# A Platform for Streaming Push Notifications to Mobile Assessors

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

## ABSTRACT

We present an assessment platform for gathering online relevance judgments for mobile push notifications that will be deployed in the newly-created TREC 2016 Real-Time Summarization (RTS) track. There is emerging interest in building systems that filter social media streams such as tweets to identify interesting and novel content in real time, putatively for delivery to users' mobile phones. In our evaluation design, all participants subscribe to the Twitter streaming API to identify relevant tweets with respect to a set of interest profiles. As the systems generate results, they are pushed in real time to our evaluation broker via a REST API. The broker then "routes" the tweets to assessors who have installed a custom app on their mobile phones. We detail the design of this platform and discuss a number of challenges that need to be tackled in this type of "Living Labs" setup. It is our goal that such an evaluation design will mitigate any issues that have arisen in traditional batch-style evaluations of this type of task.

## 1. INTRODUCTION

There is emerging interest in building push notification systems that filter social media streams, such as Twitter, to deliver relevant content to users' mobile phones. For example, the user might be a political news junkie interested in polls for the 2016 U.S. presidential elections and wishes to be notified whenever new results are posted on Twitter. She might also be interested in commentary by political pundits and reactions by the candidates. Such notifications must relevant (i.e., on topic), timely (i.e., the user desires poll results as soon as they are available), and novel (i.e., the user does not want tweets from multiple sources that cite the same poll). Techniques to address such information needs are becoming increasingly important as mobile devices continue to gain prominence for information access.

The TREC Microblog track in 2015 operationalized the push notification task in the so-called "scenario A" variant of the real-time filtering task [3]. Over the official evaluation period, which spanned ten days during July 2015, participating systems "listened" to Twitter's live tweet sample stream to identify relevant tweets with respect to 225 pre-defined "interest profiles" (each expressed through statements modeled after TREC *ad hoc* topics), 51 of which were later assessed. Each system identified up to ten tweets per day, which were ostensibly delivered to hypothetical users. In total, 14 groups submitted 37 runs to this evaluation.

As it was the first year, many aspects of the evaluation in 2015 represented substantial simplifications of the actual task. Instead of pushing notifications in real time, participating systems merely recorded the wall clock time at which the system *would have* pushed the notification. That is, all results were recorded locally and submitted in batch *after* the evaluation period ended. Thus, although the task itself is real time, the results were evaluated with standard batch protocols (pooling followed by semantic clustering; see Wang et al. [9] for more details). The goal of our work is to tackle this limitation: we present an assessment platform for gathering online relevance judgments for mobile push notifications. This platform will be deployed in the newly-created TREC 2016 Real-Time Summarization (RTS) track.

Real-Time Summarization at TREC 2016 represents a merger of the Microblog track, which ran from 2010 to 2015, and the Temporal Summarization track [1], which ran from 2013 to 2015. The creation of RTS was designed to leverage synergies between the two tracks in exploring prospective information needs over document streams containing novel and evolving information. Previously, Temporal Summarization evaluations operated by *simulating* a stream of documents using a static collection, which also represented a simplification of the underlying task model. We believe that by "joining forces", we could develop a more refined push notification task and associated evaluation infrastructure to support online user-in-the-loop evaluations, thereby growing the research community and pushing forward the state of the art.

## 2. GENERAL ARCHITECTURE

The TREC 2016 Real-Time Summarization track will operationalize a push notification task that largely follows the same task from the TREC 2015 Microblog track. Participants must deploy working systems that consume a live stream over a defined evaluation period. In 2016, this live stream will be comprised solely of tweets provided by the Twitter streaming API, but we are investigating the viability of alternate and additional sources, e.g., a stream of news stories or blog posts. The exact task definition is not

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

ISBN 978-1-4503-4069-4/16/07. . . $15.00
DOI: http://dx.doi.org/10.1145/2911451.2911463

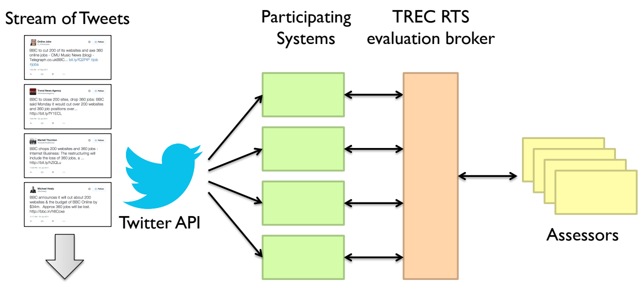


Figure 1: A high-level overview of the real-time push notification assessment platform.

relevant for the purposes of this work, which focuses on the assessment platform. The most substantial change from the previous evaluation is that systems now must submit their results in real time. This live submission is accomplished via a REST API and evaluation infrastructure that we have developed. The entire platform, including both server code and the mobile assessment app, is available open source.[1]

