Multimedia Tools and Applications (2023) 82:8921–8941
https://doi.org/10.1007/s11042-022-12183-w

1209: RECENT ADVANCES ON SOCIAL MEDIA ANALYTICS AND MULTIMEDIA SYSTEMS: ISSUES AND CHALLENGES

Classification of crisis-related data on Twitter using a deep learning-based framework

Nayan Ranjan Paul[1] · **Deepak Sahoo**[2] · **Rakesh Chandra Balabantaray**[1]

Received: 30 November 2020 / Revised: 10 August 2021 / Accepted: 8 November 2021 /
Published online: 16 February 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

In recent years, many citizens use social media platforms like Twitter to share and get the most up-to-date information regarding crisis events such as natural and man-made crisis. Automatically identifying this crisis-related information from social media data is a challenging task because of the sheer amount of data is being communicated between users during such crisis situations. The general public, response groups, and relief agencies can increase situational awareness by identifying and assessing crisis-related information from these massive amounts of social media data in real-time. Many studies have been published, that employ traditional machine learning approaches to detect crisis events, as well as others that use a deep neural network. In recent years, the models based on the deep neural network have outperformed traditional machine learning models for a variety of tasks. Two popularly used deep neural network models are Convolutional Neural Network (CNN) and the Gated Recurrent Unit (GRU). The local features can be detected by CNN in a multidimensional field, while the GRU network can learn sequential data because it can remember previously read data. In this paper, we propose two novel hybrid deep neural network models. The first model combines CNN and GRU and the second one combines CNN with SkipCNN. We evaluate our proposed models on 4 different datasets provided by CrisisNLP to show their effectiveness in detecting crisis-related information as well as identifying different types of information required for humanitarian aid. We find that from our proposed models CNN-SkipCNN is the best performing model and achieving better results than the state-of-the-art methods with an improvement of up to 16.55 absolute points for detecting crisis-related events and with an improvement of up to 21.71 absolute points in detecting different types of crisis information.

Keywords Event detection · SkipCNN · Classification · Deep learning · Crisis event

✉ Nayan Ranjan Paul
c116008@iiit-bh.ac.in

Extended author information available on the last page of the article.

1 Introduction

With the increasing growth and use of the Internet, social media platforms have become the main sources of communication. Social media platforms such as Twitter, WhatsApp, Facebook, Instagram, and others have recently enabled millions of individuals to broadcast news and real-time information about various events as they happen on the ground. These social media platforms produce huge amounts of information. These large amount of information obtained from social media are getting utilised for different applications such as event detection [14], sentiment analysis [30, 35, 43], rumour veracity detection [31], question answering system [48], recommender system [1], hate speech detection [55] and many more.

In recent years, information and communication technology (ICT) is widely being used for various crisis situations to speed up relief work [42]. During crisis situations, people use social media to post real-time information as text messages, images, or videos, to pass on information to near and dear [3]. The information that passed may include asking for help or services, damage estimation, situations monitoring, and reporting the status [34]. In the early days, people used conventional techniques like messages, phone calls, and directly asking people in need to gather information. However, nowadays by analyzing social media content may be helpful to collect real-time actionable information for damage assessment and mapping appropriately the necessary resources like food, water, shelter, doctor, and medicine at the time of need [45].

Here, we mainly focus on Twitter because Twitter is a popular micro-blogging social media platform. Twitter users can post text messages of up to 280 characters called tweets and also can post images, audio, and videos. It has 330 million monthly active users and generates more than 500 million daily tweets. Twitter has been widely utilized as a helpful source of information in a variety of crisis circumstances, including floods, earthquakes [40], fires [51], cyclones, and nuclear disasters [49].

Although the most important thing in the aforementioned crisis is actionable information to help those affected by or who will be affected during these crisis, it is not feasible for communities and various aid organizations to manually process such large numbers of twitter data in this crisis to meaningful, feasible and sensible information [17]. According to research, leveraging Twitter data in crisis circumstances is critical. It has also been demonstrated that information disseminated via Twitter may raise awareness in crisis situations among the general public, emergency agencies, and relief organizations.

There are several works that utilize supervised and unsupervised machine learning models such as language models, classifiers, and clustering to detect events from twitter data. Deep learning models have recently become a popular approach. In comparison to the traditional machine learning approaches, deep learning models deliver considerable improvements for a variety of tasks. Deep learning models have the benefit of being able to capture several levels of information. This encourages the usage of a deep neural network model in this work.

