Detecting Mass Emergency Events on Social Media: One Classification Problem or Many?

V. Pekar[1], **J. Binner**[1], **H. Najafi**[2]
[1]Business School, University of Birmingham, Birmingham, United Kingdom
[2]Computer Science and Information Systems, University of Wisconsin, River Falls, WI, USA,

Abstract *Social media proves to be a major source of timely information during mass emergencies. A considerable amount of recent research has aimed at developing methods to detect social media messages that report such disasters at early stages. In contrast to previous work, the goal of this paper is to identify messages relating to a very broad range of possible emergencies including technological and natural disasters. The challenge of this task is data heterogeneity: messages relating to different types of disasters tend to have different feature distributions. This makes it harder to learn the classification problem; a classifier trained on certain emergency types tends to perform poorly when tested on some other types of disasters. To counteract the negative effects of data heterogeneity, we present two novel methods. The first is an ensemble method, which combines multiple classifiers specific to each emergency type to classify previously unseen texts, and the second is a semi-supervised generic classification method which uses a large collection of unlabeled messages to acquire additional training data.*

Keywords: text classification, semi-supervised learning, social media analysis, disaster management

1 Introduction

Social media data offer a very promising way forward as a mechanism for facilitating the work of first responders in dealing with mass emergency events. During a crisis such as a natural disaster or a terrorist attack, social media has become a primary source of information, publishing eyewitness reports on the events in real-time. Information systems that identify and collate such eyewitness reports can provide critical situation awareness to increase the efficiency and capabilities of the emergency services, making them better equipped to detect disasters at early stages, monitor their development and tackle their consequences in the recovery operations.

The potential of social media analysis for mass emergency management has attracted many researchers in the fields of Data Mining over the past several years. Previous work has primarily focused on detecting emergency-related tweets using text classification. Limiting the problem to a single disaster type such as earthquakes or tornados has been shown to achieve high accuracy of classification (e.g., [4, 8, 9, 10]). However, because mass emergency events can differ a lot in terms of their causes, temporal and geographical spread, impacted targets and the nature of damage, it would be much more practical to have a classification method that can cover a wide range of possible disasters. This will give first responders and emergency services personnel confidence that disasters with some previously unseen characteristics would be successfully recognized by the alerting system.

This paper is concerned with the task of recognizing mass emergencies unspecified for a particular type, which could include both natural disasters such as earthquakes, floods and storms, as well as man-made ones such as explosions, collisions and shootings. This is a non-trivial classification problem, as the data is non-homogeneous: the classifier is trained and evaluated on data covering different emergency types; each characterized by its own vocabulary and correspondingly different feature distributions. We examine two possibilities to counter the problem of heterogeneous data. The first approach views this task as multiple classification problems, training one classifier for each type of known disasters; an ensemble of the classifiers is then used to classify test messages that can possibly come from an unknown disaster type. The second approach treats the task as a single classification problem: in order to better capture commonalities that exist between different types of disasters, it uses a co-training method, which obtains additional training data from a large collection of unlabeled messages. Thus our main contributions are the novel methods that are specifically suited to the task of detecting emergency events that were unseen at the training stage and their comparative evaluation.

The paper is organized as follows. In the next section we outline previous work on detecting mass emergencies using machine learning text classifiers. In Section 3 we describe the proposed ensemble classification method, and in Section 4 the co-training method. Sections 5 and 6 present the experimental setup, the results of the experiments and their discussion. Section 7 concludes and offers suggestions for future research.

2 Related work

There is a considerable body of work on detection of new events in a stream of text messages, where the type of the event of interest is not known in advance, and some of these approaches were applied to detecting mass emergency events. Such methods primarily rely on detecting “bursty” keywords [13], i.e. keywords whose frequency increases sharply within a short time window. However, bursty keywords are known to be related not only to events, but also non-events such as "viral" content. To separate them, Becker et al. [2] used a domain-independent text classifier, before applying keyword burstiness techniques.

Domain-specific methods generally have a greater accuracy than domain-independent ones, and previous work specifically on mass emergency detection was concerned with developing domain text classifiers based on machine learning. The classifiers aim to solve a binary classification problem, operating on features extracted from the entire message. Most of this work was concerned with specific types of disasters such as earthquakes [18, 22, 23], tornados [9, 11], and landslides [14].

