Online In-Situ Interleaved Evaluation of Real-Time Push Notification Systems

Adam Roegiest, Luchen Tan, and Jimmy Lin
David R. Cheriton School of Computer Science
University of Waterloo, Ontario, Canada
{aroegies,luchen.tan,jimmylin}@uwaterloo.ca

ABSTRACT

Real-time push notification systems monitor continuous document streams such as social media posts and alert users to relevant content directly on their mobile devices. We describe a user study of such systems in the context of the TREC 2016 Real-Time Summarization Track, where system updates are immediately delivered as push notifications to the mobile devices of a cohort of users. Our study represents, to our knowledge, the first deployment of an interleaved evaluation framework for prospective information needs, and also provides an opportunity to examine user behavior in a realistic setting. Results of our online in-situ evaluation are correlated against the results a more traditional post-hoc batch evaluation. We observe substantial correlations between many online and batch evaluation metrics, especially for those that share the same basic design (e.g., are utility-based). For some metrics, we observe little correlation, but are able to identify the volume of messages that a system pushes as one major source of differences.

1 INTRODUCTION

There is growing interest in systems that address prospective information needs against continuous document streams, exemplified by social media services such as Twitter. We might imagine a user having some number of "interest profiles" representing prospective information needs, and the system's task is to automatically monitor the stream of documents to keep the user up to date on topics of interest. For example, a journalist might be interested in collisions involving autonomous vehicles and wishes to receive updates whenever such an event occurs. Although there are a number of ways such updates can be delivered, we consider the case where they are immediately pushed to the user's mobile device as notifications. At a high level, these push notifications must be relevant, novel, and timely.

To date, there have been two formal evaluations of the push notification problem, at the TREC 2015 Microblog Track [15] and the TREC 2016 Real-Time Summarization (RTS) Track [16]. Despite the obvious real-time nature of this problem, systems have been assessed with a post-hoc batch evaluation methodology. It seems obvious that the push notification task should be evaluated in an online manner that better matches how content is actually delivered in operational settings.

We describe a user study of real-time push notification systems in the context of the TREC 2016 RTS Track, in which systems' notifications are delivered to users' mobile devices as soon as they are generated. This evaluation is *online*, in contrast to post-hoc batch evaluations, and *in-situ*, in that the users are going about their daily activities and are interrupted by the systems' output. Since the RTS Track deployed both this online in-situ methodology and a more traditional batch methodology, the setup provided us with an opportunity to compare the results of both.

Contributions. We view our work as having two main contributions: First, we describe, to our knowledge, the first user study and actual deployment of an interleaved evaluation for prospective notifications. Our work is based on a previously-proposed interleaving framework [18] that has only been examined in simulation. We present an analysis of user behavior in such an evaluation methodology and demonstrate that it is workable in practice.

Second, we compare results of our online methodology to a more traditional batch methodology in the same evaluation. A number of metrics for assessing push notification systems have been proposed: we observe substantial correlations between many online and batch metrics, particularly those that share the same basic design (e.g., are utility-based). This is a non-obvious finding, since all judgments in our online methodology are sparse and made locally, with respect to one tweet at a time, whereas the batch evaluation methodology takes into account all relevant tweets via a global clustering process. There are two interpretations of this finding:

- If one believes in the primacy of user-centered evaluations, our results suggest that established batch evaluation metrics are able to capture user preferences.
- On the other hand, our online evaluation methodology is less mature than the batch evaluation methodology, which has been extensively examined over the past several years; its core ideas date back at least a decade. If one takes this perspective and believes in the primacy of the established approach, then our results suggest a cheaper way to conduct evaluation of push notifications systems that yield similar conclusions.

Despite substantial correlations between many online and batch metrics, there are some metrics that exhibit no meaningful correlation. We observe that systems vary widely in the volume of messages they push, and that this is the biggest source of metric disagreement. We do not believe that the proper role of message volume in evaluating push notification systems is fully understood, but this paper elucidates key issues as an important first step.

SIGIR'17, August 7–11, 2017, Shinjuku, Tokyo, Japan

978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080808

2 BACKGROUND AND RELATED WORK

Work on prospective information needs against document streams dates back at least a few decades and is closely related to *ad hoc* document retrieval [6]. Major initiatives in the 1990s include the TREC Filtering Tracks, which ran from 1995 [13] to 2002 [21], and the research program commonly known as topic detection and tracking (TDT) [2]. The TREC Filtering Tracks are best understood as binary classification on *every* document in the collection with respect to standing queries, and TDT is similarly concerned with identifying *all* documents related to a particular event—with an intelligence analyst in mind. In contrast, we are focused on identifying a small set of the most relevant updates to deliver to users—any more than a handful of notifications per day would surely be annoying. Furthermore, in both TREC Filtering and TDT, systems must make online decisions as soon as documents arrive. In the case of push notifications, systems can choose to push older content, thus giving rise to the possibility of algorithms operating on bounded buffers. Latency is one aspect of the evaluation, allowing systems to trade off output quality with timeliness.

More recently, Guo et al. [8] introduced the temporal summarization task, whose goal is to generate concise update summaries from news sources about unexpected events as they develop. This has been operationalized in the TREC Temporal Summarization (TS) Tracks from 2013 to 2015 [4]. The task is closely related to the push notification problem that we study, and in fact the TREC Real-Time Summarization Track, which provides the context for our work, represents a merger of the TS and Microblog Tracks. However, nearly all previous evaluations, including TDT, TREC Filtering, and Temporal Summarization, merely *simulated* the streaming nature of the document collection, whereas in RTS the participants were required to build working systems that operated on tweets posted in real time (more details in Section 3).

Our online in-situ evaluation framework builds on growing interest in so-called Living Labs [22, 24] and related Evaluation-as-a-Service (EaaS) [9] approaches that attempt to better align evaluation methodologies with user task models and real-world constraints to increase the fidelity of research experiments. In this respect, our comparison between user-oriented and batch evaluations ties into a long history of research that examines the correlation between effectiveness metrics from system-oriented evaluations and metrics from user-oriented evaluations [1, 3, 10, 23, 26, 29–31]. There is, however, one important difference: all of these cited papers, with one exception [31], focus on *ad hoc* retrieval, which has received much attention over the years. Although there have been previous user studies on push notifications from the HCI perspective (e.g., [17]), there is relatively little empirical work on prospective information needs that we can draw from.

The final thread of relevant work concerns interleaved evaluations [7, 11, 19, 20, 25], which have emerged as the preferred approach to evaluating web search engines over traditional A/B testing [12]. Our work departs from this large body of literature because these papers all focus on web search ranking, whereas we tackle the push notification problem: in our task, systems must take into account temporality and redundancy, both of which are less important in web search. The length of system output (i.e., volume of pushed messages) is another major difference between our task and web ranking. These issues were explored in a recent paper by Qian et al. [18], who extended the interleaved evaluation methodology to retrospective and prospective information needs on document streams. However, their proposed approach was only validated in simulation. We take the next step by deploying an adapted version of their proposed technique in a live user study.

