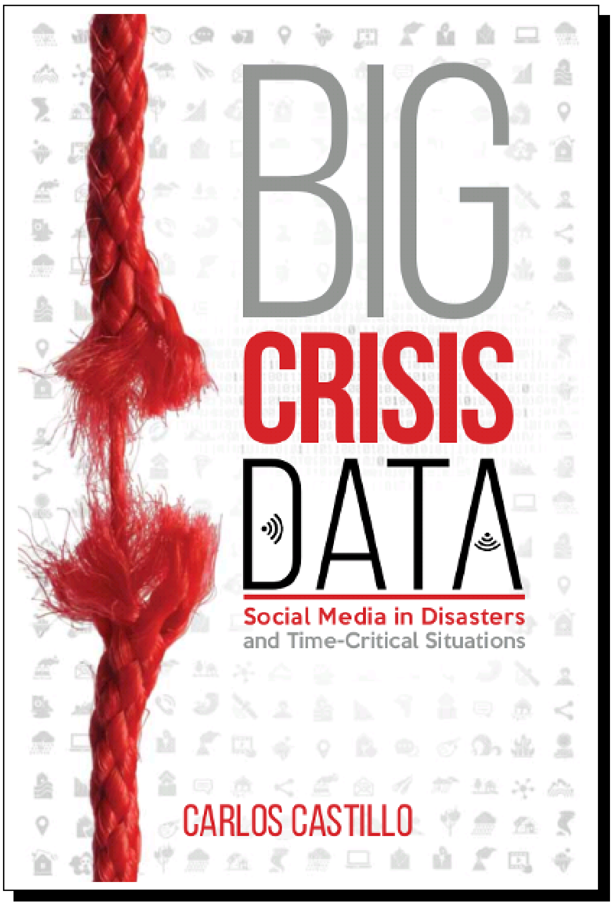
Big Crisis Data

Social Media in Disasters and Time Critical Situations

by Carlos Castillo



FREE PREVIEW

CHAPTER 6. VELOCITY

Cite as: Carlos Castillo: Big Crisis Data. Cambridge University Press, 2016

BibTex:

```
@book{castillo2016big,
      title="Big Crisis Data",
      author="Castillo, Carlos",
      publisher="Cambridge University Press"
      year=2016
}
```

Contents of Chapter 6

6

Velocity: Online Methods and Data Streams

One of the main reasons why social media is relevant for emergency response is because of its *immediacy*. For instance, the first reports on social media about the 2011 Utøya attacks in Norway appeared 12 minutes before the first news report in mainstream media (Perng et al., 2013), and in the 2013 Westgate mall attacks, social media reports appeared within a minute after the attack started, "scooping" mainstream media by more than half an hour.[1] People on the ground can collect and disseminate time-critical information, as well as data for disaster reconnaissance that otherwise would be lost due to the gap between a disaster and their arrival on site (Dashti et al., 2014).

On a lighter note, it has been speculated, jokingly but plausibly, that the damaging seismic waves from an earthquake, traveling at a mere three to five kilometers per second, can be overtaken by social media messages about them, which propagate orders of magnitude faster through airwaves and optical fiber.[2]

In this context, it is not surprising that people who associate social media with immediacy also expect a fast response from governments and other organizations, for instance, expecting help to arrive within a few hours of posting a message on social media (American Red Cross, 2012). Independently of whether those expectations are met or not in the near future, some capacity for rapid response to social media messages needs to be developed.

We recall from Section 1.5 that our main requirements are to create aggregate summaries about broad groups of messages (capturing the "big picture"), and to detect important events that require attention or action (offering "actionable insights"). We now add a new requirement: timeliness.

This chapter describes methods that ensure that the output summaries or insights are generated shortly after the input information required to create them becomes available. The way to achieve this low-latency or real-time data processing is to adapt a computing paradigm known as *online processing*, or equivalently, to consider that the input data is not a static object, but a continuously flowing *data stream*.

We begin by explaining how online processing differs from offline processing (§6.1), and present high-level operations on temporal data (§6.2). Then, we describe the framework of event detection (§6.3) and methods for finding events and subevents (§6.4). We also introduce the approach of incremental update summarization (§6.5), and end with a discussion of domain-specific approaches (§6.6).

6.1 Stream Processing

Computer algorithms can be divided into two broad classes: offline algorithms and online algorithms. In the context of crisis computing, they can be used to perform retrospective data analysis, live data analysis, or incremental data analysis.

- *Retrospective data analysis* uses an offline algorithm to process a batch of data relevant to an event, for instance, an archive containing social media messages posted during a certain time span. An example of a retrospective data analysis application could be to reconstruct a timeline of important events occurring during the first 48 hours of a disaster.
- *Live data analysis* uses an online algorithm to process a stream of data relevant to an event. Data is collected

[1] "How Useful Is A Tweet? A review of the first tweets from the Westgate Mall Attack." Nanjira Sambuli, *iHub Research*, October 2013. http://community.ihub.co.ke/blogs/16012/how-useful-is-a-tweet-a-review-of-the-first-tweets-of-the-westgate-attack.
[2] "Seismic Waves." Randall Munroe, *XKCD comic* #723. April 2010. https://xkcd.com/723/.

Table 6.1. *Comparison of retrospective data processing and live data processing.*

	Retrospective Processing	Live Processing
Algorithmic setting	Offline	Online
Data acquisition	Download	Stream
Data selection	Search	Filter
Temporal context	Complete	Past only
Storage requirements	Prop. to data size	Bounded
Main benefit	Accuracy	Immediacy

through a push/subscription/live API (see Section 2.2), and arrives after a delay in the order of a few seconds (i.e., with low latency), or a few hundred milliseconds (i.e., in "real-time"). An example of a live data analysis application could be to generate alerts of important events in a disaster as the disaster unfolds.

- *Incremental data analysis* lies somewhere in between retrospective and live data analysis. These methods often rely on algorithms that are run at regular intervals (e.g., every few minutes, every hour, or every day), processing small batches of data and keeping some memory/state in-between runs.

The trade-off between retrospective and live data can be described at a high level as a problem of accuracy versus latency. Intuitively, live data analysis is more difficult than retrospective data analysis, because we do not have the benefit of hindsight – algorithms are restricted to operate only on information from the past. We can acquire more information by waiting more, but waiting too long may not be desirable or acceptable in certain scenarios or for certain users. For instance, emergency managers would prefer a shorter wait in order to respond to a situation as it unfolds; forensic analysts would be more willing to wait until a situation is fully resolved to obtain a more accurate picture of what happened. Table 6.1 compares retrospective data processing and live data processing from a high-level perspective. For development, historical data is typically used to *simulate* live data (e.g., Guo et al., 2013a; Aslam et al., 2013).

