Real-Time Crisis Mapping of Natural Disasters Using Social Media

Stuart E. Middleton, Lee Middleton, and Stefano Modafferi, *University of Southampton IT Innovation Centre*

The proposed social media crisis-mapping platform matches location data for areas at risk of natural disaster to geoparsed real-time tweet data streams, and uses statistical analysis to generate real-time crisis maps.

With the ubiquity of mobile communication devices, people experiencing natural disaster events can publish microblogs, images, and videos in real time on social media sites, such as Facebook, Twitter, and YouTube, often live and in situ. In the humanitarian sector, this has sparked great interest in developing innovative approaches to using social media for events such as earthquakes, floods, and tornadoes to both inform the public and assist civil protection authorities in focusing response efforts.[1]

During recent natural disaster events, such as the Haiti earthquake and Russian wildfires in 2010, Hurricane Sandy in 2012, and the 2013 tornado in Oklahoma, humanitarian organizations and networks of volunteers have set up live Web-based manual crisis mapping sites.[1] These organizations check and filter crowdsourced information from news reports, social media, and civil protection agency alerts, and present it live on Web-based crisis maps for the general public to see. Challenges for these organizations include automating the huge task of real-time data fusion of large volumes of multisource heterogeneous information and maintaining the trust and credibility of this data.[1]

We've developed a real-time crisis-mapping platform capable of geoparsing tweet content. Our approach exploits readily available location information from gazetteers, street maps, and volunteered geographic information (VGI) sources. Our goal is to improve geoparsing precision of street-level tweet incident reports and empirically quantify the accuracy of the resulting social media crisis maps during natural disaster events. To our knowledge, this is the first published analysis to directly compare street-level Twitter-based crisis maps to a verified ground truth based on post-event expert assessment. Such results can help disaster management agencies assess the value of social media crisis mapping.

Current Technology

Current real-time geospatial information systems (GIS) mostly map social media microblog reports using geotag metadata with longitude/latitude coordinates.[2] This approach turns social media into a crowdsourcing virtual sensor network, allowing maps of Twitter messages to be plotted. According to the US Geological Survey (USGS), the main

1541-1672/14/$31.00 © 2014 IEEE
Published by the IEEE Computer Society

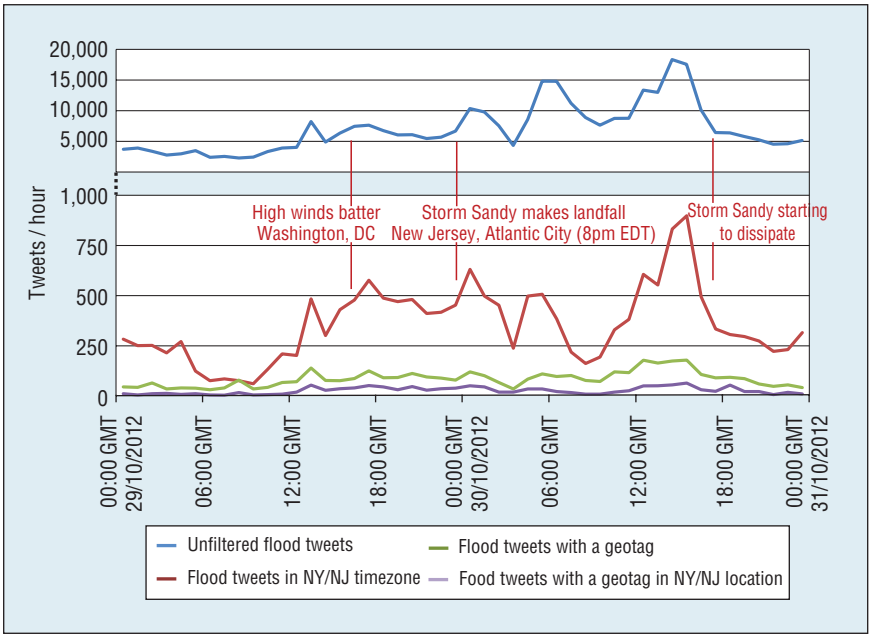


Figure 1. Twitter streaming API tweet traffic recorded using flood keywords over 48 hours as Hurricane Sandy made landfall between 29 and 30 October 2012. Peak tweet traffic was 18,000 tweets per hour, with 5 percent of tweets using the New York time zone, 1 percent of tweets containing a geotag, and 0.3 percent containing a geotag located in New York/New Jersey.

benefits of Twitter-based detection systems over sensor-based systems are their fast detection speed and low cost.[3] Social media GIS systems can be combined with conventional GIS systems deploying hardware-based sensors, such as in situ seismic sensors or remote sensing aerial photography and satellite imaging. Overall, the aim is to build a coherent situation assessment picture, and present it to emergency responders, civil protection authorities, and the general public to help coordinate response efforts and improve overall awareness.[3–5]

Unfortunately, only about 1 percent of all tweets actually contain geotag metadata. Of this 1 percent, the geotags are a mixture of genuine mobile devices (using GPS) and Twitter's default of the user's home location. In addition, the tweeted text can contain references to one or more locations geospatially distant from the location of the device sending the tweet. Although this doesn't matter when mapping course-grained earthquake regions, it does matter for finer-grained maps such as flood inundation or tornado damage. Figure 1 shows the breakdown of tweets recorded during 48 hours of Hurricane Sandy using our Twitter crawler. We've observed from our crawled tweet datasets that during events people tweet about flooding/damage to specific buildings, roads, and geographic features such as local parks, rivers, and beaches. Tweet reports are a mixture of a few first-hand reports and many retweets and comments on third-party incident reports.

