Identifying Informative Messages in Disaster Events using Convolutional Neural Networks

Cornelia Caragea
Computer Science and Engineering,
University of North Texas, Denton, TX
ccaragea@unt.edu

Adrian Silvescu
AS Research, Sunny Vale, CA
silvescu@gmail.com

Andrea H. Tapia
Information Sciences and Technology,
Pennsylvania State University, University Park, PA
atapia@ist.psu.edu

ABSTRACT

Social media is a vital source of information during any major event, especially natural disasters. Data produced through social networking sites is seen as ubiquitous, rapid and accessible, and it is believed to empower average citizens to become more situationally aware during disasters and coordinate to help themselves. However, with the exponential increase in the volume of social media data, so comes the increase in data that are irrelevant to a disaster, thus, diminishing peoples' ability to find the information that they need in order to organize relief efforts, find help, and potentially save lives. In this paper, we present an approach to identifying informative messages in social media streams during disaster events. Our approach is based on Convolutional Neural Networks and shows significant improvement in performance over models that use the "bag of words" and n-grams as features on several datasets of messages from flooding events.

Keywords

Informative tweets classification; disaster events; Convolutional Neural Networks.

INTRODUCTION

Much has been written concerning the value of using micro-blogging data from crowds of non-professional participants during disasters. Data produced through micro-blogging platforms, e.g., Twitter, is seen as ubiquitous, rapid and accessible (Vieweg, 2010), and it is believed to empower average citizens to become more situationally aware during disasters and coordinate to help themselves (Palen, Vieweg, and Anderson, 2010). Starbird, Palen, Hughes, and Vieweg (2010) assert that bystanders "on the ground are uniquely positioned to share information that may not yet be available elsewhere in the information space…and may have knowledge about geographic or cultural features of the affected area that could be useful to those responding from outside the area."

Despite the evidence of strong value to those experiencing a disaster and those seeking information concerning the disaster, there has been very little uptake of message data by large-scale, disaster response organizations (Tapia and Moore, 2014). Real-time message data being contributed by those affected by a disaster has not been incorporated into established mechanisms for organizational decision-making (Tapia, Moore, and Johnson 2013). Response organizations operate in conditions of extreme uncertainty. The uncertainty has many sources: the sporadic nature of emergencies, the lack of warning associated with some forms of emergencies, and the wide array of responders who may or may not respond to any one emergency. This uncertainty increases the need for appropriate information, which could make substantial improvements in the response process. We believe that data directly contributed by citizens and data scraped from disaster bystanders have a positive potential to give responders more accurate and timely information than it is possible by traditional information gathering methods. Still, information quality and use in any area of disaster response remains to be a challenge.

Through this research, we seek to find mechanisms to *automatically* identify the disaster-related Twitter posts (or tweets) that are informative in nature and to filter out those that are not informative to the disaster. Specifically, we formulate the problem as a classification problem and propose to use a *Convolutional Neural Network* approach for text classification to classify a tweet as either "informative" or "not informative" according to its information content. Table 1 shows examples of tweets extracted from one of the disasters in our dataset, i.e., Alberta flooding. The tweets are labeled as *informative* or *not informative*.

Tweet	Label
1. *" Shakespeare in the park. #abflood http://t.co/XW4Fn27tVy."*	*not informative*
2. *"Unreal situation with all the flooding in Calgary. Grateful that my home is safe."*	*not informative*
3. *"RT @weathernetwork: Insane photo of flooded parkade in Discovery Ridge via @GlobalCalgary: http://t.co/xAjppUJU6Y. #yyc #abflood."*	*informative*
4. *"RT @CalgaryPolice: Clarifying a rumour for #yyc. There are NO zoo animals being sheltered at the Courts. #yycflood ."*	*informative*

Table 1. Examples of tweets from the Alberta flooding labeled as informative or not informative.

