

Eyewitness: Identifying Local Events via Space-Time Signals in Twitter Feeds

John Krumm and Eric Horvitz
Microsoft Research
Microsoft Corporation
Redmond, WA USA
{jckrumm|horvitz}@microsoft.com

ABSTRACT

We present a methodology for automatically extracting and summarizing reports of significant local events from large-scale Twitter feeds. While previous work has relied on an analysis of tweet text to identify local events, we show how to reliably detect events using only time series analysis of geotagged tweet volumes from localized regions. The algorithm sweeps through different spatial and temporal resolutions and finds events as anomalous spikes in the rate of geotagged tweets. We applied the approach to a corpus of over 733 million geotagged tweets. Using a panel of 103 crowdsourced judges who tagged 2400 detected events, we achieved a local event detection precision of 70%. Using these judged events as ground truth, a decision tree classifier was able to raise the detection precision to 93%.

Categories and Subject Descriptors

I.5.3 [Pattern Recognition]: Clustering, I.5.4 [Pattern Recognition]: Applications, I.7 [Document and Text Processing]

General Terms

Algorithms, Experimentation.

Keywords

Local events, microblog, Twitter.

1. INTRODUCTION

The Twitter microblog service provides unprecedented access to accounts of local events from people at the scene. Eyewitnesses with connected devices can report on local events before any traditional news outlet could discover, interpret, and broadcast. Twitter readers and posters recognize the value of the service for providing access to information on local events as they break. Teevan *et al.* have shown that Twitter searches are more likely aimed at real-time content and breaking news compared to regular Web searches [1].

We address the challenge of identifying when Twitter posts (tweets) reference the occurrence of interesting local events. The benefits and challenges of finding local events in Twitter data are framed by the enormous volume of tweets, which was estimated at 500 million per day in a 2014 official Twitter blog [2]. With so

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
SIGSPATIAL'15, November 03-06, 2015, Bellevue, WA, USA
© 2015 ACM. ISBN 978-1-4503-3967-4/15/11...\$15.00
DOI: <http://dx.doi.org/10.1145/2820783.2820801>

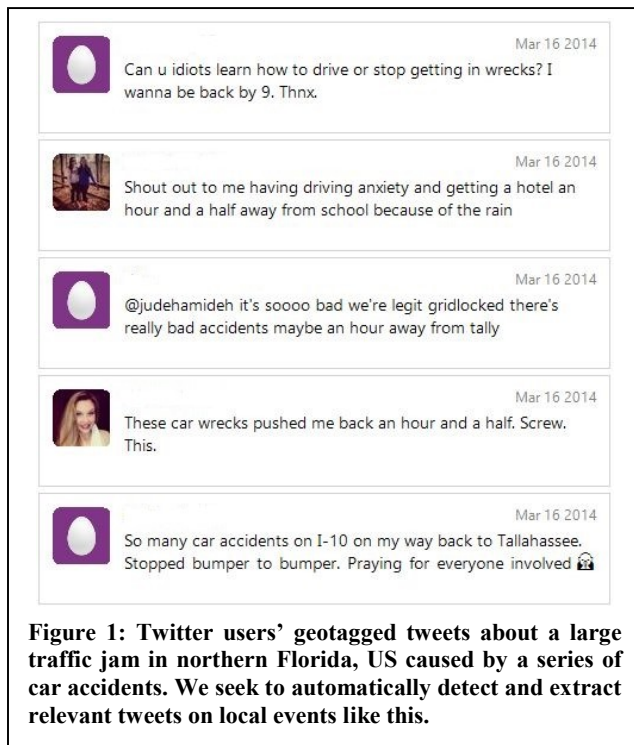


Figure 1: Twitter users' geotagged tweets about a large traffic jam in northern Florida, US caused by a series of car accidents. We seek to automatically detect and extract relevant tweets on local events like this.

many posts, it can be difficult to extract and cluster only those pertaining to a local event.

We introduce methods for automatically extracting local events from a large-scale stream of tweets. The methods address the challenge that tweets from different locations may be about local or non-local topics. We consider the spatial statistics of tweets and employ machine learning to build a classifier for local interesting tweets. We envision the methods being harnessed in a system that lets users identify and review local events across wide regions of time and space, or to monitor specific regions, such as events from their home town, college, or vacation location.

Typically, local events in Twitter are documented by multiple people, such as the multiple tweets about an anomalous traffic jam in northern Florida displayed in Figure 1. The tweets' time stamps, location data, and text give the basic facts, but the text also gives a rich account of the event in terms of emotions, meaning, and consequences.

We take our definition of a *local event* from Lee, who defines it as "something that happens at some specific time and place" [3], *i.e.* it is limited both temporally and geographically. We show that spikes in the rate of tweets over limited spans of time and space are usually associated with a local event. We verify this with human judges.

Localizing a tweet in time is straightforward given that all tweets come with a time stamp. However, it is often challenging to pinpoint the location from which a tweet was posted. The user's home location is sometimes available as a text string in their user profile, but this is imprecise. It is possible to infer a user's location from locations of their friends, but this is only at the resolution of a city [4]. Some tweets come with a latitude/longitude geotag that is automatically sensed from the user's device and attached to the post. Watanabe *et al.* estimate the percentage of geotagged tweets at 0.7% [5]. Both [5] and [6] assert that this low rate of geotagging is insufficient to support the detection of local events, and [3] claims the geotags are too imprecise for this purpose. We show, however, that using only geotagged tweets is sufficient for precision detection of local events. The traffic jam in Figure 1 is one such example.

The algorithm presented here, called *Eyewitness*, looks through a corpus of geotagged tweets, systematically scanning over localized regions of time and space for unusual spikes in the volume of tweets. For a given region on the ground, the algorithm learns a regression model that predicts how many geotagged tweets to expect as a function of time in normal, default situations. If the actual volume of tweets in a time span exceeds the prediction by a significant amount, then we declare a local event. A simple text summarization algorithm serves to extract a handful of tweets describing the event. For evaluation, a panel of over 100 anonymous, crowdsourced judges examined the extracted tweets and voted whether or not they represented a local event, giving an initial precision of 70% for the extraction process. Postprocessing with a machine learning classifier trained on the judging results increased local event detection precision to 93%, with a relative recall rate of 90%. This process provided insights about the features of events that people would consider a local event, including the event's extent in space and time.

2. PREVIOUS WORK

Some especially important local events lead to wider-ranging commentary that spreads in space and lingers in time, including news reports. The unique advantage of Twitter is that it contains posts from eyewitnesses of the local event as it happens. These posts about local events have approximately the same time, location, and topic. Previous work has looked at finding microblog posts that have one or more of these elements in common.

Tweets that share an approximate topic and time often represent an event, but not necessarily a local event. As an example, TwiCal [7] finds events in Twitter that share a topic and date, characterizing them by a named entity, event phrase, and event type, but not a location. In [8], Lee *et al.* perform a detailed analysis of tweet text to find trending keywords, identifying events that evolve over time, but are not spatially coherent. Popescu and Pennacchiotti [9] look specifically for controversial events in Twitter that occur over a one day period and share a celebrity name, using machine learning to choose which tweets to include in the event.

Looking at just the topic and location (but not time), Yin *et al.* examine geotagged Flickr comments [10]. Their clustering finds topics that are characteristic of a given location, but that are not necessarily synchronized, transient events like ours, because of the missing time dimension.

