Burst Learning to Discover Interesting Moments in Social Media Streams

Author 1
Dept.
University
Address

Abstract

This paper introduces a general technique for identifying interesting and compelling moments in social media streams without the need for domain-specific information or seed keywords. We leverage machine learning to model temporal patterns around bursts in Twitter's unfiltered public sample stream and build a classifier to identify tokens experiencing these bursts. We show our technique performs competitively with existing burst detection techniques while simultaneously providing insight into and detection of unanticipated moments. To demonstrate our approach's potential, we compare two baseline event-detection algorithms with our language-agnostic algorithm to detect key moments across three major sporting competitions (2013 World Series, 2014 Super Bowl, and 2014 World Cup). Results from this comparison show our method demonstrates similar performance in this sports domain without the need for pre-specified tokens. We then go further by transferring our sports-based models to the task of identifying earthquakes in Japan and show our method detects large spikes in earthquake-related tokens within two minutes of the actual event.

Introduction

Though researchers have presented a multitude of methods for adapting social media streams to informational and news sources for journalists or first responders, many current social media-based event detection systems rely on manually defined keywords to detect interesting events. Though straightforward and capable, such approaches are constrained to types of events for which they have prior information, potentially missing impactful but unanticipated events. Part of the difficulty in a journalist's job is to identify these unexpected but newsworthy occurrences. For instance, one can follow the frequency of words like "goal" on Twitter during the 2014 World Cup to detect when goals are scored (Cipriani 2014), but interesting occurrences like penalties or missed goals would be missed. One might respond to this weakness by tracking additional penalty-related tokens, but this approach would still be unable to identify an *unexpected* moment like Luis Suarez's biting Giorgio Chiellini during the Uruguay-Italy World Cup match; who would have thought to include "bite" as a relevant token during that event? Furthermore, relying on predefined keywords restricts these systems to those languages represented in the seed keyword set, a significant issue for international events like the World Cup. Given the shear volume of social media data (hundreds of thousands of comments, statuses, and photos generated per minute on Facebook alone as of 2012 (Pring 2012)), one could instead forgo seed keywords completely and track bursts in overall message volume (as with Vasudevan et al. (Vasudevan et al. 2013)) and sacrifice semantic information about detected events (as one would need to extract keywords causing such bursts manually). In this paper, we propose leveraging machine learning to reap the benefits of both techniques.

To explore this integration, we introduce LABurst (short for language-agnostic burst detection), a general learning method to model the temporal signatures of keyword bursts in social media streams. LAburst then tracks the number of keywords experiencing bursts in a particular moment, which serves as a predictor of an interesting or high-impact occurrence. In contrast to existing work, our approach processes *unfiltered* social media streams, discovers high-impact moments in those streams without prior knowledge of the target events, *and* yields keywords describing these discovered events. Illustrating this flexibility is a collection of experiments on Twitter streams surrounding key moments in large sporting competitions and natural disasters. These experiments include comparisons between LABurst and two existing burst detection methods: a volume-centric burst detection technique, and a similar technique with a predetermined set of sports-related keywords.

This work makes the following contributions:

- We propose a feature set and classifier for the language-agnostic discovery of impactful or interesting moments in the Twitter public sample stream. Importantly, our approach does *not* require a list of manually-defined keywords as input,
- We demonstrate our approach's performance is both competitive with existing techniques and more flexible, and
- We can transfer our models across disparate domains and still maintain comparable levels of performance to domain-specific systems.

Related Work

Though LABurst focuses on the slightly different problem of discovering interesting moments in social media streams, our work shares foundations with classical event detection research. Identifying key events from the ever-growing body of digital media has fascinated researchers for over twenty years, starting from digital newsprint to blogs and now social media (Allan, Papka, and Lavrenko 1998). Early event detection research followed that of Fung et al. in 2005, who built on the burst detection scheme presented by Kleinberg by identifying bursty keywords from digital newspapers and clustering these keywords into groups to identify bursty events (Kleinberg 2002; Fung et al. 2005). This work succeeded in identifying trending events and showed such detection tasks are feasible. Recognizing that newsprint differs substantially from social media both in content and velocity, the research community began experimenting with new social media sources like blogs, but real gains came when microblogging platforms began their rise in popularity. These microblogging platforms include Twitter and Sina Weibo and are characterized by constrained post sizes (e.g., Twitter constrains user posts to 140 characters) and broadcasting publicly consumable information.

One of the most well-known works in detecting events from microblog streams is Sakaki, Okazaki, and Matsuo's 2010 paper on detecting earthquakes in Japan using Twitter (Sakaki, Okazaki, and Matsuo 2010). Sakaki et al. show that not only can one detect earthquakes on Twitter but also that it can be done simply by tracking frequencies of earthquake-related tokens. Surprisingly, this approach can outperform geological earthquake detection tools since digital data propagates faster than tremor waves in the Earth's crust. Though this research is limited in that it requires pre-specified tokens and is highly domain- and location-specific (Japan has a high density of Twitter users, so earthquake detection may perform less well in areas with fewer Twitter users), it demonstrates a significant use case and the potential of such applications.

Along with Sakaki et al., 2010 saw two other relevant papers: Lin et al.'s construction of a probabilistic popular event tracker (Lin et al. 2010) and Petrović, Osborne, and Lavrenko's application of locality-sensitive hashing (LSH) for detecting first-story tweets from Twitter streams (Petrović, Osborne, and Lavrenko 2010a). Lin's work demonstrated that the integration of non-textual social and structural features into event detection could produce real performance gains. Like many contemporary systems, however Lin's models require seeding with pre-specified tokens to guide its event detection and concentrates on retrospective per-day topics and events. In contrast, Petrović et al.'s clustering research in Twitter avoids the need for seed keywords and retrospective analysis by instead focusing on the practical considerations of clustering large streams of data quickly. While typical clustering algorithms require distance calculations for all pairwise messages, LSH facilitates rapid clustering at the scale necessary to support event detection in Twitter streams by restricting the number of tweets compared to only those within some threshold of similarity. Once these clusters are generated, Petrović was able to track their growth over time to determine impact for a given event. This research was unique in that it was one of the early methods that did not require seed tokens for detecting events and has been very influential, resulting in a number of additional publications to demonstrate its utility in breaking news and for high-impact crisis events (Osborne et al. 2014; Petrovic et al. 2013; Rogstadius et al. 2013). Petrović's work and related semantic clustering approaches rely on textual similarity between tweets, which limits its ability to operate in mixed-language environments and differentiates LABurst and its language agnosticism.

