Semantic Wide and Deep Learning for Detecting Crisis-Information Categories on Social Media

Conference or Workshop Item

Semantic Wide and Deep Learning for Detecting Crisis-Information Categories on Social Media

Grégoire Burel, Hassan Saif, and Harith Alani

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

Abstract. When crises hit, many flog to social media to share or consume information related to the event. Social media posts during crises tend to provide valuable reports on affected people, donation offers, help requests, advice provision, etc. Automatically identifying the category of information (e.g., *reports on affected individuals*, *donations and volunteers*) contained in these posts is vital for their efficient handling and consumption by effected communities and concerned organisations. In this paper, we introduce Sem-CNN; a wide and deep Convolutional Neural Network (CNN) model designed for identifying the category of information contained in crisis-related social media content. Unlike previous models, which mainly rely on the lexical representations of words in the text, the proposed model integrates an additional layer of semantics that represents the named entities in the text, into a *wide* and *deep* CNN network. Results show that the Sem-CNN model consistently outperforms the baselines which consist of statistical and non-semantic deep learning models.

Keywords: Semantic Deep Learning, Crisis Information Processing, Social Media.

1 Introduction

Social media has become a common place for communities and organisations to communicate and share various information during crises, to enhance their situational awareness, to share requests or offers for help and support, and to coordinate their recovery efforts.

The volume and velocity of this content tend to be extremely high, rendering it almost impossible for organisations and communities to manually analyse and process the content shared during such crises [12, 16]. For example, in a single day during the 2011 Japan earthquake, 177 million tweets related to the crisis were sent [5]. In 2013, more than 23 million tweets were posted about the haze in Singapore [22].

Olteanu and colleagues study samples of tweets posts during various crisis situations, and found that crisis-related social tweets tend to bare one of the following general information categories [20]: *affected individuals, infrastructures and utilities, donations and volunteer, caution and advice, sympathy and emotional support, other useful information.* However, tools to automatically identify the category of information shared during crises are still largely unavailable.

Recent research is mostly focused on processing social media content to determine what documents are related to a crisis and what documents are not (e.g., [20]), or to detect the emergence of major crisis event (e.g., *floods* [26], *wildfires* [29], *earthquakes*

[23], *nuclear disasters* [28], etc.). However, the automatic identification of the category or type of information shared about events is still in its infancy [20].

For example, although both of the tweets '*Colorado fire displaces hundreds; 1 person missing.*' and '*If you are evacuating please dont wait, take your pets when you evacuate*' were posted during the 2012 Colorado's wildfire crisis,[1] they bare different information, i.e., while the former tweet reports information on individuals affected by the fire, the latter offers advices to the public. The approach presented in this paper is aimed at classifying such kind of documents to automatically determine which ones provide which category of information. Such a mechanism can help users (e.g., citizens, humanitarian organisation, government officials, police forces) to quickly filter big volumes of crisis-related tweets to only those that provide the types of information they are interested in.

Most current research on identifying crisis information from social media rely on the 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 new ML technique able to capture high level abstractions in data, thus providing significant improvement over traditional ML methods in certain tasks, such as in text classification [13], machine translation [2, 8] and sentiment analysis [27, 10].

Applying deep learning to enhance the analysis of crisis-related social media content is yet to be thoroughly explored [4]. In this paper, we hypothesise that the encapsulation of a layer of semantics into a deep learning model can provide a more accurate crisis-information-category identification by better characterising the contextual information, which is generally scarce in short, ill-formed social media messages.

We therefore propose Sem-CNN; a semantically enhanced *wide* and *deep* Convolutional Neural Network (CNN) model, to target the problem above. We also investigate the integration of semantic information in two different methods; (a) using semantic concept labels, and (b) using semantic concept abstracts from DBpedia.[2] Our main contributions in this paper are:

- Generation of a wide and deep learning model (Sem-CNN) to identify the category of crisis-related information contained in social media posts.
- Demonstration of two methods for enriching deep learning data representations with semantic information.
- Evaluation of the approach on three samples of the CrisisLexT26 dataset, which consists around 28,000 labelled tweets.
- Produce an accuracy that outperforms the best baselines by up to $+22.6\%$ F-measure (min $+0.5\%$), thus proving the potential of semantic deep learning approaches for processing crisis-related social media content.

The rest of the paper is structured as follows. Section 2 shows related work in the areas of event detection and deep learning. Section 3 describes our proposed deep learning model for event identification. Sections 4 and 5 show our evaluation set up and the results of our experiments. Section 6 describes our reflections and our planned future work. Section 7 concludes the paper.

---

[1] High Park fire Wikipedia article, `https://en.wikipedia.org/wiki/High_Park_fire`.
[2] DBpedia, `http://dbpedia.org`.