Several researchers have built systems to detect local events by looking at clusters in time, location, and topic simultaneously. One example is NewsStand by Samet *et al.* [11]. Instead of microblogs, it monitors over 10,000 RSS news sources, inferring the relevant locations mentioned from each article's text and providing a map-

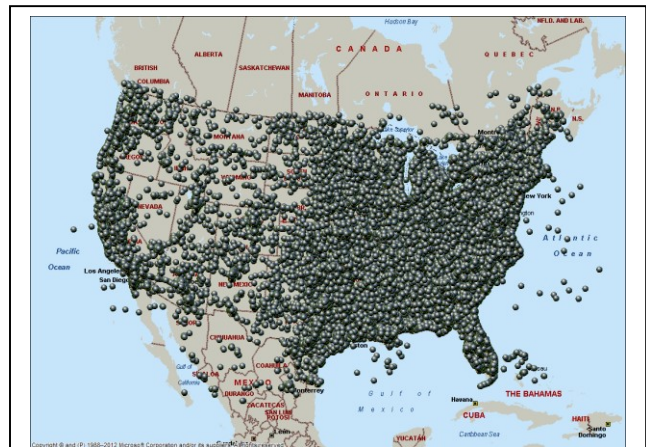


Figure 2: Random subsample of tweets showing the locations of about 0.05% of the tweets that we examined.

based browser for finding stories in locations of interest to the user. Zhou [12] presented an algorithm to create a detailed clustering of tweets and tested it on two natural disasters. The system infers the event's location from the tweet text and uses a sophisticated model to highlight tweets describing the evolution of the event. Sakaki *et al.* [13] developed a system to find tweets about earthquakes and typhoons, including an inference of the varying location of a typhoon based on Bayesian filtering of geotagged tweets. The EvenTweet system finds anomalous bursts in tweet keywords and then estimates the spatial extent of the bursts from those tweets that are geotagged [14].

Less directly related to our work are efforts to use Twitter in constructing a detailed analysis of a known event. An example is the effort by Hu *et al.* [15], who connect tweets to known events such as a political speech or debate.

We shall discuss other previous work below as it pertains specifically to event detection. As will become apparent, the novel aspects of the work presented here are:

- In contrast to assertions in [3, 5, 6], we find that geotagged tweets *are* sufficient for high-precision detection of local events. Thus, it is not necessary to infer locations from tweet text or user profiles.
- We do not need to examine tweet text to find local events. Instead, it is adequate to detect anomalous spikes in tweet volume in concentrated regions of space-time. Simple text summarization drawn from content of sets of such identified anomalous tweets can describe the event as a post-processing step.
- We can reliably detect local events in a principled way by regression analysis on time series of tweet volume.
- By scanning through different size space-time regions, we show which sizes are more likely to contain local events..
- We validate our results with over 100 human judges who assessed 2400 candidate local events, which is the largest evaluation to date of such a system.

Before explaining the event detection algorithm in detail, we describe the Twitter dataset at the core of our studies.

3. TWITTER DATA

Our experiments are based on 733,865,824 geotagged tweets collected over several months from mid-2013 to mid-2014. These came from the Twitter *firehose* via an agreement between Twitter and our institution. The tweets are limited approximately to the US

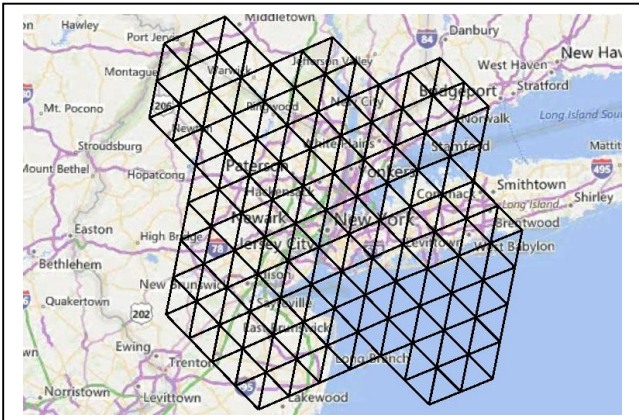


Figure 3: Depiction of some of the triangles from the hierarchical triangular mesh (HTM) at level 10 around New York City.

with a bounding rectangle. Figure 2 shows a random sample of the tweets’ locations based on their geotags. We excluded tweets whose text began with “I’m at”, because these are generally generated automatically and do not contain user-authored content. We excluded retweets and only included tweets marked as English language. If we found tweets with identical text, we only retained one of them. Eliminating retweets and repeats helped ensure that each local event tweet is a fresh observation. Each resulting tweet was represented with a time stamp, latitude/longitude, user ID, and tweet text.

4. DISCRETIZING SPACE AND TIME

Our algorithm for detecting local events is ultimately an exhaustive search over tweets through space and time. This search is made feasible by discretizing space and time into discrete pieces in space-time denoted by a space-time ordered pair (S, T) . Space and time are discretized separately, as described next.

4.1 Discretizing Space with the Hierarchical Triangular Mesh

We discretize the surface of the earth with the hierarchical triangular mesh (HTM) [16]. This is a 2D tessellation of a sphere consisting of nearly equal size, nearly equilateral triangles. The mesh comes at discrete levels of resolution, and Figure 3 shows some triangles from HTM level 10. Moving up one level in resolution consists of dividing each triangle into four smaller triangles, as illustrated in Figure 5. For this work, we scanned through HTM levels 8 through 11. Their sizes are shown in Table 1, with the levels’ areas spanning from 15.2 km² to 972.9 km².

Table 1: Sizes of triangles in our spatial grid.

HTM Level	Side Length (km)	Area (km ²)
8	62.4	972.9
9	31.2	243.2
10	15.6	60.8
11	7.8	15.2

Formally, we refer to the set of grid cells covering the earth at level L as the set S^L . Thus, the spatial component of the space-time ordered pair (S, T) is $S \in S^L$ for whichever level $L \in \{8,9,10,11\}$ we are working with.

These triangle sizes are our initial guesses at the spatial scale over which people tweet about a local event. Since we are only looking for local events, we do not use larger triangles, which would stretch the definition of local to a large swath of area. In the results, we show that some triangle sizes are more likely to correspond to local events than others.

Our uniform discretization stands in contrast to the Voronoi tessellation used by Lee and Sumiya for detecting local festivals [17]. They use k -means clustering, with a fixed k , to cluster tweets and form Voronoi regions that tend to be smaller for densely populated areas and larger otherwise. Our uniform tessellation make no implicit assumptions about how the extent of an event might vary with population nor any other factor.

Note that this discretization of the earth’s surface does not account for altitude. It would be straightforward to do so by adding a discretized vertical component, but the geotags of tweets do not have an altitude component, so two spatial dimensions are sufficient.

