On Semantics and Deep Learning for Event Detection in Crisis Situations

Grégoire Burel, Hassan Saif, Miriam Fernandez, and Harith Alani

Knowledge Media Institute, The Open University, United Kingdom
{g.burel, h.saif, m.fernandez, h.alani}@open.ac.uk

Abstract. In this paper, we introduce Dual-CNN, a semantically-enhanced deep learning model to target the problem of event detection in crisis situations from social media data. A layer of semantics is added to a traditional Convolutional Neural Network (CNN) model to capture the contextual information that is generally scarce in short, ill-formed social media messages. Our results show that our methods are able to successfully identify the existence of events, and event types (hurricane, floods, etc.) accurately ($>$ 79% F-measure), but the performance of the model significantly drops (61% F-measure) when identifying fine-grained event-related information (affected individuals, damaged infrastructures, etc.). These results are competitive with more traditional Machine Learning models, such as SVM.

Keywords: Event Detection, Semantic Deep Learning, Word Embeddings, Semantic Embeddings, CNN, Dual-CNN.

1 Introduction

Social media has emerged as a dominant channel for communities to gather and spread information during crises. Such media has proven itself as an invaluable information source in several recent natural and social crisis situations, such as floods [26], earthquakes [21], wildfires [29], nuclear disasters [28], and civil wars [4].

A survey by the American Red Cross showed that 40% of the population would use social media during a crisis, and 76% of them expect their help requests to be answered within three hours. Doing this through manual analysis, however, is far from trivial, due to the sheer data volumes and velocity. For example, in a single day during the 2011 Japan earthquake, 177 million tweets related to the crisis were sent [5].

Although information is paramount during such major crises, it is almost impossible for organisations and communities to manually absorb, process, and turn the sheer volume of social media data during crisis into sensible, actionable information [10]. Tools to automatically identify the type of emergency events reported by citizens (e.g., need shelter, trapped in building) are largely unavailable. Genuine help requests are often difficult to spot, group and validate, and many urgent aid requests by individual citizens could go unnoticed.

Several works exist in the literature that focus on detecting general and global events and themes from social media (floods, wildfires, bombings, etc.). However, the automatic identification of fine-grained emergency-related information [19] (e.g., affected individuals, infrastructure, etc.) is still in its infancy.

Current works for event identification from social media data make use of supervised and unsupervised Machine Learning (ML) methods, such as classifiers, clustering and language models [1]. More recently, deep learning has emerged as a promising ML technique able to capture high level abstractions in the data, providing significant improvement for various tasks over more traditional ML methods, such as text classification [13], machine translation [2, 7] or sentiment analysis [27, 8]. However, to the best of our knowledge deep learning has not been applied yet to the problem of fine-grained information detection in crisis situations.

An advantage of the usage of deep learning is the capacity of the model to capture multiple layers of information. Our hypothesis is that, by encapsulating a layer of semantics into the deep learning model, we can provide a better characterisation of the contextual information, generally scarce in short, ill-formed social media messages; leading to a more accurate event identification.

We therefore propose in this paper a semantically enhanced Dual-CNN deep-learning model to target the problem of event detection in crisis situations. Our results show that our proposed model is able to successfully identify the existence of an event, and the event type (hurricane, floods, etc.) with $> 79\%$ F-measure, but the performance of the model significantly drops (61% F-measure) when identifying fine-grained event-related information, showing competitive results with more traditional ML techniques, such as SVM.

Our hypothesis is that the semantics extracted from tweets may not be sufficient to capture the level of contextual information needed for an accurate fine-grained event identification. Our future work therefore aims to enhance the semantic information extracted from tweets with additional methods to enrich the data abstraction captured by our proposed deep learning model.

The contributions of this paper can therefore be summarised as follows: 1) The generation of a deep learning model (Dual-CNN) to target the problem of event identification in crisis situations, and; 2) The exploration of how semantic information can be used to enrich the deep-learning data representations.

The rest of the paper is structured as follows. Section 2 shows related work on the areas of event detection and deep learning. Section 3 describes the scenario targeted in this paper and the different types of events that we aim to identify. Section 4 describes our proposed deep learning model for event identification. Sections 5 and 6 show our evaluation set up and the results of our experiments. Section 7 describes our reflections and our planned future work. Section 8 concludes the paper.

