Using Social Media to Enhance Emergency Situation Awareness

Jie Yin, *CSIRO ICT Centre*

Andrew Lampert, *Palantir Technologies*

Mark Cameron, Bella Robinson, and Robert Power, *CSIRO ICT Centre*

The described system uses natural language processing and data mining techniques to extract situation awareness information from Twitter messages generated during various disasters and crises.

Situation awareness is "the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future."[1] This definition suggests that establishing situation awareness requires three different levels of activity: *perception*, *comprehension*, and *projection*. Enabling situation awareness in a given environment thus relies on being able to identify an appropriate set of perception elements, coupled with higher-level comprehension patterns and forecast operators. Although it initially surfaced as a concept in the military domain, situation awareness has been studied across a wide range of domains for both individual and team activities. Significantly, it's been recognized as a critical part of making successful and effective decisions for emergency response.[2,3]

In recent years, social media has emerged as a popular medium for providing new sources of information and rapid communications, particularly during natural disasters. Twitter is one such service that allows users to broadcast short textual messages, or *tweets*, of up to 140 characters to an audience of followers using Web- or mobile-based platforms. An important characteristic of Twitter is its real-time nature. Users frequently post what they're doing and thinking about and repeatedly return to the site to see what other people are doing. This generates numerous user updates from which we can find useful information related to real-world events—including natural disasters such as earthquakes, bushfires, and cyclones.[4,5]

This growing use of social media during crises offers new information sources from which the right authorities can enhance emergency situation awareness. Survivors in the impacted areas can report on-the-ground information about what they're seeing, hearing, and experiencing during natural disasters. People from surrounding areas can provide nearly real-time observations about disaster scenes, such as aerial images and photos. This is particularly useful during severe emergency situations, in which people within blackout areas would experience limited communication ability. By leveraging the public's collective intelligence, emergency authorities could better