4.2 Discretizing Time

Local events have different durations. To capture a range of events of different duration, we discretize time into periods of length ΔT , which yields a set of time periods. One set of discretized time periods is

$$T_a^{\Delta T} \stackrel{\text{def}}{=} \{ \dots, [-2\Delta T, -\Delta T), [-\Delta T, 0), [0, \Delta T), [\Delta T, 2\Delta T), \dots \}$$

These are simply disjoint time intervals with length ΔT . Events likely will not fall neatly on these time boundaries, so we also use a set of discretized time periods offset by $\Delta T/2$. This set is

$$T_b^{\Delta T} \stackrel{\text{def}}{=} \left\{ \dots, \left[-\frac{3\Delta T}{2}, -\frac{\Delta T}{2} \right), \left[-\frac{\Delta T}{2}, \frac{\Delta T}{2} \right), \left[\frac{\Delta T}{2}, \frac{3\Delta T}{2} \right), \dots \right\}$$

$T_a^{\Delta T}$ and $T_b^{\Delta T}$, illustrated in Figure 4, are the same, except offset from each other by $\Delta T/2$ in an effort to catch local events starting near even or odd multiples of $\Delta T/2$. Thus the temporal component of the space-time ordered pair (S, T) is $T \in T_a^{\Delta T}$ or $T \in T_b^{\Delta T}$. For local events, we looked at six different time lengths,

$$\Delta T \in \{20 \text{ min}, 1 \text{ hr}, 3 \text{ hr}, 6 \text{ hr}, 12 \text{ hr}, 24 \text{ hr}\}.$$

The value of ΔT corresponds roughly to our guesses for the length of time people will tweet in response to a local event. We capped the size of ΔT to 24 hours as a limit of the duration of local events we attempted to find. In the results section, we show that some durations are more likely to correspond to the rise of significant local events than others.

4.3 Space-Time Prisms

The HTM discretizes space into triangles, and we discretize time into uniform, disjoint intervals. Thus each space-time piece is a prism with one dimension representing the time period being

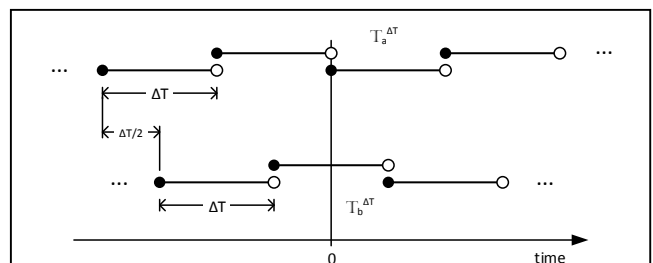


Figure 4: Two sets of time discretizations, $T_a^{\Delta T}$ and $T_b^{\Delta T}$, that are the same except for a $\Delta T/2$ offset.

considered. With four different spatial grid sizes L and six different time lengths ΔT , we examined 24 different combinations $L \times \Delta T$ of space-time discretizations. Finding an unusually large number of tweets in a space-time prism is indicative of a local event.

5. FINDING SPACE-TIME ANOMALIES

A key assumption we make is that significant local events are characterized by many people suddenly tweeting in a limited region for a limited time period. As an example, we examine tweets from an HTM triangle in northern Florida, USA shown as the center triangle in Figure 5, where our algorithm discovered a local event characterized by the tweets in Figure 1. This is a level 8 triangle ($L = 8$) whose ID happens to be $S = 831099$. This anomaly was discovered after discretizing time into periods of 3 hours, *i.e.* $\Delta T = 3$ hours.

For each HTM triangle at level L , we construct a discrete time series of tweet volume from that triangle, where the volume is measured in tweets per time period ΔT . The time series is denoted $y_{S,t}^{(L,\Delta T)}$, where $(L, \Delta T)$ refers to one of the 24 choices of spatial and temporal resolution, S indexes over the triangles in the tessellation, t indexes over the discrete time intervals, and y is the count of geotagged tweets in triangle S over discrete time period t . For the example in Figure 5, the time series would be denoted $y_{831099,t}^{(8,3 \text{ hours})}$. For simplicity of notation, we will drop the $(L, \Delta T)$ superscript and S subscript, remembering that there is a separate time series for each triangle S in the tessellation, for each triangle size L , and for each time discretization ΔT . Instead of complicating the notation to distinguish between the two offset time discretizations $T_a^{\Delta T}$ and $T_b^{\Delta T}$, we take it as implicit that we build time series for both offsets. Thus the time series representing the rate of geotagged tweets in a given triangle at some spatial and temporal resolution is simply y_t .

A partial time series for this example in northern Florida is shown as the thick, black curve in Figure 6.

The rate of tweets from inside the triangle follows a periodic, albeit noisy, daily pattern. There is also an obvious spike in the number of tweets, which our algorithm is designed to find.

5.1 Finding Anomalies

We find anomalies by first trying to predict the number of tweets from each triangle on the ground. If the prediction is significantly less than the actual number of tweets, we consider this an anomaly, likely corresponding to a local event.

Our predictions come from a regression function that estimates the time series values based on a number of parameters:

$$\hat{y}_t = f(\bar{\theta}_t)$$

where \hat{y}_t is the estimate of the number of tweets, $f(\cdot)$ is a learned regression function, and $\bar{\theta}_t$ is a vector of five numerical features.

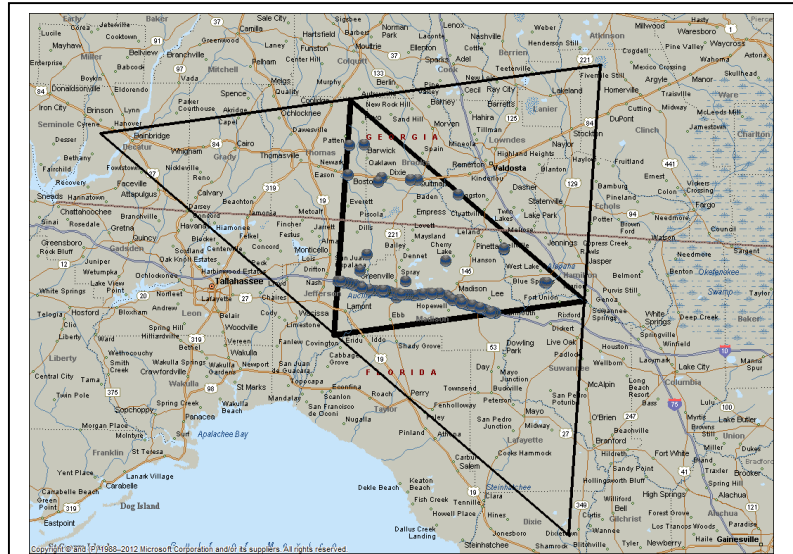


Figure 5: In this center triangle in northern Florida, USA, there was an anomalous spike in the number of tweets. The tweets' locations are shown as black dots, concentrated along U.S. interstate highway I-10.

