Event Detection on Curated Tweet Streams

Nimesh Ghelani, Salman Mohammed, Shine Wang, and Jimmy Lin
David R. Cheriton School of Computer Science
University of Waterloo, Ontario, Canada
{nghelani,salman.mohammed,shine.wang,jimmylin}@uwaterloo.ca

ABSTRACT

We present a system for identifying interesting social media posts on Twitter and delivering them to users' mobile devices in real time as push notifications. In our problem formulation, users are interested in broad topics such as politics, sports, and entertainment: our system processes tweets in real time to identify relevant, novel, and salient content. There are three interesting aspects to our work: First, instead of attempting to tame the cacophony of unfiltered tweets, we exploit a smaller, but still sizeable, collection of curated tweet streams corresponding to the Twitter accounts of different media outlets. Second, we apply distant supervision to extract topic labels from curated streams that have a specific focus, which can then be leveraged to build high-quality topic classifiers essentially "for free". Finally, our system delivers content via Twitter direct messages, supporting *in situ* interactions modeled after conversations with intelligent agents. These ideas are demonstrated in an end-to-end working prototype.

1 INTRODUCTION

We present a system for identifying interesting social media posts on Twitter and delivering them to users' mobile devices in real time as push notifications. We assume that users are interested in broad topics such as politics, technology, sports, or entertainment, and wish to keep track of "what's happening" in real time. At a high level, these updates must be relevant (actually related to the appropriate topic), salient (of significant interest), novel (not repeating previous messages), and timely (delivered as soon as possible after the event has occurred).

Our problem formulation is related to work on prospective information needs, as exemplified by the Temporal Summarization [2], Microblog [7], and Real-Time Summarization [8] evaluations at recent Text Retrieval Conferences (TRECs). However, our setup is different in that these evaluations tackle specific information needs, akin to topics in traditional *ad hoc* retrieval, whereas we focus on much broader topical categories of content.

We present an end-to-end prototype that monitors curated Twitter streams, identifies relevant, salient, and novel content, and delivers updates to users via Twitter direct messages. There are three interesting aspects to our work:

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

978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3084141

Event detection on curated streams. Although event detection on Twitter is a well-trodden area (e.g., [3, 4, 12]), we take a completely different, novel approach: instead of trying to tame the cacophony of unfiltered posts by millions of Twitter users, we exploit a smaller, but still sizeable, collection of curated streams corresponding to the accounts of different media outlets. Posts in these streams comprise the content from which we select for delivery to users.

Distant supervision for topic classification. Curated streams vary in their topical focus: some accounts have a narrow focus, i.e., they tweet only about entertainment news, while others have broad coverage, i.e., they tweet about anything newsworthy. Since in our problem formulation users "subscribe" only to categories of interest, topic classifiers are needed to further filter the curated streams. We take advantage of a novel distant supervision technique for automatically gathering noisy category labels from topically-focused streams. These can be used to train topic classifiers and applied to topically-diffuse streams to retain only those tweets that a user might be interested in.

Native content delivery and interaction support. Inspired by intelligent conversational agents such as Siri and Cortana, our system introduces a novel method for delivering push notifications to users through direct messages on Twitter itself. Users can interact with our system on the same platform, which also provides a convenient channel for gathering relevance judgments.

This paper describes an end-to-end working prototype that illustrates the above ideas. Qualitative evaluation of our system in comparison with trending stories identified by the Twitter Moments product shows that our system is effective in identifying salient tweets before they appear on Twitter Moments.

2 SYSTEM DESIGN

The starting point of nearly all event detection work on Twitter is the unfiltered torrent of tweets collectively generated by millions of users. In general, researchers try to obtain, by whatever means, as many tweets as possible—the more tweets, the better. From this cacophony, the system tries to identify events, "trending" topics, or whatever is interesting and "happening". Such a needle-in-a-haystack approach is noisy and prone to manipulation (fake news, "astro-turfing", etc.).

