The Early Bird Catches The Term: Combining Twitter and News Data For Event Detection and Situational Awareness

Nicholas Thapen  Donal Simmie  Chris Hankin
Institute for Security Science and Technology, Imperial College London

Abstract

Twitter updates now represent an enormous stream of information originating from a wide variety of formal and informal sources, much of which is relevant to real-world events. In this paper we adapt existing bio-surveillance algorithms to detect localised spikes in Twitter activity corresponding to real events with a high level of confidence. We then develop a methodology to automatically summarise these events, both by providing the tweets which fully describe the event and by linking to highly relevant news articles. We apply our methods to outbreaks of illness and events strongly affecting sentiment. In both case studies we are able to detect events verifiable by third party sources and produce high quality summaries.

1 Introduction

Updates posted on social media platforms such as Twitter contain a great deal of information about events in the physical world, with the majority of topics discussed on Twitter being news related (Kwak et al. 2010). Twitter can therefore be used as an information source in order to detect real world events. The content and metadata contained in the tweets can then be leveraged to describe the events and provide context and situational awareness. Applications of event detection and summarisation on Twitter have included the detection of disease outbreaks (Aramaki, Maskawa, and Morita 2011), natural disasters such as earthquakes (Sakaki, Okazaki, and Matsuo 2010) and reaction to sporting events (Zubiaga et al. 2012).

Using the Twitter stream for event detection yields a variety of advantages. Normally in order to automatically detect real-world events a variety of official and media sources would have to be tracked. These are usually published with some lag time, and any system monitoring them programmatically would require customisation for each source since they are not formatted in any standard way. Twitter provides a real-time stream of information that can be accessed via a single API. In addition a rich variety of sources publish information to Twitter, since it is a forum both for the traditional media and for a newer brand of citizen journalists (Hermida 2010). Tweets also contain metadata that can be mined for information, including location data, user-supplied hashtags and user profile information such as follower-friend relationships. The primary drawback of using Twitter is that it is an unstructured source that contains a great deal of noise along with its signal. Tweets can be inaccurate as a result of rumour, gossip or active manipulation via spamming.

In this paper we apply existing bio-surveillance algorithms to detect candidate events from the Twitter stream, employing customised filtering techniques to remove spurious events. We then extract the terms from the event tweets which best characterise the event and are most efficacious in retrieving related news. These terms are used to filter and rank the most informative tweets for presentation to the user along with the most relevant news articles.

Our techniques are evaluated using two case studies, both using a dataset of geo-located tweets from England and Wales collected in 2014. The primary case study is the detection of illness outbreak events. We then generalise our techniques to events strongly affecting Twitter sentiment, such as celebrity deaths and big sports matches.

In Section 2 we discuss related work in the area of event detection and situational awareness using Twitter. Sections 3 and 4 outline our methodology and results. We then discuss our conclusions in Section 5.

2 Related Work

Much of the work on event detection using social media has focused on using topic detection methods to identify breaking news stories. Streaming document similarity measures (Petrović, Osborne, and Lavrenko 2010), (Osborne et al. 2014) and online incremental clustering (Becker, Naaman, and Gravano 2011) have been shown to be effective for this purpose.

Other approaches have aimed to pick up more localised events. These have included searching for spatial clusters in tweets (Walther and Kaisser 2013), leveraging the social network structure (Aggarwal and Subbian 2012), analysing the patterns of communication activity (Chierichetti et al. 2014) and identifying significant keywords by their spatial signature (Abdelhaq, Sengstock, and Gertz 2013).

In the field of disease outbreak detection efforts have mostly focused on tracking levels of influenza by comparing them to the level of self-reported influenza on Twitter,
in studies such as (Broniatowski, Paul, and Dredze 2013) and (Li and Cardie 2013). Existing disease outbreak detection algorithms have also been applied to Twitter data, for example in a case study (Diaz-Aviles et al. 2012) of a non-seasonal disease outbreak of Enterohemorrhagic Escherichia coli (EHEC) in Germany. They searched for tweets from Germany matching the keyword “EHEC”, and used the daily tweet counts as input to their epidemic detection algorithms. Using this methodology an alert for the EHEC outbreak was triggered before standard alerting procedures would have detected it. Our study uses a modified and generalised version of this event detection approach.

Diaz-Aviles et al. also attempted to summarize outbreak events by selecting the most relevant tweets, using a customized ranking algorithm. Other studies which have summarised events on Twitter by selecting the most relevant tweets include (Zubiaga et al. 2012) and (Long et al. 2011). There has been less related work on linking or substantiating events detected from Twitter with traditional news media. One study (Abel et al. 2011) analysed various methods of contextualizing Twitter activities by linking them to news articles. The methods they examined included finding tweets with explicit URL links to news articles, using the content of tweets, hashtags and entity recognition. The best non-URL based strategy that they found was the comparison of named entities extracted from news articles using OpenCalais with the content of the tweets.