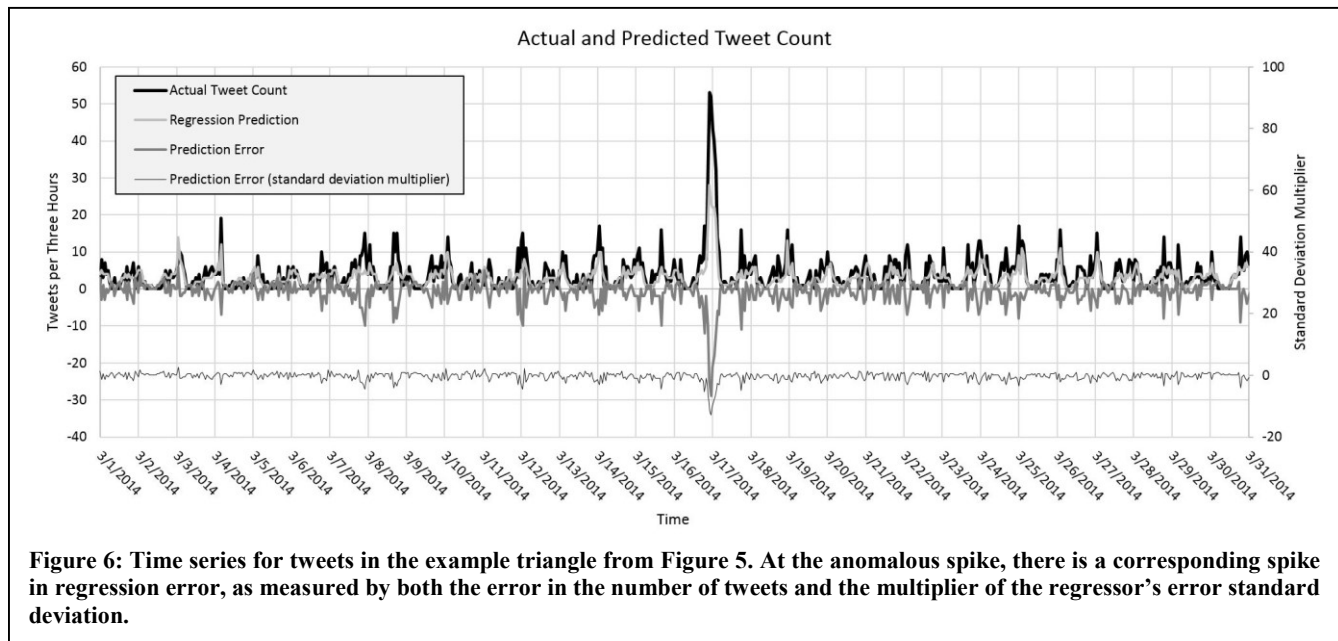


Figure 6: Time series for tweets in the example triangle from Figure 5. At the anomalous spike, there is a corresponding spike in regression error, as measured by both the error in the number of tweets and the multiplier of the regressor's error standard deviation.

These features are the time of the day, the day of the week, and the tweet counts from the three neighboring triangles, *i.e.* $y_{t,(1)}$, $y_{t,(2)}$, and $y_{t,(3)}$. Figure 5 shows the three neighboring triangles in the example. We chose to use the time of day and the day of week as features to capture the regular time-varying changes in tweet volume. For instance, if there is a consistent spike in the number of tweets on Saturday evenings, we wanted our regressor to predict this so it would not be mistaken as an anomaly.

We chose to regress on tweet counts in neighboring triangles particularly to find local events represented by a spike in volume that is *spatially isolated*. If tweets in neighboring regions rise and correctly predict a rise in the center triangle, then the increased tweet volume is likely not limited to the center triangle. This means the associated event is at a larger spatial scale than the center triangle and should be detected by the same analysis performed at a lower spatial resolution. This is why we sweep through different resolutions of triangles.

Looking for spatially isolated anomalies also means we avoid detecting large-scale news events such as a political election or celebrity news. These events are more likely to be captured by trending Twitter keywords, because the news quickly spreads from its source without regard for location.

The regression function is implemented as a FastRank regression tree, which is an efficient version of the MART gradient boosting algorithm. It learns an ensemble of decision trees, where the next tree in the ensemble is designed to correct the mistakes of the earlier trees [18]. Our particular instance was set up with a maximum number of leaves per tree of 32, minimum number of training instances in each leaf of 20, and 500 trees per ensemble. There is one ensemble of trees learned for each time series across all triangles, triangle sizes L , and time periods ΔT . By its nature, regression using decision trees produces a piecewise constant function. One convenient aspect of decision trees is that no normalizing nor preconditioning of the training data is necessary.

In Figure 6, the estimated time series \hat{y}_t is shown as a light gray curve. The prediction error at time t is $e_t = \hat{y}_t - y_t$, and a large negative error is indicative of a local event, because it means there were many more tweets than usual. The sample standard deviation of the prediction error, s_t , measures the precision of the regression function. The overall precision of the prediction varies from triangle to triangle. To account for this variation, we measure the prediction error as a multiple of the predictor’s precision, *i.e.* e_t/s_t . This value is shown as the bottom curve in Figure 6, plotted against the vertical axis on the right side. It is apparent that this value dips dramatically at the point of the anomalous spike in the rate of tweets. We declare a local event whenever this normalized error drops below -3.0, which means the actual number of tweets exceeded the predicted number by at least three times the standard deviation of the prediction error. This threshold is somewhat arbitrary, and we use it in a more refined way in a subsequent process described later. Normalizing the prediction error by the prediction error standard deviation means we can reasonably apply one threshold to all the triangles, and that it will automatically account for the variation in the regressor’s prediction error from place to place.

We find this approach to detecting local events to be valuable for the following reasons:

- The detector concentrates on local events because the regression function is designed to find only spatially isolated anomalies.

- By accounting for the effect of time of day and day of week, the detector ignores systematic temporal variations that do not indicate a local event.
- The detector automatically adjusts its sensitivity based on the precision of the regressor by using a normalized error threshold.

We refer to this portion of the Eyewitness algorithm as the *time series component* to distinguish from the local event classifier that we present in Section 7.

5.2 Space-Time Parameter Sweep

Some local events like earthquakes occur over a larger region than others, such as sports events in stadiums. Similarly, local events occur over different time durations. This is why we sweep over a set of different time periods,

$$\Delta T \in \{20 \text{ min}, 1 \text{ hr}, 3 \text{ hr}, 6 \text{ hr}, 12 \text{ hr}, 24 \text{ hr}\}$$

and different triangle levels,

$$L \in \{8,9,10,11\}.$$

As a reminder, ΔT is the time interval for each element of the tweet volume time series, which gives the count of tweets in each time interval. Our sweep of durations differs from [17] who looked at only 6-hour time periods for local festivals and from [19] and [20] who consider only 24-hour time periods.

By varying the spatial and temporal resolution of the time series, we hope to detect local events at different scales of space and time. We look at all combinations of resolutions, $L \times \Delta T$, for a total of $|L| \times |\Delta T| = 24$ different space-time resolutions. As we show in the results, this sweep shows which sizes of space-time prisms tend to capture significant local events.

Ideally for each triangle size, our algorithm would look at all the triangles in our study area covering the U.S. To save processing time, however, we look at only those triangles that cumulatively contain 95% of the total tweets over our study period. This eliminates many regions where there is very little Twitter activity and significantly speeds up processing.

5.3 Related Approaches

Other approaches to detecting anomalous events in Twitter related to ours includes the work of Hongzhi *et al.* [21], who developed a probabilistic mixture model governing the temporal evolution of keywords for a given user. It balances stable keyword topics that do not change over time and temporarily popular keywords that grow and shrink in volume. They give an example showing a temporary spike in the keywords relating to the death of Michael Jackson. In contrast, our approach ignores keywords in detecting anomalies, focusing instead on spikes in tweet volume in space-time prisms.