In this paper, we focus on the detection of events in crisis situations and detect information types mentioned in these tweets with deep learning on twitter data. Our objective is to detect crisis-related events using neural networks. We utilize two distinct neural network models which combine the popular Convolutional Neural Networks (CNNs) with Gated Neural Networks (GRU) and CNN with skippCNN and compare accuracy against regular other state of art models. The remainder of this paper is organized as follows. The related work is explained in Section 2. The application scenario and description of the model are

presented in Sections 3 and 4 respectively. The experimental setup is explained in Section 5. Result analysis and in-depth discussion are presented in Section 6 and the conclusion and future work is presented in Section 7.

2 Related work

Classifying and detecting social media messages, particularly Twitter messages about different man-made or natural crisis has been addressed by a number of researchers. Various machine learning methods have been proposed for the automatic detection and classification of tweets. All these methods which use machine learning approaches can be categorized into "classic machine learning method" and "deep learning method".

2.1 Classic machine learning methods

These methods uses traditional machine learning approaches, which depends on manual feature engineering that has been consumed by algorithms like SVM (Support Vector Machine), Naive Bayes, logistic regression [19, 39, 47].

Sakaki et al. [44] use tweets to detect crisis like earthquakes using three types of features. The first type of feature is based on tweets statistics like the number of words in a tweet, position of the query term in a tweet. The second type of feature is based on keywords like the actual words in a tweet. The third type of feature is called contextual feature which includes, words appearing near the query term. Furthermore, they use spatiotemporal information to find the event's location. They use SVM supervised algorithm for the classification task. This work did not consider identifying the type of information from these tweets.

Imran et al. [18] filters crisis-related tweets for natural disasters as well as extract valuable information nuggets relevant to disaster response. The authors consider three types of features namely binary features, scalar features, and text features. Binary features such as whether a tweet contains the @ symbol, a URL, a hashtag, an emoticon, a number, or not. Statistical features like tweet length and text features like unigrams, bigrams, POS tags are used. The authors use the Naive Bayes classifier for the classification task. This model did not automatically select the features rather uses manual feature selection for the classification task.

Karmi et al. [23] propose methods to classify whether tweets are related to disaster or not and also classify them into six different disaster types. The authors employ binary features like whether a specific hashtag, user mentions, and links are available in a tweet or not. They also employ statistical features like hashtag counts, tweet lengths, mention counts, unigrams, bigrams, and bag-of-words. SVM and multinomial Naive Bayes algorithms are used for the task. The authors did not consider the task of identifying information for humanitarian aid.

Khare et al. [27] uses both statistical features with semantic features for classifying tweets related or not related to disasters. For statistical features, the authors use a number of nouns, verbs, pronouns, words, hashtags, tweet length, and unigrams. For semantic features, they use semantic annotation using Bablify and Babelnet by extracting every direct hypernym from the Bablenet Knowledge base and semantic filtering to filter every generic concepts with low discriminative power. The authors use SVM with a linear kernel for the classification task. The dataset used in this task is small which has only 3206 tweets, which

affects the accuracy of classification. They also did not consider the task of identifying information for humanitarian aid.

Kevin et al. [47] classify tweets that are relevant or not to disaster events. The authors use statistical features like unigrams and bigrams count, binary features such as to indicate whether a tweet is a retweet or not, they also used time of tweets, POS-tags as well as named entities. For the classification task SVM, Naive Bayes, and maximum entropy models are used and SVM gives better results than the other two. This work did not do well for classifying fine-grained information due to manual feature selection because the features used are not working well compared to deep learning-based models. Another bottleneck of this work is the data sparsity as many classes lack positive examples.

Zhang et al. [56] uses a semi-supervised learning method to classify tweets related to a disaster or not. Authors use Brown clustering and clustering based on word2vec with feature selection based on traditional bag-of-words. For learning algorithms, they use logistic regression with L2 regularization. One bottleneck with this method is that the accuracy is strongly influenced by the available number of tweets. They did not consider the task of identifying information for humanitarian aid.

Resch et al. [41] analyses social media data by using a method that combines spatial and temporal analysis along with semantic machine learning technique(LDA) to identify the place and damage caused by the disaster. The central advantage of LDA is its transferability to other text corpora, and languages because of the rapid adaptation of the unsupervised learning approach to another text corpus. The problem with this method is the creation of a consistent technique for finding the optimum parameters for semantic analysis.

Verma et al. [50] presented a method based on NLP techniques along with NaiveBayes and maximum entropy to identify tweets that contain situational awareness information during crisis situations. One problem with this work is false positives, which are of greater concern since they represent noise that could be misleading. This work uses manual feature selection for this task.

