SPOTHOT: Scalable Detection of Geo-spatial Events in Large Textual Streams

Erich Schubert
Institut für Informatik
LMU Munich
Germany
schube@dbs.ifi.lmu.de

Michael Weiler
Institut für Informatik
LMU Munich
Germany
weiler@dbs.ifi.lmu.de

Hans-Peter Kriegel
Institut für Informatik
LMU Munich
Germany
kriegel@dbs.ifi.lmu.de

ABSTRACT

The analysis of social media data poses several challenges: first of all, the data sets are very large, secondly they change constantly, and third they are heterogeneous, consisting of text, images, geographic locations and social connections. In this article, we focus on detecting events consisting of text and location information, and introduce an analysis method that is scalable both with respect to volume and velocity. We also address the problems arising from differences in adoption of social media across cultures, languages, and countries in our event detection by efficient normalization.

We introduce an algorithm capable of processing vast amounts of data using a scalable online approach based on the SigniTrend event detection system, which is able to identify unusual geo-textual patterns in the data stream without requiring the user to specify any constraints in advance, such as hashtags to track: In contrast to earlier work, we are able to monitor every word at every location with just a fixed amount of memory, compare the values to statistics from earlier data and immediately report significant deviations with minimal delay. Thus, this algorithm is capable of reporting "Breaking News" in real-time.

Location is modeled using unsupervised geometric discretization and supervised administrative hierarchies, which permits detecting events at city, regional, and global levels at the same time. The usefulness of the approach is demonstrated using several real-world example use cases using Twitter data.

CCS Concepts

•**Information systems** → **Data streaming; Data stream mining;** *Summarization;* •**Theory of computation** → *Bloom filters and hashing;* •**Computing methodologies** → *Anomaly detection;*

Keywords

Local event detection; bursty topic detection; change detection; online control charts; time-series analysis; anomaly detection; trend detection; geo-social media; rich geo-spatial data; scalable real-time data analysis; streaming algorithm

SSDBM '16, July 18-20, 2016, Budapest, Hungary

DOI: http://dx.doi.org/10.1145/2949689.2949699

1. INTRODUCTION AND MOTIVATION

Social media such as Twitter produces a fast-flowing data stream with thousands of new documents every minute, containing only a short fragment of text (up to 140 characters), some annotated entities and links to other users as well as web sites, and sometimes location information on where the message originated from. Many traditional analysis methods do not work well on such data: the informal language with many special abbreviations, emoji icons and constantly changing portmanteau words prevents many advanced linguistic techniques from understanding the contents. Furthermore, many methods in topic modeling need to visit data multiple times to optimize the inferred structure, which is infeasible when thousands of new documents arrive every minute. Thus, algorithms for such data sets are usually designed around simple counting, or require the data set to be reduced via predefined keywords and location filters. Such algorithms often yield unsatisfactory results because the output is determined too much by quantity: the most frequent terms usually contain well-known information such as the popularity of Justin Bieber on Twitter, "good morning" in the morning and "dinner" at night, and the TV series currently on air. As a result, interesting events are hard to discover due to the sheer quantity of everyday chat. Furthermore, when users are not evenly distributed such "mainstream" behavior in large countries tends to obscure interesting developments in other parts of the world. Using absolute counting for event detection only works if the events have already been reported on TV and other mass media and we observe the global echo in social media. Instead, we need approaches that can identify significant patterns without reporting the obvious. Due to spam and noise identifying the interesting subset is hard.

When the attack on Charlie Hebdo occurred at around 11:30, it still took 45 minutes to yield a significant number of mentions of the hashtag *#CharlieHebdo*, all of them originating in Paris. By 13:00, the news then had spread to the U. K. and by 13:30 most European countries were discussing the attacks on social media. By 14:00, the new hashtag *#JeSuisCharlie* had emerged. These observations demonstrate how geographic locality may help gauge the importance —add interesting detail—to understanding an event. It also demonstrates the usefulness of a tool capable of analyzing such developments with little delay, without having to neither rerun the analysis, nor specify the topics of interest in advance. While *Charlie Hebdo* yields an obvious portmanteau keyword to use as hashtag on Twitter, this is not always the case, and we need to monitor words that have not been manually chosen as a keyword by the user.

Thus, we are interested in a system capable of tracking unusual occurrences of arbitrary words at arbitrary locations in real-time, without having to specify the terms of interest in advance. Absolute counts—and the absolute increase—need to be adjusted for differences in usage.

1.1 Problem Definition

Given a high-throughput textual data stream (too large to be effectively stored and queried) with geographic metadata, the goal is to identify terms that receive a unusual strong increase in their frequency as well as the locations and time associated with this unusual increase. The resulting events should not include spurious events that do not exhibit above characteristics, so that they can be used as unsupervised input for a "Breaking News" detection system. The user must not be required to specify a limited set of candidate terms or locations, but instead the system must be able to learn the relevant vocabulary and locations from the data.

1.2 Challenges

Velocity and Volume: When processing a Twitter feed, the obvious challenges are the velocity and volume of the data. The `statuses/sample` endpoint of Twitter is reported to include 1% of all tweets (4–5 million per day, along with 1–2 million deletions; about 15 GB of uncompressed raw data per day). Only the surprisingly small percentage of 1.5% of these tweets include geographic coordinates: Retweets never contain coordinates, and often users only check in with an approximate location such as a point of interest. We use a different feed which both has a larger volume and a higher rate of tweets with coordinates (at the cost of having no retweets), so that we have to process over 5 million geo-tagged tweets per day, between 3000 and 5000 tweets per minute. The number of tweets is not the sole parameter affecting scalability: the complexity of the analysis also influences the scalability a lot. Performing pairwise comparisons or computing a distance matrix is impossible if the data is too large to visit multiple times.

Spam: Twitter contains various kinds of spam in large quantities. There are advertisements for products, services, and Twitter followers, and some spammers try to manipulate the trending topics. But there are also teenagers sending mass love messages to their idols; bots announcing trending topics, earthquakes, and weather forecasts; and "follow sprees" where users mass-follow each other to raise their follow count to appear more popular. We found the analysis results to become much more interesting when ignoring tweets that either include "follow" or "trend". We utilize a basic near-duplicate filter to drop most spam and thus remove many false-positives. This filter is robust to randomly generated words often used to defeat Twitters built-in duplicate detector. After removing such garbage words we compare every tweet to the last 10,000 tweets and skip those that are duplicate. Due to space limitations, we do not provider further details in this article.

Handling of geography: Tweets may come with either coordinates, or points of intererest ("check ins"). Neither information is very trustworthy, and we do not have precision information on the location. Both spammers and bots may fake their location on Twitter. For example, we have observed one large spammer to use locations randomly sampled from $[0; 1]$, in the ocean south-west of Africa. Earthquake and weather bots also do not use their physical location for their tweets, but an estimate of the earthquake location or the city the weather forecast is produced for. For simplicity, we only use the tweets with coordinates, but not the points of interest. It is easy to remove the false locations at $[0; 1]$, but we do not remove the earthquake and weather bots.

Variation of Tweet Density: Twitter adoption varies heavily from country to country: Twitter is popular e.g. in the USA, Brazil, Argentina, Indonesia, and Turkey. On the other hand, users from China, India, and Germany (where apparently Twitter usage and location sharing raise privacy concerns) are underrepresented in this data set. Table 1 shows the distribution of countries and locations within our data set from September 10, 2014 to February 19, 2015. This demonstrates the need to look for relative increase of volume, and to use geographic location for normalization. If we want to obtain meaningful results for all areas of the world (including e.g. Berlin), we need to design an algorithm that is capable of adapting to local variations in the tweet density. Methods based solely on counting the most popular terms will be biased towards countries with many users and unable to detect events in less populated areas.

Table 1: Geographic distribution of twitter data.

region	mil.	share	region	mil.	share
u. s. a.	287.7	25.4%	russian fed.	18.2	1.6%
brazil	165.6	14.6%	mexico	17.6	1.6%
argentina	73.6	6.5%	florida	17.4	1.5%
indonesia	72.0	6.4%	new york	17.3	1.5%
turkey	59.3	5.2%	kanto (japan)	17.2	1.5%
japan	52.4	4.6%	west java (ind.)	16.9	1.5%
united kingdom	49.3	4.4%	saudi arabia	15.4	1.4%
são paulo	40.6	3.6%	colombia	14.1	1.2%
england	40.3	3.6%	selangor (mal.)	14.1	1.2%
california	38.6	3.4%	ohio	13.6	1.2%
rio de janeiro	35.9	3.2%	kinki (japan)	13.5	1.2%
spain	34.8	3.1%	santa catarina	13.1	1.2%
buenos aires	33.9	3.0%	los angeles	12.8	1.1%
philippines	32.1	2.8%	ile-de-france	12.5	1.1%
france	31.8	2.8%	minas gerais (br.)	12.3	1.1%
malaysia	31.3	2.8%	porto alegre (br.)	12.2	1.1%
texas	31.2	2.8%	manila (phil.)	11.9	1.1%
marmara region	30.1	2.7%	jakarta (indon.)	11.6	1.0%
istanbul	20.3	1.8%	pennsylvania	11.3	1.0%
rio grande do sul	20.0	1.8%	aegean region	10.6	0.9%
thailand	19.4	1.7%	italy	10.1	0.9%
selected further regions $< 1\%$:					
region	mil.	share	region	mil.	share
canada	9.4	0.83%	india	4.8	0.43%
portugal	9.3	0.83%	egypt	4.2	0.37%
uruguay	9.0	0.80%	australia	4.0	0.35%
london	7.6	0.67%	ukraine	3.8	0.33%
new york city	7.5	0.66%	germany	3.5	0.31%
tokyo	7.4	0.66%	nigeria	2.6	0.23%
chile	7.0	0.62%	pakistan	1.6	0.14%
scotland	5.5	0.48%	china	1.2	0.11%
the netherlands	4.8	0.43%	berlin	0.5	0.05%

Streaming Analysis: Designing an algorithm that processes every record only once and that has to perform under substantial memory and time constraints—a streaming or "on-line" algorithm—yields a different scalability challenge.