Geoparsing systems can parse text documents to extract likely geographic tokens or "named entities" (for example, places or regions such as "New York").[6-8] When coupled with a geocoder, which can look up location names on a map and return the geotags, this provides a way to associate geotags with locations mentioned in microblog reports. Such systems often use *named entity recognition*. Using this technique, the text is first tokenized to extract sentences and words. Each token (word) is classified using a language-specific parts-of-speech (POS) classifier, identifying a lexical category (for example, ADJ for adjective, N for noun, and NP for proper noun). Lexical patterns can then be used to identify groups of tokens that are likely to refer to named entities. Challenges for named entity recognition include acquiring enough labeled training data, handling poorly structured text from sources like Twitter, and multilanguage scalability.[6]

Real-Time Crisis-Mapping Platform

We're interested in mapping real-time tweet flood reports for at-risk coastal areas near known geological fault lines that have the potential to cause a tsunami. Real-time monitoring is important because early wave-impact assessments can be used to warn people further along the coastline, allowing them to get to safety. Another key issue for decision makers in early warning control centers is keeping crisis map false alarm rates to a minimum, because this undermines credibility in the data source.[9]

To evaluate the accuracy of geoparsing of locations from Twitter data, we compare location matches from our platform against expert manual labels for tweet datasets covering disaster zones located in the US, New Zealand, Italy, and Turkey. To evaluate how social media crisis maps compare to expert impact assessments, we directly compare both our flood tweet crisis map for Hurricane Sandy in 2012 and our tornado tweet crisis map for the Oklahoma tornado in 2013, to official US National Geospatial Agency (NGA) post-disaster impact assessment maps compiled from verified satellite and aerial imagery.

Our system differs from existing crisis-mapping approaches in that we geoparse tweet text in real time rather than only using the tweet's geotag. Thus, we can access all crawled tweets

rather than just the 1 percent with a geotag. We avoid the need for language- and location-specific training sets by preloading available gazetteer, street map, and VGI data for areas at risk of disaster. This lets us work at a building- and street-level resolution as opposed to only working with higher-level administrative regions. Finally, we use statistical analysis techniques to identify a baseline noise signal and use this to reduce false positives in our crisis maps.

System Architecture

Figure 2 shows the architecture of our social media crisis mapping platform, which is split into sets of offline and real-time services. The offline services prepare a geospatial database for the local region of interest and calculate baseline statistics during a historical period when no disasters occurred. The real-time services crawl tweets live from Twitter, identifying mentions of known locations and displaying them as a live street- and/or region-level crisis map.

During the offline phase, we use a set of geospatial data-extraction tools to download geospatial data. We use OpenStreetMap to access street-level information, and the GooglePlaces API to access VGI such as buildings and local features. We use the global gazetteers Geonames and GEOnet Names (GNS), as well as local gazetteers, to get region names. This geospatial information is stored in a MySQL database, along with any OpenGIS shape data for later visualization on a map. When downloading geospatial data, we request the language native to the local area of interest.

A batch process geocodes each location's address, returning a well-formed address string and a specific coordinate on a map. We use the GoogleGeocoding API for our geocoding. For place data (such as buildings and rivers), geocoding lets us fill in blank address fields or correct errors or inconsistencies. We've found that building data uploaded by the general public varies greatly in its use of name, street, and address fields. For street data, geocoding parses the address field into subcomponents, providing us with short name variants in addition to the official road names. This is important because people often use short names or abbreviations in tweets.

The last offline step is to create baseline match statistics for each location in the database over a historical period when no disaster event occurred. Baseline match statistics are useful for reducing false positives associated with location names that pop up often in Twitter conversations (for example, "Hollywood"). This baseline is used as a threshold above which location matches can be considered relevant. An ingest tool is used to import the historical dataset to a MySQL database.

The real-time system is driven by a Twitter crawler tasked with a set of keywords. We use a set of European multilingual keywords for the event type we're looking to record (for example, for flooding we use "flood," "tsunami," "inondation," "sel," and "alluvione"). The TwitterStreaming API is used to receive tweets, which we store in our MySQL database, splitting SQL tables into one-month blocks to ensure a fast table query response.