Imran et al. [19] used NaiveBayes classifiers for classifying different types of informative tweets. After classifying, the authors used a sequence labeling algorithm CRF, to extract useful information nuggets. One disadvantage of CRF is the high computational complexity of the training stage of the algorithm. This fact makes it more difficult to re-train the model when new training data samples become available.

Imran et al. [21] used a random forest classifier that learns from a source disaster to classify a target disaster. They also showed that the data related to previous disasters can be helpful in clustering similar types of current disasters across various languages.In this work they did not consider how to detect humanitarian information from the disaster tweets.

Some authors have also used Latent Dirlect Allocation [22, 52], ensemble learning learning [25], Naive Bayes models [33], and random forest [24] for this task. All these works mentioned here uses traditional machine learning approaches which require manual feature engineering but deep learning approaches did not require manual feature engineering rather it automatically detect the features, So we prefer to use deep learning method for this problem.

2.2 Deep learning methods

These methods use deep artificial neural networks with stacked multiple layers of neurons which learns abstract features from input data for the classification of disaster-related tweets. Here, the inputs may be simple raw text data or any feature encoding. However, the main difference from classical machine learning methods is that this type of model does

not need manual feature engineering and the input features may not be used directly for classification, rather it uses a multi-layer structure to learn features directly from the input. Due to this reason, deep learning-based methods mostly focus on developing good network structures rather than focusing on manual feature engineering. These models automatically extract features from simple input feature representation. We notice that there is a growing trend in literature towards approaches focused on deep learning.

To the best of our understanding, approaches in these categories include [6–8, 26, 28, 36], which uses simple word or character-based encoding as input features to their model. The most popular architecture is Convolutional Neural Network(CNN) and Recurrent Neural Network(RNN) and their variations. Caragea et al. [9] first employed CNN for the classification of tweets during disaster events, which is able to predict the informative tweets and filter out the tweets that are not informative in nature. The authors show that accuracy of classification increases in comparison to the state-of-the-art classical ML techniques. Burel et al. [7] presented a variation of CNN called Sem-CNN which is a semantically enhanced wide and deep convolutional network model to identify categories of crisis-related information. This model integrates an additional layer of semantic that represents the named entities in the text into a wide and deep CNN network. D.T. Nguyen et al. [36] uses the CNN model and a variation of CNN called MLP-CNN, which is a CNN model with a multi-layer perceptron. it has outperformed all the classic models for classifying disaster-related tweets. Burel et al. [8] presented another variation of CNN called Dual-CNN which extends Kim's model [28] with an additional semantic representation layer representing the named entities in tweets and their associated semantic sub types. The authors use word embedding for one CNN and semantic embedding for another CNN. Kirsten et al. [26] use the same CNN model proposed by Kim [28] for classifying disaster-related tweets. Burel et al. [6] introduced CREES (Crisis Event Extraction Service), a web API that allows the automatic classification of crisis-related tweets. This API uses the CNN model. Firoj et al. [2] propose a semi-supervised framework that combines CNN with a graph-based network that learns internal representations of the input for classifying tweets in a disaster situation. Ashutosh et al. [5] proposed a novel hybrid model based on CNN to classify tweets in crisis situations. They also provide a method to find a ranked list of tweets for respective topics and also mapped identified need tweets with their corresponding availability tweets.

The above-mentioned methods use the most popular network architecture CNN and its variations. CNN can obtain local features from the tweets by considering that all words of tweets are loosely coupled with each other but actually the words have intrinsic dependencies with each other. These word dependencies are very much vital to capture the latent clues of the information categories. This information can be captured by using Recurrent Neural Networks(RNN). To the best of our knowledge, no work has explored combining CNN and RNN for this task. There are several studies conducted that are found to be more effective when CNN and RNN are combined than structures solely based on CNN or RNN in tasks such as gesture recognition [10], activity recognition [54], and named entity recognition [53], recommender system [32].

3 Application scenario

During a crisis, individuals frequently utilize various social media platforms to submit messages and photos of the disaster. As a result, these sites are flooded with millions of messages. However, many of these messages are irrelevant or uninformative. According to

Olteanu et al. [37], crisis reports could be divided into three main categories of informativeness. They are 'related and informative', 'related but not informative' and 'not related'". During the crisis, the proportion of relevant and useful social reporting varies between 10% and 65%, according to [46].