A general approach for text classification is to use a learning model, e.g., Support Vector Machines (SVMs) or Naïve Bayes, on the "bag-of-words" (*tf* or *tf-idf*) representation of the documents. However, the word order from the text is lost and the performance can decrease for some tasks. To avoid this type of information loss, researchers proposed models that consider both unigrams as well as *n*-grams with $n>1$ (an *n*-gram is defined as a sequence of *n* contiguous words from a text). Unfortunately, this approach can increase the risk of over-fitting especially when the training set size is small. A recently introduced approach to text classification that is able to effectively make use of the word order in text is an adaptation of Convolutional Neural Networks (CNNs) from images to text data (Johnson and Zhang, 2015). CNNs for images are neural networks that make use of the 2-dimensional structure of image data through convolution layers (LeCun, Bottou, Bengio, and Haffner, 1986). In CNNs, each computation unit corresponds to a small patch from the input image. The analogous CNNs for text make use of the 1-dimensional structure of text data through convolution layers.

Contributions. We explore the application of CNNs for text classification to the task of identifying informative tweets during disaster events. The automated detection of informative data within micro-blogging platforms is still in its infancy. To our knowledge, we are the first to use state-of-the-art Artificial Intelligence technology, i.e., CNNs, to identify informative tweets in disasters. We show empirically on several real world flooding datasets that CNNs outperform SVMs and fully connected neural networks.

RELATED WORK

Micro-blogging has been under the lens of researchers with regards to its use in disasters and other high profile events (Dai, Hu, Wu, and Dai, 2014). However, in times of crises, micro-blogging can create a lot of noise in which stakeholders need to sift through to find relevant information. Machine learning and natural language processing have made great leaps in extracting, processing, and classifying social media feeds (Imran, Castillo, Diaz, and Vieweg, 2013a). For example, Mendoza, Poblete, and Castillo (2010) studied the propagation of rumors and misinformation from the Chilean earthquake using social media posts. Caragea, McNeese, Jaiswal, Traylor, et al. (2011) built models for classifying short text messages from the Haiti earthquake into classes representing people's most urgent needs so that NGOs, relief workers, people in Haiti, and their friends and families can easily access them. Dailey and Starbird (2014) explored techniques such as visible skepticism to help control the spread of false rumors. Li, Guevara, Herndon, Caragea, et al. (2015) used a domain adaptation approach to study the usefulness of labeled data from a source disaster, together with unlabeled data from a target disaster to learn classifiers for the target and showed that source data can be useful for classifying target data. Similarly, Imran, Elbassuoni, Castillo, Diaz, and Meier (2013b) explored domain adaptation for identifying information nuggets using conditional random fields and data from two disasters, Joplin 2011 tornado (as source) and Hurricane Sandy (as target). Caragea, Squicciarini, Stehle, Neppalli, and Tapia (2014) automatically classified the sentiment of users' posts during the Hurricane Sandy and studied the association of tweets' sentiments and their geo-locations.

Several works have particularly focused on identifying disaster-related information in Twitter. For example, Olteanu, Castillo, Diaz, and Vieweg (2014) built a lexicon for collecting and filtering micro-blogged tweets

from crisis events and showed that a crisis lexicon can improve the recall in detecting information relevant to a disaster. Moreover, Olteanu, Vieweg, and Castillo (2015) studied the type of information that is posted during different crisis events so that stakeholders know what information content to expect and what information sources are prevalent. The authors performed a statistical analysis of dependencies between types of crises and types of messages posted during these crises. In contrast, we use machine-learning techniques to identify information content in Twitter during crisis events. Similar to our work, there are a few other works that used machine learning for detecting useful information during crisis events. For example, Ashktorab, Brown, Nandi, and Culotta (2014) used a combination of classification, clustering, and extraction methods to extract actionable information for disaster responders. Imran, Elbassuoni, Castillo, Diaz, and Meier (2013c) trained classifiers to identify informative tweets in a dataset collected during the Joplin 2011 tornado, and subsequently classified the informative tweets into more specific types, such as *casualties and damage*, *donations, etc*. Finally, they extracted information nuggets such as location and time, for different types of tweets. Starbird and Palen (2010) studied information propagation in Twitter during mass emergencies through the re-tweet feature of Twitter, using North Dakota Red River floods and Oklahoma Wild fires. They mainly focused on the retweet aspect and analyzed the percentage of the retweets among the collected tweets to show that retweeting plays a major role in information sharing.

SUPERVISED LEARNING ALGORITHMS

We address the problem of identifying informative tweets during disaster events as a binary supervised classification problem, where the task is to predict if a tweet is informative (+ class) or not-informative (- class). We propose the use of Convolutional Neural Networks (CNN) and compare them with Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs), using unigrams, unigrams + bigrams, and unigrams + bigrams + trigrams. We review these models for the binary case in this section.