2 Related Work

Recently, several works have introduced the use of deep learning for event detection [6, 9, 17, 11, 31]. Unlike traditional ML feature-based methods, deep learning models do not generally require heavy feature engineering, and are therefore less prone to error propagation, caused by using external NLP and text processing tools. Also, deep learning models are more generic and tolerant to domain and context variations than feature-based models, as the former use word embeddings as a more general and richer representation of words. [17]

Pioneer works in this vein include [6, 9, 17]. These works address the problem of event detection at the sentence and/or phrase level by first identifying the event triggers in a given sentence (which could be a verb or nominalisation) and classifying them into specific types. For example, the word *"release"* in *"The European Unit will release 20 million euros to Iraq"* is a trigger for the event *"Transfer-Money"*. Multiple deep learning models have been proposed to address the above problem. For example, Nguyen and Grishman [17] use a Convolutional Neural Network (CNN) [15] with three input channels, corresponding to word embeddings, word position embeddings and entity type embeddings, to learn a word representation and use it to infer whether a word is an event trigger or not. Chen et al. [6] argue that a sentence may contain two or more events and that using a traditional CNN model with a max-pooling layer[1] often leads to capture clues of one event in the sentence but to miss the rest. To address this issue, the authors propose using a CNN with a dynamic multi-pooling layer to obtain a maximum value for each part of a sentence and therefore cover more valuable clues of the events within it.

Feng et al. [9] use a hybrid neural network model for cross-language event detection. The proposed model incorporates both, a bidirectional LSTM (Bi-LSTM) [24] and CNN component. Bi-LSTM captures contextual semantics of a given word by means of its preceding and following information in the text, while CNN is used to capture structure information from the local contexts (i.g., sentence chunks). Results show that the proposed model achieves relatively high and robust performance when applied to data of multiple languages including English, Chinese and Spanish, in comparison with traditional feature-based approaches.

It is worth noting that the above works experiment with their approach on the ACE 2005 event extraction corpus [30], which consists of a set of news articles collected from several online newspapers.

Our work in this paper differs from the above works in two main aspects: First, while the above works target the problem at the sentence level, our proposed model aims to detect events related to crisis situations at different detection levels (see Section 3). Secondly, in addition to using word embeddings, our model uses the conceptual semantics word embeddings (i.e., semantics extracted from external knowledge sources) as additional input layer to better capture the events' contextual and conceptual clues in the tweets as described in Section 4.

3 Scenario

During crises, a very large number, sometimes in the millions, of messages are often posted on various social media platforms by using the hashtags dedicated to the crises at hand. However, a good percentage of those messages are irrelevant or uninformative.

Olteanu and colleagues observe that crises reports could be classified into three main categories of informativeness; related and informative, related but not informative, and not related [19]. The percentage of relevant and informative social reports during crises varies a great deal, ranging from 10% in some cases[2] to 65% in others [25]. However, buried under very many mundane and irrelevant tweets, sometime one emerge that needs an urgent response.

---

[1] in a CNN, a max-pooling layer applies a max operation over the representation of an entire sentence to capture the most useful information.

[2] Behavioral & Linguistic Analysis of Disaster Tweets, `http://irevolution.net/2012/07/18/disaster-tweets-for-situational-awareness/`.

Our goal in this paper is to develop models to efficiently identify the messages of sufficient relevance and value. For this purpose, and based on the event types identified by [19] we consider the following three tasks when developing our approach:

- *Task1 - Crisis vs. non crisis related messages*: The goal of this task is to differentiate those posts that are *related to a crisis situation vs. those posts that do not*.
- *Task2 - Type of crisis*: The goal of this task is to identify the different types of crises the message is related to. Following the work of [19] we consider the following types of natural and human-induced types of crises: *shooting, explosion, building collapse, fires, floods, meteorite fall, haze, bombing, typhoon, crash, earthquake and derailment*.
- *Task3 - Type of information*: the goal of this task is to provide a fine-grained information detection in crises situations. Following the work of [19] we consider the following categories of crisis-related information: *affected individuals, infrastructures and utilities, donations and volunteer, caution and advice, sympathy and emotional support, useful information, other*.