3 Methodology

3.1 Problem Definition

Our definition of a real-world event within the context of Twitter is taken from (Becker, Naaman, and Gravano 2011), with the exception that we have added a concept of event location.

Definition 1. (Event) An event is a real-world occurrence e with (1) an associated time period \(T_e\) and (2) a time-ordered stream of Twitter messages \(M_e\), of substantial volume, discussing the occurrence and published during time \(T_e\). The event has a location \(L_e\) where it took place, which may be specific or cover a large area, and the messages have a set of locations \(L_{M_1}, \ldots, L_{M_n}\) which they were sent from.

When given a time-ordered stream of Twitter messages \(M\), the event detection problem is therefore one of identifying the events \(e_1, \ldots, e_n\) that are present in this stream and their associated time periods \(T_e\) and messages \(M_e\). It is also valuable to identify the primary location or locations \(L_{M_1}\) that messages have originated from, and if possible the event location \(L_e\). The situational awareness problem is one of taking the time period \(T_e\) and messages \(M_e\) and producing an understandable summary of the event and its context.

3.2 Overview

Our approach to the event detection problem incorporates location by detecting deviations from baseline levels of tweet activity in specific geographical areas. This allows us to track the location of messages relating to events, and in some cases determine the event location itself. We break down the problem by defining classes of events which we are interested in and formulating a set of groups of keywords which describe each class. In this paper we have examined two distinct classes:

- Outbreaks of symptoms of illness, such as coughing or itching
- Events triggering emotional states, such as happiness or sadness

We track the number of tweets mentioning each keyword in each of our areas and use modified bio-surveillance algorithms to detect spikes in activity which we can classify as events.

Initially we designed the system with health symptom event detection as the primary use case. This led to a system design focused around keywords and aliases for their keywords, since a limited range of illness symptoms characterises most common diseases and the vocabulary used to describe these symptoms is also relatively limited. After several iterations of this approach we noted that it could be viable as a general event detection and situational awareness method, so we added another event class, emotion-based events, to test out the feasibility of the general approach.

Our situational awareness approach is based on identifying terms from the event tweets which characterise the events and using them to retrieve relevant news articles and identify the most informative tweets. The news search uses metrics based on cosine similarity to ensure that searches return related groups of articles.

3.3 Architecture

The general approach can be described by the architecture in Figure 1. Every new event class requires a list of keyword groups. Optionally a domain specific data pre-processing step can also be included. For example in the health symptom case we employ a machine learning classifier to remove noise (those tweets not actually concerning health). These are the only two aspects of the design that need to be altered to provide event detection and situational awareness to a new problem domain.

3.4 Event Classes

We now go into a more detailed explanation of our event classes and how we formulated the keyword groups. Each keyword group consists of a primary keyword which is used to identify the group, e.g. vomit, and a number of aliases that expand the group, e.g. throwing up, being sick, etc.

Illness Symptoms To build up a list of symptoms and related keywords we searched Freebase for /medicine/symptom. Each of these symptoms is defined as a primary keyword. They are returned with a list of aliases that are used as related keywords.

The next step in creating a symptom list was to filter these symptoms by their frequency in the Twitter data, since only those words actually used on Twitter are of interest. All symptoms with less than 10 mentions in the Twitter data were removed from this candidate list. This excluded a large proportion of symptoms, reducing the set from 2000 to 200.
We further limited the set by removing symptoms not related to infectious diseases. We also added primary keywords and aliases for some common conditions such as hayfever and flu. This step resulted in 46 symptom groups.

**Emotion States** For a list of emotion states and associated keywords we used the work of Shaver et al. They conducted research (Shaver et al. 1987) to determine which sets of words were linked to emotions and how these clusters together. We took the six basic emotions identified in the work as primary keywords: love, joy, surprise, sadness, anger and fear. Shaver’s work associated each of these with a list of terms to form a tree. We took the terms from lower leaves on the tree for each emotion as our alias sets (See Table 1 for examples). The only alteration we made was that after some initial analysis we discovered that the term “happy” from the “joy” category was a very strong signal of special events such as Valentine’s Day, Mother’s Day and Easter. It was also very often used on a daily basis due to people offering birthday greetings. We therefore separated “happy” into its own category separate from “joy”.

In addition we employed SentiStrength (Thelwall et al. 2010), a sentiment analysis tool, to classify our tweets into positive and negative emotional sentiment. We took those classified as being very positive and very negative as additional emotion states.

### 3.5 Data Collection

Using Twitter’s live streaming API we collected geo-tagged tweets between 11th February 2014 and 11th October 2014. Tweets were collected from within a geographical bounding box containing England and Wales. Retweets were excluded due to our focus on tweets as primary reports or reactions to events. This resulted in a data-set of 95,852,214 tweets from 1,230,015 users. 1.6% of users geo-tag their tweets (Lee-taru et al. 2013), so our data is a limited sample of the total tweet volume from England and Wales during this period.

We chose to use only geo-tagged tweets since they contain metadata giving an accurate location for the user. This allows us to locate each tweet within our geographical model.