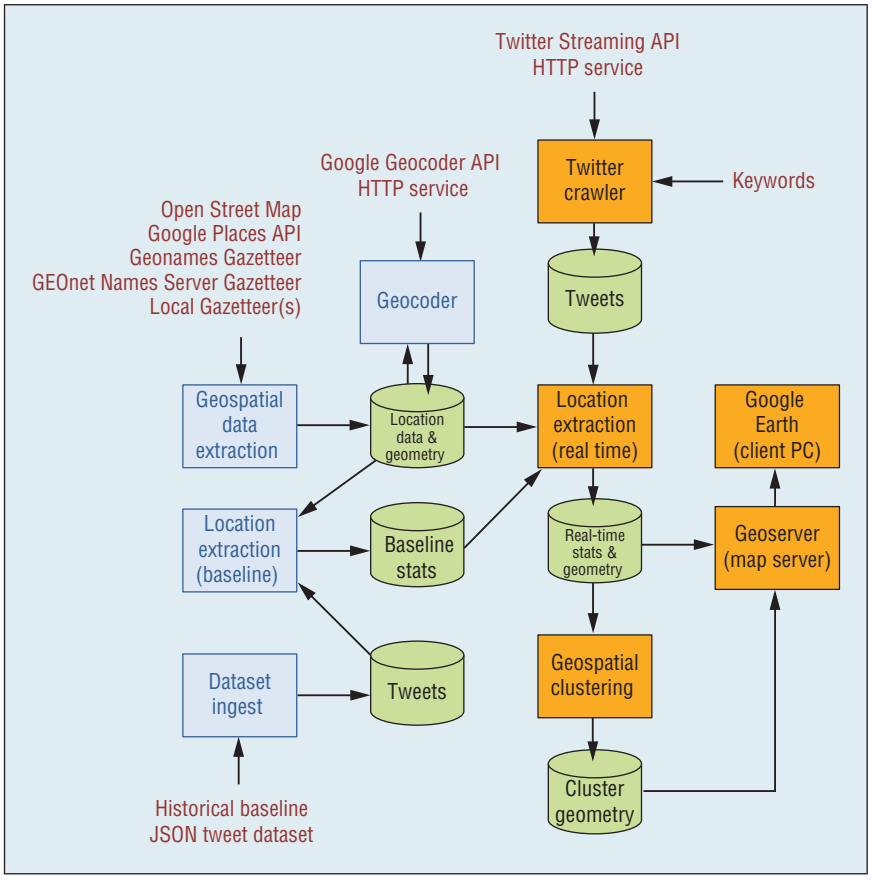


Figure 2. System architecture and information flow. Offline services (in blue) prepare a geospatial database for the local region of interest and calculate baseline statistics during a historical period when no disasters occurred. Real-time services (in orange) crawl tweets live from Twitter, identifying mentions of known locations and displaying them as a live street- and/or region-level crisis map. (JSON = JavaScript Object Notation.)

We use regex expressions to check for retweets, looking for prefixes such as RT, because Twitter retweet metadata is unreliable.

We filter tweets outside of the local region's time zone to help restrict our analysis to people located in the affected area, as opposed to people located in another state/country commenting on news reports. We also filter retweets, which usually don't report new information and thus tend to artificially inflate a location's frequency count.

Our real-time location-extraction service runs in parallel to the crawler, processing tweets as they arrive in the database. This service preloads locations for the spatial area of interest, tokenizes each of them, and creates an in-memory hash table of tokens ready for efficient real-time matching. Baseline statistics are also preloaded into memory. As new tweets are read from the database, they're cleaned, tokenized, and named entity matching is performed, matching location tokens to tweet text tokens. Location matches are logged to a rolling in-memory buffer of configurable size, usually between 6 and 24 hours long, which forms the basis of a rolling sample window. The sample period is usually between 1 and 5 minutes, ensuring that we have up-to-date statistics in the database for map display. All match statistics are saved to the database as soon as they're ready along with the OpenGIS geometry to plot on the crisis map.

We run a parallel geospatial clustering service to continuously cluster spatial areas of high activity and produce an easily visible polygon map overlay. This service applies a standard hierarchical clustering algorithm to compute clusters from location geometry.

The mapping visualization is performed using Geoserver, an open source map server. Map layers are driven from the geometry columns in MySQL database tables, plotting buildings (points), streets (lines), regions (points), and clusters of activity (polygons). We render our maps using Google Earth, although Geoserver supports a variety of mapping formats such as Web Mapping Service (WMS) and OpenLayers.

Geoparsing Tweets to Get Locations

Both our real-time analysis and offline baseline location extraction services use the same geoparsing algorithm. We support English, Italian, Portuguese, and Turkish, languages native to the coastal regions in the North Eastern Atlantic and Mediterranean (NEAM) regions of our tsunami early warning use case. Because we know a priori the spatial region of interest, we preload all possible location entities into an efficient in-memory lookup table. This avoids the need to use named entity-recognition approaches, such as a POS classifier coupled with a context grammar, to extract free text location phrases and then geocode them at runtime. Most online geocoding services, including the GoogleGeocoder API, have strict usage rate limitations, making geocoding on the fly impractical for the throughput of tweets we receive.