4 A Semantic Deep Learning Approach for Event Detection

Event detection in the context of Twitter is a text classification task where the aim is to identify if a given document (post) describes or is related to an event. In this section we describe our proposed Dual-CNN model, a semantically enriched deep learning model for event detection on Twitter.

Besides relying on word embeddings, the proposed model also learns a semantic embeddings representation from word concepts that aims at better capturing the latent clues of the event description in tweets and consequently enhance the automatic detection of events.

The pipeline of our model consists of five main phases as depicted in Figure 1:

1. *Text Processing*: A collection of input tweets are cleaned and tokenised for later stages;
2. *Word Vector Initialisation*: Given a bag of words produced in the previous stage and a pre-trained word embeddings, a matrix of word embedding is constructed to be used for model training;
3. *Concept Extraction*: This phase run in parallel with the previous phase. Here the semantic concepts of named-entities in tweets are extracted using an external semantic extraction tool;
4. *Concepts Vector Initialisation*: this stage constructs a vector representation for each of the extracted entities as well as the entities' associated concepts;
5. *Dual-CNN Training*: in this phase our proposed Dual-CNN model is trained from both, the word embeddings matrix and the semantics embeddings matrix.

In the following subsections we describe each of the phases of the pipeline in more detail.

4.1 Text Preprocessing

Tweets are usually composed of incomplete, noisy and poorly structured sentences due to the frequent presence of abbreviations, irregular expressions, ill-formed words and

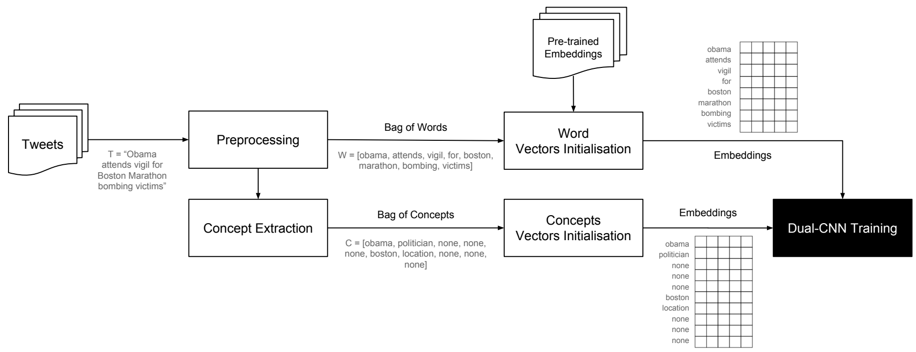

Fig. 1: Pipeline of the proposed semantic Dual-CNN deep learning model event detection model.

non-dictionary terms. This phase therefore applies a series of preprocessing steps to reduce the amount of noise in tweets including, for example, the removal URLs, and all non-ASCII and non English characters. After that, the processed tweets are tokenized into words that are consequently passed as input to the *word embeddings* phase.

4.2 Word Vector Initialisation

An important part for applying deep neural networks to text classification is to use word embeddings. As such, this phase aims to initialise a matrix of word embeddings for training the event classification model.

Word embeddings is a general name that refers to a vectorised representation of words, where words are mapped to vectors instead of a one dimension space [3]. The main idea is that semantically close words should have a similar vector representation instead of a distinct representation. Different methods have been proposed for generating embeddings such has Word2Vec [16] and GloVe [20] and they have shown to improve the performance in multiple NLP tasks. Hence, in this wok we choose to bootstrap our model with Google's pre-trained Word2Vec model [16] to construct our word embeddings matrix, where rows in the matrix represent embeddings vectors of the words in the Twitter dataset.

4.3 Concept Extraction and Semantics Vector Initialisation

As mentioned in the previous step, using word embeddings for training deep learning classification models has shown to substantially improve classification performance. However, conventional word embeddings methods merely rely on the context of a word in the text to learn its embeddings. As such, learning word embeddings from Twitter data might not be as sufficient for our training our classifier because tweets often lack context due to their short length and noisy nature.