### 3.6 Location Assignment

Our methodology relies on the collection of baseline levels of tweet activity in an area, so that alarms can be triggered when this activity increases. We therefore amalgamated the fine-grained location information from the geo-coded tweets by assigning them to broader geographical areas. We used a data driven approach to generate the geographical areas rather than using administrative areas such as towns or counties. This technique allowed us to select only those areas with a minimum level of tweet activity, and also did not require any additional map data. It would therefore be reusuable for any region or country with a sufficient level of Twitter usage.

We began by viewing a sample of the collected tweets as geo-social points. Viewed on a map these clearly clustered in the densely populated areas of England and Wales. We therefore decided to use a clustering algorithm on these points in order to separate out areas for study. We employed the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm (Ester et al.) for clustering, as this does not require a priori knowledge of the number of clusters in the data. The features provided to DBSCAN were the latitudes and longitudes of the tweets.

The clusters produced by the algorithm matched the most populated areas, corresponding to the largest cities or towns in the UK as shown in Figure 2. They also separated most cities into distinct clusters (a notable exception being the conglomeration of Liverpool and Manchester). In total 39 clusters were created for England and Wales and each was given an ID and a label. We then created a convex hull around each cluster, providing a polygon that can be used to check whether a point is in the cluster or outside it. Points outside all of the clusters were assigned to a special ‘noise’ cluster, and not included in the analysis. Overall 80% of tweets were assigned to specific clusters and the remainder to noise, giving us good coverage of geo-tagged tweets using our cluster areas.

### 3.7 Tweet Processing

As tweets are received by our system they are processed and assigned to the symptom and emotion state classes via key-

---

**Table 1: Selected emotion keyword groups and some of their aliases:** keyword groups contain a primary keyword and aliases (taken from Shaver et al. ).

| Keyword |_aliases |
|---------|---------|
| surprise | amazed, astonished, surprised... |
| sadness  | depressed, unhappy, crying... |
| joy      | glad, delighted, pleased... |
word matching. They are assigned a location by checking whether they fall into one of our cluster areas.

For the illness symptoms we introduce a noise removal stage at this point. It is particularly relevant for this class of events because there are many fewer tweets relating to illness than showing emotion states. This means that the signal is more easily blocked out by random noise. To remove noise we construct a machine learning classifier with the aim of removing tweets containing alternative word usages or general illness discussion rather than reporting of illness events. The classifier we use is a linear SVM trained on a semi-supervised cascading training set (Sadilek, Kautz, and Silenzio 2012). This classifier uses the LibSVM (Chang and Lin 2011) library, and achieves a classification accuracy of 96.1% on a test set of manually classified tweets.

The number of tweets assigned to each class in each area are then saved on a daily basis. These counts are first normalised to take account of Twitter’s daily effect pattern, which shows more tweeting on weekends than weekdays. Event detection is run daily since we are attempting to pick up temporally coarse-grained events. Disease outbreaks take weeks to develop, and events that shift public sentiment or emotion will generally take hours or days to unfold.

3.8 Detecting Events

Our event detection methodology leverages considerable existing syndromic surveillance research by using an algorithm designed and developed by the Centers for Disease Control and Prevention (CDC), the Early Aberration Reporting System (EARS) (Hutwagner et al. 2003).

Definition 2. (Alarm) An alarm is an alert produced by the first stage of our event detection system. The alarm has an associated symptom and location. It also has a start and end date, and associated tweet counts for each date within this period. When certain criteria are met an alarm is deemed to be an event.

We employ the C2 and C3 variants of EARS. These algorithms operate on a time series of count data, which in our case is a count of daily symptomatic tweet activity. The C2 algorithm uses a sliding seven day baseline, and signals an alarm for a time \( t \) when the difference between the actual count at \( t \) and the moving average at \( t \) exceeds 3 standard deviations. The C3 algorithm is based on C2, and in effect triggers when there have been multiple C2 alarms over the previous 3 days.

These C2 and C3 candidate alarms are then grouped together so that alarms for the same keyword set and area on consecutive days are treated as a single alarm. An alarm is therefore made up of one or more days, each with an observed count of tweets.

Some of our Twitter count time series data is zero-skewed and non-normal, since the number of geo-tagged users reporting illness can be low. The number of standard deviations from the mean used in the C2 and C3 algorithms can be an unreliable measure of central tendency in those circumstances. Hence to determine how far above general baseline activity an observed count is we employ the median of the series to date and the Median Absolute Deviation (MAD) to produce a new metric of alarm severity. The number of Median Absolute Deviations from the median, \( \mu \), gives a comparable figure across alarms as to how sharp a rise has been over expected levels. This figure is produced from the following equation:

\[
\mu = \frac{(\text{observation} - \text{median})}{\text{MAD}} \quad (1)
\]

We then find the highest metric for an alarm, \( \mu_{\text{max}} \), by finding the highest value of \( \mu \) within the observations making up the alarm.

\[
\mu_{\text{max}} = \arg \max_{\mu} (\text{observations in alarm}) \quad (2)
\]

The \( \mu_{\text{max}} \) is the primary statistic which we use to determine which events are real and which have just been generated by random noise. Details of the threshold value which we use for this and how we selected it are contained in Section 4.