In their work on earthquake detection from Twitter, Sakaki *et al.* [13] observed that tweet volumes from local events often follow a decaying exponential curve that is indicative of a homogenous Poisson process. From this they were able to compute the probability of an event occurrence. They also build a support vector machine classifier to detect events based on the text of tweets that contain a given query word. We instead detect events based purely on the count of tweets coming from a region without regard to the tweets’ content. This means we can find events of any type without requiring any explicit query words to target certain event types.

Diao *et al.* [22] combine the idea of mixtures and the Poisson distribution to model tweets within a topic as a mixture of two Poisson distributions corresponding to the steady state background

and bursts of volume. The mixture moves between full background mode and full burst mode as a Markov chain through time.

Guille and Favre [23] detect events based on discovered keywords and “mentions” in Twitter, which are user names of other Twitter users. They build a probabilistic model of a tweet containing a keyword and a mention in a time slice. Their work is distinctive in that it exploits the social phenomenon of mentions and that it can also string together time slices for an event-specific estimate of the event’s duration.

In their search for festivals based on tweets, Lee and Sumiya [17] make box plots of the number of tweets, users, and incoming users in a spatial region. They then declare a local event when combinations of these features exceed a permissible range.

Lee [3] presents a method for finding real time event topics in Twitter. One of their features for weighting candidate events is the “burst score”, which is a function of the expected and actual arrival rate of key words. In [20], the authors model tweets from a topic as a binomial distribution of tweet frequencies and compute a “burst probability” from a sigmoid function based on parameters of the distribution.

TwitInfo [24] finds tweets with user-specified keywords and forms a time series giving the number of tweets per minute. Using a weighted average, the detection algorithm computes the expected frequency of tweets and the expected absolute deviation from the mean. TwitInfo declares an event around the keywords if the normalized absolute difference between the actual and expected frequency exceeds a threshold, where the normalization factor is the expected absolute deviation.

The anomaly detection algorithm closest to ours comes from Chae *et al.* [19]. Based on an extracted topic phrase, they extract all matching tweets and compute a time series giving the number of tweets per day. Using a seasonal trend decomposition procedure, they decompose the time series into the sum of a seasonal part, a trend part, and a remainder. If the remainder is large enough, this indicates an unusual volume of tweets for that topic phrase.

Compared to previous approaches to anomaly detection in Twitter, Eyewitness is novel in the following ways:

- We examine several different spatial resolutions and time slice durations to find local events with different extents in space and time.
- We detect anomalies without requiring any analysis of the tweet text, using only time stamps and geotagged locations. This eliminates the introduction of possible text-based bias.
- By looking at neighboring regions, the detector is set up to find only anomalies from a limited region on the ground, ignoring events that are not local.

5.4 Event Summary

The detection of a local event gives a space-time prism (S, T) that gives the event’s location and time. We expect that the tweets associated with the local event will give a meaningful summary of the event for human consumption. To summarize the event succinctly, we examine the text of constituent tweets and extract five for presenting. This stands in contrast to previous approaches for local event detection from Twitter, which normally examine the tweet text as part of the detection process. Instead, we look at the text only after the event is detected.

The specific task at this stage is to take the text from a group of tweets and choose tweets that are most representative of the group. This problem has been addressed by Inouye and Kalita [25] who

compared several summarization algorithms for tweets. They concluded:

Overall, it seems from these results that the simple frequency based summarizers, namely SumBasic and Hybrid TFIDF, perform better than summarizers that incorporated more information or more complexity such as LexRank, TextRank or MEAD.

Based on these results, we used SumBasic [26] to choose five tweets to summarize each local event detected by our algorithm. SumBasic was originally designed to pick out sentences to summarize a document. For our task, each tweet from the event is considered a sentence. After removing stop words from each tweet, SumBasic computes the frequency of each word in the collection of tweets. It then proceeds iteratively to choose those tweets with the highest frequency words. However, once a tweet is chosen, the frequency values of its constituent words are reduced, leading to a diversity of words in the choice of subsequent tweets. This is how we selected the tweets shown in the sample event in Figure 1.

5.5 Real-Time Modifications

Our detection algorithm is set up to process a corpus of stored tweets to find local events. As described, it is not capable of real time detection. Recall that we use a regression function $\hat{y}_t = f(\hat{\theta}_t)$ to predict the number of tweets at time t inside a triangle. The vector of regression parameters $\hat{\theta}_t$ includes tweet counts from the three neighboring triangles at time t (which are $y_{t,(i)}, i \in \{1,2,3\}$), meaning we have to wait for those counts to accumulate before making the prediction. This introduces a temporal lag of ΔT . For real time detection, we propose using a modified regression function that replaces the current counts in neighboring triangles with counts from the previous time step (which would be $y_{t-1,(i)}, i \in \{1,2,3\}$). We leave this for future work.

6. HUMAN EVALUATION


Evaluating everyday event detection has traditionally been difficult due to a lack of ground truth. One of the best previous examples of such an evaluation is from Popescu and Pennacchiotti [9] who presented a method to detect controversial events in Twitter. They had two expert judges label 800 detected events for testing. Even then, however, it is difficult to estimate a recall rate, since there is no guarantee that the ground truth contains every relevant event.


For our testing, we chose 100 detected events from the sweep through each of our 24 space-time resolutions, giving 2400 candidate local events. We used a panel of 103 crowd-sourced judges from our institution’s human judging system, which is similar to Mechanical Turk [27]. Each judge was required to be an English-speaker in the U.S., corresponding to the U.S.-based, English language tweets we used in our experimental corpus.


Each detected event was presented to its judges as an image and a multiple choice question as shown in Figure 7. The image shows the five summary tweets selected by the SumBasic algorithm along with a map showing the location of the event at three different zoom levels. Circles on the map approximate the triangle where the event was found. The maps were presented solely to make the judging task more interesting, which helps judges concentrate.


For each candidate local event, each judge answered the question, “Do three or more of these tweets seem related to the same local event as each other?” The available answers were “Yes”, “No”, and “Unsure”. We formulated this question carefully to pick out those sets of tweets that, in aggregate, seemed to be related to the same local event, because this was the goal of our system. We did not ask the judges to verify the existence nor location of the event. Before


Do three or more of these tweets seem related to the same local event as each other? Yes No Unsure

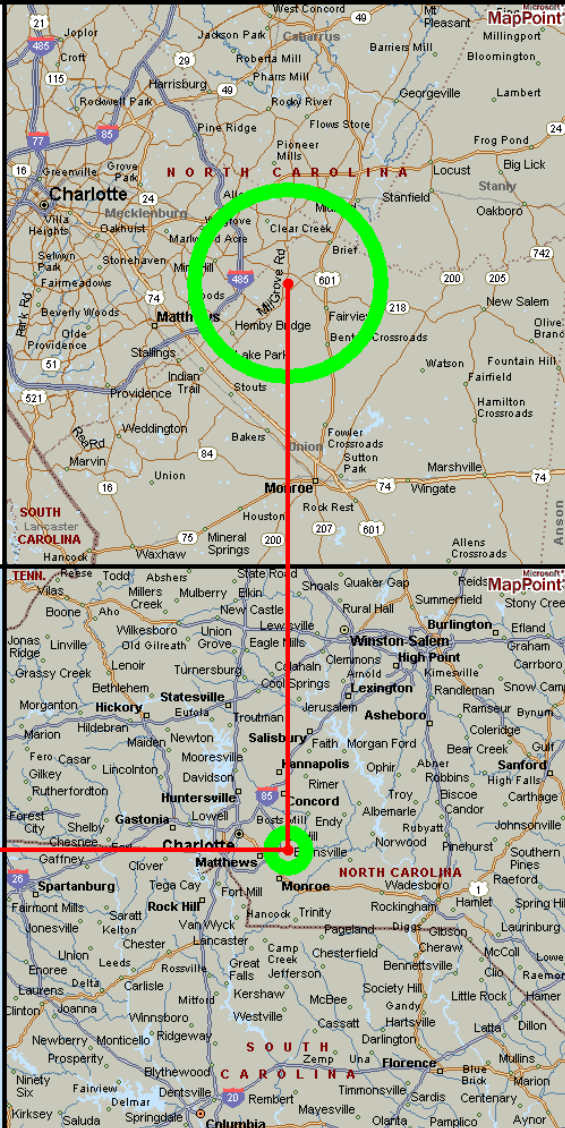
 Dec 31 2013
Wow!!! The giant disco ball at the Avett Bros. concert just fell and crashed through the stage... <http://t.co/5nK64X3fui>


