Using Social Media to Enhance Emergency Situation Awareness: Extended Abstract*

Jie Yin[†], **Sarvnaz Karimi**[†], **Andrew Lampert**[‡], **Mark Cameron**[†],
Bella Robinson[†], **Robert Power**[†]
[†]Digital Productivity Flagship, CSIRO, Australia
{jie.yin, sarvnaz.karimi, mark.cameron, bella.robinson, robert.power}@csiro.au
[‡]Palantir Technologies
andrew@thoughtlets.org

Abstract

Social media platforms, such as Twitter, offer a rich source of real-time information about real-world events, particularly during mass emergencies. Sifting valuable information from social media provides useful insight into time-critical situations for emergency officers to understand the impact of hazards and act on emergency responses in a timely manner. This work focuses on analyzing Twitter messages generated during natural disasters, and shows how natural language processing and data mining techniques can be utilized to extract situation awareness information from Twitter. We present key relevant approaches that we have investigated including burst detection, tweet filtering and classification, online clustering, and geotagging.

1 Introduction

Social media has emerged as a popular channel for providing new sources of information and rapid communications, particularly during mass emergencies. First-hand information from people on the scene often conveys timely and actionable information, which is greatly valuable for official authorities to better respond to the emergencies. Augmenting the traditional communication means such as phones with social media significantly increases the amount of information exchanged between affected people and emergency responders. This leads to higher situational awareness which in return translates to faster and more informed decisions and actions. Disaster management using Twitter has recently been studied for humanitarian crises and natural disasters such as earthquakes, bushfires, and cyclones [Sakaki *et al.*, 2010; Vieweg *et al.*, 2010; Li *et al.*, 2012; McMinn *et al.*, 2014].

Harnessing credible information on emergency events from social media is however very challenging. Twitter for example only allows short textual messages called tweets of up to 140 characters. It thus encourages its users to use abbreviations to shorten their tweets. Language of Twitter users can be largely different to formal text, with users often communicating in colloquial language which is not handled effectively by well-established natural language processing techniques. The sheer volume of information that is spread by millions of users is also overwhelming. It requires text mining algorithms that can handle massive amounts of data in real time and filter through this data to find the right information. Twitter users also often avoid using automatic geotagging of their tweets. While that is often to protect their privacy, it poses challenges to the applications that require location information to make sense of the tweet content.

To overcome these challenges, we investigate several key text and data mining techniques for extracting situation awareness information from Twitter. These techniques cover a series of sub-problems ranging from detecting events from streaming data, classifying and filtering large datasets for relevant information, to discovering geolocations in user posts. Due to unexpected nature of natural disasters, statistical methods that detect a sudden increase in the frequency of counts are often candidate approaches. We adapt burst detection techniques for identifying early indicators of unexpected events. To avoid information overload, we develop classification and online clustering methods for filtering and summarizing disaster-related information from Twitter messages. Geolocating the tweets based on their content helps identify where help is needed or what areas are affected during emergencies. We present our new algorithm that aggregates location clues from tweet and infers a coherent locational focus.

2 Burst Detection

Burst detection focuses on monitoring a feed of Twitter messages, and raising an alert for immediate attention when an unexpected event is detected. In the data mining area, burst detection techniques have been studied to identify emergent patterns from data streams [Fung *et al.*, 2005; Kleinberg, 2003]. To achieve real-time efficiency, we adopt a parameter-free algorithm [Fung *et al.*, 2005] to identify bursty words from Twitter text streams. The basic idea is to determine whether a word is bursty based on its probability distribution in a time window. Specifically, we compute the probability of the number of tweets that contain a word v_j in a time window

[*] This paper is an extended abstract of the IEEE IS journal publication [Yin *et al.*, 2012].

W_i, denoted as $P(n_{i,j})$, using a binomial distribution:

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

where N is the number of tweets in a time window. It is worth noting that, although the number of tweets, N_i, in each time window may be different, we can re-scale it in all time windows by adjusting the frequencies of words, such that all N_i become the same and N is thus not considered as a parameter.