3 EVALUATION METHODOLOGY

Although the push notification problem is applicable to document streams in general, we focus on social media posts: the public nature of Twitter makes tweets the ideal source for shared evaluations. In particular, Twitter provides a streaming API through which clients can obtain a sample (approximately 1%) of public tweets—this level of access is available to anyone who signs up for an account. In order to evaluate push notification systems in a realistic setting, the TREC 2016 RTS Track defined an official evaluation period during which all participants "listened" to the tweet sample stream to identify relevant and novel tweets with respect to users' interest profiles in a timely manner. The evaluation period began Tuesday, August 2, 2016 00:00:00 UTC and lasted until Thursday, August 11, 2016 23:59:59 UTC.

Interest profiles, which represent users' information needs, followed the standard TREC *ad hoc* topic format of "title", "description", and "narrative". These were made available to all participants a few weeks prior to the beginning of the evaluation period. Given the prospective nature of the profiles, it is difficult to anticipate what topics will be discussed during the evaluation period and what events will be "interesting". Instead, the organizers adopted the strategy of "overgenerate and cull": in total, 203 interest profiles were provided to the participants, more than there were resources available for assessment, with the anticipation of letting users decide what profiles should be assessed (more details below).

3.1 Online Evaluation Setup

The TREC 2016 RTS Track contained two separate tasks: push notifications (so-called "Scenario A") and email digests (so-called "Scenario B"). In this paper we are only concerned with push notifications, but for more details we refer the reader to the track overview [16].

The overall evaluation framework is shown in Figure 1. Before the evaluation period, participants "registered" their systems with the evaluation broker to request unique tokens (via a REST API), which are used in subsequent requests to associate submitted tweets with specific systems.[1] During the evaluation period, whenever a system identified a relevant tweet with respect to an interest profile, the system submitted the tweet id to the evaluation broker (also via a REST API), which recorded the submission time. Each system was allowed to push at most ten tweets per interest profile per day; this limit represents an attempt to model user fatigue.

Once the evaluation broker recorded a system's submission, the tweet was *immediately* delivered to the mobile devices of a group of users, where it was rendered as a push notification containing both the text of the tweet and the corresponding interest profile.

[1] As is standard in TREC, each participant was permitted to submit multiple "runs" (usually system variants), but for the purposes of this discussion we refer to them as different systems.

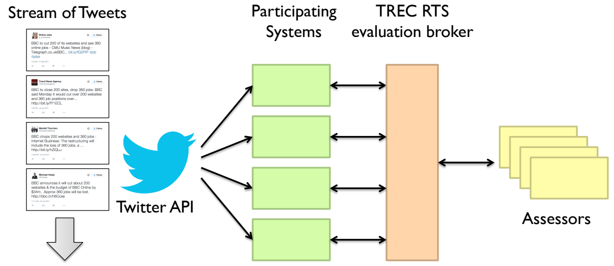

Figure 1: Evaluation setup for push notifications: systems "listen" to the live Twitter sample stream and send results to the evaluation broker, which then delivers push notifications to users.

The user may choose to attend to the tweet immediately, or if it arrived at an inopportune time, to ignore it. Either way, the tweet is added to a queue in a custom app on the user's mobile device, which she can access at any time to examine the queue of accumulated tweets. For each tweet, the user can make one of three judgments with respect to the associated interest profile: *relevant*, if the tweet contains relevant and novel information; *redundant*, if the tweet contains relevant information, but is substantively similar to another tweet that the user had already seen; *not relevant*, if the tweet does not contain relevant information. As the user provides judgments, results are relayed back to the evaluation broker and recorded. Users have the option of logging out of the app, at which point they will cease to receive notifications completely (until they log back in).

Our setup has two distinct characteristics: First, judgments happen online as systems generate output, as opposed to traditional batch post-hoc evaluation methodologies, which consider the documents some time (typically, weeks) after they have been generated by the systems. Note that although the push notifications are delivered in real-time, it is not necessarily the case that judgments are provided in real time since users can ignore the notifications and come back to them later. Second, our judgments are *in situ*, in the sense that the users are going about their daily activities (and are thus interrupted by the notifications). This aspect of the design accurately mirrors the intended use of push notification systems. Furthermore, from the evaluation perspective, we believe that this setup yields more situationally-accurate assessments, particularly for rapidly developing events. With post-hoc batch evaluations, there is always a bit of disconnect as the assessor needs to "imagine" herself at the time the update was pushed. With our evaluation framework, we remove this disconnect.

Our entire evaluation was framed as a user study (with appropriate ethics review and approval). A few weeks prior to the beginning of the evaluation period, we recruited users from the undergraduate and graduate student population at the University of Waterloo, via posts on various email lists as well as paper flyers on bulletin boards. The users were compensated $5 to install the mobile assessment app and then $1 per 20 judgments provided.

As part of the training process, users installed the custom app described above on their mobile devices. In addition, they subscribed, using an online form, to receive notifications for interest profiles they were interested in, selecting from the complete list of 203 interest profiles provided to all systems. To encourage diversity, we did not allow more than three users to select the same profile (on a first come, first served basis).

The evaluation broker followed the temporal interleaving strategy proposed by Qian et al. [18], which meant that tweets were pushed to users as soon as the broker received the submitted tweets from the systems. Although Qian et al. only discussed interleaving the output of two systems, it is straightforward to extend their strategy to multiple systems. The broker made sure that each tweet was only pushed once (per profile), in the case where the same tweet was submitted by multiple systems at different times. Although one can imagine different "routing" algorithms for pushing tweets to different users that have subscribed to a profile, we implemented the simplest possible algorithm where the tweet was pushed to *all* users that had subscribed to the profile. This meant that the broker might receive more than one judgment per tweet.

3.2 Online Metrics

The output of our online in-situ evaluation is a sequence of judgments, which need further aggregation before we can use the results to compare the effectiveness of different systems. Note that this aggregation is more complicated than a similar process in interleaved evaluations for web search because systems can vary widely in tweet volume (i.e., how many tweets they push). In standard interleaving techniques for evaluating web search, both variant algorithms being tested contribute to the final ranking for *all* queries—thus, it usually suffices to count the number (or fraction) of clicks to determine the winner (e.g., [7, 24]). However, in our case, there isn't a query that lends itself to a natural paired comparison. Some systems are quite profligate in dispatching notifications, while other systems are very quiet.

Another implication of our interleaved evaluation setup is that a user will encounter tweets from different systems, which makes the proper interpretation of "redundant" judgments more complex. A tweet might only be redundant because the same information was contained in a tweet pushed earlier by another system (and thus not the "fault" of the particular system that pushed the tweet). In other words, the interleaving itself was directly responsible for introducing the redundancy. This observation was made by Qian et al. [18], who proposed a heuristic for more accurate credit assignment when interleaving two systems. However, we decided to adopt a much simpler approach (explained below), which is justified by our experimental results (more details later).