1541-1672/12/$31.00 © 2012 IEEE
Published by the IEEE Computer Society

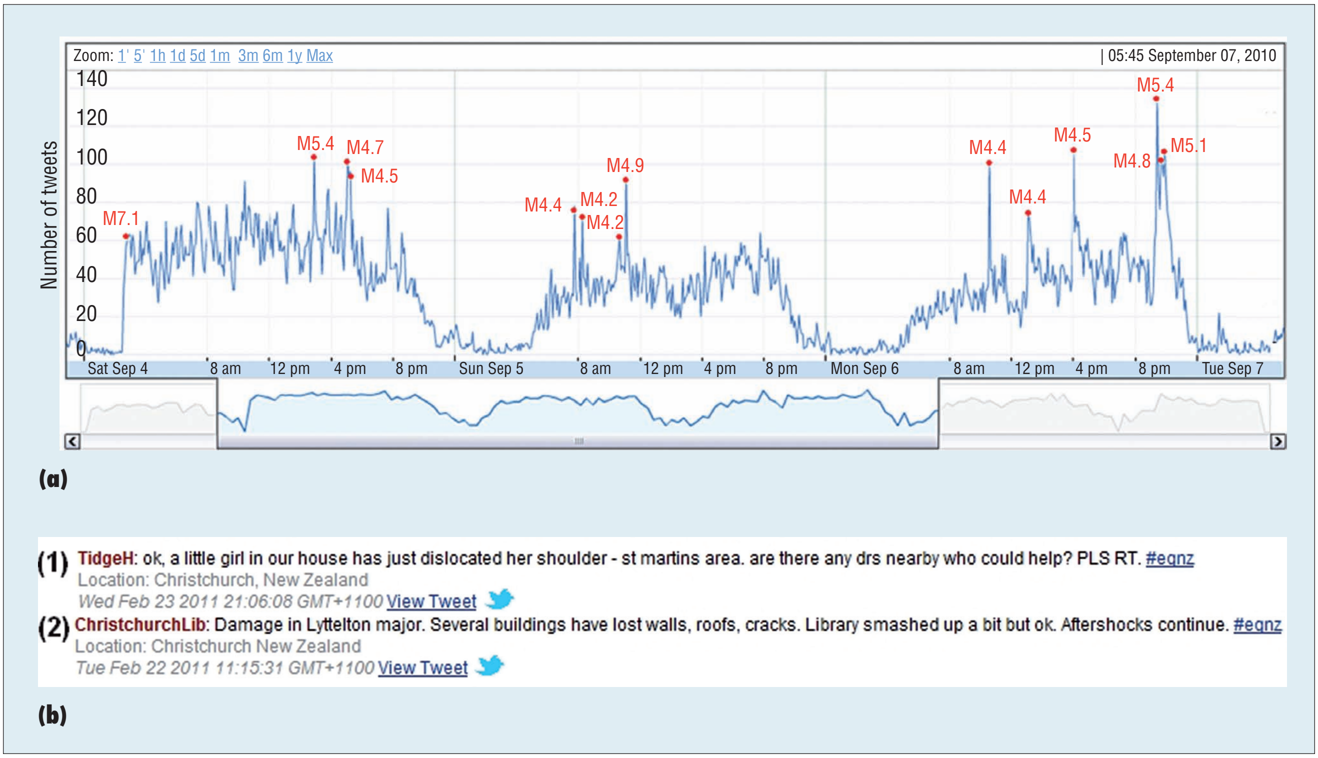

Figure 1. Using social media (such as Twitter) during the Christchurch, New Zealand, earthquakes. (a) Correlation of Twitter traffic and the September 2010 earthquake and its aftershocks. The *x*-axis denotes the date and time, and the *y*-axis denotes the number of tweets in a 5-minute period. Each red dot represents a spike in the number of tweets, and the associated label indicates the magnitude of the earthquake or aftershock on the Richter scale. (b) We can see tweets representing (1) a request for help and (2) an infrastructure status report of damage, both from Christchurch shortly after the earthquake in February 2011.

understand "the big picture" during critical situations, and thus make the best, most informed decisions possible for deploying aid, rescue, and recovery operations.

Here, we present a system architecture for leveraging social media to enhance emergency situation awareness. It differs from existing systems[2,3] in that the data sources are high-speed text streams retrieved from Twitter during natural disasters and crises. These text streams provide important situation awareness information, such as community responses to emergency warnings, near-real-time notification of incidents, and first-hand reports of an incident's impact. Such information, if extracted and analyzed properly and rapidly, can effectively contribute to enhancing the perception level of situation awareness.

Motivation

Because of its growing ubiquity, communication rapidity, and cross-platform accessibility, Twitter is increasingly being considered as a means for emergency communication during and after natural disasters.[6] In most urban areas, different types of networks, such as fixed-line, Wi-Fi, cellular, and WiMax, can provide overlapping coverage for Internet connectivity. So, during times of crises, when a certain type of telecommunication infrastructure is destroyed, people can still use other means to keep in touch via social media. As reported by Craig Fugate (the administrator of the US Federal Emergency Management Agency)[7] with regard to the catastrophic 2010 Haiti earthquake, even when an area's physical infrastructure was completely destroyed, the cellular tower bounced back quickly, allowing survivors to request help from local first responders and emergency managers to relay important disaster-related information via social media sites.

We've seen strong evidence of this by capturing Twitter data during several natural disasters, such as the earthquakes in Christchurch, New Zealand, in September 2010 and February 2011. Figure 1a illustrates the correlation between peaks in the volume of tweets that people in Christchurch posted and the magnitude of the September 2010 earthquake and its aftershocks. The *x*-axis denotes the time and date, and the *y*-axis denotes the number of tweets in a 5-minute period. The red dots indicate when an aftershock of 4.2 magnitude or stronger hit Christchurch, and they correlate with a spike in the number of tweets that people posted. This illustrates that when earthquakes or aftershocks occurred, people

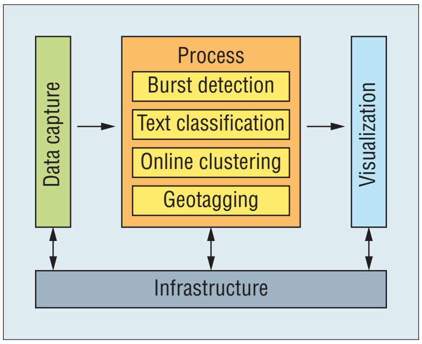

Figure 2. Architecture for emergency situation awareness. Key components include burst detection, text classification, online clustering, and geotagging, along with visualization interfaces for incident exploration.

actively broadcast information, wishes, and other messages on Twitter. Manually inspecting these tweets confirms that in the hours after the earthquake, Twitter users local to the crisis were providing on-the-ground information, including expressions of fear, requests for help, and the disaster's impact on the community. Figure 1b shows tweets indicating a request for help and an infrastructure status report of damage.