In the above distribution, p_j is the expected probability of the tweets that contain a word v_j in a random time window, and is therefore the average of the observed probability of v_j in all time windows containing v_j, which is defined as $p_j = \frac{1}{L} \sum_{i=0}^{L} P_o(n_{i,j})$, where $P_o(n_{i,j}) = \frac{n_{i,j}}{N}$ and L is the number of time windows containing v_j.

We determine whether a word v_j is bursty or not by comparing the actual probability $P_o(n_{i,j})$ that the word v_j occurs in the time window W_i against the expected probability p_j of the word v_j occurring in a random window. If $P_o(n_{i,j})$ is noticeably higher than the expected probability of the word $v_j(p_j)$, it indicates that v_j exhibits an abnormal behavior in W_i, and we thus consider v_j as a bursty word in W_i.

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

For evaluation, we annotated about 8,400 features in a six-month Twitter dataset collected in 2010. A burst is defined as one word that suddenly occurs frequently in a time window and its occurrence lasts more than one minute. The performance of burst detection was evaluated using two metrics: *detection rate* and *false alarm rate*. Our experiments show that, our burst detection mechanism achieves an overall detection rate of 72.1% and a false alarm rate of 1.4%.

3 Tweet Filtering

When monitoring Twitter for a specific event, a large volume of tweets published every second are considered irrelevant. Even when the tweets are discussing an event of interest, depending on the application one may be interested in prioritizing different classes of messages.

To address this need in the context of disaster management, we study three different tweet classification settings: *disaster or not* [Karimi *et al.*, 2013], *disaster type* [Karimi *et al.*, 2013], and *impact assessment* [Yin *et al.*, 2012]. Disaster or not is a binary classifier that classifies whether or not a tweet is talking about a disastrous event. Disaster type classifier classifies tweets into a representative disaster type: earthquake, flooding, fire, storm, other disasters (e.g., traffic accident and civil disorders), and non-disaster.

Another level of filtering is to identify tweets reporting a damage to infrastructure during a disaster. Infrastructure is defined as roads, bridges, railways, airports, commercial and residential buildings, water, electricity, gas, and sewerage supplies. Identifying tweets that contain such information assists authorities to better plan their response to disasters.

A cross-disaster classification setting is also studied [Karimi *et al.*, 2013] as an extension to disaster type classification. The goal is to investigate if a classifier is trained on specific disasters, such as earthquake, fire, and flooding, how would that perform in identifying other disaster types such as storm. This is important because we may not always have enough training data for all the possible disaster types.

3.1 Methodology

Support Vector Machines (SVM) and Naïve Bayes classifiers are shown the most effective for text classification. We therefore used the two classification algorithms in our experiments. To extract features we examined a number of different feature combinations. For infrastructure classification these features were used: word unigrams, word bigrams, word length, the number of hashtags, the number of user mentions, whether a tweet is retweeted, and whether a tweet is replied to by other users. Disaster or not and disaster type classifiers utilized word unigram, word bigram, hashtag, hashtag count (number of hashtags in a tweet), mention, mention count (number of user mentions in a tweet), link (a binary feature whether or not a link exists in a tweet), and tweet length.

3.2 Evaluation

We evaluated the classifiers using *accuracy* metric, which indicates the percentage of correctly classified tweets.

For impact assessment classification, 10-fold cross-validation of a manually annotated dataset of 450 tweets was used. Using all the features, Naïve Bayes and SVM achieve classification accuracy of 86.2% and 87.5%, respectively.

For disaster or not and disaster type classifiers, we used a time-split evaluation scheme [Karimi *et al.*, 2015] that prevents any biases in evaluations that may occur due to the dependencies among Twitter data. In our time-split evaluation, a dataset of 5,747 tweets was sorted in chronological order and divided into two sets of training and testing. Therefore, the training data represents older tweets based on their tweet time. For both of the classifiers, using a combination of hashtags and word unigrams performed better than other feature combinations once at least 50% of the data was used for training. Once trained on 90% of the data, both of the classifiers were accurate over 90% of the time. SVM classifier consistently outperformed Naïve Bayes in all the feature settings and therefore Naïve Bayes was left unreported.