Verma et al. [21] conducted experiments on how well a classifier trained on one type of emergency would perform on messages representing a different emergency type. They ran all pairwise comparisons between four datasets, which represented two flood events, one earthquake and one wildfire, and found that testing on an emergency type other than the one used for training results in worse classification accuracy; the F-measure ranging between 29 and 83 depending on a specific pair. Ashktorab et al. [1] trained one generic classifier on data from twelve different emergency events, achieving the F-measure between 50 and 65 depending on the learning method; the evaluation was done however by randomly splitting all the data into a test and train sets, i.e., the train and test data contained data representing different disasters in similar proportions. Pekar et al. [17] showed that if a classifier is trained on some disaster types, but evaluated on others, the performance of the classifier drops by up to 70%, when compared with training and testing on the same set of disasters.

3 Ensemble classification

Ensemble or committee-based classification methods aim to leverage advantages of different models trained on the same dataset, in order to improve on performance of individual models [6]. The different models in the ensemble can be learned using different subsets of the training data, different classification parameters of the same learning algorithm, or using different learning algorithms. For a review of ensemble methods applied to text classification, see e.g. [7].

In the context of detecting disaster-related text messages, we create a classifier ensemble through dividing the training instances by disaster type. We then trained one classifier specific to each type, using the same learning algorithm. Each of the classifiers is thus expected to be more effective at classifying just its own disaster type, than a classifier trained on other types or a generic classifier. Test instances representing an unknown disaster would then be classified more effectively by some classifiers than others. This is because the unknown disaster will be more similar to some of the disaster types observed during training than others.

Majority vote among classifiers is the simplest way to derive the eventual class label for the test instance, but in the case of highly heterogeneous data the majority class will seldom be the correct one. Our initial experiments showed that the negative class almost always got the majority vote. Therefore in our implementation the test instance is given the class label of the classifier that assigned it with the highest confidence.

```
Input:
  Training documents $D_{train}$
  Testing documents $D_{test}$
  Emergency types $E$
  Class labels $Y = \{True, False\}$

Training phase:
For each $e$ in $E$:
  Train classifier $c_e$ on a subset of $D_{train}$ each of which belongs
  to $e$

Testing phase:
For each $d$ in $D_{test}$:
  For each $c_e$ in $C = \{c_1, c_2, \dots c_{|E|}\}$:
    Obtain label $y_e$ and classifier confidence score $s_e$
  Assign $y_i$ to $d$, such that $s_i$ is the maximum value in
  $\{s_{e1}, s_{e2}, \dots, s_{|E|}\}$
```

Algorithm 1. Ensemble classification method.

4 Co-Training

Co-training [3] is a semi-supervised learning technique that is aimed to overcome the problem of insufficient training data, if large amounts of unlabeled data are available. It is also commonly used for domain adaptation of a classifier, when train data available for one domain is used to obtain train data for a different domain, thus tuning the classifier to perform more effectively on the new domain. The technique was previously used for domain adaptation for various NLP tasks, including dependency parsing [20], co-reference resolution [15], and sentiment classification of texts [5].

The general co-training algorithm starts with choosing two different "views" on the data, which are disjoint subsets of the entire set of classification features. The subsets are created to satisfy two conditions: (1) each view must be able to learn the classification problem with sufficient accuracy and (2) the views must be conditionally independent of each other. Two classifiers are trained using each view on available labeled training data. They are then used to classify unlabeled data. Instances which either classifier labeled with high confidence are added to the train set, thus the classifiers help each other, adding one's most confident classifications into the other's train set. The classifiers are then re-trained on the new dataset and re-applied to the unlabeled data. These steps are repeated until a certain stopping criterion is reached. After that, the main classifier is trained on the augmented train set and evaluated on the test data.