SVMs: SVMs are binary classification models, commonly used for text classification (Bishop, 2007). Given a set of labeled inputs $(\mathbf{x}_i, y_i)_{i=1,\ldots,l}$, $\mathbf{x}_i$ a feature vector and $y_i \in \{-1,+1\}$, learning an SVM is equivalent to learning a binary decision function $f(\mathbf{x})$ whose sign represents the class assigned to an input $\mathbf{x}$. This can be achieved by solving a quadratic optimization problem. During classification, $\mathbf{x}_{test}$ is classified based on the sign of the decision function, $sign(f(\mathbf{x}_{test}))$ (i.e., if $f(\mathbf{x}_{test})>0$, then $\mathbf{x}_{test}$ is assigned to the positive class; otherwise, $\mathbf{x}_{test}$ is assigned to the negative class). We used SVM with a linear kernel and its SVM^{Light} implementation.

ANNs: ANNs are classification models that are able to represent highly non-linear functions (Bishop, 2007). The ANNs have one input layer, one or more hidden layers and one output layer. Each layer has one or more neurons. The number of input neurons is equal to the dimension of the feature vector $\mathbf{x}$; the number of output neurons is equal to 1 for binary classification problems; the number of hidden neurons is an input parameter. The *activation* of each neuron i in a hidden or output layer j is given by $a_i^{(j)} = s(W_i^{(j-1)T} a^{(j-1)} + b)$, $j>2$, $a^{(1)}=\mathbf{x}$. s is a non-linear activation function, e.g., $s(x) = \frac{1}{1+e^{-w^T x}}$ (the sigmoid function) or $s(x) = \max(0, x)$ (the rectified linear units, ReLU). $W_i^{(j-1)T}$ is the i^{th} row of the weight matrix $W^{(j-1)}$, $a^{(j-1)}$ is the input to layer j, and b is the bias term. Figure 1 (left side) shows the architecture of a fully connected neural network with a single hidden layer. The network in the example has five input neurons (i.e., the dimensionality of the input space is five), three hidden neurons and one output neuron. A connection exists between any neuron in one layer to any neuron in the next layer (e.g., between any neuron from the input layer to any neuron from the hidden layer).

The weights $W^{(j-1)}$, for all j>2, of an ANN are learned using the backpropagation algorithm, which employs the gradient descent to minimize the sum of squared errors (L2 loss) between the network output values and the target (i.e., the actual) values for these outputs, over the training examples. During classification, for an input $\mathbf{x}_{test}$, the network returns the probability of $\mathbf{x}_{test}$ belonging to the positive class, $P(y=+1|\ \mathbf{x}_{test}) = s(W^{f-1)T} a^{(f-1)} + b)$, *f-1* is the layer before the final *f* (or output) layer, whereas $P(y=-1|\ \mathbf{x}_{test}) = 1 - P(y=+1|\ \mathbf{x}_{test})$.

Feature representation for SVMs and ANNs: For text classification, the input to SVMs and ANNs is often the "bag of words" or "bag of *n*-grams" vectors. A vocabulary is first constructed, which contains all unique words or *n*-grams in a collection of documents. A document is then represented as a vector $\mathbf{x}$ with as many entries as the words in the vocabulary. An entry i in $\mathbf{x}$ represents the frequency (in the document) of the i^{th} word or *n*-gram from the vocabulary, denoted by x_i. For each component in the vector $\mathbf{x}$, we used $\log(x_i + 1)$ and normalized $\mathbf{x}$ to unit vectors. Using these representations, we trained SVMs and ANNs classifiers. A general problem with the "bag of words" is that it does not preserve the word order, whereas the "bag of *n*-grams"

results in high data sparsity for large values of *n* and hence, it is neither efficient nor effective, especially when the size of the labeled dataset is small. The CNNs for text classification make use of the internal structure of the data (the word order) and internally learn features that are useful for classification.