Similar to Petrović, Weng and Lee's 2011 paper on EDCoW, short for Event Detection with Clustering of Wavelet-based Signals, is also able to identify events from Twitter without seed keywords (Weng and Lee 2011). After stringent filtering (removing stop words, common words, and non-English tokens), EDCoW uses wavelet analysis to isolate and identify bursts in token usage as a sliding window advances along the social media stream. Besides the heavy filtering of the input data, this approach exhibits notable similarities with the language-agnostic method we describe herein with its reliance on bursts to detect event-related tokens. These methods, however, operate retrospectively, focusing on daily news rather than breaking event detection on which our research focuses. Becker, Naaman, and Gravano's 2011 paper on identifying events in Twitter also fall under this label of retrospective analysis, but their findings also demonstrate reasonable performance in identifying events in Twitter by leveraging classification tasks to separate tweets into those on "real-world events" versus non-event messages (Becker, Naaman, and Gravano 2011b; Becker, Naaman, and Gravano 2011a). Similarly, Diao et al. also employ a retrospective technique to separate tweets into global, event-related topics and personal topics (Diao et al. 2012).

Several domain-specific research efforts have also targeted sporting events specifically(Vasudevan et al. 2013; Zhao et al. 2011; Lanagan and Smeaton 2011). Lanagan and Smeaton's work is of particular interest because it relies almost solely on detecting bursts in Twitter's per-second message volume (Lanagan and Smeaton 2011). Though naive, this frequency approach is able to detect large bursts on Twitter in high-impact events without reliance on complex linguist analysis and performs well in streaming contexts as little information must be kept in memory. Detecting such bursts provide evidence of an event, but this approach makes it difficult to gain insight into what that event actually was without additional processing. LABurst addresses this potential disadvantage by identifying both the overall burst and the keywords related to that burst.

More recently, Xie et al.'s 2013 paper on TopicSketch seeks to perform real-time event detection from Twitter streams "without pre-defined topical keywords" by maintaining acceleration features across three levels of granularity: individual token, bigram, and total stream. As with Petrović's use of LSH, Xie et al. leverage "sketches" and dimensionality reduction to facilitate event detection and also relies on language-specific similarities. Furthermore, Xie et al. focus only on tweets from Singapore rather than the

worldwide stream. In contrast, our approach is differentiated primarily in its language-agnosticism and its use of the unfiltered stream from Twitter's global network.

Despite this extensive body of research, it is worth asking how event detection on Twitter streams differs from Twitter's own offerings on "Trending Topics," which they make available to all their users. When a user visit's Twitter's website, she is immediately greeted with her personal feed as well as a listing of trending topics for her city, country, worldwide, or nearly any location she chooses. These topics offer insight into the current popular topics on Twitter, but the main differentiating factor is that these popular topics are not necessarily connected to specific events. Rather, popular memetic content like "#MyLovelifeInMoveTitles" often appear on the list of trending topics. Additionally, Twitter monetizes these trending topics as a form of advertising (Sydell 2011). These trending topics also can be more high-level than the interesting moments we seek to identify: for instance, during the World Cup, particular matches or the tournament in general were identified as trending topics by Twitter, but individual events like goals or penalty cards in those matches were not. It should be clear then that Twitter's trending topics serves a different purpose than the streaming event detection described herein.

Moment Discovery Defined

This paper demonstrates the LABurst algorithm's ability to discover and describe impactful moments from social media streams *without* pre-specified knowledge of the types or domains of these target moments. First though, we lay LABurst's foundations by defining the problem LABurst seeks to solve and presenting the model around which LABurst is built.

Problem Definition

Given an unfiltered (though potentially downsampled) stream S of messages m consisting of various tokens w, our objective is to determine whether each minute t contains a compelling or high-impact moment and, if so, *extract* a set of tokens that describe that moment. These tasks are difficult to perform separately because, by the time one can react with a separate analysis tool to identify relevant tokens, the moment may have passed, and to our knowledge, no existing algorithm satisfies this joint problem in the streaming context. By focusing on individual minutes of activity, we also avoid the complexities of defining "events" and the hierarchies among them.

More formally, if we let E denote the set of all minutes t in which an interesting moment occurs, then the indicator function $\mathbb{1}_E(S_t, t)$ takes the stream S up to time t and returns a 1 for all times t in which such a moment occurs, and 0 for all other values of t. We can then define the moment discovery task as constructing a function that approximates this indicator function $\mathbb{1}_E(S_t, t)$. We also include a function $B_E(S_t, t)$ that returns a set of words w that describe the discovered moment at time t if $t \in E$ and an empty set otherwise.

To account for possible lag in experiencing the event, typing out a message about the event, and the message actually posting to a social media server, we include a parameter τ to control for tolerance of delay. This parameter relaxes the task slightly by constructing the set E' where, for all $t \in E$, $t, t+1, t+2, ..., t+\tau \in E'$. Since our evaluation is a comparison between two methods that share the same ground truth, and controlling τ affects the ground truth consistently for both methods, comparative results should be unaffected. In this paper, we use $\tau = 2$.

False positives/negatives and true positives/negatives follow in the normal way for some candidate function $\widehat{\mathbb{1}_{E'}}(S_t, t)$: a false positive is any time t such that $\widehat{\mathbb{1}_{E'}}(S_t, t) = 1$ and $\mathbb{1}_{E'}(S_t, t) = 0$; likewise, a false negative is any t such that $\widehat{\mathbb{1}_{E'}}(S_t, t) = 0$ and $\mathbb{1}_{E'}(S_t, t) = 1$. True positives/negatives then follow as expected.