In our implementation, feature subsets are created as follows. The first one consists of unigrams and bigrams, i.e. lexical features extracted from the cleaned version of the message. The second one includes grammatical features (part-of-speech tags) and features extracted from Twitter metadata (hashtags, mentions, presence of URLs, etc.). The two subsets were found to produce models of similar classification accuracy (for details of the features used and experiments with feature subsets, see Section 6). Unlabeled instances are added to the training set in such a way as to preserve the ratio of positive and negative instances that was found in the original train data. As a stopping criterion, we used the maximum number of automatically added training instances, which we set at 25% of the original training set.

```
Input:
Labeled documents $L$
Unlabeled documents $U$
Data views $V$
Classifier confidence threshold $t$
Desired size of labelled documents $m$

Initialize:
Augmented labeled set $L' \leftarrow L$

Loop:
While $|L'| < m$ and $|U| > 0$:
    for $v$ in $V$:
        Train classifier $c_v$ on $L'$ using features from $v$
    for $u$ in $U$:
        for $c_v$ in C:
            Classify $u$ with $c_v$, obtaining class label $y_v$ and
            confidence score $s_v$
        Select $y_i$ from $\{y_1, y_2, \ldots, y_{|V|}\}$ such that $s_i$ is the
        maximum in $\{s_1, s_2, \ldots, s_{|V|}\}$
        if $s_i > t$:
            Assign $y_i$ to $u$ and add $u$ to $L'$
            Remove $u$ from $U$

Output:
Augmented labeled set $L'$
```

Algorithm 2. Acquisition of labeled data in the co-training algorithm.

5 Experimental setup

5.1 Labeled data

In the experiments we use the labeled part of the CrisisLexT26 dataset [16], which includes tweets on twenty six mass emergencies that occurred between 2012 and 2013. The types of emergencies are very diverse and range from terrorist attacks and train derailment to floods and hurricanes. Some examples are Colorado wildfires in 2012, Venezuela refinery explosion in 2012, and Boston bombings in 2013. The dataset was created by first retrieving tweets based on a set of search terms relating to mass emergencies, and thus is representative of data that is likely to be found in real-world use cases after initial keyword-based filtering.

The evaluation included three classification tasks, which are of different practical value for emergency responders and at the same time differ in terms of the difficulty of the classification problem:

i. Relatedness: separating messages related to a mass emergency from unrelated ones,

ii. Informativeness: separating informative messages (whether the message contributes to better understanding of the crisis situation) from uninformative ones (refers to the crises but involves sympathy, jokes, etc.),

iii. Eyewitnesses: detecting eyewitness accounts of mass emergencies (first-hand descriptions of the events).

Figure 1 shows examples of positive and negative messages for the three tasks (examples taken from Olteanu et al. [16]).

Table 1 describes the size of the positive and negative classes in the three classification tasks in the CrisisLexT26 dataset.

	Positive	Negative
Relatedness	24581	2863
Informativeness	16849	7732
Eyewitnesses	2193	22396

Table 1. The sizes of the positive and negative classes in the classification tasks.

	Positive	Negative
Relatedness	RT @NWSBoulder Significant flooding at the Justice Center in #boulderflood	#COstorm you are a funny guy lol
Informativeness	Flash floods wash away homes, kill at least one near Boulder via @NBCnews	Pray for Boulder, Colorado #boulderflood
Eyewitnesses	Outside sounds like it is going to shatter my bedroom windows any sec now #bigwet #qld	RT @RedCrossAU: Everyone affected by #qldfloods, let people know you’re safe: http://t.co/..

Figure 1. Examples of messages belonging to positive and negative classes of the three classification tasks.

5.2 Unlabeled data

To obtain unlabeled data for co-training experiments, we created 24 search terms describing different types of mass emergencies: *avalanche, blizzard, cyclone, earthquake, flood, landslide, heat wave, eruption, storm, tornado, tsunami, wildfire, bushfire, crash, explosion, collision, disaster, shooting, accident, capsize, sank, stampede, collapse, massacre*. Submitting the search terms to the Twitter Search API we continuously retrieved tweets that were published between February 23, 2016 and March 08, 2016, obtaining 2,479,079 tweets in total.