Crisis Coordination and Emergency Response

To better understand how to extract emergency situation awareness information from social media, we worked with the Australian government's recently established Crisis Coordination Centre. The CCC is a dedicated 24/7 facility that supports a whole-of-government response to national security and natural disaster incidents. It's responsible for hazard monitoring and situation awareness, and for the timely and accurate dissemination of information on emerging risks and threats to government ministers, police, emergency services, and other agencies.

Our work with the CCC focuses on how best to provide watch officers with additional, near-real-time situation awareness information drawn from high-volume social media text streams. A significant motivation stems from the Royal Commission on Australia's 2009 Victorian bushfires (www.royalcommission.vic.gov.au). The commission heard evidence that situation awareness information was reported in near-real-time on social networking and blog sites but wasn't visible to state or federal crisis coordination teams. So, our work aims to assist CCC watch officers in gathering such information from social media to improve emergency management and crisis coordination.

The CCC regularly experiences multiple common modes of operation:

- *A quiet day at the office*, which is the most frequent mode, given that emergency events are expected to occur infrequently.
- *Urgent emergency response,* which requires gathering, verifying, coordinating, and rapidly disseminating information to relevant government ministers and agencies.
- *Issue management*, which focuses on exploring and analyzing the details and impacts of an identified incident.

Significant challenges face watch officers in performing these tasks. First, officers must continually monitor a large amount of high-volume social media streams to maintain situation awareness for potential incidents. Second, the content published on social media is intrinsically noisy and arrives at a high rate, making it difficult for watch officers to manually monitor and analyze such texts. Third, watch officers are typically time constrained, whereas the information they're seeking is both time critical and infrequent in text streams.

Designing an intelligent system can thus help watch officers more effectively identify situation awareness information of operational and strategical relevance from the large information space of social media within the time constraints.

System Architecture

Social media brings new challenges about how to sift relevant information from the sheer volume of data being broadcast over time. User-generated content is intrinsically noisy and embodies language uses that are markedly different from conventional documents, which makes traditional natural language processing techniques inapplicable. To deal with these difficulties, we developed a coherent set of integrated components for extracting situation awareness by using various data mining techniques, including burst detection, text classification, online clustering, and geotagging. We adapted and optimized these techniques to deal with real-time, high-volume text streams, which provide capabilities that include identifying early indicators of unexpected incidents, exploring the impact of identified incidents, and monitoring incidents' evolution. Figure 2 shows our high-level system architecture.

The *data capture* component manages the system's reliable access to Twitter messages using the available streaming and search APIs. It gathers raw tweets and forwards them to the *process* component, which processes the tweets via various methods, including burst detection, text classification, online clustering, and geotagging. Finally, the results from the process component go to the *visualization* component for display to users. This component can display any combination of raw tweets with outputs from any of the processing methods (for example, groups of tweets clustered by topic or tweets placed on a map based on location information). Underlying these components is the *infrastructure* layer, which encapsulates shared libraries and low-level components for interacting with data from Twitter.

Data Capture

Using Twitter APIs, we've been capturing tweets for specific areas of interest within Australia and New Zealand since March 2010. Over this time period, we've captured on the order of 66 million tweets from approximately 2.51 million distinct Twitter profiles that cover a range of natural disasters and security incidents, including

- tropical cyclone Ului (March 2010),
- the Brisbane storms (June 2010),
- the gunman in Melbourne (June 2010),
- the Christchurch earthquake (September 2010),
- the Qantas A380 incident (November 2010),
- the Brisbane floods (January 2011),
- tropical cyclone Yasi (February 2011), and
- another Christchurch earthquake (February 2011).

Our data capture module uses the Twitter API for search and stream captures. The challenge for stream capture is to obtain tweets relevant to incidents of interest. Because tracking from the stream feed delivers tweets from all over the world, not only those of interest in a locality, we mainly use Twitter's location-based search API to provide a feed of tweets from people within a region of interest.

Burst Detection for Unexpected Incidents

To identify unexpected incidents, we developed a burst-detection module that continuously monitors a Twitter feed and raises an alert for immediate attention when it detects an unexpected incident. To achieve real-time efficiency, we adopt a parameter-free algorithm[8] to identify bursty words from Twitter text streams in our system. The basic idea is to determine whether a word is bursty on the basis of its probability distribution in a time window. Specifically, we compute the probability of the number of tweets that contain the word f_j in the time window W_i, denoted as $P(n_{i,j})$, using a binomial distribution as follows:

$$P(n_{i,j}) = \binom{N}{n_{i,j}} p_j^{n_{i,j}} (1-p_j)^{N-n_{i,j}}, \quad (1)$$

where N is the number of tweets in a time window. Note that, although the number of tweets N_i in each time window might be different, we can rescale this number in all time windows by adjusting word frequencies, such that all N_i become the same; thus, we don't consider N as a parameter in the method.

