Interleaved Evaluation for Retrospective Summarization and Prospective Notification on Document Streams

Xin Qian, Jimmy Lin, and Adam Roegiest
David R. Cheriton School of Computer Science
University of Waterloo, Ontario, Canada

ABSTRACT

We propose and validate a novel interleaved evaluation methodology for two complementary information seeking tasks on document streams: retrospective summarization and prospective notification. In the first, the user desires relevant and non-redundant documents that capture important aspects of an information need. In the second, the user wishes to receive timely, relevant, and non-redundant update notifications for a standing information need. Despite superficial similarities, interleaved evaluation methods for web ranking cannot be directly applied to these tasks; for example, existing techniques do not account for temporality or redundancy. Our proposed evaluation methodology consists of two components: a temporal interleaving strategy and a heuristic for credit assignment to handle redundancy. By simulating user interactions with interleaved results on submitted runs to the TREC 2014 tweet timeline generation (TTG) task and the TREC 2015 real-time filtering task, we demonstrate that our methodology yields system comparisons that accurately match the result of batch evaluations. Analysis further reveals weaknesses in current batch evaluation methodologies to suggest future directions for research.

1. INTRODUCTION

As primarily an empirical discipline, evaluation methodologies are vital to ensuring progress in information retrieval. The ability to compare system variants and detect differences in effectiveness allows researchers and practitioners to continually advance the state of the art. One such approach, broadly applicable to any online service, is the traditional A/B test [12]. In its basic setup, users are divided into disjoint "buckets" and exposed to different treatments (e.g., algorithm variants); user behavior (e.g., clicks) in each of the conditions is measured and compared to assess the relative effectiveness of the treatments. As an alternative, information retrieval researchers have developed an evaluation methodology for web search based on *interleaving* results from two different comparison systems into a single ranked list [8, 17, 6, 15, 7, 3, 16, 19], as well as recent extensions to more than two system [20]. Instead of dividing the user population into disjoint segments, all test subjects are exposed to these interleaved results. Based on user clicks, it is possible to assess the relative effectiveness of the two input systems with greater sensitivity than traditional A/B testing [17, 3], primarily due to the within-subjects design.

This paper explores interleaved evaluation for information seeking on document streams. Although we focus on a stream of social media updates (tweets), nothing in our formulation is specific to tweets. In this context, we tackle two complementary user tasks: In the retrospective summarization scenario, which is operationalized in the tweet timeline generation (TTG) task at TREC 2014 [13], the user desires relevant and non-redundant posts that capture key aspects of an information need. In the prospective notification scenario, operationalized in the real-time filtering task ("scenario A") at TREC 2015 [14], the user wishes to receive timely, relevant, and non-redundant updates (e.g., via a push notification on a mobile phone).

The contribution of this paper is the development and validation of an interleaved evaluation methodology for retrospective summarization and prospective notification on document streams, consisting of two components: a temporal interleaving strategy and a heuristic for credit assignment to handle redundancy. Although we can draw inspiration from the literature on interleaved evaluations for web search, previous techniques are not directly applicable to our tasks. We face a number of challenges: the important role that time plays in organizing and structuring system output, differing volumes in the number of results generated by systems, and notions of redundancy that complicate credit assignment. Our evaluation methodology addresses these complexities and is validated using data from the TREC 2014 and 2015 Microblog evaluations. Specifically, we simulate user interactions with interleaved results to produce a decision on whether system A is better than system B, and correlate these decisions with the results of batch evaluations. We find that our methodology yields accurate system comparisons under a variety of settings. Analysis also reveals weaknesses in current batch evaluation methodologies, which is a secondary contribution of this work.

2. BACKGROUND AND RELATED WORK

We begin by describing our task models, which are illustrated in Figure 1. We assume the existence of a stream of timestamped documents: examples include news articles coming off an RSS feed or social media posts such as tweets.

SIGIR '16, July 17 - 21, 2016, Pisa, Italy

ISBN 978-1-4503-4069-4/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2911451.2911494

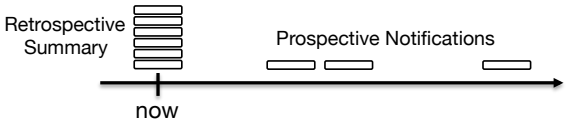

Figure 1: Illustration of our task models. At some point in time ("now"), the user develops an information need: she requests a retrospective summary of what has happened thus far and desires prospective notifications of future updates.

In this context, we consider a pair of complementary tasks: suppose at some point in time the user develops an information need, let's say, about an ongoing political scandal. She would like a retrospective summary of what has occurred up until now, which might consist of a list of chronologically-ordered documents that highlight important developments. Once she has "come up to speed", the user might wish to receive prospective notifications (on her mobile phone) regarding future updates, for example, statements by the involved parties or the emergence of another victim. Retrospective summarization and prospective notification form two complementary components of information seeking on document streams. In both cases, users desire relevant and novel (non-redundant) content—they, for example, would not want to see multiple tweets that say essentially the same thing. In the prospective notification case, the user additionally desires timely updates—as close as possible to the actual occurrence of the "new development". This, however, isn't particularly important for the retrospective case, since the events have already taken place.

In this work, we present and evaluate an interleaved evaluation methodology for the retrospective summarization and prospective notification tasks described above. Although there has been substantial work on interleaved evaluation in the context of web search [8, 17, 6, 15, 7, 3, 16, 19], we face three main challenges:

1. Temporality plays an important role in our tasks. In web search, ranked lists from different systems can be arbitrarily interleaved (and in some cases the relative ordering of documents swapped) without significantly affecting users' interpretation of the results. In our task, however, the temporal ordering of documents is critical for the proper interpretation of system output.
2. We need to interleave results of different lengths. In web search, most interleaving strategies assume ranked lists of equal length, while this is not true in our case—some systems are more verbose than others.
3. We need to account for redundancy. In our tasks the notion of novelty is very important and "credit" is only awarded for returning non-redundant tweets. This creates a coupling effect between two systems where one's result might "mask" the novelty in the other. That is, a system's output becomes redundant only because the interleaving algorithm injected a relevant document before the document in question.

Nevertheless, there is a rich body of literature from which we can draw inspiration. In particular, we employ a simulation-based approach that is well-established for validating interleaved evaluations [6, 7, 16].

Our retrospective summarization and prospective notification tasks are grounded in the Microblog track evaluations at TREC: specifically, the tweet timeline generation (TTG) task at TREC 2014 [13] and the push notification scenario ("scenario A") in the real-time filtering task at TREC 2015 [14]. Although there has been a substantial amount of work on developing systems that try to accomplish the tasks we study (see the TREC overview papers for pointers into the literature), our focus is not on the development of algorithms, but rather in evaluating system output. We adopt the framework provided by these tracks: relevance judgments and submitted runs are used in simulation studies to validate our interleaved evaluation methodology.