Our work adopts a completely different approach: we begin with the observation that there already exist many human-curated streams of interesting events, corresponding to the Twitter accounts of various media outlets. The news editors at CNN, for example, tweet breaking news from @cnn and related accounts. Almost every media outlet, large and small, has their own Twitter account. We wonder, why not build event detection on these curated streams? Especially for "head events" of broad interest to large populations of

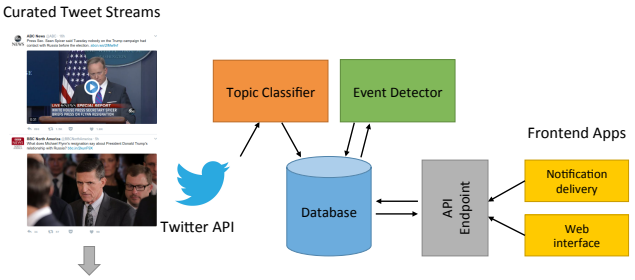

Figure 1: Overall architecture of our system.

users, such an approach seems intuitive. In addition, this approach skirts many thorny issues in event detection, such as the definition of an event, which has been the subject of much debate dating back over a decade [1]. To us, an event is simply what the editors of the underlying curated streams deem interesting. Nevertheless, there remains two challenges:

- First, although our techniques operate on curated streams of tweets, the combined volume of these streams is still beyond what any human can consume. Furthermore, there are many duplicate tweets corresponding to reports by different media outlets. Thus, even over curated streams, we must still identify what the *salient* and *novel* tweets are.
- Second, curated streams vary in their topical focus. Some accounts have a narrow focus, e.g., they tweet only about business news, while others have broad coverage, i.e., they tweet about anything that is newsworthy. Since users are often only interested in particular topics, we need topic classifiers to properly categorize content.

Our prototype addresses both challenges with the architecture shown in Figure 1. The Twitter API is used to subscribe to curated streams corresponding to the accounts of various media outlets (more details below). Observed tweets are sorted into different categories by the topic classifier and stored in a MongoDB collection. The event detector runs a sliding window over tweets from the database: salient and novel tweets are written to another database collection. A Flask[1] server provides a REST API for frontend applications to query the database. We've implemented two separate frontends that poll this API endpoint: a web-based interface for displaying interesting events and a push notification delivery mechanism that takes advantage of the Twitter direct message platform. We detail each of these components below.

2.1 Topic Classification

Facebook published an article in May 2016 providing an overview of their Trending Topics algorithm [11]. The article provided a list of RSS URLs, mapped to countries and topics, that their algorithm uses to identify breaking events. Most of those URLs correspond to popular media outlets such as CNN and ESPN. We used the Google Search API to find Twitter accounts associated with the domains of those URLs. Although the Facebook data contained RSS feeds in many languages, in this work we only focus on English. Based on a few simple heuristics and manual verification, we obtained a list of 293 Twitter accounts corresponding to media outlets in

[1]http://flask.pocoo.org/

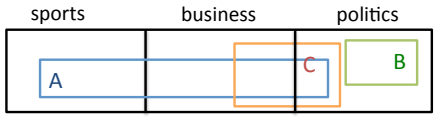

Figure 2: Topic grid showing accounts with different coverage. Tweets from B provide a source of topic labels "for free".

the Facebook dataset. Tweets from these accounts serve as the input to our system. Note that collectively, these accounts post a volume of tweets that would be impossible for a human to consume directly—over an evaluation period of 21 days from late 2016, we observed an average of around 16,000 tweets per day.

Our first challenge is to categorize all of these tweets into topics that users might be potentially interested in: business, politics, health, science, tech, sports, entertainment, and gaming. The streams, however, vary in their topical focus. Some accounts have a *narrow* focus while others have a *broad* focus. This is illustrated by the topic grid in Figure 2; here, we see that stream B has a narrow focus on politics, whereas streams A and C have a broader focus. As a specific example, @espn tweets almost exclusively about sports, whereas @cnn posts about nearly everything.

For event detection, we benefit from broad coverage accounts for signal, but we must develop topic classifiers to discard tweets that a particular user would not be interested in. We can exploit tweets from *narrow* accounts to train topic classifiers using distant supervision, which can then be used to classify tweets from *broad* accounts—thus maximizing both coverage as well as relevance.

This approach, which builds on previous work applying distant supervision to social media [5, 6, 9, 16], is detailed in a separate paper [10]. As our system simply uses this technique as a component, we refer the reader to the detailed exposition, but here we provide a brief summary: Our classifier gathers distantly-supervised labels as described above to train a logistic regression classifier using scikit-learn, based on tf-idf features from tweet text. To combat topic drift, we reweight the training data, placing higher weights on more recent data. In the current setup, the topic classifier is trained on tweets from the past 30 days and retrained every hour.