In Equation 1, p_j is the expected probability of the tweets that contain the word f_j in a random time window and is thus the average of the observed probability of f_j in all time windows containing f_j:

$$p_j = \frac{1}{L}\sum_{i=0}^{L} P_o(n_{i,j}), \quad (2)$$

$$P_o(n_{i,j}) = \frac{n_{i,j}}{N}, \quad (3)$$

where L is the number of time windows containing f_j.

We determine whether a word f_j is bursty by comparing the actual probability $P_o(n_{i,j})$ that the word f_j occurs in the time window W_i against the expected probability p_j of the word f_j occurring in a random window. If $P_o(n_{i,j})$ is noticeably higher than the expected probability of the word $f_j(p_j)$, this indicates that f_j exhibits an abnormal behavior in W_i, and we consider f_j as a bursty feature in W_i.

In our implementation, we used a training set of around 30 million tweets captured between June and September 2010. We preprocessed the tweets by removing stop words and stemming words, which resulted in a set of roughly 2.6 million distinct features, based on which we built our background alert model. In the online phase, we devised an alerting scheme that evaluates a sliding 5-minute window of features against the alert model every minute.

For evaluation, we annotated roughly 2,400 features in a six-month Twitter dataset that we collected in 2010. We define an actual burst as one feature that suddenly occurs frequently in a time window and whose occurrence lasts more than 1 minute. We evaluate our burst-detection module using two commonly used metrics: detection rate and false-alarm rate. We compute the detection rate as the ratio of the number of correctly detected bursty features to the total number of actual bursty features, and the false-alarm rate as the ratio of the number of nonbursty features that are incorrectly detected as bursty features to the total number of nonbursty features. Our experimental results show that our burst-detection module achieves an overall detection rate of 72.13 percent and a false-alarm rate of 1.40 percent. For example, we can identify interesting bursts, such as `a380`, `earthquake`, and `cyclone`, after some real-world emergencies occurred.

Classification for Impact Assessment

In large-scale crises, understanding incidents' impact is critical to successfully restoring safety and recovering essential services. To support issue management for an incident, we provide tools to help identify high-value messages from Twitter. In our discussions with CCC staff, they highlighted a need to help them understand an incident's impact so that they could better plan their response. To address this need, we built statistical classifiers that automatically identify tweets containing information about the infrastructure status, where the infrastructure includes assets such as roads, bridges, railways, hospitals, airports, commercial and residential buildings, water, electricity, gas, and sewerage supplies.

Our training dataset consists of roughly 450 tweets posted during the February 2011 Christchurch earthquake that contain the `#eqnz` hashtag. We manually labeled each tweet with a binary annotation based on whether it contained information about the disaster's impact on infrastructure. As an example, the tweet in Figure 1c indicates the status of buildings and is thus labeled as a positive example of an infrastructure status tweet.

We experimented with two machine learning methods for tweet classification—naive Bayes and support vector machines (SVM), which both work well for text classification tasks.[9] To extract useful features, we preprocessed the dataset by removing a list of stop words and tokenizing the tweets. We then constructed lexical features and Twitter-specific features for classification. These features include

- word unigrams;
- word bigrams;
- word length;
- the number of hashtags "#" contained in a tweet;
- the number of user mentions, "@username";
- whether a tweet is retweeted; and
- whether a tweet is replied to by other users.

After feature extraction, we performed experiments using a 10-fold cross-validation over our training data. Initial results of this work have been promising: naive Bayes and SVM achieve classification accuracy of 86.2 percent and 87.50 percent, respectively, over a baseline result of roughly 60 percent using only word unigrams.

Online Clustering for Topic Discovery

To discover important topics from Twitter, we also developed an online incremental clustering algorithm that automatically groups similar tweets into topic clusters, so that each cluster corresponds to an event-specific topic. For this task, the desirable clustering algorithm should be scalable to handle the sheer volume of incoming tweets and not require a priori knowledge of the number of clusters, given that tweet contents are constantly evolving over time. So, partitional clustering algorithms such as k-means and expectation-maximization (EM)[10] aren't suitable for this problem, because they require the number of clusters as input. Hierarchical clustering algorithms are also inappropriate because they rely on a fully specified similarity matrix, which doesn't scale to our data's growing size.