Our objective in this research is to develop several models in order to effectively detect the relevant messages. We evaluate the following task for this purpose, based on the event categories provided by [37].

- *Task 1 - Tweets related to crisis Vs Not related to crisis :* The goal of this task is to discriminate between tweets that are related to a crisis and tweets that are not related to a crisis.
- *Task 2 - Type of Information :* The aim of the task is to automatically obtain fine-grained crisis-related information for humanitarian aid. We consider six different categories of crisis-related information which is based on the work of (Olteanu, Vieweg, et al. 2015 [37]). These are: 'Affected people', 'Donations and volunteering', 'Infrastructure and utilities', 'Sympathy and support', 'Other useful information', and 'Not related or irrelevant'.

4 Proposed model architectures

In the crisis event classification task, our hypothesis is that a structure that combines a CNN with RNN can be more effective as it may recognize co-occurring words as useful patterns for classification. To illustrate this consider the tweet "Dua's for all those affected by the earthquake in India, Nepal and Bhutan. Stay safe and help others in any form ...". In this tweet each of the individual words 'affected', 'earthquake', 'India', 'Nepal', 'Bhutan', 'safe', 'help', Others" alone are always not indicative features of crisis event because they can also be used in any other context. However combinations such as ('affected by', 'earthquake'), ('earthquake', 'India'), ('earthquake', 'Nepal'), ('earthquake', 'Bhutan'), ('earthquake', 'stay safe'), ('earthquake', 'help others') may be used as more indicative features. These pairs of words show some kind of dependency on each other. These can not be captured by n-gram like features.

We propose two deep neural models that may capture such types of features automatically. First, we described a traditional CNN model which will act as a base model. Our proposed model is also based on this traditional CNN model. Our first model combines a traditional CNN with a GRU (Gated Recurrent Unit). Our second model combines the traditional CNN with some skipped CNN layers which will serve as a skip-gram extractor to be called as 'SkipCNN'. Both the proposed models modify the common traditional CNN (Section 4.3.1) that acts as extracting n-gram features while the modified CNN-GRU (Section 4.3.2) and CNN-SkipCNN (Section 4.3.3) are expected to obtain the dependent sequence of words as described above.

The framework is presented in Fig. 1 which contains three main phases as follows:

1. *Tweet preprocessing* - All of the tweets available in the datasets are cleaned and tokenized for further use.
2. *Word vector initialisation* - A word embedding matrix is produced in this step using a pre-trained word embedding and a bag of words from the previous stage. The model will be trained in the following stage using this word embedding matrix.

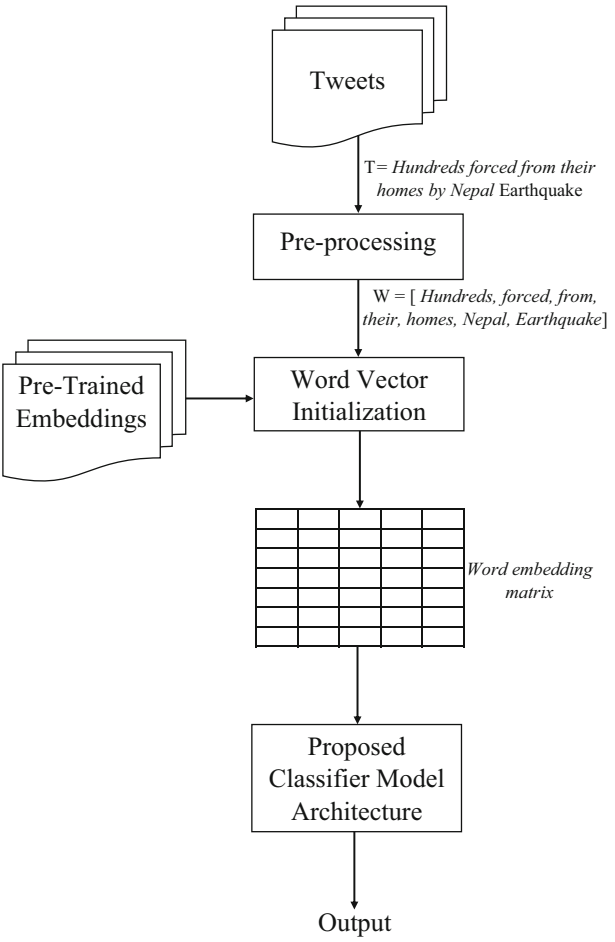

Fig. 1 The proposed Model framework

3. *CNN-GRU and CNN-SkipCNN model training* - The models are trained using the word embedding matrix acquired in the previous phase.

