Bootstrapping Language-Agnostic Event Detection in Social Media Streams with Sporting Events

Cody Buntain
Dept. of Computer Science
University of Maryland
College Park, Maryland 20742
cbuntain@cs.umd.edu

Jimmy Lin
College of Information Studies
University of Maryland
College Park, Maryland 20742
jimmylin@cs.umd.edu

Jen Golbeck
College of Information Studies
University of Maryland
College Park, Maryland 20742
golbeck@cs.umd.edu

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam tristique quam dolor, sed fermentum eros commodo eget. Nunc sem ante, tempor gravida gravida quis, vehicula nec dui. Integer hendrerit laoreet mi eu commodo. Integer lacus metus, suscipit at dignissim eget, blandit eget velit. Aenean ac porta metus, ac ultrices sem. Integer tincidunt arcu tortor, id sollicitudin lorem finibus nec. Morbi dignissim purus eget est porta interdum. Nulla faucibus lacinia dignissim. Praesent at nibh dignissim, placerat arcu sit amet, interdum orci.

In volutpat gravida turpis, nec ornare quam. Nam elementum elit non risus rhoncus, facilisis feugiat massa commodo. Vestibulum tempor felis et porttitor vestibulum. Nullam odio neque, ornare in interdum vel, volutpat quis ipsum. Integer vel finibus erat. Quisque eu mi vehicula erat imperdiet vestibulum. Sed laoreet eros lacinia aliquam mollis. Donec fringilla ante id ex pellentesque ultrices. Interdum et malesuada fames ac ante ipsum primis in faucibus.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining;
H.3.3 [**Information Search and Retrieval**]: Information Filtering

Keywords

event detection, twitter, social networks, temporal features

1. INTRODUCTION

Social media's ubiquity has drastically altered the velocity with which we spread and consume information, transforming how news and opinions propagate across the planet. This democratization of information dissemination has both researchers and traditional media organizations considering social media as a viable complement and/or alternative to real-time news sources. For instance, existing research demonstrates social media can perform on par with existing newswire sources for certain types of breaking stories [13]. While such social media-based systems are able to detect noteworthy events rapidly and in near real time, they often come with restrictive assumptions to facilitate the event-detection task. These assumptions might be requiring pre-specified classes of interesting event, limiting target languages, setting event-centric queries, and preprocessing/filtering data through expensive language models. Resulting systems are brittle and inflexible in application and are difficult to adapt to new domains, languages, or events. Forgoing these simplifying restrictions and processing an unfiltered social media stream is a more difficult task given the volume of noise in such sources, but advantages like multi-language event detection and unexpected event discovery have many practical and important applications currently unsatisfied by existing technology.

Existing work already achieves acceptable performance in detecting events within social media streams when event types and keywords are known, but what if we could achieve the same level of performance in a language-agnostic way and without the need to specify keywords or filter the streams? Given the "unreasonable effectiveness of data" and the shear volume of social media content generated per minute (hundreds of thousands of comments, statuses, and photos on Facebook alone as of 2012 [14]), it is possible that usage patterns alone could provide interesting cues for detecting events without relying on linguistic understanding and related drawbacks. This paper pursues such an alternative avenue for unfiltered event detection by disregarding language semantics and focusing on the temporal patterns that comprise breaking news and high-impact events.

To explore these questions, we propose a set of features for learning patterns of increased token usage, or bursts, in the Twitter stream in response to key moments in large sporting competitions. From these features, we then build an ensemble classifier to detect such bursting tokens from the Twitter stream. Then, using frequencies of these classified tokens, we demonstrate the feasibility of detecting new events of interest from unfiltered Twitter data. To ground this exploration, we start by modeling large sporting competitions as such events are both highly followed and occur regularly but also include unpredictable patterns of sub-events around points scored, fouls committed, and other occurrences of high interest. Experiments on several such sporting competitions illustrate how a language-agnostic, burst-centric technique performs in comparison to baseline keyword-centric meth-

ods while simultaneously providing additional insight across languages and without text normalization or filtering. Finally, we apply the models learned in the sporting domain to earthquake detection and show performance commensurate with existing research.