The LABurst Model

In LABurst, we sought to combine the language-agnostic flexibility of general burst detection techniques with the informative capabilities of pre-defined seed keyword burst detectors by leveraging machine learning to model bursts in token usage. At a high level, we ingest a social media stream, maintain a sliding window of frequencies for *each* token contained within the stream, and use the number of bursty tokens in a given minute as an indicator of the impact of that moment. Critically, these tokens can be of any language and are *neither* stemmed nor normalized. As an example, after a goal is scored in a World Cup match, one would expect to see many different forms of the word "goal" (both different languages and different variations, such as "goooooaaal!") experiencing bursts within a minute of the score; the more such tokens experience a burst, the higher impact the event.

At a lower level, LABurst runs a sliding window over the incoming data stream S and divides it into slices of a fixed number of seconds δ such that time $t_i - t_{i-1} = \delta$. LABurst then combines a set number ω of these slices into a single window (with an overlap of $\omega - 1$ slices), splits each message in that window into a set of tokens, and tabulates each token's frequency. By maintaining a list of frequency tables from the past k windows up to time t (see Figure 1), we can construct features describing how a token's frequency changes over time and apply tools from machine learning to separate these tokens into two classes: bursty tokens $\mathbf{B}_t$, and non-bursty tokens $\mathbf{B}'_t$. Following this classification, if the number of bursty tokens exceeds some threshold $|\mathbf{B}_t| \geq \rho$, LABurst flags this window at time t as containing a high-impact moment. In this manner, LABurst approximates the target indicator function with $\widehat{\mathbb{1}_{E'}}(S_t, t) = |\mathbf{B}_t| \geq \rho$ and yields $\mathbf{B}_t$ as the set of descriptive tokens for the given moment.

To avoid spurious bursts generated by endogenous network affects, it is worth noting we discard duplicate messages such as retweets or shares since existing literature shows retweets propagate extremely rapidly, potentially leading to false positives (Kwak et al. 2010).

Temporal Features To capture the dynamics of how a token's usage changes, LABurst extracts a number of temporal features for each token across k windows. Each feature is normalized into the range $[0, 1]$ to avoid biases from scale.

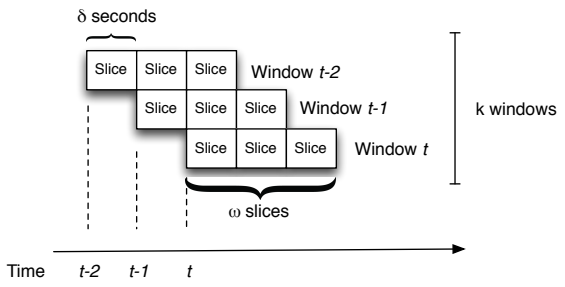


Figure 1: LABurst Sliding Window Model

- **Frequency Regression** Given the logarithm of a token's frequency at each window, take the slope of the best-fitting line.
- **Message Frequency Regression** Given the logarithm of the number of messages in which a token appears for each window, take the slope of the best-fitting line.
- **User Frequency Regression** Given the logarithm of the number of authors using a token for each window, take the slope of the best-fitting line.
- **Average Frequency Difference** The difference between the token's frequency in the most recent window and the average frequency across the previous $k-1$ windows.
- **Message Average Frequency Difference** The difference between the number of messages in which a token appears in the most recent window and the average frequency across the previous $k-1$ windows.
- **User Average Frequency Difference** The difference between the number of users who use a token in the most recent window and the average frequency across the previous $k-1$ windows.
- **Inter-Arrival Time** The average number of seconds between token occurrences in the previous k windows.
- **Entropy** The entropy of the set of messages containing a given token.
- **TF-IDF** The term frequency, inverse document frequency for a each token.
- **TF-PDF** A modified version of TF-IDF called term frequency, proportional document frequency (Bun and Ishizuka 2002).
- **BursT** Weight using a combination of a given token's actual frequency and expected token frequency (Lee, Wu, and Chien 2011).

LABurst's Bursty Token Classification LABurst's key component is its ability to differentiate between bursty and non-bursty tokens. To make this determination, LABurst integrates these temporal features into feature vectors for each token and processes them using an ensemble of known classification algorithms, specifically support vector machines (SVMs) and random forests (RFs) integrated using AdaBoost. Training these burst detection classifiers requires both positive and negative samples of bursty tokens. Obtaining positive samples of bursty tokens is relatively straightforward in that we can identify a number of high-impact, real-world events and construct a set of seed tokens that *should* experience bursts along with the event, following the same workflow as many typical seed-based event detection approaches. Determining negative samples is more difficult, however, since one cannot know all events occurring around the world at a given moment. Fortunately, we can rely on a trick of linguistic and leverage stop words as negative samples as stop words are in general highly used but used consistently (i.e., stop words are intrinsically non-bursty). Therefore, in the experiments described herein, we train LABurst on a set of known events with specified bursty tokens and stop words in both English and Spanish. As this task is semi-supervised, we also include a self-training phase after initial training is complete to identify additional bursty tokens.

Evaluation Framework

Having established the details of our model, we now turn to frameworks for evaluating LABurst compared to existing methods. To explore such comparisons, we first look to other, similar methods for detecting interesting events from social media streams and compare their relative accuracies with LABurst. We then include a second experiment to demonstrate LABurst's domain independence and utility in an emergency scenario.

Comparative Accuracy in Event Discovery

Our first and primary research question is as follows:

- **RQ1** Is LABurst competitive at event discovery when compared with existing systems?

To answer this question, we construct an experiment for enumerating high-impact moments during major sporting competitions. Such competitions are interesting given their large followings (many fans to post on social media), thorough coverage by sports journalists (high-quality ground truth), and regular occurrence (large volume of data) make them ideal for both data collection and evaluation. Such events are also complex in that they include multiple types of events and unpredictable patterns of events around scores, fouls, and other compelling moments of play.