The high-level architecture of our assessment platform is shown in Figure 1. Our general approach builds on growing interest in so-called Living Labs [6] and related Evaluation-as-a-Service (EaaS) [2] approaches that attempt to better align evaluation methodologies with user task models and real-world constraints to increase the fidelity of research experiments. In our architecture, participating systems all subscribe to the Twitter streaming API (a sample stream is freely available to all registered users) to identify relevant tweets with respect to interest profiles. Since tweets are being posted in real time, the evaluation organizers do not distribute any data ahead of time—they listen to the stream just like all participants to gather an archival copy of the tweets. A pilot study in 2015 [4] confirmed that multiple geographically-distributed listeners to the public Twitter sample stream receive effectively the same tweets (Jaccard overlap of 0.999 across six independent crawls over a three day sample in March 2015). As the participating systems identify relevant tweets, they are pushed in real time to the evaluation broker, which then routes the tweets to assessors who have installed a custom app on their mobile phones. We intend to recruit students to serve as the assessors.

This setup has a number of distinct advantages:

- Gathering relevance judgments in an online fashion has the potential to yield 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.
- An online evaluation platform allows for the possibility of user-submitted information needs, thus giving assessors the ability to judge tweets for interest profiles they are genuinely interested in.
- An online evaluation platform opens the door to providing realistic, online feedback to participants, thus potentially facilitating active learning approaches. There are, of course, many additional complexities that remain unresolved (discussed below), and in the near term we do not anticipate providing this option.

Note that online evaluation using our assessment platform does not preclude the use of standard batch evaluation methodologies that have been well studied and appropriately validated. Indeed, it is the plan that the Real-Time Summarization track in TREC 2016 will also apply the batch evaluation methodology developed for the Microblog track in 2015 in a post hoc manner. This dual evaluation approach will help us validate the reliability and stability of our online mobile assessment platform.

# 3. PLATFORM COMPONENTS

## 3.1 Evaluation Workflow

We have designed our assessment platform around the following workflow:

1. Systems listen to the live Twitter sample stream to identify interesting and novel tweets with respect to a set of pre-defined interest profiles. Although we provide boilerplate code to get started, participants are responsible for building and running their own systems.
2. Whenever a system identifies a relevant tweet with respect to an interest profile, the system submits the result to the evaluation broker via a REST API, which records the submission time.
3. The broker routes the tweet to the mobile phone of an assessor, where it is rendered as a push notification containing both the text of the tweet and the corresponding interest profile.
4. The assessor may choose to judge the tweet immediately, or if it arrives at an inopportune time, to ignore it. Either way, the tweet is added to a judging queue in a custom app on the assessor's mobile phone, which she can access at any time to judge the queue of accumulated tweets.
5. As the assessor judges tweets, the results are relayed back to the evaluation broker and recorded.

While the information retrieval community is already familiar with a variety of batch relevance assessment approaches, our platform explores a largely untouched space in evaluation design. We anticipate that a number of experimental design decisions will need to be empirically verified:

- **Number of assessors**: The size of the assessor pool will be largely dependent on the number of topics, participating systems, and the incentive structure for providing judgments.
- **Topic assignment**: There are many possibilities for mapping between topics and assessors. As with standard pooled batch evaluations, assigning all tweets for a given topic to a single assessor increases the consistency of judgments, but might result in an unbalanced load across assessors. Alternatively, we could split each topic across multiple assessors, but this approach may increase inconsistency.
- **Assessor modeling**: It is likely that assessors will vary in quality, will respond to push notifications with varying degrees of latency, and will provide differing numbers of judgments in a given time period. We anticipate the need to construct post hoc assessor models in order to understand and account for these assessor differences.

[1] https://github.com/trecrts/trecrts-eval

- **Tweet interleaving**: It is unlikely that we will be able to devote an assessor to exclusively judge the output of a single system. Nor would this approach be desirable, since it would magnify the impact of assessor differences when performing system comparisons. Thus, assessors will see interleaved results from multiple systems. We have been separately investigating tweet interleaving strategies for evaluation [5], and we will apply the lessons learned from those experiments as appropriate.

## 3.2 Evaluation Broker

The evaluation broker (see Figure 1) serves as an intermediary between systems participating in the evaluation and the mobile assessors. The main role of the broker is to distribute interest profiles to participating systems, record system submissions, route submitted results to mobile assessors, and record judgments rendered by the assessors.