Recognizing the issues discussed above, we computed two aggregate metrics based on user judgments:

Online Precision. A simple and intuitive metric is to measure precision, or the fraction of relevant judgments:

$$\frac{\text{relevant}}{\text{relevant} + \text{redundant} + \text{not relevant}} \tag{1}$$

We term this "strict" precision because systems don't get credit for redundant judgments. As an alternative, we could compute "lenient" precision, where the numerator includes redundant judgments. Extending this further, redundant judgments could in principle be assigned fractional credit, but as we discuss later, such schemes do not appear to have any impact on our overall findings.

Two minor details are worth mentioning for the proper interpretation of this metric: First, tweets may be judged multiple times since a tweet is pushed to all users who had subscribed to the profile. For simplicity, all judgments are included in our calculation. Second, our precision computation represents a micro-average (and *not* an average across per-profile precision). This choice was made due to the sparsity of judgments: macro-averaging would magnify the effects of profiles with few judgments.

Online Utility. As an alternative to online precision, we could take a utility-based perspective and measure the total gain received by the user. The simplest method would be to compute the following:

$$\text{relevant} - \text{redundant} - \text{not relevant} \tag{2}$$

which we refer to as the "strict" variant of online utility. Paralleling the precision variants above, we define a "lenient" version of the metric as follows:

$$(\text{relevant} + \text{redundant}) - \text{not relevant} \tag{3}$$

Of course, we could further generalize with weights for each type of judgment. However, we lack the empirical basis for setting the weights. Furthermore, experimental analyses show that our findings are insensitive to weight settings.

To summarize: from user judgments, we compute two aggregate metrics—online precision and online utility. Note that there is no good way to compute a recall-oriented metric since we have no control over when and how frequently user judgments are provided. This is a fundamental limitation of this type of user study.

3.3 Batch Evaluation Setup

In order to mitigate the risk inherent in any new evaluation methodology, the TREC 2016 RTS Track also deployed a more traditional post-hoc batch evaluation methodology—specifically, the approach developed for the Tweet Timeline Generation (TTG) task at the TREC 2014 Microblog Track [14], which was also used in 2015 [15]. The methodology has been externally validated [31] and can be considered mature due to its deployment in multiple formal evaluations. The assessment workflow proceeded in two major stages: relevance assessment and semantic clustering. Here we provide only a brief overview, referring the reader to the cited papers above for additional details.

Tweets returned by participating systems were judged for relevance by NIST assessors via pooling. Note that this occurred after the live evaluation period ended, so it was possible to gather all tweets pushed by all participating systems. NIST assessors began a few days after the end of the evaluation period to minimize the "staleness" of tweets. Each tweet was assigned one of three judgments: not relevant, relevant, or highly-relevant. After the relevance assessment process, the NIST assessors proceeded to perform semantic clustering on only the relevant and highly-relevant tweets. Using a custom interface, they grouped tweets into clusters in which tweets share substantively similar content, or more colloquially, "say the same thing". The interpretation of what this means operationally was left to the discretion of the assessor. In particular, they were not given a particular target number of clusters to form; rather, they were asked to use their judgment, considering both the interest profile and the actual tweets. The output of the cluster annotation process is a list of tweet clusters; each cluster contains tweets that are assumed to convey the same information.

3.4 Batch Evaluation Metrics

As previously discussed, push notifications should be relevant, non-redundant, and timely. One challenge, however, is that there is little empirical work on how users perceive timeliness. Therefore, instead of devising a single-point metric that tries to combine all three characteristics, the organizers decided to separately capture output quality (relevance and redundancy) and timeliness (latency). In this paper, we only focus on output quality metrics. In short, RTS batch evaluation metrics attempt to capture precision, recall, and overall utility. We elaborate below:

Expected Gain (EG) for an interest profile on a particular day is defined as $\frac{1}{N}\sum \text{G}(t)$, where N is the number of tweets returned and $\text{G}(t)$ is the gain of each tweet: not relevant tweets receive a gain of 0; relevant tweets receive a gain of 0.5; highly-relevant tweets receive a gain of 1.0. Once a tweet from a cluster is retrieved, all other tweets from the same cluster automatically become not relevant. This penalizes systems for returning redundant information. Expected gain can be interpreted as a precision metric.

Normalized Cumulative Gain (nCG) for an interest profile on a particular day is defined as $\frac{1}{\mathcal{Z}}\sum \text{G}(t)$, where $\mathcal{Z}$ is the maximum possible gain (given the ten tweet per day limit). The gain of each individual tweet is computed in the same way as above. Note that gain is not discounted (as in nDCG) because the notion of document ranks is not meaningful in this context. We can interpret nCG as a recall-like metric.

The score for a run is the average over scores for each day over all interest profiles. An interesting question is how scores should be computed for days in which there are no relevant tweets: for rhetorical convenience, we call days in which there are no relevant tweets for a particular interest profile (in the pool) "silent days", in contrast to "eventful days" (when there are relevant tweets). In the EG-1 and nCG-1 variants of the metrics, on a silent day, the system receives a score of one (i.e., a perfect score) if it does not push any tweets, or a score of zero otherwise. In the EG-0 and nCG-0 variants of the metrics, for a silent day, all systems receive a gain of zero no matter what they do. For more details about this distinction, see Tan et al. [28].

Therefore, under EG-1 and nCG-1, systems are rewarded for recognizing that there are no relevant tweets for an interest profile on a particular day and remaining silent (i.e., the system does not push any tweets). The EG-0 and nCG-0 variants of the metrics do not reward recognizing silent days: that is, it never hurts to push tweets. We show later in our analyses that EG-0 and nCG-0 are poorly-formulated metrics.

Gain Minus Pain (GMP) is defined as $\alpha \cdot \sum G - (1-\alpha) \cdot P$, where G (gain) is computed in the same manner as above, pain P is the number of non-relevant tweets that the system pushed, and α controls the balance of weights between the two. We investigated three α settings: 0.33, 0.50, and 0.66. Note that this metric is the same as the linear utility metrics used in the TREC Filtering [13, 21] and Microblog [27] Tracks, although our formulation takes a slightly different mathematical form.

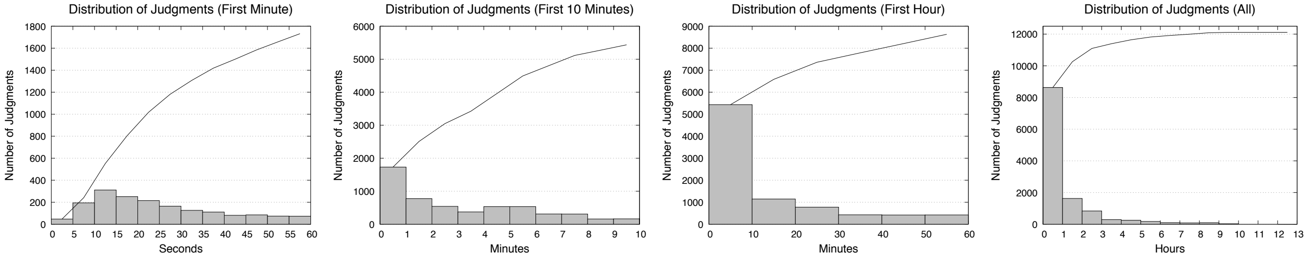

Figure 2: Distribution of response times over the first minute (14.3%), first ten minutes (44.9%), first hour (71.3%), and across the entire evaluation period. Percentages in parentheses show how many judgments were received in the corresponding period.