2. RELATED WORK

With social media's explosive popularity and the ease with which users can post information using a variety of means (mobile applications, the Web, or via text message), the huge volumes of data now being published has proven useful for a variety of purposes, the most popular of which concentrates on event detection and summarization. In 2009, researchers began transferring expertise on event detection from traditional news and blog data to social network-based microblogs like Twitter. Nagarajan et al. adapted existing spatio-temporal analysis to their Twitris framework to identify and localize specific themes according to region [8]. Twitris relies on Google's "Insights for Search" to identify trending keywords for a given location or time interval; these keywords are fed into Twitter's search API to bootstrap data collection. Twitris clusters these trending tokens to identify thematically similar content and present groups of related tokens as retrospective event "storylines." Though Twitris and similar systems [17, 7] are powerful, their event summaries are too coarse to detect individual occurrences in sports and rely heavily on traditional media to bootstrap the detection process.

Cataldi et al. take a different approach in their 2010 paper on detecting emerging topics in Twitter by leveraging "user authority" as calculated using the well-known Google PageRank algorithm [3]. Though this approach might be useful for identifying authoritative sources like sports journalists or official team Twitter accounts, our approach forgoes user characteristics. Instead, we prioritize token-centric burst information over a user's network influence since only a limited number of fans tweeting about some event may be authoritative in the network.

Work by Petrović, Osborne, et al. is perhaps the most similar system to ours [11, 13, 9]. This system, called ReDites, relies on locality sensitive hashing (LSH) to enable near real-time tweet clustering for event detection but is restricted to only for English-language tweets. LSH allows for fast similarity calculations to determine a message's "nearest neighbor," which enables high-speed clustering for theme/event generation. Once ReDites constructs these themes, it can perform a retrospective analysis to identify the first story related to a given theme. Our approach has a language flexibility that ReDites lacks, and we focus more on identifying events as they occur rather than retrospectively, but ReDites's ability to handle many Twitter messages at an extremely high rate is an impressive benchmark for which we are striving for future versions.

Only recently has event detection specific to sports gathered more attention. Lanagan and Smeaton tried to align changes in tweet volume of a filtered Twitter stream with an annotated audio/video analysis and showed that bursts in Twitter usage co-occurred with high-impact events[5]. Zhao et al. used a lexicon of American football-related terms to refine the Twitter stream and detect events during the 2010-2011 NFL football season within 40 seconds [19]. Vasudevan et al. used Twitter to identify events specific to American football and found that events could be detected within a few minutes of the actual event [18]. The common threads among all these approaches, however, are prior knowledge of event type and a pre-specified set of event keywords, which limits their applicability to international sports and other languages.

3. TECHNIQUES FOR STREAMING EVENT DETECTION

This paper's primary goal is to demonstrate the feasibility of detecting significant events from social media streams without relying on pre-specified queries and instead by identifying temporal bursts in token usage and analyzing their frequencies (we refer to this language-agnostic, burst-centric technique as LABurst). To this end, we compare LABurst's performance with that of a baseline token-based technique in detecting specific events within a variety of major sporting competitions. Then, we show applications outside the sports domain by applying LABurst to the task of detecting earthquakes from social media soon after they occur. This section's remainder defines these detection tasks, presents our experimental framework, details the data sources used, and outlines both the baseline technique and our LABurst method.

3.1 Problem Definition

Prior to any deep description on methods, we must first define an "event" and a "burst" within the context of this research. Though the term "event" is vague and overridden in many contexts, we can borrow from existing works like Allan et al.'s preliminary report on event detection and tracking and Raimond and Abdallah's Event Ontology [15, 1]. From these sources, one could define an event as "something that happens at a particular time and place" and can be composed of one or more brief "sub-events" that occur within the context of the larger event. For the purposes of this paper, however, we further restrict our definition to *instantaneous* events and sub-events, or those events that occur at a specific place and time *and* have a negligible amount of time between start and end. Examples of such instantaneous events include a point scored in a game, a particular explosion during some terrorist attack, or a particular tremor in an earthquake. These instantaneous events are interesting and harder to identify than their longer-term counterparts because one may not know tokens relevant to these brief events until the event is occurring, and by the time one can react, such a short-term event would already be over.

Corresponding to these brief events, we define a burst in a token's usage as a sudden increase in the frequency with which that token appears in a data stream. Like "event," "burst" is also vague and can be applied at various levels of temporal granularity (seconds, minutes, hours, days, etc.). As an example, "Obama" would have experienced a burst in usage over a period of months leading into the 2008 Democratic Presidential Primary as President Obama was relatively unknown prior to that. Tokens like "earthquake" or "superbowl," on the other hand, would experience more drastic bursts in much shorter periods just around the time of a specific event. We refer to such tokens as "bursty." Again,

for our purposes, we focus only on bursts that occur within a minute (or just a few minutes). It's worth noting here that bursts need not be symmetric; that is, a drastic uptick in frequency need not be followed by an equally dramatic dip in usage.