During system start-up, we take each location in our spatial region of interest and tokenize it into one-gram tokens using the Natural Language Toolkit's (NLTK)[10] Treebank word tokenizer, then compute a final *n*-gram token from a sequential combination of the 1-gram tokens. An *n*-gram token is a phrase made up of one to *N* words, in our case up to a maximum of five words. For example, the address "London Street" generates a two-gram token "london street." For buildings and street addresses, we use our own multilingual corpus of building and street types, along with common variants and abbreviations. This lets us expand token sets to include common usage variants of certain phrases. For example, "London Street" becomes "london street" + "london st."

We remove any tokens that match the NLTK's multilingual stopword list, holding words with low information value such as "the." We also remove tokens that match the NLTK's name corpus of common male and female names, avoiding false matches such as "Chelsea," which is both a location and a girl's name. We use weak token stemming to remove plurals, because locations are proper names and stronger stemming would cause false positives. We filter any place and address tokens that are identical to region names, because a region match is most likely in this case. We reject any one-gram token phrases for place and street names, because these tend to be ill-defined (for example, "station") and prone to overmatching. Lastly, we compute a one-gram hashtag token by removing all spaces, because hashtags are often used in tweets (for example, "#newyork").

During live real-time tweet processing, we remove URLs and email addresses from tweets that might generate false tokens. We then use the NLTK's Punkt sentence tokenizer before executing the Treebank word tokenization as before. We compute all sequential combinations of *n*-gram tokens from each tweet's text and use this as the basis for location token matching. Our location-match algorithm first checks for places' tokens, then streets, and finally regions. At each stage, we remove previously matched tokens from the tweet token pool to avoid text with street names such as "london street" being used to also match a region such as "london."