Another statistic which we employ in order to filter out noise is the tweet-user ratio. This is the ratio of tweets in an event to that of distinct users involved in an event. A high value of this statistic would imply that some users have tweeted a large number of times across a short time period, which is an indication that they may be spammers and that the alarm is spurious.
In summary, we use the output from EARS to produce alarms. We filter the alarms to a set of high likelihood events by using the $\mu_{max}$ and tweet-user ratio parameters.

### 3.9 Situational Awareness

Once an event has been identified our next objective is to automatically provide additional context for it, which may provide an explanation of the underlying cause. A human interpreter could achieve this by reading all of the tweets and synthesizing them into a textual explanation, which might be some text such as ‘People reacting to the death of Robin Williams’. We do this in two main ways: by providing the most representative tweets from those that triggered the alarm, and by linking to relevant news articles. The steps involved in the Terms, News and Tweets (TNT) Event Summarisation process are detailed in Algorithm 1. The steps and terminology are then explained in more detail.

**Algorithm 1 Terms, News and Tweets (TNT) Event Summarisation**

1. Fetch gist tweets and baseline tweets
2. if |gist tweets| < 30 then
   3. Do not attempt to summarise event
   else
5. Extract unigrams and bigrams appearing in at least 5% of the gist tweets
6. for all ngrams extracted do
7. Perform Fisher’s Exact Test to determine whether ngram is significantly more likely to appear in gist than baseline
8. for Top 2 most significant unigrams and bigrams and the primary keyword do
9. Search news database using ngram for the alarm’s date range and return the top 10 documents
10. for ngrams with PCSS values above threshold do
11. Compute title similarity PCSS between ngram documents and those for each other ngram
12. Good search terms ← term with title similarity PCSS above threshold
13. Good articles ← documents returned from good search terms
14. Filtered tweets ← tweets containing a good search term
15. Rank good articles by cosine similarity to average vector of good news articles
16. Rank filtered tweets by cosine similarity to average vector of filtered tweets

1. The first step is to retrieve the relevant tweets from the processed tweet and alarm databases. Tweets are fetched for both the alarm gist and from a historical baseline. We discard those events with fewer than 30 tweets as we found that they did not contain sufficient data to produce good summarisation results.

**Definition 3. (Gist)** The gist consists of the tweets for the time period of the event which match the event’s keyword group and area.

**Definition 4. (Baseline)** The baseline consists of the tweets for the same keyword group and area as an event from the 28 days prior to that event.

5. The next task is to find unigrams and bigrams that are more prevalent in the gist than in the baseline. These are likely to come from tweets discussing the event and will thus be characteristic of the event. We first extract the most common unigrams and bigrams from both sets of tweets, after removal of stopwords. Our list of stopwords includes a standard list, plus the 200 most frequent words from our tweet database. We select all non-stopwords that appear in at least 5% of the tweets.

7. We then do a Fisher’s Exact Test to determine which of the common unigrams and bigrams in the gist appear significantly more frequently ($\alpha < 0.05$) here than in the baseline set. Our candidate terms are the top two most significant unigrams and bigrams. We select the top two as this was found to give the best results on our test examples. To this set we append the primary keyword that triggered the alarm.

9. Using these candidate terms we then perform a search on Google for documents published in the United Kingdom during the time period of the alarm. Due to Google’s Terms of Service this step was performed manually. A fully automated system would replace this step with a search of a news database, which could be created by pulling down news articles from RSS feeds of major content providers.

10. We take the first 10 documents retrieved for each search term, remove stopwords and apply stemming using a Lancaster stemmer. We then convert each document into a Term Frequency/Inverse Document Frequency (TF/IDF) vector. In order to determine whether the search term has retrieved a coherent set of related documents we define a metric based on cosine similarity, the Pairwise Cosine Similarity Score (PCSS).

- **The Pairwise Cosine Similarity Score** of a group of TF/IDF vectors is calculated by taking the cosine similarity between each pair of vectors and adding them to a set. The standard deviation of this set is subtracted from its mean to form a score.

The PCSS rewards articles which are similar and penalises any variance across those article similarities, this reduces the effect of some articles being strongly related in the document set and others being highly unrelated. Any term which retrieves a set of documents with a score below a threshold value (determined by a parameter selection process detailed in section 4) is not considered further.

It is possible for a search term to hit on a coherent set of documents purely by chance, perhaps by finding news articles related to another event in a different part of the world. In order to guard against this we institute another check to ensure that the set of documents returned from a search term is sufficiently closely related to the set returned from at least one other search term.