User	Judgments	Profiles	Messages	Response
1	53	4	1619	3.27%
2	3305	10	7141	46.28%
3	136	10	5860	2.32%
4	327	8	3795	8.62%
5	949	12	6330	14.99%
6	28	12	7211	0.39%
7	281	10	4162	6.75%
8	1908	15	7754	24.61%
9	3791	33	16654	22.76%
10	680	16	7257	9.37%
11	107	43	22676	0.47%
12	324	2	938	34.54%
13	226	12	7058	3.20%

Table 1: User statistics. For each user, columns show the number of judgments provided, the number of interest profiles subscribed to, the maximum number of push notifications received, and the response rate.

To summarize: we have multiple batch metrics for evaluating push notification systems: EG-1 and EG-0 (both of which measure precision), nCG-1 and nCG-0 (both of which measure recall), and GMP with $\alpha = \{0.33, 0.50, 0.66\}$ (which capture utility).

4 USER BEHAVIOR

The evaluation methodology for push notifications detailed above was deployed in the TREC 2016 Real-Time Summarization Track. In total, 18 groups from around the world participated, submitting a total of 41 systems (runs). Over the evaluation period, these runs pushed a total of 161,726 tweets, or 95,113 unique tweets after de-duplicating within profiles.

To simplify app development, we only targeted users of Android devices. For our evaluation, we recruited a total of 18 users, 13 of whom ultimately provided judgments. Of these, 11 were either graduates or undergraduate students at the University of Waterloo. In total, we received 12,115 judgments over the assessment period, with a minimum of 28 and a maximum of 3,791 by an individual user. Overall, 122 interest profiles received at least one judgment; 93 received at least 10 judgments; 67 received at least 50 judgments; 44 received at least 100 judgments.

We begin with descriptive characterizations of user behavior: a breakdown is shown in Table 1. The second column lists the number of judgments each user provided and the third column shows the number of profiles that each user subscribed to. The fourth column shows the sum of all push notifications for the profiles that each user subscribed to: this count captures the maximum number of push notifications that the user *could have received* during the evaluation period. Note that we do not have the *actual* number of notifications each user received because the user could have logged out during some periods of time or otherwise adjusted the local device settings (e.g., to disable notifications). The final column shows the response rate, computed as the fraction between the second and fourth columns (which is a lower-bound estimate). From this table, we see that some users are quite diligent in providing judgments, while others provide judgments more sporadically.

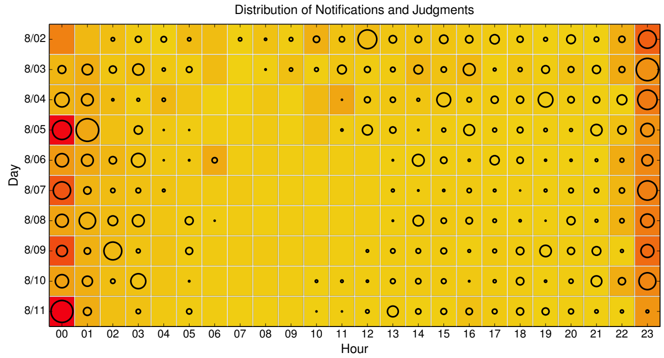

Figure 3: Heatmap showing the volume of push notifications, overlaid with circles whose areas are proportional to the number of received judgments.

How quickly do users provide judgments? The plots in Figure 2 answer this question, showing the distribution of response times over the first minute, first ten minutes, first hour, and across the entire evaluation period. The bars show bucketed counts, while the line graph shows cumulative counts. Normalizing, we find that 14.3% of judgments arrive within the first minute after the push notification has been delivered, 44.9% of judgments arrive in the first ten minutes, and 71.3% of judgments arrive in the first hour. We find that users are quite responsive to interruptions!

Finally, Figure 3 provides an overview of the entire evaluation period. In the heatmap, each box represents one hour across the ten-day evaluation period: the color reflects the total number of pushed tweets by all systems across all profiles that at least one user subscribed to. A deeper red indicates more tweets pushed. The overlaid circles represent judgments received from all users, where

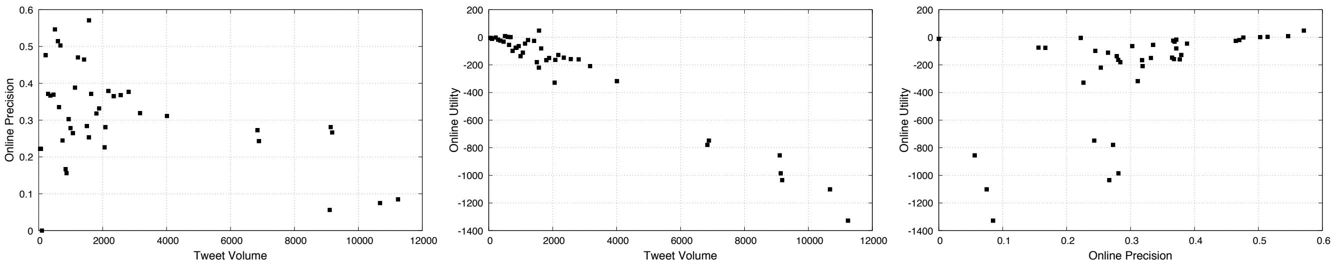

Figure 4: Analyses of online metrics. The left and middle plots show tweet volume vs. online precision and online utility. The right plot shows almost no correlation between online precision and online utility because systems with roughly the same online precision can vary widely in push volume.

the area is proportional to the number of judgments. Note that time is given in UTC; for reference, 00:00:00 UTC translates into 20:00:00 in the local time zone of the users.

A few interesting observations follow: we find that relatively more tweets are pushed by systems in the first and final hours of each day. We believe that this is mostly an evaluation artifact: recall that each system receives a quota of ten tweets per day per profile. At the beginning of each day (hour 00), the quota resets—thus allowing systems that have used up their quota the previous day to start pushing notifications again. At the end of each day (hour 23), we believe that the rise in tweets corresponds to systems "using up" the remainder of their quota.

Looking at the circles, which represent the volume of judgments, we see that they mostly line up with the push volume. That is, darker red cells generally have larger circles—the more tweets systems push, the more judgments we receive. However, there are some deviations, which represent delayed judgments—for example, a burst of tweets that wasn't examined until some time later. It is also interesting to note that with the exception of night time when users are asleep, there does not appear to be a consistent diurnal cycle across our population of users. The users are exposed to a pretty constant stream of push notifications throughout the day (and indeed during sleeping hours also), but there doesn't appear to be a time of the day when we consistently receive more judgments.