3. TASK AND METRICS

We begin by describing evaluations from TREC 2014 and 2015 that operationalize our retrospective summarization and prospective notification tasks.

3.1 Retrospective Summarization

Tweet timeline generation (TTG) was introduced at the TREC 2014 Microblog track. The putative user model is as follows: "At time T, I have an information need expressed by query Q, and I would like a summary that captures relevant information." The system's task is to produce a summary timeline, operationalized as a list of non-redundant, chronologically ordered tweets. It is imagined that the user would consume the entire summary (unlike a ranked list, where the user might stop reading at any time).

Redundancy was operationalized as follows: for every pair of tweets, if the chronologically later tweet contains substantive information that is not present in the earlier tweet, the later tweet is considered novel; otherwise, the later tweet is redundant with respect to the earlier one. Thus, redundancy and novelty are antonyms; we use them interchangeably in opposite contexts. Due to the temporal constraint, redundancy is *not* symmetric. If tweet A precedes tweet B and tweet B contains substantively similar information found in tweet A, then B is redundant with respect to A, but not the other way around. The task also assumes transitivity. Suppose A precedes B and B precedes C: if B is redundant with respect to A and C is redundant with respect to B, then by definition C is redundant with respect to A.

The TTG assessment task can be viewed as semantic clustering—that is, we wish to group relevant tweets into clusters in which all tweets share substantively similar information. Within each cluster, the earliest tweet is novel; all other tweets in the cluster are redundant with respect to all earlier tweets. The track organizers devised a two-phase assessment workflow that implements this idea. In the first phase, all tweets are pooled and judged for relevance. In the second phase, relevant tweets for each topic are then clustered. We refer the reader to previous papers for more details [13, 22], but the final product of the human annotation process is a list of tweet clusters, each containing tweets that represent a semantic equivalence class.

In TREC 2014, TTG systems were evaluated in terms of set-based metrics (precision, recall, and F-score) at the cluster level. Systems only received credit for returning one tweet from each cluster—that is, once a tweet is retrieved, all other tweets in the cluster are automatically considered not relevant. In this study, we performed our correlation analysis against recall, for reasons that will become apparent later. The track evaluated recall in two different ways: unweighted and weighted. In the relevance assessment pro-

cess, tweets were judged as not relevant, relevant, or highly relevant. For unweighted recall (also called S-recall [23] and I-recall [18]), relevant and highly-relevant tweets were collapsed to yield binary judgments and all clusters received equal weight. For weighted recall, each cluster is assigned a weight proportional to the sum of relevance grades from every tweet in the cluster (relevant tweets receive a weight of one and highly-relevant tweets receive a weight of two).

3.2 Prospective Notification

In the real-time filtering task at TREC 2015 [14], the goal is for a system to identify interesting and novel content for a user in a timely fashion, with respect to information needs (called "interest profiles" but in actuality quite similar to traditional *ad hoc* topics). In the push notification variant of the task ("scenario A"), updates are putatively delivered in real time as notifications to users' mobile phones. A system was allowed to return a maximum of ten tweets per day per interest profile. The official evaluation took place over a span of ten days during July 2015, where all participating systems "listened" to Twitter's live tweet sample stream to complete the evaluation task; the interest profiles were made available prior to the evaluation period.

The assessment workflow was the same as the TTG task in TREC 2014 (see Section 3.1): relevance assessment using traditional pooling followed by semantic clustering. The task likewise used three-way judgments: not relevant, relevant, and highly relevant. We refer the reader to the TREC overview paper for more details [14].

The two metrics used to evaluate system runs were expected latency-discounted gain (ELG) and normalized cumulative gain (nCG). These two metrics are computed for each interest profile for each day in the evaluation period (explained in detail below). The final score of a run is the average of daily scores across all interest profiles.

The expected latency-discounted gain (ELG) metric was adapted from the TREC temporal summarization track [2]:

$$\frac{1}{N}\sum \mathrm{G}(t) \tag{1}$$

where N is the number of tweets returned and $\mathrm{G}(t)$ is the gain of each tweet: not relevant tweets receive a gain of 0, relevant tweets receive a gain of 0.5, and highly-relevant tweets receive a gain of 1.0.

As with the TTG task, redundancy is penalized: a system only receives credit for returning one tweet from each cluster. Furthermore, per the track guidelines, a latency penalty is applied to all tweets, computed as $\mathrm{MAX}(0, (100 - d)/100)$, where the delay d is the time elapsed (in minutes, rounded down) between the tweet creation time and the putative time the tweet was delivered. That is, if the system delivers a relevant tweet within a minute of the tweet being posted, the system receives full credit. Otherwise, credit decays linearly such that after 100 minutes, the system receives no credit even if the tweet was relevant.

The second metric is normalized cumulative gain (nCG):

$$\frac{1}{\mathcal{Z}}\sum \mathrm{G}(t) \tag{2}$$

where $\mathcal{Z}$ is the maximum possible gain (given the ten tweet per day limit). The gain of each individual tweet is computed as above (with the latency penalty). Note that gain is not discounted (as in nDCG) because the notion of document ranks is not meaningful in this context.

Due to the setup of the task and the nature of interest profiles, it is possible (and indeed observed empirically) that for some days, no relevant tweets appear in the judgment pool. In terms of evaluation metrics, a system should be rewarded for correctly identifying these cases and not generating any output. We can break down the scoring contingency table as follows: If there are relevant tweets for a particular day, scores are computed per above. If there are no relevant tweets for that day, and the system returns zero tweets, it receives a score of one (i.e., perfect score) for that day; otherwise, the system receives a score of zero for that day. This means that an empty run (a system that never returns anything) may have a non-zero score.

4. INTERLEAVING METHODOLOGY

The development of an interleaved evaluation methodology requires answering the following questions:

1. How exactly do we interleave the output of two systems into one single output, in light of the challenges discussed in Section 2?
2. How do we assign credit to each of the underlying systems in response to user interactions with the interleaved results?

4.1 Interleaving Strategy

We begin by explaining why existing interleaving strategies for web search cannot be applied to either retrospective summarization or prospective notification. Existing strategies attempt to draw results from the test systems in a "fair" way: In balanced interleaving [8], for example, the algorithm maintains two pointers, one to each input list, and draws from the lagging pointer. Team drafting [17], on the other hand, follows the analogy of selecting teams for a friendly team-sports match and proceeds in rounds. Both explicitly assume (1) that the ranked lists from each system are ordered in decreasing probability of relevance (i.e., following the probability ranking principle) and (2) that the ranked lists are of equal length. Both assumptions are problematic because output in retrospective summarization and prospective notification must be chronologically ordered: a naïve application of an existing web interleaving strategy in the retrospective case would yield a chronologically jumbled list of tweets that is not interpretable. In the prospective case, we cannot "time travel" and push notifications "in the future" and then "return to the past". Furthermore, in both our tasks system outputs can vary greatly in verbosity, and hence the length of their results. This is an important aspect of the evaluation design as systems should learn when to "keep quiet" (see Section 3.2). Most existing interleaving strategies don't tell us what to do when we run out of results from one system. For these reasons it is necessary to develop a new interleaving strategy.

