Information Abstraction from Crises Related Tweets Using Recurrent Neural Network

Mehdi Ben Lazreg, Morten Goodwin, Ole-Christoffer Granmo

Information Abstraction from Crises Related Tweets Using Recurrent Neural Network

Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo

University of Agder, Grimstad, Norway,
{mehdi.ben.lazreg, morten.goodwin, ole.granmo}@uia.no,
http://www.uia.no

Abstract. Social media has become an important open communication medium during crises. The information shared about a crisis in social media is massive, complex, informal and heterogeneous, which makes extracting useful information a difficult task. This papers presents a first step towards an approach for information extraction from large Twitter data. In brief, we propose a Recurrent Neural Network based model for text generation able to produce a unique text capturing the general consensus of a large collection of twitter messages. The generated text is able to capture information about different crises from tens of thousand of tweets summarized only in a 2000 characters text.

Keywords: information abstraction, recurrent neural network, twitter data, crisis management

1 Introduction

Social media has become the de facto open crises communication medium [1]. It plays a pivotal role in most crises today, from getting life signs from people affected to communicating with responders [2]. However, processing and extracting useful information and inferring valuable knowledge from such social media messages is difficult for several reasons. The messages are typically brief, informal, and heterogeneous (mix of languages, acronyms, and misspellings) with varying quality, and it may be required to know the context of a message to understand its meaning. Moreover, people also post information on other mundane events, which introduces additional noise into the data.

The state-of-the-art in the area of information discovery using machine learning mostly centres on supervised learning techniques. Those techniques are based on training an algorithm on sets of text from each topic to learn a predictive function, which in turn is used to classify new texts into a previously learnt topic [3]. A limitation of this approach is the scope of the topics: If a new text about an unforeseen topic is presented to the algorithm, such as a new crisis, it will wrongly classify it as one of the existing topics. Another challenge is that crises are diverse and the number of topics discussed in social media during a single crisis is large, dynamic, and changing from crisis to crisis. Moreover, applying a classifier trained on data from previous disasters on the next disaster may

not perform well in practise. This can be explained by the fact that the next disaster will typically be more or less unique compared to the previous ones. Accordingly, a loss of accuracy occurs even if the crises have many similarities. Alternatively, unsupervised techniques try to look for co-occurrences of terms in the text as a metric of similarity [4] or infer the word distribution of a set of words the text contains and use it for document clustering [5]. Moreover, different methods based on graph theory has been used to extract information from a document. As an example, a graph based ranking model for document processing was adopted to extract key words from a text document [6]. In addition, a stochastic graph based method has also been employed to extract the most important sentence in a text document [7].

Recurrent Neural Networks and its long-short term memory variant have emerged as an efficient model in a variety of application involving sequential data [8]. This includes handwriting recognition [9], speech recognition [10], and video analysis [11]. As an example, RNN was trained on Wikipedia articles for text generation with great success. The power of recurrent neural network comes from their high dimensional hidden state with non-linear dynamics which has the ability to remember and process past input information [12]. The goal of this paper is to make a model that summarises and reproduces content from massive Twitter streams. The model is based on recurrent neural network to predict the next character in a stream of text. The approach allows to generate a text that compresses the information contained in the text that the network has been trained on.

This paper is organised as follow. Section 2 gives an overview of the state-of-the-art twitter analysis in crises situations. Section 3 introduces recurrent neural network and illustrates its basic features. Section 4 proposes a recurrent network based model for topic discovery in crisis related twitter data. Section 5 presents tests and results of the model. Finally, Section 6 concludes and provides pointers to further work.

2 Twitter analysis for crisis situations

There is no doubt that valuable, high throughput data is produced on social media only seconds after a crisis occurs [1]. To cope with the complexity of the social media data, and extract information from crisis related messages, machine learning techniques have been applied [2]. Two main approaches were investigated: supervised and unsupervised learning.