Now, if we let E denote the set of all minutes t in which an event occurs, then the indicator function $\mathbb{1}_E(t)$ returns a 1 for all times t in which an event occurs, and 0 for all other values of t. We can now define the event detection task as constructing a function that approximates this indicator function $\mathbb{1}_E(t)$. 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 relax this task slightly by using the set E' where, for all $t \in E$, $t, t+1, t+2 \in E'$. False positives/negatives and true positives/negatives follow in the normal way for some candidate function $\widehat{\mathbb{1}_E}(t)$: a false positive is any time t such that $\widehat{\mathbb{1}_E}(t) = 1$ and $\mathbb{1}_E(t) = 0$; likewise, a false negative is any t such that $\widehat{\mathbb{1}_E}(t) = 0$ and $\mathbb{1}_E(t) = 1$. True positives/negatives then operate as expected.

3.2 Experimental Framework

Having established the vocabulary and tasks, we can now turn our attention to the actual mechanics of comparing our LABurst technique with the baseline, which primarily focuses on the sports context. Sporting competitions adhere well to our definition of event in that a sporting event has a well-defined place and time and can contain many unpredictable sub-events, such as scores, fouls, ejections, or other dramatic moments of play. Additionally, sporting events with large followings occur fairly often and with a good degree of regularity, greatly simplifying data collection. As such, our main experiment here is a comparative study in detecting a collection of events in three major sporting competitions: the 2013 Major League Baseball (MLB) World Series, 2014 National Football League (NFL) Super Bowl, and 2014 Fédération Internationale de Football Association (FIFA) World Cup.

In each competition, we extract the times of four basic types of events (beginning of the game, end of the game, scores, and penalties) from existing blog posts, news articles, and social media data to construct game timelines. These events then comprise the ground truth against which we compare LABurst and the baseline. That is, positive instances of events are those minutes in which one of these events occurs, and negative instances are those minutes in which no event occurs. From this data, we generate receiver operating characteristic (ROC) curves for both LABurst and the baseline across each sport and then a composite ROC curve for all sports and compare the area under the curves for both methods.

We then follow up this experiment with an additional test of whether LABurst can detect earthquake events similar to Sakaki's earthquake detection task. This investigation compares LABurst with the frequency of the keyword "earthquake" by applying LABurst's models learned in the first experiment to social media captured during two large earthquakes: 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.

3.2.1 Data Collection and Additional Events

Though this work should apply to any social media stream, much of the development leverages Twitter since a large amount of research on this social network already exists, and data is relatively easy to acquire. Our data on the main sporting events used for evaluation (the World Series, Super Bowl, and World Cup) come from Twitter's 1% public sample stream using source code from Jimmy Lin's twitter-tools library[1].

For training LABurst, however, we require additional data to model bursty events, so we leverage two other Twitter data sources as well: an excerpt from the Edinburgh Twitter Corpus [12] and a selection of tweets from the Twitter Firehose covering Argentina in November of 2011. From these sources, we are able to generate a series of timestamped event timelines and related tokens for several additional sporting events as well:

- The 2010 NFL National Football Championship game,
- Four premier league soccer games in November of 2012,
- The National Hockey League's (NHL) 2014 playoffs,
- The National Basketball Association's (NBA) 2014 playoffs,
- The 2014 Kentucky Derby and Belmont Stakes races,
- And a selection of early games from the 2014 FIFA World Cup.

3.3 Baseline Event Detection

Inspired by existing work like that from Twitter's blog on detecting goals during the World Cup and others, we constructed a simple event detector that uses per-minute frequencies for a small set of target tokens [4]. These target tokens should be keywords that are likely to exhibit bursts in usage during related events, such as "goal" for goals in soccer/football or hockey or "run" for runs scored in baseball. Our baseline implementation also supports some rudimentary normalization to collapse modified words to their originals (e.g., "goooooooaaaallll" down to "goal"). We know such a technique is 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 ("earthquake" in Sakaki's work for example [16]).