2.2 Event Detection

Having classified tweets into topics, our next two challenges are to discover *salient* and *novel* content. We define saliency as the property characterizing tweets that are of interest to users. In our formulation, saliency is independent and distinguished from relevance, in that a tweet may be on topic (i.e., about entertainment) but not worth delivering to users as a push notification—for example, news about a minor celebrity. At the same time, we must ensure that our system does not push multiple updates that say the same thing, i.e., notifications must be novel. This is a real concern because breaking news stories are frequently reported by multiple media outlets within a short span of time.

Our current approach attempts to address both issues simultaneously: our intuition is that an event is salient if we observe similar content from multiple accounts within a short amount of time. That is, if multiple media outlets tweet about the same thing, those tweets are likely to be of interest. At the same time, if we are able to identify multiple tweets that say the same thing, then by

definition we have addressed the novelty problem. In other words, we try to kill two birds with one stone.

This idea is operationalized in a simple yet effective algorithm. Event detection is performed on a sliding window over tweets posted within the past 30 minutes (for a particular topic). We consider a tweet to be salient if there are at least K other tweets in the window that are paraphrases of the tweet under consideration (i.e., posted by other media outlets). Two tweets are considered paraphrases of each other if their Jaccard similarity is above some threshold, i.e., $\text{sim}_{\text{Jaccard}}(\text{tweet}_a, \text{tweet}_b) > T$, based on the NLTK Tweet Tokenizer. This algorithm runs every minute.

A salient tweet is also novel if it is not redundant with respect to any tweet that was pushed within the last 24 hours. Again, we measure redundancy via Jaccard similarity, but the threshold for identifying duplicate content (Q) is different from the threshold for identifying a salient tweet (T).

Intuitively, our algorithm identifies as a salient tweet the first post that is "confirmed" by multiple media outlets (as controlled by the K and T parameters). The Q parameter allows us to control how likely we are to report subsequent developments of the same news story. Based on qualitative inspection of output, we set the parameters in our prototype as follows: $K = 6, T = 0.35, Q = 0.2$.

2.3 Content Delivery

Tweets identified as salient and novel are stored in our database by topic (see Figure 1). An API endpoint provides a method for frontend applications to access the stored content. Currently, we have built two such applications, described below.

Direct Messages. Direct messages are private channels in Twitter for multi-party conversations, similar to messaging apps such as Facebook Messenger or WeChat. Users, however, need not be conversing with other humans—in fact, the direct message platform can be used to build software agents. The emergence of intelligent conversational agents such as Siri and Cortana, as well as the prevalence of software "chatbots", suggests that such forms of interaction are widely accepted today.

Building on this idea, our system delivers push notifications to users via a software agent, represented by a Twitter account that users follow and sign up for notifications via direct messages. There are several advantages to this content delivery approach: Since we are using the Twitter platform itself, this *in situ* notification mechanism presents a seamless user experience. There is no need for specialized mobile apps (e.g., see [15]) that we would have to write ourselves for multiple mobile operating systems—our notification mechanism will run on any platform for which there is already a Twitter client. This tight integration also allows users to have more fine-grained control over how the notifications are actually rendered on their mobile devices (for example, whether the notification is accompanied by an audible chime), and to adjust these settings accordingly if they do not wish to be disturbed.

Finally, the direct message platform provides a mechanism for interacting with our system. To initiate communications with the agent, the user can just send an arbitrary message and the agent will reply with a *help* message describing available commands. The user can subscribe and unsubscribe to topics via the text interface. Furthermore, direct messages provide a channel through which

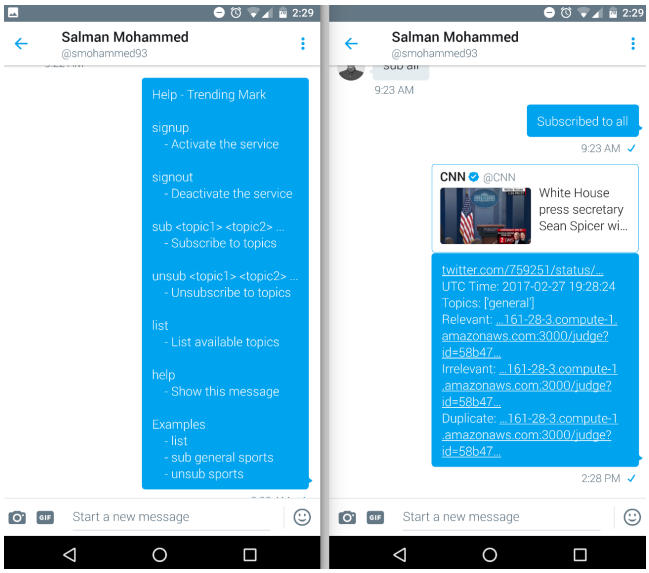

Figure 3: Delivery of push notifications and support for interactions via Twitter direct messages.