Online algorithms on complex data streams. In computing, an *online algorithm* is a series of discrete operations executed over discrete pieces of data, where the whole input is not available from the start. Online algorithms that operate on large data streams also follow the *streaming model of computation*, in which it is assumed that every item in the stream is seen only once, and it is not possible to store all the items in memory.

Large systems performing streaming computation often use *event-driven architectures*. In an event-driven architecture, a system receives external events (social media messages in our case), and passes them through a series of modules. Each module can, in turn, generate events to be processed by other modules or to be produced as output.

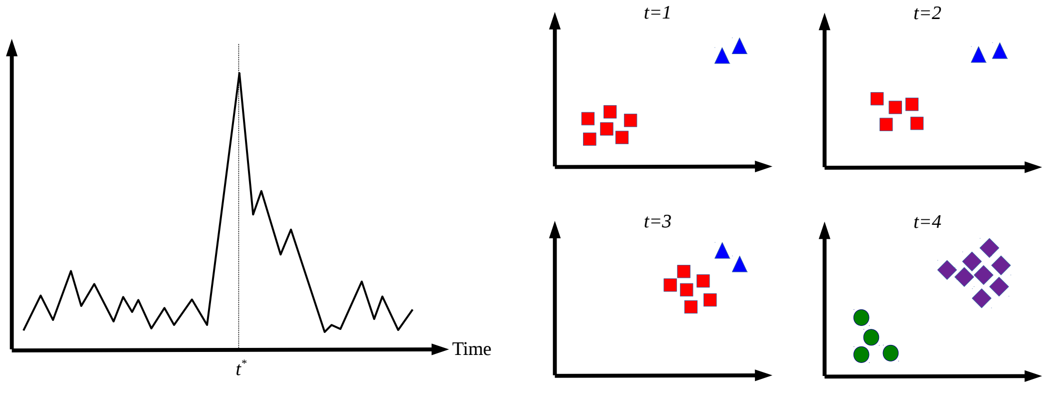
Additionally, computational systems in the emergency response space typically perform *complex event processing*, meaning that instead of processing a homogeneous stream of events from a single source, they must be able to process heterogeneous events from a variety of sources.

6.2 Analyzing Temporal Data

The analysis of temporal data seeks to find temporal patterns in the data. In general, the kinds of patterns that can be described depend on the dimensionality of the data, as depicted in Figure 6.1.

Figure 6.1(a) depicts a *time series* – a sequence of observations in time – for a single scalar variable. The X axis represents the time, while the Y axis describes a quantity of interest. In crisis-related messages, this variable could be, for instance, the number of mentions of a certain keyword (e.g., the word "injured") or the number of messages that are classified within a certain category (e.g., messages reporting injured people). The operation on this series that will be our main focus of interest is the detection of *events*, which intuitively correspond to significant changes. In the figure, it can be argued that an event has occured at $t = t^*$.

Figure 6.1(b) depicts a two-dimensional time series at different moments in time, which could be minutes, hours, or days apart. In this series, objects are represented by two numerical values, which may correspond to different dimensions of the message after applying a topic modeling method (e.g., using LDA, see Section 4.3). This can be interpreted as the extent to which two topics are expressed in each message. Points that are close to



(a) One dimension (b) Two dimensions

Figure 6.1. Illustration of time series analysis in one dimension and multiple dimensions (two in the figure). In the one-dimensional case, we are depicting an event happening at time t^*. In the two-dimensional case, we are depicting two clusters (squares and triangles) at $t = 1, 2, 3$ that merge at $t = 4$ (diamonds), plus a new, emerging cluster at $t = 4$ (circles).

each other should correspond to messages having similar characteristics. The multidimensional case is analogous to the two-dimensional case.

In Figure 6.1(b), a clustering method has been applied at every time step, and clusters at different time steps have been associated to each other by using symbols: squares, triangles, diamonds and discs. For instance, while the six messages posted at $t = 1$ and depicted by a square are not the same as the five messages posted at $t = 2$, they are both in the same region of the figure in similar time intervals, so we assume they correspond to the same theme or topic. The operations of interest in this series are: to detect topics at each timestep, and to track those topics by mapping them across time steps, including noticing if they merge, disappear, or split.

Temporal Information Retrieval. The study of collections of documents having timestamps or other temporal information, is known as *Temporal Information Retrieval*. Traditionally, these studies used news articles as a data source; currently, many other types of documents are analyzed, including Web pages and postings in social media. Temporal Information Retrieval is more challenging in social media than in other document collections. Social media streams are larger and arrive at a faster rate than documents in other collections, and include short and unstructured content which is quite different from traditional news articles.

In general, the process of tracking how events develop and unfold over time in a timestamped document collection is known as *Topic Detection and Tracking* (TDT). Topic detection and tracking includes various techniques such as story segmentation, topic detection, new event detection, and topic tracking. Story segmentation is usually applied to continuous broadcasts such as newscasts on radio or television; it aims at automatically determining when a news story ends and the next story begins. Topic detection aims at grouping news articles into coherent stories. New event detection decides if an article is part of a new story or belongs to an already reported story. Topic tracking follows how an event unfolds over time. For a survey of topic detection and tracking, see Allan (2002).

6.3 Event Detection

An event is the occurrence of something significant at a specific time and in a specific location (Brants et al., 2003). Important events in social media are often characterized by an increase in the volume of messages associated to a specific topic, and/or to specific entities such as certain people or places (Dou et al., 2012). These increases are sometimes described as “trending topics” or “bursts” of activity.

Types of event. While significant events cause social media activity to increase, not all increases of social media activity correspond to significant events: the volume of messages in social media can also increase for other reasons.

Crane and Sornette (2008) distinguish increases due to external causes (*exogenous*) and increases that are due to viral propagation of a piece of information originating within a social media platform itself (*endogenous*). Increases in activity due to exogenous causes are often described in the crisis computing literature as corresponding to "actual events" or "real-world events," for instance, an earthquake or a tornado. Increases in activity due to endogenous causes are in general not related to disasters. In Twitter, for instance, an endogenous event is the increase in frequency of a popular hashtag such as "*#musicmonday*," which is used to suggest music on Mondays, or "*#followfriday/#ff*," which are used to suggest people to follow on Fridays.

Events can be specified or unspecified (Atefeh and Khreich, 2013). The distinction is similar to the one between supervised and unsupervised approaches in machine learning. *Specified event detection* is concerned with detecting events of a specific type, for instance, an increase in messages about an infectious disease. *Unspecified event detection* is concerned with finding any kind of event in an input stream.