5.3 Preprocessing

We apply the following preprocessing steps to the data, which are commonly used for Twitter messages before performing text classification on them in order to reduce the amount of noisy features (see, e.g.,[12]):

- **Text normalization**. Before processing the text of the message with a PoS tagger, the text was normalized: mentions (e.g., @username) and URLs removed; sequences of hashtags at the start and end of the message removed; hashtags appearing in the middle of the text were kept, but the hash symbol removed from the hashtags; long non-alphanumeric symbol sequences, which tend to be emotions, were removed; word tokens consisting of digits were replaced with a unique tag.
- **Part-of-speech tagging**. The normalized text was tagged with the PoS tagger in the Pattern library [19].
- **Stopword removal**. The usual stoplist was used to remove stopwords.
- **Additional metadata**. The CrisisLexT26 data contains the Twitter id of the message, its raw content, and its timestamp. We retrieve via Twitter Search API additional metadata fields, such as the retweet count.

5.4 Classification method

To train classifiers, we use the Maximum Entropy and the Linear Support Vector Machines algorithms[1], which in our previous study on the same dataset proved to be the top performing algorithms[17]. Based on the same study, we use classification features, which were found to positively contribute to the precision of the MaxEnt and SVM classifiers, in order to maximize the quality of automatically added train instances. The features included:

`Unigrams`: whitespace-separated word tokens (nominal: *please, help, fire*).

`Bigrams`: token sequences with the length of two (nominal: *was_scary, we_complained*).

`PartOfSpeechTags`: separate features are created from part-of-speech (PoS) categories, as assigned by a PoS tagger (nominal: *NNS, JJ, VBD*).

`ContainsHashtags`: whether or not the tweet contains any hashtags (Boolean).

`RetweetCount`: the number of times the message has been retweeted (continuous).

`ContainsURL`: whether the tweet contains a URL (Boolean).

Prior to training and classification, all features are converted to continuous values.

A usual experimental setup for text classification involves randomly splitting labeled data into a train and a test set. However in our experiments, to better reflect intended real-world use cases, the train-test split was done in a way that the test data contained tweets only on those disasters that were not included into the train data, i.e., simulating the conditions when a disaster needs to be detected before any manually labelled data relating to it are available. Thus we create nine train-test splits, so that in each split data on 23 disasters were used for training and data on 3 remaining crises were used for testing. The performance of the classifiers was measured in terms of precision, recall and F-measure rates averaged across the nine train-test splits. In the following sections, we report them only for the positive class that has main practical interest.

6 Results and Discussions

As mentioned in Section 4, we create two feature subsets to be used in the co-training algorithms: View 1, consisting of lexical features, and View 2, consisting of grammatical features and features derived from metadata found in the messages. To verify both views achieve reasonable performance, we evaluated them on the labelled data. Tables 2 and 3 show the results for the three classification tasks.

The results indicate that for the first two tasks both views have a similar level of accuracy: the differences between them are no more than 4 points for either precision and recall, for both MaxEnt and SVM. The eyewitness detection task, however, seems much harder than the other two. With MaxEnt, View 1 achieves noticeably better precision, while View 2 is better in terms of recall. This finding is in agreement with our previously published experiments [17] that showed that lexical features contribute to higher precision, while non-lexical ones improve recall. For SVM, the picture is the opposite: View 1 has a higher recall but lower precision than View 2.

We then compared the results achieved by a general classifier, i.e. one that is training on the entire train set of labeled data, to the ensemble and the co-training classification methods. These results for the Relatedness task are shown on Figure 2 (MaxEnt) and Figure 3 (SVM).

[1] We use the implementation in the scikit-learn library: http://scikit-learn.org/stable/

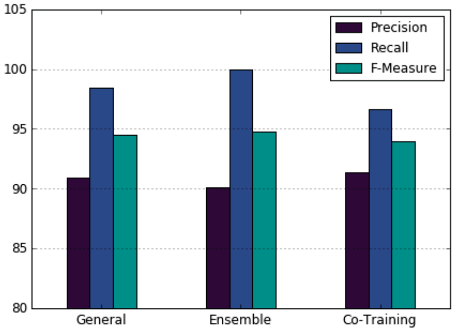

Figure 2. Performance of the general classifier, the ensemble method and the co-training method, Relatedness task, MaxEnt.