To address this issue, we propose to enrich the training process of our proposed model with the semantic embeddings of words in order to better capture the context of tweets. To this end, we use AlchemyAPI[3] to first extract named entities from tweets (e.g. *'Oklahoma', 'Obama', 'Red Cross'*) and map them to their corresponding semantic sub-

[3] Alchemy API, `http://www.ibm.com/watson/alchemy-api.html`.

types (e.g. *'Location' , 'Politician', 'Non-Profit Organisation'*) using multiple semantic knowledge bases including DBpedia[4] and Freebase.[5]

After that, we represent each of the extracted entities and semantic types as a vector using an approach similar to the word embeddings. As a result, the semantic representation of documents (i.e. the entities and their associated semantic subtypes) become represented as a semantic embedding matrix, which is used for training the proposed Dual-CNN model.

4.4 Dual-representation CNN Model for Text Classification

This phase aims to train our Dual-CNN model from the word and semantic embeddings matrices. Below we describe our CNN-Model along with the proposed training procedure.

As discussed in section 2, CNN can be used for classifying sentences or documents [13]. The main idea is to use word embeddings coupled with multiple convolutions of varying sizes that extract important information from a set of words in a given sentence, or a document, and then apply a softmax function that predict its class.

Kim's model [13] is a simple CNN model widely used for text classification. It consists of a convolution layer (with three region sizes and multiple filters per region) followed by a max-pooling phase and a fully connected layer where the softmax function is applied for predicting the document classes.

In this paper, we propose to extend the aforementioned CNN model with an additional semantic representation layer representing the named entities in tweets and their associated semantic subtypes. Although, in principle, the most logical method for adding a semantic representation to an existing word-embedding CNN model is to use an additional channel, as it is commonly used in image classification, it requires one-to-one mappings between each embedding channel; meaning that the words and semantic tokenisations of a document need to match exactly (i.e. for a given document, the word and semantic embeddings need to have the same length and width).

Nonetheless, one-to-one mappings between word tokens and their meanings can not be enforced. For example, a document D = '***Obama attends vigil for Boston Marathon bombing victims.***' may be tokenised as $T_w = [\text{`}obama\text{'}, \text{`}attends\text{'}, \text{`}vigil\text{'}, \text{`}for\text{'}, \text{`}boston\text{'}, \text{`}marathon\text{'}, \text{`}bombing\text{'}, \text{`}victims\text{'}]$ by a word tokeniser whereas a semantic tokeniser may split D as $T_s = [\text{`}obama\text{'}, \text{`}politician\text{'}, \text{`}none\text{'}, \text{`}none\text{'}, \text{`}none\text{'}, \text{`}boston\text{'}, \text{`}location\text{'}, \text{`}none\text{'}, \text{`}none\text{'}, \text{`}none\text{'}]$ using entity and entity-type tokens. In this context, the embedding of both T_w and T_s cannot be used directly as different channels of the embedding representation of D, as they have different length.

In order to deal with this particular issue, we decided to add a parallel convolutional layer that is computed separately from the word embeddings. This is done before a merging step, that concatenates the max-pooling steps for each representation layer, and before applying the softmax step that classifies individual documents as depicted in Figure 2

[4] DBpedia, `http://dbpedia.org`.
[5] Freebase, `http://www.freebase.com`.

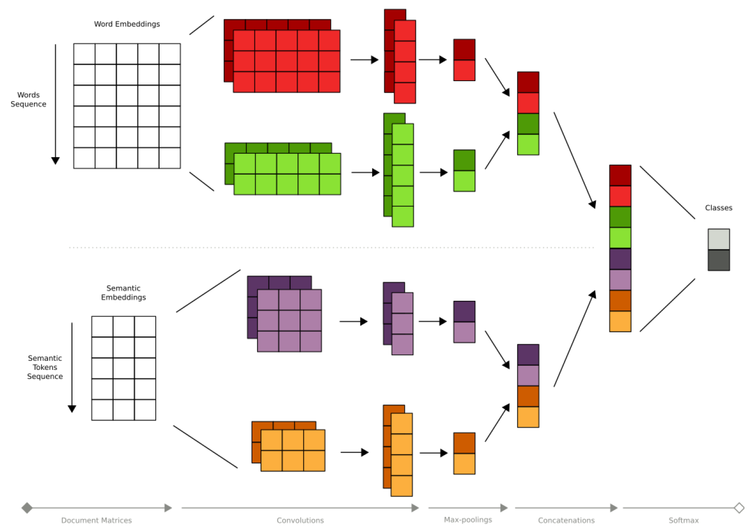

Fig. 2: Dual-representation Convolutional Neural Network (CNN) for text classification with word embeddings and semantic embeddings representations