Our first step here was to collect data for a number of popular competitions and identify important or key moments in each event. We captured these moments and their times from sport journalism articles, game highlights, box scores, blog posts, and social media messages. These moments then comprise the ground truth against which we can compare LABurst to baseline approaches. From there, we introduced a pair of baseline methods: first, a general burst detection algorithm using raw message frequency following the approaches of Vasudevan et al. and the "activity peak detection" method set forth by Lehmann et al. (Vasudevan et al. 2013; Lehmann et al. 2012), and second, a seed keyword-based algorithm in the pattern of Cipriani and Zhao et al. (Cipriani 2014; Zhao et al. 2011). We then evaluate the relative performance for LABurst and both baselines as described below.

Sporting Competitions To minimize bias, these competitions covered several different sporting types, from horse racing to the National Football League (NFL), to Fédération Internationale de Football Association (FIFA) premier league soccer, to the National Hockey League

(NHL), National Basketball Assoc. (NBA), and Major League Baseball (MLB). Each competition also contained four basic types of events: beginning of the game, end of the game, scores, and penalties. Table 1 lists sources of events we identified and the number of key moments in each.

Table 1: Sporting Competition Data

Sport	Key Moments
2010 NFL Division Championship	13
2012 Premier League Soccer Games	21
2013 MLB World Series	15
2014 NFL Super Bowl	13
2014 NHL Stanley Cup Playoffs	24
2014 NBA Playoffs	3
2014 Kentucky Derby Horse Race	3
2014 Belmont Stakes Horse Race	3
2014 FIFA World Cup	98
Total	193

In 2012, we tracked four Premier League games in November. Likewise, we tracked only a subset of games during the NHL Stanley Cup and NBA playoffs. For the 2013 World Series between the Boston Red Sox and the St. Louis Cardinals, we explore the final two games on 28 October and 30 October of 2013. Similarly for the 2014 World Cup, our analysis covers a number of early matches during stages 1 and 2 and the the final two matches of tournament: the 12 July match between the Netherlands and Brazil for third place, and the final match on 13 July between Germany and Argentina for first place.

To provide for separation between training and testing data, we split these competitions with the testing data covering the 2013 MLB World Series, 2014 NFL Super Bowl, and the final two matches of the 2014 FIFA World Cup, and the remaining data being used for training.

Burst Detection Baselines The LABurst algorithm straddles the line between raw burst detection algorithms and token-centric burst detectors. Therefore, to evaluate LABurst properly, we implemented two such baselines for comparison. The first baseline, to which we refer as RawBurst, uses a known method for detecting activity peaks by taking the difference between the number of messages seen in the current time slice and the average number of messages seen over the past k time slices (Vasudevan et al. 2013; Lehmann et al. 2012).

Formally, we define a series of time slices $t \in T$ segmented into δ seconds and a social media stream S containing messages m such that S_t contains all messages in the stream between $t-1$ and t. We then define the frequency of a given time slice t as $\text{freq}(t, S) = |S_t|$ and the average over the past k time slices as $\text{avg}(k, t, S)$, shown in Eq. 1.

$$\text{avg}(k, t, S) = \frac{\sum_{j=t-k}^{t} \text{freq}(j, S)}{k} \tag{1}$$

Given these functions, we take the difference $\Delta_{t,k}$ between the frequency at time t and the average over the past k slices such that $\Delta_{t,k} = \text{freq}(t, S) - \text{avg}(k, t, S)$. If this difference exceeds some threshold ρ such that $\Delta_{t,k} \geq \rho$, we say an event was detected at time t.

Following the course of Cipriani from Twitter's Developer Blog, we then modify the RawBurst algorithm to detect events using frequencies of a small set of seed tokens $w \in W$, to which we will refer as TokenBurst (Cipriani 2014). To convert RawBurst into TokenBurst, we simply modify the $\text{freq}(t, S)$ function to return the summed frequency of all seed tokens, as shown in Eq. 2 where $\text{count}(w, S_t)$ returns the frequency of token w in the stream S during time slice t. These seed tokens chosen such that they are likely to exhibit bursts in usage during the key moments of our sporting event data, such as "goal" for goals in soccer/football or hockey or "run" for runs scored in baseball. This TokenBurst implementation also includes some rudimentary normalization to collapse modified words to their originals (e.g., "gooaalllll" down to "goal"). Such a technique should be effective because, as seen in the related work, many existing stream-based event detection systems use just such an approach to track specific types of events.

$$\text{freq}(t, S) = \sum_{w \in W} \text{count}(w, S_t) \tag{2}$$

Since our analysis covers three separate types of sporting competitions, the seed keyword list for this method should include tokens from the vocabulary of each. We avoid separate keyword lists for each sport to provide an even comparison to the general nature of our language-agnostic technique. The keywords for which we searched are shown in Table 2, and we took the union of these token sets. Additionally, the following regular expressions collapsed deliberately misspelled tokens down to their normal counterparts: "g+o+a+l+" →"goal", "g+o+l+" →"gol", "g+o+l+a+z+o+" →"golazo", "sco+red?" →"score".

Table 2: Predefined Seed Tokens

Sport	Tokens
World Series	"run", "home", "homerun"
Super Bowl	"score", "touchdown", "td", "fieldgoal", "points"
World Cup	"goal", "gol", "golazo", "score", "foul", "penalty", "card", "red", "yellow", "points"

Performance Evaluation Having defined LABurst, RawBurst, and TokenBurst, we now can compare the three algorithms by constructing a series of receiver operating characteristic (ROC) curves across test sets of our sports data. We then evaluate relative performance between the approaches by comparing their respective areas under the curves (AUCs) by varying the threshold parameters for each method. In RawBurst and TokenBurst, this threshold parameter refers to ρ in $\Delta_{t,k} \geq \rho$. For our LABurst method, the ROC curve is generated by varying the minimum the ρ in $\widehat{\mathbb{1}_{E'}}(S_t, t) = |\mathbf{B}_t| \geq \rho$. The AUC of the ROC curve is useful in this instance because it is robust against imbalanced classes, which we expect to see in such an event detection task. Then, by

comparing these AUC values, we can provide an answer to **RQ1**.