After preliminary exploration, we developed an interleaving strategy, called *temporal interleaving*, where we simply interleave the two runs by time. The strategy is easy to implement yet effective, as we demonstrate experimentally. Temporal interleaving works in the prospective case because time is always "moving forward". An example is shown in Figure 2, where we have system A on the left and system B on the right. The subscript of each tweet indicates its timestamp and the interleaved result is shown in the middle (note that tweet t_{28} is returned by both systems). One potential

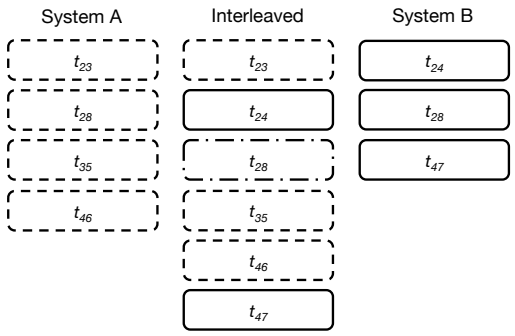

Figure 2: Illustration of temporal interleaving. Note that tweet t_{28} is returned by both systems.

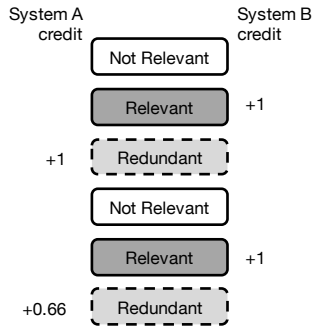

Figure 3: Example of interleaving credit assignment and redundancy handling.

downside of this strategy is that all retrieved documents from both systems are included in the interleaved results, which increases its length—we return to address this issue in Section 5.3.

Our simple temporal interleaving strategy works as is for TTG runs, since system outputs are ordered lists of tweets. For push notifications, there is one additional wrinkle: which timestamp do we use? Recall that in prospective notification there is the tweet creation time and the push time (when the system identified the tweet as being relevant). We base interleaving on the push time because it yields a very simple implementation: we watch the output of two prospective notification systems and take the output as soon as a result is emitted by either system. However, we make sure to apply de-duplication: if a tweet is pushed by two systems but at different times, it will only be included once in the interleaved results.

4.2 User Interactions and Credit Assignment

In our interleaved evaluation methodology, output from the two different test systems are combined using the temporal interleaving strategy described above and presented to the user. We assume a very simple interaction model in which the user goes through the output (in chronological order) from earliest to latest and makes one of three judgments for each tweet: not relevant, relevant, and relevant but redundant (i.e., the tweet is relevant but repeats information that is already present in a previously-seen tweet). This extends straightforwardly to cases where we have graded relevance judgments: for the relevant and redundant judgments, the user also indicates the relevance grade. In retrospective summarization, the user is interacting with static system output, but in the prospective notification case, output is presented to the user over a period of time. This is called the "simple task", for reasons that will become clear shortly. We assume that users provide explicit judgments, in contrast to implicit feedback (i.e., click data) in the case of interleaved evaluations for web search; we return to discuss this issue in Section 5.4.

Based on user interactions with the interleaved results, we must now assign credit to each of the test systems, which is used to determine their relative effectiveness. Credit assignment for the relevant label is straightforward: credit accrues to the system that contributed the tweet to the interleaved results (or to both if both systems returned the tweet). However, credit assignment for a tweet marked redundant is more complex—we do not know, for example, if the redundancy was actually introduced by the interleaving. That is, the interleaving process inserted a tweet (from the other run) before this particular tweet that made it redundant.

We can illustrate this with the diagram in Figure 3. A dotted border represents a tweet contributed by system A (on the left) and a solid border represents a tweet contributed by system B (on the right). Suppose the assessor judged the tweets as they are labeled in the figure. The second and fifth tweets are marked relevant, and so system B gets full credit twice. Now let's take a look at the third tweet, contributed by system A, which is marked redundant—we can confidently conclude in this case that the redundancy was introduced by the interleaving, since there are no relevant tweets above that are contributed by system A. Therefore, we can give system A full credit for the third tweet. Now let's take a look at the sixth tweet: generalizing this line of reasoning, the more that relevant tweets above are from system B, the more likely that we're encountering a "masking effect" (all things being equal), where the redundancy is an artifact of the interleaving itself. To capture this, we introduce the following heuristic: the amount of credit given to a system for a tweet marked redundant is multiplied by a discount factor equal to the fraction of relevant and redundant tweets above that come from the *other* system. In this case, there are two relevant tweets above, both from system B, and one redundant tweet from system A, so system A receives a credit of 0.66.

More formally, consider an interleaved result S consisting of tweets $s_1 \ldots s_n$ drawn from system A and system B. We denote S_A and S_B as those tweets in S that come from system A and B, respectively. For a tweet s_i judged redundant, if $s_i \in S_A$, then we multiple its gain by a discount factor D_A as follows:

$$D_A(s_i) = \frac{|\{s_j | j < i \wedge I(s_j) \wedge s_j \in S_B\}|}{T(s_i)} \tag{3}$$

$$T(s_i) = |\{s_j | j < i \wedge I(s_j) \wedge s_j \in S_A\}| + |\{s_j | j < i \wedge I(s_j) \wedge s_j \in S_B\}| \tag{4}$$

where $I(s)$ is an indicator function that returns one if the user (previously) judged the tweet to be either relevant or redundant, or zero otherwise. On the other hand, if $s_i \in S_B$, we apply a discount factor D_B that mirrors D_A above (i.e., flipping subscripts A and B). If s_i is both in S_A and S_B, we apply *both* equations and give each system a different amount of credit (summing up to one).

We emphasize, of course, that this way of assigning credit for redundant judgments is a heuristic (but effective, from

our evaluations). For further validation, we introduce an alternative interaction model that we call the "complex task": in this model, the user still marks each tweet not relevant, relevant, and redundant, but for each redundant tweet, the user marks the source of the redundancy, i.e., which previous tweet contains the same information. With this additional source of information, we can pinpoint the exact source of redundancy and assign credit definitively (zero if the source of redundancy was from the same run, and one if from the other run). Of course, such a task would be significantly more onerous (and slower) than just providing three-way judgments, but this "complex task" provides an upper bound that allows us to assess the effectiveness of our credit assignment heuristic.