Many methods require the specification of a set of candidate words to track, or only track hashtags to reduce the data size by filtering. However, usage of Twitter changes over time, and hashtags only emerge after an event has occurred. Events are even more interesting if they are significant before people have agreed on which hashtag to use. At the same time, single words may be not unique enough to identify an event. If we want to produce the best possible results, we need to design a system capable of tracking all the words at the same time.

2. RELATED WORK

As the daily flood of information available in social media is continuously increasing and the commercial interest to obtain valuable information from this data grows, the field of emerging event detection gains more and more attention. For this purpose, both commercial systems and academic prototypes like the CMU system [29], the UMass system [4], Blogscope [6], Meme-Detection system [16], and SigniTrend [23] have been established. Application scenarios in literature include the detection of earthquake events [20] and the prediction of flu trends [14].

Approaches using text only: Most related work on event and emerging topic detection does not use geographic metadata. Kleinberg [13] uses an infinite state automaton to model frequency bursts of incoming documents such as emails. This approach has also been applied to blog posts [19] and to dynamic topic models [24]. TwitterMonitor [18] and EnBlogue [5] both use the increase in term frequencies to detect topics. SigniTrend [23] tracks the average and standard deviation of term and term pair frequencies and which is thus able to quantify the significance of an event.

Geographical Approaches: Twitter publishes a list of trending topics for a manually curated list of about 500 locations (with only includes a limited choice of cities in each country). The exact algorithm used is not published, but it does not include topics—like *#justinbieber*—which are always frequent.[1] We assume it is based on a relative increase in frequency at a predefined location, i.e. the geographical information is used as a filter to split the data, but not in the actual analysis. Sakaki et al. [20] monitor Twitter users to detect earthquake and typhoon events. However, this approach requires that the user specifies query terms to select the topic of interest. Crowd behavior is used to detect geo-social events in [15]. They partition the world into smaller regions of interest (RoI). For each RoI, they monitored the number of tweets, number of users and crowd movement. When unusual features are measured, tweets belonging to the corresponding RoI are reported as an emerging geo-social event. This approach does not track individual topics. Points of interest (POIs) are analyzed in [17] to identify expert users for predefined topics and POIs. Kim et al. [12] use predefined topics such as Weather, TV, and Sport, and build a state level correlation matrix for each. Wang et al. [26] identify local frequent keyword co-occurrence patterns, but do not determine emerging or trending ones. For a spatially limited subset it can still be feasible to count—for every word and every location—the number of users. Such approaches were used in [1, 2], but these approaches do not scale to large data sets. EvenTweet [2] first identifies keywords by exact counting and comparison to history information, then for every keyword analyzes the spatial distribution. For this, it needs to store large amounts of data and cannot operate on a live stream.

Our approach is motivated by GeoScope [8], which tries to solve a similar problem (detecting events in geo-tagged tweets), but our algorithm is built upon SigniTrend [23] to overcome the limitations of GeoScope. GeoScope also uses Count–min sketch data structures [10] for approximate counting, but they independently use such tables for identifying the most frequent locations and the most frequent hashtags (the most frequent words would be stop words, thus the need to constrain their approach to hashtags; which is not a problem for our significance-based approach). An event in GeoScope is required to both having a unusually frequent term at a single location, and an usually frequent single location for this term. While this works well for extreme events that are observed in a single location only, for example the Super Bowl is watched all over the world, and will thus not be recognized as event. When decreasing the thresholds enough to capture less frequent topics and locations, the number of reported results grows exponentially.

Our approach improves substantially over the capabilities of GeoScope, because it is based on a notion of significance instead of frequency. We can detect the most significant events in large-scale regions (e.g. SuperBowl on TV) as well as neighborhood resolution because we are not constrained to tracking the globally most popular words and locations only. Our approach also can distinguish between local keywords that are always frequent (such as `#nyc` in New York City), and local keywords that exhibit unusual activity.

[1]Source: https://support.twitter.com/entries/101125

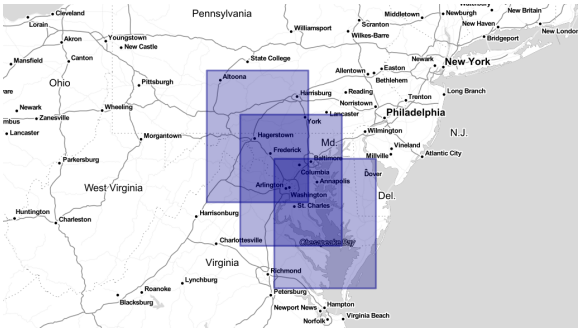


Figure 1: Grid tokens for a location in Washington, DC
Background © OpenStreetMap contributors, tiles by Stamen Design.

3. EVENT DETECTION APPROACH

In order to integrate textual data and geographic information, we map both into a sequence of tokens. Textual data is processed by Porter stemming, stop word removal, unification of emoji and hashtags, and a simple entity extraction approach trained on Wikipedia that recognizes common phrases and maps them to DBPedia concepts. For geographic information, we used a more complicated approach to avoid boundary effects, yet remain scalable and retain enough data to be able to achieve statistical significance. Figure 2 shows the tokenization process for an example tweet. After tokenization, documents are logically represented as a set of tokens that either correspond to a word token w or a location token l.

3.1 Symbolic Representation of Location

We employ two different methods in parallel to generate a symbolic representation of the tweet location. This dual strategy is necessary because of the different strengths: the grid-based approach covers the complete earth, with a uniform resolution and guaranteed coverage, whereas the OpenStreetMap approach provides administrative boundaries at different resolutions; but does for example not cover the oceans. Sometimes, administrative boundaries may cut areas of interest into two distinct regions, e.g. the "Twin Cities" Minneapolis and Saint Paul, or the Niagara Falls which are partially in Canada and partially in the United States.

Grid-based token generation: The first token generator is based on coordinate grids spaced at every even degree of latitude and longitude (i.e. using a grid width of 2°), which is an empirically chosen tradeoff between precision (the grid must not be too coarse) and abstraction: a too fine resolution reduces the chance of seeing a statistically significant number of mentions in the same location, while a too coarse resolution results in too many false positives. In order to avoid boundary effects (where the grid boundary would split a city into two separate grid cells), we use three overlapping grids, offset by one third and two thirds of the grid width. Chan [9] used the pigeonhole principle to prove that it is impossible to be closer than 1/6th of the grid width to all three grids at the same time. Because of this, points within $1/6 \cdot 2^\circ = 1/3^\circ$ degree must be in the same grid cell in at least one of the grids (for details, see Chan [9]). Points farther away than $\sqrt{2} \cdot 2^\circ \approx 2.82^\circ$ (the diagonal of a grid cell) are guaranteed to be in different grid cells in every grid. Due to the spherical nature of earth, the length of a degree varies: at the poles, $1^\circ \approx 110$km ≈ 69miles; at Oslo and Anchorage this reduces to approximately half (i.e. 55km ≈ 34miles). Further north, we do not see many tweets anyway. Tweets within about 18 km or 11 miles will thus produce the same symbol at least once; the expected distance for a grid cell collision is about 90 km or 54 miles, while events farther than 300 km or 200 miles have entirely different symbols. In-between of this range, we still have a high chance of at least one grid cell detecting the event.

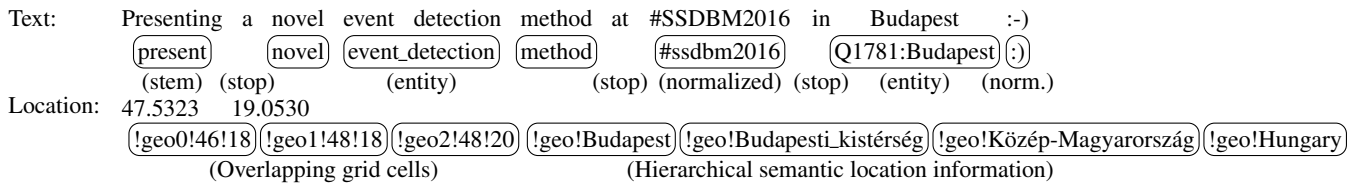


Figure 2: Tweet tokenization and generation of geo tokens of an example tweet

Figure 1 visualizes the geographic grid tokens produced for a location in Washington, DC. This approach is an improvement over both the Z-curve approach used by Wang et al. [26] and the grid-based approach of Abdelhaq et al. [1]. These approaches would cut e.g. Greenwhich into two halves. By using overlapping grids our approach does not suffer from such boundary effects (proven [9] by using the pigeonhole principle).