2 Related Work

Crisis-related data analysis is often divided into three main tasks [20]. First, *crisis-related* posts are separated from *non-related* documents. This allows the filtering of documents that may have used a *crisis-related* term or hashtag, but does not contain information that is relevant to a particular crisis event. Second, the type of events mentioned (e.g., *fires, floods, bombing*) are identified from each remaining post in order to identify the main type of event discussed in a document. Third, the category of information contained in these crisis-related tweets are determined. Olteanu and colleagues observed that there is a small number of information categories that most crisis-related tweets tend to bare [20]. These categories are shown in Table 1 along with examples of tweets related to the Colorado's Wildfires. Crisis-information-category can be used by responders to better asses an event situation as they tend to be more actionable than the more general event categories.

In our previous work [4], we showed that the first two tasks can be performed relatively successfully with traditional classification techniques (e.g., SVM), achieving higher than 80% in precision and recall values. However, the automatic identification of crisis information categories proved to be a more challenging task.

Table 1: Crisis information categories from [20], and tweet examples

Category	Example
Affected Individuals	'*Colorado fire displaces hundreds; 1 person missing: Firefighters in Colorado and New Mexico are battling wind-fu*'
Caution and Advice	'*If you are evacuating please dont wait, take your pets when you evacuate #HighParkFire*'
Donations and Volunteering	'*RT @username: we are available to house a displaced kitty or two if needed #flagstafffire cc @username*'
Infrastructure and Utilities	'*Homes at risk from Colorado wildfire: Hundreds of families took refuge early Monday at a northern Colorado*'
Sympathy and Support	'*Pray for rain! RT @username: #HighParkFire is now at 36,930 acres.*'
Other Useful Information	'*Photo of the Colorado wildfire from space (via @NASA) URL #HighParkFire*'

Identifying *information categories* from social media is a commonly used step in event detection literature, and several recent works used deep learning for event detection in different contexts. The advantage brought by deep learning models over traditional ML feature-based methods is the lightweight feature engineering they require and their reliance instead on word embeddings as a more general and richer representation of words [18].

Pioneer works in this field include [6, 11, 18], which 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 categories. Multiple deep learning models have been proposed to address this

problem. For example, Nguyen and Grishman [18] use CNNs [15] with three input channels, corresponding to *word embeddings*, *word position embeddings*, and *ACE entity type embeddings*[3], to learn a word representation and use it to infer whether a word is an event trigger or not. Contrary to the general DBpedia entities and concepts that we use in our research, ACE entities are limited to only a few concepts and cannot be associated to concept or entity descriptions or abstracts.

We investigated the use of semantics for crises-event detection with deep learning methods in [4], where we added a CNN layer to a traditional CNN model by combining two parallel layers that join word embeddings and semantic embeddings initialised from extracted concepts. Although the model performed well for identifying crisis-related tweets and the general crisis events they mention, its performance in identifying information categories could not outperform the more traditional classification methods such as SVM. This was perhaps due to the training complexity of CNN and the semantic embeddings as the amount of semantics in each document is limited.

The approach introduced in this paper differs from [4] by using a variation of the *wide* and *deep* learning model [7] that is designed for balancing the *richness* of semantic information with the *shallowness* of textual content of documents. In particular, it reuses the strength of CNN models for dealing with textual content and a more traditional linear model for dealing with the richness of semantic information. Contrary to the approach in [4], our new model also considers entity and concept abstracts in its semantic input for allowing a better representation of the document semantics.

3   The Sem-CNN Approach for Identifying Crisis Information Categories

In the context of Twitter,[4] the identification of the category of information contained in crises-related tweets is a text classification task where the aim is to identify which posts contain which category of crisis-related information. In this section we describe our proposed Sem-CNN model, which is a semantically enriched deep learning model for identifying crisis-related information categories on Twitter.

The proposed approach is a *wide* and *deep* learning model [7] that jointly integrates shallow textual information in a *deep* Convolutional Neural Network (CNN) model with semantic annotations in a *wide* generalised linear model.

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] Automatic   Content   Extraction   (ACE)   Entities,   `http://ldc.upenn.edu/ collaborations/past-projects/ace`.

[4] Twitter, `http://twitter.com`.

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 (e.g, TextRazor[5], Alchemy API[6], DBpedia Spotlight [9]);

4. *Semantic Vector Initialisation*: This stage constructs a vector representation for each of the entities and concepts extracted in the previous phase. The vector is either constructed from DBpedia concept labels or from DBpedia concept abstracts;

5. *Sem-CNN Training*: In this phase the proposed Sem-CNN model is trained from both, the word embeddings matrix and the semantic term-document vector (concept names or concept abstracts).
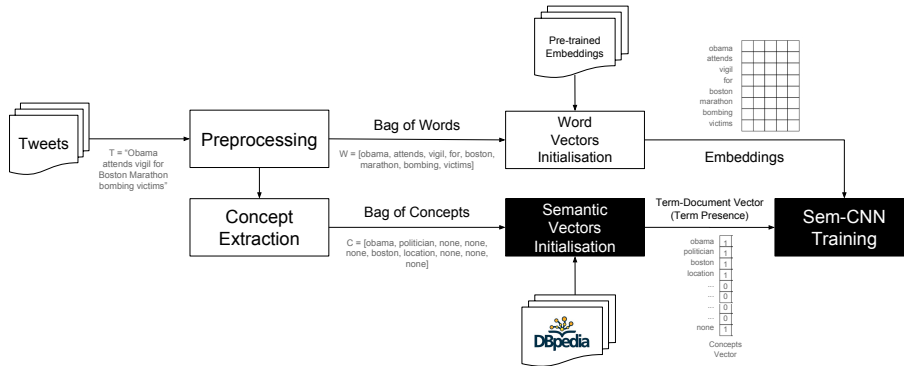

Fig. 1: Pipeline of the proposed semantic Sem-CNN deep learning model for detecting crises information categories .