One final detail: In the prospective task, we still apply a latency penalty to the assigned credit, as in ELG. Thus, in the case of a tweet that was pushed by both systems, but at different times, they will receive different amounts of credit. In the interleaved results, of course, the tweet will appear only once—from that single judgment we can compute the credit assigned to each system.

To recap: we have presented a temporal interleaving strategy to combine system output, introduced a model for how users interact with the results, and devised a credit assignment algorithm (including redundancy handling) that scores the systems based on user interactions. From this, we arrive at a determination of which system is more effective. Do these decisions agree with the results of batch evaluations? We answer this question with simulation studies based on runs submitted to TREC 2014 (for retrospective summarization) and TREC 2015 (for prospective notification).

5. SIMULATION RESULTS

5.1 Retrospective Summarization

To validate our interleaved evaluation methodology for retrospective summarization, we conducted user simulations using runs from the TREC 2014 TTG task. In total, 13 groups submitted 50 runs to the official evaluation. For each pair of runs, we applied the temporal interleaving strategy described above and simulated user interactions with the "ground truth" cluster annotations. Each simulation experiment comprised 67,375 pairwise comparisons, which we further break down into 63,415 comparisons of runs from different groups (inter-group) and 3,960 comparisons between runs from the same group (intra-group). Wang et al. [22] were able to elicit two completely independent sets of cluster annotations, which they refer to as the "official" and "alternate" judgments. Thus, we were able to simulate user interactions with both sets of clusters.

First, we ran simulations using binary relevance judgments. Results are shown in Table 1. When comparing simulation results (which system is better, based on assigned credit) with the batch evaluation results (unweighted recall), there are four possible cases:

- The compared runs have different batch evaluation results and the simulation was able to detect those differences; denoted (Agree, Δ).
- The compared runs have the same batch result and the simulation assigned equal credit to both runs; denoted (Agree, $\neg\Delta$).
- The compared runs have different batch evaluation results but the simulation was not able to detect those differences; denoted (Disagree, Δ).
- The compared runs have the same batch result and the simulation falsely ascribed differences in effectiveness between those runs; denoted (Disagree, $\neg\Delta$).

In the first two cases, the batch evaluation and interleaved evaluation results are consistent and the interleaving can be said to have given "correct" results; this is tallied up as (Agree, Total) in the results table. In the last two cases, the batch evaluation and interleaved evaluation results are inconsistent and the interleaving can be said to have given "incorrect" results; this is tallied up as (Disagree, Total) in the results table.[1]

With "official" and "alternate" clusters, there are four ways we can run the simulations: simulate with official judgments, correlate with batch evaluation results using official judgments (official, official); simulate with alternate judgments, correlate with batch evaluation results using alternative judgments (alternate, alternate); as well as the symmetrical cases where the simulation and batch evaluations are different, i.e., (official, alternate) and (alternate, official). Table 1 shows all four cases, denoted by the first two columns. Finally, the two vertical blocks of the table denote the results of the "simple task" (simulated user provides three-way judgments) and the "complex task" (simulated user additionally marks the source of a redundant tweet).

There is a lot of information to unpack from Table 1. Focusing only on "all pairs" with the "simple task", we see that our simulation results agree with batch evaluation results 92%-93% of the time, which indicates that our interleaved evaluation methodology is effective. The inaccuracies can be attributed to the credit assignment heuristic for redundant labels—this can be seen from the "complex task" block, where accuracy becomes 100% if we ask the (simulated) user to mark the source of the redundancy. Of course, this makes the task unrealistically onerous, so we argue that our credit assignment heuristic strikes the right balance between accuracy and complexity.

With the (official, official) and the (alternate, alternate) conditions, we are simulating user interactions and computing batch results with the same cluster assignments. With the other two conditions, we simulate with one set of clusters and perform batch evaluations with the other—the difference between these two sets quantifies inter-assessor differences. Results suggest that the effect of using different assessors is relatively small—this finding is consistent with that of Wang et al. [22], who confirmed the stability of the TTG evaluation with respect to assessor differences.

The inter-group and intra-group comparisons suggest how well our interleaved evaluation methodology would fare under slightly different conditions. Runs by the same group (intra-group) often share similar algorithms (perhaps varying in parameters), which often yield runs that are similar in effectiveness (or the same). This makes differences more difficult to detect, and indeed, Table 1 shows this to be the

[1] Methodologically, our approach differs from many previous studies that take advantage of click data. For example, Chapelle et al. [3] studied only a handful of systems (far fewer than here) but across far more queries, and hence are able to answer certain types of questions that we cannot. Also, most previous studies do not consider system ties, with He et al. [6] being an exception, but they do not explicitly break out the possible contingencies as we do here.

Simulation	Judgment	Simple Task						Complex Task					
		Agree			Disagree			Agree			Disagree		
		Δ	¬Δ	Total	Δ	¬Δ	Total	Δ	¬Δ	Total	Δ	¬Δ	Total
All Pairs													
official	official	89.6%	3.7%	93.3%	3.0%	3.7%	6.7%	92.6%	7.4%	100.0%	0	0	0
alternate	alternate	88.8%	3.6%	92.4%	3.5%	4.1%	7.6%	92.3%	7.7%	100.0%	0	0	0
official	alternate	88.4%	3.6%	92.0%	3.8%	4.2%	8.0%	89.2%	5.6%	94.8%	3.1%	2.1%	5.2%
alternate	official	88.7%	3.5%	92.2%	3.9%	3.9%	7.8%	89.2%	5.6%	94.8%	3.4%	1.8%	5.2%
Inter-Group Pairs Only													
official	official	91.1%	2.8%	93.9%	2.8%	3.3%	6.1%	93.9%	6.1%	100.0%	0	0	0
alternate	alternate	90.3%	2.7%	93.0%	3.3%	3.7%	7.0%	93.6%	6.4%	100.0%	0	0	0
official	alternate	89.9%	2.7%	92.6%	3.6%	3.8%	7.4%	90.7%	4.4%	95.1%	2.9%	2.0%	4.9%
alternate	official	90.2%	2.6%	92.8%	3.7%	3.5%	7.2%	90.7%	4.4%	95.1%	3.2%	1.7%	4.9%
Intra-Group Pairs Only													
official	official	65.8%	18.1%	83.9%	5.8%	10.3%	16.1%	71.6%	28.4%	100.0%	0	0	0
alternate	alternate	65.1%	17.8%	82.9%	6.5%	10.6%	17.1%	71.6%	28.4%	100.0%	0	0	0
official	alternate	64.3%	17.9%	82.2%	7.3%	10.5%	17.8%	65.3%	24.0%	89.3%	6.3%	4.4%	10.7%
alternate	official	64.7%	17.6%	82.3%	7.0%	10.7%	17.7%	65.3%	24.0%	89.3%	6.3%	4.4%	10.7%

Table 1: TTG simulation results for both the "simple task" and the "complex task". The "Agree" columns give the percentages of cases where the simulation results agree with the batch evaluation results, when the runs actually differ (Δ), and when the runs don't differ (¬Δ). The "Disagree" columns give the percentages of cases where the simulation results disagree with the batch evaluation results, when the runs actually differ (Δ), and when the runs don't differ (¬Δ).