In a supervised approach the goal is to classify a social media message as part of one particular crisis event. To achieve this, the algorithm learns a predictive function so that it can classify any new unknown message as part of one of the categories of crises. A number of approaches have been investigated including Naïve Bayes, Support Vector Machine (SVM) [13], Random Forests [14], and Logistic Regression [15]. Further, some research focus on only analysing tweets containing certain keywords [2]. In this way, the tags can replace manual labelling for training. As an example, SVM was used to classify tweets related to

earthquakes and landslides [3, 16]. In a supervised approach, labels are necessary for training the classifiers, but they might be highly difficult to obtain especially in the case of multi-language messages or context knowledge [2]. To address this problem, unsupervised learning techniques are used.

Unsupervised methods are used to identify patterns in unlabelled data. They are most useful when the information seekers do not know specifically what information to look for in the data –which is the case in many crises situations. An example is grouping tweets into stories (clusters of tweets) after a keyword filter [17]. This method reduces the number of social media messages to be handled by humans since it groups equivalent messages together. Another application using unsupervised learning identifies events related to public and safety using a spatio-temporal clustering approach [18]. In addition to strictly clustering elements into groups, soft clusters have been used to allow items to simultaneously belong to several clusters with variant degrees. In this approach, the tweets similarity is based on words they contain and the length of the tweets [19]. The approach was applied on the Indonesia earthquake (2009) data and detected different aspects related to the crisis (relief, deaths, missing persons, and so on).

3 Artificial Neural Networks

An Artificial Neural Network (ANN) is a machine learning algorithm developed to mimic the human brain and reach its information processing capabilities [20]. An ANN is a network of processing units (analogous to neurons) joined by weighted connections (analogous to synapse). The network is activated by giving an input to some or all of the units. This activation is then spread throughout the hidden layers of the network until it reaches the output layer (see Figure 1).

Many varieties of ANN exist all with different sets of properties [20]. One major distinction is between networks where the connections are acyclic called Feedforwad Neural Networks (FNN) and networks where the connections form cycles called Recurrent Neural Networks (RNN). RNN is most suited for tasks that involve sequential input such as speech and language. A RNN processes an input sequence one element at a time, and maintains information about the history of the past elements in the sequence. This ability makes it suitable for learning patterns form text since a text is a series of correlated characters. RNN is successfully used to predict the next word in a sequence of semantically related words [8]. It also has some success in predicting the next character in a sequence of characters which is used to generate text, and in machine translation [8].

3.1 Recurrent Neural Networks

A Recurrent neural network (RNN) is an ANN where the connections between neurons are allowed to form a circle (see Figure 1) [20]. As Figure 1 shows, the connections between units on the same layer allow mapping the history of previous inputs to the output vectors. For each unit k in the RNN, the activation

a_k of that unit depends on the inputs $\{x_1, x_2, ..., x_n\}$ of the unit and the weights $\{w_1, w_2, ..., w_n\}$ of their respective connections as shown in Equation 1.

$$a_k = f(\sum_{i=1}^{n} w_i x_i + b_k) \tag{1}$$

The most widely used activation functions are sigmoid, hyperbolic tangent (in this case the unit is called a logistic unit) and linear functions (in this case the unit is called linear unit) [20]. b_k is a bias term that represents the expected mean value of the activation when all the inputs are zeroes.

During the training phase of an RNN, the aim is to update the weights so that for a given input, the output produced minimises a loss function that measures the similarity between the output of the network and the desired output [20]. The training of a RNN goes through three major steps [20]:

1. Initialise the weights w_i to a generally small value (in the range $[-0.1, 0.1]$).
2. Forward pass: Computes the activations a_k of all the unit in the RNN.
3. Backward pass: Updates the weights of the network in a manner that minimises the loss function between the output of the RNN and the desired output. This is performed using gradient descent. Backpropagation is used to efficiently compute the gradient and update the weights.

The three steps are repeated until a minimum of the loss function is reached. Note that the solution converged to may actually represent a local minimum.