In the following subsections we detail each phase in the pipeline.

3.1 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 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 of URLs, and all non-ASCII and non English characters. After that, the processed tweets are tokenised into words that are consequently passed as input to the *word embeddings* phase. Although different methods can be used for preprocessing textual data, we follow the same approach used by Kim in the CNN sentence classification model [13].

3.2 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 information 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

---

[5] TextRazor, https://www.textrazor.com/.

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

instead of a distinct representation. Different methods have been proposed for generating embeddings such has Word2Vec [17] and GloVe [21] and they have shown to improve the performance in multiple NLP tasks. Hence, in this work we choose to initialise our model with Google's pre-trained Word2Vec model [17] to construct our word embeddings matrix, where rows in the matrix represent embedding vectors of the words in the Twitter dataset.

3.3 Concept Extraction and Semantic 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 embedding 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 training our classifier because tweets often lack context due to their short length and noisy nature.

One possible approach to address this issue is to enrich the training process of our proposed model with the semantic embeddings of words in order to better capture the context of tweets. This approach we pursued in [4] was to add semantic embeddings (i.e., a vectorised representation of semantic concepts) to a two layer CNN model [4]. However, since tweets are small documents the number of unique concepts available within a corpus of documents is much lower than the number of words present in the corpus. As a consequence, the number of available concepts may not allow the efficient training of the semantic embeddings.

In this context, rather than using semantic embeddings, we propose to use the more traditional vector space model representation of documents where the semantics of each document is represented as a vector that identifies the presence of individual semantic concepts as vector indexes within a concept space. We also represent the presence of individual semantic concepts (or associated abstract words) rather than the frequency of concepts within a tweet since tweets are short textual documents.

Before converting the tweets' semantics into the vector space model representation, we first extract the named-entities in tweets (e.g., *'Oklahoma', 'Obama', 'Red Cross'*) and map them to their associated semantic concepts (aka semantic types) (e.g., *'Location', 'Politician', 'Non-Profit Organisation'*) using multiple semantic knowledge bases including DBpedia and Freebase.[7]

We decided to use the TextRazor tool due to its higher accuracy, coverage, and performance in comparison with other entity extraction and semantic linking tools [24].

We use the extracted entities along with their concepts to enrich the training process in the Sem-CNN model. We investigate two different methods for integrating the semantics into the vector space model: 1) the usage of the semantic concepts and entities labels, and; 2) the usage of the DBpedia descriptions of semantic concepts and entities (i.e., concept abstracts). In the following subsections we describe these methods in more detail.

3.3.1 Semantic Concepts Vector Initialisation

The first method for converting the concepts and entities extracted from tweets using the TextRazor tool is to use, when available, their semantic labels (`rdfs:label`) from

---

[7] Freebase, `http://www.freebase.com`

DBPedia. When such labels are unavailable, the labels that are returned from TextRazor are used directly instead.

The method used for converting a given document using the semantic concepts vector initialisation method is displayed in Figure 1. For an example document $D =$ '*Obama attends vigil for Boston Marathon bombing victims.*', the concepts and entities labels are extracted and tokenised using a semantic extraction tool and DBpedia so that the words that do not have extracted semantics are converted to a *none* label. Using this method the document D may be tokenised as $T_s = ['obama', 'politician', 'none', 'none', 'none', 'boston', 'location', 'none', 'none', 'none']$ using entity and entity-type tokens. The tokenised version is then converted into the vector space model that the depends on the concept space size $N_s = ['obama', 'politician', 'boston', 'location', \cdots, 'none']$ of size n_s, where n_s represents the total number of concepts and entities in the corpus of documents where D is extracted from. Using the previous concept space, T_s can be converted to the following vector space model $V_s = [1, 1, 1, 1, 0, 0, 0, \cdots, 1]$.

3.3.2 Semantic Abstracts Vector Initialisation

The second method uses, when available, the first sentence of the DBpedia abstracts (`dbo:abstract`) rather than the semantic labels (`rdfs:label`). This has the potential advantage of providing richer contextual representation of the semantics contained in the tweets as DBpedia abstracts normally contain additional implicit semantics that are not available in the `rdfs:label`. In particular, since DBpedia abstracts are extracted from Wikipedia articles,[8] the first sentence of each abstract tends to contain highly descriptive terms that are effectively semantic concepts even though they are not explicitly represented as such (i.e., DBpedia concepts). For example, for the semantic concept `dbpedia:Barack_Obama`, the first sentence of the `dbo:abstract` property is '*Barack Hussein Obama II; born August 4, 1961) is an American politician serving as the 44th President of the United States, the first African American to hold the office.*'. This sentence contains multiple implicit entities and concepts such as `dbo:President_of_the_United_States`, `dbo:Politician`, `dbpedia:United_States`. As a consequence, by using the `dbo:abstract` of the concepts and entities found in the documents, we effectively increase the concept space size of the concept vectors and increase the contextual semantics of the document.