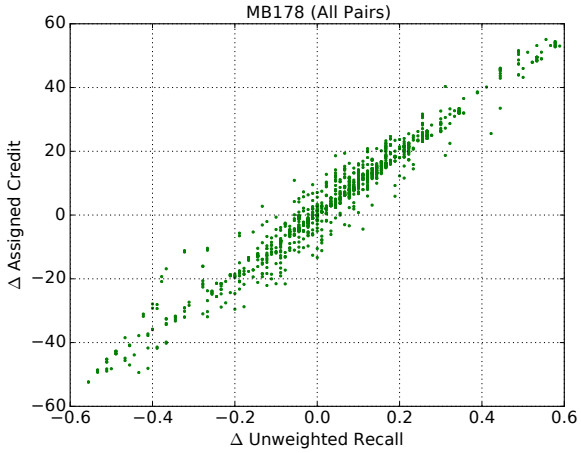

Figure 4: Scatterplot showing batch vs. simulation results for topic MB178.

case (lower agreement). In contrast, differences in effectiveness in runs between groups (inter-group) are slightly easier to detect, as shown by the slightly higher agreement.

To help further visualize our findings, a scatterplot of simulation results is presented in Figure 4 for a representative topic, MB 178, under the all-pairs, (official, official) condition. Each point represents a trial of the simulation comparing a pair of runs: the x coordinate denotes the difference based on the batch evaluation, and the y coordinate denotes the difference in assigned credit based on the simulation. We see that there is a strong correlation between simulation and batch results. Plots from other topics look very similar, except differing in the slope of the trendline (since credit is not normalized, but recall is).

The previous results did not incorporate graded relevance judgments. Our next set of experiments examined this refinement: relevant tweets receive a credit of one and highly-relevant tweets receive a credit of two. The simulated user now indicates the relevance grade for the relevant and redundant cases. There is, however, the question of which batch metric to use: the official TREC evaluation used weighted recall, where the weight of each cluster was proportional to the sum of the relevance grades of tweets in the cluster. This encodes the simple heuristic that "more discussed facets are more important", which seems reasonable, but Wang et al. [22] found that this metric correlated poorly with human preferences, suggesting that cluster size is perhaps not a good measure of importance. We ran simulations correlating against official weighted recall: the results were slightly worse than those in Table 1, but still quite good. For example, we achieved 90% accuracy in the (official, official) condition on the simple task, as opposed to 93%.

However, given the findings of Wang et al., these simulations might not be particularly meaningful. As an alternative, we propose a slightly different approach to computing the cluster weights: instead of the sum of relevance grades of tweets in the cluster, we use the highest relevance grade of tweets in the cluster. That is, if a cluster contains a highly-relevant tweet, it receives a weight of two; otherwise, it receives a weight of one. This weighting scheme has the effect that scores are not dominated by huge clusters. The results of these simulations are shown in Table 2.

From these experiments, we see that accuracy remains quite good, suggesting that our interleaved evaluation methodology is able to take advantage of graded relevance judgments. One important lesson here is that capturing "cluster importance" in TTG is a difficult task, and that it is unclear if present batch evaluations present a reasonable solution. Without a well-justified batch evaluation metric, we lack values against which to correlate our simulation outputs. Thus, these results reveal a weakness in current batch evaluations (indicating avenues of future inquiry), as opposed to a flaw in our interleaved evaluation methodology.

5.2 Prospective Notification

For prospective notification, we validated our interleaved evaluation methodology using runs submitted to the TREC 2015 real-time filtering task ("scenario A"). In total, there

		Simple Task						Complex Task					
		Agree			Disagree			Agree			Disagree		
Simulation	Judgment	Δ	¬Δ	Total	Δ	¬Δ	Total	Δ	¬Δ	Total	Δ	¬Δ	Total
All Pairs													
official	official	89.9%	3.1%	93.0%	4.1%	2.9%	7.0%	93.3%	5.5%	98.8%	0.7%	0.5%	1.2%
alternate	alternate	89.0%	3.0%	92.0%	4.7%	3.3%	8.0%	92.9%	5.7%	98.6%	0.8%	0.6%	1.4%
official	alternate	88.6%	3.0%	91.6%	5.1%	3.3%	8.4%	89.9%	4.4%	94.3%	3.8%	1.9%	5.7%
alternate	official	88.7%	3.0%	91.7%	1.3%	3.0%	8.3%	90.0%	4.4%	94.4%	4.0%	1.6%	5.6%
Inter-Group Pairs Only													
official	official	91.3%	2.2%	93.5%	3.9%	2.6%	6.5%	94.6%	4.2%	98.9%	0.6%	0.5%	1.1%
alternate	alternate	90.4%	2.2%	92.6%	4.5%	2.9%	7.4%	94.2%	4.5%	98.7%	0.7%	0.6%	1.3%
official	alternate	90.0%	2.1%	92.1%	4.9%	3.0%	7.9%	91.4%	3.3%	94.7%	3.5%	1.8%	5.3%
alternate	official	90.2%	2.1%	92.3%	5.0%	2.7%	7.7%	91.5%	3.3%	94.7%	3.8%	1.5%	5.3%
Intra-Group Pairs Only													
official	official	66.5%	17.2%	83.7%	7.4%	8.9%	16.3%	72.5%	25.0%	97.5%	1.4%	1.1%	2.5%
alternate	alternate	65.8%	16.8%	82.6%	8.2%	9.2%	17.4%	72.6%	25.0%	97.6%	1.4%	1.0%	2.4%
official	alternate	64.7%	17.0%	81.7%	9.3%	9.0%	18.3%	66.3%	21.9%	88.2%	7.7%	4.1%	11.8%
alternate	official	65.3%	16.9%	82.2%	8.6%	9.2%	17.8%	66.3%	22.0%	88.3%	7.6%	4.1%	11.7%

Table 2: TTG simulation results with graded relevance judgments, organized in the same manner as Table 1.

were 37 runs from 14 groups submitted to the official evaluation. This yields a total of 33,966 pairwise comparisons; 32,283 inter-group pairs and 1,683 intra-group pairs.

Simulation results (with graded relevance judgments) are shown in Table 3 for correlations against ELG and in Table 4 for correlations against nCG. The table is organized in the same manner as Tables 1 and 2, with the exception that we only have one set of cluster annotations available, so no "official" vs. "alternate" distinction.