To capture tweets' textual information, we represent each tweet using a vector of terms weighted using *term frequency* (TF) and *inverse document frequency* (IDF). Specifically, a tweet represents a data point in d-dimensional space, $V_i = (v_1, v_2, \ldots, v_d)$, where d is the size of the word vocabulary, and v_j is the TF-IDF weight of the jth word in tweet V_i.

We propose an online incremental clustering algorithm that extends the single-pass algorithm proposed elsewhere.[11] Given a Twitter stream in which the tweets are sorted according to their published time, the basic idea of incremental clustering is as follows. First, the algorithm takes the first tweet from the stream and uses it to form a cluster. Next, for each incoming tweet, T, the algorithm computes its similarity with any existing clusters. Let C be the cluster that has the maximum similarity with T. If $sim(T, C)$ is greater than a threshold δ, which is to be determined empirically, tweet T is added to the cluster C; otherwise, a new cluster is formed based on T. We define the function $sim(T, C)$ to be the similarity between tweet T and cluster C. In the clustering process, whenever a new tweet T is added to a cluster C, the centroid of C is updated as the normalized vector sum of all the tweets in C.

In our algorithm, we use two similarity measures: cosine similarity and Jaccard similarity. We define these as

$$sim_{\cos}(T_i, T_j) = \frac{v_i \cdot v_j}{\|v_i\| \times \|v_j\|}, \tag{4}$$

$$sim_{\text{jac}}(T_i, T_j) = \frac{|v_i \cap v_j|}{|v_i \cup v_j|}, \tag{5}$$

where $V_i \cdot V_j$ is the dot product of vector V_i and vector V_j. Here, $|V_i \cup V_j|$ denotes the number of distinct words either in tweet V_i or in V_j, and $|V_i \cap V_j|$ denotes the number of common words in both V_i and V_j.

To take into account the temporal dimension, we add another time factor to the similarity measure that favors a tweet to be added to the clusters whose time centroids are close to the tweet's publication time. So, we define our modified similarity measure as

$$\hat{sim}(T_i, T_j) = sim(T_i, T_j) \cdot e^{-\frac{(t_{T_i} - t_{T_j})^2}{2\sigma^2}}, \tag{6}$$

where t_{T_i} and t_{T_j} are the publication times of tweets T_i and T_j, respectively. The similarity measure depends not only on the similarity between the vectors of two tweets but also on the time distance between them.

Specifically, our clustering algorithm maintains a list of active clusters. Each cluster is represented by a centroid feature vector computed from the tweets that it contains, and a time centroid that is the average publication time of all the tweets forming the cluster. If no more tweets are added to a cluster for a period of time, which is determined based on application needs, the cluster is considered inactive and removed from the active list. The algorithm considers only those clusters in the active list as candidates to which a new tweet can be added.

Our clustering algorithm is efficient because it considers each tweet at once and thus can scale to a growing amount of Twitter messages. However, because of Twitter's noisy nature, our algorithm could lead to a large number of clusters, many of which might not correspond to events of interest. We overcome this problem by filtering out unimportant tweets using the burst-detection module and allowing only tweets that contain bursty features to form clusters. We thus dramatically reduce the number of clusters and only maintain a list of topic clusters associated with real-world events.

To evaluate the algorithm, we performed experiments on 3,500 tweets collected during the February 2011 Christchurch earthquake. We measure clustering quality using the Silhouette score,[12] which is a metric-independent measure designed to describe the ratio between cluster coherence and separation. Initial results show that using Jaccard similarity and cosine similarity, our clustering algorithm can achieve a Silhouette score of 0.42 and 0.34, respectively. This indicates that Jaccard similarity achieves higher clustering accuracy than cosine similarity. This might be because the TF-IDF vectors are very sparse owing to the tweets' limited length, and thus Jaccard similarity can better capture the similarity between tweets.