At a high level, we compare the most recent count of these target tokens against the average count over the past few minutes, and if the difference between the two is above some threshold, we claim an event just occurred. More formally, we define the following: a time series T segmented into m minutes, a set of event-related seed tokens S such that $s_i \in S$ is one of these event-related tokens, and a function `count`(s_i, t_j) that returns the frequency of token s_i in minute t_j. The frequency for a given minute t_j is then defined by the function `freq`(t_j) shown in Eq. 1. We also

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

construct a sliding window w of size $|w|$ such that w_k includes minutes t_k to $t_{k+|w|-1}$ and define an average over this window as $\texttt{avg}(w_k)$, shown in Eq. 2.

$$\text{freq}(t_j) = \sum_{i=0}^{|S|} \text{count}(s_i, t_j) \tag{1}$$

$$\text{avg}(w_k) = \frac{\sum_{j=k}^{k+|w|-1} \text{freq}(t_j)}{|w|} \tag{2}$$

Given these functions, we take the difference Δ_k between the frequency at time $t_{k+|w|-1}$ and the average for window w_k such that $\Delta_k = \texttt{freq}(t_{k+|w|-1}) - \texttt{avg}(w_k)$. If this difference exceeds some threshold Z such that $\Delta_k > Z$, we say an event was detected at time $t_{k+|w|-1}$.

Since our analysis covers three separate types of sporting competitions, the seed keyword list for this method must include tokens from the vocabulary of each. We avoid separate keyword lists for each sport to provide a more even comparison to the general nature of our language-agnostic technique. The keywords for which we searched were as follows: "goal", "gol", "golazo", "score", "foul", "penalty", "card", "red", "yellow", "touchdown", "td", "fieldgoal", "points", "run", "home", "homerun". 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".

3.4 Language-Agnostic Event Detection

As mentioned, existing research achieves acceptable performance in streaming event detection by tracking frequencies for a small set of bursty keywords when a real-world event occurs. We extend such approaches by developing a technique for automatically identifying these bursty tokens and detecting based on those frequencies, which should hopefully provide more insight and additional flexibility. To this end, we constructed a set of temporal features to model the characteristics of a bursty token. Like other approaches, these temporal features were built around a sliding window of several minutes with each window further divided into overlapping slices as shown in Figure 1. In this manner, we construct a windowed time series containing the most recent frequencies for each token, from which our features are generated.

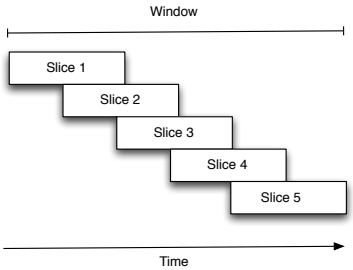

Figure 1: Sliding Window and Overlapping Slices

From these features, we use AdaBoost to combine support vector machines (SVMs) and random forests (RFs) into an ensemble classifier to discriminate between bursty and non-bursty tokens. To cast this task as a machine learning problem, however, we require positive and negative token samples on which our classifier can be trained. While obtaining positive samples of bursty tokens is straightforward (we can at least use keywords from the baseline as well as event-specific tokens like names of scoring players or locations of events), determining negative samples is more difficult since it is hard to know all the events that may burst on Twitter at any moment. Fortunately, we can circumvent this difficulty by automatically classifying all stop words as negative, non-bursty tokens, and Python's NLTK[2] library provides convenient lists of stop words for several languages.

It is important to note here that we perform this classification without any semantic or language-based filtering or normalization, and the only data we discard in this analysis are retweets and hashtags.

3.4.1 *Temporal Features*

While previous work already covers burst detection, only a subset is relevant to the streaming case presented herein, and we integrated those into our system. Beyond these existing features, we also developed a number of features that should yield higher weights for tokens that deviate significantly from normal posting frequencies:

- **Frequency Regression** Given the log of a token's frequency at each slice in the current window, take the slope of the best-fitting line.
- **Message Frequency Regression** Given the log of the number of tweets in which a token appears for each slice in the current window, take the slope of the best-fitting line.
- **User Frequency Regression** Given the log of the number of users using a token at each slice in the current window, take the slope of the best-fitting line.
- **Average Frequency Difference** The difference between the token's frequency in the most recent slice and the average frequency across the current window.
- **Message Average Frequency Difference** The difference between the number of messages in which a token appears in the most recent slice and the average number of messages containing that token across the current window.
- **User Average Frequency Difference** The difference between the number of users who use a token in the most recent slice and the average number of users across the current window.
- **Inter-Arrival Time** The average number of seconds between token occurrences in the given window.
- **Entropy** The entropy of the set of tweets containing a given token.
- **TF-IDF** The term frequency, inverse document frequency for a each token.

[2]http://www.nltk.org

- **TF-PDF** A modified version of TF-IDF called term frequency, proportional document frequency [2].
- **BursT** Weight using a combination of a given token's actual frequency and expected token frequency [6].