5 Experimental Setup

Here we present the experimental setup used to assess our event detection model. As mentioned in Section 3, we aim to apply and test the proposed model in three different tasks. As such, our evaluation setup requires the selection of (i) Twitter datasets, (ii) the semantic extraction tool, and (iii) baseline models for cross-comparison.

5.1 Dataset

To assess the performance of the event detection model we require the use of datasets where each tweet is annotated with: whether or not it relates to a crisis event, the type of crisis (earthquake, flood, etc.) and the type of information (affected individuals, infrastructures, etc.) - see Section 3 for more details. For the purpose of this work we use the CrisisLexT26 dataset.[18]

CrisisLexT26 includes tweets collected during 26 crisis events in 2012 and 2013. Each crisis contains around 1,000 annotated tweets for a total of around 28,000 tweets with labels that indicate if a tweet is related or unrelated to a crisis event (i.e. *related/unrelated*, Task1)

For the second task (see Section 3), we need a list of crisis types. In order to obtain such information, we consider that the annotated tweets that are from the same sub-collection belong to the same type of event. Using this approach we obtain 12 different crisis types (*shooting, explosion, building collapse, fires, floods, meteorite fall, haze, bombing, typhoon, crash, earthquake and derailment*) (Task 1).

The CrisisLextT26 tweets are also annotated with additional labels indicating the type of information present in the tweet (*affected individuals, infrastructures and utilities, donations and volunteer, caution and advice, sympathy and emotional support, and* *useful information and unknown*, Task 3). More information about the CrisisLexT26 dataset can be found on the CrisisLex website.[6]

Since the annotations tend to be unbalanced, we also create a balanced version of the dataset for each task by performing biased random undersampling using tweets from each sub-collection. As a result, the first task dataset is reduced to 6703 tweets (24%), the second task to 12997 tweets (46.5%) and the final task to 9105 tweets (32.6%).

5.2 Semantic Extraction

As mentioned in Section 4, the Dual-CNN model integrates the conceptual semantics of words as semantic embeddings to better capture event clues in tweets. We take conceptual semantics to refer to the semantic types (e.g. '*Location'* , *'Politician', 'Non-Profit Organisation'*) of named-entities (e.g. '*Oklahoma'* , *'Obama', 'Red Cross'*) in tweets. To extract this type of semantic from our Twitter datasets we use the AlchemyAPI semantic extraction tool due to its accuracy and high coverage of semantic types in comparison with other semantic extraction services [22, 23]. Nevertheless, only 16.6% of the dataset tweets get annotated by the semantic extraction tool.

6 Evaluation

In this section, we report the results obtained from using the proposed Dual-CNN model for crisis event detection of tweets under three evaluation tasks: (Task1) Crisis vs. non crisis related tweets, (Task2) type of crisis, and (Task3) type of information. Our baselines of comparison are three traditional machine learning classifiers: Naive Bayes, Classification and Regression Trees (CART), and SVM with RBF kernels trained from words unigrams. We initialise our CNN models with the Google News 3 million words and phrases pre-trained word embeddings data.[7] Results for all experiments are computed using 5-fold cross validation. For each task, we perform the evaluation on the full and undersampled versions of the dataset.

We train the CNN model using 300 long word embeddings vectors with $F_n = 128$ convolutional filter of sizes $F_s = [3, 4, 5]$. For the Dual-CNN model, we use the same parameters except that for the semantic embeddings, we use 30 long vectors since we have very few semantic concepts compared to the size of words lexicon. For avoiding over-fitting, we use a dropout of 0.5 during training and use the ADAM gradient decent algorithm [14]. We perform 400 iterations with a batch size of 256.