 Dec 31 2013
Soo the ball dropped on New Year's Eve at the @theavettbros show...it also broke the stage.

 Dec 31 2013
Giant disco ball fell and crashed through the stage at @theavettbros show. Between acts, looks like no one hurt. <http://t.co/INHaHGukAX>

 Dec 31 2013
Giant New Year's Eve disco ball just fell and crashed through the stage. Indeed, 2014. Indeed. <http://t.co/0wOVF7yY9>

 Dec 31 2013
Ok y'all this huge disco ball just fell from the set and crashed through the stage at this concert. Literally the biggest disco ball ever





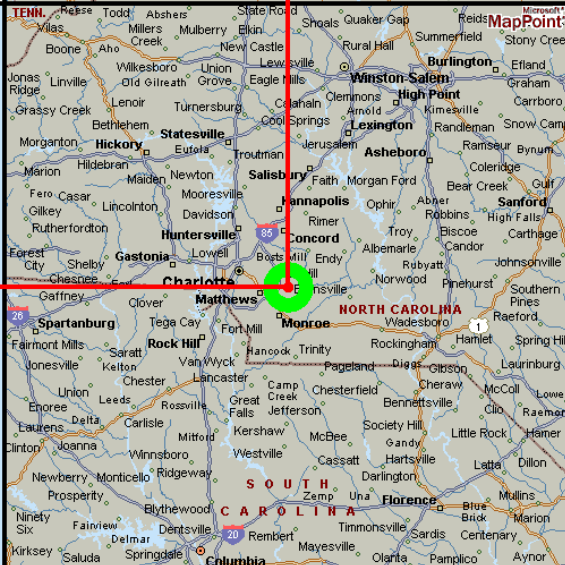


Figure 7: Example of query presented to judges.

any judging, each judge was presented with the following instructions:

Tweets Related to Local Event

Each task shows a group of tweets from the same general area on a map. Please read the tweets and tell us if at least three of them seem to be talking about the same local event.

A local event is an event that draws attention from people nearby. For instance, an earthquake is a local event, because it covers a limited area.

A presidential election is not a local event, because it covers such a wide area. Local events are often the types of stories you see on the local television news.

We are interested in any type of local event, but some example local events we want to find are sports, extreme weather, natural disasters, crimes, accidents, protests, gatherings, concerts, festivals, sports games, conventions, and conferences.

We are only interested in tweets that look like they came from someone who is actually experienced the event, either at the event's location or on TV, radio, or the Web.

You will likely see tweets related to a team sporting event, like football, basketball, baseball, soccer, etc. Commonly, the maps shows the location of the game, which is often the home field of one of the teams or else the home of the visiting team.

Even though the tweeters may be watching on TV, it should be considered a local event, because the event is being experienced by people in a limited region who are especially interested in the event.

The green circles on the maps show the location of the tweets at three different zoom levels. You *do not* have to tell us if the tweets came from the green circles.

The maps are there just for your curiosity to see the reported location of the event. You can ignore the maps if you like.

Human Judges on Local Events

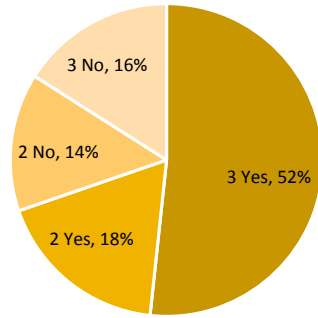
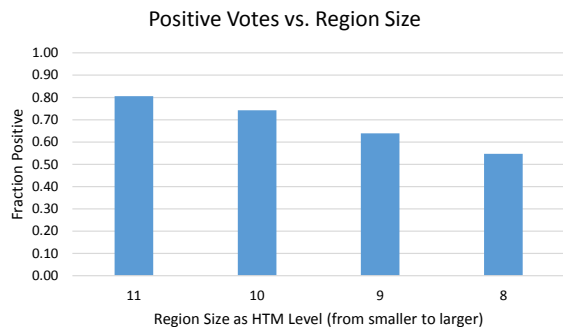


Figure 8: 70% of detected local events were judged as local events by a majority of our human judges.

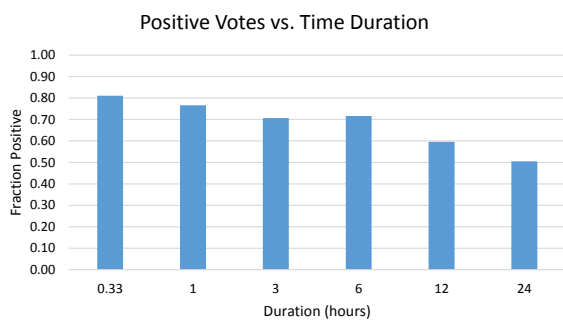
The judges were paid \$US 0.05 per answer, and we limited each judge to a maximum of 200 event judgments. Each candidate event was judged by three different judges.

Figure 1 is an example of tweets from an event that was unanimously judged as a local event.

Out of 2400 candidate events that were judged, 2339 received a majority of “Yes” or “No” votes. The remainder were less determinant with the addition of “Unsure” votes. Of these 2339, about 70% had a majority of votes confirming it as a local event,



(a) Positive votes vs. triangle size



(b) Positive votes vs. time duration

Figure 9: Rise in number of positive votes with smaller region sizes and shorter time intervals.

with the remainder rejected as not a local event with a majority “No” vote. The distribution of votes is shown in Figure 8.

We might expect that local events occur over relatively small extents in space and time. The votes from human judges give insight into which size space-time prisms are more likely to host a local event. Figure 9 plots the number of “Yes” votes for each for each triangle size L and time duration ΔT that our judges evaluated. Note that the triangle sizes and time durations were distributed uniformly over the 2400 test events, so each was fairly represented in the judged candidates. We see from the plots that smaller spatial regions and shorter temporal extents tend to account for relatively more local events.

A precision level of 70% is likely acceptable for many applications. However, this figure can be improved with a machine learning classifier, which we describe next.