12. In order to perform this check we compare the titles of the articles returned from the two different searches using
a similar process to our earlier document comparison. We found it more effective to compare titles than whole documents, since sets of documents with similar topics can contain similar language even for fairly unrelated search terms. For example the terms “ebola” and “flu” will both return health-related documents containing similar language, but we would not wish to say that these search terms are related.

To convert the titles to TF/IDF vectors we remove stopwords but do not apply stemming. Since the titles are so short we include all unigrams, bigrams and trigrams in the vector representation. We then compute a PCSS between the two document sets, pairing each document in the first set with each in the second and vice versa.  A search term must be related to at least one other term for it to be used going forward.

\[ \text{Once TNT has identified good search terms we then return the news articles fetched using those terms.} \]

In order to rank the top news articles for a search we take the average TF/IDF vector and then rank the articles by cosine similarity to this average vector. We return the top ranked articles from each search term.

\[ \text{In order to return the most explanatory tweets we find the gist tweets that contain at least one of the good search terms. We then convert these into TF/IDF vectors and compute the average vector. The tweets are then ranked in the same way, by cosine similarity to the average vector, and we return the top 5 tweets.} \]

4 Results

There are three individual components to our event detection and situational awareness platform that require evaluation:

1. Event detection
2. Situational awareness
   (a) Linkage of relevant news articles
   (b) Ranking most informative tweets

4.1 Example Cases

To effectively evaluate all of these components required a varied set of example events and alarms. These were used in order to choose values for our threshold parameters. We compiled an initial set of 13 focus examples. These were taken from events that the authors knew had happened in the evaluation time period and from those alarms in our dataset with low and high values of \( \mu_{max} \). The event ID which will be used to refer to these events is composed of the first two letters of the event keyword followed by a 1-2 letter area code. The final part of the ID is the day and month of the event start date.

The focus examples were used to find sensible values that separated the high-confidence events from the low-confidence events. The most important threshold parameter in the context of the event detection is the \( \mu_{max} \) figure which measures the deviation of the alarm counts from the median level. Examining the distribution of the number of alarms for each value of \( \mu_{max} \) revealed that it started to tail off sharply at \( \mu_{max} \geq 5 \). We therefore took this as a value to segment additional test examples, drawing ten more at random with a \( \mu_{max} \) less than 5 and ten with a \( \mu_{max} \) greater than or equal to 5.

4.2 Event Detection Evaluation

Method It is difficult to provide a completely automated evaluation procedure for detecting previously unknown events. Diaz et al. used the time to detection on a known outbreak as their evaluation criterion (Diaz-Aviles et al. 2012). In our case we do not know a priori that these are genuine outbreaks or events. Hence we need to make an assessment of the alarms produced to see what they refer to and if there is a way of externally verifying that they are genuine events. For all 33 of the selected alarms the authors read the tweets and determined whether they described a real world event. The coders found 26 YES answers, 5 NO answers and 2 DISAGREED answers, producing a 94% agreement. Where an event was present they wrote a short summary.

For external verification of events two different methods were used, depending on whether the event was symptom-related or emotion-based. For symptom related events the activity spike was checked against official sources for the same time period. The General Practitioner (GP) in hours bulletin for England and Wales (Public Health England 2014) was used and an event was deemed verified if the symptom exhibited an increasing trend for that period. This detail is noted in the summary document produced by Public Health England for that reporting period. Emotion-based events were verified by checking if there were any articles (via Web search) that could corroborate the cause of the event (as given by the summary).

We manually investigated all examples from the initial focus set and found initial parameters for the score functions in our algorithms that worked reasonably well. These provided possible ranges of values which were evaluated more systematically over the entire alarm set. For event detection we evaluated which alarms were flagged as events by the system for each parameter value against whether those events were externally verifiable. The final evaluation for all algorithms contains all 33 of the selected alarms in both sets, not just the twenty expanded ‘test’ examples.

Results To determine if an alarm is an event that we should be concerned about we consider two properties of the alarm. The first is the tweet-user ratio. This provides a naive spam filter, as when this is high an alarm is mostly caused by one user tweeting multiple times. From exploratory testing we found a value of 1.5 separated our spam and genuine alarms very well, leaving only a small number of alarms with large tweet sets and some spam. The spam detection problem should be straightforward and will be addressed more completely in future work.

The second figure which gives the strength of the activity above the usual baseline is the \( \mu_{max} \) figure. This is the essence of the modified EARS algorithm and the value of this figure should generally separate events from non-events.

The criterion for selecting the best threshold for \( \mu_{max} \) is context dependent. We have used the balanced F1 measure for this scenario as that is a fair representation of both pre-
### Table 2: Evaluation set of events: showing whether they were externally verifiable and their $\mu_{max}$ value. *Note: this event not confirmed by the GP in hours report of that week. However, the following week showed an increase and it is possible that social media detected increased Influenza activity before this was confirmed by GP visits.