In performance testing, our location-extraction algorithm performs

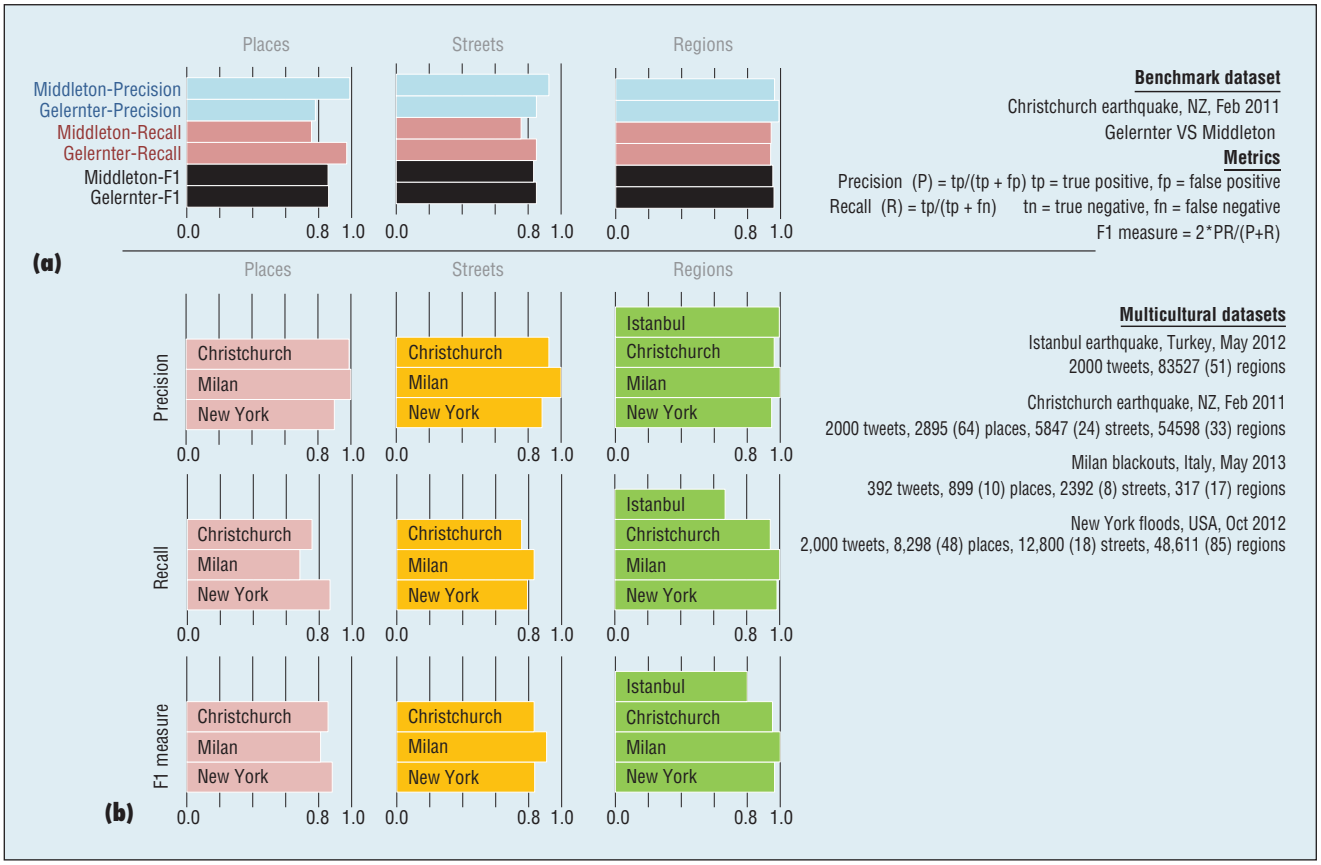


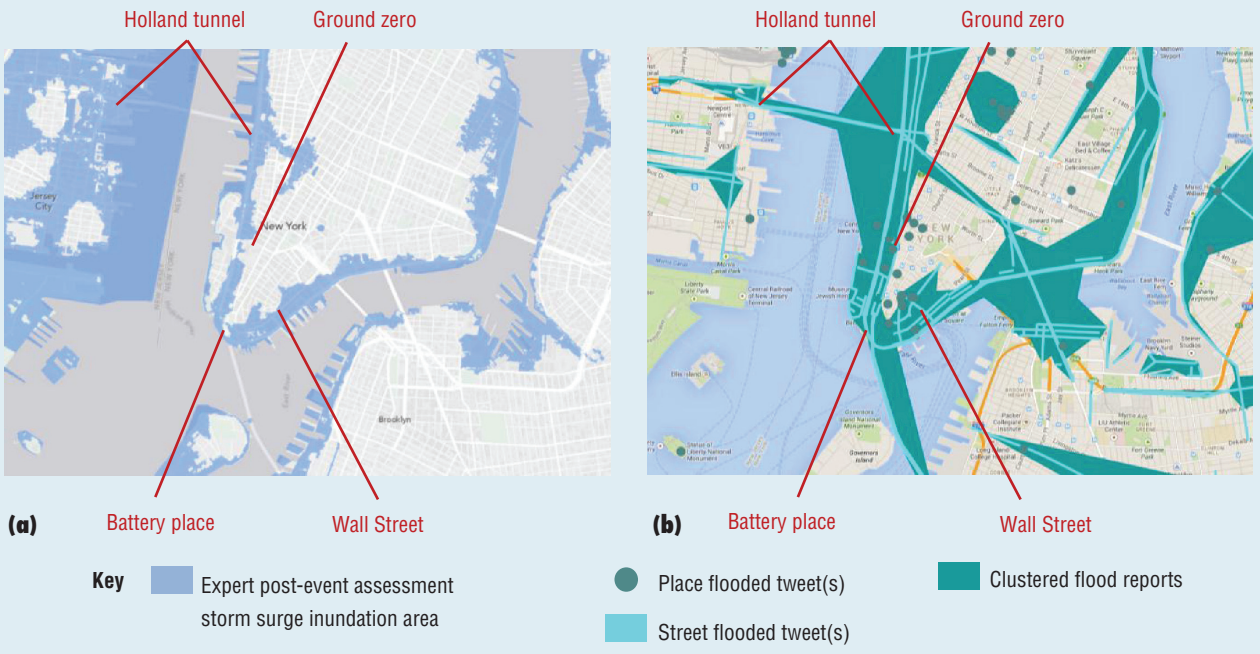
Figure 3. Geoparsing evaluation results. (a) Benchmark results for the February 2011 earthquake in Christchurch, New Zealand, comparing published[7] results to ours using the same 2,000-tweet labeled dataset. (b) Multicultural datasets show our results for a variety of European locations. Each dataset has a set of labeled tweets and a set of locations, of which a small subset of locations appeared in tweets (for example, the Istanbul database has 83,527 regions across Turkey, of which 51 were mentioned in tweets).

three times faster than the peak levels of tweet throughput found in our recorded datasets. The end-to-end processing speed, including all of the database I/O, was about 270,000 tweets per hour for a 20,000-location dataset on an 8-Gbyte RAM 2.5-GHz CPU laptop. Our performance scales much better than linearly as more locations are added to the database.

We first evaluated multicultural geoparsing accuracy on some tweet datasets recorded by our crawler over the last two years. These tweet datasets were manually annotated with places, streets, and regions. The datasets vary in native language and size of area affected, with localized blackouts in Milan and widespread floods in New York. The Istanbul earthquake caused no building or street damage, hence we only matched region labels. We counted true/false positives and negatives, where a true positive occurred if the matched location was the same as the expert label, and computed precision, recall, and F1 measures to evaluate the overall accuracy of our approach.

As the results in Figure 3b show, all locations reported a high matching precision, but the Turkish dataset had an unusually low recall for region data. This was largely due to how location names are written in Turkish grammar. For example, "izmir" is a Turkish location, but it might appear as "izmirda," "izmirdan," or "izmira," depending on whether someone is going to, from, or into a location. This result highlights potential limitations of our language-independent matching process.