Our token generation process is designed for maximal performance, but alternate designs could of course provide higher precision if necessary. For each grid $i = 1, 2, 3$ we apply the function $f(l, o_i) = \lfloor l/2 + o_i \rfloor \cdot 2$ to each latitude and longitude, using the grid offsets $o_1 = 0$, $o_2 = 2/3$ and $o_3 = 4/3$. We then encode these tokens as "`!geo<i>!<lat>!<lon>`" (cf. Figure 2). These tokens cannot occur in regular text, and thus we can subsequently treat them as if they were regular words.

Tokens based on administrative boundaries: For aggregation at coarser resolutions, we employ a different strategy. We use a fast reverse geocoding library [22], for which we extracted polygons from OpenStreetMap and built a fast lookup table with 0.01 degree resolution, containing 64,597 regions while using just 30 MB of RAM, and allowing constant-time lookups at a rate of 1.5 million operations per second on a modern CPU. We reverse-geocode each coordinate to a hierarchy of regions, such as *Budapest* (City of Budapest), *Budapesti_kistérség* (Budapest subregion), *Közép-Magyarország* (Central Hungary), and *Hungary* as seen in Figure 2. We use each region returned by the lookup as a geographic token for our process; and this hierarchy allows us to detect events at different levels such as cities, counties, states and countries.

Again, we encode these tokens and add them to the tweets (e.g. "`!geo!Hungary`"). Subsequently we can treat them as regular words, except that we do not need to track pairs of these geo tokens.

3.2 Significance of Events

Given a word token w and a location token l—candidates will be obtained from a hash table detailed in Section 3.6—we use a classic model from statistics to measure the significance: Let $f_t(w, l)$ be the relative frequency of this pair of tokens within the documents $D_t = \{d_1, \ldots, d_n\}$ at time t, i.e.

$$f_t(w, l) := \frac{|\{w \in d \wedge l \in d \mid d \in D_t\}|}{|D_t|}$$

then we can use the series of previous values $f_1, \ldots, f_{t-1}$ to compute an estimated value and a standard deviation. To facilitate aging of the data and to avoid having to store all previous values, we employ the exponentially weighted moving average ($\mathit{EWMA}[f(w,l)]$) and moving standard deviation ($\mathit{EWMVar}[f(w,l)]$). With these estimates, we can compute the z-score of the frequency:

$$z_t(w, l) := \frac{f_t(w, l) - \mathit{EWMA}[f(w, l)]}{\sqrt{\mathit{EWMVar}[f(w, l)]}}$$

To avoid instability if a pair (w, l) was not seen often before, we employ a bias term β as introduced by SigniTrend [23].

$$z_t(w, l) := \frac{f_t(w, l) - \max\{\mathit{EWMA}[f(w, l)], \beta\}}{\sqrt{\mathit{EWMVar}[f(w, l)] + \beta}}$$

The term β is a Laplace-style smoothing term motivated by the assumption that there might have been $\beta \cdot |D|$ documents that contained the term, but which have not been observed. For Twitter, the suggested value for this term is $\beta = 10/|D|$: intuitively we consider 10 occurrences to be a by chance observation. This also adjusts for the fact that we do not have access to the full Twitter data.

The observed counts are standardized by subtracting the expected rate, and normalizing with the (exponentially weighted) standard deviation. The resulting "z-score" is a scale-free factor measuring how unusual the observed frequency is.

This normalization is one key improvement of our method over GeoScope. Because the pair (w, l) are the local occurrences with respect to l only, we are more robust against geographic differences. If the frequency increases drastically at a location l it can be detected as a significant event, even if the frequency is low compared to more popular locations. GeoScope would miss such events, because it only considers the most frequent locations. In particular, our approach can detect the same word w at different locations at the same time. We have observed such situations e.g. with TV events that are on-air in neighboring countries.

This statistic works at different granularities (e.g. detecting Super Bowl as trend in the whole U.S.A., but also very localized events such as an earthquake in Guam), and for highly populated areas as well as much less populated areas: in New York the expected rate, but also variance, will be much higher, than in Twitter-agnostic Germany. This is an important improvement, since cities like Istanbul, New York City, Tokyo and London have many more tweets than all of Germany, as seen in Table 1. An approach that does not take local tweet density into account would be unable to detect an event in a city in Germany, because these locations are never frequent with respect to the global Twitter volume.

3.3 Updating the Moving Averages

The statistics we use for normalization, *EWMA* and *EWMVar*, are a weighted mean and variance. In contrast to their traditional counterparts, we need to update them incrementally, based on the previous statistic and the new observations only: we cannot afford to store all previous values or revisit the data stream.

Welford [27], West [28] and Finch [11] provide update equations for these statistics that suit our needs very well:

$$\begin{aligned}
\Delta &\leftarrow f_t(w, l) - \mathit{EWMA}[f(w, l)] \\
\mathit{EWMA}[f(w, l)] &\leftarrow \mathit{EWMA}[f(w, l)] + \alpha \cdot \Delta \\
\mathit{EWMVar}[f(w, l)] &\leftarrow (1 - \alpha) \cdot (\mathit{EWMVar}[f(w, l)] + \alpha \cdot \Delta^2)
\end{aligned}$$

These (numerically stable) equations only rely on the new frequency $f_t(w, l)$, the old estimates *EWMA* and *EWMVar*, and an aging factor α which can easily be derived from the half-life time $t_{1/2}$—the time after which the influence of the data has reduced to half of the initial influence—as $\alpha = 1 - \exp\left(\log\left(\frac{1}{2}\right)/t_{1/2}\right)$.

The main requirement to use these equations is a good estimate of $f_t(w, l)$, and we need to improve scalability because we cannot afford to store these estimates for every pair (w, l).

3.4 Event Detection Algorithm

To obtain a smooth estimate of our frequency $f_t(w, l)$, we need to aggregate the data over large enough time windows to not exhibit random fluctuations, which would cause false alerts and a too high variance. Experimentally 15–60 minutes work well on Twitter, containing about 100,000 usable (non-spam, non-duplicate, geo-tagged) documents. The statistics are slow-moving averages, therefore this update rate is sufficient as long as we can *detect* deviations earlier: A delay of one hour is not acceptable to detect "Breaking News". Therefore, we split the process into two parts: a batch interval (1 minute) for event detection and the "epoch" interval (15–60 minutes) for statistics updates. We will give details on the detection process in Section 3.6.

In the batch interval of 1 minute tweets are tokenized, stemmed, spam and duplicates are removed. We can afford to count the frequencies of all pairs in this batch due to its small size. But computing our statistics for every pair (w, l) is infeasible due to the large number of different words and locations occurring in a fast-flowing social media stream. SigniTrend provides an approach derived from count-min sketches to efficiently maintain these statistics for huge data sets via hashing, which we adopt for our needs.

The frequencies counted in the 1 minute window are then fed into a count-min *aggregation sketch*, and at the end of the (15–60 minute) epoch, the *statistics sketch* with the moving averages is updated from the frequencies in this sketch, and we can start a new aggregation sketch for the next epoch. The old sketch is now used as *detection sketch* during the next epoch. This sketch is also updated from the 1 minute window counts; but because it is pre-filled with the counts of the previous epoch, we obtain a less noisy estimate of the average frequency: it contains both the new data and the data of the previous epoch. This is important in particular during the first minutes of each epoch: the detection sketch provides reliable estimates for detecting a sudden increase in frequency; whereas the new aggregation sketch is still too noisy and unreliable to be used. This way we can reduce the amount of false positive alerts substantially. At the end of the next epoch, this sketch is discarded and replaced with the next aggregation sketch. We do not use the detection sketch to update the statistics, because it contains redundant data. Because we only update the statistics at the end of each epoch, the aggregation sketch is better suited for this purpose. In order to detect events, we compare the values of the detection sketch with the estimates obtained from the statistics sketch. In order to filter repeated alerts, we use another *threshold sketch* that simply stores the last value that triggered an alert, and which slowly expires every epoch. The *last-seen sketch* simply stores the last pair (w, l) that was counted in each bucket, to be able to translate bucket numbers into understandable pairs.

At the end of each epoch, the statistics sketch is updated from the aggregation sketch. The observed absolute counts are normalized with the number of documents $|D|$ in the current epoch to obtain relative frequencies, and the moving average and variances are updated. Then, the aggregation sketch becomes the next detection sketch, and a new aggregation sketch is started. In the threshold sketch, the last reported values ν_i are decreased to slowly forget earlier trends. These operations are designed so that they can be vectorized to efficiently update the statistics table, as seen in Figure 3c. The overall process is summarized in Algorithm 1.

This procedure can be implemented in a watermarking stream system such as Google's Millwheel system [3]. Data is processed and added to the counting and alerting sketches as it arrives (because within each Epoch we do not rely on the data order), but statistics updates are delayed until the watermark guarantees the data of the Epoch is complete. If we switch from a time-based to a count-based epoch duration (e.g. every epoch has 100,000 tweets) we may even want to not use watermarking at all and ingest data as it arrives, not as it was generated.