Results of the simulation, shown under the rows marked retaining "quiet days", are quite poor. Analysis reveals that this is due to the handling of days for which there are no relevant tweets. Note that for days without any relevant tweets, there are only two possible scores: one if the system does not return any results, and zero otherwise. Thus, for interest profiles with few relevant tweets, the score is highly dominated by these "quiet days". As a result, a system that does not return anything scores quite highly; in fact, better than most submitted runs [14]. To make matters worse, since 2015 was the first year of this TREC evaluation, systems achieved high scores by simply returning few results, in many cases for totally idiosyncratic reasons—for example, the misconfiguration of a score threshold.

This property of the official evaluation is problematic for interleaved evaluations since it is impossible to tell without future knowledge whether there are relevant tweets for a particular day. Consider the case when system A returns a tweet for a particular day and system B does not return anything, and let's assume we know (based on an oracle) that there are no relevant tweets for that day: according to our interleaved evaluation methodology, neither system would receive any credit. However, based on the batch evaluation, system B would receive a score of one for that day. There is, of course, no way to know this at evaluation time when comparing only two systems, and thus the interleaved evaluation results would disagree with the batch evaluation results. The extent of this disagreement depends on the number of days across the topics for which there were no relevant tweets. Since the interest profiles for the TREC 2015 evaluation had several quiet days each, our interleaved evaluation methodology is not particularly accurate.

We argue, however, that this is more an artifact of the current batch evaluation setup than a flaw in our interleaved evaluation methodology per se; see Tan et al. [21] for further discussion. As the track organizers themselves concede in the TREC overview paper [14], it is not entirely clear if the current handling of days with no relevant tweets is appropriate. While it is no doubt desirable that systems should learn when to "remain quiet", the current batch evaluation methodology yields results that are idiosyncratic in many cases.

To untangle the effect of these "quiet days" in our interleaved evaluation methodology, we conducted experiments where we simply discarded days in which there were no relevant tweets. That is, if an interest profile only contained three days (out of ten) that contained relevant tweets, the score of that topic is simply an average of the scores over those three days. We modified the batch evaluation scripts to also take this into account, and then reran our simulation experiments. The results are shown in Table 3 and Table 4 under the rows marked discarding "quiet days". In this variant, we see that our simulation results are quite accurate, which confirms that the poor accuracy of our initial results is attributable to days where there are no relevant tweets. Once again, this is an issue with the overall TREC evaluation methodology, rather than a flaw in our interleaving approach. These findings highlight the need for additional research on metrics that better model sparse topics. In order to remove this confound, for the remaining prospective notification experiments, we discarded the "quiet days".

Our credit assignment algorithm is recall oriented in that it tries to quantify the total amount of relevant information a user receives, and so it is perhaps not a surprise that credit correlates with nCG. However, experiments show that we also achieve good accuracy correlating with ELG (which is precision oriented). It is observed in the TREC 2015 evaluation [14] that there is reasonable correlation between systems' nCG and ELG scores. There is no principled explanation for this, as prospective notification systems could very well make different precision/recall tradeoffs. However, there is the additional constraint that systems are not allowed to push more than ten tweets per day, so that a high-

Condition	Simple Task						Complex Task					
	Agree			Disagree			Agree			Disagree		
	Δ	¬Δ	Total	Δ	¬Δ	Total	Δ	¬Δ	Total	Δ	¬Δ	Total
Retaining "quiet days"												
All Pairs	45.6%	16.6%	62.2%	37.7%	0.1%	37.8%	45.6%	16.6%	62.2%	37.6%	0.2%	37.8%
Inter-Group Pairs	46.4%	15.3%	61.7%	38.2%	0.1%	38.3%	46.5%	15.3%	61.8%	38.1%	0.1%	38.2%
Intra-Group Pairs	29.0%	41.4%	70.4%	29.1%	0.5%	29.6%	29.5%	41.7%	71.2%	28.6%	0.2%	28.8%
Discarding "quiet days"												
All Pairs	56.7%	34.8%	91.5%	8.4%	0.1%	8.5%	56.8%	34.8%	91.6%	8.3%	0.1%	8.4%
Inter-Group Pairs	58.0%	33.9%	91.9%	8.0%	0.1%	8.1%	58.1%	33.9%	92.0%	7.9%	0.1%	8.0%
Intra-Group Pairs	32.0%	52.7%	84.7%	14.8%	0.5%	15.3%	32.5%	53.0%	85.5%	14.3%	0.2%	14.5%

Table 3: Results of push notification simulations, correlating against ELG.

Condition	Simple Task						Complex Task					
	Agree			Disagree			Agree			Disagree		
	Δ	¬Δ	Total	Δ	¬Δ	Total	Δ	¬Δ	Total	Δ	¬Δ	Total
Retaining "quiet days"												
All Pairs	52.8%	16.7%	69.5%	30.0%	0.5%	30.5%	52.8%	16.8%	69.6%	30.0%	0.4%	30.4%
Inter-Group Pairs	53.5%	15.3%	68.8%	30.7%	0.5%	31.2%	53.6%	15.4%	69.0%	30.6%	0.4%	31.0%
Intra-Group Pairs	38.3%	43.8%	82.1%	17.3%	0.6%	17.9%	37.8%	44.1%	81.9%	17.8%	0.3%	18.1%
Discarding "quiet days"												
All Pairs	62.5%	35.2%	97.7%	2.1%	0.2%	2.3%	62.7%	35.3%	98.0%	2.0%	0	2.0%
Inter-Group Pairs	63.7%	34.1%	97.8%	2.1%	0.1%	2.2%	63.8%	34.3%	98.1%	1.9%	0	1.9%
Intra-Group Pairs	41.2%	55.4%	96.6%	2.9%	0.5%	3.4%	40.9%	55.8%	96.7%	3.3%	0	3.3%

Table 4: Results of push notification simulations, correlating against nCG.

	Summarization	Notification
All Pairs	92.8%	96.9%
Inter-Group Pairs	94.0%	97.9%
Intra-Group Pairs	73.7%	76.5%

Table 5: Lengths of interleaved results as a percentage of the sum of the lengths of the individual runs.

volume low-precision system would quickly use up its "daily quota". Additionally, we suspect that since TREC 2015 represented the first large-scale evaluation of this task, teams have not fully explored the design space.

5.3 Assessor Effort: Output Length

We next turn our attention to two issues related to assessor effort: the length of the interleaved system output (this subsection) and the effort involved in providing explicit judgments in our interaction model (next subsection).