The method used for converting the extracted semantics to the vector space model is the same as the one used when doing the semantic concepts vector initialisation except that the concept and entity labels are replaced with the first sentence of the DBpedia abstracts. Effectively, we obtain longer vectors for each documents since the semantic vocabulary space n_a, is larger than the label-only semantic space n_s ($n_a \gg n_s$).

In principle, it is possible to use both the content of abstracts and semantic concepts labels together. However, it does not necessarily increase the amount of semantics found each semantic vector since each `dbo:abstract` already contains the labels of the extracted concept and entities found in tweets. As a consequence, we focus our research on the semantic concepts vectorisation and the semantic abstracts vectorisation approaches individually.

---

[8] Wikipedia, `http://en.wikipedia.org`

3.4 A Wide and Deep Semantic CNN Model for Text Classification

This phase aims to train our Sem-CNN model (Figure 2) from the word embeddings matrices and semantic vectors described in the previous section. Below we describe the *wide* and *deep* CNN model that we propose to tackle the task of identifying fine-grained information within crisis-related documents.
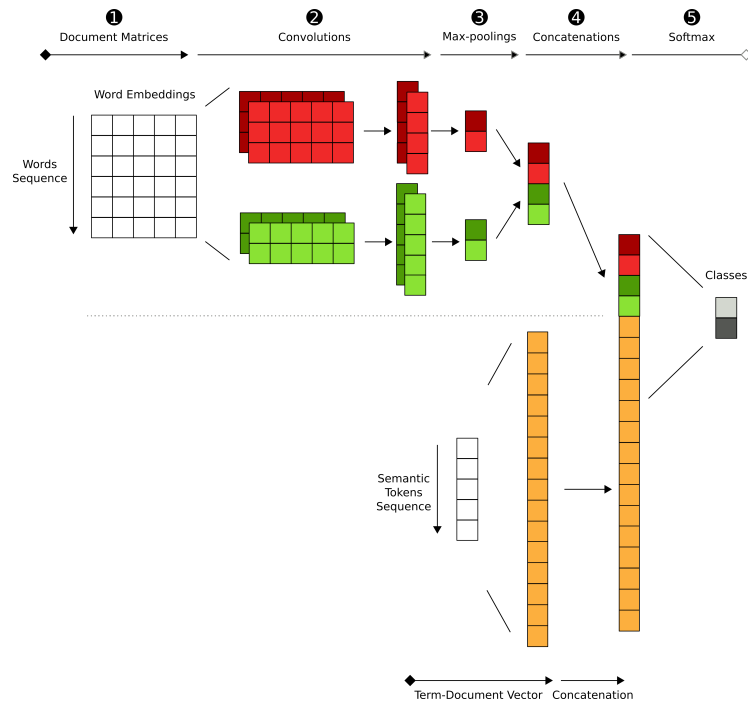

Fig. 2: *Wide* and *Deep* Convolutional Neural Network (CNN) for text classification with word embeddings and semantic document representations: 1) A word embedding matrix is created for a document: 2) Multiple convolutional filters of varying sizes generate features vectors; 3) Max pooling is performed on each features vector; 3) The resulting vectors are concatenated with the semantic vector representation of the document, and; 5) A softmax layer is used for classifying the document.

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 predicts its class.

CNN models can be also extended with semantic embeddings in order to use contextual semantics when classifying textual documents [4]. However, there are some drawbacks in simply adding an additional parallel layer of convolutions that integrates these extracted semantic embeddings (Section 2).

First, the limited number of available semantics across tweets is low, which limits the usefulness of embeddings since little data is available for training them. Second, the

CNN networks takes into account the location of the entities within the tweets. Although this might be beneficial in principle, the number of non-annotated terms in tweets makes this less useful, and make the model more complicated to train.

A potential solution to those problems is to create a deep learning model that takes into account the *richness* and depth of the semantics contained in the entities and concepts extracted from documents (tweets) and the *shallowness* of the textual content.

The *wide* and *deep* learning model [7] is a deep learning model that jointly trains a wide linear model and a deep neural network. This approach can be potentially useful for our particular task where we need to combine *shallow* textual information with the *richer* semantic information. In particular, we can use the deep neural network on the textual part of documents whereas the wide part is trained on the entities extracted from documents. This means that we effectively balance the *shallowness* of textual content with the *richer* information of semantic concepts and entities.