3.2 Long-short term memory

The main benefit of a RNN is its ability to use the input at previous time steps to produce an output. Nevertheless, in a standard RNN the range of past inputs that can influence output is quite limited because gradients can either decay or blow up exponentially as feedback cycles around the network recurrent connections [20]. This problem is known as the vanishing/exploding gradient problem [21]. To address this problem a Long-Short Term Memory (LSTM) architecture was proposed [9]. In a LSTM architecture, the unit in the hidden layer of a RNN is replaced by a block (see Figure 2) analogous to a memory block. The block is composed of :

- A linear unit "c_t" presenting the state of the block a time t.
- A logistic input unit "i_t": analogous to a write gate that updates the value in "c_t" when on (outputs a value close to 1)
- A logistic output unit "o_t": analog to a read gate that retrieves the value in "c_t" when on.
- A logistic forget unit "f_t": analog to a keep gate that maintains the value in "c_t" when on.

The state of units "i_t", "o_t" and "f_t" are updated based on the input x_t, the output of the block at the previous time step h_{t-1}, and the output of other blocks in the LSTM network. This architecture is proved successful at a range of tasks that require long range memory including text generation, speech recognition and handwriting recognition [8].

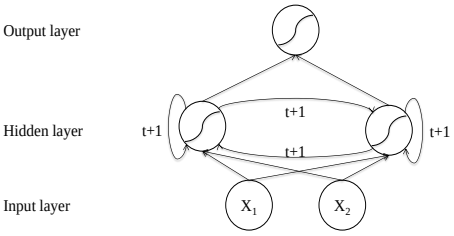

Fig. 1. Example of a RNN.

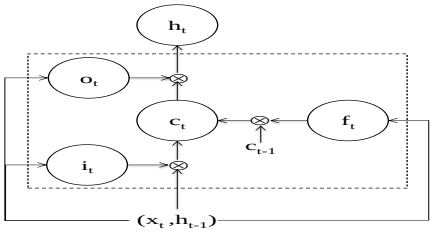

Fig. 2. A block of LSTM architecture.

4 Approach

Our approach aims at learning patterns form crises related tweets, and later generate a compressed text deducing the main topic (see Figure 3). This means that the model learns characteristics of tweets, understands the context and is able to reproduce similar tweets. Note that this is very different from reciting tweets in that the model has learnt concepts. It does not copy Twitter text. It is designed to capture the main concept even with a noisy set of tweets i.e. a collection of tweets about different topics. However, the model does not aim at presenting a comprehensive assessment of the crisis at this point. It only present fragments of the crisis present in the generated text. To achieve this goal, we train a character based LSTM architecture on crises related tweets.

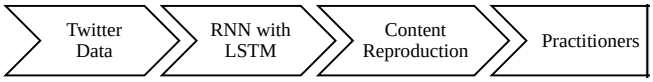

Fig. 3. A hight level overview of the proposed model.

The used LSTM architecture contains multiple hidden layers, each containing multiple LSTM blocks presented in Figure 2. The input of the network is a vector representing the current character x_t where the character can be a letter or a special character. The output node of the network h_t is a softmax distribution over characters [20]. The softmax function produces an output in the $[0, 1]$ interval that represents the probability of the next character given the input of the node. For a training set $\{(x^1, y^1), (x^2, y^2), .., (x^n, y^n)\}$, the softmax distribution is defined by Equation 2.

$$p(y^i = k|x^i) = \frac{\exp(\theta_i x^i)}{\sum_{j=1}^{K} \exp(\theta_j x^i)} \tag{2}$$

Where $x^i \in \mathbb{N}^K$ is vector coding of the input of the softmax (note that in our case the input of the softmax is the output of the hidden layers of the LSTM architecture), $y^i = \{1, 2, .., K\}$ is the index of the output character, K

is the number of possible characters. θ_i is the parameter of the softmax to be determined during the training phase to minimize the loss function in Equation 3. The loss function represents the sum of the negative log-likelihood of y^i knowing x^i. By minimizing the loss function, the probabilities that the correct character is predicted approaches one.