5 ANALYSIS

By design, the TREC 2016 RTS Track employed both the online in-situ interleaved evaluation methodology as well as the more traditional post-hoc batch evaluation methodology. This means that for the same systems and interest profiles, we have independently-derived metrics from two very different approaches. For the batch metrics, NIST assessors fully judged 56 interest profiles (relevance judgments and clusters). Section 3.3 and Section 3.4 provide an overview, but since this is not the focus of our work, we refer the reader to details provided in the track overview [16].

We begin by presenting separate analyses of online and batch metrics, and then describe results of correlation analyses between them. In particular, comparing online and batch metrics allows us to explore two questions: From the perspective of the user, do user preferences correlate with batch metrics? From the perspective of system-centered evaluations, can unreliable online judgments replace high-quality NIST assessors?

In considering the online metrics, there is a question regarding which metric to use—the "strict" or "lenient" variant of online precision and online utility (see Section 3.2). We performed analyses with both: All plots look very similar, except for systematic shifts due to the metric variants; for example, all the absolute precision values increase from "strict" to "lenient" precision, but the overall relationships between the points remain largely unchanged. Therefore, we only report the "strict" variants here for brevity. This also suggests that the credit assignment heuristic of Qian et al. [18], which lies somewhere between the strict and lenient variants, is also unlikely to alter our findings.

5.1 Online Metrics

Three different analyses of the online metrics are shown in Figure 4. We organize our findings around two themes:

Precision is an intrinsic metric of push notification quality, while utility is a convenient composite metric. Online precision computes the fraction of relevant user judgments, but does not factor in the volume of tweets that a system pushes. Online utility implies a particular precision target with volume as a scaling factor, and thus serves as a convenient composite metric. To see why this is so, consider a system that achieves a precision of 0.5: setting aside relevance grades for now, the expected utility per tweet is zero (for $\alpha = 0.5$) and the overall expected utility is also zero, regardless of how many tweets the system pushes. A system with lower precision has a negative expected utility per tweet, and the total expected utility is simply that value multiplied by the volume of tweets. Since the precision of most systems in the evaluation falls below 0.5, we observe a strong negative correlation between tweet volume and utility: this can be clearly seen in the middle plot in Figure 4, which shows tweet volume against online utility. Here, volume is measured as the number of tweets pushed by the system for all interest profiles that received at least one judgment.

Our argument can be generalized to other ways of computing utility. Of course, one could assign different weights to non-relevant tweets, but for every weighting scheme, there is an implied precision at which the expected utility per tweet is zero. Only systems that have higher precision can provide positive utility; otherwise, negative utility is directly proportional to push volume. The same idea can be straightforwardly extended to relevance grades: all utility-based metrics encode (at least implicitly) a breakeven point between "good" results and "bad" results.

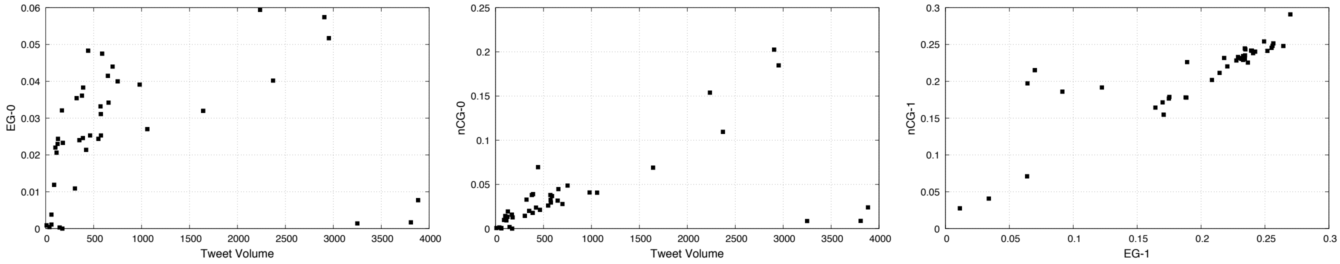

Figure 5: Analyses of batch metrics. The left and middle plots show tweet volume vs. EG-0 and nCG-0, illustrating the dominant effect of tweet volume, which is a major flaw in those metrics. The right plot shows a strong correlation between EG-1 and nCG-1 (with the exception of a few outliers).

Tweet volume is an independent and important measure of system output. Building on the previous observation, the independence of tweet volume and precision can be clearly seen in the right plot of Figure 4, where we observe almost no relationship between online precision and online utility. This is further reinforced in the left plot, which shows tweet volume vs. online precision. Although we observe a negative correlation overall, the effect is primarily due to outliers. If we focus only on systems with tweet volume under 2000, there is little correlation between online precision and volume. In particular, in the band from 0.3 to 0.4 precision, systems vary widely in volume. This leads to the wide spread of precision values for systems that have similar utility in the right plot. Thus, from the user perspective, we believe that online precision and tweet volume are the two fundamental inputs to metrics for measuring system effectiveness.

5.2 Batch Metrics

Analyses of batch metrics are shown in Figure 5. Our two main findings are as follows:

EG-0 and nCG-0 are flawed metrics. Recall that these variants do not reward systems for recognizing that there is no relevant information and staying silent, and thus it never hurts to push notifications. As a result, these two metrics reward systems that push a large volume of tweets without necessarily differentiating the quality of those tweets. This is most evident in the middle plot in Figure 5, which shows tweet volume against nCG-0. Tweet volume here is measured as the total number of tweets pushed across the interest profiles that were evaluated by NIST assessors. Due to the much more involved batch evaluation methodology, the NIST assessors considered a smaller set of interest profiles than users in the online evaluation, and thus the plots report smaller tweet volumes. While it is possible to push a large number of non-relevant tweets (bottom right corner of the middle plot), in general, the more tweets a system pushes, the higher its nCG-0 score.

We note a similar effect for EG-0, although less pronounced, from the left plot in Figure 5, which shows tweet volume against EG-0. Once again, disregarding the outliers in the bottom right corner, higher tweet volumes correlate with higher EG-0 scores. Since under EG-0 all systems receive EG scores of zero for silent days when there are no relevant tweets, it never hurts to "guess" by pushing tweets. Thus, we believe that EG-0 and nCG-0 are flawed metrics since it is unlikely that users desire high-volume systems that push tweets of questionable quality. We advocate that these metrics be dropped in future evaluations, and we remove EG-0 and nCG-0 from subsequent analyses in this paper.

EG-1 and nCG-1 are highly correlated. This correlation can be seen in the right plot in Figure 5. In contrast to EG-0 and nCG-0, these metrics reward systems for remaining silent on days when there is no relevant content. The plot shows that systems with higher gain (utility) also tend to achieve higher precision.