We benchmarked the accuracy of our geoparsing on the well-studied 2011 earthquake in Christchurch, New Zealand. Our gold standard for comparison is a recently reported tweet-geoparsing system by Judith Gelernter and Shilpa Balaji, which is based on state-of-the-art language-specific named entity recognition, lexio-semantic heuristics, and a spell checker.[7] We tested our system using the same dataset of 2,000 manually labeled tweets, provided by Carnegie Mellon University, with annotations showing places, streets, and regions. We used the same experimental conditions as Judith Gelernter and Shilpa Balaji,[7] including the



Crisis map accuracy for an 8 x 8 map segmentation at different map thresholds

Threshold	Places	Streets	Precision	Recall	F1
dev_sma > 0	66	101	0.78	0.76	0.77
dev_sma > 0.0168	19	22	0.81	0.44	0.57
dev_sma > 0.1	4	1	1.00	0.09	0.17

Figure 4. Crisis map comparison for New York's 2012 flooding. (a) The ground truth post-event US National Geospatial Agency (NGA) impact assessment showing storm surge inundation. (b) A 5-day tweet flood map (dev_sma > 0) for tweets between 29 October and 2 November 2012. Red annotations show major incidents. (Source: Federal Emergency Management Agency [FEMA] Modeling Task Force [MOTF] storm Sandy impact analysis field-verified interim high-resolution report, November 2012. Mapping courtesy of ArcGIS ESRI portal and Google Maps.)

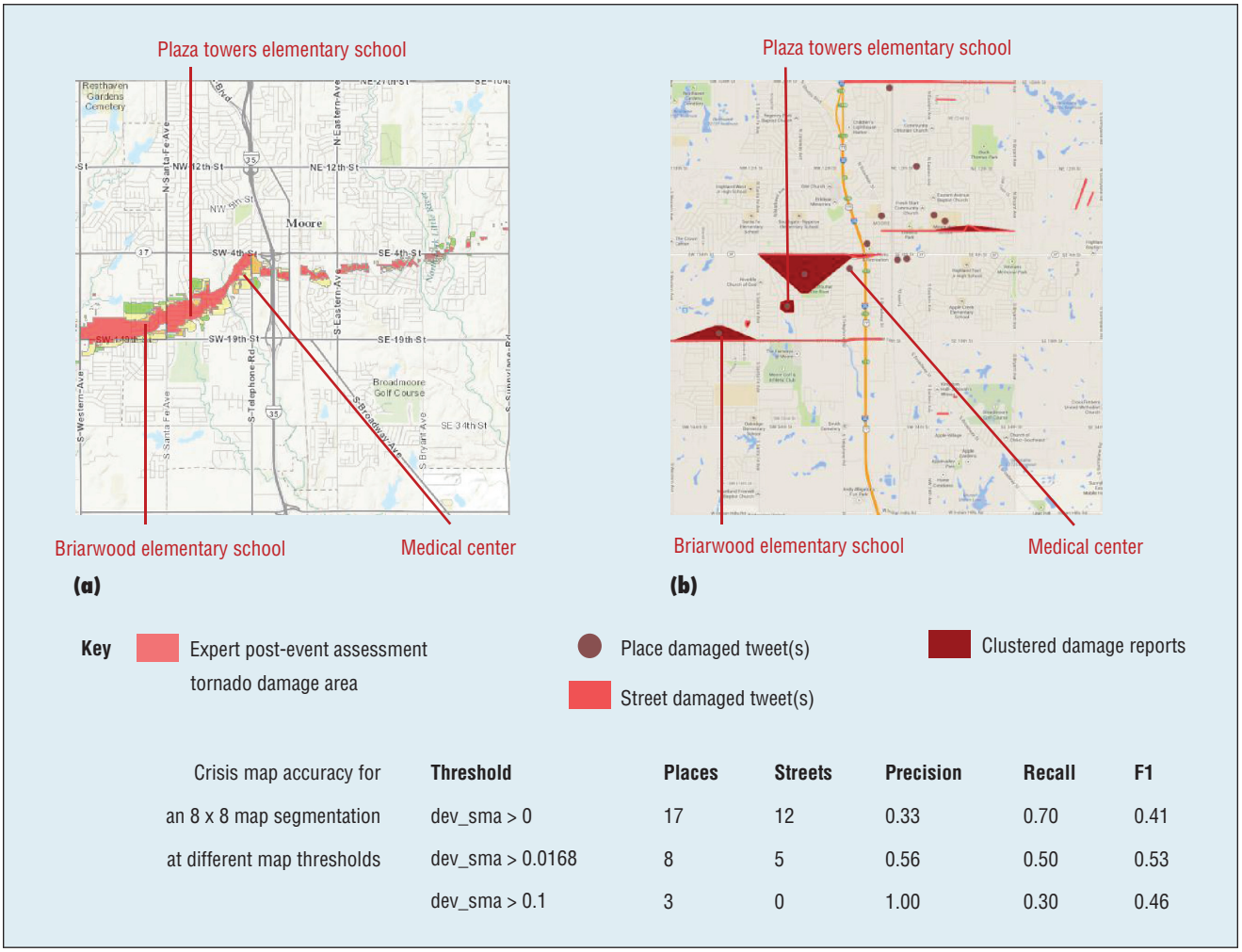
GNS Gazetteer and a local gazetteer, as well as additional map data from OpenStreetMap. Figure 3a shows our benchmarked results by place, street, and region. Although our approach's F1 measure is similar to the gold standard, the street-level precision is much higher. This is an attractive result, given that a low false-positive rate is an important requirement for control room staff to avoid wasting time during a crisis situation.[9]