A cross-disaster setting was also evaluated [Karimi *et al.*, 2013]. We evaluated all the possible settings between four types of disasters (earthquake, flooding, fire, and storm) in which three disasters were kept for the training and one for the testing. The results showed that more generic features, such as hashtag count or mention count, were more effective than incident-specific features, such as actual hashtag or mention values, for identifying previously unseen types of disasters. Accuracies for the best setting (unigram plus hashtag count) varied between 60% to 73%.

4 Online Clustering

Online clustering aims to automatically group similar tweets into a set of clusters, such that each cluster corresponds to an event-specific topic. For this task, desirable clustering algorithms should be scalable to handle the massive volume of tweets under the one-pass constraint of streaming scenarios.

4.1 Algorithm Description

To meet this need, we propose a new clustering algorithm [Yin, 2013], which intelligently divides the computational process into two phases, *i.e.*, an online discovery phase and an offline cluster merging phase.

Online Discovery Phase The online phase provides a one scan algorithm over the incoming Twitter stream to identify base clusters, with each cluster consisting of a set of similar tweets. To represent textual content of tweets, we employ a traditional vector-space model, in which a tweet $\mathbf{m}_i$ is represented as a vector of words $(v_1, v_2, \ldots, v_d)$, where d is the size of word vocabulary and v_j is the TF-IDF weight of j^{th} term in tweet $\mathbf{m}_i$. In streaming scenarios, because word vocabulary dynamically changes over time, it is very computationally expensive to recalibrate the inverse document frequency of TF-IDF. We thus use term frequency as the term weight and adopt a sparse matrix representation to deal with dynamically changing vocabulary in our clustering algorithm.

In order to cluster tweets into temporally-related groups, we incorporate temporal information for measuring the similarity between two tweets. The similarity measure used in our clustering algorithm is defined as

$$\mathtt{sim}(\mathbf{m}_i, \mathbf{m}_j) = \cos(\mathbf{m}_i, \mathbf{m}_j) \cdot \exp(-\frac{|\mathbf{m}_i^t - \mathbf{m}_j^t|}{\lambda}). \quad (2)$$

We use cosine similarity to measure textual similarity between tweets $\mathbf{m}_i$ and $\mathbf{m}_j$ and penalize the similarity if their publication times are far away. $|\mathbf{m}_i^t - \mathbf{m}_j^t|$ indicates the difference between tweets' publication times, represented as the number of days, and λ is the number of days of one month, whose value is application dependent.

To maintain sufficient statistics about clusters, for each cluster C_i, we store the *textual centroid* C_i^v, which is a feature vector where each element indicates the average weight of the corresponding words for all tweets in cluster C_i, *time centroid* C_i^t, which is the mean publication time of all tweets forming cluster C_i, and cluster size $|C_i|$, which is the number of tweets included in cluster C_i.

Given a Twitter stream in which the tweets are sorted according to their published time, our algorithms works as follows. First, it takes the first tweet from the stream and uses it to form a cluster. Next, for each incoming tweet, say $\mathbf{m}$, it computes its similarity with any existing clusters C_i, that is, $\mathtt{sim}(\mathbf{m}, C_i) = \cos(\mathbf{m}, C_i^v) \cdot \exp(-\frac{|\mathbf{m}_i^t - C_i^t|}{\lambda})$. Let C^* be the cluster having the maximum similarity with $\mathbf{m}$. If $sim(\mathbf{m}, C^*)$ is greater than a threshold δ, which is to be determined empirically, the tweet $\mathbf{m}$ is added to the cluster C^*; otherwise, a new cluster is created. To further improve efficiency, we maintain a list of active clusters. If no more tweets are added to a cluster for a period of time, the cluster is considered inactive and it is removed from the active list. The algorithm only considers those clusters that are in the active list as candidates to which a new tweet can be added.