7. LEARNING TO DETECT EVENTS

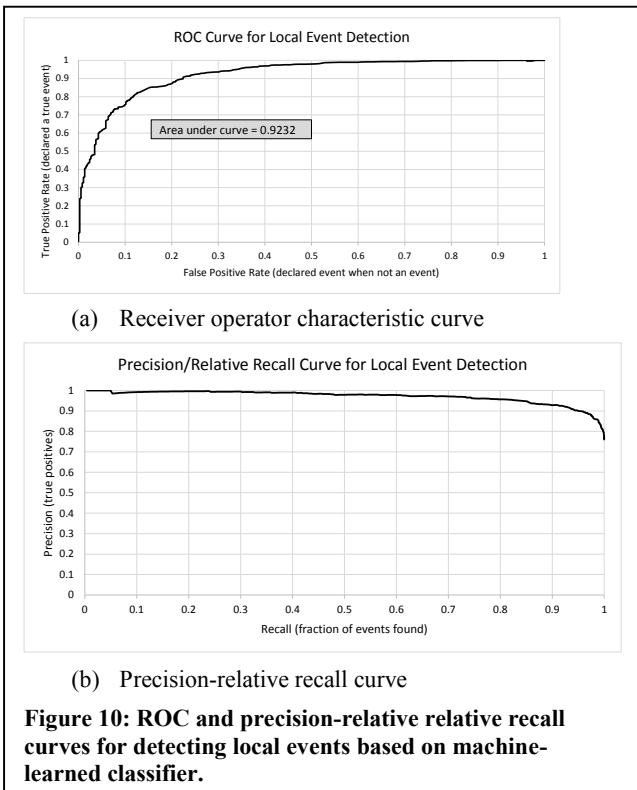
Our local event detector works by finding space-time prisms where the number of actual tweets exceeds the number of predicted tweets by three times the standard deviation of the prediction error. This straightforward approach achieves 70% precision as explained above. However, the results of human judging give a ground truth set of positive and negative instances of local events that we can use for training and testing a local event classifier. Specifically, we can compute a feature vector for each candidate local event and learn a binary classifier that estimates the probability of the candidate actually being a local event. This has the potential of increasing precision beyond the 70% achieved with the time series part of the algorithm.

From the human judgments in Section 6, we have 1209 events that were unanimously judged as a local event by the human judges (*i.e.* three “Yes” votes) and another 374 with a unanimous “No” vote. From these we trained and tested a binary classifier. As with our regression function, we again used a FastRank classifier that results in an ensemble of decision trees, this time with the aim of producing a probability indicating the likelihood that a candidate local event is actually a local event. The features that we used for the candidate event in the space-time prism (S, T) are:

1. Spatial size of space-time samples, from $L \in \{8,9,10,11\}$
2. Duration of space-time samples, from $\Delta T \in \{20 \text{ min}, 1 \text{ hr}, 3 \text{ hr}, 6 \text{ hr}, 12 \text{ hr}, 24 \text{ hr}\}$
3. Day of week, 0-6
4. Weekend or weekday, binary
5. Number of tweets in space-time prism, *i.e.* y_t
6. Tweet count prediction error, *i.e.* $e_t = \hat{y}_t - y_t$
7. Prediction error divided by number of tweets, *i.e.* e_t/y_t
8. Prediction error divided by standard deviation of error of regression function, *i.e.* e_t/s

Training and testing with 10-fold cross validation, we swept through different parameter settings of the decision tree ensemble. Optimizing for area under the ROC curve (AUC), the best parameters were a maximum number of leaves per tree of 32, minimum number of training instances in each leaf of 50, and 20 trees per ensemble. This led to an AUC of 0.92. The ROC curve is shown in Figure 10.

The classifier returns a class probability, and the final classification is based on thresholding this probability. This gives flexibility in adjusting the sensitivity of the classifier. A low threshold will lead to more local event detections. This corresponds to operating farther to the right on the ROC curve (more false positives and more true positives).



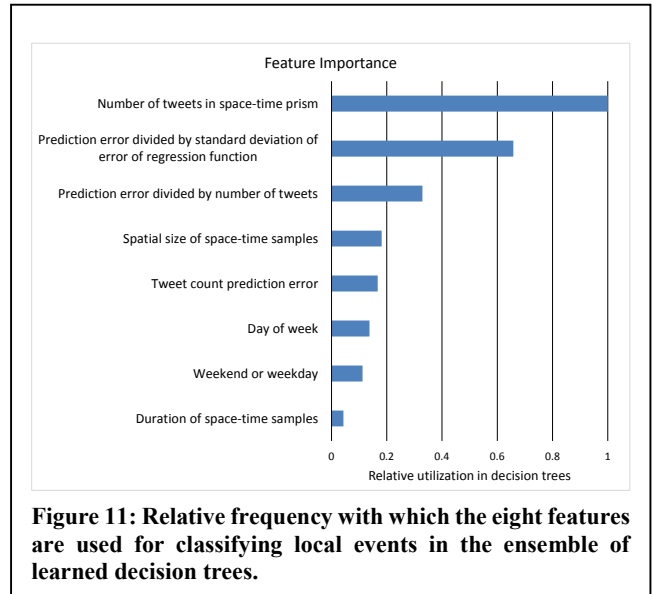
Another way to evaluate our algorithm is in terms of precision and recall. Precision is the same as the true positive rate. Recall measures the fraction of actual local events detected out of all local events. Clearly this is difficult to measure, for the list of actual local events is practically unknowable. This is because there is not an unambiguous definition of a local event, and because there is no ground truth repository of *all* local events against which to compare. Many local events go unreported in any formal way. (This is one reason that a system like Eyewitness is important: we can extract local events from Twitter that might otherwise be missed.)

Our universe of local events are those detected in the signal processing phase of our algorithm and then unanimously judged as a local event by our anonymous judges. Thus we cannot report recall in the traditional sense. Instead, our “recall” measures the ability of our algorithm to pick out those local events from a limited universe. To avoid confusion, we will use the term “relative recall” to convey this more limited measure of recall. Figure 10 shows the precision-relative recall curve. One feasible operating point gives a relative recall of 90% and a precision of 93%. Recalling that the signal processing phase of our algorithm achieved 70% precision, the use of a classifier in this phase raised the precision by 23 percentage points.

The relative importance of the eight classification features is shown in Figure 11. Importance is measured based on how many times the feature is used in the ensemble of learned decision trees. The three leading features are all related to the number of tweets in the space-time prism, led by the raw number of tweets (y_t) and then two ways of measuring the size of the anomaly: the time series prediction error normalized by the standard deviation of the time series prediction error (e_t/s) and the time series prediction error normalized by the number of tweets (e_t/y_t). The size of the triangles (L) is the fourth most important feature. The fifth most important is the raw prediction error (e_t), followed by the day of

the week, weekday vs. weekend, and finally the sampling time of the time series (ΔT).