![Event Detection Max MAD Parameter Selection](image)

**Figured 3: $\mu_{max}$ event detection parameter selection**

...
A technique that extracts sets of terms characterising each topic in a group of documents. The success and error types used to compute the F0.5 measure are:

- **True positive**: relevant news returned for newsworthy event
- **False positive**: news returned for an event with no genuine news
- **True negative**: no news returned for an event with no genuine news
- **False negative**: no news returned for newsworthy event

The evaluation is presented in Figure 4 as well as the different levels of article PCSS that were iterated over to find the maximum F0.5 value in a step-wise procedure. It is clear from these images that the TNT algorithm has a higher F0.5 at all tested values of the article PCSS, due to its higher recall. The outcome of the parameter selection process was that a PCSS threshold of −0.08 produced the best results. Using this value the F0.5 was 0.79, showing that our system was successful in retrieving relevant news for the sample events.

Table 3: **STT tweet ranking evaluation**: The STT tweet summary fully matched the human-coded event summarisation in 21 cases. This yields a full match fraction of 0.81.

Selecting top ranked relevant news articles is one part of our situational awareness contribution. The second is the selection of tweets that provide a representative summary of an event.

**Top Ranked Tweets Evaluation** We select the summary tweets by choosing the top 5 tweets ranked by calculating the maximum cosine similarity between an average tweet TF/IDF vector and all tweets in the candidate set. This tweet set can be: 1) all tweets in the gist, 2) those filtered by selecting the extracted terms or 3) those from the filtered term set, that is, the extracted term set less any that don’t have a good news match. 1) is always available and is labelled the *Gist Top Tweets* (GTT). If terms have been found to be significantly different in frequency from the baseline then set 2) is available for use and if terms from that set have good news matches then set 3) can be used. The *Summary Top Tweets* (STT) are from set 3) if it exists and fallback to set 2) if the good news match terms are not available. If no terms were found to be significantly different from the baseline then only the GTT is available.

We have employed two evaluations for the tweet ranking exercise: comparison to human-coded event explanation and comparison between GTT and STT. The human-coded event explanations were created by both authors after reading through all of the tweets linked to each event. There were 26 alarms that had an identifiable cause. The tweet ranking match (to human-coded event assessment) performance is presented in Table 3. The tweets were considered a full match if a human summary of the 5 top ranked tweets would match the human-coded event explanation for the whole set of tweets.

The partial matches were: FRL-30-05 (Fever: London, May) and FLP-06-10 (Flu: Birmingham, October). These events had more than one explanatory cause. Currently our algorithms work best in the single event case. The three cases that did not match were: JONO-23-02 (Joy: Norwich, February), STL-26-08 (Sore throat: London, August) and SUN-29-08 (Surprise, Newcastle, August). The coders disagreed as to whether STL-26-08 was actually an event. The remaining two examples were not summarised well by the significant tweets as they both exhibited high disparity in terms used to describe a contextually related event and SUN-29-08 also included a number of spam tweets that distorted the results of TNT.

The second evaluation for the tweet ranking exercise was a comparison between the GTT and the STT. A qualitative assessment of the tweets led to the conclusion that STT tweets were better in 11 out of 33 cases and there was no sig-

| Match | Count |
|-------|-------|
| Full  | 21    |
| Partial | 2    |
| No    | 3     |
nificant difference between the two for 21 cases out of 33. In one case, FLP-06-10, the GTT included a mention of “flu jab” (one of the manually selected terms) which the STT did not include. Hence the STT provides an improvement over ranking based off the alarm tweets in 1/3 instances.

### 4.4 Notable Examples Discussion

We now discuss four example events that highlight the strengths and limitations of our approach. These examples are listed in Table 4.

The first example case is JONO-23-02. From a reading of the tweets there were definitely some relating to a single event: Norwich City Football Club beating Tottenham Hotspur Football Club 1 – 0 in a football match. Both TNT and LDA term extraction failed to find terms representative of this event. This was due to the disparity of the language used; the following example tweets should help elucidate this point:

- #canarycall absolutely delighted with the win :) good performance, good result
- #yellows almost didn’t go today glad i did
- so glad i chose to come today!#ncfc

It is difficult for a term-based solution to find any common thread here. Finding the cause of this event would require contextual knowledge of football matches, team names and commonly employed aliases. The news linkage algorithm did initially find a news story for the term “joy” on this date. The British Prime Minister “let out a little cry of joy” over David Bowie Scottish independence comments (Telegraph, Feb 24, 2014). The articles returned all concerned this story and were found to be closely related, but were dropped from the news linkage because they did not match those returned from the other search terms. This highlights the benefits of searching with multiple terms and ensuring that the results are related.