Offline Cluster Merging Phase The base clusters generated during the online phase serve as an intermediate statistical representation of the Twitter stream. The offline phase is utilized to merge a list of relevant clusters into event-based clusters. There is no need to process the voluminous tweets, but the compactly stored summary statistics of clusters.

For a particular event, since users tend to convey the same or a similar meaning using different words, the online phase would organize the tweets reporting the same event, but expressed using different words, into different clusters. Thus, our algorithm merges together the base clusters that are related with respect to the same event. The principle is to merge a pair of clusters that have a larger inter-cluster similarity, calculated as $\mathtt{link}(C_i, C_j) = \cos(C_i^v, C_j^v) \cdot \exp(-\frac{|C_i^t - C_j^t|}{\lambda})$.

The offline phase provides the flexibility for an analyst to perform queries about clusters and retrieve event-based clusters upon demand at any time horizon. Given a list of clusters generated during the online phase, our algorithm iteratively merges two clusters C_{i^*} and C_{j^*} such that $\mathtt{link}(C_{i^*}, C_{j^*})$ is maximized. We use the notion of *separation* to measure the clustering quality, which is defined as $S(k) = \frac{1}{N(N-1)} \sum_i \sum_j link(C_i, C_j)$, where N is the number of clusters obtained at step k. The smaller value this metric has, the better clusters are separated from each other. Based on this metric, we design a criterion to decide whether or not to stop the merging process. At each step k, given two candidate clusters to be merged, we compute a validation index as $\Delta_k = \frac{S(k+1) - S(k)}{S(k)}$, which represents the relative change in inter-cluster similarity after a merge is made. If $\Delta_k < 0$, that means a cluster merge can improve the separation of clusters. We then proceed with merging the two clusters. Otherwise, if $\Delta_k \geq 0$, we stop the cluster merging process. In this way, the optimal number of clusters can be automatically determined.

4.2 Evaluation

The dataset we used for evaluation is an annotated corpus of tweets collected from July 2011 to September 2011 [Petrović *et al.*, 2012]. The corpus was distributed as a set of tweet IDs and their annotations. We re-retrieved the tweets using Twitter search API and obtained a set of 2,633 tweets. Each tweet was annotated as one out of 27 real-world events, such as London riots or Earthquake in Virginia.

We compared our algorithm with two baselines: (1) A standard incremental clustering (IC) algorithm [Becker *et al.*, 2011]. It determines the assignment of a tweet solely based on its similarity to the textual centroids of clusters; (2) IC-Time, which is a variant of our proposed algorithm that only uses the online phase to discover clusters. F-measure [Larsen and Aone, 1999] was used as the evaluation metric.

Using bag-of-words and hashtags as features, our proposed algorithm achieved an F-measure of 0.958, in comparison to

0.892 achieved by IC, and 0.905 by IC-Time. By removing the # symbol and treating hashtags as normal words, the clustering accuracies for three algorithms were observed to increase; our proposed algorithm achieved an F-measure of 0.966, while IC and IC-Time achieved an F-measure of 0.899, and 0.910, respectively. We believe this improvement is because removing the # symbol contributes to increasing the term frequency of the same topic word in the tweets, which leads to better clustering performance.

5 Geotagging Locational Focus

Geotagging is the process of finding all mentions of textual references to geographic locations in a text (toponym recognition), and determining where on the map these toponyms collectively refer to (toponym recognition). Because a tweet can contain textual references to more than one locations, we aim to find the *locational focus* of the tweet that may be geographically identifiable on a map. In our work, we define a location as a combination of both *geographic location*, such as country, state, city, or suburb, and *Point of Interest* (POI), such as hotel, shopping center, or restaurant. While the majority of the existing studies focus on estimating the users' locations [Cheng *et al.*, 2010; Li *et al.*, 2011; Mahmud *et al.*, 2012], we are interested in deriving a coherent locational focus that is referred to in a tweet if it exists.