Although in general the Sem-CNN model (Figure 2) is philosophically similar to the *wide* and *deep* learning model, the proposed model has three major differences:

1. Rather than using a set of fully connected layers for the *deep* part of the model, we use a set of convolutions since this is known to perform well for text classification tasks [13, 4].
2. In the standard *wide* and *deep* learning model, the wide and deep layers use the same input features encoded in different formats (i.e., feature embeddings and feature vectors) whereas our model uses two different feature sets for each part of the model (i.e., word embeddings and concept/entity feature vectors).
3. The standard *wide* and *deep* learning model uses cross product transformations for the feature vectors in the *wide* part of the model. In the Sem-CNN model we omit this transformation due to the small size of the semantic vocabulary and the number of semantics extracted in each document.

The design of the Sem-CNN model allows the integration of semantics in different ways as long as the semantic layer is encoded as a vector space model. In particular, Sem-CNN can integrate semantics using the two semantic vectorisation approaches discussed in section 3.3. In the next section, we compare both integration approaches in the particular context of fine-grained information identification in crisis-related tweets.

4 Experimental Setup

Here we present the experimental setup used to assess our event detection model. As described earlier, the aim is to apply and test the proposed model on the task of information-category detection in crisis-related tweets. As such, our evaluation requires the selection of: 1) a suitable Twitter dataset; 2) the identification of the most appropriate semantic extraction tool, and; 3) the identification of baseline models for cross-comparison.

4.1 Dataset

To assess the performance of our model we require a dataset where each tweet is annotated with an information-category label (e.g. *affected individuals*, *infrastructures*, etc.). For the purpose of this work we use the CrisisLexT26 dataset [19].

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*).

The tweets are also annotated with additional labels, indicating the information categories present in the tweet as listed in Table 1. More information about the CrisisLexT26 dataset can be found on the CrisisLex website.[9]

Note that in our experiments (Section 5) we discard the tweets' *related* and *unrelated* labels and keep only the *information type* labels since the task we experiment with focuses on the identification of information categories within crisis-related tweets.

Three data sets are used in this experiment:

– **Full Dataset**: This consists of all the 28,000 labeled tweets mentioned above.
– **Balanced Dataset 1**: Since the annotations tend to be unbalanced, we create a balanced version of our dataset by performing biased random under-sampling using tweets from each of the 26 crisis events present in the CrisisLexT26 dataset. As a result, 9105 tweets (32.6%) are extracted from the full dataset.
– **Balanced Dataset 2**: Besides the previous under-sampled dataset, we also consider an under-sampled dataset where only tweets that contain at least two semantic entities or concepts are extracted. The aim of this dataset is to better understand the availability of semantic annotations on the Sem-CNN dataset. After under-sampling the model with at least two entities and concepts for each tweet, we obtain 1194 tweets (4.3% of the tweets present in the full dataset).

Table 2 shows the total number of tweets and unique words under each of the three dataset subsets.

Table 2: Statistics of the three Twitter datasets used for the evaluation.

Dataset	No. of Tweets	No. of Words	No. of Word Embeddings	No. of Tweets with Extracted Entities
Full Dataset	27,933	57,563	16,617	18,298
Balanced Dataset 1	9,105	26,933	10,429	5,420
Balanced Dataset 2	1,194	5,671	3,540	1,194

4.2 Concept Extraction

As mentioned in Section 3, the Sem-CNN model integrates both, the entities' semantic concepts and abstracts of these concepts into the training phase of the classifier in order to better capture information-category clues in tweets.

Using TextRazor, we extract 4,022 semantic concept and entities and from those concepts and entities, we manage to match them to 3,822 unique abstract.

Looking at the different datasets, we notice that most of the semantics found in our dataset refer either to a type of event (e.g., *Earthquake*, *Wildfire*) mentioned in the tweets or to the place (e.g., *Colorado*, *Philippines*) where the event took place. This shows the value of using these types of semantics as discriminative features for event detection in tweets and may be beneficial for the identification of crisis-related information types.

---

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

4.3 Baselines

As discussed in Section 2, the task of event detection and information category identification in crisis-related documents in social media has been typically targeted by traditional machine learning classifiers (e.g., Naive Bayes, MaxEnt, SVM). Hence, in our evaluation we consider the following baselines for comparison:

- **SVM (TF-IDF):** A linear kernel SVM classifier trained from the words' TF-IDF vectors extracted from our dataset.
- **SVM (Word2Vec):** A linear kernel SVM classifier trained from the Google pre-trained 300-dimensional word embeddings [17].

In order to provide a thorough evaluation for our model, we also consider two additional variations of SVM as baselines: a SVM trained from the semantic concepts of words (SVM-Concepts) as well as a SVM trained from the semantic abstracts (SVM-Abstracts). Note that in [4], SVM was found to outperform other ML methods such as Naive Bayes and CART in various tasks on crisis-related tweets, and hence we focus our comparison here to SVM only.

5 Evaluation

In this section, we report the results obtained from using the proposed Sem-CNN model for identifying crisis-related information categories from social media posts. Our baselines of comparison is the SVM classifiers trained from TF-IDF, Word2Vec (pre-trained word embeddings), semantic concepts, and semantic abstracts features, as described in Section 4.3.

Table 3: Crisis information category detection performance of baselines and our proposed Sem-CNN model on the full and under-sampled datasets.