Table 1 shows the results of our event detection classifiers for the three evaluation tasks on the full and undersampled versions of the dataset. In particular, the table reports the precision (P), recall (R), and F1-measure (F1) for each evaluation Task and model. The table also reports the types of features and embeddings used to train the different classifiers.

6.1 Baselines Results

As seen in Table 1, the results for each task and each baseline show that the first two tasks are relatively easy to predict whereas predicting information types is much more complex. In general we also observe that SVM is the best performing algorithm followed

[6] CrisisLex T26 Dataset, `http://www.crisislex.org/data-collections.html\#CrisisLexT26`.

[7] Google Word2Vec, https://code.google.com/archive/p/word2vec

Table 1: Event detection performance of baselines and our proposed CNN models under the three evaluation Tasks on full and undersampled datasets. PT-Embed: Pre-trained word embeddings. PTS-Embeddings: pre-trained word embeddings and semantic word embeddings.

			Related/Unrelated			Event Types			Information Types		
Model	Data	Features	P	R	$F1$	P	R	$F1$	P	R	$F1$
NAIVE BAYES	*Full*	TF-IDF	0.846	0.684	0.733	0.941	0.927	0.933	0.600	0.570	0.579
CART	*Full*	TF-IDF	0.742	0.707	0.723	0.992	0.992	0.992	0.506	0.491	0.497
SVM	*Full*	TF-IDF	0.870	0.738	0.785	0.997	0.996	0.997	0.642	0.604	0.616
CNN	*Full*	PT-Embed	0.861	0.744	0.797	0.991	0.986	0.988	0.634	0.590	0.609
DUAL-CNN	*Full*	PTS-Embed	0.857	0.762	0.798	0.990	0.985	0.988	0.648	0.581	0.601
NAIVE BAYES	*Sample*	TF-IDF	0.795	0.787	0.785	0.929	0.928	0.928	0.558	0.563	0.556
CART	*Sample*	TF-IDF	0.770	0.769	0.769	0.988	0.988	0.988	0.471	0.464	0.464
SVM	*Sample*	TF-IDF	0.833	0.830	0.829	0.995	0.995	0.995	0.606	0.609	0.605
CNN	*Sample*	PT-Embed	0.839	0.838	0.838	0.983	0.983	0.983	0.610	0.610	0.610
DUAL-CNN	*Sample*	PTS-Embed	0.835	0.833	0.833	0.985	0.985	0.985	0.615	0.615	0.613

by CART and Naive Bayes. For the first two tasks with the full data, each method achieve *precision*, *recall* and $F_1 > 0.72$ and SVM appears to be the best model with $F_1 = 0.785$ for identifying crisis related tweets and $F_1 = 0.997$ for identifying event types. The task of identifying information types show much lower F_1 accross the board. This is probably due to the fact that compared to the previous tasks, information types probably contain much more general terms in each class. Similarly to the prevous tasks, SVM performs the best with $F_1 = 0.616$.

With the balanced datasets, the results are similar. However, the predictions for the first task increase by around $+4.8\%$. This results is likely due to the fact that the first task was the most imbalanced task and benefits the most from the undersampling process.

The high precision and recall results observed for the second task ($F_1 = 0.997$) suggests that the different models overfit the data. The issue was not resolved by undersampling the data with an F_1 of 0.995. Looking at the data in more details, we observe that each category contains very clear category indicators. For instance, 77% of the tweets about *meteorite falls* contain the word *meteor*, whereas 76.2% of the tweets about *explosions* contains the word *Boston*. In order to reduce such issue, we could for instance remove some of these words from the dataset so the models become less tied to practical event instances (e.g the Boston bombings).

6.2 CNN and Dual-CNN Results

In general, applying CNNs with pre-trained word embeddings (*PT-Embed*) for both the full and undersampled data does not improve significantly over SVM. Using the full dataset, we obtain an F_1 of 0.797 for the crises related tweets and full dataset, 0.988 for event types detection and 0.616 for information type identification. We also observer very little difference between the CNN model and Dual-CNN model despite adding an additional semantic layer.

When using the undersampled datasets, the results are similar to the previous observations with an increase of $+3.7\%$ in F_1 for the first task. There is also a slight improvement for the last task with $+0.6\%$ in F_1.