The broker is implemented in Node.js and backed by a MySQL database for persistent storage of result submissions and assessor judgments. Broker functionalities are implemented via different REST API endpoints. For example, systems submit a tweet for a particular interest profile using the following call:

```
POST /tweet/:topid/:tweetid/:clientid
```

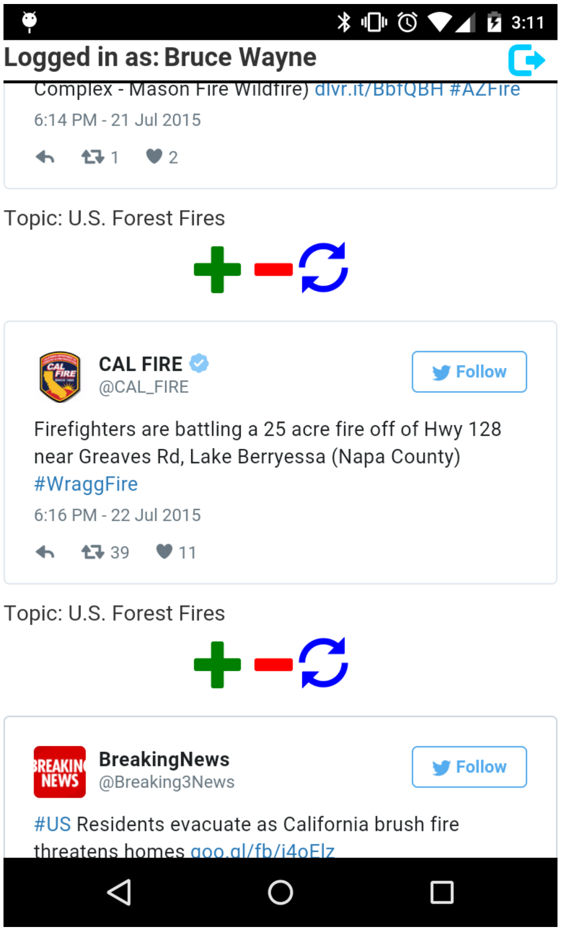
where `:topid` specifies the topic (interest profile) identifier, `:tweetid` specifies the unique tweet identifier of the post, and `:clientid` specifies the client's unique identifier. The broker returns a 204 status code on success. It is expected that all participating systems will properly interface with the appropriate API endpoint during the evaluation period.

Currently, we have designed the broker to rate-limit the number of submissions by a system to ten tweets per topic per day, following the TREC 2015 Microblog track protocol. The broker further employs some common sense safeguards, e.g., to not bombard any individual assessor with an undue number of push notifications. The actual routing policy of how tweets are assigned to assessors is still currently under development, although we have already implemented a basic round robin approach.

## 3.3 Mobile Assessment App

For TREC 2016, we plan to recruit students from the University of Waterloo to serve as mobile assessors in a user study centered around the Real-Time Summarization task. As discussed, assessors will receive tweets as they are identified by participating systems in real time, on their mobile phones as push notifications through our custom app. A screenshot of the current app is shown in Figure 2. We envision the experimental study to proceed as follows:

- Assessors will be given a brief description of the task and an invitation to install the assessment app on their mobile phones.
- Assessors will log in to the app to indicate that they are ready to receive tweets. They may log out to stop receiving tweets when they unavailable (e.g., during an exam).
- Assessors will receive new tweets to judge as they become available (i.e., are pushed by participating systems). Tweets will be assessed with respect to the interest profile provided with the tweet.



3:11

Logged in as: Bruce Wayne

Complex - Mason Fire Wildfire) dlvr.it/BbfQBH #AZFire

6:14 PM - 21 Jul 2015

1 2

Topic: U.S. Forest Fires

+ − ↻

CAL FIRE
@CAL_FIRE

Follow

Firefighters are battling a 25 acre fire off of Hwy 128 near Greaves Rd, Lake Berryessa (Napa County) #WraggFire

6:16 PM - 22 Jul 2015

39 11

Topic: U.S. Forest Fires

+ − ↻

BreakingNews
@Breaking3News

Follow

#US Residents evacuate as California brush fire threatens homes goo.gl/fb/i4oElz

Figure 2: Screenshot of the mobile assessment app.

- At any point, assessors will be able to end their participation in the task and to opt out of further work. After ending participation they will receive a code that can be exchanged for remuneration (at a rate yet to be determined).
- Otherwise, assessors will be able to judge tweets until the end of the evaluation period, after which they will receive the renumeration code.

We have built the mobile assessment app for both iOS and Android using the Cordova framework. Cordova is designed to leverage existing web technologies for the creation of mobile apps without requiring extensive platform-specific API knowledge. This makes cross-platform development (Android and iOS) significantly less labor-intensive compared to writing two separate native apps. The tradeoff is performance, which is not a concern for us since our assessment task is not processing intensive.

Note that the broker does not directly deliver tweets to the assessors. Rather, tweet identifiers are sent to the mobile phone and tweets are rendered by the assessment app from these identifiers via Twitter's OEmbed API,[2] which displays any inline media content for the tweet. This mechanism was adopted to comply with the Twitter terms of service, which prohibit the distribution of tweet content, but

[2] https://dev.twitter.com/rest/reference/get/statuses/oembed

the mechanism also allows us to offload tweet rendering (including complex multimedia content) to the device. Using OEmbed, we can render a tweet natively with minimal effort yet still provide a user experience similar to that of the official Twitter client.