Events can be recurring or new. A *recurring event* is an event closely resembling a similar event observed in the past, such as the different games during a long sports tournament. A *new event* is different from past events (Yang et al., 2009), where this difference is quantified through a suitable metric (Kumaran et al., 2004).

Events and subevents. The word "event" for our purposes does not always mean a large-scale event, but sometimes a small-scale event or even a subevent of some larger situation.

Many disaster events, such as severe weather storms and earthquakes, can be detected and described accurately through meteorological observations and seismic sensors. In places having a good coverage of meteorological stations and seismic sensors, social media is probably not the first choice as a tool for detecting that one of these events has occurred. In these cases, event detection methods using social media may be used to complement existing sensor data (Musaev et al., 2014).

Crisis events spanning many hours or days include *subevents* of smaller scale. For instance, the 2012 shooting in Aurora, Colorado, in the United States, included at least the following subevents: (i) a shooting has taken place in a cinema, (ii) one person has been arrested, (iii) the arrested suspect has been identified, (iv) increased security measures have been taken in other cinemas (McCreadie et al., 2014). These are some of the subevents of interest that can be identified through automatic methods. In the following, we do not make a distinction between events and subevents except when necessary.

Describing events through multidocument summarization. After a system detects an event, it needs to describe it in a manner that makes sense to end-users. For instance, some organizations would like to receive event reports using a particular format or structure. Common elements across these reports include time, location, and a description of the event.

An heuristic that has been used for describing an event discovered in crisis-related social media, is to find named entities (Section 3.4), particularly locations (Mathioudakis and Koudas, 2010; Dou et al., 2012; Avvenuti et al., 2014). Other descriptors of an event obtained via information extraction include actors (people and organizations), actions, objects, and dates (Khurdiya et al., 2012; Nguyen et al., 2015).

The framework of *multidocument text summarization* provides a series of methods to generate a brief summary from a set of documents – in our case, from a set of messages. There are two basic methods for performing multidocument text summarization: *extracting* words or sentences from the input messages, and methods for *abstracting*, that is, synthesizing new sentences. A well-established method for extractive text summarization is to look for messages that are "central," that is, messages that are similar to many other messages. This heuristic is applied to the summarization of social media streams by Lee et al. (2013), among others. There are many methods for multidocument text summarization; for a survey, see Nenkova and McKeown (2011).

Delimiting phases/stages of a crisis. In addition to events, other time-sensitive elements of a crisis are its phases. Killian (2002, p. 51) describes four phases of emergency events: warning, impact, emergency, and recovery. Information needs are different in different phases. For instance, during the warning phase, the focus may be monitoring the situation, while, during the emergency phase, rescue and similar activities occur.

Event phases can also be observed in the social media response to a crisis. Supervised classification models can be used to classify messages according to the stages described earlier (Appling et al., 2014), or to a simplified version of it, such as *before*, *during*, and *after* an event (Iyengar et al., 2011). This can be of particular interest in the case of events that are not anticipated. The same classification can be attempted through lexical methods based on identifying discriminative words for each phase (Chowdhury et al., 2013).

6.4 Event-Detection Methods

Single-word frequency. A simple yet often effective method for detecting events is to assume that a sharp increase in the frequency of a word is indicative of an event. This method requires to maintain the frequency of words, such as counters of how many messages contain a given word in each discrete period of time. Typical periods of time can range from one hour to one day. An event is declared whenever the current counter for a word exceeds by a sufficient margin the previous counters, or some statistic computed from them, such as moving average or median. To avoid increasing indefinitely the memory usage over time, older counters can be discarded.

An example is the *TwitInfo* system by Marcus et al. (2011), which collects all tweets containing an input query (e.g., "earthquake"). The system maintains a historical average of the frequency of tweets per minute, and reports a new event whenever the current frequency is more than two standard deviations above the historical average frequency. Robinson et al. (2013) continuously monitor Twitter for tweets geotagged in Australia and New Zealand and keywords related to earthquakes (e.g., "*earthquake*" and "*#eqnz*"), and trigger an alert through the Emergency Situation Awareness (ESA) platform (Cameron et al., 2012) whenever the observed frequency exceeds a certain threshold. Earle et al. (2011) trigger alerts based on a statistic dependent on Short-Term Averages (STA) and Long-Term Averages (LTA) of frequency: $C(t) = \frac{STA}{m \times LTA + b}$ where $m > 1$ and $b \geq 0$ are tunable parameters and an earthquake is declared whenever $C(t) > 1$.

Multiword frequency. A natural extension of methods based on the increase of frequency of a single word, is to consider groups of words. These methods also maintain per-word counters, that are then aggregated according to some similarity function between words. This similarity function can be computed based on time series correlations, or by other means, such as grouping together synonyms.

The *TwitterMonitor* system described by Mathioudakis and Koudas (2010) detects events by first finding individual words showing a sharp increase in frequency, and then by grouping together words by co-occurrence (i.e., if they frequently appear in the same messages). A variant of this method focuses on hashtags instead of general keywords (Corley et al., 2013).

This can be extended by creating a cross-correlation graph, in which each node is a frequently occurring keyword, phrase, or hashtag, and nodes are connected by weighted edges based on the cross-correlation of the time series of the keywords they represent (Sayyadi et al., 2009). Weng and Lee (2011) compute dense subgraphs in the cross-correlation graph, and detect an event whenever a relatively small subgraph exhibiting large cross-correlations among its words is found. The cross-correlation graph might be pruned to discard spurious words that are not related to an event, for instance, by applying k-core decomposition (Meladianos et al., 2015).

An alternative way of using the frequency of multiple terms to detect events is to look for changes in the distribution of frequencies. Events of high significance and disaster events tend to capture the conversation in social media, which exhibits more concentrated frequencies of hashtags and words (Kenett et al., 2014; Rudra et al., 2015, among others). In other words, these high-impact events are characterized by fewer hashtags and fewer words capturing a larger share of messages. Kenett et al. (2014) apply a concentration metric borrowed from economics, to evaluate the distribution of hashtags at different moments of time, and suggest to generate an alert when this concentration increases significantly.

Classification-based and clustering-based methods. These methods exploit the redundancy in crisis-related social media by grouping similar messages together, and reporting that a new event has occurred if: (i) a group of messages becomes "too large," or (ii) a message that is "unlike" any of the groups seen so far appears.

For the first case, a supervised classification method can be used to count the number of messages belonging to a particular class per unit of time. Significant peaks in this time series can be reported as new events (Avvenuti et al., 2014). Messages can also be classified in an unsupervised manner, clustered by applying an *offline clustering* method (e.g., LDA, see Section 4.3) on a set of recently seen documents. An event is reported whenever the number of messages per unit of time on one of these clusters exceeds a certain threshold, for instance, if it is larger by a certain margin measured in units of standard deviation of previous observations for that topic (Dou et al., 2012).