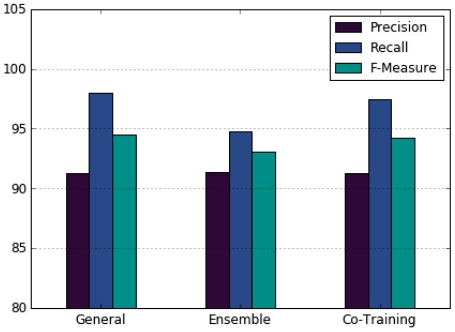

Figure 3. Performance of the general classifier, the ensemble method and the co-training method, Relatedness task, SVM.

The three methods perform very similarly, all achieving both precision and recall of over 90 points. The ensemble classifier improves on the general method by 4 points in terms of recall, but drops several points in precision, which results in an insignificant for F-measure. The co-training method fails to obtain any improvement on the general classifier, for both learning methods, trading a minor gain in precision for a somewhat greater loss in recall.

On the Informativeness task (Figures 4 and 5), the picture is similar. Co-training results for both precision and recall are almost the same as for the general method, the difference being no more than 1 point. The ensemble method, as in the Relatedness task, shows a five points gain on the general method in terms of recall, but loses nine points in terms of precision.

On the Eyewitnesses task (Figure 6 and 7), both ensemble and co-training improve on the general method in terms of F-measure, by 11 and 17 points, respectively. The ensemble method shows a particularly large increase in recall (by 62 points), although it also loses a lot in precision (53 points). The co-training method gains 14 points in recall, but loses 27 in precision.

Thus, the experimental results demonstrate that the Relatedness and Informativeness tasks are not affected by the data heterogeneity problem; a general classifier that can separate positive and negative classes with a high accuracy levels (F-measure of over 85 points) can be trained on modest amounts of data; ensemble or co-training methods offer no significant improvement over the baseline method. Although direct comparison with previous work is problematic, because previous studies used different evaluation datasets, evaluation metrics and slightly different definitions of “informativeness”, the results we have attained on these two tasks are similar to previous studies who also aimed to detect informative tweets. Verma et al. [21] classified tweets into those that contribute and situational awareness and those that do not, finding that the best classification method achieves the accuracy of 0.88. Imran et al (2014) classify tweets into informative and non-informative, reporting the AUC rate of 0.8.

	View 1			View 2		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Relatedness	90.1	99.9	94.6	90.87	97.8	94.1
Informativeness	81.8	93.2	87.1	83.4	89.7	86.3
Eyewitnesses	52.7	1.3	2.6	45.9	3.5	6.1

Table 2. Performance of MaxEnt classifiers trained on two views used in the co-training algorithm, on three classification tasks.

	View 1			View 2		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Relatedness	90.4	99.0	94.4	90.0	99.8	94.6
Informativeness	84.6	89.0	86.7	83.9	89.3	86.4
Eyewitnesses	55.9	5.3	9.4	58.8	2.3	4.4

Table 3. Performance of SVM classifiers trained on two views used in the co-training algorithm, on three classification tasks.

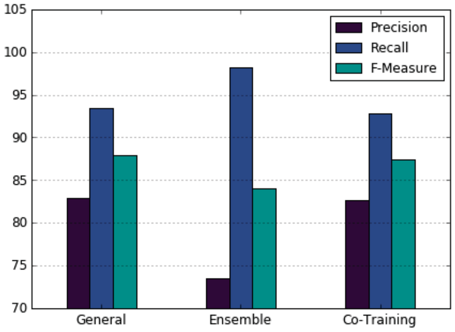

Figure 4. Performance of the general classifier, the ensemble method and the co-training method, Informativeness task, MaxEnt.

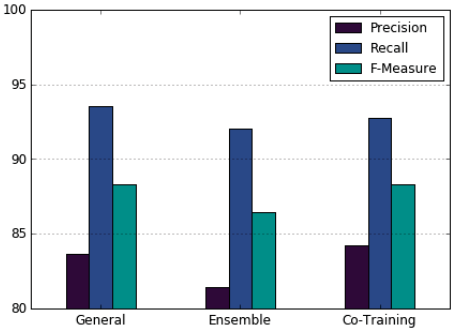

Figure 5. Performance of the general classifier, the ensemble method and the co-training method, Informativeness task, SVM.