Statistical Analysis of Location Matches

We calculate a statistical baseline for each location to allow us to compute a threshold level for tweet mentions before which each location is displayed on the crisis map. We use a configurable sample period (for example, 5 minutes) and sample window (for example, 6 hours) over which to calculate our statistics. For each location, we count the number of tweets per sample period in which the location is mentioned, using a historical dataset for the baseline in which no disasters occurred. We then calculate a simple moving average and triangular weighted moving average across the dataset as a whole for the moving sample window. Both case studies reported here use a one-month baseline tweet dataset with just less than 1 million tweets each.

The same per-location match statistics are calculated for a moving sample window of real-time tweet data. The deviation of real-time metric values from baseline metric values is calculated every sample period, and compared to a configurable threshold before displaying each location on the crisis map.

Our central hypothesis is that locations mentioned many times in a sample window are more likely to be coherent and credible disaster-related location reports than those with only one or two mentions. In the case studies reported next, we use the simple moving average metric, and show how raising the threshold



Crisis map accuracy for	Threshold	Places	Streets	Precision	Recall	F1
an 8 x 8 map segmentation	dev_sma > 0	17	12	0.33	0.70	0.41
at different map thresholds	dev_sma > 0.0168	8	5	0.56	0.50	0.53
	dev_sma > 0.1	3	0	1.00	0.30	0.46

Figure 5. Crisis map comparison for Oklahoma's 2013 tornado. (a) The ground truth official post-event US NGA impact assessment showing building damage. (b) A 5-day tweet damage map (dev_sma > 0) for tweets between 20 and 24 May 2013. Red annotations show major incidents. (Source: US NGA damage assessment using aerial [FEMA, BAE Systems] and satellite images [World View 1], May 2013. Mapping courtesy of ArcGIS ESRI portal and Google Maps.)

level for acceptance increases precision. Ultimately, this threshold value will be tailored to suit each crisis-management control room, reflecting the error tolerance of the final decision makers.

Crisis-Mapping Case Studies

We conducted two case studies to evaluate the quality of our tweet maps. The first event studied was Hurricane Sandy (October 2012), which caused major flooding in New York and New Jersey. The second event was the May 2013 tornado that devastated the town of Moore, south of Oklahoma.

For the New York flooding event, we ran our crisis mapping with a sample window of 6 hours and a sampling rate of 5 minutes. Three maps were computed using a high/medium/low threshold setting for the allowed deviation of simple moving average from baseline (dev_sma). The tweet dataset covered 5 days, contained 597,022 tweets (15,175 after time zone and retweet filtering), of which 4,302 mentioned a location. Our New York location database has 8,298 places, 12,800 streets, and 48,661 regions available for matching. We have all regions (cities, suburbs, neighbourhoods, and so on) for New Jersey from our gazetteers, and coastal street data from OpenStreetMap and GooglePlaces covering all of Manhattan.

We compared each map to a ground truth storm-surge map from the official post-event impact assessment produced by the US NGA. Figure 4 shows the post-event storm-surge map alongside our 5-day tweet map. To empirically evaluate our map, we segmented it into an 8 × 8 grid and compared each grid cell to the ground truth map. True positives were reported for any cell that has both a tweeted location reported and some storm-surge activity on the

THE AUTHORS

Stuart E. Middleton is a senior research engineer at the University of Southampton IT Innovation Centre. His main research interests are social media, sensor systems, data fusion, and ontologies. Middleton has a PhD in computer science from the University of Southampton. Contact him at sem@it-innovation.soton.ac.uk.

Lee Middleton is a senior research engineer at the University of Southampton IT Innovation Centre. His main research interests are computer vision, machine learning, and pattern analysis. Middleton has a PhD in electrical and electronic engineering from the University of Auckland, New Zealand. Contact him at ljm@it-innovation.soton.ac.uk.

Stefano Modafferi is a senior research engineer at the University of Southampton IT Innovation Centre. His main research interests are information modelling and software architectures. Modafferi has a PhD in information engineering from the Politecnico di Milano. Contact him at sm@it-innovation.soton.ac.uk.