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} 1\{y^i = j\}\log(p(y^i = j|x^i)) \tag{3}$$

$$1\{y^i = j\} = 1 \text{ if } y^i = j \text{ ; } 0 \text{ otherwise} \tag{4}$$

The hypothesis is that the generated text should provide a description of the tweets the network has been trained on. Hence, a practitioner can easily get an overview of the underlying topics of the tweets.

5 Tests and results

The model described in the previous section was used to learn patterns from Twitter data related to several crises and then generate a unique text containing information present in the initial tweets.

5.1 The data

The crises data used to train our model is provided by *CrisisLex* [22] a platform for collecting and filtering communications during a crisis. Table 1 refers to more than fifty four thousand tweets about several crises. The data is a mix of tweets where some are related to the respective crisis and some are not. The percentage of unrelated tweets for each crisis ranges from 38% to 44% of the whole set of Twitter messages. Moreover, in the case of the Alberta flood, only 30% out of the related tweets gave concrete useful information about the crisis. The remaining tweets include other mundane topics. The percentage of informative tweets out of the related tweets goes up to a maximum 48% in the Queensland flood data.

Table 1. Crises related twitter data

Crisis	Year	Number of tweets	Percentage of unrelated tweets
Boston bombing	2013	11012	40%
Texas explosion	2013	11006	44%
Alberta flood	2013	11036	44%
Hurricane Sandy	2012	10008	38%
Queensland flood	2011	11233	43%

5.2 Experiments

The network was trained on a Twitter data collect from several crises (see Section 5.1). Table 2 summarises the empirical results of the model tested with different setups. The performance of the model can be measured with the training loss indicating the difference between the predicted and true value during the training period (Equation 3), and the validation loss indicating the difference between the predicted and true value over a validation data (additional data over which the model is applied after training).

Table 2. Experiments results

Parameter	Value	Training loss	Validation loss
Architecture	RNN	1.97	1.81
	LSTM	1.48	1.50
Number of nodes	128	1.81	1.65
	256	1,48	1.50
	512	1.17	1.47
Number of layers	1	1.53	1.57
	2	1.48	1.50
	3	1.56	1.49
Dropout	0	0.80	1.90
	0.50	1.48	1.50
	0.90	2.00	1.90
Batch size	50	1.48	1.61
	100	1.48	1.50
	500	1.66	1.62
	1000	1.85	1.79
Sequence length	30	1.52	1.48
	50	1.48	1.50
	140	1.49	1.51

The first conducted experiment was to show the improvement in training and validation loss the LSTM brings over a simple RNN. Table 2 shows that the validation loss drops from 1.81 for a simple RNN to 1.5 for LSTM. Similarly, by showing the evolution of the training loss over the amount of train data for RNN and LSTM architectures, Figure 4 illustrates the margin in training loss between LSTM and RNN even for small amount of training data. The training loss ends at a value of 1.97 for RNN and 1.48 for LSTM.

The LSTM architecture possesses different parameters that can be tuned to improve the model's ability to learn patterns from the training set and predict the next character in a sequence. An important parameter is the number of units (or nodes) in the network. Table 2 shows that increasing the number of nodes in the network improves the validation loss from 1.65 with 128 nodes to 1.47 with 512 nodes. Figure 5 displays the same trend: A 512 units network reaches

a training loss of 1.17 while the loss for 128 units network remains at 1.81. However, with 512 nodes the validation loss is significantly higher then training loss strongly indicating that the network is over-fitting the data. Over-fitting will cause the generated text later on to be a copy of the tweets existing in the training set which will present no value added. Moreover, using 512 nodes will require more processing time for little gain in validation loss over 256 nodes (1.5 for 256 units and 1.47 for 512 units). An equilibrium is reached at 265 units where the validation loss 1.5 is slightly higher then the training loss 1.48.