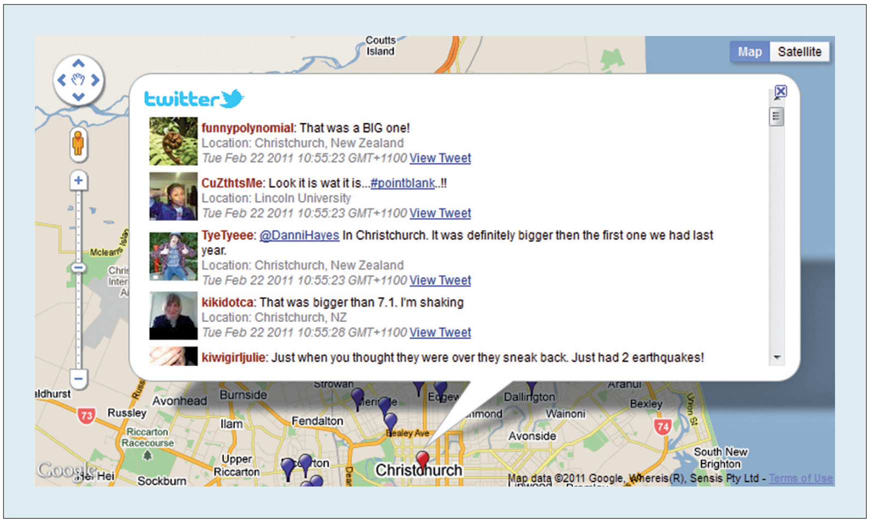

Figure 3. Geotagging tweets from the February 2011 Christchurch earthquake. The marker colors indicate the volume of tweets captured at a specific location, and viewers can click each marker to display recent tweets from that location.

Geotagging

To facilitate spatial exploration of tweets, we also developed a geotagging module that displays the content of a tweet at its geographic location on a map. We do this by using a tweet's coordinates if it's geotagged, or the location information from the user's profile. Specifically, if a tweet is geotagged, we display it at its latitude/longitude coordinates. Otherwise, we use the location field of the user profile to determine a latitude/longitude position. We first pass the location string to the Yahoo geocoding service (http://developer.yahoo.com/geo/placefinder/) and retrieve the top-five matches worldwide. We then select the most suitable one using state or country constraints. Figure 3 shows an example of geotagging tweets from the February 2011 earthquake. This figure displays the distribution of tweets that can be geotagged; the marker colors indicate the volume of tweets captured at a specific location. For further investigation, users can click each marker to display recent tweets from that location.

Visualization

To assist CCC watch officers in monitoring unexpected and known incidents, we developed a suite of visualization interfaces for exploring and interacting with the information our system generated as well as the raw data extracted from Twitter.

To explain our visualization tools, we use the September 2010 Christchurch earthquake and an incident involving a Qantas A380 airplane for illustration. The historical alert clustering tool can replay stored alerts, cluster tweets, and track the incident's evolution. As Figure 4 shows, the viewer has a component to identify a reference date-time and a slider to move the current time point within an hour interval. Our tool maps bursty features to different sizes and colors according to how statistically different the observed number of occurrences is with respect to the alert model. The viewer enables an operator to track bursty features and use them to seed topic clusters. The historical alert clustering tool can also display tweets belonging to a cluster or associated with a bursty feature.

On 4 September 2010, a magnitude 7.1 earthquake occurred at 04:35 NZDT (16:35, 3 September UTC) 40 km west of Christchurch