For the second case, it is common to use an *online clustering* method, that runs incrementally on each new message. One method for online clustering works as follows: a set of clusters is maintained, and every new message seen is compared to the current set of clusters, by performing a *nearest neighbor* search. If the message is similar enough to an existing cluster, it is added to it, and if it is not, a new cluster is created (Phuvipadawat and Murata, 2010). In this case, the creation of a new cluster indicates that something new, unlike what has seen before, can be a signal that an "event" has happened.

In addition to this nearest-neighbor based method, other clustering methods, such as density-based clustering (Lee et al., 2013), or self-organizing maps (Pohl et al., 2012) have been used for event detection in social media. In some cases, the clustering algorithms process specific aspects of the messages, instead of their entire text. Khurdiya et al. (2012) extract elements from messages such as actors, actions, dates, and locations; then similarity computation for clustering is done considering only these elements.

Regarding efficiency, a naïve implementation of an online clustering method can be very expensive, because every new item must be compared against many pre-existing items. A first idea to make this process more efficient is to maintain a cluster *centroid*, which is an hypothetical "average message" computed from the word vectors of the messages in a cluster. New elements are compared against this centroid, instead of against all elements in previous clusters (Becker et al., 2011). Another idea is to speed up the nearest neighbor search by reducing the dimensionality of the messages, by hashing them to a vector of a smaller dimension, and ensuring that two similar messages have similar hashes. This can be done by using LSH (Charikar, 2002), which has been applied to the detection of events and subevents in Twitter (Petrović et al., 2010; Rogstadius et al., 2013).

Spatial clustering corresponds to the identification of geographical areas exhibiting an abnormally high activity in social media during a relatively short time frame. Cheng and Wicks (2014) show that even without the content of the messages but merely by knowing that a message has been posted from a given location at a given time, one can detect events of interest. These events correspond to large clusters in a spatiotemporal sense, that is, large groups of messages posted from nearby places during a short time span. Chen and Roy (2009) note that, given the sparsity of geotagged messages, a spatial smoothing operation can be used to preprocess geographical information before event detection. Specifically, they apply the wavelet transform to transforms clouds of points into contiguous regions on a map.

Timeliness is an important consideration when performing online clustering, including the frequency-based and spatial cases. We are often more interested in clusters that represent recent messages, instead of clusters that represent all messages, where the exact definition of "recent" depends on the application (Aggarwal et al., 2003). In the case of social media messages during a crisis, and as expected when considering the different phases of a disaster situation (Baird, 2010), the topics that people speak about change as the crisis unfolds. In this scenario, it might be reasonable to consider that clusters reflecting the themes present in recent messages might be more relevant then clusters reflecting the complete set of messages posted during the entire crisis. Some event detection methods indeed incorporate the idea of demoting or removing older messages, or clusters that have not acquired new elements recently, from the computation of events (see, e.g., Lee et al., 2013).

Basic evaluation of event detection. In general, detection methods for crisis events in social media data are used to identify events that are *exogenous*, *specified*, and *new*. Their purpose often is to generate early alerts about changes in social media motivated by a real-world development of a specific type that has important and immediate consequences for affected populations.

Evaluating the quality of an event detection system is not trivial. For systems that trigger an alarm whenever they believe an event has happened, a standard way of performing this evaluation is in terms of sensitivity and false discovery rate. The *sensitivity* is the fraction of events that are detected, for example, if there are four events in an area, and the system produces an alarm for only one of them, then its sensitivity is $1/4 = 25\%$. The *false discovery rate* is the fraction of alarms that are incorrect,[3] for example, if a system generates five alarms, but only two of those correspond to actual events, the false discovery rate is $1 - 2/5 = 3/5 = 60\%$. Figure 6.2 illustrates the concepts of sensitivity and false discovery rate. Compare with Figure 4.1 depicting precision and recall, which to be applied to event detection would require a definition of negative elements, for instance, "a day without events."

There are trade-offs between these two aspects. Trivially, a system that triggers too many alerts may have good sensitivity but will also incur in a large false discovery rate. A system that triggers too few alerts will have poor sensitivity, but may also have a smaller false discovery rate. Usually a system allows for a number of choices in these parameters. The specific choice used depends on the costs associated to both types of errors – not triggering when we should, and triggering when we should not. These costs, in turn, depend on the actions that are taken as

[3] This is not the same as the false positive rate, which is the probability of triggering an alarm every time a nonevent happens, which requires discrete time steps at which event detection can happen, or a discretization of time. In contrast, the false discovery rate can be defined even for continuous time.

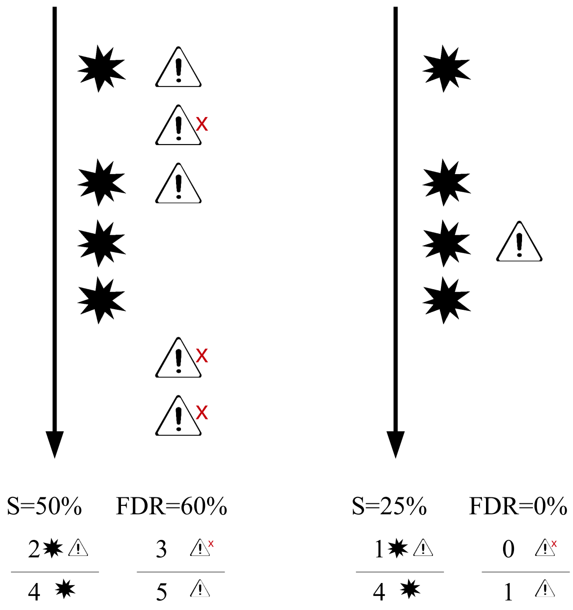


Figure 6.2. Visual depiction of sensitivity (S) and false discovery rate (FDR), comparing two hypothetical systems. The arrows represent time, the "explosions" are events, and the warning signs represent alarms issued. The system on the right has lower sensitivity, as it issues an alarm only in one of the four events. The system on the left is twice as sensitive, but it also issues three false alarms.

a consequence of the system's output. An evaluation of some event detection systems described on the literature according to these metrics is presented on Section 9.6.

6.5 Incremental Update Summarization

Current methods for temporal summarization aim at creating a *timeline* of information collected from a data stream (Aslam et al., 2013; Guo et al., 2013b) this is different from retrospective/offline text summarization.

An incremental update summarization method takes as input a stream of documents that are somehow topically related, for instance, they are received from a push API using a keyword-based query. Then, online processing or incremental data analysis is used to produce a sequence of relevant updates, in which each update contains information that is *relevant* and *novel* with respect to previous updates.