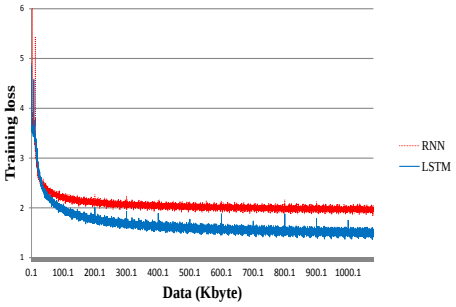

Fig. 4. Training loss for RNN and LSTM architectures.

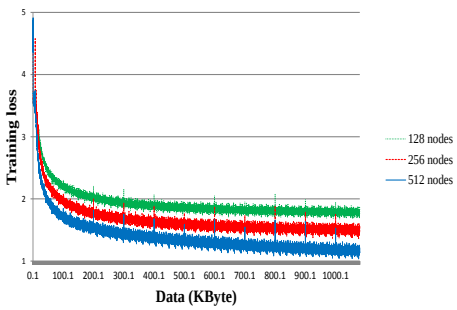

Fig. 5. Training loss for different number of nodes.

Another parameter of LSTM is the number of hidden layers in the network. Table 2 suggests that increasing the number of hidden layers slightly improves the validation loss from 1.57 for one layer to 1.5 and 1.49 for 2 and 3 layers respectively. Nevertheless, the validation loss barely changes between 2 and 3 hidden layer. Likewise, the training loss is not greatly influenced by the number of layers going from 1.53 to 1.48 and 1.56 for 1, 2 and 3 layers respectively.

The dropout intends to avoid over-fitting by dropping out each node in the network with a certain probability at each training step [23]. Table 2 shows that 0 dropout improves the training loss to 0.8 compared to 2 for a dropout of 0.9. Nevertheless, the validation loss remains high, which again indicates over-fitting. In contrast, a high dropout causes both losses to be high (training loss of 2 and a validation loss of 1.9) which suggest under-fitting. Under-fitting means that the model cannot capture data patterns and fails to fit the data well enough.

The batch size determines how many examples the model looks at before making a weight update [24]. Lower batch sizes should intuitively improve the validation and training loss. However, the change is no longer compelling after reaching an batch size of 100. As Table 2 shows, the training loss goes from 1.85 to 1.48 for a change of batch size from 1000 to 100 and stays at 1.48 for a batch size of 50. Similarly, the validation loss goes from 1.79 to 1.5 for a change of batch size from 1000 to 100 but then increases to 1.61 for a batch size of 50.

The sequence length is the maximum number of characters that remains in the network memory to perform a prediction. Our experiments tested 3 values:

the most frequent length of tweets (30 characters), the average length of tweets (50 characters) and the maximum length of tweets (140 characters). The results presented in Table 2 show an improved validation loss with shorter sequence moving from 1.51 to 1.48 for sequence lengths of 140 and 30 respectively. The model starts degrading in fitting the training data for shorter sequence shown by an increase of training loss from 1.49 to 1.52. The best combination of validation and training loss for 50 sequences.

We will in the remainder of the paper apply the configuration and parameters that provide the best performance above. The best setup consists of an LSTM architecture with 2 hidden layers and 256 hidden nodes, which represents approximately 400 thousand parameters to train. We used a dropout of 0.5, a batch size of 100, and the network keeps a memory of last 50 characters to use in its predictions.

5.3 Results

The model was successful in generating a 2000 character text for each crisis. A sample of the generated text is presented in Table 3. The aim is to generate tweets which are concise, explanatory, and extract the main topic of the big twitter data. The generated text is unique and only generated by the model. This means that all generated text is different and is not contained in the training data. Hence, from a structural point of view, the model was able to learn the basic component of a tweet: RT (retweets), hashtags, the "@" to address a specific person and the hypertext links. It was also able to predict an open bracket following two points, :(, for a sad smiley face in the first hurricane Sandy related text.