The second example is ASL-02-04. This event was due to increased levels of air pollution observed in London at the beginning of April, caused by a Saharan dust cloud. This event had a $\mu_{max}$ of 12 indicating a significant increase in baseline activity for the alert period. It was well summarised by all aspects of our situational awareness algorithm. The top ranked tweets provided by our summary method (STT) produced tweets more representative of the event than those from all tweets in the gist. This is demonstrated by the top tweet selected by both:

- STT top tweet: i can’t breathe #asthma #smog
- GTT top tweet: my asthma is literally so bad

Here selecting the top tweets from the filtered event set captures tweets representative of the event as opposed to the baseline illness activity. The news linkage for this example worked well, with all five of the top selected articles being representative of the event. The top article, “Air pollution reaches high levels in parts of England - BBC”, gives the cause of the event in the first few lines: “People with health problems have been warned to take particular care because of the pollution - a mix of local emissions and dust from the Sahara.”

The third case is VOL-20-04. Reading the tweets makes it clear that this one day event is caused by people feeling sick after eating too much chocolate on Easter Sunday. In this case the TNT summary and all tweet ranking return similar tweets as there is little baseline activity and that baseline activity is not strongly related. The top tweets from both sets therefore both produce good summaries:

- STT top tweet: seriously i feel sick having all this chocolate
- GTT top tweet: eaten too much chocolate feel sick

While the top ranked tweets are similar the event tweet filtering does remove baseline tweets referring to general illness. No good news searches were found in this case. This event may be valid in the context of social media but it is not newsworthy.

The fourth example is SAL-11-08 which is the UK Twitter reaction to the death of Robin Williams. These tweets from the sadness keyword group exhibit both the highest $\mu_{max}$ (20) and the highest overall tweet count for any single event (4472). The prominence of celebrity deaths within our detected events mirrors earlier findings (Petrović, Osborne, and Lavrenko 2010). As with all of our high $\mu_{max}$ events the TNT tweet ranking and news linkage work well. The top news article returned is an article reporting the death of Mr. Williams: “Robin Williams dies aged 63 in suspected suicide” (Telegraph, August 12, 2014). The top five ranked tweets by TNT tweet filtering are better than those ranked on all tweets as they remove baseline general sadness tweets from the ranking:

- STT top tweet: rip robin williams. sad day
- GTT top tweet: yep , very sad

### 5 Conclusion

We have presented techniques for event detection and situational awareness based on Twitter data. We have shown that they are robust and generalisable to different event classes. New event classes could be added to this system simply by producing a list of keywords of interest and an optional noise filter. Our event detection is based on the EARS biosurveillance algorithm with a novel filtering mechanism. The maximum Median Absolute Deviations from the median provides a robust statistic for determining the strength of relative spikes in count-based time series. As it is based on the median, this measure handles cases where data is non-normal as was the case for some of our symptom based geotagged tweets. The event detection approach achieved an F1 score of 0.9362 on our event examples.

By filtering to terms that are significantly different ($\alpha < 0.05$) in frequency from baseline levels we have extracted terms to search news sources for related articles. Where good news matches are found these revise our event term list. We have created two novel algorithms that provide additional situational awareness about an event from these event terms.
Firstly, we rank the filtered set of news articles to produce the top five representative articles. The news linkage, weighted towards precision, achieved an F0.5 score of 0.79 on our example set, with no false positives.

Secondly, we produce a top five ranked list of tweets that summarise an event. These ranked tweets are calculated from the tweet set, filtered by those that contain the extracted event terms. The top ranked tweets fully matched our human-coded event summaries in 21 out of 26 cases.

In future work we aim to improve our news linkage algorithm with a final step checking whether the articles returned are similar to the event tweets, using cosine similarity or other features such as entities identified in the news articles. Additional improvements to event detection would lie in improving spam detection and adding sentiment classification to our emotion model as a classifier. Collecting data over longer time periods would also allow us to look into using bio-surveillance algorithms which require seasonal baseline information.

6 Acknowledgements

This research was carried out in cooperation with the UK Defence Science and Technology Laboratory. It was funded by the U.S. Department of Defense’s Defense Threat Reduction Agency (DTRA), through contract HDTRA1-12-D-0003-0010.

References

[Abdelhaq, Sengstock, and Gertz 2013] Abdelhaq, H.; Sengstock, C.; and Gertz, M. 2013. Eventweat: Online localized event detection from twitter. Proceedings of the VLDB Endowment 6(12):1326–1329.

[Abel et al. 2011] Abel, F.; Gao, Q.; Houben, G.-J.; and Tao, K. 2011. Semantic enrichment of twitter posts for user profile construction on the social web. In The Semantic Web: Research and Applications. Springer. 375–389.

[Aggarwal and Subbian 2012] Aggarwal, C. C., and Subbian, K. 2012. Event detection in social streams. In SDM, volume 12, 624–635. SIAM.

[Aramaki, Maskawa, and Morita 2011] Aramaki, E.; Maskawa, S.; and Morita, M. 2011. Twitter catches the flu: Detecting influenza epidemics using twitter. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP ‘11, 1568–1576. Stroudsburg, PA, USA: Association for Computational Linguistics.

[Becker, Naaman, and Gravano 2011] Becker, H.; Naaman, M.; and Gravano, L. 2011. Beyond trending topics: Real-world event identification on twitter. ICWSM 11:438–441.