Evaluating Domain Independence

Beyond LABurst's ability to discover and describe interesting moments, we also claim it to be domain independent. To justify this claim, we must answer our second research question:

- **RQ2** Can LABurst discover events in a domain completely separate from its training domain?

Detecting key moments within sporting competitions as described above is a useful task for areas like advertising or automated highlight generation, but a more compelling and worthwhile task would be to detect higher-impact events like natural disasters. The typical frequency-based approach is difficult here as it is impossible to know what events are about to happen where, and a list of target keywords to detect all such events would be long and lead to false positives. LABurst could be highly beneficial here as one need not know details like event location, language, or type. This new context presents a good opportunity to evaluate LABurst on in a new domain and compare it to existing work by Sakaki, Okazaki, and Matsuo (Sakaki, Okazaki, and Matsuo 2010). Thus, to answer **RQ2**, we can take the LABurst model as trained on sporting events presented for **RQ1** and apply them directly to this task.

For this earthquake detection task, we compare LABurst with the TokenBurst baseline using the keyword "earthquake," following Sakaki, Okazaki, and Matsuo. Also following Sakaki et al., we target earthquakes in Japan over the past two years and select two of the most severe: the 7.1-magnitude quake off the coast of Honshu, Japan on 25 October 2013, and a 6.5-magnitude quake off the coast of Iwaki, Japan on 11 July 2014. Rather than generating ROC curves for this comparison, we take a more straightforward approach and compare lag between the actual earthquake event and the point in time in which the two methods detect the earthquake. If the lag between TokenBurst and LABurst is sufficiently small, we will have good evidence for an affirmative answer to **RQ2**.

Data Collection

While the algorithms described herein are general enough to be applied to any sufficiently active social network stream, the ease with which one can access and collect Twitter data makes it an attractive target for our research. To this end, we leveraged two existing Twitter corpora and created our own corpus of tweets from Twitter's 1% public sample stream[1] using code from Jimmy Lin's twitter-tools library[2]. In collecting from Twitter's public sample stream, we connect to the Twitter API endpoint (provide *no filters*), and a sampling of 1% of all public tweets are returned as a stream we can then collect into files for local streaming and analysis.

The two existing corpora we used were the Edinburgh Corpus (Petrović, Osborne, and Lavrenko 2010b), which covered the 2010 NFL division championship game, and an existing set of tweets pulled from Twitter's firehose source targeted at Argentina during November of 2012, which covered the four Premier League soccer games. All remaining data sets were extracted from Twitter's sample stream over the course of October 2013 to July 2014.

Where possible, for each event (both sporting and earthquake), we extracted tweets starting an hour before the target event and ending an hour after the event, which yielded a total of over 15 million tweets. Table 3 shows the breakdown of tweets collected per event.

Table 3: Per-Event Tweet Counts

Event	Tweet Count
2010 NFL Division Championship	109,809
2012 Premier League Soccer Games	1,064,040
2013 Honshu Earthquake	444,018
2013 MLB World Series	1,563,822
2014 NFL Super Bowl	1,024,367
2014 NHL Stanley Cup Playoffs	2,421,065
2014 NBA Playoffs	500,170
2014 Kentucky Derby Horse Race	233,172
2014 Belmont Stakes Horse Race	226,160
2014 FIFA World Cup	7,843,976
2014 Iwaki Earthquake	358,966
Total	15,789,565

Experimental Results

Setting Model Parameters

Prior to carrying out the experiments described above, we first had to discern appropriate parameters for window sizes and LABurst's classifiers. For LABurst's parameters regarding slice size δ, window size ω, and k previous windows, preliminary experimentation yielded good results with the following settings: $\delta = 60$, $\omega = 3$, and $k = 10$. We also used these δ and k parameters in both RawBurst and TokenBurst as well.

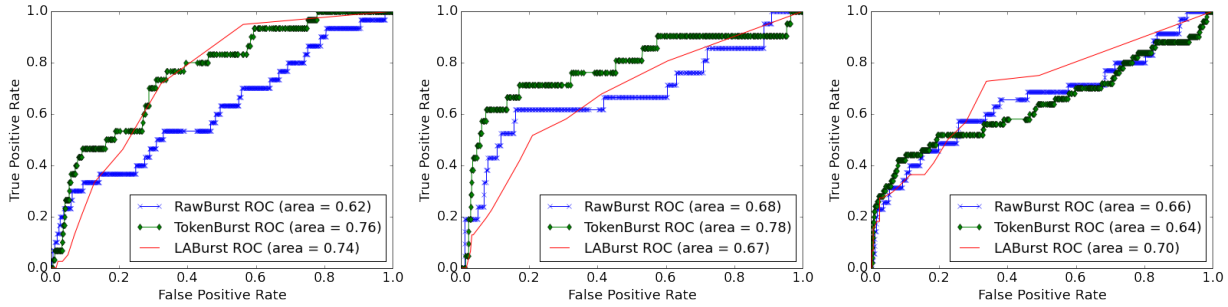
Regarding LABurst's classifier implementations, we used the Scikit-learn Python package for SVMs and RFs as well as an implementation of the ensemble classifier AdaBoost [3]. Both SVMs and RFs have tunable parameters to select before integrating into AdaBoost, however, so we employed a grid search strategy to select parameters based on the F1 scores on our training and testing data.

For SVMs, we first had to decide whether to use a traditional linear model or employ a kernel. Initial experiments showed linear SVMs performed quite poorly, and after using principal component analysis to reduce the dimension and looking at the labeled data, it fit well in a sphere rather than a clear linear plane. As a result, we use the radial basis kernel, which has two parameters: cost c and kernel coefficient γ. In searching the space of c and γ, the grid covers powers of two such that $c = 2^x$, $x \in [-2, 10]$ and $\gamma = 2^y$, $y \in [-2, 5]$. For each pair of parameter values, we train thirty different

[1] https://dev.twitter.com/streaming/reference/get/statuses/sample

[2] https://github.com/lintool/twitter-tools

[3] http://scikit-learn.org/



(a) 2013 World Series (b) 2014 Super Bowl (c) 2014 World Cup

Figure 2: Per-Sport ROC Curves

classifiers using repeated random subsampling, take the average F1 score, and select the parameter set with the highest F1 score. Selecting parameter values for RFs is similar for the number of estimators n and feature count c' such that $n = 2^x$, $x \in [0, 10]$ and $c' = 2^y$, $y \in [1, 11]$. This training procedure yielded the results shown in Table 4. These two classifiers are then combined using the Scikit-learn's AdaBoost implementation with four estimators.