# 4. BASELINE SYSTEM

Although the main focus of this demonstration is the mobile assessment platform itself, we also present a baseline implementation to facilitate system development and participation in the track. Our baseline system, called YoGosling, is a modified and extended version of the system that generated the best performing automatic run in TREC 2015 [7]. Following TREC, we performed error analysis and ablation studies to distill the original system down to the components that contributed most to overall effectiveness [8]. This system is built on the Anserini project, which is the University of Waterloo's Lucene-based search framework. The system is able to incrementally index the Twitter sample stream and provide real-time search capabilities.

YoGosling converts the title field of interest profiles into queries for searching Lucene and applies a simple relevance scoring method to rank tweets. After relevance scoring, duplicate and near-duplicate tweets are identified by a novelty detection component (based on simple Jaccard similarity), so that users are not given repetitiously annoying information. One of the most important lessons learned in building YoGosling is the proper setting of score thresholds and ignoring tweets that fall below the thresholds. Our simple scoring model is amenable to a *global* threshold that yields reasonable effectiveness, thus obviating the need for per-topic tuning (difficult due to the paucity of training data). Another interesting feature of YoGosling is a simple relevance feedback mechanism whereby users assess tweets once a day, and these judgments are used to set the score threshold for the next day. We experimentally show that this technique can yield substantial gains in effectiveness [8].

# 5. FUTURE WORK

One of the main enhancements we are planning to add is the ability for assessors to supply their own interest profiles directly through an API, so that they can actually receive tweets of personal interest. In TREC 2016, we plan to develop interest profiles manually based on assessor input, but the profiles will be vetted by the track organizers before distribution to participating systems. An automatic API for interest profile submission creates several non-trivial problems, including near-duplicate detection, limiting the release of personal information, and topic termination (since some topics may only be important for a limited time). These issues are exacerbated if there is the expectation that behavioral traces will be released as part of a training set (as is the case with TREC evaluations), and ethics issues similar to the public release of web query logs come into focus.

At present, we have a few ideas of how to assign interest profiles to assessors, but lack concrete empirical evidence as to what strategy will actually "work". As discussed, foreseeable issues include inter-assessor consistency, assessor latency (how quickly they respond to push notifications), and assessment volume (how many judgments they ultimately provide). Related, the strategy used for interleaving tweets has the potential to affect the judgments rendered, and in turn the relative scoring of systems. Assessor differences introduce additional confounds. We have been exploring in parallel some of these questions via simulations [5], but the literature otherwise offers little guidance and we will have to make decisions about evaluation design for TREC 2016 based on limited empirical evidence.

# 6. CONCLUSIONS

In this demonstration, we described the architecture of an assessment platform for gathering online relevance judgments for mobile push notifications that will be deployed in the TREC 2016 Real-Time Summarization (RTS) track. Although the infrastructure is "feature complete" in terms of implementation, we freely admit that there are many unresolved issues regarding evaluation design. There is no substitute for operational experience in running a TREC evaluation, and we anticipate many lessons to be learned.

# 7. ACKNOWLEDGMENTS

This work was supported in part by the U.S. National Science Foundation under awards IIS-1218043 and CNS-1405688 and the Natural Sciences and Engineering Research Council of Canada (NSERC). Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the sponsors.

# 8. REFERENCES

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

[2] A. Hanbury, H. Müller, K. Balog, T. Brodt, G. V. Cormack, I. Eggel, T. Gollub, F. Hopfgartner, J. Kalpathy-Cramer, N. Kando, A. Krithara, J. Lin, S. Mercer, and M. Potthast. Evaluation-as-a-Service: Overview and outlook. *arXiv:1512.07454*, 2015.

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

[4] J. H. Paik and J. Lin. Do multiple listeners to the public Twitter sample stream receive the same tweets? *SIGIR Workshop on Temporal, Social and Spatially-Aware Information Access*, 2015.

[5] X. Qian, J. Lin, and A. Roegiest. Interleaved evaluation for retrospective summarization and prospective notification on document streams. *SIGIR*, 2016.

[6] A. Schuth, K. Balog, and L. Kelly. Overview of the Living Labs for Information Retrieval Evaluation (LL4IR) CLEF Lab 2015. *CLEF*, 2015.

[7] L. Tan, A. Roegiest, and C. L. A. Clarke. University of Waterloo at TREC 2015 Microblog track. *TREC*, 2015.

[8] L. Tan, A. Roegiest, C. L. A. Clarke, and J. Lin. Simple dynamic emission strategies for microblog filtering. *SIGIR*, 2016.

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