[Becker, Naaman, and Gravano 2011] Becker, H.; Naaman, M.; and Gravano, L. 2011. Beyond trending topics: Real-world event identification on twitter. ICWSM 11:438–441.

[Broniatowski, Paul, and Dredze 2013] Broniatowski, D. A.; Paul, M. J.; and Dredze, M. 2013. National and local influenza surveillance through twitter: An analysis of the 2012–2013 influenza epidemic. PloS one 8(12):e85672.

[Chang and Lin 2011] Chang, C.-C., and Lin, C.-J. 2011. LIB-SVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2:27:1–27:27.

[Chierichetti et al. 2014] Chierichetti, F.; Kleinberg, J.; Kumar, R.; Mahdian, M.; and Pandey, S. 2014. Event detection via communication pattern analysis. In Eighth International AAAI Conference on Weblogs and Social Media.

[Diaz-Aviles et al. 2012] Diaz-Aviles, E.; Stewart, A.; Velasco, E.; Denecke, K.; and Nejdl, W. 2012. Epidemic intelligence for the crowd, by the crowd. In ICWSM.

[Ester et al. 2014] Ester, M.; Kriegel, H.-P.; Sander, J.; and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise.

[Hermida 2010] Hermida, A. 2010. Twittering the news: The emergence of ambient journalism. Journalism Practice 4(3):297–308.

[Hutwagner et al. 2003] Hutwagner, M. L.; Thompson, M. W.; Seeman, G. M.; and Treadwell, T. 2003. The bioterrorism preparedness and response early aberration reporting system (ears). Journal of Urban Health 80(1):i89–i96.

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

[Leetaru et al. 2013] Leetaru, K.; Wang, S.; Cao, G.; Padmanabhan, A.; and Shook, E. 2013. Mapping the global twitter heartbeat: The geography of twitter. First Monday 18(5).

[Li and Cardie 2013] Li, J., and Cardie, C. 2013. Early stage influenza detection from twitter. arXiv preprint arXiv:1309.7340.

[Long et al. 2011] Long, R.; Wang, H.; Chen, Y.; Jin, O.; and Yu, Y. 2011. Towards effective event detection, tracking and summarization on microblog data. In Web-Age Information Management. Springer. 652–663.

[Osborne et al. 2014] Osborne, M.; Moran, S.; McCreadie, R.; Von Lunen, A.; Sykora, M. D.; Cano, E.; Ireson, N.; Macdonald, C.; Ounis, I.; He, Y.; et al. 2014. Real-time detection, tracking, and monitoring of automatically discovered events in social media.

[Petrović, Osborne, and Lavrenko 2010] Petrović, S.; Osborne, M.; and Lavrenko, V. 2010. Streaming first story detection with application to twitter. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 181–189. Association for Computational Linguistics.

Table 4: Example cases and the terms extracted for them: top terms selected either by TNT or LDA.

| ID          | TNT Terms                          | LDA Terms                      |
|-------------|------------------------------------|--------------------------------|
| JONO-23-02  | joy, enjoy                         | enjoy, glad, loss               |
| ASL-02-04   | asthma, air pollution, smog, pollution | asthma, smog, pollution, attack air |
| VOL-20-04   | vomit, chocolate, easter           | chocolate, eaten, easter, vomit, headache |
| SAL-11-08   | sadness, robin williams, sad news, robin, williams | sad, robin, williams, rip, ripobinwilliams |

11:438–441.
[Public Health England 2014] Public Health England. 2014. Gp in hours bulletin 2014. https://www.gov.uk/government/publications/gp-in-hours-bulletin. Accessed: 2014-08-04.

[Sadilek, Kautz, and Silenzio 2012] Sadilek, A.; Kautz, H. A.; and Silenzio, V. 2012. Predicting disease transmission from geo-tagged micro-blog data. In AAAI.

[Sakaki, Okazaki, and Matsuo 2010] Sakaki, T.; Okazaki, M.; and Matsuo, Y. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In Proceedings of the 19th international conference on World wide web, 851–860. ACM.

[Shaver et al. 1987] Shaver, P.; Schwartz, J.; Kirson, D.; and O’connor, C. 1987. Emotion knowledge: further exploration of a prototype approach. Journal of personality and social psychology 52(6):1061.

[Thelwall et al. 2010] Thelwall, M.; Buckley, K.; Paltoglou, G.; Cai, D.; and Kappas, A. 2010. Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology 61(12):2544–2558.

[Walther and Kaiser 2013] Walther, M., and Kaiser, M. 2013. Geo-spatial event detection in the twitter stream. In Advances in Information Retrieval. Springer. 356–367.

[Zubiaga et al. 2012] Zubiaga, A.; Spina, D.; Amigó, E.; and Gonzalo, J. 2012. Towards real-time summarization of scheduled events from twitter streams. In Proceedings of the 23rd ACM conference on Hypertext and social media, 319–320. ACM.