Table 4: Classifier Parameter Scores

Classifier	Params	F1-Score
SVM	$c = 64$, $\gamma = 4$	0.588410
RF	trees = 128, features = 9	0.575301

Event Discovery Results

To restate, the first research question (**RQ1**) posed in this work is to determine whether LABurst can perform as well as existing methods in detecting events, with a focus on sporting competitions. We answer this question across three separate sporting events: the final two games of the 2013 MLB World Series, the 2014 NFL Super Bowl, and the final two matches of the 2014 FIFA World Cup.

Prior to presenting comprehensive results, we first present performance curves for each sporting competition. For the 2013 World Series, RawBurst's AUC is 0.62, TokenBurst's UAC is 0.76, and LABurst's is 0.74. From 2a, LABurst clearly dominates RawBurst and exhibits performance similar to TokenBurst (with a difference of only 0.02) During the Super Bowl, RawBurst, TokenBurst, and LABurst achieve an AUC of 0.68, 0.78, and 0.67 respectively, with the difference between RawBurst and LABurst dropping significantly (shown in Figure 2b). Unlike the World Series and Super Bowl, however, during the 2014 World Cup, LABurst (AUC=0.70) outperformed both RawBurst (AUC=0.66) and TokenBurst (AUC=0.64), as seen in Figure 2c.

Composite Results

To compare comprehensive performance, we look to Figure 3, which shows ROC curves for all three methods across all three testing events. From this figure, we see LABurst (AUC=0.71) outperforms RawBurst (AUC=0.65) and performs nearly identically to TokenBurst (AUC=0.72). Assuming equal cost for false positives and false negatives and optimizing for the largest difference between true positive rate (TPR) and false positive rate (FPR), TokenBurst shows a TPR of 0.5581 and FPR of 0.1408 with a difference of 0.4174 at a threshold value of 13.2. LABurst, on the other hand, has a TPR of 0.7105 and FPR of 0.3518 with a difference of 0.3587 at a threshold value of 2. From these values, we see LABurst achieves a higher true positive rate at the cost of a higher false positive rate. Therefore, it seems the answer to **RQ1** is that, yes, LABurst can be competitive with existing methods.

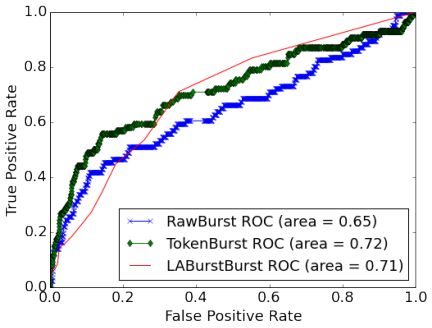
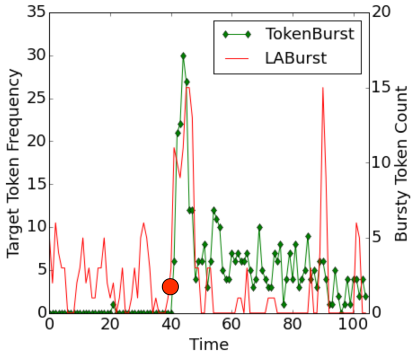


Figure 3: Composite ROC Curves

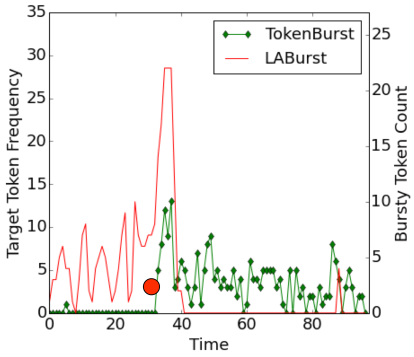
Earthquake Detection

Our second research question (**RQ2**) seeks to determine whether transferring LABurst's models, as trained on the previously mentioned sporting events, can compete with existing techniques in a different domain.

Figures 4a and 4b show the detection curves for both methods for the 2013 and 2014 earthquakes respectively; the red dots indicate the earthquake times as reported by the United States Geological Survey (USGS). The left vertical axis for each figure reports the frequency of the "earthquake" token, and the right axis shows the number of tokens classified as bursty by LABurst. From the TokenBurst



(a) Honshu, Japan Earthquake - 25 October 2013



(b) Iwaki, Japan Earthquake - 11 July 2014

Figure 4: Japanese Earthquake Detection

curve, one can see the token "earthquake" sees a significant increase in usage when the earthquake occurs, and LABurst experiences a similar increase at the same moment for both events. It is worth noting that LABurst exhibits bursts prior to the earthquake event, but these peaks *unrelated* to the earthquake event as LABurst does not differentiate between the earthquake and other high-impact events that could be happening on Twitter. In addition, the peak occurring about 50 minutes after the earthquake on 25 October 2013 potentially represents an aftershock event[4]. Given the minimal lag between the LABurst and TokenBurst's detection, we can answer **RQ2** in the affirmative.

One can now ask what tokens we identified as bursting when the earthquakes occurred. Many of the tokens are in Japanese, and tokens at the peak of the earthquake events are shown in Table 5. We also extracted several tweets that contain the highest number of these tokens for the given time period, a selection of which include, "地震だあああああああああああああああああああああああ," "今回はチト使ってないから地震わからなかった," and "地震だ." Google Translate[5] translates these tweets as "Ah ah ah ah ah ah ah ah Aa's earthquake," "I did not know earthquake because not using cheat this time," and "Over's earthquake" respectively.