It is interesting to observe that such a strong correlation exists between EG-1 and nCG-1, since EG is quite similar to precision and nCG is recall-like: in principle, systems could make tradeoffs along these two dimensions independently. However, this might simply be a statement about the current state of push notification techniques. Nevertheless, we do observe some outliers: the group of runs around 0.06 in EG-1 and around 0.2 in nCG-1 are those that push a high volume of tweets. What they lack in the overall quality of individual tweets, they make up in volume, leading to higher nCG-1 than their EG-1 scores would otherwise suggest (i.e., the points lie above the trend).

5.3 Online vs. Batch Metrics

Scatter plots correlating various batch metrics against online utility and online precision are shown in Figure 6. In the top row we show correlations between EG-1, nCG-1, and GMP ($\alpha = 0.50$) against online utility; in the bottom row, the same metrics against online precision. Note that we removed EG-0 and nCG-0 from consideration given the discussion above. For GMP, the choice of α does not change the shape of the plots and does not affect our conclusions, so for brevity we omit GMP with $\alpha = \{0.33, 0.66\}$.

When performing correlational studies on retrieval experiments, outlier runs may have a disproportional influence on the results. For example, poor performing systems are easy to distinguish, and most metrics can easily identify poor systems. Therefore, including such systems tends to increase correlations in ways that are not particularly helpful in discriminating systems that are not outliers. The outliers in our case are systems that push a large volume of tweets and those that push very few tweets. From Figure 4 and Figure 5 we can identify the outliers as those runs that push more than 4000 tweets in the online evaluation and more than 1500 tweets in the batch evaluation. There are eight such systems and both criteria identify exactly the same systems. On the whole, these are systems that perform poorly. In the plots, we identify

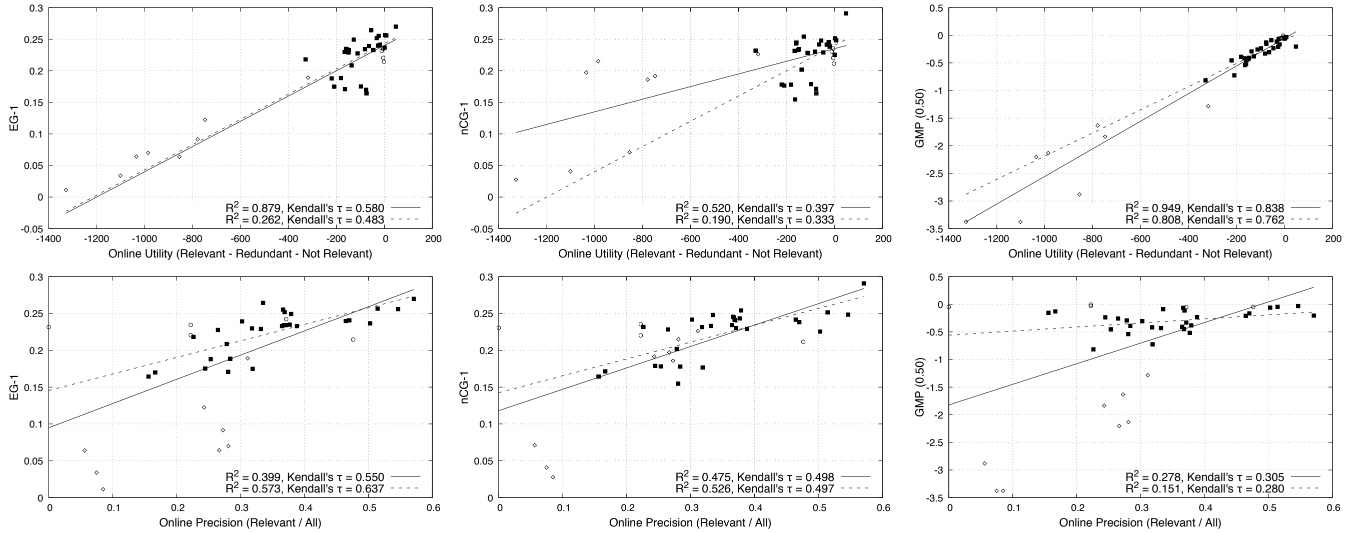

Figure 6: Scatter plots comparing EG-1, nCG-1, and GMP ($\alpha = 0.50$) to online utility (top row) and to online precision (bottom row). High-volume systems are represented by empty diamonds and low-volume systems are represented by empty circles. Solid lines denote best fit lines with all points; dotted lines denote best fit lines discarding high- and low-volume systems.

these runs separately as empty diamonds. At the other end of the spectrum, we have runs that push very few tweets. We arbitrarily set this threshold to be less than 100 tweets pushed based on the batch evaluation. Since the batch evaluation considered 56 interest profiles spanning 10 days, this translates into less than two tweets per interest profile (over the entire span), which is close to a system that basically does nothing. There are five such runs, identified as empty circles in the plots. This leaves us with 28 systems (runs) whose tweet volumes fall somewhere in the middle, identified by the solid squares in the plots.

For each analysis, we considered two separate conditions: First, with *all* runs. The results of linear regressions are shown as solid lines. Second, we discarded high- and low-volume systems (as described above); the results of linear regressions in these cases are shown as dotted lines. The second condition attempts to remove the influence of these outliers: in some cases, it affects the findings, but in other cases, not. For both conditions, alongside the coefficient of determination for the linear regression shown in the legend, we also report rank correlation in terms of Kendall's τ, the standard metric of rank stability in information retrieval experiments.

There is a lot to unpack in our results, and so we organize our findings around several major themes:

Online utility is highly correlated with GMP. Our strongest finding is a high correlation between online utility and GMP (see Figure 6, top row, right plot). The correlation weakens slightly if we discard the high-volume and low-volume systems, but is still substantial. Because there are many points packed in the top right corner of the plot, the Kendall's τ we observe is a bit lower compared to the coefficient of determination, but still solidly in the range that would be considered good agreement for retrieval experiments.

At first glance, this might seem like an obvious finding since online utility and GMP are both utility-based metrics, but this is a non-obvious result for several reasons: GMP is computed over cluster judgments from pooled tweets and therefore represents a global view over tweets from all systems. In particular, tweets are grouped into clusters and systems do not get credit for pushing tweets that say the same thing. In contrast, online utility captures only a local view of content—users are making decisions about each tweet pushed to them, and the redundant judgments are subjected to the fallacies of human memory (i.e., users may have forgotten having seen similar tweets).

In addition, GMP is computed using "dense" judgments over a smaller set of profiles gathered by pooling, whereas online utility is computed from sparse judgments over uncontrolled samples, since we have no control over when and how frequently users provide judgments. It is surprising that sporadic, unpredictable, in-situ judgments from a multitude of users yield results that are highly-correlated with the careful deliberations of professional NIST assessors.