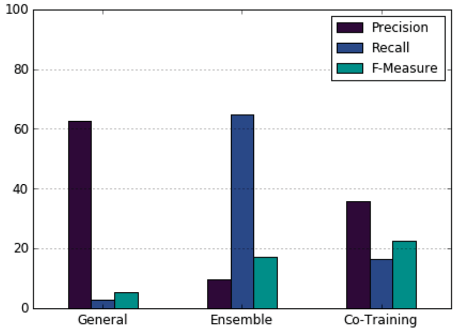

Figure 6. Performance of the general classifier, the ensemble method and the co-training method, Eyewitnesses task, MaxEnt.

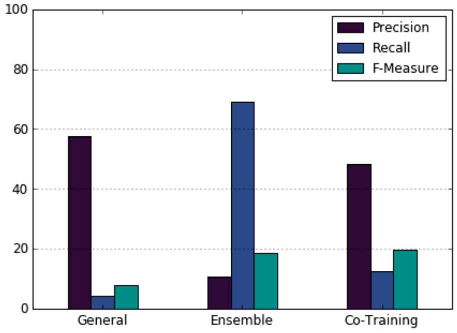

Figure 7. Performance of the general classifier, the ensemble method and the co-training method, Eyewitnesses task, SVM.

The Eyewitnesses task, however, proved a much harder problem. The baseline classifier achieves a precision rate of 60, but also an extremely low recall (under 10 points), which suggests that the model overfits and is not able to generalize sufficiently to the test data. The ensemble method makes it possible to improve recall to over 60 points, at the expense of precision, but nonetheless improving on the baseline in the F-measure. The co-training technique produces a similar effect, also beating the baseline in terms of the F-measure, which is also somewhat higher than that of the ensemble method. To our knowledge, Imran et al.'s study [8] is the only previous paper that evaluated the ability of a classifier to detect eyewitness accounts of mass emergencies. Their Naïve Bayes classifier achieved the F-measure of 60 points, also suggesting that this task is harder than that of informativeness or relatedness classification. These results are also higher than ours, but it should be noted that a direct comparison is not possible because of different experimental settings: specifically, Imran et al. [8] trained and tested their classifier on data representing the same mass emergency event.

7 Conclusion

In this paper we examined the task of detecting social media messages related to a mass emergency event, when the type of the event is not known at the training stage. We studied two ways to overcome the problem of data heterogeneity that affects the classifier in this situation: an ensemble classifier which combines predictions of classifiers specific to known types of disasters and a co-training method, which aims to reduce data heterogeneity by adding more train instances acquired from unlabeled messages in a bootstrapping manner.

In our experiments we studied three problems: detection of messages related to a disaster, detection of informative messages that contribute to situation awareness of first responders, and detection of first-hand accounts of the mass emergency events. We find that the first two tasks are relatively easy and good classification accuracy (an F-measure close to 80 or higher) can be achieved by a single generic classifier, even if the type of the disaster of the test data is not known in advance; ensemble or co-training methods did not offer any great advantage over the general classifier. The task of detecting eyewitness account proves much harder, but this is where the strengths of the ensemble and the co-training methods come to light: in comparison to using a general classifier, they both lead to significant gains in recall and F-measure.

Our results also suggest that there is considerable room for improvement for the both methods on the Eyewitness task. Thus, our future work will focus on exploring parameters of the ensemble and the co-training method, such as the effect of the amount of automatically added training data, as well as other semi-supervised techniques such as active learning, in the context of this classification scenario. The proposed classification methods will eventually be incorporated into a practical system for detection and monitoring of mass emergency events on social media and evaluate their utility within simulated real-world use cases, where their benefits will be measured in terms of the efficiency of semi-automated detection of emergency situations by first responders, ultimately enabling early warning and more expedient recovery operations.

8 References

[1] Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. Tweedr: Mining Twitter to inform disaster response. In Proc. of ISCRAM.

[2] Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond trending topics: Real-world event identification on Twitter. Proc. of ICWSM. 438–441.