Algorithm 1: Event detection algorithm

```
aggregation sketch ← new count-min sketch
detection sketch ← new count-min sketch
last-seen sketch ← new last-seen-word sketch
threshold sketch ← new threshold sketch
statistics sketch ← new SigniTrend sketch
while new batch D do
    foreach document d in batch D do
        Tokenize text of document d
        Add geo-tokens for document d
        foreach word w or pair (w, l) do
            Compute h hash codes
            Update aggregation sketch
            Update detection sketch
            if detection sketch was modified then
                Update last-seen sketch
    Compare detection sketch and threshold table
    Report detected events
    if end of epoch interval then
        Update statistics sketch using aggregation sketch
        detection sketch ← aggregation sketch
        aggregation sketch ← new count-min sketch
        Expire old events from threshold sketch
```

3.5 Sketch Data Structures

The sketches (data structures to approximate the data with less memory) used in SPOTHOT are inspired by bloom filters [7], count-min sketches and similar heavy-hitters algorithms. Each sketch consists of a table of size 2^b, where $b \geq 20$ produced excellent results as shown in Section 4. Tables with other sizes could be used, but powers of two can be more efficiently handled with bit mask operations. All sketches are indexed using the same hash functions $H_1(x) \ldots H_h(x)$ and thus must have the same size. The number of hash functions h is a small integer, and the value $h{=}3$ was used throughout our experiments.

The following sketches and update procedures are used:

1. The *aggregation sketch* and *detection sketch*: A simplified count-min [10] sketch (using a single table, instead of a separate table for each hash function).

   To update, read buckets $H_1(x) \ldots H_h(x)$, compute the minimum, then increment buckets $H_1(x) \ldots H_h(x)$ only if their current value was the minimum. (cf. Figure 3a)

2. The *last-seen sketch* stores the latest word-location pair, word or word pair seen in every bucket.

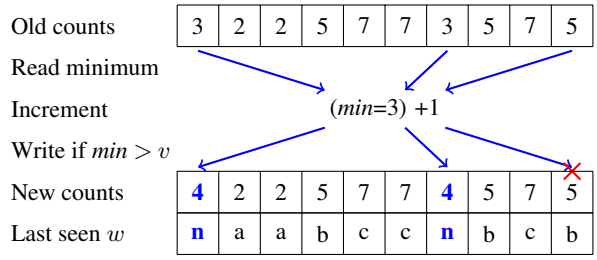
   If we increment the counter in the *detection sketch* sketch, we also update the entry attached to the bucket.

3. The *statistics sketch* is a SigniTrend sketch, which stores the exponentially weighted moving average and variance.

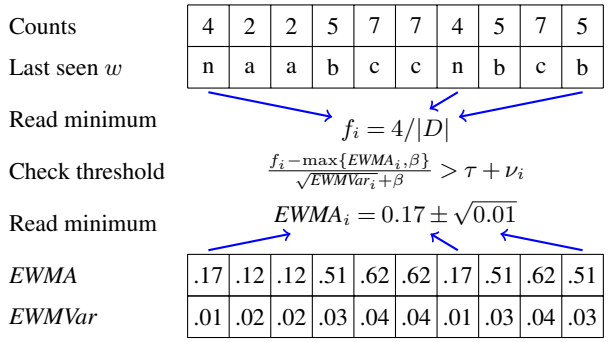
   At the end of every epoch, this sketch is updated using the update equations for moving averages (*EWMA*) and variances (*EWMVar*) discussed in Section 3.3 (cf. Figure 3c).

4. The *threshold sketch* caches the standard deviation and last reported significance.

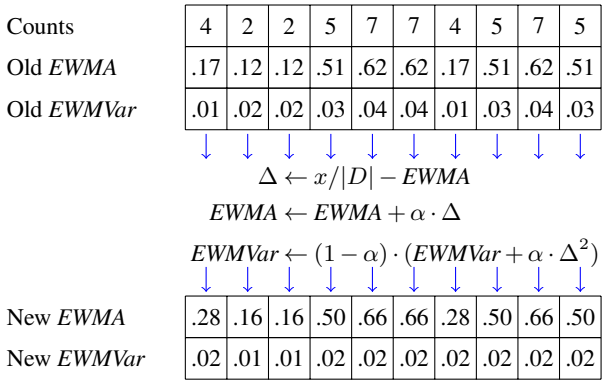
   At the end of each epoch, the last reported significance is heuristically reduced by multiplication with 0.75, to allow events to recur after a pause. The standard deviation is obtained by computing the square root of the variance stored in the SigniTrend sketch, to avoid repeated calculations.



(a) Count-min sketch update (new token: n)



(b) Check thresholds for new events



(c) Vectorized statistics table update

Figure 3: Hash table maintenance

These sketches have the following interesting properties:

1. Operations on the sketches are either confined to the affected hash buckets $H_1(x) \dots H_h(x)$, or affect all buckets the same way, and can be efficiently executed in parallel with SIMD instructions.

2. The sketches could be distributed to multiple machines by horizontal partitioning, using the first bits of every bucket as partition key. Aggregation sketches could also be partitioned vertically, but then we would need to frequenly aggregate these partitions. Because a single machine was already able to process all data, we have not further explored options for a parallel implementation.

3. The sketches used are lossy in the same way that a count-min sketch can overestimate the count of an object: the majority of word pairs in Twitter are unique combinations that will never constitute a significant event. Collisions can occur, but the more frequent entry is expected to win. So unless we have h collisions, each with a more frequent entry, we will not lose an event.

3.6 Detection of Events

Because computing the statistics for every pair of tokens would quickly exhaust memory resources, we employ heavy-hitters style sketches as introduced by SigniTrend [23]. This way, we are capable of monitoring all words and word pairs in a data stream by probabilistic counting with constant memory. But this yields a new challenge: we need to be able to infer the actual event detected from the hash tables. For this purpose we introduce the last-seen sketch into the model, and perform event detection as follows: At the end of every batch (i.e. in 1 minute intervals) we have to compute $H_i(x)$ for every observed combination $x = (w, l)$ to update the sketches. When updating the detection sketch, we get an estimated count, which we can normalize with the number of documents added to the sketch to get an estimated frequency $f(x)$. From the statistics sketch we can also read the corresponding estimates of $EWMA[x]$ and $EWMVar[x]$, such that we can apply our significance model explained in Section 3.2 to this combination. If the resulting significance exceeds the last reported value (stored in the threshold sketch, to filter repeated alerts) by at least the threshold τ, then this combination is reported as an event, and the threshold sketch is updated with its significance to temporary silence repeated notifications.

At this point in our analysis pipeline, we can only detect single word-location combinations (w, l). We also incorporate SigniTrend [23] to additionally obtain single-word and word-pair trends, then cluster the results as explained in Section 3.7 to obtain more complex event descriptions.

The hashing strategy ensures that we count frequent combinations exactly (with high probability), and that we never underestimate the frequency of *rare* combinations. Because rare combinations do not constitute events in our model, hash collisions do not effect event detection if the hash tables are large enough. To further reduce the chance of missing an event, we use multiple hash functions. Only if every hash function has a collision with a more popular combination, then we may miss a true event. It was demonstrated [23] that hash tables with 20 bit are sufficient to not lose important information, and our experiments in Section 4 indicate this also holds for our extension despite tracking additional pairs. Hash partitions can easily be parallelized, which allows this process to be easily distributed onto multiple hosts if necessary.

In addition to the tables for counting, moving average, and moving standard deviation employed by SigniTrend, we also maintain a threshold table which contains the alerting thresholds and the last reported values ν. The 1-minute batch interval is used for counting, and the resulting frequencies (aggregated over the active epoch) are compared to these thresholds. If the observed count exceeds the previously reported value ν by a significance threshold τ, the event will immediately be reported. Every time it exceeds the previously reported value by another τ standard deviations, it will be reported again. At the end of each epoch, the moving average table is updated—the longer update interval yields more reliable frequency estimates—and the last reported value ν is also exponentially decreased, to allow recurrent events.

By using moving averages, we also do not need to remove individual tweets from our statistics but the data "disappears" due to exponential weighting. Other approaches such as GeoScope [8] have to maintain a buffer storing the most recent tweets to be able to remove them from their counts when they have expired, and thus require much more memory and additional processing to clean up their data structures. In our statistical approach, data expiry is solely managed by the exponential weighting of the averages. as introduced by SigniTrend [23].

Table 2: Recall and precision compared to exact counting.

Table Bits	Recall	Precision	F_1-Measure	$\sigma \geq 20$
12	0.52%	100.00%	1.04%	5
13	4.47%	89.36%	8.51%	47
14	33.56%	89.64%	48.84%	338
15	66.96%	93.42%	78.01%	1717
16	87.52%	96.03%	91.57%	4604
17	97.54%	99.94%	98.16%	7493
18	99.16%	100.00%	99.54%	9330
19	99.22%	100.00%	99.61%	9791
20	99.23%	100.00%	99.61%	9929

3.7 Clustering

Reporting trends to users is an important step that needs a carefully chosen balance between fast notifications (a single term that is currently exceeding its alerting threshold) and meaningful aggregation into clusters containing multiple related terms. Clustering trending terms will provide topics and help the user to better understand the corresponding story. For our experiments, we choose an hierarchical agglomerative clustering as we do not know the number of trending clusters (stories) in advance, but can specify a qualitative similarity threshold. We chose average linkage as our linkage criteria to construct the cluster dendrogram, because the data may be dirty, and individual terms may be correlated to multiple clusters (albeit our clustering can only assign them to one). Average linkage was returning the empirically most meaningful results.