Online precision exhibits moderate correlations with EG-1 and nCG-1. This is shown in Figure 6, bottom row, left and center plots. Since the definition of EG-1 shares similarities with online precision, one might expect this, and since EG-1 and nCG-1 are correlated (right plot, Figure 5), it is no surprise to find that online precision also correlates with nCG-1. As with GMP above, we emphasize that online precision is computed from tweets evaluated in isolation, whereas EG-1 and nCG-1 are based on cluster annotations, which take into account the global cluster structure of tweets relevant to the interest profile.

In both cases, the correlation strengthens if we discard high- and low-volume systems (although for nCG-1, Kendall's τ is essentially unchanged). An empty run (i.e., a system that does nothing) would receive a score of 0.2339 for EG-1 and nCG-1, which is simply the fraction of silent days when there are no relevant tweets. Therefore, low-volume systems receive scores that are close to the score of an empty run, and this throws off the correlation.

Online utility exhibits at best a weak correlation with EG-1 and nCG-1. This is shown in Figure 6, top row, left and center plots. In both these cases, the correlation weakens substantially if we remove the high- and low-volume systems. Therefore, outliers are giving the impression of a stronger correlation than one that actually exists. In a sense, it is not surprising that we observe little correlation between a utility metric vs. precision-oriented and recall-like metrics.

Online precision exhibits almost no correlation with GMP. This is shown in Figure 6, bottom row, right plot. This finding is consistent with what we see in Figure 4. Precision doesn't capture tweet volume, whereas tweet volume has a substantial impact on utility, as previously discussed.

6 DISCUSSION AND LIMITATIONS

One potential objection to our evaluation methodology is our reliance on explicit user judgments. This, of course, stands in contrast to web ranking, which benefits from a tremendous amount of implicit judgments that are collected as a byproduct of users searching. However, we argue that explicit judgments are an important component of push notifications since they, by definition, interrupt the user. Given that the interruption has already occurred (if the user has chosen to attend to the notification), allowing the user an opportunity to provide feedback seems like good design. Thus, we argue that our evaluation setup is realistic, mirroring how a production push notification system might be deployed. For example, in some implementations today, notifications already appear with a "dismiss" option for users to take explicit action; adding options for quality feedback would incur minimal extra cost.

At a high-level, our work supports three findings: First, that online in-situ interleaved evaluations of push notifications systems are workable in practice. It is indeed possible to recruit users and they are willing to provide sufficient judgments (quite diligently, in fact) to meaningfully evaluate systems. This seems consistent with our arguments above regarding the role of explicit judgments in push notification systems. Second, we observe substantial correlations between online and batch evaluation metrics that share the same design (e.g., are precision-oriented or utility-based). Third, the volume of messages that a system pushes is an important aspect of system evaluation, but its role is not fully understood.

With respect to the second finding—the substantial correlation between online and batch metrics—this is by no means obvious, given the large body of literature that has shown divergences between user- and system-oriented metrics (see Section 2). We have touched on some of the main differences between the online and batch methodologies, but they bear additional emphasis:

- The batch evaluation considered all tweets that all systems pushed during the evaluation period. That is, all push notifications were included in the pool and so the judgments are exhaustive from the perspective of the participants. This stands in contrast to the online judgments, which are best characterized as a small convenience sample by users (see response rates in Table 1). We have no control over when and how many judgments are provided—and whether there are any systematic biases, for example, a user who only marks relevant tweets but ignores non-relevant tweets (i.e., a bias against explicit negative judgments).
- The batch metrics all operate at the level of semantic clusters, taking into account redundancy. These clusters are formed from the pool and therefore contain a "global" view of tweets pushed by all systems. Accordingly, systems are penalized for retrieving multiple tweets that say that same thing. In contrast, user judgments occur tweet-by-tweet and represent a "local" view—our users assess only the tweets in front of them. Furthermore, redundant judgments are made with respect only to tweets the users had previously assessed, and are subject to the effects of imperfect memory. Another consequence of this setup is that from batch judgments we can characterize silent days (at least within the limitations of pooling), whereas with online judgments there is no way for the user to obtain this information. Thus, it is not possible for an online metric to reward systems for "staying quiet". Finally, high-volume systems (which tend to have lower precision) are disproportionately represented.
- The batch evaluation used professional NIST assessors, many with decades of experience. They have become the gold standard against which human judgments are compared [5]. Contrast this with our users: since they are simply going about their daily lives (which is indeed the point), we have no idea in what context they are assessing tweets—were they alone in a quiet setting considering tweets with care or hurriedly skimming tweets while multi-tasking? We assume our users were acting in good faith and judging the tweets to the best of their ability (and we have no reason to suspect otherwise), but the overall fidelity and quality of judgments are likely to be lower than the NIST assessors who operated in a carefully-controlled environment.

These differences considered, we find the correlations between online and batch metrics non-obvious and interesting. There are two different interpretations to these results:

If one believes in the primacy of user-centered evaluations, our findings suggest that established batch evaluation metrics are able to capture user preferences. That is, the batch metrics are capturing aspects of what users care about in *useful* systems. This result nicely complements the findings of Wang et al. [31], who validated batch metrics for the related task of retrospective timeline summarization over tweet streams.

On the other hand, one might put more faith in a mature batch evaluation methodology that has been through the gauntlet of multiple deployments, and whose core ideas date back at least a decade. If one takes this perspective and believes in the primacy of the established approach, then our results suggest a cheaper way to conduct evaluations of push notification systems that yield similar conclusions. Of course, these two perspectives are not necessarily conflicting. Instead, they point to more work that is necessary to fully align user- and system-oriented perspectives to assessing push notification systems.

It makes sense to discuss some of the limitations of this work. Our users are compensated for their participation in the study and thus can be assumed to operate under certain social norms. One might argue that their behavior would be different had they "organically" discovered an app for push notifications. While this is certainly a legitimate criticism, it could be leveled against any user study that involves compensation—potential differences between paid subjects and "real users" are beyond the scope of this study.

A closely-related issue is the fact that our users subscribed to interest profiles that were not "their own", i.e., they did not come up with the information needs themselves. This concern, however, is mitigated by having users select from a broad range of profiles (a couple hundred) to match their interests. Therefore, our evaluation is less likely to have suffered from user indifference.

Another limitation of our study is that it captures only a snapshot of current technology. This, of course, is an implicit qualification of any evaluation, not just our work. For example, consider attempts to control the volume of push notifications and to recognize when there is no relevant content: such techniques are nascent at best, since the community is just beginning to understand the nuances of systems "learning when to shut up". We have found that these issues are confounding variables when trying to correlate online and batch metrics, but as the technology evolves and matures, the nature of the confound might change. As another example, we empirically observe that EG-1 and nCG-1 are correlated, even though in principle systems can operate in a tradeoff space in which the measures are not correlated. Nevertheless, we are unable to speculate on future developments that have yet to happen—we can only draw conclusions based on the data at hand. The only way to address this limitation is a follow-up study that considers systems once push notification techniques have substantially progressed.

7 CONCLUSIONS

This paper describes a formal user study of push notification systems with two distinct characteristics: tweets are assessed *online* and *in-situ*. As the infrastructure for conducting our evaluation can be reused (all software deployed in this study is open source), future iterations will take less effort. Therefore, we hope to see more of these evaluations as the methodology becomes "just another hammer" in the toolbox of information retrieval researchers and practitioners.