Table 5: Tokens Classified as Busting During Events

Match	Bursty Tokens
Honshu, Japan – 25 October 2013	ç 丈, 地, 夫, 怖, 波, 注, 津, 源, 福, 震
Iwaki, Japan – 11 July 2014	び, ゆ, ビビ, 地, 怖, 急, 福, 警, 速, 震

Analysis

In comparing LABurst with the baseline techniques, it is important to understand the strengths and weaknesses of each baseline: RawBurst requires no prior information but provides little in the way of semantic information regarding detected events, while TokenBurst provides this semantic information at the cost of missing unknown tokens or significant events that do not conform to its prior knowledge. LABurst combines the benefits of these two approaches in that it not only directly supports undirected event discovery while simultaneously yielding insight into these moments of high interest by tagging relevant bursting tokens.

Identifying Event-Related Tokens

As mentioned, where the baselines require sacrificing flexibility or information, LABurst jointly attacks these problems and yields event-related tokens automatically. These tokens may include misspellings, colloquialisms, and cross language boundaries, which makes them hard to know before hand. The 2014 World Cup presents an interesting case for finding these otherwise unexpected tokens because the event has enormous international viewership; as such, many Twitter users of many different languages are likely tweeting about the same event.

To explore the tokens generated during these high-profile events, we look to those tokens identified as bursting during several events in the final two World Cup matches. Table 6 shows a selection of events from these matches and a subset of those tokens classified as bursting during the events (one should note the list is not exhaustive owing to formatting and space constraints).

Several interesting artifacts emerge from this table, first of which is that one can get an immediate sense of the detected event from tokens our algorithm presents. For instance, the prevalence of the token "goal" and its variations clearly indicate a team scored in the first and third events in Table 6; similarly, bursting tokens associated with the middle event regarding Oscar's yellow card reflect his penalty for diving. Beyond the pseudo event description put forth by the identified tokens, this reference to diving and to specific player and teams names in the first and third events are also of significant interest. In the first event, one can infer that the Netherlands scored since "holandaaaa" is flagged along with "persie" from the Netherlands' player Van Persie, and like-

[4]http://ds.iris.edu/spud/aftershock/9761021

[5]http://translate.google.com

Table 6: Tokens Classified as Busting During Events

Match	Event	Bursty Tokens
Brazil v. Netherlands, 12 July 2014	Netherlands' Van Persie scores a goal on a penalty at 3', 1-0	0-1, 1-0, 1:0, 1x0, card, goaaaaaaal, goal, gol, goool, holandaaaa, kırmızı, pen, penal, penalti, pênalti, persie, red
Brazil v. Netherlands, 12 July 2014	Brazil's Oscar get's a yellow card at 68'	dive, juiz, penalty, ref
Germany v. Argentina, 13 July 2014	Germany's Götze scores a goal at 113', 1-0	goaaaaallllllll, goallllll, godammit, goetze, gollllll, goooooool, gotze, gotzeeee, götze, nooo, yessss, ドイツ

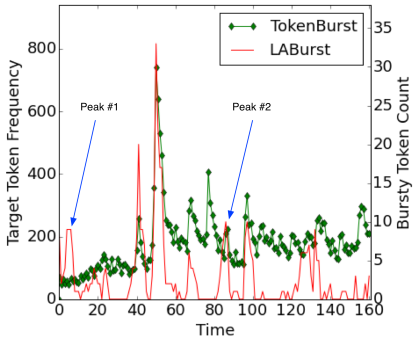


Figure 5: Baseline and LA Bursty Frequencies

wise for Germany's Götze in the third event (and the accompanying variations of his name). These terms would be difficult to capture beforehand as would be required in the baseline and would likely not be related to every event or every type of sporting event.

Finally, the last artifact of note is that the set of bursty tokens displayed includes tokens from several different languages: English for "goal" and "penalty," Spanish for "gol" and "penal," Brazilian Portuguese for "juiz" (meaning "referee"), as well as the Arabic for "goal" and Japanese for "Germany." Since these words are semantically similar but syntactically dissimilar, typical normalization schemes could not capture these connections. Instead, capturing these words in the baseline would require a pre-specific keyword list in all possible languages or the inclusion of an expensive machine translation system that was also capable of normalizing within different languages (to collapse "goool" down to "gol" for example).

Discovering Unanticipated Moments

Our experimental results show LABurst is competitive with the domain-specific TokenBurst, but TokenBurst's specificity makes it unable to detect unanticipated moments, and we can see instances of such omissions in the last game of World Cup. Figure 5 shows target token frequencies for TokenBurst in green and bursty tokens for our method in red. From this graph, we can see the first, obvious incidence in Peak #1 where LABurst exhibits a peak missed by TokenBurst in the first few points of data. The primary tokens appearing in this peak are "puyol," "gisele," and "bundchen," which correspond to former Spanish player Carles Puyol and model Gisele Bundchen, who presented the World Cup trophy prior to the match. At peak #2, slightly more than eighty minutes into the data (which is sixty minutes into the match), LABurst sees another peak that is otherwise relatively minor in the TokenBurst curve. Upon further exploration, tokens present in this peak refer to Argentina's substituting in Agüero for Lavezzi at the beginning of the match's second half.

Since LABurst is both language agnostic and domain independent, it likely detects additional high impact events but not part of the game start/end, score, and penalty events present in our ground truth. For instance, during the Super Bowl, spectators tweet about more moments than just sports plays; they tweet about the half-time show, particularly good or bad commercials, and massive power outages. Since our ground truth disregards such moments, its higher false-positive rate is perhaps unsurprising.

Conclusions

To revisit our motivations, the goal for this experiment was to demonstrate the potential of a language-agnostic, machine learning-based approach to discovering highly compelling or interesting moments through analyzing temporal characteristics from unfiltered Twitter data streams. Our results show that by leveraging temporal characteristics to identify bursty tokens and using the volume of these bursty tokens, we can detect significant events across a collection of disparate sporting competitions and other domains with a level of performance nearly equivalent to an existing, domain-specific baseline.