We also normalize each window's feature vectors into the range $[0, 1]$ to avoid biases from scale during classification by taking the maximum and minimum values for each feature in the current window.

3.4.2 *Training the Ensemble Classifier*

Discriminating between bursty tokens and stop words necessitated the use of a classification algorithm, and many different such algorithms exist. In particular, the Scikit-learn Python package provides implementations for SVMs and RFs as well as an implementation of the ensemble classifier AdaBoost [10]. Both SVMs and RFs have tunable parameters to select before integrating into AdaBoost, however, so we developed a grid search strategy to select parameters based on the F1 scores on our training and testing data.

For SVMs, we used the radial basis kernel, which has two parameters: cost c and kernel coefficient γ. In searching the space of c and γ, the grid covered 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 trained thirty different classifiers using repeated random subsampling, took the average F1 score, and selected the parameter set with the highest F1 score. Selecting parameter values for RFs was 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 1.

Table 1: Classifier Parameter Scores

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

These two classifiers were then combined using the Scikit-learn's AdaBoost implementation with four estimators. We then applied the resulting AdaBoost classifier to all the training data and expanded our set of known bursty tokens with those tokens that had a greater than 90% likelihood of being part of the bursty class in one round of self-training (Scikit's AdaBoost implementation provides likelihoods for tokens it predicts).

Regarding sliding window and slice size, preliminary investigations seemed to illustrate a window size of ten minutes with a slice size of three minutes (each slice overlapped the next by two minutes) lead to acceptable results.

4. EXPERIMENTAL RESULTS

To restate, the research question posed in this work is to determine whether a language-agnostic, streaming event detection scheme can perform as well as a domain-specific frequency-based method in detecting events in 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.

For each event, we generated ROC curves by varying a threshold parameter and calculating the true and false positive rates. In the baseline approach, our threshold parameter controlled the minimum difference between current frequency and average frequency over the sliding window. For our LABurst method, the ROC curve was generated by varying the minimum number of tweets a window must contain for an event to be detected. We then compared the area under the curve (AUC) metric to determine the difference in performance between the two methods. Prior to presenting comprehensive results, we first present performance curves for each event type.

4.1 2013 World Series

For the 2013 World Series between the Boston Red Sox and the St. Louis Cardinals, we explored only the final two games on 28 October and 30 October of 2013. These games contained fourteen events of interest: four game starts/ends and ten instances of points scored. The two tested techniques exhibited similar performance, with a difference of only 0.02 (the baseline with 0.76 and the language-agnostic bursty method with 0.74). Figure 2 shows the how these two curves compare graphically, and we can see that neither curve completely dominates the other, and both perform better than random guessing.

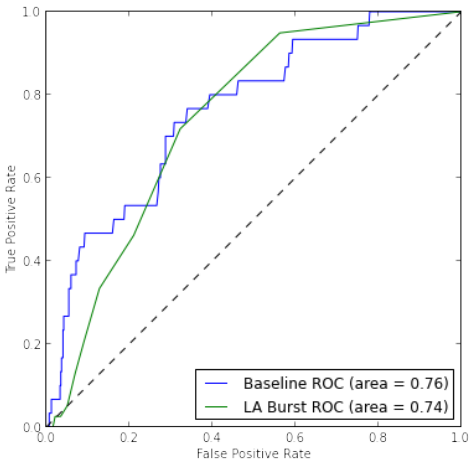

Figure 2: ROC Curves for the 2013 World Series

4.2 2014 Super Bowl

Since the 2014 Super Bowl was a single-day competition, we covered the entire game between the Seatlle Seahawks and the Denver Broncos on 2 February 2014, which contained twelve events: a game start, an end, and ten instances of points scored. The difference between the two mechanisms (shown in Figure 3) is more pronounced with a difference in AUC near 0.1, with the baseline performing better. LABurst exhibited a much higher false-positive rate in comparison to the baseline, which may be explained later in 5.2.

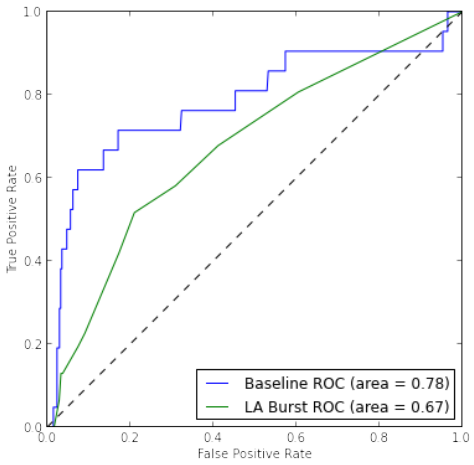

Figure 3: ROC Curves for the 2014 Super Bowl

4.3 2014 World Cup

As with the World Series, our World Cup analysis covered 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. These matches contained the most events with a total of seventeen: four game starts/ends, nine penalty cards issued, and four goals scored. Unlike the World Series and Super Bowl, however, the difference between the baseline and LABurst shows our LABurst method actually outperforms the baseline here with a difference in AUC of approximately 0.05.