From a content point of view, the text contains misspelling and the sentences are unstructured. However, they clearly present valuable information that exists in the training data. An example is the Boston bombing, the first tweets clearly indicates the event of a bomb at the Boston marathon and presents a name: "Jeff Bauran". The name is actually a misspelling of "Jeff Bauman", a witness who identified one of the attackers [25]. The second text indicates that the FBI released a video about the Bombing. What actually happened, and present in the training data, is that the FBI released pictures and videos of the attackers [25]. An arrest was also made as the training data suggest and this was also captured by the model in the third tweet.

For the Texas explosion in West Fertilizer Company, the model was able to capture the number 60 surrounded by "killed" and "injured" in the first tweet. Actually, in the training data, this number appears sometimes as the number of killed in the explosion and other times as the number of injured. It is worth noticing that the number of deaths declared by the authorities was 14 [26]. However, this number was not part of the training Twitter data. This is an example of misleading information that can be present in the generated text cause by people spreading rumors through Twitter. The second tweet is unrelated to the crises and handles mundane topic present in the training data. The same applies to the third tweet about Alberta flood. Moreover, the generated tweets about the Texas explosion do not explicitly state the nature of the crises.

Table 3. Generated text

Crisis	Generated tweets
Boston bombing	1. Jeff Bauran, was bomb at #Boston Marathon and rew for the Boston marathon. My thoughts and prayers go out to everyone affected by today. 2. RT @CSDiggren: Wook, suspect in Boston Marathon bombings: FBI releases video of the #BostonMarathon bombing: Any are peosled of sole. Same name gun... 3. RT @SportsCenter: BREAKING: An arrest has been made in the Boston Marathon bombings, CNN reports.
Texas explosion	1. ?earcharliess killed 60 injured) http://t.co/jaKhPJfNND 2. he's saying we won't knight. 3. Eeproike for ouf the texas #PrayForWest PHOTO: Fertilizer plant explosion httexas. I gagrings! Thoughts To fertilizer plant explosionor the mesting, #WestTXClr4MwWUp
Alberta flood	1. now video all the stay safe. They work to help the flood victims to help. Everyone pock the floods to my content for calgary floods: RT @nenshi: 2. This pic of an awesome firefighter in Mission needs to go viral. Please share! #abflood http://t.co/x1Y 3. #job #hiring http://t.co/MDEIzdl5lN
Hurricane Sandy	1. nice in plays of Hurricane Sandy election to some going hurricane sandy. Tomorrow :(2. school destroyed http://t.co/nIsP2NJK
Queensland flood	1. Queensland's flood crisis deepens - The Australian: Sydney Morning 2. Photos: Flood water rises in Australia: http://t.co/eFEOl

This might be due to the fact that the training data related the crisis present the highest percentage of unrelated tweets which influence the generated text. Therefore, we tried to remove the unrelated tweets from the train data related to the Texas explosion, retrain the model and generate a new text. The last tweet about the Texas explosion represents a sample of what we obtain. Even though the nature of the crisis is explicitly stated in the tweet, the tweet contains much more misspellings and unstructured. This is caused by reducing the training data (eliminating unrelated tweets), the model had not enough data to capture the structure and learn words.

For the remaining crises in the data, the tweets indicate the type of crisis, its location and provide some update on it status like the school destroyed during

hurricane sandy, and the deepening of the Queensland flood. Note that some tweets are related to the crises but do not provide value added information about its status, like the firefighters mission in the Alberta flood second tweet. Nevertheless, these tweets present a significant chunk of the training data (see Section 5.1) linked to the crisis but not presenting useful information about it.

When we tried to generate the text anew, the information present in the new text was similar to previously discussed text. It is also worth noticing that some valuable information was present in the training data that was not extracted by the generated text which could be an area of further improvement in addition to automatically displaying the text in a manner that improves situational awareness further.