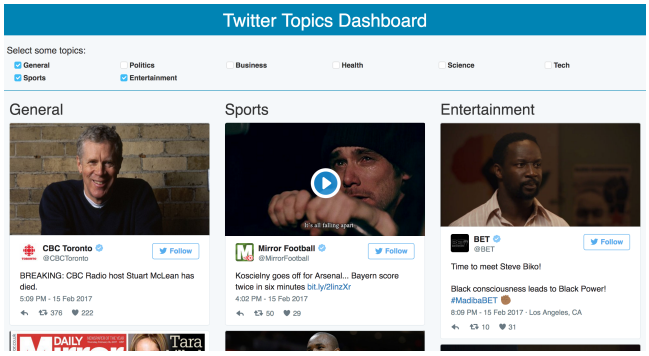

Figure 4: Screenshot of the web portal displaying salient and novel tweets for different topic categories.

users can supply judgments to help us evaluate the quality of the pushed content (cf. [13, 14]). See Figure 3 for example interactions.

Web portal. In addition to push notification delivery via the Twitter direct message platform, we have also built a web portal that users can visit to browse "what's happening" in multiple topic categories. A screenshot of this interface is shown in Figure 4. Based on users' topic selections, the web portal calls the API endpoint to request the desired information and continues to poll the server to dynamically update the user interface whenever salient and novel tweets are discovered.

3 CURRENT DEPLOYMENT

We are currently working on a live deployment of our end-to-end prototype with a group of users to evaluate the system in a rigorous manner. Here, we present an informal qualitative comparison of our system's output against the Twitter Moments product,[2] which

[2]https://twitter.com/i/moments

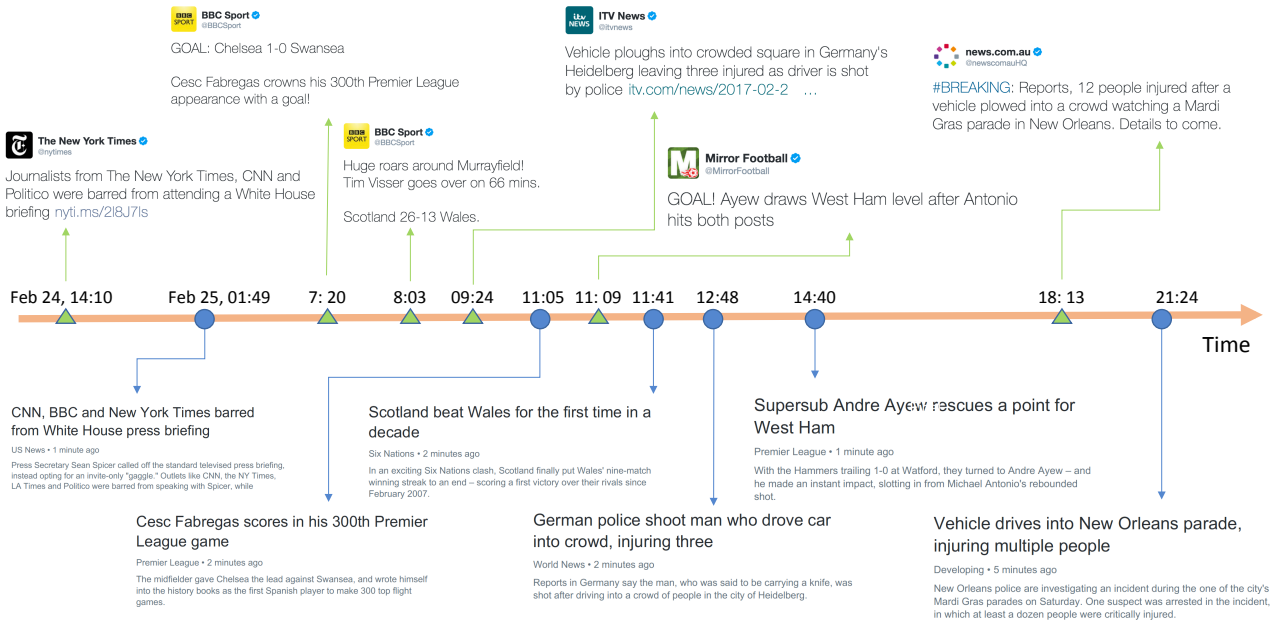

Figure 5: Timeline comparing tweets identified by our system and stories taken from Twitter Moments on February 25, 2017.

lists curated stories to showcase current trending events on Twitter. "Moments" occupies a tab in the top-level Twitter interface and is divided into five different sections: Today, News, Sports, Entertainment, Fun. The "Today" section is just a summary of prominent stories from the other sections. For our purposes, "News", "Sports", and "Entertainment" are the most relevant since they directly correspond to our topic categories.