CNNs: CNNs for text classification (Johnson and Zhang, 2015) consist of a sequence of one or multiple pairs of convolutional and pooling layers, which can be arranged in a stack or in parallel. The output layer returns a prediction based on features that are learned internally by previous layers. Each convolutional layer has a variable number of computational units, with each unit corresponding to an *n*-gram (also known as short region) from the input text. The weights in a convolutional layer are shared across all short regions. Specifically, for an input **x**, the "activation" of a unit in a convolution layer is given by $s(W^T r_l(x)+b)$, where $r_l(x)$ is a region vector corresponding to the l^{th} *n*-gram in **x**, and s is a non-linear activation function as in ANNs , e.g., the sigmoid logistic function or ReLU. Figure 1 (right hand side) shows the architecture of a convolutional neural network with only one pair of convolutional and pooling layers. The network has three computation units in the convolutional layer and one output unit. Examples of region vector representations $r_l(x)$ of region size 2 corresponding to the input tweet "pray for Alberta !" and assuming a vocabulary $\mathcal{V}$ given as follows $\mathcal{V}$ = {"floods", "crisis", "!", "Alberta", "for", "pray"} are "pray for:"{0,0,0,0,0,1|0,0,0,0,1,0}, "for Alberta:"{0,0,0,0,1,0|0,0,0,1,0,0}, and "Alberta !:"{0,0,0,1,0,0|0,0,1,0,0,0}.

The weights of CNNs are learned using the backpropagation algorithm as in ANNs. The algorithm employs the stochastic gradient descent to minimize the sum of squared errors (L2 loss objective function) between the network output values and the target (i.e., the actual) values for these outputs, over the training examples. During classification, for an input $\mathbf{x}_{test}$, the network returns the probability of $\mathbf{x}_{test}$ belonging to the positive class.

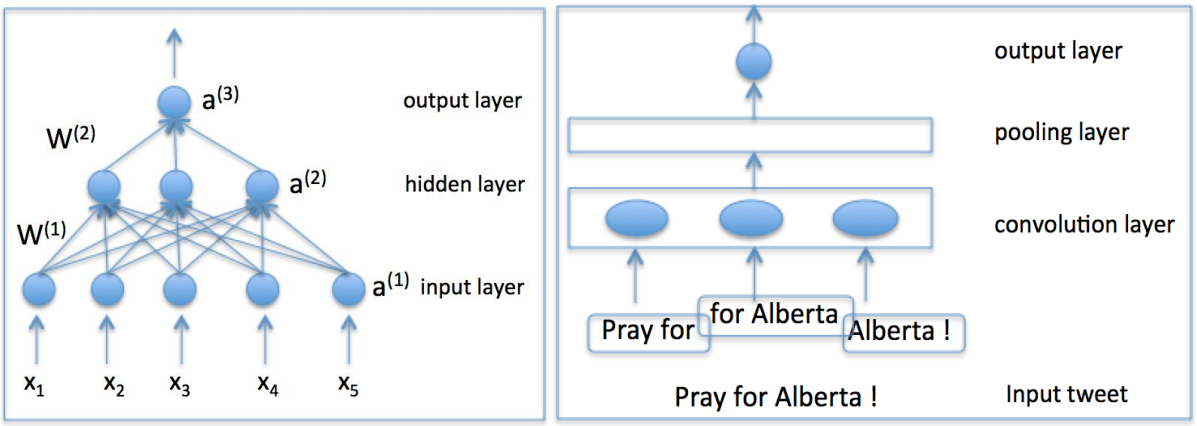

Figure 1. Fully connected neural network (left hand side); Convolutional Neural Network (right hand side).

In experiments, we used the implementation of ANNs and CNNs available online, [1] as described in (Johnson and Zhang, 2015). The number of neurons in the hidden layer of ANNs and the number of neurons (weight vectors) in the convolution layer of CNNs are both set to 1000 (based on a development set).

TWITTER DATA

In order to evaluate the effectiveness of CNN models for identifying informative tweets during crisis events, in our experiments we used a subset of the Twitter data available from the CrisisLex project.[2] Specifically, we used data from six flood events, which are available from the CrisisLexT26 collection (Olteanu et al., 2015). This collection contains tweets from 26 crises that are manually annotated by crowd-sourced workers with tweet informativeness (*informative* or *not informative*). There are about 1000 tweets manually annotated in each of the 26 crises. A summary of the data used in our experiments is shown in Table 2.

[1] riejohnson.com/cnn_data.html

[2] http://crisislex.org/

Disaster Name (Year)	Disaster Initial	Num. Positive	Num. Negative	Total
Philipinnes floods (2012)	P	761	145	906
Colorado floods (2013)	C	768	157	925
Queensland floods (2013)	Q	728	191	919
Sardinia floods (2013)	S	631	294	925
Alberta floods (2013)	A	684	297	981
Manila floods (2013)	M	628	293	921

Table 2. Summary of disaster data used in experiments.