Because we also track every word-pair (like SigniTrend does) in addition to all word-geo pairs, we can estimate the frequencies of any word combination at any given time from our counting table. We can also include words that cooccur with the recent event, but are still slightly below the regular trigger threshold τ, or that have triggered early and are currently active, albeit not novel events themselves. We first collect all active words that participate in a current or recent event. For agglomerative clustering, we use a simple combination of coocurrence frequency $f_t(w_i, w_j)$ and trend significance $z_t(w_i, w_j)$. We then build a similarity matrix using a modified Jaccard similarity J' based on the pair frequencies, but use similarity 1 if the word pair is currently significant together:

$$J'(w_i, w_j) = \begin{cases} 1 & \text{if } z_t(w_i, w_j) \geq \tau \\ \frac{f_t(w_i, w_j)}{f_t(w_i) + f_t(w_j) - f_t(w_i, w_j)} & \text{otherwise} \end{cases}$$

Thus, we amplify words that did trend together by setting their similarity to the maximum of 1. We then run hierarchical clustering with average linkage, and cut the dendrogram at height 0.5. The resulting clusters resemble topics as found by typical topic models: At the day of the Scotland's independence referendum voting the terms *poll*, *England*, *referendum*, *result* and *people* formed a single cluster. More example cluster can be found in Table 4.

4. EXPERIMENTS

To evaluate the approximation quality of our algorithm, we compare to an implementation that uses expensive exact counting on all pairs (old pairs are forgotten if they have not been observed for several hours to reduce memory consumption, but we need 16 GB of RAM nevertheless to be able to keep all candidates in memory). In Table 2 we give recall and precision comparing our approach to this ground truth, using all events with over $\sigma \geq 20$, and consider a match positive if the event is also found by the other method with at least $\sigma \geq 10$ and with at most 1 hour difference. Recall measures how many events found by exact counting are also found by the approximate methods, precision measures how many of the events found using the approximate method can be confirmed using exact counting. Figure 4 visualizes these results. Precision remains high (i.e. few incorrect events are reported), but only very few events are detected and the recall is thus very low. The ongoing increase of $\sigma \geq 20$ events are mostly redundant reportings (the same event reported multiple times with increasing σ). Similar to the experiments using synthetic data in SigniTrend [23], we can observe a saturation effect on Twitter data at a table size of 18 bits even with the additional tokens we added for geographic information.

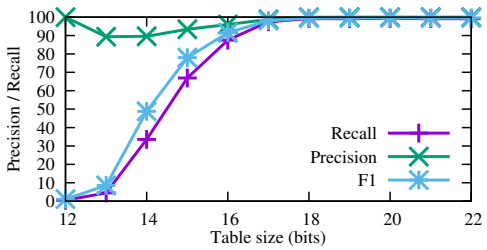


Figure 4: Recall and precision compared to exact counting.

The data feed we use contains 5–6 million tweets per day (slightly more than the popular `statuses/sample` endpoint; no retweets, and all tweets are geo-tagged). The data period is September 10, 2014 to February 19, 2015. Due to the Twitter terms of service, we are not allowed to make this data set available for download. Our implementation is using Java, and we used a desktop PC with an Intel Core i7-3770 CPU running Debian Linux and OpenJDK 1.8 with the fastutil and ELKI [21] libraries. Large parts of the implementation were done using low-level data structures to achieve high throughput.

Hash based counting is easy to distribute if additional scalability is needed, because it is embarrassingly parallel. The sketch tables can be split into multiple slices by a prefix of the hash key (i.e. horizontal partitioning), and distributed onto multiple computers. Furthermore, incoming Tweets can also be processed by multiple nodes in parallel (vertical partitioning), if the tables are then aggregated for example after each batch (this corresponds to a local aggregation as often done using a "combiner" in MapReduce). We did not use a distributed implementation, because we could already run the full analysis of one hour of data in 9 seconds on a single core. The overall lag from the live Twitter feed to the final output is constantly less than 2 seconds, and we could process 400× as much data without upgrading the CPU, adding additional cores, or having the overhead of a distributed implementation. We estimate that the 5–6 million geotagged Tweets we are receiving are up to 1/3 of the total geo-tagged tweets, and thus our method should be able to process all of Twitter on a single node easily.

4.1 Most Significant Regional Events

In the first experiment, we inspect the most significant local events detected in our data set, as this shows both the ability to detect trends as well as the ability to locate them geographically. For presentation, we selected the most significant occurrence of each keyword only, and of each locations only the most significant keyword. The top 20 most significant events selected this way are shown in Table 3. There are four events that we attribute to a highly active Twitter spammer in Turkey, but all others can be attributed to major media events or celebrities. Most of the events in Table 3 are detected at country level, which makes sense as they are based on events in TV, and did not happen at the users location. We observe that Twitter users mostly comment on what they see on TV and on the Internet, and much less on the physical world around them.

Table 3: Most significant events in their most significant location each

σ	Time	Word	Location	Explanation
2001.8	2014-10-29 00:59	#voteluantvz	Brazil	Brazilian Music Award 2014
727.8	2014-09-23 02:21	allahımsenbüyüksün	Denizli (Turkey)	Portmanteau used in spam wave
550.1	2015-02-02 01:32	Missy_Elliott	United States of America	Super Bowl Halftime Show
413.5	2014-09-18 21:29	#gala1gh15	Spain	Spanish Big Brother Launch
412.2	2014-11-11 19:29	#murrayftw	Italy	Teen idol triggered follow spree
293.8	2014-10-21 12:05	#tarıkgüneşttyapıyor	Marmara Region	Hashtag used in spam wave
271.2	2015-02-02 02:28	#masterchefgranfinal	Chile	MasterChef Chile final
268.1	2015-01-30 19:28	#سپارکیز	Saudi Arabia	Amusement park “Sparky’s”
257.7	2014-11-16 21:44	gemma	United Kingdom	Gemma Collins at jungle camp opening
249.1	2014-10-08 02:56	rosmeri	Argentina	Rosmery González joined Bailando 2014
223.1	2015-01-21 18:51	otortfv	Central Anatolia Region	Keyword used in spam wave
212.7	2014-09-11 18:58	#catalansvote9n	Catalonia	Catalan referendum requests
208.4	2014-12-02 20:00	#cengizhangençtürk	Northern Borders Region	Hashtag used in spam wave
205.3	2015-01-04 15:56	hairul	Malaysia	Hairul Azreen, Fear Factor Malaysia
198.7	2014-12-31 15:49	あけましておめでとうございます	Japan	New Year in Japan
198.5	2015-01-10 20:19	вк	Russian Federation	“Russian Facebook” VK unavailable
179.7	2014-10-04 16:28	#hormonestheseries2	Thailand	Hormones: The Series Season 2
174.7	2014-11-28 21:29	chespirito	Mexico	Comedian “Chespirito” died
160.9	2014-09-21 21:27	#ss5	Portugal	Secret Story 5 Portugal launch
157.3	2014-09-24 01:57	maluma	Colombia	Maluma on The Voice Kids Colombia

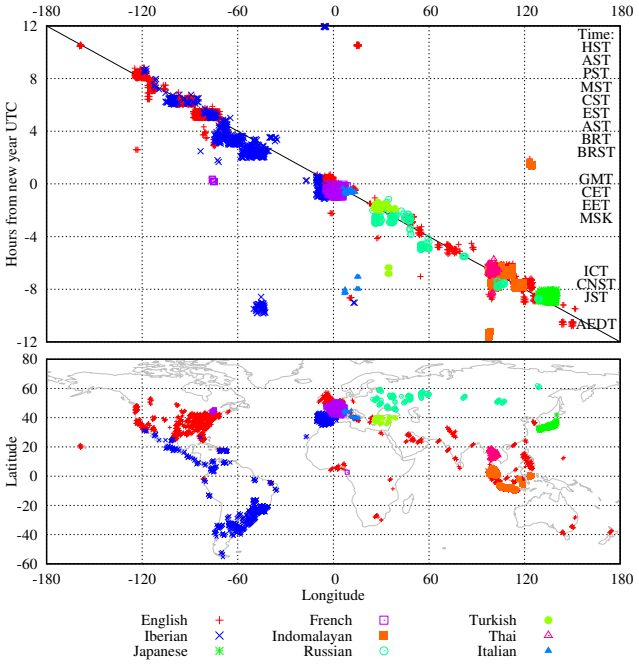


Figure 5: New Year around the world at $\sigma \geq 3$

4.2 New Year’s Eve

We analyzed trending topics on New Year’s Eve, and were able to identify New Year greetings in several languages. Interesting patterns emerge if we plot longitude versus time, as seen in Figure 5. In the first Figure, the x-axis is geographic, but the y-axis is temporal, so we can see the arrival of the New Year in different time zones. All events with $\sigma \geq 3$ were included, if we were able to identify them as New Year related. To remove visual clutter, we did not include other events, or festive emoji. Several cultural patterns arise in this Figure: Chinese New Year and Hindu New Year are on a different date, and thus neither China nor India does not show a lot of activity. Some vocabulary such as “Silvester” refers to the whole day before New Year. Italians started tweeting “capodanno” in the afternoon, but wish “buon anno” after midnight. Despite the fact that Twitter is not used much in Germany, our variance-based approach however can account for this, and was still able to detect some German New Year’s wishes, but the English greetings were more popular in Germany on Twitter. Sydney has the first fireworks at 9pm already (for the kids), and we can indeed observe an event “fireworks” in Australia at 10:04 UTC. Throughout the evening, we see New Year mentions, and the event at midnight is rather small compared to other countries. In Québec, we observe more French wishes around GMT than at midnight, but closer inspection revealed that most originate from the same user. In Russia, we can see Yakutsk, Ulan-Ude and Irkutsk, and Novosibirsk celebrate in different time zones prior to the more densely populated areas beginning with Yekaterinburg.