US NGA impact assessment map. We counted the number of true/false positives and negatives and calculated precision, recall, and F1 measures. As expected, when we increase the mapping threshold (dev_sma), the map precision increases at the expense of recall.

For the Oklahoma tornado event, we also ran our crisis mapping using a 6-hour sample window and sampling period of 5 minutes, generating three maps with the same threshold values as the New York case study. The tweet dataset covered 5 days and contained 877,527 tweets (92,300 after time zone and retweet filtering), of which 42,434 included a location mention. Our Oklahoma location database has 625 places, 3,930 streets, and 18,599 regions available for matching.

Our ground truth map was a US NGA post-event impact assessment showing structural damage across the town of Moore. Figure 5 shows this post-event damage assessment alongside our 5-day tweet map. We segmented each map into an 8×8 grid as before and compared each cell, counting the true/false positives and negatives. The results again show that we can raise the overall map precision at the expense of recall by raising the mapping threshold level.

Both of our case studies demonstrate that it's possible to obtain high-precision (90 percent or higher) geoparsing from real-time Twitter data by exploiting large databases of preloaded location information for at-risk areas. Such data are readily available online from mapping services, VGI sources, and gazetteers. These case studies also show that crisis maps generated from social media data can compare well to gold-standard post-event impact assessments from national civil protection authorities. This matches well with the requirements of use cases such as tsunami early warning centers, which require real-time crisis mapping with minimal false positive rates.

When applying our approach in the future, it's important to consider the spatial size and significance of the natural disaster, as the quality of the crisis map is directly related to the number of people tweeting information about the disaster zone. Large-scale newsworthy events usually receive more tweets than events in small, localized areas, or areas in remote locations with limited mobile communication. However, as the uptake of social media around the world increases with time, we feel that the role that this type of social intelligence has to play in assisting civil protection authorities will also increase.

For next steps, we're experimenting with approaches for language-specific context filtering to be applied as a type of secondary filter on the subset of tweets that match the initial geoparsing stage. This context filter would look at the natural language context in which locations are mentioned and try to classify patterns associated with specific classes of response, such as flooded transport systems, positive/negative reports, cries for help, and reports with high levels of urgency. We will also look at using retweets for adding credibility to original reports.

Currently, each instance of our location-extraction process looks for location matches in a single region of interest. In the future, we'll scale our approach across a computing cluster to handle many spatial regions of interest simultaneously. This offers the possibility of country-wide area map coverage and/or collections of processes that can be adaptively tasked to monitor new spatial areas on demand.

The prototype has been successfully deployed as part of the award-winning TRIDEC project, allowing potential users to assess the social media crisis-management platform first-hand and start the user evaluation process and progress toward adopting this early-stage technology. ■

Acknowledgments

The work presented in this article is part of the research and development in the TRIDEC project (contract number 258723), supported by the Seventh Framework Program of the European Commission.

References

1. P. Meier, "New Information Technologies and Their Impact on the Humanitarian Sector," *Int'l Rev. Red Cross*, vol. 93, 2011, article no. 844.
2. T. Sakaki, M. Okazaki, and Y. Matsuo, "Tweet Analysis for Real-Time Event Detection and Earthquake Reporting System Development," *IEEE Trans.*

Knowledge and Data Eng., vol. 25, no. 4, 2013, pp. 919–931.
3. P.S. Earle, D.C. Bowden, and M. Guy, "Twitter Earthquake Detection: Earthquake Monitoring in a Social World," *Annals Geophysics*, vol. 54, no. 6, 2011; doi:10.4401/ag-5364.
4. N.R. Adam, B. Shafiq, and R. Staffin, "Spatial Computing and Social Media in the Context of Disaster Management," *IEEE Intelligent Systems*, vol. 27, no. 6, 2012, pp. 90–96.
5. J. Yin et al., "Using Social Media to Enhance Emergency Situation Awareness," *IEEE Intelligent Systems*, vol. 27, no. 6, 2012, pp. 52–59.
6. X. Liu et al., "Named Entity Recognition for Tweets," *ACM Trans. Intelligent Systems and Technology*, vol. 4, no. 1, 2013, article no. 3.
7. J. Gelernter and S. Balaji, "An Algorithm for Local Geoparsing of Microtext," *GeoInformatica*, vol. 17, no. 4, 2013, pp. 635–667.
8. G. Shi and K. Barker, "Extraction of Geospatial Information on the Web for GIS Applications," *Proc. 10th IEEE Int'l Conf. Cognitive Informatics & Cognitive Computing*, 2011, pp. 41–48.
9. A. Zielinski et al., "Spatio-Temporal Decision Support System for Natural Crisis Management with TweetComP1," *Proc. Workshop Exploring New Directions for Decisions in the Internet Age* (EWG-DSS 13), F. Dargam et al., eds., 2013, p. 33.
10. A. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python—Analyzing Text with the Natural Language Toolkit*, O'Reilly Media, 2009.

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