5.1 Algorithm Description

To this aim, we propose a novel algorithm [Yin *et al.*, 2014], which identifies all location mentions from tweets and then uses a gazetteer to infer the most probable locational focus.

Finding Location Mentions Identifying location mentions is the first crucial step in inferring the locational focus of a tweet. Because tweets are often very short, tweet text alone cannot provide sufficient evidence for disambiguating location mentions and finding a coherent locational focus. Therefore, we take three sources of information into account: tweet text, hashtags, and user profiles.

We retrained Stanford Named Entity Recognizer on Twitter data to tag location mentions in tweet text, following our prior work [Lingad *et al.*, 2013]. Because hashtags may contain locational clues (*e.g.*, #sydneyfire, #ukfloods), we adopt a hashtag segmentation algorithm which uses a greedy maximal matching method that refers to a word list to break hashtags. The word list is composed of an English dictionary augmented with major states and cities in Australia and New Zealand, as well as their short abbreviations. We also use location information that the users register in their user profiles, especially when specific location information is missing in the tweet content. However, we weight user profile locations less than location mentions found in tweet text and hashtags.

Inferring Locational Focus After location mentions are identified, the next step is to infer a locational focus being referred to by a tweet. Because not all the location mentions equally contribute to the focus, we use a gazetteer database as an external knowledge source to help disambiguate the location mentions and infer the locational focus.

Our gazetteer is built based on two freely available data sources, including gazetteer of Australia 2010 and GeoNames New Zealand gazetteer, augmented with the information from OpenStreetMap on roads, streets, highways, and POIs. We organize our gazetteer as a hierarchy of different geographic substructures: (1) Level 1: Country; (2) Level 2: State/Territory/Region; (3) Level 3: City/Suburb/Town/Non-specific POIs (*e.g.*, mountains or national parks); and (4) Level 4: Specific POIs (*e.g.*, school or airport).

Our algorithm first queries the gazetteer for each location mention identified in a tweet, and builds an inference tree to summarize all the matches against the gazetteer. Each match is a full path from the matched leaf node via its ancestors to the root. For each matched leaf node, we calculate an *importance score* that balances the granularities of places in the gazetteer and how well their names match the terms of a given query (measured using Jaccard similarity). To capture both explicit and implicit matches, we also calculate an importance score for each intermediate node in the inference tree, by recursively aggregating the scores of its child nodes in a bottom-up manner. Finally, our algorithm performs a top-down traversal over the inference tree to find an optimal path having the maximum score, which is the locational focus of the tweet. It uses an entropy threshold δ to decide at which granularity level the traversal should stop.

5.2 Evaluation

We evaluated the effectiveness of our proposed algorithm using a dataset of 1,441 tweets annotated with their location mentions and locational focus. We used a random validation set of 80 tweets to tune the entropy threshold ($\delta = 3.4$) and the rest of 1,318 tweets for testing. When calculating the base scores, we set the weights as 0.6, 0.3, and 0.1, for tweet text, hashtags, and user profile, respectively.

We tested the accuracy of our algorithm when different information sources are used. Among others, a combination of all the three sources (tweet text, hashtags, and user profiles) significantly outperformed each of the individual sources; 89.9% accuracy was achieved for identifying countries (Level 1), 73.5% for states/regions (Level 2), 51.0% for cities/suburbs (Level 3), and 29.7% for specific POIs (Level 4). Hashtags were found useful in finding countries, states, or cities, as people often use hashtags to indicate the type of an event together with its associated city or country (*e.g.*, #sydneyfire). User profile locations were helpful for identifying country, but had almost no contribution to detecting POIs. Expectedly, finding POIs was most difficult with only about 30% correctly identified when all the three sources were used.

We also compared our proposed algorithm with a service provided by Yahoo! BOSS Geo services[1], called PlaceFinder. While there was no significant difference at the country level (around 90%), our algorithm remarkably outperformed this service at finer levels of granularities; for example, our algorithm achieved an accuracy of 73.5% and 51.0% for identifying states/regions, and cities/suburbs, respectively, in comparison to 59.1% and 23.5% by PlaceFinder.