4.3 WikiTimes Events

We use the WikiTimes data set [25] to validate events found by our detection system. Table 4 lists events validated by comparing the resulting word cluster (using hierarchical clustering) with Wikipedia headlines for the same day. Our algorithm was able to identify several important events of contemporary history, and was able to produce both meaningful geographical information as well as related keywords. Only one of these events was detected with a hashtag, indicating that restricting the input data to hashtags—as required for GeoScope—may obscure important events. Named entity recognition however helped in detection, as it can be used to normalize different spellings and abbreviations of names. Due to the higher frequency of the disambiguated terms, we have a more stable signal and thus less variance and earlier detection.

4.4 Earthquakes

Natural disasters, such as earthquakes, are of broad interest to the public. It has been proposed to detect them using Twitter [20], thus we evaluate this scenario despite the fact that earthquakes are better detected using seismic sensors, and the earliest tweets for most quakes are automatic tweets by bots. The relative frequency of the term “earthquake” regardless of its geographical origin is shown in Figure 6a. This yields a noisy signal with a high background activity and many low-significance events. The frequency shown is normalized by the total number of tweets at each corresponding day to avoid seasonal patterns (e.g. more tweets on weekends than on workdays). If we narrow down the scope to a specific ge-

Table 4: Events validated via WikiTimes [25]

Date	σ	Event Terms	Event description © Wikipedia, The Free Encyclopedia
09-17	7.5	Scotland, U_K, poll, England, referendum, result, people	Voters in Scotland go to the polls to vote on the referendum on independence.
09-17	5.6	house, rebel, arm, Syria, approv	The United States Senate passes a budget measure authorizing President Barack Obama to equip and train moderate rebels to fight ISIL in Syria.
09-18	12.4	England, Scotland, referendum, U_K, Greater_London, London	Voter turnout in the referendum hits 84.5%, a record high for any election held in the United Kingdom since the introduction of universal suffrage.
09-18	25.6	Scotland, U_K, uk, England, Greater_London, London, David_Cameron	Prime Minister David Cameron announces plans to devolve further powers to Scotland, as well as the UK's other constituent countries.
09-18	15.0	England, referendum, Greater_London, U_K, Alex_Salmond, Scotland, resign, London, salmond, Glasgow_City	Alex Salmond announces his resignation as First Minister of Scotland and leader of the Scottish National Party following the referendum.
09-18	9.5	Philippines, cancel, flood, Metro_Manila, UK	Manila is inundated by massive flooding causing flights to the international airport to be cancelled and businesses to shut down.
09-22	40.1	Isis, U_S_A, Syria, airstrikes, bomb, target, islamic_state, u_s, strike, air	The United States and its allies commence air strikes against Islamic State in Syria with reports of at least 120 deaths.
09-23	17.7	Syria, strike, air, Isis	The al-Nusra Front claims its leader Abu Yousef al-Turki was killed in air strikes.
09-24	3.1	Algeria, french, behead	Algerian jihadist group Jund al-Khilafah release a video showing French tourist Hervé Gourdel being killed.
09-26	3.5	Iraq, Isis, air, strike	The Parliament of the United Kingdom approves air strikes against ISIS in Iraq by 524 votes to 43.
10-08	60.5	di, patient, thoma, duncan, eric, dallas, hospital, diagnos, texas	The first person who was diagnosed with Ebola in the United States, Thomas Eric Duncan, a Liberian man, dies in Dallas, Texas.
10-10	44.7	kailash, satyarthi, India, Nobel_Peace_Prize, malala, Malala_Yousafzai, congratul, #nobelpeaceprize, indian, pakistani, peace	Pakistani child education activist Malala Yousafzai and Indian children's rights advocate Kailash Satyarthi share the 2014 Nobel Peace Prize.
10-12	11.0	texas, worker, posit, tests, health_care, supplement	A Texas nurse tests positive for Ebola. The health care worker is the first person to contract the disease in the United States of America.
10-14	34.4	Republic_of_Ireland, ireland, U_K, Germany, England, John_O'Shea, Leinster, County_Dublin, Scotland	Ireland stuns world champion Germany in Gelsenkirchen, with Ireland drawing the match at 1–1 when John O'Shea scores in stoppage time.
10-14	30.6	Albania, Serbia, U_K, England, London, match, drone, flag	The game between Albania and Serbia is abandoned after a drone carrying a flag promoting the concept of Greater Albania descends onto the pitch in Belgrade, sparking riots, mass brawling and an explosion.
10-15	17.8	posit, worker, tests, Ebola_virus_disease, texas, health	A second health worker tests positive for the Ebola virus in Dallas, Texas.
10-17	13.1	czar, Barack_Obama, klain, ron, Ebola_virus_disease, U_S_A, Travis_County	Barack Obama names lawyer and former political operative Ron Klain as "ebola czar" to coordinate US response to the Ebola outbreak.
10-22	26.4	soldier, Canada, Ottawa, shoot, Ontario, insid, canada, parliament	A gunman shoots a Canadian Forces soldier outside the Canadian National War Memorial.

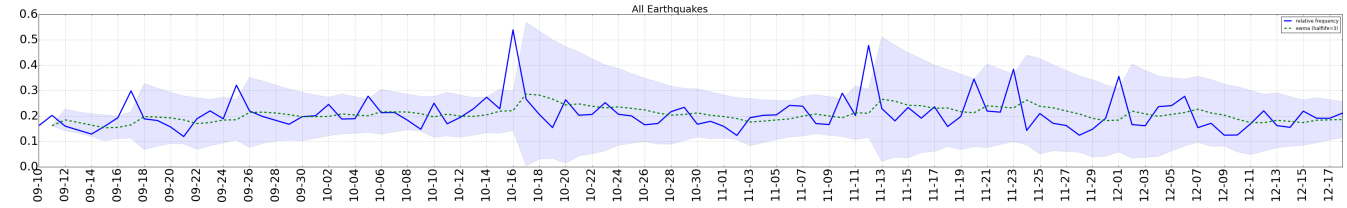
ographic region, we get a much cleaner signal. Figure 6b shows the frequency only near Guam within latitude 144 ± 1 and longitude 13 ± 1, where we can observe two events. To validate our observations, we can use metadata from the *United States Geological Survey's (USGS) Earthquake Hazards Program* as our validation data source for earthquake events. The first peak in Figure 6b at September 17, 2014 refers to a strong earthquake with a magnitude of 6.7 located 47km northwest of Hagatna (Guam), which was included in the USGS *Significant Earthquake Archive*.[2] The second, smaller peak on October 29, 2014 refers to a small earthquake 36km southeast of Hagatna with a lower magnitude of 4.7. In Figure 6c we plotted the frequency for earthquake mentions around Dallas, Texas within latitude -97 ± 1 and longitude 33 ± 1.

The total number of significant earthquakes included by the USGS in their list during the time of our Twitter crawl is 30 (ranging from the first earthquake reported on September 17, 2014 to February 13, 2015). By exact counting the tweets corresponding to these events, we found that 19 of them received less than 10 mentions because they happened far away from cities. Because such low frequencies are too little to be statistically significant, we excluded them from our evaluation. While detecting earthquakes is a popular use case for event detection on Twitter, it cannot be used as a replacement for seismic sensors: 19 out of 30 significant earthquakes were not significantly discussed on Twitter. The explanation is that most earthquakes happen off-shore, such as the last (Sacramento) earthquake in this list, which was 40 miles off-shore south-west of Eureka, CA. It was mostly reported by Twitter earthquake bots, and did not cause many additional Tweets. If a substantial earthquake happens, we must even assume the network to become unavailable.

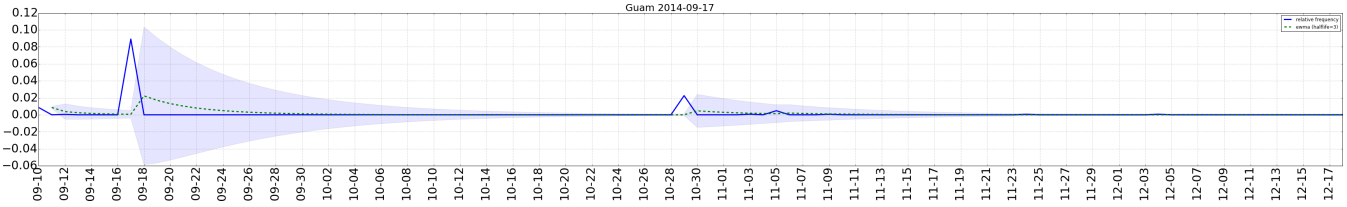
Many earthquakes would also have been detected by SigniTrend without our extension using geo locations, because people mention their location in the text: only the 1st (Oklahoma) and 9th (Texas) earthquake would have been missed otherwise. By including the geo information and tracking statistics for the term-location pairs (*earthquake*, *Oklahoma*) respectively (*earthquake*, *Texas*) they were identified, furthermore our approach is able to identify them earlier than SigniTrend. As shown in Table 5, we are able to detect 9 out of 11 earthquakes that were classified as significant by the USGS and present in our data set. GeoScope only detected a single *#earthquake* hashtag at January 7, 2015. The columns *Time* and *M* report the USGS reported earthquake date and magnitude. Δ contains the delay in minutes from the earthquake to our first detected event that surpassed the threshold of 3σ. $T_{t,q}$ denotes the set of all tweets within a 1-hour window of the event matching the query term $q =$ *"earthquake"*. $T_{t,q,g} \subseteq T_{t,q}$ denotes the subset of tweets, which also match our geo token g of the earthquake's location. *Event Keywords* are the terms and pairs that were significant. In the last columns we indicate whether *GeoScope* and *Our* method were able to identify this event.