Similar performance to a baseline is only part of the story, however, as our approach offers notable flexibility in identifying bursting tokens without normalization and across language boundaries. With this versatility also comes support for event description since we no longer rely on predetermined keywords; that is, we can get a sense of the occurring event by inspecting the bursty tokens. Finally, these advantages culminate in powerful tool for event *discovery* in that it can unanticipated instances of high interest that we did not expect, regardless of the source language, which makes this technique particularly useful for journalists and newswire sources who have a need to know about events on the ground, as they happen but cannot know a priori what the event may be about in all cases. In short, this LABurst moment discovery algorithm is able to compete with existing techniques, performs well beyond its training domain, *and* automatically yields tokens describing discovered events.

Acknowledgments

This work made use of the Open Science Data Cloud (OSDC), which is an Open Cloud Consortium (OCC)-sponsored project. The OSDC is supported in part by grants

from Gordon and Betty Moore Foundation and the National Science Foundation and major contributions from OCC members like the University of Chicago.

References

[Allan, Papka, and Lavrenko 1998] Allan, J.; Papka, R.; and Lavrenko, V. 1998. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 37–45. ACM.

[Becker, Naaman, and Gravano 2011a] Becker, H.; Naaman, M.; and Gravano, L. 2011a. Beyond Trending Topics: Real-World Event Identification on Twitter. *ICWSM* 11:438–441.

[Becker, Naaman, and Gravano 2011b] Becker, H.; Naaman, M.; and Gravano, L. 2011b. Beyond Trending Topics: Real-World Event Identification on Twitter - Technical Report.

[Bun and Ishizuka 2002] Bun, K. K., and Ishizuka, M. 2002. Topic Extraction from News Archive Using TF*PDF Algorithm. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering*, WISE '02, 73–82. Washington, DC, USA: IEEE Computer Society.

[Cipriani 2014] Cipriani, L. 2014. Goal! Detecting the most important World Cup moments. Technical report, Twitter.

[Diao et al. 2012] Diao, Q.; Jiang, J.; Zhu, F.; and Lim, E.-P. 2012. Finding bursty topics from microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 536–544. Association for Computational Linguistics.

[Fung et al. 2005] Fung, G. P. C.; Yu, J. X.; Yu, P. S.; and Lu, H. 2005. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, 181–192. VLDB Endowment.

[Kleinberg 2002] Kleinberg, J. 2002. Bursty and hierarchical structure in streams. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, 91–101. New York, NY, USA: ACM.

[Kwak et al. 2010] Kwak, H.; Lee, C.; Park, H.; and Moon, S. 2010. What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, 591–600. ACM.

[Lanagan and Smeaton 2011] Lanagan, J., and Smeaton, A. F. 2011. Using twitter to detect and tag important events in live sports. *Artificial Intelligence* 542–545.

[Lee, Wu, and Chien 2011] Lee, C.-H.; Wu, C.-H.; and Chien, T.-F. 2011. BursT: a dynamic term weighting scheme for mining microblogging messages. In *Proceedings of the 8th international conference on Advances in neural networks - Volume Part III*, ISNN'11, 548–557. Berlin, Heidelberg: Springer-Verlag.

[Lehmann et al. 2012] Lehmann, J.; Gonçalves, B.; Ramasco, J. J.; and Cattuto, C. 2012. Dynamical Classes of Collective Attention in Twitter. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, 251–260. New York, NY, USA: ACM.

[Lin et al. 2010] Lin, C. X.; Zhao, B.; Mei, Q.; and Han, J. 2010. PET: a statistical model for popular events tracking in social communities. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, 929–938. New York, NY, USA: ACM.

[Osborne et al. 2014] Osborne, M.; Moran, S.; McCreadie, R.; Von Lunen, A.; Sykora, M.; Cano, E.; Ireson, N.; Macdonald, C.; Ounis, I.; He, Y.; and Others. 2014. Real-Time Detection, Tracking, and Monitoring of Automatically Discovered Events in Social Media. *Association for Computational Linguistics*.

[Petrovic et al. 2013] Petrovic, S.; Osborne, M.; McCreadie, R.; Macdonald, C.; Ounis, I.; and Shrimpton, L. 2013. Can Twitter replace Newswire for breaking news? In *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media*, volume 2011.

[Petrović, Osborne, and Lavrenko 2010a] Petrović, S.; Osborne, M.; and Lavrenko, V. 2010a. Streaming First Story Detection with Application to Twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, 181–189. Stroudsburg, PA, USA: Association for Computational Linguistics.

[Petrović, Osborne, and Lavrenko 2010b] Petrović, S.; Osborne, M.; and Lavrenko, V. 2010b. The Edinburgh Twitter Corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, WSA '10, 25–26. Stroudsburg, PA, USA: Association for Computational Linguistics.

[Pring 2012] Pring, C. 2012. 100 social media statistics for 2012. *TheSocialSkinny.com*.

[Rogstadius et al. 2013] Rogstadius, J.; Vukovic, M.; Teixeira, C. A.; Kostakos, V.; Karapanos, E.; and Laredo, J. A. 2013. CrisisTracker: Crowdsourced social media curation for disaster awareness. *IBM Journal of Research and Development* 57(5):4:1–4:13.

[Sakaki, Okazaki, and Matsuo 2010] Sakaki, T.; Okazaki, M.; and Matsuo, Y. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, 851–860. New York, NY, USA: ACM.

[Sydell 2011] Sydell, L. 2011. How Twitter's Trending Algorithm Picks Its Topics.

[Vasudevan et al. 2013] Vasudevan, V.; Wickramasuriya, J.; Zhao, S.; and Zhong, L. 2013. Is Twitter a good enough social sensor for sports TV? In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, 181–186. IEEE.

[Weng and Lee 2011] Weng, J., and Lee, B.-S. 2011. Event Detection in Twitter. In *ICWSM*.

[Zhao et al. 2011] Zhao, S.; Zhong, L.; Wickramasuriya, J.; and Vasudevan, V. 2011. Human as Real-Time Sensors of Social and Physical Events: A Case Study of Twitter and Sports Games. *CoRR* abs/1106.4.