EXPERIMENTS AND RESULTS

We compare results of experiments obtained using supervised classification based on CNNs with those obtained using supervised classification based on SVMs and ANNs. The SVM and ANN classifiers are trained on unigrams, unigrams + bigrams, and unigrams + bigrams + trigrams. In our experiments, we used the set of tweets from Philippines, Colorado, and Queensland floods as the training set, denoted by PCQ, the set of tweets for Manila floods as the development set, denoted by M, and the set of tweets from Alberta and Sardinia floods as two independent test sets, denoted by A and S, respectively. The development set was used to estimate model hyper-parameters, e.g., the number of neurons in a layer, or the value of *n* in *n*-grams. We report the accuracy on each test set independently, as well as the average classification accuracy of both test sets. We did not perform stemming, and did not remove stop-words or punctuation.

Table 3 shows the results of the comparison of CNNs of region size 2 with SVM classifiers, trained using three feature types: unigrams: SVM(1), unigrams + bigrams: SVM(2), and unigrams + bigrams + trigrams: SVM(3). The table shows also the results of the comparison of CNNs of region size 2 with ANN classifiers, trained using unigrams + bigrams: ANN(2). ANN(2) resulted in the highest performance among ANN(1) and ANN(3) (data not shown). As can be seen from the table, the CNNs outperform SVM and ANN classifiers trained using unigrams and *n*-gram features alone or in combination. This suggests that the CNNs effectively exploit the internal structure of the textual data to generate features that are used by the top layer to make predictions.

Train/Test	Naïve Approach	SVM(1)	SVM(2)	SVM(3)	ANN(2)	CNN(2)
PCQ/M	68.18	77.74	78.39	78.50	80.46	**82.52**
PCQ/S	68.21	70.59	71.24	71.57	74.49	**75.90**
PCQ/A	69.72	76.96	78.29	78.19	77.88	**79.31**
PCQ/S+A: Average performance	68.96	73.77	74.76	74.88	76.18	**77.61**

Table 3. Summary of disaster data used in experiments.

The performance of SVMs is generally lower than or very similar to the performance of ANNs on both test sets. For example, the best performance of SVM on Sardinia floods is 71.57% using unigrams + bigrams + trigrams, i.e., SVM(3), whereas the performance of ANN using unigrams and bigrams is 74.49%. A naïve approach that classifies every example in the majority class shows an accuracy of 68.21% on Sardinia floods. The CNN(2) outperforms both SVM and ANN generally by at least 1.5%. The fact that millions of tweets are posted during disaster events on social media sites, the 1.5% improvement in performance adds substantial value to using CNNs in disasters events to find informative messages and filter out not informative messages.

SUMMARY AND CONCLUSION

Previous research suggests that data gleaned from social media contributions have both significant value to emergency responders and are difficult to use. Responders seek an enhanced operational picture during any disaster, which grants them better situational awareness. The strongest contribution of this paper is the improvement in accuracy of identifying informative tweets during disaster events using state-of-the-art Artificial Intelligence (AI) technology, i.e., Convolutional Neural Networks (CNNs). We showed that the CNNs are able to predict the informative tweets and filter out the tweets that are not informative in nature. In time, such AI technologies could pinpoint the joy of having survived a falling tree, the horror of a bridge washing out or the

fear of looters in action. This is one strong step along the path to providing official responders with truly actionable information in real time based on social media data. Using domain adaptation techniques in conjunction with Convolutional Neural Networks would be an interesting future direction to pursue.

ACKNOWLEDGMENTS

We thank the National Science Foundation for support from the grants IIS #1526542 and IIS #1526678 to Cornelia Caragea and Andrea Tapia. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the National Science Foundation. We also wish to thank our anonymous reviewers for their constructive comments.