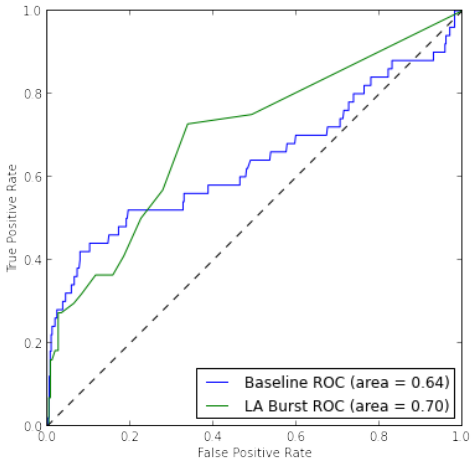

Figure 4: ROC Curves for the 2014 World Cup

4.4 Composite Results

To compare comprehensive performance, we look to Figure 5, which shows ROC curves for both methods across all three event types. From this figure, we see the two methods perform nearly identically with AUC values of 0.7187 for the baseline and 0.7052 for our language-agnostic technique. 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), the baseline method shows 0.5581 and 0.1408 respectively with a difference of 0.4174 at a threshold value of 13.2. Our language-agnostic method, 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 our approach achieves a higher true positive rate but at a cost of a higher false positive rate as a result.

4.5 Earthquake Detection

Detecting sub-events within sporting competitions as described above is a useful task for areas like advertising or automated highlight generation, but a more interesting and worthwhile task would be to detect higher-impact events like natural disasters. The typical frequency-based approach is more difficult here as it is impossible to know what events are about to happen, and a list of target keywords to detect all such events would be long, leading to false positives. Our method could be highly beneficial here as one would not need to know the target language or other pre-specified information. Since Sakaki showed the feasibility of detecting earthquakes using Twitter, we pulled Twitter data for two earthquakes in Japan: a 7.1-magnitude quake off the coast of Honshu on 25 October 2013, and a 6.5-magnitude quake off the coast of Iwaki on 11 July 2014.

To determine whether our approach could detect these earthquake events, we applied the classifier trained and tested for the sporting domain to these Twitter sets and tracked the frequency of the term "earthquake" simultaneously. Figures 6 and 7 show the frequencies 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). From these figures, one can see the token "earthquake" sees a significant increase in usage when the earthquake occurs, and our language-agnostic method experiences a similar increase at the same moment for both events. That is, both techniques identify the earthquake simultaneously.

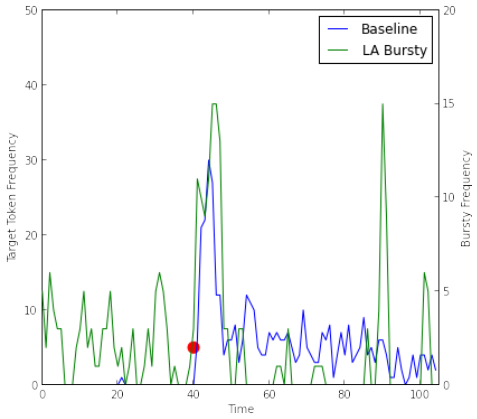

Figure 6: Honshu, Japan Earthquake - 25 October 2013

Given our method's success here, 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 2. We