One downside of our temporal interleaving strategy is that the interleaved results are longer than the individual system outputs. Exactly how much longer is shown in Table 5, where the lengths of the interleaved results are shown as a percentage of the sum of the lengths of the individual runs. The lengths are not 100% because the individual system outputs may contain overlap, and comparisons between runs from the same group contain more overlap. Nevertheless, we can see that temporal interleaving produces output that is substantially longer than each of the individual system outputs. This is problematic for two reasons: first, it means a substantial increase in evaluation effort, and second, the interleaving produces a different user experience in terms of the verbosity of the system.

There is, however, a simple solution to this issue: after temporal interleaving, for each result we flip a biased coin and retain it with probability p. That is, we simply decide to discard some fraction of the results. Figure 5 shows the results of these experiments. On the x axis we sweep across p, the retention probability, and on the y axis we plot the simulation accuracy (i.e., agreement between simulation credit and batch results). The left plot shows the results for retrospective summarization using unweighted recall and the (official, official) condition; the rest of the graphs look similar and so we omit them for brevity. In the middle plot, we show accuracy against ELG for prospective notification and against nCG on the right (both discarding quiet days). Since there is randomness associated with these simulations, the plots represent averages over three trials.

We see that simulation results remain quite accurate even if we discard a relatively large fraction of system output. For the prospective task, accuracy is higher for lower p values because there are many intra-group ties. At $p = 0$, accuracy is simply the fraction of "no difference" comparisons. Based on these results, an experiment designer can select a desired tradeoff between accuracy and verbosity. With p around 0.5 to 0.6, we obtain an interleaved result that is roughly the same length as the source systems—and in that region we still achieve good prediction accuracy. It is even possible to generate interleaved results that are shorter than the input runs. Overall, we believe that this simple approach adequately addresses the length issue.

5.4 Assessor Effort: Explicit Judgments

Another potential objection to our interleaved evaluation methodology is that our interaction model depends on explicit judgments for credit assignment, as opposed to implicit judgments (i.e., clicks) in the case of interleaved evaluations for web ranking. This issue warrants some discussion, because the ability to gather implicit judgments based on be-

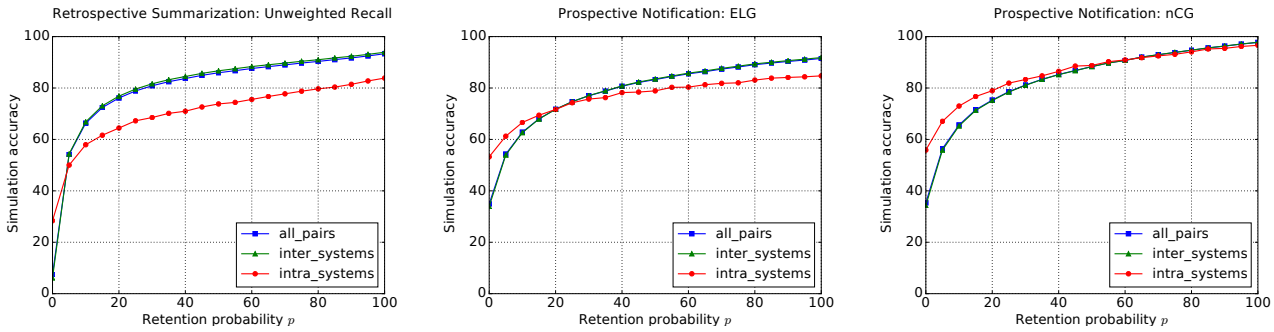

Figure 5: Simulation accuracy as a function of retention probability p for unweighted recall on retrospective summarization (left); ELG (middle) and nCG (right) for prospective summarization.

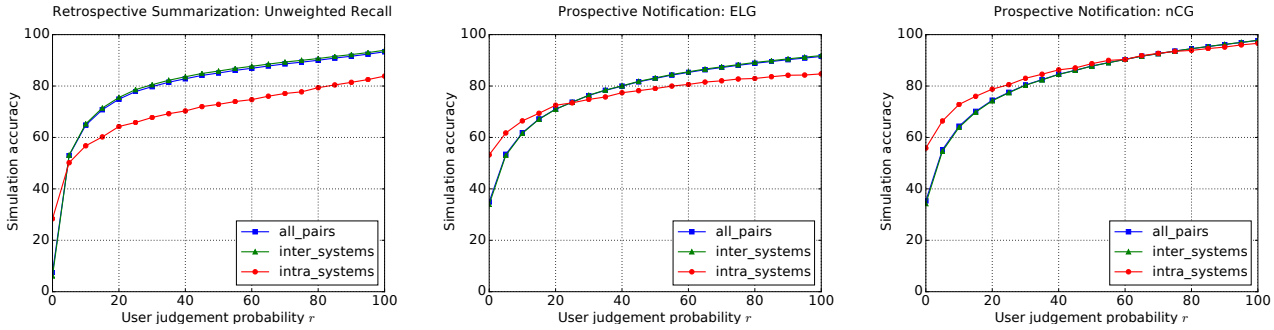

Figure 6: Simulation accuracy as a function of user judgment probability r for unweighted recall on retrospective summarization (left); ELG (middle) and nCG (right) for prospective summarization.

havioral data greatly expands the volume of feedback we can easily obtain (e.g., from log data).

We have two responses: First, it is premature to explore implicit interactions for our tasks. For web search, there is a large body of work spanning over two decades that has validated the interpretation of click data for web ranking preferences—including the development of click models [1, 4], eye-tracking studies [5, 9], extensive user studies [10], and much more [11]. In short, we have a pretty good idea of how users interact with web search results, which justifies the interpretation of click data. None of this exists for retrospective summarization and prospective notification. Furthermore, interactions with tweets in our case are more complex: some tweets have embedded videos, images, or links. There are many different types of clicks: the user can "expand" a tweet, thereby showing details of the embedded object and from there take additional actions, e.g., play the embedded video directly, click on the link to navigate away from the result, etc. Not taking any overt action on a tweet doesn't necessary mean that the tweet is not relevant—the succinct nature of tweets means that relevant information can be quickly absorbed, perhaps without leaving any behavioral trails. Thus, any model of implicit interactions we could develop at this point would lack empirical grounding. More research is necessary to better understand how users interact with retrospective summarization and prospective notification systems. With a better understanding, we can then compare models of implicit feedback with the explicit feedback results presented here.

Our second response argues that in the case of prospective notifications, an explicit feedback model might not actually be unrealistic. Recall that such updates are putatively delivered via mobile phone notifications, and as such, they are presented one at a time to the user—depending on the user's settings, each notification may be accompanied by an auditory or physical cue (a chime or a vibration) to attract the user's attention. In most implementations today the notification can be dismissed by the user or the user can take additional action (e.g., click on the notification to open the mobile app). These are already quite explicit actions with relatively clear user intent—it is not far-fetched to imagine that these interactions can be further refined to provide explicit judgments without degrading the user experience.