We go through each phase of the framework in further detail in the subsections below.

4.1 Tweet preprocessing

Tweets frequently contain grammatical mistakes, poorly constructed phrases, and incomplete sentences as a result of the frequent usage of ill-formed words, irregular expressions, short forms, emoticons, and non-dictionary terms. As a result, we use tweet preprocessing, which is a series of preprocessing procedures applied to each tweet in the datasets, to decrease tweet noise. All URLs, emojis, non-English and non-ASCII characters, hashtag symbols, and user mentions are removed. The preprocessing step also involves replacing contractions with its equivalent words, like "can't" will become "can not", elongated word normalization, like "yeeees" will become "yes", and segmentation of hashtags like

"#StaySafe" will become "Stay Safe". Then we apply lemmatization on each word to bring it to its dictionary form. Following the application of the aforementioned approaches, these tweets are tokenized into word tokens, which are then fed into the word vector initialization step.

4.2 Word vector initialization

This phase's major goal is to create a word embedding matrix, which is required to train classification models. This word embedding matrix is essential for using deep neural networks for classification tasks. The word tokens obtained from the tweet preprocessing step is passed to a word vector initialization phase, which transforms the sequence of words into a numeric vector called word embedding [4]. The basic idea behind word embedding is that similar meaning words should have a similar vector representation. Specifically in word embedding each word in the sequence is mapped to a fixed dimension numeric vector. Generally, word embeddings use a large corpus of data for training, but we do not have that large corpus of data as our datasets are small. So we used pre-trained word embedding 'GloVe' [38], which is publicly available. One problem with the use of pre-trained word embedding is Out-Of-Vocabulary (OOV) words because tweets use some words which may not be meaningful. Thus during preprocessing, this type of meaningless words can be filtered out which intern reduces the scale of OOV. For example, by hashtag segmentation, we transform an OOV "#NepalEarthquake" into "Nepal" and "Earthquake". These two words have a higher chance of being included in the pre-trained embedding model. To handle this OOV we are planning to use some other embeddings like: Crisis embedding, Conceptnet Numberbatch embedding in future to handle this task in a better way.

4.3 CNN-GRU and CNN-SkipCNN model training

This is the phase in which the proposed models are trained. For training the models, the word embedding matrix produced from the preceding word vector initialization phase is supplied into the proposed CNN-GRU and CNN-SkipCNN models. Both the proposed models are based on a traditional CNN model which acts as a base model. This base model("Base CNN Model") is described first in Section 4.3.1 and then the proposed CNN-GRU and CNN-SkipCNN model is described in Sections 4.3.2 and 4.3.3 respectively.

4.3.1 The base CNN model

The base CNN model is illustrated in Fig. 2. The word embedding matrix with a shape of 100 X 300 obtained from the word vector initialization phase is given as input to three different convolutional layers. These three convolutional layers use 100 filters each with a stride of 1 along with three different window sizes of length 2, 3, and 4 respectively. Here each CNN layer may be considered to act as bigram, tri-gram, and quad-gram feature extractors. The 'ReLU' function is used as activation for these CNN layers. The output of these CNN layers passes through dropout layers with a dropout rate fixed at 0.2. The output of these dropout layers are fed to the 1D max-pooling layer with a pool size of 4 and a stride of 4 for further down sampling the features. We also add another 1D max-pooling layer on top of that with the same configuration as the first one. Because we found that by adding another pooling layer we get an improvement of f1 score in most cases. The output of the max-pooling layer is given as input to the final softmax layer which will predict the probability distribution for all classes that will depend on different datasets.