Adding semantics seems to not improve much the accuracy compared to the standard CNN model. This result may be explained by different factors. First, the size of our semantic concepts and entities vocabulary is much smaller than the word lexicon with only 265 semantic terms compared to 57,577 words. Second more than 83.5% of the

Tweets appear to not have any concepts. This means that very little semantic context is available for each Tweet and that the extracted semantic information has little impact on the predictive power of our model. Such issue could be alleviated by using better semantic extraction techniques or using a more complex semantic representation of Tweets. We could also increase the number of iterations and the size of the batches to improving the performance of the model.

7 Discussion and Future Work

In this paper we introduced the use of conceptual semantics embeddings in deep learning CNN models for detecting events on Twitter. This section discusses the limitations of the presented work as well as different areas of future investigations.

We experimented with our proposed Dual-CNN model on three event detection tasks (Section 3) and observed that identifying crisis related events and event types in tweets (i.e. Task 1 and Task 2) with high accuracy appears to be a relatively easy task that can be fulfilled well with both traditional models such as SVM and CNN models. Identifying the types of information provided in crisis related tweets (Task 3) is much more challenging as tweets mentioning event information types tend to contain much more general terms in each classes than the tweets that are related or unrelated to crises or are discussing different types of events.

Looking into the details of the second task, we observed that for this task, the models were generally overfitted even after balancing the data. The reason seems to be associated with the presence of very clear category indicators (e.g., place names). In order to reduce such an issue, we could remove place names from training instances or try to collect additional data so that the associations between event types and locations is reduced.

Despite using the semantic concepts of words in the proposed Dual-CNN model, we found no significant improvement compared to the original CNN model. As stated in the previous section, the lack of improvement is probably linked to the small size of the semantic vocabulary as well as the ability of the Alchemy API semantic extraction tool to extract concepts from tweets (only 16.5% of the tweets had semantic concepts extracted by the Alchemy API). Also, we observed that some of the extracted concepts were too abstract (e.g., Location) and were mapped to entities in both, event-related and event-unrelated tweets. This might affect the discrimination power of such concepts and lead to inaccurate event classification.

As future work, we plan to investigate methods to improve both, the extraction and the integration of words' conceptual semantics into our proposed model. For the semantic extraction part, we plan to increase the number as well as the specificity of the conceptual semantics, perhaps with the aid of Linked Data or using alternative extractors, such as TextRazor.[8]

Concerning the event detection model, we plan to improve the Dual-CNN model by adding additional convolutional layers and performing parameter optimisation. For instance we could try to improve the results by modifying the size of the model filters as well as the number of filters. We could also increase and optimise the number of training steps in order to obtain better results.

[8] TextRazor, `http://www.textrazor.com`.

Our proposed dual layer model is built on top of a CNN network, which assumes that all inputs (i.e., words and semantic concepts) are loosely coupled with each other. However, it might be the case that the latent clues of an event can be determined based on the intrinsic dependencies between the words and semantic concepts of a tweet. Hence, room for future work is to incorporate these information in our event detection model, probably by using recurrent neural networks (RNN) [12] due to their ability to capture sequential information in text.

8 Conclusions

We proposed Dual-CNN, a deep learning model that uses the conceptual semantics of words for fine-grained event detection in crisis situations. We based our analysis on Twitter data since it is a social media platform that is widely used during crisis events. We investigated how named-entities in tweets can be extracted and used, together with their corresponding semantic concepts as an additional CNN layer to train a deep learning model for event detection on Twitter. We used our Dual-CNN model on a Twitter dataset of 26 different crisis events and tested its performance under three event detection tasks. Results show that our model is able to successfully identify the existence of events, and event types with $> 79\%$ F-measure, but the performance of the model significantly drops (61% F-measure) when identifying fine-grained event-related information. These results are competitive with more traditional Machine Learning models, such as SVM.

Acknowledgment: This work has received support from the European Union's Horizon 2020 research and innovation programme under grant agreement No 687847 (COMRADES).