Model	Features	Semantics	Full Dataset P	R	$F1$	Balanced Dataset 1 P	R	$F1$	Balanced Dataset 2 P	R	$F1$
SVM	TF-IDF	-	0.644	0.604	0.617	0.608	0.610	0.607	0.555	0.548	0.540
SVM	Word2Vec	-	0.565	0.499	0.508	0.539	0.548	0.541	0.611	0.618	0.609
SVM	TF-IDF	*Concepts*	0.644	0.606	0.618	0.612	0.615	0.612	0.549	0.547	0.542
SVM	Word2Vec	*Concepts*	0.572	0.500	0.509	0.543	0.552	0.544	0.577	0.586	0.576
SVM	TF-IDF	*Abstracts*	0.633	0.590	0.603	0.595	0.598	0.595	0.499	0.499	0.495
SVM	Word2Vec	*Abstracts*	0.541	0.455	0.467	0.506	0.517	0.506	0.502	0.511	0.497
SEM-CNN	CNN-Embed	*Concepts*	0.645	0.600	0.621	0.627	0.625	0.626	0.675	0.601	0.636
SEM-CNN	CNN-Embed	*Abstracts*	0.646	0.604	0.624	0.628	0.628	0.628	0.676	0.608	0.640

We train the proposed Sem-CNN model using 300 long word embeddings vectors with $F_n = 128$ convolutional filter of sizes $F_s = [3, 4, 5]$. For avoiding over-fitting, we use a dropout of 0.5 during training and use the ADAM gradient decent algorithm [14]. We perform 2,000 iterations with a batch size of 256.

Table 3 shows the results computed using 5-fold cross validation for our crisis information category classifiers on the full dataset, the balanced dataset sample, and the two balanced dataset samples. In particular, the table reports the precision (P), recall (R), and F1-measure (F1) for each model and dataset. The table also reports the types of features and embeddings used to train the different classifiers.

5.1 Baselines Results

From Table 3 we can see that identifying information categories within crisis-related messages is a challenging tasks, where both, the SVM models produce relatively low results that vary between 46.7% and 61.8% in average F_1, based on the type of training features and dataset.

For the full dataset, we notice that SVM trained from Word2Vec features only gives 56.5%, 49.9% and 50.8% in P, R and F_1 measures respectively. However, using SVM with TF-IDF features improves the performance substantially by around $+18.83\%$ yielding in 64.4% P, 60.4% R and 61.7% F_1.

A similar performance trend can be observed under the balanced dataset 1, where SVM with TF-IDF gives higher performance than SVM with Word2Vec features although the performance of SVM with either type of feature on this datasets stays similar to the one reported under the full dataset.

For the balanced dataset 2, we notice a different trend. Here, Word2Vec features seem to outperform TF-IDF features by $+10.6\%$ in all measures on average. This might be due to the small size of this dataset in comparison with the size of the full dataset and the balanced dataset 1 as shown in Table 2. This issue is further discussed in Section 6.

The second part of Table 3 shows the performance of our baselines when semantic features are added to the feature space of the SVM models. Here we can observe that SVM classifiers trained either from concepts or abstract features do not have much impact on the overall performance. In particular, SVMs trained from concept features under both, the full and balanced dataset 1 give up to 61.8% F_1, which is in general similar to F_1 of a SVM trained from TF-IDF features solely. Nonetheless, on the balanced dataset 2 the performance when using concept features with SVM drops. It is also worth noting that using semantic abstracts as features for event information classification yields in more noticeable changes in the classification performance. In essence, the performance in this case drops even further compared with the concept features.

The above results suggest that plainly using semantic concepts or abstracts with traditional machine learning classifiers (SVM in this case) for identifying crises-related information categories has no additional value on the performance of these classifiers and that more complex classifier are necessary in order to integrate semantic concepts and entities efficiently.

5.2 Sem-CNN Results

In general, we observe that the Sem-CNN models needs relatively few steps in order to obtain the best F_1 results with the models converging around 400-600 steps (Figure 3).

The third part of Table 3 depicts the results of the proposed Sem-CNN model. From these results, we notice that Sem-CNN trained either from the concepts or abstract features yields noticeable improvement in the identification performance on all the three datasets. In particular, applying Sem-CNN on the first two datasets (full and balanced dataset 1) increases $P/R/F_1$ on average by $+1.19\%$ compared to SVM with TF-IDF and concepts features (the best performing baseline model).