[3] Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In Proceedings of COLT, pp.92–100.

[4] Cornelia Caragea, Nathan McNeese, Anuj Jaiswal, Greg Traylor, H. Kim, Prasenjit Mitra, Dinghao Wu, A. Tapia, Lee Giles, Bernard J. Jansen, and others. 2011. Classifying text messages for the Haiti earthquake. In Proc. of ISCRAM.

[5] Minmin Chen, Kilian Q. Weinberger, John C. Blitzer. 2011. Co-Training for Domain Adaptation. In Proc. NIPS-2011.

[6] Thomas G. Dietterich. 2001. Ensemble methods in machine learning. In Kittler, J., Roli, F., eds.: Multiple Classifier Systems. LNCS Vol. 1857, Springer (2001) 1–15.

[7] Yan-Shi Dong, Ke-Song Han. 2004. A comparison of several ensemble methods for text categorization. In Proc. IEEE International Conference on Services Computing. pp. 419-422.

[8] Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, Patrick Meier. 2013. Extracting Information Nuggets from Disaster-Related Messages in Social Media. In Proc. of ISCRAM.

[9] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. 2014. AIDR: Artificial intelligence for disaster response. In Proc. of WWW (Companion). IW3C2, 159–162.

[10] Rui Li, Kin Hou Lei, Ravi Khadiwala, and KC-C Chang. 2012. Tedas: A Twitter-based event detection and analysis system. In Proc. of ICDE. IEEE, 1273–1276.

[11] Benjamin Mandel, Aron Culotta, John Boulahanis, Danielle Stark, Bonnie Lewis, and Jeremy Rodrigue. 2012. A demographic analysis of online sentiment during hurricane Irene. In Proc. of the Second Workshop on Language in Social Media (LSM '12). Association for Computational Linguistics, Stroudsburg, PA, USA, 27-36.

[12] Huina Mao, Xin Shuai, Apu Kapadia. 2011. Loose Tweets: An Analysis of Privacy Leaks on Twitter. In Proc. WPES’11, Chicago, Illinois, USA.

[13] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. 2011. Twitinfo: Aggregating and visualizing microblogs for event exploration. In Proc. of CHI. 227–236.

[14] Aibek Musaev, De Wang, and Calton Pu. 2014. LITMUS: Landslide detection by integrating multiple sources. Proc. of ISCRAM.

[15] Vincent Ng and Claire Cardie. 2003. Bootstrapping Coreference Classifiers with Multiple Machine Learning Algorithms.

[16] Alexandra Olteanu, Sarah Vieweg, Carlos Castillo. 2015. What to Expect When the Unexpected Happens: Social Media Communications Across Crises. In Proceedings of the ACM 2015 Conference on Computer Supported Cooperative Work and Social Computing (CSCW '15). ACM.

[17] Viktor Pekar, Jane Binner, Hossein Najafi, Chris Hale. 2016. Selecting Classification Features for Detection of Mass Emergencies on Social Media. In Proc. of the International Conference on Security and Management. Las Vegas, NV.

[18] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: Real-time event detection by social sensors. In Proc. of WWW. ACM, 851–860.

[19] Tom De Smedt and Walter Daelemans. (2012). Pattern for Python. Journal of Machine Learning Research. 13, 2063-2067.

[20] Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, P. Ruhlen, S. Baker, and J. Crim. 2003. Bootstrapping statistical parsers from small datasets. In Proc. the EACL.

[21] Sudha Verma, Sarah Vieweg, William J. Corvey, Leysia Palen, James H. Martin, Martha Palmer, Aaron Schram, and Kenneth Mark Anderson. 2011. Natural language processing to the rescue? Extracting “Situational Awareness” tweets during mass emergency. In Proc. of ICWSM.

[22] Jie Yin, Andrew Lampert, Mark Cameron, Bella Robinson, and Robert Power. 2012. Using social media to enhance emergency situation awareness. IEEE Intelligent Systems 27, 6, 52–59.

[23] Andrea Zielinski and Ulrich Buegel. 2012. Multilingual Analysis of Twitter News in Support of Mass Emergency Events. In of the 9th International ISCRAM Conference – Vancouver, Canada, April 2012.