Nevertheless, the issue of assessor effort in providing explicit judgments is still a valid concern. However, we can potentially address this issue in the same way as the length issue discussed above. Let us assume that the user provides interaction data with probability r. That is, as we run the simulation, we flip a biased coin and observe each judgment with only probability r. In the prospective notification case, we argue that this is not unrealistic—the user "pays attention" to the notification message with probability r; the rest of the time, the user ignores the update.

Figure 6 shows the results of these experiments (averaged over three trials). On the x axis we sweep across r, the interaction probability and on the y axis we plot the simulation accuracy. The left plot shows the results for retrospective summarization using unweighted recall and the (official, official) condition. In the middle plot, we show accuracy against ELG for prospective notification and against nCG on the right (once again, discarding quiet days in both cases). Experiments show that we are able to accurately decide the relative effectiveness of the comparison systems even with limited user interactions.

The next obvious question, of course, is what if we combined both the length analysis and interaction probabil-

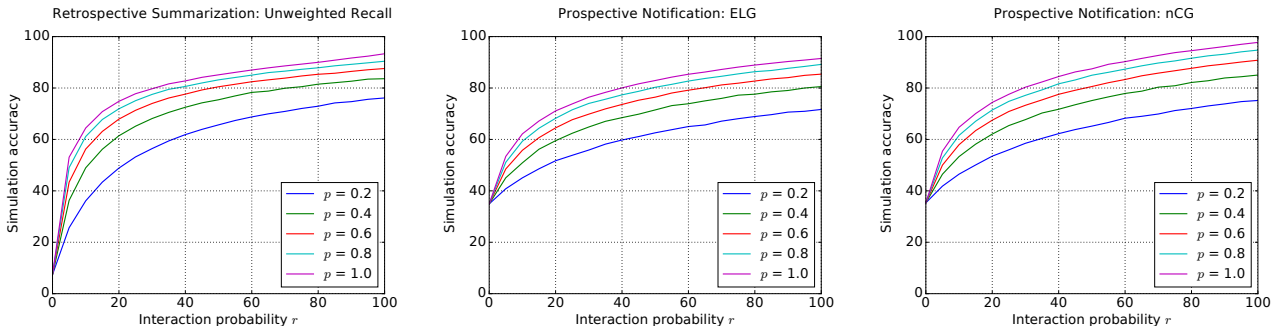

Figure 7: Simulation accuracy combining both retention probability p and interaction probability r.

ity analysis? These results are shown in Figure 7, organized in the same manner as the other graphs (also averaged over three trials). For clarity, we only show results for all pairs. The interaction probability r is plotted on the x axis, with lines representing retention probability $p = \{0.2, 0.4, 0.6, 0.8, 1.0\}$. As expected, we are able to achieve good accuracy, even while randomly discarding system output, and with limited interactions. Given these tradeoff curves, an experiment designer can strike the desired balance between accuracy and verbosity.

6. CONCLUSIONS

In this paper, we describe and validate a novel interleaved evaluation methodology for two complementary information seeking tasks on document streams: retrospective summarization and prospective notification. We present a temporal interleaving strategy and a heuristic credit assignment method based on a user interaction model with explicit judgments. Simulations on TREC data demonstrate that our evaluation methodology yields high fidelity comparisons of the relative effectiveness of different systems, compared to the results of batch evaluations.

Although interleaved evaluations for web search are routinely deployed in production environments, we believe that our work is novel in that it tackles two completely different information seeking scenarios. Retrospective summarization and prospective notification are becoming increasingly important as users continue the shift from desktops to mobile devices for information seeking. There remains much more work, starting with a better understanding of user interactions so that we can develop models of implicit judgment and thereby greatly expand the scope of our evaluations, but this paper takes an important first step.

7. ACKNOWLEDGMENTS

This work was supported in part by the U.S. National Science Foundation (NSF) under awards IIS-1218043 and CNS-1405688 and the Natural Sciences and Engineering Research Council of Canada (NSERC). All views expressed here are solely those of the authors. We'd like to thank Charlie Clarke and Luchen Tan for helpful discussions.

8. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. *SIGIR*, 2006.

[2] J. Aslam, M. Ekstrand-Abueg, V. Pavlu, F. Diaz, R. McCreadie, and T. Sakai. TREC 2014 Temporal Summarization Track overview. *TREC*, 2014.

[3] O. Chapelle, T. Joachims, F. Radlinski, and Y. Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM TOIS*, 30(1):Article 6, 2012.

[4] O. Chapelle and Y. Zhang. A Dynamic Bayesian Network click model for web search ranking. *WWW*, 2009.

[5] L. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in WWW search. *SIGIR*, 2004.

[6] J. He, C. Zhai, and X. Li. Evaluation of methods for relative comparison of retrieval systems based on clickthroughs. *CIKM*, 2009.

[7] K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. *CIKM*, 2011.

[8] T. Joachims. Optimizing search engines using clickthrough data. *KDD*, 2002.

[9] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM TOIS*, 25(2):1–27, 2007.

[10] D. Kelly. Understanding implicit feedback and document preference: A naturalistic user study. *SIGIR Forum*, 38(1):77–77, 2004.

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

[12] R. Kohavi, R. M. Henne, and D. Sommerfield. Practical guide to controlled experiments on the web: Listen to your customers not to the HiPPO. *KDD*, 2007.

[13] J. Lin, M. Efron, Y. Wang, and G. Sherman. Overview of the TREC-2014 Microblog Track. *TREC*, 2014.

[14] J. Lin, M. Efron, Y. Wang, G. Sherman, and E. Voorhees. Overview of the TREC-2015 Microblog Track. *TREC*, 2015.

[15] F. Radlinski and N. Craswell. Comparing the sensitivity of information retrieval metrics. *SIGIR*, 2010.

[16] F. Radlinski and N. Craswell. Optimized interleaving for online retrieval evaluation. *WSDM*, 2013.

[17] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? *CIKM*, 2008.

[18] T. Sakai, N. Craswell, R. Song, S. Robertson, Z. Dou, and C. Lin. Simple evaluation metrics for diversified search results. *EVIA*, 2010.

[19] A. Schuth, K. Hofmann, and F. Radlinski. Predicting search satisfaction metrics with interleaved comparisons. *SIGIR*, 2015.

[20] A. Schuth, F. Sietsma, S. Whiteson, D. Lefortier, and M. de Rijke. Multileaved comparisons for fast online evaluation. *CIKM*, 2014.

[21] L. Tan, A. Roegiest, J. Lin, and C. L. A. Clarke. An exploration of evaluation metrics for mobile push notifications. *SIGIR*, 2016.

[22] Y. Wang, G. Sherman, J. Lin, and M. Efron. Assessor differences and user preferences in tweet timeline generation. *SIGIR*, 2015.

[23] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. *SIGIR*, 2003.