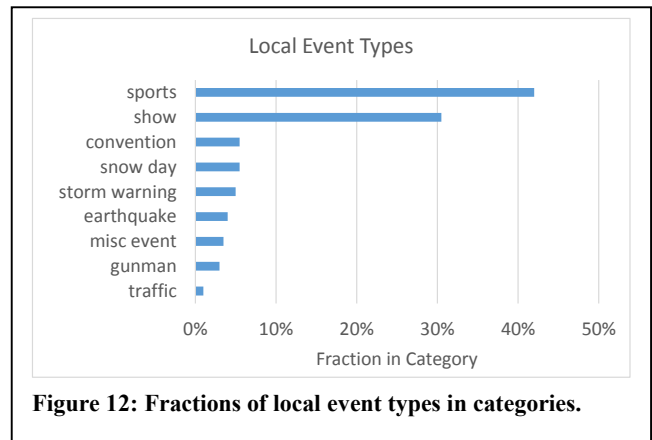
The classifier from this section is the second major stage of our algorithm, following the signal processing step described earlier. The signal processing step can be considered a filter to select only those space-time prisms that have a high likelihood of hosting a local event. However, the classifier stage works on features from any space-time prism, not just those that passed the first stage. This



means the classifier could serve as a stand-alone local event detector. In our experiment, however, the first stage was important to extract an adequate proportion of positive candidate events (70%) to give our human judges and subsequent classifier training enough positive samples.

8. LOCAL EVENT TYPES

A cursory look at the events detected by our system showed they fell into a relatively small number of well-defined categories. We employed a professional linguist to categorize 200 randomly chosen local events from the 2339 that were voted to be an actual local event by our judges. The linguist’s categories and fractions of events in each category are shown in Figure 12. One notable category is “gunman” at 3%, which came from armed criminals on college campuses.



9. CONCLUSIONS

We have presented our Eyewitness system for detecting local events from geotagged tweets. Although only a small fraction of tweets are geotagged, we showed that this fraction is adequate for detecting local events based only on finding spatially localized anomalies in the rate of geotagged tweets. The detection system does not require any analysis of the tweets' text. The algorithm was tested on time series of tweets from different spatial and temporal resolutions. A panel of human judges determined that 70% of 2400 detected events were local events. A decision tree classifier was able to boost the precision to 93% while maintaining a relative recall rate of 90%.

Eyewitness is a retrospective tool for finding local events. For future work, it should be feasible to make small changes to detect events in real time. With more work, it may be possible to detect local events even as they evolve based on a careful examination of the trajectory of tweet volumes in localized areas.

10. REFERENCES

- [1] J. Teevan, D. Ramage, and M. R. Morris, "#TwitterSearch: a comparison of microblog search and web search," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM 2011)*, 2011, pp. 35-44.
- [2] Twitter, "The 2014 #YearOnTwitter," in *The Official Twitter Blog* vol. 2014, ed. 2014.
- [3] C.-H. Lee, "Mining Spatio-Temporal Information on Microblogging Streams Using a Density-Based Online Clustering Method," *Expert Systems with Applications*, vol. 39, pp. 9623-9641, 2012.
- [4] C. A. Davis Jr, G. L. Pappa, D. R. R. de Oliveira, and F. de L. Arcanjo, "Inferring the Location of Twitter Messages based on User Relationships," *Transactions in GIS*, vol. 15, pp. 735-751, 2011.
- [5] K. Watanabe, M. Ochi, M. Okabe, and R. Onai, "Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs," in *20th ACM International Conference on Information and Knowledge Management (CIKM 2011)*, Glasgow, UK, 2011, pp. 2541-2544.
- [6] C.-H. Lee, H.-C. Yang, T.-F. Chien, and W.-S. Wen, "A Novel Approach for Event Detection by Mining Spatio-Temporal Information on Microblogs," in *International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2011)*, 2011, pp. 254-259.
- [7] A. Ritter, O. Etzioni, and S. Clark, "Open Domain Event Extraction from Twitter," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 1104-1112.
- [8] C.-H. Lee, T.-F. Chien, and H.-C. Yang, "An Automatic Topic Ranking Approach for Event Detection on Microblogging Messages," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, 2011, pp. 1358-1363.
- [9] A.-M. Popescu and M. Pennacchiotti, "Detecting Controversial Events from Twitter," in *19th ACM International Conference on Information and Knowledge Management (CIKM '10)*, 2010, pp. 1873-1876.
- [10] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, "Geographical Topic Discovery and Comparison," in *Proceedings of the 20th International World Wide Web Conference (WWW 2011)*, 2011, pp. 247-256.
- [11] H. Samet, J. Sankaranarayanan, M. D. Lieberman, M. D. Adelfio, B. C. Fruin, J. M. Lotkowsky, *et al.*, "Reading News with Maps by Exploiting Spatial Synonyms," *Communications of the ACM*, vol. 57, pp. 64-77, 2014.
- [12] X. Zhou and L. Chen, "Event Detection over Twitter Social Media Streams," *The VLDB Journal—The International Journal on Very Large Data Bases*, vol. 23, pp. 381-400, 2014.
- [13] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake Shakes Twitter Users: Real-Time Event Detection by Social Sensors," in *19th International Conference on World Wide Web (WWW '10)*, Raleigh, NC USA, 2010, pp. 851-860.
- [14] H. Abdelhaq, C. Sengstock, and M. Gertz, "EvenTweet: Online Localized Event Detection from Twitter," *Proceedings of the VLDB Endowment*, vol. 6, pp. 1326-1329 August 2013 2013.
- [15] Y. Hu, A. John, D. D. Seligmann, and F. Wang, "What Were the Tweets About? Topical Associations Between Public Events and Twitter Feeds," in *6th International Conference on Weblogs and Social Media (ICWSM-12)*, 2012.
- [16] A. S. Szalay, J. Gray, G. Fekete, P. Z. Kunszt, P. Kukul, and A. Thakar, "Indexing the Sphere with the Hierarchical Triangular Mesh," Microsoft Research, Redmond, WA USA, Technical Report MSR-TR-2005-123, August 2005 2005.
- [17] R. Lee and K. Sumiya, "Measuring Geographical Regularities of Crowd Behaviors for Twitter-Based Geo-Social Event Detection," in *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, 2010, pp. 1-10.
- [18] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol. 29, pp. 1189-1232, 2001.
- [19] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, *et al.*, "Spatiotemporal Social Media Analytics for Abnormal Event Detection and Examination Using Seasonal-Trend Decomposition," in *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, 2012, pp. 143-152.
- [20] C. Li, A. Sun, and A. Datta, "Twevent: Segment-Based Event Detection from Tweets," in *21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 155-164.
- [21] H. Yin, B. Cui, H. Lu, Y. Huang, and J. Yao, "A Unified Model for Stable and Temporal Topic Detection from Social Media Data," in *29th IEEE International Conference on Data Engineering (ICDE 2013)*, 2013, pp. 661-672.
- [22] Q. Diao, J. Jiang, F. Zhu, and E.-P. Lim, "Finding Bursty Topics from Microblogs," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 2012, pp. 536-544.
- [23] A. Guille and C. Favre, "Mention-anomaly-based Event Detection and tracking in Twitter," in *EEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, 2014, pp. 375-382.
- [24] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "Twitinfo: Aggregating and Visualizing Microblogs for Event Exploration," in *SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 227-236.
- [25] D. Inouye and J. K. Kalita, "Comparing Twitter Summarization Algorithms for Multiple Post Summaries," in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom)*, 2011, pp. 298-306.
- [26] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, "Beyond SumBasic: Task-Focused Summarization with Sentence Simplification and Lexical Expansion," *Information Processing & Management*, vol. 43, pp. 1606-1618, 2007.
- [27] M. Buhmester, T. Kwang, and S. D. Gosling, "Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data?," *Perspectives on psychological science*, vol. 6, pp. 3-5, 2011.