References

1. Atefeh, F., Khreich, W.: A survey of techniques for event detection in Twitter. Computational Intelligence 31(1), 132–164 (2015)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
3. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of machine learning research 3(Feb), 1137–1155 (2003)
4. Bercovici, J.: Why time magazine used instagram to cover hurricane sandy. Obtenido de http://www. forbes. com/sites/jeffbercovici/2012/11/01/why-time-magazine-used-instagram-to-cover-hurricane-sandy (2012)
5. Campanella, T.J.: Urban resilience and the recovery of New Orleans. Journal of the American Planning Association 72(2), 141–146 (2006)
6. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multi-pooling convolutional neural networks. In: ACL (1). pp. 167–176 (2015)
7. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014) (2014)
8. Dos Santos, C.N., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: COLING. pp. 69–78 (2014)
9. Feng, X., Huang, L., Tang, D., Qin, B., Ji, H., Liu, T.: A language-independent neural network for event detection. In: The 54th Annual Meeting of the Association for Computational Linguistics. p. 66 (2016)
10. Gao, H., Barbier, G., Goolsby, R.: Harnessing the crowdsourcing power of social media for disaster relief. IEEE Intelligent Systems 26(3), 10–14 (2011)

11. Ghaeini, R., Fern, X.Z., Huang, L., Tadepalli, P.: Event nugget detection with forward-backward recurrent neural networks. In: The 54th Annual Meeting of the Association for Computational Linguistics. p. 369 (2016)
12. Graves, A.: Supervised sequence labelling. In: Supervised Sequence Labelling with Recurrent Neural Networks, pp. 5–13. Springer (2012)
13. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014) (2014)
14. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2014)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
17. Nguyen, T.H., Grishman, R.: Event detection and domain adaptation with convolutional neural networks. In: ACL (2). pp. 365–371 (2015)
18. Olteanu, A., Castillo, C., Diaz, F., Vieweg, S.: CrisisLex: A lexicon for collecting and filtering microblogged communications in crises. In: ICWSM (2014)
19. Olteanu, A., Vieweg, S., Castillo, C.: What to expect when the unexpected happens: Social media communications across crises. In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing. pp. 994–1009. ACM (2015)
20. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014)
21. Qu, Y., Huang, C., Zhang, P., Zhang, J.: Microblogging after a major disaster in China: a case study of the 2010 Yushu earthquake. In: Proceedings of the ACM 2011 conference on Computer supported cooperative work. pp. 25–34. ACM (2011)
22. Rizzo, G., Troncy, R.: NERD: Evaluating named entity recognition tools in the web of data. In: Workshop on Web Scale Knowledge Extraction (WEKEX11). vol. 21 (2011)
23. Saif, H., He, Y., Alani, H.: Semantic sentiment analysis of twitter. In: Proc. 11th Int. Semantic Web Conf. (ISWC). Boston, MA (2012)
24. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45(11), 2673–2681 (1997)
25. Sinnappan, S., Farrell, C., Stewart, E.: Priceless tweets! a study on Twitter messages posted during crisis: Black Saturday. ACIS 2010 Proceedings 39 (2010)
26. Starbird, K., Palen, L., Hughes, A.L., Vieweg, S.: Chatter on the red: what hazards threat reveals about the social life of microblogged information. In: Proceedings of the 2010 ACM conference on Computer supported cooperative work. pp. 241–250. ACM (2010)
27. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015). pp. 1422–1432 (2015)
28. Thomson, R., Ito, N., Suda, H., Lin, F., Liu, Y., Hayasaka, R., Isochi, R., Wang, Z.: Trusting tweets: The Fukushima disaster and information source credibility on twitter. In: Proceedings of the 9th International ISCRAM Conference. pp. 1–10 (2012)
29. Vieweg, S., Hughes, A.L., Starbird, K., Palen, L.: Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: Proceedings of the SIGCHI conference on human factors in computing systems. pp. 1079–1088. ACM (2010)
30. Walker, C., Strassel, S., Medero, J., Maeda, K.: ACE 2005 multilingual training corpus. Linguistic Data Consortium, Philadelphia 57 (2006)
31. Zeng, Y., Yang, H., Feng, Y., Wang, Z., Zhao, D.: A convolution BiLSTM neural network model for chinese event extraction. In: International Conference on Computer Processing of Oriental Languages. pp. 275–287. Springer (2016)