On the balanced dataset 2, we noticed that Sem-CNN gives the highest detection performance with 63.6% F-measure for concepts features and 64% F-measure for the abstracts features. This represents $+17.71\%$ F-measure average increase in performance

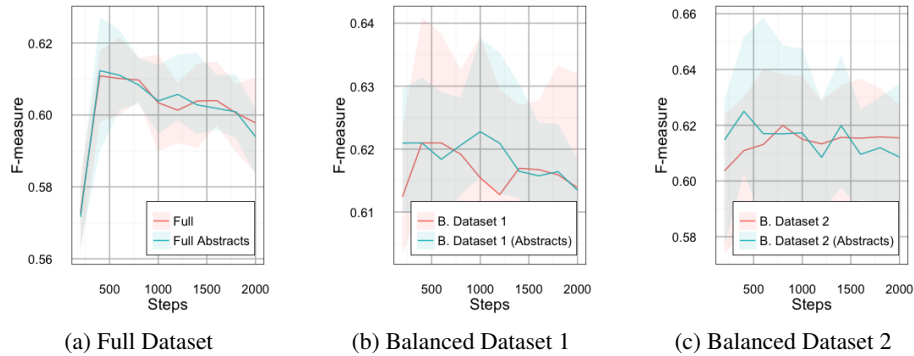(a) Full Dataset   (b) Balanced Dataset 1   (c) Balanced Dataset 2

Fig. 3: F-measure against the number of training steps for Sem-CNN on each dataset with concept labels and concept abstracts.

upon using the traditional SVM classifier on this dataset. These results show that our semantic deep learning model is able to use the semantic features of words more efficiently than SVM and find more specific and insightful patterns to distinguish between the different types of event-related information in tweets.

The significance of the results obtained by Sem-CNN against the best semantic baselines (SVM TF-IDF with concepts or abstract) can be compared by performing paired t-tests. We observe that the Sem-CNN with concepts and Sem-CNN with abstracts models mostly significantly outperform their SVM TF-IDF counterparts in term of F-measure (with $p < 0.001$ for Sem-CNN with abstracts for the balanced dataset 1 and 2; $p < 0.01$ for Sem-CNN with concept for the balanced dataset 2, and; $p < 0.05$ for Sem-CNN with concept for the full dataset). The only non-significant cases appears to be Sem-CNN with abstracts on the full dataset ($p = 0.062$) and Sem-CNN with concepts on the balanced dataset 2 ($p = 0.146$). The difference in F-measure for the Sem-CNN with abstract and Sem-CNN with concepts is non-significant ($0.395 < p < 0.092$) meaning that in general both approaches can be used with similar results.

6 Discussion and Future Work

In this paper we presented Sem-CNN, a semantic CNN model designed for identifying information categories in crisis-related tweets. This section discusses the limitations of the presented work and outlines future extensions.

We evaluated the proposed Sem-CNN model on three data samples of the CrisisLexT26 dataset and investigated two related methods for integrating semantic concepts and entities into the *wide* component of our model. Results showed that identifying information categories in crisis-related posts is a highly challenging task since tweets belonging to a given event contain, in many cases, general terms that may correspond to several categories of information [4]. Nevertheless, we showed that our deep learning model outperforms the best machine learning baselines, with an average gain between $+0.48\%$ and $+22.6\%$ in F-measure across each dataset subset. Compared to the best baselines, the proposed models significantly outperformed the best baselines in 67% of the cases ($p < 0.05$).

When creating our model, we used the DBpedia abstracts (`dbo:abstract`) of concepts in addition to their labels (`rdfs:label`) in order to add additional semantic context to the Sem-CNN model. Results showed a minimal average increase of $+0.3\%$ ($0.395 < p < 0.092$) in F-measure when using DBpedia abstracts in comparison with solely using semantic concepts. Despite the non-significance of such improvement, we can speculate that such small increase in F-measure might be attributed to the inclusion of more detailed descriptions of the abstract concepts that are often identified by entity extraction tools. This can be taken as a small demonstration of the potential value of expanding beyond the simple labels of concepts in such analysis scenarios. One obvious next step would be to replace, or extend, these abstracts in our model with semantics extracted from these abstracts. This could help refining and extending the concept labels used in the Sem-CNN model.

The proposed semantic *wide* and *deep* CNN model is built on top of a CNN network and a *wide* generalised linear model. Our model assumes that all inputs (i.e., words and semantic concepts and entities) are loosely coupled with each other. However, it might be the case that the latent clues of the information categories 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 this information in our detection model, probably by using recurrent neural networks (RNN) [8] due to their ability to capture sequential information in text or by using Hierarchical Attention Network (HAN) [30] in order to allow the model to focus on key semantic concepts and entities. Another direction would be by moving from the back-of-concepts representation used in our model to the back-of-semantic-relations [25]. This can be done by extracting the semantic relations between named-entities in tweets (e.g., $Tsunami < location > Sumatra$, $Evacuation < place > HighPark$) and use them to learn a more effective semantic vector representation similarly.

We also plan to better optimise our model by adding additional layers and performing parameter optimisation. Results could also be improved modifying the size of the model filters as well as the number of filters present in the *deep* part of Sem-CNN. In our experiments, we used the general Google pre-trained 300-dimensions word embeddings. Although previous work showed that not using pre-trained embeddings only slows down the learning phase of similar CNN models,[4] it would be interesting to experiment with embeddings tailored to social media such as pre-trained Twitter embeddings.[10]

In our evaluation we merely relied on SVM as a baseline and a case study of traditional machine learning baseline. This is because in our previous work [4] SVM showed to outperform other ML models (e.g., Naive Bayes, MaxEnt, J48, etc.) in identifying information categories in tweets. Those results are discussed in detail in [4].