Example update summarization methods. Guo et al. (2013b) introduce a general online method for incremental update summarization. Every message that is received is broken down into sentences. Then, every sentence is scored individually using a regression model learned on historical data. Sentences which low scores are discarded. Each remaining sentence is considered a candidate update, and compared against previously issued updates. If the candidate contributes new information, it is written in the output as an update.

Incremental data analysis can also be used. McCreadie et al. (2014) perform incremental data analysis using fixed-size time intervals (e.g., one hour). From all the messages posted during each interval, a fixed number of diverse and representative sentences are extracted (e.g., ten sentences). These are candidate updates, which are filtered to remove those that are too similar to previously issued updates. Finally, a rank-cutoff method is used to decide how many of these updates are relevant enough to be included in the output, which consists of a variable number of updates per hour (from zero to ten).

Kedzie et al. (2015) also perform incremental data analysis using fixed-time intervals. In each interval, the salience (relevance) of all sentences is estimated using a regression model trained on manually annotated sentences. This model uses a series of features from the sentences, including superficial text features, query-specific features (such as the presence of the word "earthquake" and synonyms), geographical features, and temporal features. Next, a graph of sentence similarity is created for all sentences in the interval, and a variant of the Affinity Propagation (AP) clustering algorithm is used to group sentences (Frey and Dueck, 2007). The selection of pivots (exemplars) from each cluster is biased toward high-salience sentences. The generated sentences are candidate updates, which are then postprocessed to discard the ones that are too similar to previously issued updates, and to ensure a minimum degree of salience.

Rudra et al. (2015) also operate on fixed-time intervals. Their method first determines which messages in the interval contribute to situational awareness by using a supervised classification method. Then, it scores each word in the set according to its frequency, and generates a set of messages covering the highest-scoring content words. The total length of this set of messages is constrained, so the system tends to produce summaries containing few short messages.

Evaluating incremental updates methods. In addition to sensitivity and false discovery rate, metrics for evaluating systems for incremental update summarization include other elements such as verbosity, redundancy, and latency (Aslam et al., 2013; Kenter et al., 2015).

Aslam et al. (2013) describe the evaluation methodology used in the TREC 2013 Temporal Summarization track, in which participants must generate a stream of updates, corresponding to relevant subevents in a stream of documents. In their evaluation, a series of "nuggets" of information are extracted manually from each event as ground truth. Each information nugget is a piece of information that the system should generate, and it is associated to a time (the first time it appears in the stream) and a measure of importance (low, medium, or high). Under this evaluation framework, a good system should have high relevance (retrieve the high-importance nuggets), low verbosity (do not generate many updates, or updates that are too long and do not contain nuggets), no redundancy (do not repeat the same nugget again), and low latency (generate an update containing a nugget soon after the nugget's timestamp).

An evaluation framework can also consider multiple users that visit a system at different times. Concepts of latency and redundancy need to be modified in this case, because they have to be computed with respect to the other updates a user has seen since she logged in (Baruah et al., 2015).

Value tracking. Some quantifiable information about a disaster, for instance the number of people missing or displaced, cannot be known with 100% certainty as the disaster is unfolding (Neubig et al., 2011). These figures tend to be revised significantly as a crisis evolves, and hence require special processing to reflect the most recent – hopefully more accurate – estimate so far. The *value tracking* task, described in Aslam et al. (2013, sec. 2.2), corresponds to producing a sequence of estimations for a *specified* query (e.g., number of injured people), in which the estimation can be revised with each message that arrives. Experimentally, the evaluation of these systems is based on how close the estimation is to the actual value, which is gathered and validated after the fact.

Rudra et al. (2015) present a method to find *unspecified* quantities of interest. They apply a heuristic in which a POS tagger and a dependency parser are used to find numerals modifying a verb (e.g., "*67 missing*").

6.6 Domain-Specific Approaches

As in many applications involving noisy and ambiguous data, the application of domain-specific heuristics tends to improve the results obtained using open-domain methods. This is particularly applicable in our case given that activity in social media increases in response to various events that are not related to crises (as explained in Section 2.1).

Twitter-specific features. Heuristics based on observed behavior on a specific social media platform can be applied to improve the computation of candidate events or to postprocess them. For instance, when computing the similarity of two messages, Phuvipadawat and Murata (2010) boost the similarity of vector components corresponding to hashtags or usernames. In other words, hashtags in common between two messages are considered a stronger signal of similarity than keywords in common.

After applying an online clustering method, many spurious, nonevent clusters appear; Twitter-specific heuristics can be applied to remove such clusters. Becker et al. (2011) note that a high number of retweets typically signals an endogenous event, while a high percentage of user mentions is common during actual (exogenous) events. Similarly, multiword hashtags such as *#musicmonday* and *#followfriday* are also indicative of endogenous events. Phuvipadawat and Murata (2010) also postprocess the clusters and rank them according to certain heuristics, including whether they include names of people (extracted using a named entity tagger).

Crisis-specific features. Methods that exploit crisis-specific features often rely on a model to determine which are the clusters or candidate events of interest. This is typically a supervised model, created from messages posted in past events that have been deemed to be relevant. The type of model can be a simple lexical model (i.e., "a crisis event is one that contains a particular set of keywords") or a more complex statistical model, such as a machine-learned classification model (Section 4.2). At a high level, these methods scan social media for groups of crisis-relevant messages that are created approximately at the same time and approximately in the same location.

For instance, Sakaki et al. (2013) use a specific set of crisis-related keywords and a supervised classifier to find messages that are crisis-related, then use a statistical model of an event based on observations that are close by in time and space. This model is further refined in Sakai and Tamura (2015), and is designed based on observations from previous events, particularly from earthquakes. A similar method can be used to detect smaller-scale events, such as car crashes (Schulz et al., 2013). Rudra et al. (2015) also uses a supervised learning approach to filter messages of interest.

Crisis-specific features can also be used to rank updates or subevents. Verma et al. (2011) noted that the messages that can contribute to situational awareness tend to express facts more than opinions, and tend to be written in a formal, impersonal tone. These elements can be modeled computationally and combined to determine the probability that a sentence or message contributes to situational awareness. Li et al. (2012) consider a group of messages classified on a particular category and close by in time and space as a candidate event, and evaluate them as a set using a linear regression model. The model generates an "importance" score for the candidate based on aspects computed from all the messages in a candidate set, such as the number of people participating, and the frequency of certain keywords such as "killed" or "death." The parameters of the model are learned on previously seen crises.

A further extension is to consider more than one type of event. *SaferCity* (Berlingerio et al., 2013) identifies several types of public safety events in social media, classified using a supervised classification method into various types such as "traffic accident" or "theft and burglary," and clustered using a spatio-temporal approach that incorporates information about the content of the messages.