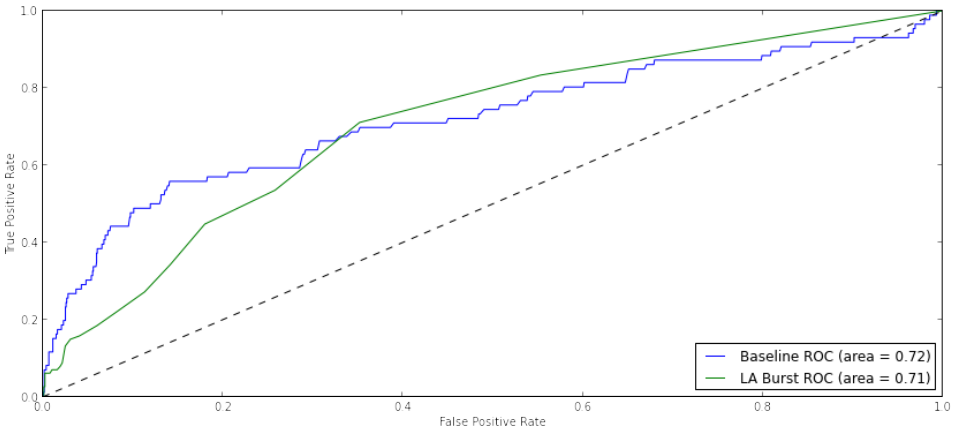

Figure 5: Composite ROC Curves

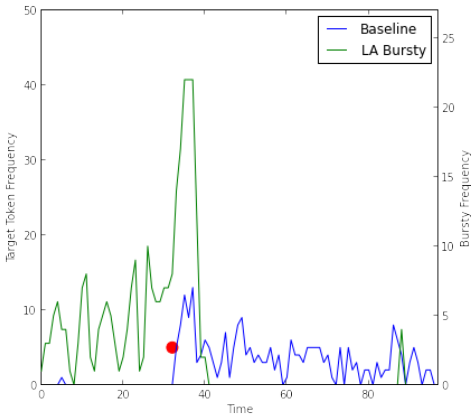

Figure 7: Iwaki, Japan Earthquake - 11 July 2014

also extracted tweets from our data 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," "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[3] translates these tweets as "Ah 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 2: Tokens Classified as Busting During Events

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

5. ANALYSIS

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

In comparing the baseline and language-agnostic techniques, it is important to understand the baseline provides little in the way of discovering previously unknown tokens or significant events that do not conform to a priori knowledge. The real power of the language-agnostic method described herein addresses such deficiencies directly by identifying specific tokens that burst along with a significant event *and* by capturing unexpected events.

5.1 Identifying Event-Related Tokens

As mentioned, where the baseline requires the user to specify interesting or event-related tokens prior to any data processing or analysis, our approach identifies these 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 3 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 3; 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 likewise for Germany's Götze in the third event (and the

Table 3: 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	goaaaaalllllllll, goallllll, godammit, goetze, gollllll, gooooool, gotze, gotzeeee, götze, nooo, yessss, ドイツ

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).

5.2 Undocumented Event Discovery

One particular weakness present in the baseline is that it is unable to capture unexpected events or events that do not conform to the keyword list. This deficiency means analysts might miss significant events within these competitions, especially if they are not directly related to scores or penalties, such as Uruguay's Luis Suarez's biting the Italy's Giorgio Chiellini during a World Cup match on 24 June since no foul was called at the time. Other instances of particularly dramatic play or events that happen at the larger sporting event but not necessarily on the field might be missed as well.

We can see instances of such omissions in the last game of World Cup. Figure 8 shows the frequencies for target tokens for the baseline in blue and bursty tokens for our method in green. From this graph, we can see the first, obvious incidence in Peak #1 where our bursty method exhibits a peak that is missed by the baseline 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. Peak #2, slightly more than eighty minutes into the data (which is sixty minutes into the match), our burst analysis sees another peak that is otherwise relatively minor in the baseline graph. 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.

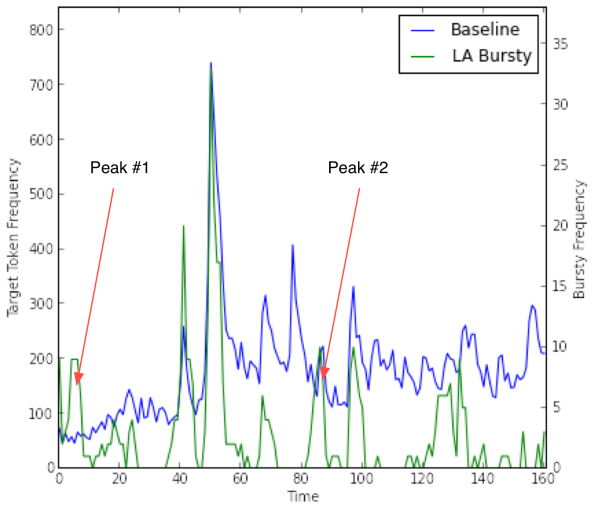

Figure 8: Baseline and LA Bursty Frequencies

Given our detection of these additional non-score-/non-penalty-related events, it is perhaps unsurprising that our burst detection technique exhibits a higher false-positive rate compared to the baseline. Since our approach is both language- and domain-agnostic, it makes sense that it would detect additional events beyond the game start/end, score, and penalty events our data counts as ground truth. A better comparison between the accuracies of the two approaches may be to identify only those peaks in which the keywords used in the baseline are identified as bursty in our method, but such a test disregards our approach's additional power.