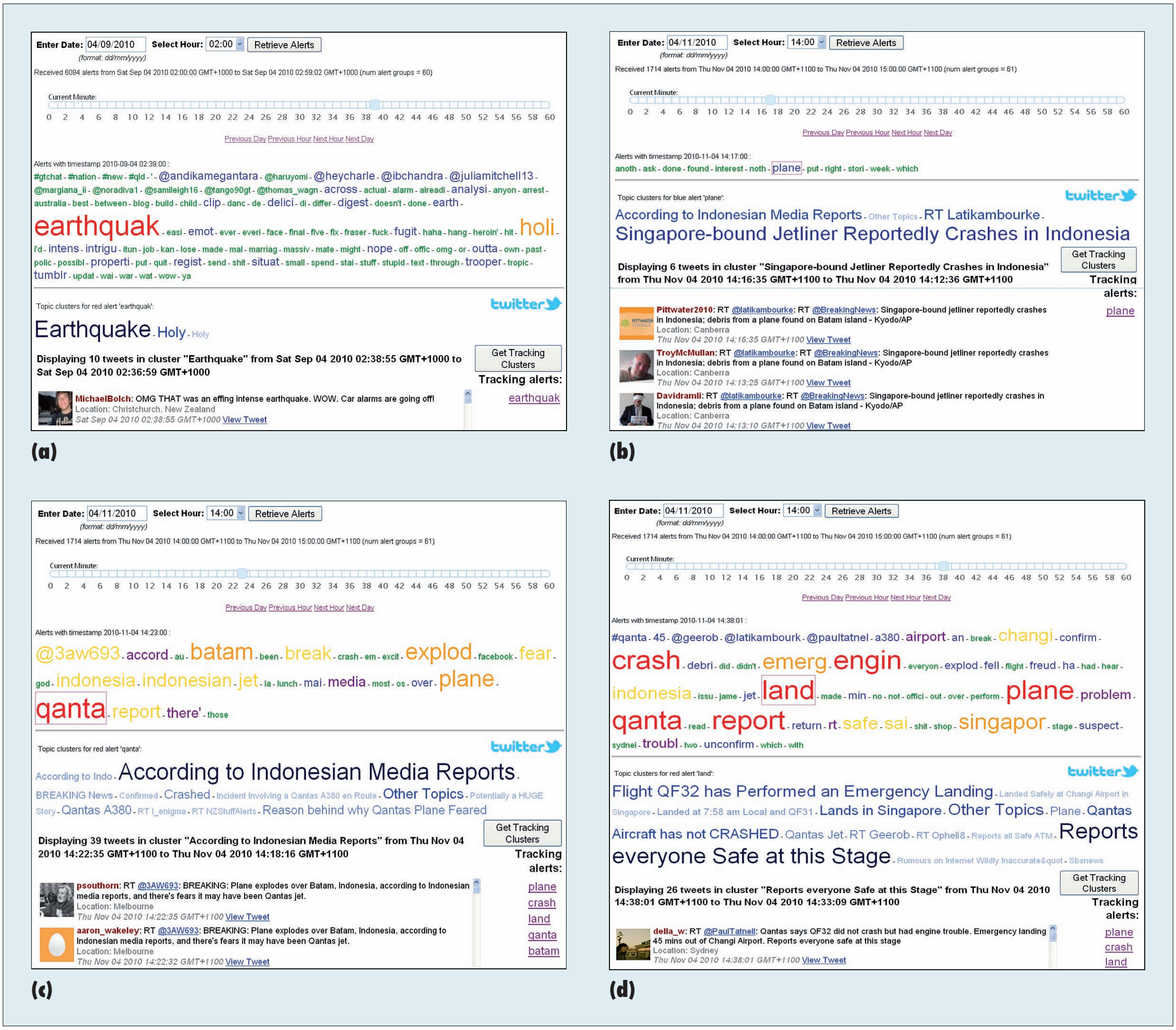

Figure 4. Visualizations for (a) the Christchurch 2010 earthquake and (b)–(d) the Qantas A380 incident. We can see the data evolving from the bursting of `plane` to the emergence of a topic cluster for `land`.

(http://en.wikipedia.org/wiki/2010_Canterbury_earthquake). Our first tweet containing `earthquake` was received at 04:36:59 NZDT, with 10 more tweets occurring within the next minute. Our burst detector identified the word stem `earthquak` as bursting at 04:39, as Figure 4a shows. Other word stems and hashtags occurring during the incident and the time at which they first appeared as bursts included `power` at 04:40, `#CHCHQuake` at 04:41, `#earthquake` at 04:42, and `#eqnz` at 05:13.

The Qantas A380 incident concerns an airplane's engine failure and subsequent emergency landing at Singapore's Changi Airport on 4 November 2010 (http://en.wikipedia.org/wiki/Qantas_Flight_32). The engine failure occurred at around 10:01 SGT (02:01 UTC). At 11:45 SGT, the crew safely landed the aircraft, but it took several hours to shut down another engine before passengers could disembark. This event is interesting in that the first reports of the incident were from Indonesian media reporting on debris falling on Batam. Our first tweet mentioning Qantas and Batam arrived at 14:02 SGT, and our burst detector identified `plane` as bursting at 14:17, soon followed by `batam` at 14:18 and `explod` at 14:19. As the incident unfolded, the topic clusters tracked many sides of the story, from the plane reportedly having crashed to it safely making an emergency landing in Singapore. Figures 4b through 4d show the data evolving from the bursting of `plane` to the emergence of a topic cluster for `land`.

Our proposed architecture can clearly provide useful situation awareness information through a set of tightly integrated components. It can thus provide on-the-ground information from the general public, as reported in Twitter, to help establish and enhance timely situation awareness across a range of crisis types.

In the future, we will conduct more experiments on large-scale datasets to evaluate our system's overall performance. We also plan to improve the performance of burst detection and tweet classification by using additional external resources to compensate for tweets' terseness. Finally, we will explore the use of smoothing techniques to tackle the data sparsity problem for better topic clustering.