Incorporating multiple sources. Multiple heterogeneous data feeds can be processed simultaneously, engaging in complex event processing. For instance, *SaferCity* (Berlingerio et al., 2013) is intended to incorporate data from both social media as well as traditional news media. *STED* (Hua et al., 2013) uses data from traditional news media to reduce the amount of manual labeling required over social media messages, by propagating labels from news articles to social media messages.

LITMUS (Musaev et al., 2014) detects landslides using data collected from multiple sources, including seismic sensors, rainfall data, and social media. Seismic data is obtained from the U.S. Geological Survey (USGS), rainfall data from NASA's Tropical Rainfall Measuring Mission (TRMM), and social media data from providers such as Twitter, YouTube, and Instagram.

6.7 Research Problems

Making predictions. A key aspect of time series analysis is *forecasting*, that is, making inferences about the behavior of a series in the future. In the crisis domain, even events that cannot be in general forecasted, such as a spontaneous demonstration or protest, can be anticipated to some extent by observing signals in social media and other data sources (Ramakrishnan et al., 2014).

Preserving privacy while mining. Recent studies have analyzed call detail records during emergencies, finding that there are anomalous patterns that are well correlated with crisis events such as an explosion in a factory (Aleissa et al., 2014) or a bombing attack (Young et al., 2014). Currently these approaches require access to private data that is only accessible to mobile telephony companies. A privacy-preserving data mining approach (Aggarwal and Philip, 2008) could be used to analyze this data and detect events without compromising the privacy of users (more on privacy on Section 11.1).

Performing effective text summarization at a scale. A general problem of dealing with crisis data from a system perspective is dealing with drastic variations in the flow of data. Ideally, a system is able to detect minor but significative changes in the flow of messages, while at the same time being able to cope with huge surges of several orders of magnitude, which are not uncommon. In particular, many algorithms for text summarization require a large number of comparisons between sentences or messages, often quadratic, which makes them prohibitively expensive for social media crisis data.

Quantifying without classifying. Results from the task of *text quantification* suggest that methods willing to quantify the amount of messages belonging to a certain category do not necessarily need to be optimized for categorization accuracy. Instead, one could try to directly estimate the number of messages in the category of interest (Sebastiani, 2014; Gao and Sebastiani, 2015).

6.8 Further Reading

The textbook on models and algorithms on data streams by Aggarwal (2007, ch. 2 and 3) describes clustering and classification on data streams. Allan (2002) surveys several aspects of topic detection and tracking. Atefeh and Khreich (2013) survey methods for event detection in Twitter, including unsupervised, supervised, and hybrid approaches – the latter applies a supervised classifier to determine the messages of interest, and then a clustering algorithm to group them.

The TREC Temporal Summarization track,[4] which is a long-running conference and competition in Information Retrieval, is a good source to track research developments on current methods for update summarization (Aslam et al., 2013).

[4] TREC Temporal Summarization track: http://www.trec-ts.org/.

Bibliography

Aggarwal, Charu (ed). 2007. *Data streams: Models and algorithms*. Advances in Database Systems. Springer.

Aggarwal, Charu C, and Philip, S Yu. 2008. *A general survey of privacy-preserving data mining models and algorithms*. Springer.

Aggarwal, Charu C., Han, Jiawei, Wang, Jianyong, and Yu, Philip S. 2003 (Sept.). A framework for clustering evolving data streams. Pages 81–92 of: *Proceedings of 29th International Conference on Very Large Databases*, vol. 29. VLDB Endowment, Berlin, Germany.

Aleissa, Faisal, Alnasser, Riyadh, Almaatouq, Abdullah, Jamshaid, Kamran, Alhasoun, Fahad, Gonzalez, Marta, and Alfaris, Anas. 2014. Wired to connect: Analyzing human communication and information sharing behavior during extreme events. Pages 12–15 of: *KDD-LESI 2014: Proceedings of 1st KDD Workshop on Learning about Emergencies from Social Information at KDD'14*. New York, USA: ACM.

Allan, James. 2002. *Topic detection and tracking: Event-based information organization*. Springer.

American Red Cross. 2012 (August). *More americans using mobile apps in emergencies*. Online and phone survey.

Appling, Scott, Briscoe, Erica, Ediger, David, Poovey, Jason, and McColl, Rob. 2014. Deriving disaster-related information from social media. Pages 16–22 of: *KDD-LESI 2014: Proceedings of 1st KDD Workshop on Learning about Emergencies from Social Information at KDD'14*. New York, USA: ACM.

Aslam, Javed, Ekstrand-Abueg, Matthew, Pavlu, Virgil, Diaz, Fernado, and Sakai, Tetsuya. 2013 (Nov.). Overview of the TREC 2013 temporal summarization track. In: *Proceedings of 22nd Text Retrieval Conference (TREC)*.

Atefeh, Farzindar, and Khreich, Wael. 2013. A survey of techniques for event detection in Twitter. *Computational Intelligence*, 132–164.

Avvenuti, Marco, Cresci, Stefano, Marchetti, Andrea, Meletti, Carlo, and Tesconi, Maurizio. 2014 (Aug.). EARS (Earthquake Alert and Report System): A real time decision support system for earthquake crisis management. Pages 1749–1758 of: *Proceedings of 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, USA.

Baird, Malcolm E. 2010. *The "phases" of emergency management*. Tech. rept. Vanderbilt Center for Transportation Research (VECTOR).

Baruah, Gaurav, Smucker, Mark D., and Clarke, Charles LA. 2015. Evaluating streams of evolving news events. Pages 675–684 of: *Proceedings of 38th International Conference on Research and Development in Information Retrieval (SIGIR)*. Santiago, Chile: ACM.

Becker, Hila, Naaman, Mor, and Gravano, Luis. 2011. Beyond trending topics: Real-world event identification on Twitter. Pages 438–441 of: *Proceedings of 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*. Barcelona, Spain: ACM.

Berlingerio, Michele, Francesco, Calabrese, Giusy, Di Lorenzo, Xiaowen, Dong, Yiannis, Gkoufas, and Mavroeidis, Dimitrios. 2013. SaferCity: A system for detecting and analyzing incidents from social media. Pages 1077–1080 of: *Proceedings of 13th International Conference on Data Mining Workshops ICDMW*. Dallas, Texas, USA: IEEE.

Brants, Thorsten, Chen, Francine, and Farahat, Ayman. 2003 (July). A system for new event detection. Pages 330–337 of: *Proceedings of 26th International Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, Toronto, Canada.

Cameron, Mark A., Power, Robert, Robinson, Bella, and Yin, Jie. 2012. Emergency situation awareness from Twitter for crisis management. Pages 695–698 of: *Social Web for Disaster Management (SWDM), Companion: Proceedings of 21st International Conference on World Wide Web Conference (WWW)*. Lyon, France: ACM.