We wrote a script to scrape the Twitter Moments page at five minute intervals on February 25, 2017 from 00:00 to 23:59 and used these results to qualitatively assess the output of our system. Figure 5 shows a timeline containing a selection of salient and novel tweets (green triangles) identified by our system and corresponding stories taken from Twitter Moments (blue circles) for comparison purposes. In these cases, our system identifies the trending tweets a few hours before they appear on Twitter Moments. We noticed, in fact, that our system achieves broader coverage than Twitter Moments for the entertainment category—most likely due to the limited screen real estate in Twitter's interface and the abundance of entertainment content on Twitter in general. Of course, this is not a rigorous evaluation, but overall the simple techniques presented in this paper appear to perform well.

4 CONCLUSION

Our work takes a different approach to event detection on Twitter compared to most existing techniques: instead of trying to sift through as many tweets as possible, we take advantage of human-curated streams created by various media outlets. In a sense, much of the work has already been done for us: each account posts tweets corresponding to what human editors deem interesting, and our system's primary task is to synthesize and aggregate these decisions. This problem formulation leads to a different set of challenges, to which we present simple yet effective solutions.

REFERENCES

[1] James Allan. 2002. *Topic Detection and Tracking: Event-Based Information Organization.* Kluwer Academic Publishers, Dordrecht, The Netherlands.
[2] Javed Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Tetsuya Sakai. 2015. TREC 2015 Temporal Summarization Track Overview. In *TREC*.
[3] Farzindar Atefeh and Wael Khreich. 2015. A Survey of Techniques for Event Detection in Twitter. *Computational Intelligence* 31, 1 (2015), 132–164.
[4] Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond Trending Topics: Real-World Event Identification on Twitter. In *ICWSM*. 438–441.
[5] Stephanie D. Husby and Denilson Barbosa. 2012. Topic Classification of Blog Posts Using Distant Supervision. In *Workshop on Semantic Analysis in Social Media*. 28–36.
[6] Sheila Kinsella, Alexandre Passant, and John G. Breslin. 2011. Topic Classification in Social Media Using Metadata from Hyperlinked Objects. In *ECIR*. 201–206.
[7] Jimmy Lin, Miles Efron, Yulu Wang, Garrick Sherman, and Ellen Voorhees. 2015. Overview of the TREC-2015 Microblog Track. In *TREC*.
[8] 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*.
[9] Walid Magdy, Hassan Sajjad, Tarek El-Ganainy, and Fabrizio Sebastiani. 2015. Distant Supervision for Tweet Classification Using YouTube Labels. In *ICWSM*. 638–641.
[10] Salman Mohammed, Nimesh Ghelani, and Jimmy Lin. 2017. Distant Supervision for Topic Classification of Tweets in Curated Streams. *arXiv:1704.06726*.
[11] Justin Osofsky. 2016. Information About Trending Topics. (2016). http://newsroom.fb.com/news/2016/05/information-about-trending-topics/
[12] Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming First Story Detection with Application to Twitter. In *HLT/NAACL*. 181–189.
[13] Xin Qian, Jimmy Lin, and Adam Roegiest. 2016. Interleaved Evaluation for Retrospective Summarization and Prospective Notification on Document Streams. In *SIGIR*. 175–184.
[14] Adam Roegiest, Luchen Tan, and Jimmy Lin. 2017. Online In-Situ Interleaved Evaluation of Real-Time Push Notification Systems. In *SIGIR*.
[15] Adam Roegiest, Luchen Tan, Jimmy Lin, and Charles L. A. Clarke. 2016. A Platform for Streaming Push Notifications to Mobile Assessors. In *SIGIR*. 1077–1080.
[16] Arkaitz Zubiaga and Heng Ji. 2013. Harnessing Web Page Directories for Large-Scale Classification of Tweets. In *WWW Companion*. 225–226.

Acknowledgments. This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and by a Google Founders Grant.