We did *not* configure our system specifically for this purpose by specifying query terms beforehand (in contrast to specialized systems such as Sakaki et al. [20] that require keyword selectors). For this experiment, we later extracted earthquake related events from the general trend report file, that also includes all other events which occurred during this time period. The top other events detected on each day are presented in Table 6, along with the tokens clustered as explained in Section 3.7.

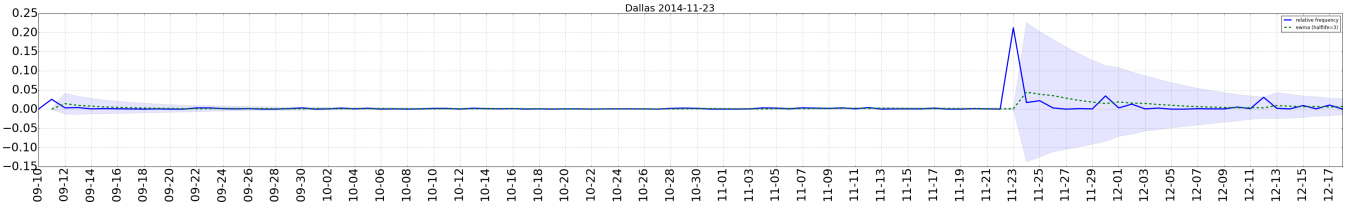
[2]http://earthquake.usgs.gov/earthquakes/



(a) Mentions of term “earthquake” regardless of location



(b) Mentions of term “earthquake” in Guam matching at latitude 144 ± 1 and longitude 13 ± 1



(c) Mentions of term “earthquake” in Dallas, Texas matching at latitude -97 ± 1 and longitude 33 ± 1

Figure 6: Frequency of the term “earthquake” globally vs. locally.

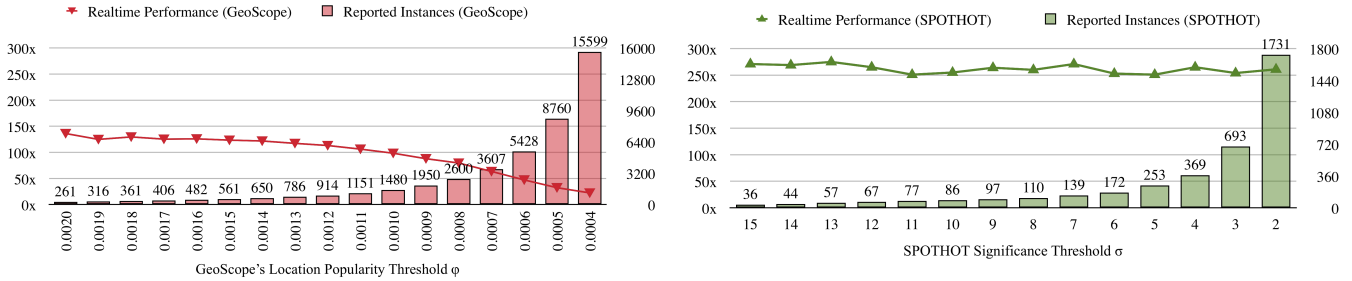
Table 5: Significant earthquakes that exhibit a minimum of 10 tweets within 1 hour of the actual time.

Time	Lat	Lng	Nearby City (distance)	M	Δ	σ	$\frac{\lvert T_{t,q,g}\rvert}{\lvert T_{t,q}\rvert}$	Event Keywords	[8]	Our
14-09-17	144.4	13.8	Hagatna, Guam (47km)	6.7	2.3	21.7	93/114	guam, {guam, earthquake}, {guam_county, earthquake}	✗	✓
14-10-02	-98.0	37.2	Wichita, KA (73km)	4.3	-	-	13/51	-	✗	✗
14-10-10	-96.8	35.9	Oklahoma, OK (86km)	4.2	53.7	3.0	41/54	oklahoma, {oklahoma, earthquake}	✗	✓
14-11-12	-97.6	37.3	Wichita, KA (53km)	4.9	2.0	18.0	138/322	kansas, {earthquake, felt}, {kansas, earthquake}	✗	✓
14-11-20	-121.5	36.8	Santa Cruz, CA (65km)	4.2	0.2	18.7	134/148	california, {california, earthquake}, {monterey_county, earthquake}	✗	✓
14-11-21	127.1	2.3	Bitung, Indon. (228km)	6.5	-	-	13/19	-	✗	✗
14-12-01	-111.7	35.0	Phoenix, AZ (179km)	4.7	3.4	22.8	119/148	arizona, {arizona, earthquake}, {earthquake, flagstaff}	✗	✓
14-12-30	-118.3	33.6	Long Beach, CA (21km)	3.9	0.7	22.5	213/245	california, {california, earthquake}, {los_angeles, earthquake}	✗	✓
15-01-07	-96.9	32.8	Irving, TX (6km)	3.6	22.9	3.1	269/334	{denton_county, earthquake}	✓	✓
15-01-20	-121.0	36.4	Santa Cruz, CA (80km)	4.4	3.4	8.0	37/51	california, {california, earthquake}, {monterey_county, earthquake}	✗	✓
15-01-28	-124.6	40.3	Sacramento, CA (330km)	5.7	51.1	3.0	25/37	earthquake	✗	✓

Note: data and method were *not* optimized for earthquake detection, but *all* events were tracked at the same time.
The detected earthquakes were significant at a significance level of $\sigma \geq 3$ in the full Twitter feed.

Table 6: Top events detected at the same day as the earthquakes in Table 5.

Date	σ	event terms	description
2014-09-17T02:50	247.8	#selfiesfornash, nash, #selfiefornash, #sefiesfornash	Teenager star “Nash Grier” asked his fans for selfies
2014-12-01T02:56	223.0	United_States_of_America, #soultrainawards	Soul Train Music Awards
2014-12-01T02:00	167.2	#thewalkingdead, beth, di, cry, #ripbeth, #asknorman, daryl	TV Series “The Walking Dead”
2015-01-07T22:00	135.1	hopkins, kati, England, #cbb, United_Kingdom, London, Greater_London, North_West_England	Katie Hopkins passes judgement in TV Show “Celebrity Big Brother”
2014-11-12T23:00	122.2	#followmehayes, hay	Fans send “folow me please” wishes to Teenager star “Hayes Grier”
2014-09-17T20:00	113.1	England, yaya, #gbbo, United_Kingdom, Greater_London, London, North_West_England	Famous TV Show “The Great British Bake Off" live
2014-11-20T04:05	102.3	United_States_of_America, #ahsfreakshow, New_York, Massachusetts	Fans of TV show “American Horror Story” talking about new Release Date
2014-09-17T20:51	98.0	boateng, goal	Jerome Boateng (Bayern Munich) scored the winning goal against Manchester City
2014-10-10T21:00	83.8	#5sosgoodgirls, #5sosgoodgirlsmusicvideo, 5so	Famous Teenager Band release new video
2014-10-10T10:42	70.3	kailash, satyarthi, Nobel_Peace_Prize, malala, Malala_Yousafzai	Malala and Kailash Satyarthi win Nobel Peace Prize
2014-11-12T04:38	66.1	#soa, #soafx, #finalride, abel	TV Series “Sons of Anarchy”
2015-01-28T21:00	59.8	Lionel_Messi, Neymar, suarez	Famous soccer players in Atlético Madrid against Barcelona.
2015-01-28T21:53	55.3	eriksen, Greater_London, London	Christian Eriksen scored a soccer goal



(a) GeoScope performance for decreasing thresholds.

(b) SPOTHOT with increasing significance threshold σ.

Figure 7: Scalability of GeoScope and SPOTHOT.
Left Y-axis: real-time performance. Right Y-axis: Average number of reported events per hour.

4.5 Scalability

In this section we demonstrate the efficiency of our algorithm. The following experiments were executed on a laptop with a 2.4 GHz Core i7 CPU and 8 GB RAM. We re-implemented GeoScope [8] and using their suggested parameter values obtained similar results to those reported by Budak et al. [8]. As long as we use only hashtags (as done by Budak et al.), we obtained a performance of around $480\times$ real-time: Processing one day of tweets took about 3 minutes. Because concepts like hashtags are likely to be absent in data sources other than Twitter, we then dropped this restriction and instead process the complete stemmed text of each tweet, as done in our approach. This makes the methods more comparable, but the increased amount of data reduced the throughput of GeoScope to around $130\times$ real-time. In Figure 7a we compare the performance of the method to the number of reported location-topic events within a window size of one hour. If we decrease GeoScope's threshold parameters the total number of reported instances increases exponentially from 261 to 15,599 whereas the runtime performance decreases substantially from $136\times$ to $32\times$ real-time because the thresholds are less selective. Because the hashing strategy used by SPOTHOT focuses on the interesting combinations directly (instead of tracking the most frequent words and locations independently) the performance is both faster (at around $240\times$ real-time) and does not degrade as much for reasonable values of σ, as seen in Figure 7b.