Charikar, Moses S. 2002. Similarity estimation techniques from rounding algorithms. Pages 380–388 of: *Proceedings of 34th Annual ACM Symposium on Theory of Computing*. Montreal, Quebec, Canada: ACM.

Chen, Ling, and Roy, Abhishek. 2009. Event detection from Flickr data through wavelet-based spatial analysis. Pages 523–532 of: *Proceedings of 18th ACM Conference on Information and Knowledge Management (CIKM)*. Hong Kong, China: ACM.

Cheng, Tao, and Wicks, Thomas. 2014. Event detection using Twitter: A spatio-temporal approach. *PLoS ONE*, **9**(6), e97807.

Chowdhury, Soudip Roy, Imran, Muhammad, Asghar, Muhammad Rizwan, Amer-Yahia, Sihem, and Castillo, Carlos. 2013. Tweet4act: Using incident-specific profiles for classifying crisis-related messages. In: *Proceedings of 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*. Baden Baden, Germany: ISCRAM.

Corley, Courtney D., Dowling, Chase, Rose, Stuart J., and McKenzie, Taylor. 2013. Social sensor analytics: Measuring phenomenology at scale. Pages 61–66 of: *Proceedings of IEEE International Conference on Intelligence and Security Informatics ISI*. Seattle, Washington, USA: IEEE.

Crane, Riley, and Sornette, Didier. 2008. Robust dynamic classes revealed by measuring the response function of a social system. *PNAS*, **105**(41), 15649–15653.

Dashti, Shideh, Palen, Leysia, Heris, Mehdi P., Anderson, Kenneth M., and Anderson, Scott. 2014. Supporting disaster reconnaissance with social media data: A design-oriented case study of the 2013 Colorado Floods. In: *Proceedings of 11th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*. University Park, Pennsylvania, USA: ISCRAM.

Dou, Wenwen, Wang, Xiaoyu, Skau, Drew, Ribarsky, William, and Zhou, Michelle X. 2012. Leadline: Interactive visual analysis of text data through event identification and exploration. Pages 93–102 of: *Proceedings of IEEE Conference on Visual Analytics Science and Technology VAST*. Seattle, Washington, USA: IEEE.

Earle, Paul S., Bowden, Daniel C., and Guy, Michelle. 2011. Twitter earthquake detection: Earthquake monitoring in a social world. *Annals of Geophysics*, **54**(6), 708–715.

Frey, Brendan J., and Dueck, Delbert. 2007. Clustering by passing messages between data points. *Science*, **315**(5814), 972–976.

Gao, Wei, and Sebastiani, Fabrizio. 2015 (Aug.). Tweet sentiment: From classification to quantification. In: *Conference on Advances in Social Networks Analysis and Mining (ASONAM)*.

Guo, Qi, Diaz, Fernando, and Yom-Tov, Elad. 2013a. Updating users about time critical events. Pages 483–494 of: *Advances in Information Retrieval*. Lecture Notes in Computer Science. Springer Berlin Heidelberg.

Guo, Qi, Diaz, Fernando, and Yom-Tov, Elad. 2013b. Updating users about time critical events. Pages 483–494 of: *Advances in Information Retrieval*. Springer.

Hua, Ting, Chen, Feng, Zhao, Liang, Lu, Chang-Tien, and Ramakrishnan, Naren. 2013 (Aug.). STED: Semi-supervised Targeted-interest Event detection in Twitter. Pages 1466–1469 of: *Proceedings of 19th ACM International Conference on Knowledge Discovery and Data Mining SIGKDD*. ACM, Chicago, Illinois, USA.

Iyengar, Akshaya, Finin, Tim, and Joshi, Anupam. 2011. Content-based prediction of temporal boundaries for events in Twitter. Pages 186–191 of: *Proceedings of IEEE 3rd International Conference on Privacy, Security, Risk and Trust PASSAT*. Boston, Massachusetts, USA: IEEE.

Kedzie, Chris, McKeown, Kathleen, and Diaz, Fernando. 2015 (July). Predicting salient updates for disaster summarization. In: *Proceedings of 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kenett, Dror Y., Morstatter, Fred, Stanley, H. Eugene, and Liu, Huan. 2014. Discovering social events through online attention. *PLoS ONE*, **9**(7), e102001.

Kenter, Tom, Balog, Krisztian, and de Rijke, Maarten. 2015. Evaluating document filtering systems over time. *Information Processing & Management*, 791–808.

Khurdiya, Arpit, Dey, Lipika, Mahajan, Diwakar, and Verma, Ishan. 2012. Extraction and compilation of events and sub-events from Twitter. Pages 504–508 of: *Proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology WI-IAT*. Macau, China: IEEE Computer Society.

Killian, Lewis M. 2002. Methods for disaster research: Unique or not? Pages 49–93 of: Stallings, Robert A. (ed), *Methods of Disaster Research*. Philadelphia, USA: Xlibris.

Kumaran, Girdhar, Allan, James, and McCallum, Andrew. 2004. *Classification models for new event detection*. Tech. rept. University of Massachusetts Amherst.

Lee, Pei, Lakshmanan, Laks VS., and Milios, Evangelos E. 2013. Event evolution tracking from streaming social posts. *arXiv preprint arXiv:1311.5978*, 13 pages.

Li, Rui, Lei, Kin Hou, Khadiwala, Ravi, and Chang, KC-C. 2012 (Apr.). Tedas: A Twitter-based event detection and analysis system. Pages 1273–1276 of: *Proceedings of 28th International Conference on Data Engineering ICDE*. IEEE, Washington DC, USA.

Marcus, Adam, Bernstein, Michael S., Badar, Osama, Karger, David R., Madden, Samuel, and Miller, Robert C. 2011. Twitinfo: Aggregating and visualizing microblogs for event exploration. Pages 227–236 of: *Proceedings of Conference on Human Factors in Computing Systems (SIGCHI)*. Vancouver, Canada: ACM.

Mathioudakis, Michael, and Koudas, Nick. 2010. TwitterMonitor: Trend detection over the Twitter stream. Pages 1155–1158 of: *Proceedings of International Conference on Management of Data (SIGMOD)*. Indianapolis, USA: ACM, for ACM.

McCreadie, Richard, Macdonald, Craig, and Ounis, Iadh. 2014. Incremental update summarization: Adaptive sentence selection based on prevalence and novelty. Pages 301–310 of: *Proceedings of 23rd ACM International Conference on Conference on Information and Knowledge Management*. Shanghai, China: ACM, for ACM.