6 Conclusion

During a crises, valuable and substantial information about the crisis is shared on social media. However, the complexity and heterogeneity of tweeted text render extraction of useful information a difficult task for machine learning techniques. This papers presents a first step towards an approach for information extraction from large Twitter data collections. We propose training an LSTM architecture on crises related tweets and then use the trained model to generate a novel text. The model is able to capture valuable information from tens of thousands of tweets summarized only in an adjustable 2000 character text. Work is still to be made to filter useful crises related information and present it automatically in a more intuitive manner. Another future direction is to adopt a diversification mechanism that uses scores to rank the generated text based on their similarities and the information they bring.

References

1. Zielinski, A., Middleton, S.E., Tokarchuk, L.N., Wang, X.: Social media text mining and network analysis for decision support in natural crisis management. Proc. ISCRAM. Baden-Baden, Germany (2013)
2. Imran, M., Castillo, C., Diaz, F., Vieweg, S.: Processing social media messages in mass emergency: a survey. ACM Computing Surveys (CSUR) **47**(4) (2015)
3. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web. (2010)
4. Dumais, S.T.: Latent semantic analysis. Annual review of information science and technology **38**(1) (2004)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. the Journal of machine Learning research **3** (2003)
6. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. Conference on Empirical Methods in Natural Language Processing (2004)
7. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research (2004)
8. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553) (2015)

9. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on **31**(5) (2009)
10. Farahat, M., Halavati, R.: Noise robust speech recognition using deep belief networks. International Journal of Computational Intelligence and Applications **15**(1) (2016)
11. Kahou, S., Michalski, V., Konda, K., Memisevic, R., Pal, C.: Recurrent neural networks for emotion recognition in video. ICMI 2015 - Proceedings of the 2015 ACM International Conference on Multimodal Interaction (2015)
12. Karpathy, A., Johnson, J., Li, F.F.: Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078 (2015)
13. Yin, J., Lampert, A., Cameron, M., Robinson, B., Power, R.: Using social media to enhance emergency situation awareness. IEEE Intelligent Systems **27**(6) (2012)
14. Imran, M., Castillo, C., Lucas, J., Meier, P., Vieweg, S.: Aidr: Artificial intelligence for disaster response. In: Proceedings of the companion publication of the 23rd international conference on World wide web companion. (2014)
15. Ashktorab, Z., Brown, C., Nandi, M., Culotta, A.: Tweedr: Mining twitter to inform disaster response. Proc. of ISCRAM (2014)
16. Musaev, A., Wang, D., Pu, C.: Litmus: Landslide detection by integrating multiple sources. In: 11th International Conference Information Systems for Crisis Response and Management (ISCRAM). (2014)
17. Rogstadius, J., Vukovic, M., Teixeira, C., Kostakos, V., Karapanos, E., Laredo, J.A.: Crisistracker: Crowdsourced social media curation for disaster awareness. IBM Journal of Research and Development **57**(5) (2013)
18. Berlingerio, M., Calabrese, F., Di Lorenzo, G., Dong, X., Gkoufas, Y., Mavroeidis, D.: Safercity: a system for detecting and analyzing incidents from social media. In: Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on, IEEE (2013)
19. Kireyev, K., Palen, L., Anderson, K.: Applications of topics models to analysis of disaster-related twitter data. In: NIPS Workshop on Applications for Topic Models: Text and Beyond. Volume 1., Canada: Whistler (2009)
20. Graves, A.: Supervised sequence labelling. Springer (2012)
21. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. arXiv preprint arXiv:1211.5063 (2012)
22. Olteanu, A., Castillo, C., Diaz, F., Vieweg, S.: Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In: ICWSM. (2014)
23. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1) (2014)
24. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.R.: Efficient backprop. In: Neural networks: Tricks of the trade. Springer (2012)
25. FBI, B.: Updates on investigation into multiple explosions in boston. https://www.fbi.gov/news/updates-on-investigation-into-multiple-explosions-in-boston (2013)
26. Gillam, C., MacLaggan, C.: Ammonium nitrate stores exploded at texas plant: state agency. http://www.reuters.com/article/us-usa-explosion-texas-idUSBRE9460GP20130507 (2013)