REFERENCES

1. Ashktorab, Z., Brown, C., Nandi, M., Culotta, A. (2014) Tweedr: Mining Twitter to Inform Disaster Response. In: Proceedings of ISCRAM 2014, 354-358, University Park, PA.
2. Bishop, C. (2007). Pattern Recognition and Machine Learning. Springer. 2007.
3. Caragea, C., McNeese, N., Jaiswal, A., Traylor, G., Kim, H.-W., Mitra, P., Wu, D., … Yen, J. (2011). Classifying Text Messages for the Haiti Earthquake. In: ISCRAM 2011, Lisbon, Portugal.
4. Caragea, C., Squicciarini, A., Stehle, S., Neppalli, K., & Tapia, A. Mapping Moods: Geo-Mapped Sentiment Analysis during Hurricane Sandy. In: ISCRAM 2014, University Park, Pennsylvania, USA.
5. Castillo, C., Mendoza, M., & Poblete, B. (2011). Information Credibility on Twitter. WWW '11 Proceedings of the 20th international conference on World Wide Web (pp. 675–684). ACM.
6. Dai, W., Hu, H., Wu, T., and Dai, Y. (2014) "Information Spread of Emergency Events: Path Searching on Social Networks." The Scientific World Journal Volume 2014 (2014).
7. Dailey, D., & Starbird, K. (2014). Visible Skepticism: Community Vetting after Hurricane Irene. In Proceedings of the 11th International ISCRAM Conference. University Park, Pennsylvania, USA. 777 - 781.
8. Imran, M., Castillo, C., Diaz, F., Vieweg, S. (2013a). Processing Social Media Messages in Mass Emergency: A Survey. Journal of the ACM Computing Surveys (CSUR) Volume 47 Issue 4, July 2015, Article No. 67.
9. Imran, M., Elbassuoni, S., Castillo, C., Diaz, F. and Meier, P. (2013b) Practical Extraction of Disaster-Relevant Information from Social Media. In: Proceedings of WWW 2013, 1021-1024, Rio de Janeiro, Brazil.
10. Imran, M., Elbassuoni, S., Castillo, C., Diaz, F. and Meier, P. (2013c) Extracting Information Nuggets from Disaster-Related Messages in Social Media. In: Proceedings of the ISCRAM 2013, 791-800.
11. Johnson, R., and Zhang, T. (2015). Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In Proceedings of NAACL 2015.
12. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. Gradient-based learning applied to document recognition. In Proceedings of the IEEE, pages 2278–2324.
13. Li, H., Guevara, N., Herndon, N., Caragea, D., Neppalli, K., Caragea, C., Squicciarini, A., Tapia, A. (2015). Twitter Mining for Disaster Response: A Domain Adaptation Approach. In: ISCRAM 2015.
14. Mendoza, M., Poblete, B., & Castillo, C. (2010). Twitter under Crisis: Can we trust what we RT? New York (pp. 71–79). ACM Press.
15. Olteanu, A., Vieweg, S., Castillo, C. 2015. What to Expect When the Unexpected Happens: Social Media Communications Across Crises. In CSCW '15. ACM, Vancouver, BC, Canada.
16. Olteanu, A., Castillo, C., Diaz, F., Vieweg, S. 2014. CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises. In: ICWSM'14, AAAI Press, Ann Arbor, MI, USA.
17. Palen, L., Vieweg, S., & Anderson, K. M. (2010). Supporting "Everyday Analysts" in Safety- and Time-Critical Situations. The Information Society, 27(1), 52–62.
18. Starbird, K., and Palen, L. (2010). Pass It On? : Retweeting in Mass Emergency. In Proceedings of the 7th ISCRAM Conference, Seattle, WA.
19. Starbird, K., Palen, L., Hughes, A. L., & Vieweg, S. (2010). Chatter on the Red: What Hazards Threat Reveals About the Social Life of Microblogged Information. In: CSCW '10 (pp. 241–250).
20. Tapia, A. and Moore, K. (2014), Good Enough is Good Enough: Overcoming Disaster Response Organizations' Slow Social Media Data Adoption Special Issue on Technologies for Disaster Response. Journal of Computer

Supported Cooperative Work. Online First, Springer.

21. Tapia, A. Moore, K. Johnson, N. (2013) "Beyond the Trustworthy Tweet: A Deeper Understanding of Microblogged Data Use by Disaster Response and Humanitarian Relief Organizations", In: ISCRAM.
22. Vieweg, S. (2010). Microblogged Contributions to the Emergency Arena: Discovery, Interpretation and Implications. CSCW 2010, February 6-10 (pp. 515–516). Savanah, GA: ACM.