Meladianos, Polykarpos, Nikolentzos, Giannis, Rousseau, François, Stavrakas, Yannis, and Vazirgiannis, Michalis. 2015 (May). Degeneracy-based real-time sub-event detection in Twitter stream. In: *Proceedings of 9th International AAAI Conference on Web and Social Media (ICWSM)*.

Musaev, Aibek, Wang, De, and Pu, Calton. 2014. LITMUS: Landslide Detection by Integrating Multiple Sources. In: *Proceedings of 11th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*. University Park, Pennsylvania, USA: ISCRAM.

Nenkova, Ani, and McKeown, Kathleen. 2011. *Automatic summarization*. Vol. 5. Now Publishers.

Neubig, Graham, Matsubayashi, Yuichiroh, Hagiwara, Masato, and Murakami, Koji. 2011. Safety information mining-what can nlp do in a disaster. Pages 965–973 of: *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP)*, vol. 11. Chiang Mai, Thailand: ACL.

Nguyen, Minh-Tien, Kitamoto, Asanobu, and Nguyen, Tri-Thanh. 2015. TSum4act: A framework for retrieving and summarizing actionable tweets during a disaster for reaction. Pages 64–75 of: *Advances in Knowledge Discovery and Data Mining*. Springer.

Perng, Sung-Yueh, Büscher, Monika, Wood, Lisa, Halvorsrud, Ragnhild, Stiso, Michael, Ramirez, Leonardo, and Al-Akkad, Amro. 2013. Peripheral response: Microblogging during the 22/7/2011 Norway attacks. *International Journal of Information Systems for Crisis Response and Management*, **5**(1), 41–57.

Petrović, Saša, Osborne, Miles, and Lavrenko, Victor. 2010. Streaming first story detection with application to Twitter. Pages 181–189 of: *Proceedings of 48th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Phuvipadawat, Swit, and Murata, Tsuyoshi. 2010. Breaking news detection and tracking in Twitter. Pages 120–123 of: *Proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology WI-IAT*, vol. 3. Toronto, Canada: IEEE, for IEEE.

Pohl, Daniela, Bouchachia, Abdelhamid, and Hellwagner, Hermann. 2012. Automatic identification of crisis-related subevents using clustering. Pages 333–338 of: *Proceedings of 11th International Conference on Machine Learning and Applications (ICMLA)*, vol. 2. Boca Raton, Florida, USA: IEEE, for IEEE.

Ramakrishnan, Naren, Butler, Patrick, Muthiah, Sathappan, Self, Nathan, Khandpur, Rupinder, Saraf, Parang, Wang, Wei, Cadena, Jose, Vullikanti, Anil, Korkmaz, Gizem, Kuhlman, Chris, Marathe, Achla, Zhao, Liang, Hua, Ting, Chen, Feng, Lu, Chang Tien, Huang, Bert, Srinivasan, Aravind, Trinh, Khoa, Getoor, Lise, Katz, Graham, Doyle, Andy, Ackermann, Chris, Zavorin, Ilya, Ford, Jim, Summers, Kristen, Fayed, Youssef, Arredondo, Jaime, Gupta, Dipak, and Mares, David. 2014. "beating the news" with EMBERS: Forecasting civil unrest using open source indicators. Pages 1799–1808 of: *Proceedings of 19th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. New York, USA: ACM.

Robinson, Bella, Power, Robert, and Cameron, Mark. 2013. A sensitive Twitter earthquake detector. Pages 999–1002 of: *Social Web for Disaster Management (SWDM), Companion: Proceedings of 22nd International Conference on World Wide Web Conference (WWW)*. Rio de Janeiro, Brazil: ACM, for IW3C2.

Rogstadius, Jakob, Vukovic, Maja, Teixeira, Claudio, Kostakos, Vassilis, Karapanos, Evangelos, and Laredo, Jim Alain. 2013. Crisistracker: Crowdsourced social media curation for disaster awareness. *IBM Journal of Research and Development*, **57**(5), 4–1.

Rudra, Koustav, Ghosh, Subham, Ganguly, Niloy, Goyal, Pawan, and Ghosh, Saptarshi. 2015 (Oct.). Extracting situational information from microblogs during disaster events: A classification-summarization approach. In: *Proceedings of Conference on Information and Knowledge Management (CIKM)*.

Sakai, Tatsuhiro, and Tamura, Keiichi. 2015. Real-time analysis application for identifying bursty local areas related to emergency topics. *SpringerPlus*, **4**(1), 1–17.

Sakaki, Takeshi, Okazaki, Masahide, and Matsuo, Yoshikazu. 2013. Tweet analysis for real-time event detection and earthquake reporting system development. *Knowledge and Data Engineering, IEEE Transactions on*, **25**(4), 919–931.

Sayyadi, Hassan, Hurst, Matthew, and Maykov, Alexey. 2009. Event detection and tracking in social streams. Pages 311–314 of: *Proceedings of 3rd International AAAI Conference on Weblogs and Social Media (ICWSM)*. San Jose, California, USA: AAAI Press.

Schulz, Axel, Ristoski, Petar, and Paulheim, Heiko. 2013. I see a car crash: Real-time detection of small scale incidents in microblogs. Pages 22–33 of: *The Semantic Web: ESWC 2013 Satellite Events*. Springer.

Sebastiani, Fabrizio. 2014. Text quantification. Pages 819–822 of: de Rijke, Maarten, Kenter, Tom, de Vries, ArjenP., Zhai, ChengXiang, de Jong, Franciska, Radinsky, Kira, and Hofmann, Katja (eds), *Advances in Information Retrieval*. Lecture Notes in Computer Science, vol. 8416. Springer International Publishing.

Verma, Sudha, Vieweg, Sarah, Corvey, William J, Palen, Leysia, Martin, James H, Palmer, Martha, Schram, Aaron, and Anderson, Kenneth Mark. 2011 (July). Natural language processing to the rescue? extracting "situational awareness" tweets during mass emergency. In: *Proceedings of 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*.

Weng, Jianshu, and Lee, Bu-Sung. 2011. Event detection in Twitter. Pages 401–408 of: *Proceedings of 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*. Barcelona, Spain: AAAI Press.

Yang, Christopher C., Shi, Xiaodong, and Wei, Chih-Ping. 2009. Discovering event evolution graphs from news corpora. *IEEE Transactions on Systems, Man and Cybernetics*, 850–863.

Young, William Chad, Blumenstock, Joshua E., Fox, Emily B., and McCormick, Tyler H. 2014. Detecting and classifying anomalous behavior in spatiotemporal network data. Pages 29–33 of: *KDD-LESI 2014: Proceedings of 1st KDD Workshop on Learning about Emergencies from Social Information at KDD'14*. New York, USA: ACM.