As an outstanding issue, we believe that the proper role of notification volume in evaluating systems is not yet fully understood. As we have empirically observed, systems with the same precision can vary widely in the volume of notifications they push. However, the question remains: how much content should a system actually push? Even assuming that systems can achieve high precision—let's say, 90% or greater—are more notifications really better? Intuitively, one would expect that, at some point, user fatigue sets in, even for a stream of high-quality tweets. We might imagine the user having access to a "volume dial" to provide feedback: "yes, these are all good tweets, but too many!" As we have shown, tweet volume is not directly captured in existing metrics, but the problem lies deeper: our understanding of how users perceive notifications in response to prospective information needs remains quite poor, especially when factoring in the cost of interruptions. More work on fundamental issues along these lines is needed.

8 ACKNOWLEDGMENTS

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, with additional contributions from the U.S. National Science Foundation under IIS-1218043 and CNS-1405688. Any findings, conclusions, or recommendations expressed do not necessarily reflect the views of the sponsors.

REFERENCES

[1] Azzah Al-Maskari, Mark Sanderson, Paul Clough, and Eija Airio. 2008. The Good and the Bad System: Does the Test Collection Predict Users' Effectiveness? In *SIGIR*. 59–66.

[2] James Allan. 2002. *Topic Detection and Tracking: Event-Based Information Organization*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

[3] James Allan, Ben Carterette, and Joshua Lewis. 2005. When Will Information Retrieval Be "Good Enough"? User Effectiveness as a Function of Retrieval Accuracy. In *SIGIR*. 433–440.

[4] Javed Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Tetsuya Sakai. 2015. TREC 2015 Temporal Summarization Track Overview. In *TREC*.

[5] Peter Bailey, Nick Craswell, Ian Soboroff, Paul Thomas, Arjen P. de Vries, and Emine Yilmaz. 2008. Relevance Assessment: Are Judges Exchangeable and Does it Matter? In *SIGIR*. 667–674.

[6] Nicholas J. Belkin and W. Bruce Croft. 1992. Information Filtering and Information Retrieval: Two Sides of the Same Coin? *CACM* 35, 12 (1992), 29–38.

[7] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. 2012. Large-Scale Validation and Analysis of Interleaved Search Evaluation. *ACM TOIS* 30, 1 (2012), Article 6.

[8] Qi Guo, Fernando Diaz, and Elad Yom-Tov. 2013. Updating Users about Time Critical Events. In *ECIR*. 483–494.

[9] Allan Hanbury, Henning Müller, Krisztian Balog, Torben Brodt, Gordon V. Cormack, Ivan Eggel, Tim Gollub, Frank Hopfgartner, Jayashree Kalpathy-Cramer, Noriko Kando, Anastasia Krithara, Jimmy Lin, Simon Mercer, and Martin Potthast. 2015. Evaluation-as-a-Service: Overview and Outlook. *arXiv:1512.07454*.

[10] William Hersh, Andrew Turpin, Susan Price, Benjamin Chan, Dale Kramer, Lynetta Sacherek, and Daniel Olson. 2000. Do Batch and User Evaluations Give the Same Results? In *SIGIR*. 17–24.

[11] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2011. A Probabilistic Method for Inferring Preferences from Clicks. In *CIKM*. 249–258.

[12] Ron Kohavi, Randal M. Henne, and Dan Sommerfield. 2007. Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO. In *KDD*. 959–967.

[13] David D. Lewis. 1995. The TREC-4 Filtering Track. In *TREC*. 165–180.

[14] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. 2014. Overview of the TREC-2014 Microblog Track. In *TREC*.

[15] Jimmy Lin, Miles Efron, Yulu Wang, Garrick Sherman, and Ellen Voorhees. 2015. Overview of the TREC-2015 Microblog Track. In *TREC*.

[16] Jimmy Lin, Adam Roegiest, Luchen Tan, Richard McCreadie, Ellen Voorhees, and Fernando Diaz. 2016. Overview of the TREC 2016 Real-Time Summarization Track. In *TREC*.

[17] Abhinav Mehrotra, Veljko Pejovic, Jo Vermeulen, Robert Hendley, and Mirco Musolesi. 2016. My Phone and Me: Understanding People's Receptivity to Mobile Notifications. In *CHI*. 1021–1032.

[18] Xin Qian, Jimmy Lin, and Adam Roegiest. 2016. Interleaved Evaluation for Retrospective Summarization and Prospective Notification on Document Streams. In *SIGIR*. 175–184.

[19] Filip Radlinski and Nick Craswell. 2010. Comparing the Sensitivity of Information Retrieval Metrics. In *SIGIR*. 667–674.

[20] Filip Radlinski and Nick Craswell. 2013. Optimized Interleaving for Online Retrieval Evaluation. In *WSDM*. 245–254.

[21] Stephen Robertson and Ian Soboroff. 2002. The TREC 2002 Filtering Track Report. In *TREC*.

[22] Alan Said, Jimmy Lin, Alejandro Bellogín, and Arjen P. de Vries. 2013. A Month in the Life of a Production News Recommender System. In *CIKM Workshop on Living Labs for Information Retrieval Evaluation*. 7–10.

[23] Mark Sanderson, Monica Paramita, Paul Clough, and Evangelos Kanoulas. 2010. Do User Preferences and Evaluation Measures Line Up? In *SIGIR*. 555–562.

[24] Anne Schuth, Krisztian Balog, and Liadh Kelly. 2015. Overview of the Living Labs for Information Retrieval Evaluation (LL4IR) CLEF Lab 2015. In *CLEF*.

[25] Anne Schuth, Katja Hofmann, and Filip Radlinski. 2015. Predicting Search Satisfaction Metrics with Interleaved Comparisons. In *SIGIR*. 463–472.

[26] Mark Smucker and Chandra Jethani. 2010. Human Performance and Retrieval Precision Revisited. In *SIGIR*. 595–602.

[27] Ian Soboroff, Iadh Ounis, Craig Macdonald, and Jimmy Lin. 2012. Overview of the TREC-2012 Microblog Track. In *TREC*.

[28] Luchen Tan, Adam Roegiest, Jimmy Lin, and Charles L. A. Clarke. 2016. An Exploration of Evaluation Metrics for Mobile Push Notifications. In *SIGIR*. 741–744.

[29] Andrew Turpin and William R. Hersh. 2001. Why Batch and User Evaluations Do Not Give the Same Results. In *SIGIR*. 225–231.

[30] Andrew Turpin and Falk Scholer. 2006. User Performance versus Precision Measures for Simple Search Tasks. In *SIGIR*. 11–18.

[31] Yulu Wang, Garrick Sherman, Jimmy Lin, and Miles Efron. 2015. Assessor Differences and User Preferences in Tweet Timeline Generation. In *SIGIR*. 615–624.