5.3 Real-Time Usage and Event Persistence

In comparing these two event detection methods, we must also address their abilities to handle streaming data and the lag between an event and its detection by either of these mechanisms. The baseline technique processes data minute by minute and therefore has at most a minute of lag between input and discovery. Our technique, on the other hand, exhibits a lag equal to the slice size, which in the case of this paper is three minutes.

Another important aspect to consider is the length of time in which an event's peak persists. For the baseline, event detection achieves its highest accuracy when events are flagged in the minute they occur and the minute immediately following. Our approach logically follows the slice length such that events persist for three minutes.

6. FUTURE WORK

While the experiments outlined herein establish the utility of language-agnostic event detection, several avenues of research can follow up on this foundation. First, though

this work is applied in the streaming context, little effort was made to enforce near-real-time computation constraints; with the growing popularity of stream-centric processing frameworks like Apache Storm (as used by Petrović et al.) or Apache Spark Streaming, one has considerable latitude in exploring ways to enforce such constraints. Secondly, our selection of classifiers used in our ensemble was based primarily on ease of use given our features; deep learning systems are more complex but could provide enhanced capabilities in detecting bursting patterns in social media streams. Thirdly, we specifically targeted short-term, instantaneous events, but it is possible that these same techniques could be applied at different levels of granularity (per hour or per day for instance) to detect events of larger scales; one could then apply Kleinberg's notion of event hierarchies across multiple temporal granularities and track different aspects of the same event. Additionally, an implicit assumption made in this work is that tokens that experience bursts at the same time are related in some way, which allows us to detect events using token frequency; this assumption has interesting consequences with regarding to language-agnostic topic detection. That is, one could cluster tokens with similar temporal signatures to identify topics across languages, which would otherwise be impossible if one were to rely solely on semantic similarity or would require applying machine translation to reduce all text in the stream to the same language. Finally, though this paper exclusively leverages data from Twitter's public stream, our techniques should be applicable to streams from other social networks as well, and how events burst on more media-centered networks like Flickr or Pinterest might reveal interesting photographic representations of an event.

7. CONCLUSIONS

To revisit our motivations, the goal for this experiment was to demonstrate the feasibility of detecting events and event-related tokens through analyzing temporal characteristics from unfiltered Twitter data streams. While many social media-based event detection systems require some form of query-based filtering and language model processing, our approach is more flexible, lighter weight, and easily adaptable to different domains. Our results show that by leveraging temporal characteristics to identify bursty tokens and using frequency of these bursty tokens, we can detect significant events across a collection of disparate sporting competitions with a level of performance nearly equivalent to an existing, domain-specific baseline.

Similar performance to the 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 detect events we did not expect to occur, 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.

8. 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.

9. REFERENCES

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

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

[3] M. Cataldi, L. Di Caro, C. Schifanella, U. Torino, and L. D. Caro. Emerging topic detection on Twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, MDMKDD '10, pages 4:1—-4:10, New York, NY, USA, 2010. ACM.

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

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

[6] C.-H. Lee, C.-H. Wu, and T.-F. Chien. 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, pages 548–557, Berlin, Heidelberg, 2011. Springer-Verlag.

[7] R. Long, H. Wang, Y. Chen, O. Jin, and Y. Yu. Towards effective event detection, tracking and summarization on microblog data. In *Proceedings of the 12th international conference on Web-age information management*, WAIM'11, pages 652–663, Berlin, Heidelberg, 2011. Springer-Verlag.

[8] M. Nagarajan, K. Gomadam, A. P. Sheth, A. Ranabahu, R. Mutharaju, and A. Jadhav. Spatio-Temporal-Thematic Analysis of Citizen Sensor Data: Challenges and Experiences. In *Proceedings of the 10th International Conference on Web Information Systems Engineering*, WISE '09, pages 539–553, Berlin, Heidelberg, 2009. Springer-Verlag.

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

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*,

12:2825–2830, 2011.

[11] S. Petrović, M. Osborne, and V. Lavrenko. 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, pages 181–189, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

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

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

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

[15] Y. Raimond and S. Abdallah. The Event Ontology, 2007.

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

[17] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. TwitterStand: news in tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 42–51, New York, NY, USA, 2009. ACM.

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

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