[1] http://developer.yahoo.com/boss/geo/

6 Conclusion

The growing use of social media during natural disasters and crises provides on-the-ground information reported from the general public. This work focused on analyzing Twitter messages generated during humanitarian crises, and presented key relevant methods for burst detection, tweet filtering and classification, online clustering, and geotagging. Development and evaluation of these methods showed that if the right information is sifted through social media, it can facilitate the right authorities to enhance their awareness of time-critical situations and make better decisions for emergency response.

References

[Becker *et al.*, 2011] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on Twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, pages 438–441, Barcelona, Spain, 2011.

[Cheng *et al.*, 2010] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: A content-based appraoch to geolocating Twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 759–768, Toronto, Canada, 2010.

[Fung *et al.*, 2005] Gabriel P. C. Fung, Jeffrey X. Yu, Philip S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pages 181–192, Trondheim, Norway, 2005.

[Karimi *et al.*, 2013] S. Karimi, J. Yin, and C. Paris. Classifying microblogs for disasters. In *Proceedings of the 2013 Australasian Document Computing Symposium*, pages 26–33, Brisbane, Australia, 2013.

[Karimi *et al.*, 2015] S. Karimi, J. Yin, and J. Baum. Evaluation methods for statistically dependent text. *Computational Linguistics*, (Accepted November 2014, to appear), 2015.

[Kleinberg, 2003] J. Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.

[Larsen and Aone, 1999] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–22, San Diego, CA, August 1999.

[Li *et al.*, 2011] W. Li, P. Serdyukov, A.P. de Vries, C. Eickhoff, and M. Larson. The where in the tweet. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 2473–2476, Glasgow, Scotland, UK, 2011.

[Li *et al.*, 2012] R. Li, K. H. Lei, R. Khadiwala, and K. Chen-Chuan Chang. TEDAS: A Twitter-based event detection and analysis system. In *Proceedings of the IEEE 28th International Conference on Data Engineering*, pages 1273–1276, Arlington, VA, 2012.

[Lingad *et al.*, 2013] J. Lingad, S. Karimi, and J. Yin. Location extraction from disaster-related microblogs. In *Proceedings of WWW Companion*, pages 1017–1020, Rio de Janeiro, Brazil, 2013.

[Mahmud *et al.*, 2012] J. Mahmud, J. Nichols, and C. Drews. Where is this tweet from? Inferring home location of Twitter users. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, pages 511–514, Dublin, Ireland, 2012.

[McMinn *et al.*, 2014] A. J. McMinn, D. Tsvetkov, T. Yordanov, A. Patterson, R. Szk, J. A. Rodriguez Perez, and J. M. Jose. An interactive interface for visualizing events on Twitter. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1271–1272, Gold Coast, Australia, 2014.

[Petrović *et al.*, 2012] S. Petrović, M. Osborne, and V. Lavrenko. Using paraphrases for improving first story detection in news and Twitter. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–346, Montreal, Canada, 2012.

[Sakaki *et al.*, 2010] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th World Wide Web Conference*, pages 851–860, Raleigh, NC, 2010.

[Vieweg *et al.*, 2010] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging during two natural hazards events: What twitter may contribute to situational awareness. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, pages 1079–1088, Atlanta, GA, 2010.

[Yin *et al.*, 2012] J. Yin, A. Lampert, M. Cameron, B. Robinson, and R. Power. Using social media to enhance emergency situation awareness. *IEEE Intelligent Systems*, 27(6):52–59, 2012.

[Yin *et al.*, 2014] J. Yin, S. Karimi, and J. Lingad. Pinpointing locational focus in microblogs. In *Proceedings of the 2014 Australasian Document Computing Symposium*, pages 66–72, Melbourne, Australia, 2014.

[Yin, 2013] J. Yin. Clustering microtext streams for event identification. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 719–725, Nagoya, Japan, 2013.