If a user wants to discover local events at geo-locations with low Twitter activity (as noted before, all of Germany has less Tweets than New York City alone; Table 1), GeoScope's threshold parameters have to be set to low values to enable tracking for less frequent locations. For example on the day of the Guam earthquake (c.f. Table 5) the location *Guam County* ranked only 834th most frequent; thus GeoScope's ϕ parameter needs to be much smaller than $\phi \leq \frac{1}{834} \approx 0.001$ in order to be included. With such low threshold values, the number of reported instances increases to thousands of reported location-topic combinations (c.f. Figure 7a).

GeoScope does not use a measure of unusual increase in frequency, but focuses on top-k frequent words that occur in a single top-k frequent location only. This includes many trivial combinations such as the hashtags *#ElPaso*, *#Milano*, and *#NYC* being reported as interesting "trends" for *El Paso County*, *Milan*, and *New York City* respectively. Such trivial combinations account for a large share of the instances reported by GeoScope, and make it hard to find the really interesting events due to the result set size. The significance score used by our new SPOTHOT method is normalized by the expected (local) frequency and thus avoids such uninteresting combinations that occur every day at these locations, and thus our result size remains much smaller.

5. CONCLUSION

We enhanced and refined the SingiTrend approach with the ability to use geographic information to detect events in fast and very large data streams, and showed that

1. Mapping coordinates to a token representation allows efficient and effective integration of geographic information in an analysis pipeline originally designed for textual data.
2. While grid-based tokens give a guaranteed resolution, tokens based on administrative boundaries allow detecting events at different granularities.
3. Co-occurrence of terms and location yields insight into events happening around the globe and enables data scientists to study for example new years celebrations in a geo-temporal context, or the influence of TV shows onto Twitter usage.
4. Variance-based normalization yields more interesting topics and events by adjusting for differences in user density and activity. By using incremental statistics, we remove the need to expire old data and improve performance.
5. Decoupling data aggregation and statistics tracking permits the use of short timeframes for counting and alerting, while at the same time we can use larger timeframes for more robust statistics.
6. Probabilistic counting using hash tables allows to scale this approach to very large data sets, well capable of performing such an analysis in real-time on thousands of tweets per second.
7. Due to the scalability improvements, the a priori specification of interesting topics and regions of interest required by earlier approaches is no longer necessary.

Future research directions include: (i) the classification of events into categories such as sports and politics; (ii) the handling of recurring events such as weekly TV shows; (iii) a visualization suitable for non-expert users that integrate both the cluster structure of the tokens as well as their geographic affinity; (iv) a drill-down functionality to study a specific subset of the data only.

References

[1] H. Abdelhaq, M. Gertz, and C. Sengstock. "Spatio-temporal Characteristics of Bursty Words in Twitter Streams". In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), Orlando, FL*. 2013, pp. 194–203. DOI: 10.1145/2525314.2525354.

[2] H. Abdelhaq, C. Sengstock, and M. Gertz. “EvenTweet: Online localized event detection from Twitter”. In: *Proceedings of the VLDB Endowment* 6.12 (2013), pp. 1326–1329.

[3] T. Akidau, A. Balikov, K. Bekiroglu, S. Chernyak, J. Haberman, R. Lax, S. McVeety, D. Mills, P. Nordstrom, and S. Whittle. “MillWheel: Fault-Tolerant Stream Processing at Internet Scale”. In: *Proceedings of the VLDB Endowment* 6.11 (2013), pp. 1033–1044.

[4] J. Allan, V. Lavrenko, D. Malin, and R. Swan. “Detections, bounds, and timelines: UMass and TDT-3”. In: *Proceedings of Topic Detection and Tracking (TDT–3)*. 2000, pp. 167–174.

[5] F. Alvanaki, S. Michel, K. Ramamritham, and G. Weikum. “See what’s enBlogue: real-time emergent topic identification in social media”. In: *Proceedings of the 15th International Conference on Extending Database Technology (EDBT), Berlin, Germany*. 2012, pp. 336–347.

[6] N. Bansal and N. Koudas. “Blogscope: a system for online analysis of high volume text streams”. In: *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB), Vienna, Austria*. 2007, pp. 1410–1413.

[7] B. H. Bloom. “Space/time trade-offs in hash coding with allowable errors”. In: *Communications of the ACM* 13.7 (1970), pp. 422–426.

[8] C. Budak, T. Georgiou, D. Agrawal, and A. El Abbadi. “GeoScope: Online detection of geo-correlated information trends in social networks”. In: *Proceedings of the VLDB Endowment* 7.4 (2013), pp. 229–240.

[9] T. M. Chan. “Approximate Nearest Neighbor Queries Revisited”. In: *Discrete & Computational Geometry* 20.3 (1998), pp. 359–373. DOI: 10.1007/PL00009390.

[10] G. Cormode and S. Muthukrishnan. “An improved data stream summary: the count-min sketch and its applications”. In: *J. Algorithms* 55.1 (2005), pp. 58–75. DOI: 10.1016/j.jalgor.2003.12.001.

[11] T. Finch. *Incremental calculation of weighted mean and variance*. Tech. rep. University of Cambridge, 2009.

[12] H.-G. Kim, S. Lee, and S. Kyeong. “Discovering hot topics using Twitter streaming data social topic detection and geographic clustering”. In: *Proc. ASONAM*. 2013.

[13] J. Kleinberg. “Bursty and hierarchical structure in streams”. In: *Data Mining and Knowledge Discovery* 7.4 (2003), pp. 373–397. DOI: 10.1023/A:1024940629314.

[14] V. Lampos, T. De Bie, and N. Cristianini. “Flu detector-tracking epidemics on Twitter”. In: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), Barcelona, Spain*. 2010, pp. 599–602. DOI: 10.1007/978-3-642-15939-8_42.

[15] R. Lee and K. Sumiya. “Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection”. In: *Proc. LBSN*. 2010.

[16] J. Leskovec, L. Backstrom, and J. Kleinberg. “Meme-tracking and the dynamics of the news cycle”. In: *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Paris, France*. 2009, pp. 497–506.

[17] W. Li, C. Eickhoff, and A. P. de Vries. “Geo-spatial Domain Expertise in Microblogs”. In: *Advances in Information Retrieval - Proceedings of the 36th European Conference on IR Research (ECIR), Amsterdam, Netherlands*. 2014, pp. 487–492. DOI: 10.1007/978-3-319-06028-6_46.

[18] M. Mathioudakis and N. Koudas. “Twittermonitor: trend detection over the Twitter stream”. In: *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Indianapolis, IN*. 2010, pp. 1155–1158.

[19] M. Platakis, D. Kotsakos, and D. Gunopulos. “Searching for events in the blogosphere”. In: *Proceedings of the 18th International Conference on World Wide Web (WWW), Madrid, Spain*. 2009, pp. 1225–1226.

[20] 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), Raleigh, NC*. 2010, pp. 851–860.

[21] E. Schubert, A. Koos, T. Emrich, A. Züfle, K. A. Schmid, and A. Zimek. “A Framework for Clustering Uncertain Data”. In: *Proceedings of the VLDB Endowment* 8.12 (2015), pp. 1976–1979. DOI: 10.14778/2824032.2824115.

[22] E. Schubert and OpenStreetMap Contributors. *Fast Reverse Geocoder using OpenStreetMap data*. Open Data LMU. Dec. 2015. DOI: 10.5282/ubm/data.61.

[23] E. Schubert, M. Weiler, and H.-P. Kriegel. “SigniTrend: Scalable Detection of Emerging Topics in Textual Streams by Hashed Significance Thresholds”. In: *Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), New York, NY*. 2014, pp. 871–880. DOI: 10.1145/2623330.2623740.

[24] Y. Takahashi, T. Utsuro, M. Yoshioka, N. Kando, T. Fukuhara, H. Nakagawa, and Y. Kiyota. “Applying a Burst Model to Detect Bursty Topics in a Topic Model”. In: *Advances in Natural Language Processing – Proceedings of the 8th International Conference on NLP, JapTAL 2012, Kanazawa, Japan, October*. 2012, pp. 239–249. DOI: 10.1007/978-3-642-33983-7_24.

[25] G. B. Tran and M. Alrifai. “Indexing and analyzing Wikipedia’s current events portal, the daily news summaries by the crowd”. In: *Proceedings of the 23rd International Conference on World Wide Web (WWW), Seoul, Korea*. 2014, pp. 511–516.

[26] X. Wang, Y. Zhang, W. Zhang, and X. Lin. “Efficiently identify local frequent keyword co-occurrence patterns in geo-tagged Twitter stream”. In: *Proceedings of the 37th International Conference on Research and Development in Information Retrieval (SIGIR), Gold Coast, QLD, Australia*. 2014, pp. 1215–1218.

[27] B. P. Welford. “Note on a Method for Calculating Corrected Sums of Squares and Products”. In: *Technometrics* 4.3 (1962), pp. 419–420. DOI: 10.2307/1266577.

[28] D. H. D. West. “Updating mean and variance estimates: an improved method”. In: *Communications of the ACM* 22.9 (1979), pp. 532–535. DOI: 10.1145/359146.359153.

[29] Y. Yang, T. Pierce, and J. Carbonell. “A study of retrospective and on-line event detection”. In: *Proceedings of the 32nd International Conference on Research and Development in Information Retrieval (SIGIR), Boston, MA*. 1998, pp. 28–36.