We experimented with the SVM model using TF-IDF and Word2Vec features. Results showed that while TF-IDF features outperform Word2Vec features on both, the full and balanced 1 datasets, Word2Vec gives higher performance on the balanced dataset 2. This might be due the small size of the balanced dataset 2. As shown in Table 2, the balanced dataset 2 comprises 4.3% of the tweets in full dataset only, which may have had impact on the performance of these two types of features. We plan to further investigate

---

[10] Twitter Word2Vec model, `http://www.fredericgodin.com/software`.

this issue by extending our experiments to cover more datasets with different sizes and characteristics.

7 Conclusion

Very large numbers of tweets are often shared on Twitter during crises, reporting on crisis updates, announcing relief distribution, requesting help, etc. In this paper we introduced Sem-CNN, a *wide* and *deep* CNN model that uses the conceptual semantics of words for detecting the information categories of crisis-related tweets (e.g., *affected individuals, donations and volunteer, emotional support*).

We investigated the addition of the semantic concepts that appear in tweets to the learning component of the Sem-CNN model. We also showed that using semantic abstracts can marginally (i.e. non-significantly) improve upon semantic labels when integrating semantics into deep learning models.

We used our Sem-CNN model on a Twitter dataset that covers 26 different crisis events, and tested its performance in classifying tweets with regards to the category of information they hold. Results showed that our model generally outperforms the baselines, which consist of traditional machine learning approaches.

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. Burel, G., Saif, H., Fernandez, M., Alani, H.: On semantics and deep learning for event detection in crisis situations. In: Proceedings of the workshop on Semantic Deep Learning (SemDeep) at 14th Extended Semantic Web Conf. (ESWC). Portoroz, Slovenia (2017)
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: Proc. Annual Meeting of the Association for Computational Linguistics (ACL). Beijing, China (2015)
7. Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al.: Wide & deep learning for recommender systems. CoRR abs/1606.07792 (2016)
8. 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: Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar (2014)
9. Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N.: Improving efficiency and accuracy in multilingual entity extraction. In: Proceedings of the 9th International Conference on Semantic Systems (I-Semantics) (2013)
10. Dos Santos, C.N., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: Proc. Int. Conf. on Computational Linguistics (COLING). Dublin, Ireland (2014)

11. Feng, X., Huang, L., Tang, D., Qin, B., Ji, H., Liu, T.: A language-independent neural network for event detection. In: Proc. Annual Meeting of the Association for Computational Linguistics (ACL). Berlin, Germany (2016)
12. Gao, H., Barbier, G., Goolsby, R.: Harnessing the crowdsourcing power of social media for disaster relief. IEEE Intelligent Systems 26(3), 10–14 (2011)
13. Kim, Y.: Convolutional neural networks for sentence classification. In: Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar (2014)
14. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: Proc. Int. Conf. on Learning Representations (ICLR). Banff, Canada (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. Meier, P.: Digital Humanitarians: How Big Data Is Changing the Face of Humanitarian Response. Taylor & Francis Press (2015)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
18. Nguyen, T.H., Grishman, R.: Event detection and domain adaptation with convolutional neural networks. In: Proc. Annual Meeting of the Association for Computational Linguistics (ACL). Beijing, China (2015)
19. Olteanu, A., Castillo, C., Diaz, F., Vieweg, S.: CrisisLex: A lexicon for collecting and filtering microblogged communications in crises. In: Proc. Int. Conf. Weblogs and Social Media (ICWSM). Oxford, UK (2014)
20. Olteanu, A., Vieweg, S., Castillo, C.: What to expect when the unexpected happens: Social media communications across crises. In: Proc. ACM Conf. on Computer Supported Cooperative Work & Social Computing (CSCW). Vancouver, Canada (2015)
21. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar (2014)
22. Prasetyo, P.K., Ming, G., Ee-Peng, L., Scollon, C.N.: Social sensing for urban crisis management: The case of singapore haze. In: Proc. Int. Conf. on Social Informatics (SocInfo). Kyoto, Japan (2013)
23. Qu, Y., Huang, C., Zhang, P., Zhang, J.: Microblogging after a major disaster in China: a case study of the 2010 Yushu earthquake. In: Proc. ACM Conf. on Computer supported cooperative work (CSCW). Hangzhou, China (2011)
24. Rizzo, G., van Erp, M., Troncy, R.: Benchmarking the extraction and disambiguation of named entities on the semantic web. In: LREC. Reykjavik, celand (2014)
25. Saif, H., Dickinson, T., Leon, K., Fernandez, M., Alani, H.: A semantic graph-based approach for radicalisation detection on social media. In: European Semantic Web Conference. Portoroz, Slovenia (2017)
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: Proc. ACM Conf. on Computer supported cooperative work (CSCW). Savannah, Georgia, USA (2010)
27. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP). Lisbon, Portugal (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: Proc. Int. ISCRAM Conf. Vancouver, Canada (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: Proc. Conf. Human Factors in Computing Systems (CHI). Atlanta, GA, USA (2010)
30. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for document classification. In: HLT-NAACL. pp. 1480–1489 (2016)