Acknowledgments

The Australian government financially supports this project through the National Security Science and Technology Branch within the Department of the Prime Minister and Cabinet. We also appreciate the kind support of the Crisis Coordination Centre staff from the Department of the Attorney-General.

References

1. M.R. Endsley, "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors*, vol. 37, no. 1, 1995, pp. 32–64.
2. A. Blandford and B.L.W. Wong, "Situation Awareness in Emergency Medical Dispatch," *Int'l J. Human-Computer Studies*, vol. 61, no. 4, 2004, pp. 421–452.
3. R.G. Madey, G. Szabó, and A.-L. Barabási, "Wiper: The Integrated Wireless Phone-Based Emergency Response System," *Proc. Int'l Conf. Computational Science*, LNCS, 2006, pp. 417–424.
4. T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake Shakes Twitter Users: Real-Time Event Detection by Social Sensors," *Proc. 19th World Wide Web Conf.*, ACM, 2010, pp. 851–860.
5. S. Vieweg et al., "Microblogging during Two Natural Hazards Events: What Twitter May Contribute to Situational Awareness," *Proc. 28th Int'l Conf. Human Factors in Computing Systems* (CHI 10), ACM, 2010, pp. 1079–1088.
6. H. Gao, G. Barbier, and R. Goolsby, "Harnessing the Crowdsourcing Power of Social Media for Disaster Relief," *IEEE Intelligent Systems*, vol. 26, no. 3, 2011, pp. 10–14.
7. C. Fugate, *Five Years Later: An Assessment of the Post Katrina Emergency Management Reform Act*, tech. report, US Federal Emergency Management Agency, Oct. 2011.
8. G.P.C. Fung et al., "Parameter Free Bursty Events Detection in Text Streams," *Proc. 31st Int'l Conf. Very Large Databases* (VLDB 05), VLDB Endowment, 2005, pp. 181–192.
9. Y. Dang, Y. Zhang, and H. Chen, "A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews," *IEEE Intelligent Systems*, vol. 25, no. 4, 2010, pp. 46–53.
10. R. Xu and D.C. Wunsch II, "Survey of Clustering Algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, 2005, pp. 645–678.
11. Y. Yang, T. Pierce, and J. Carbonell, "A Study of Retrospective and Online Event Detection," *Proc. 21st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, ACM, 1998, pp. 28–36.
12. P.J. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," *J. Computational and Applied Mathematics*, vol. 20, no. 1, 1987, pp. 53–65.

THE AUTHORS

Jie Yin is a senior research scientist in the Information Engineering Laboratory at the CSIRO ICT Centre, Australia. Her main research interests include data mining and machine learning, and their applications to sensor-based activity recognition, pervasive computing, and social media analysis. Yin has a PhD in computer science from the Hong Kong University of Science and Technology. Contact her at jie.yin@csiro.au.

Andrew Lampert is a forward deployed engineer at Palantir Technologies and chief scientist for Cruxly, a company focused on actionable content detection in email text. His research focuses on conversational text analysis, particularly email communication. Lampert has an MS in speech and language from Macquarie University. He's a member of the Association for Computational Linguistics and serves as the secretary and treasurer of the Australasian Language Technology Association. Contact him at andrew@thoughtlets.org.

Mark Cameron is a research team and project leader at the CSIRO ICT Centre, Australia, where he leads a team developing a platform and tools to deliver situation awareness from near real-time social media text streams. His research interests include spatial databases, spatial information systems, spatial decision support systems, information integration, and Web-service composition planning. Cameron has a PhD in computer science and technology from Australian National University. Contact him at mark.cameron@csiro.au.

Bella Robinson is a senior software engineer at the CSIRO ICT Centre, Australia, where she works on developing tools to deliver situation awareness from near real-time social media text streams. Her research interests include intelligent transport systems, spatial information systems, scalable vector graphics, health data integration, and Web service integration. Robinson has a BS in information technology from James Cook University. Contact her at bella.robinson@csiro.au.

Robert Power is a senior software engineer at the CSIRO ICT Centre, Australia, where he investigates the use of social networks to aid emergency management, and Web interfaces to support disaster planning. His research interests include database systems, geospatial information systems, Web technologies, and software engineering. Power has a BS in computer science from the Australian National University. Contact him at robert.power@csiro.au.

cn *Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.*