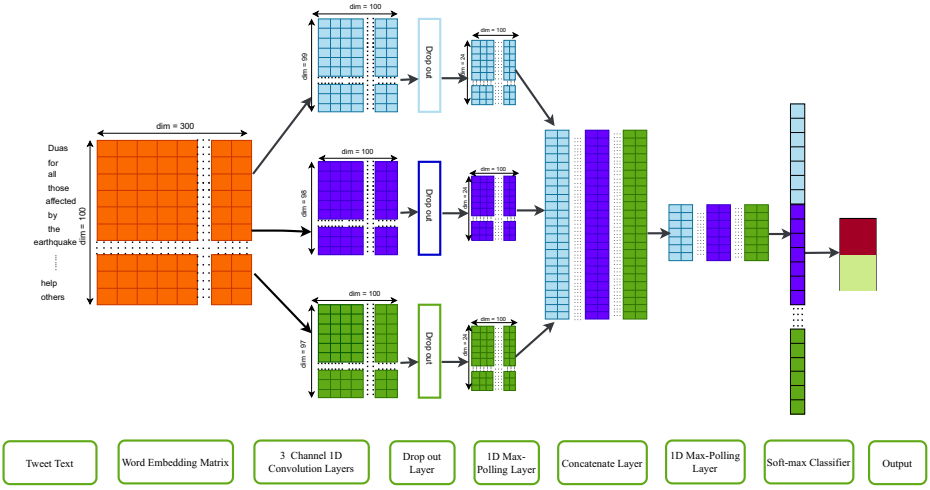

Fig. 2 The base CNN model with three window sizes of 2,3, and 4 to extract features

4.3.2 CNN-GRU

This proposed model is an extension of the base CNN model described in Section 4.3.1 and presented in Fig. 3. A GRU layer is added above the CNN layer. We use GRU instead of the most popular LSTM because GRU has only two gates namely *reset* and *update gate* whereas LSTM has three gates namely *input*, *output*, and *forget gates*. So in comparison to LSTM, GRU has a simpler structure, so fewer parameters to be trained. This makes the GRU run faster to train than LSTM and generalizes well in the small dataset. It is shown in the literature that GRU achieves comparable results to LSTM [11].

The output of the max-pooling layer of the previous CNN model is given as input to the concatenation layer which concatenates the outputs of the max-pooling layer and fed

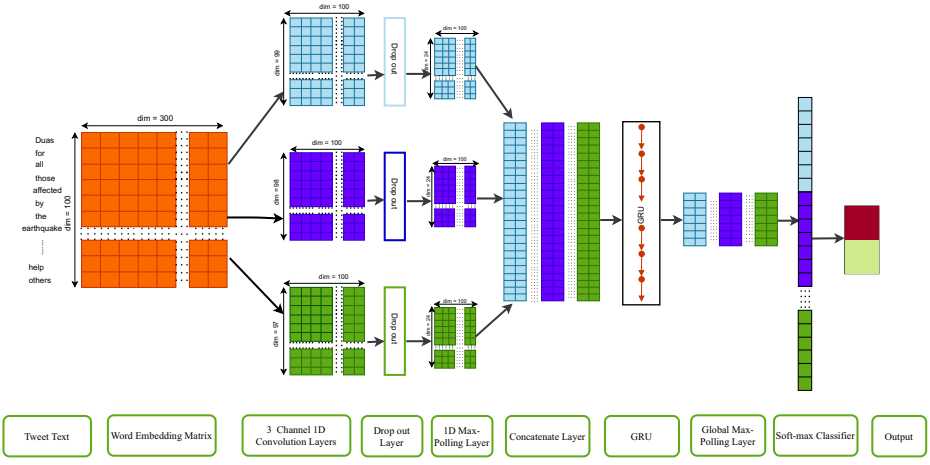

Fig. 3 The CNN-GRU Model

to the GRU layer. This outputs 100 hidden units per time step as it treats the features as timesteps. The GRU layer's output is supplied into a global max-pooling layer, which filters the output space by picking the highest value in each timestep dimension and outputs a feature vector, which is then fed into the softmax layer. Our intention is to choose the features with the highest score to represent a tweet that performs better than the usual setup.

We have used 1D max-pooling as well as global max-pooling layers in this model. The global max-pooling has the pool size the same as the input size so that the max vector over the steps dimension of the entire input is computed as the output value. Where as the 1D max-pooling takes the max over the steps too but is constrained to a pool size for each stride. In this model, we use 1D max-pooling to down sample the feature space, and global max-pooling is used to choose the feature with the highest score.

For the processing of text in the classification task, it is important to know the ordering of words as well as the dependency of words with each other in the text. The GRU layer can capture such orderings and learn dependency from the word n-grams provided by the previous CNN layers. As a result, these co-occurring words can be used as useful patterns for the classification task. We have already outlined such a pair of words and phrases before.

4.3.3 CNN-SkipCNN(SkCNN)

This model is an extension of the base CNN model. We add other CNN layers on top of the base CNN model. These extra CNN layers present on top of the base CNN uses 'skipped-window' which extracts features from the input and we called them 'SkipCNN'. If the inputs at certain adjacent locations of the window are skipped, then it is called a skipped window, which means these locations with the window is 'deactivated', while other locations are 'activated'. Algorithm 1 describes how to apply skipped-window to produce multiple shapes of windows (Fig. 4).

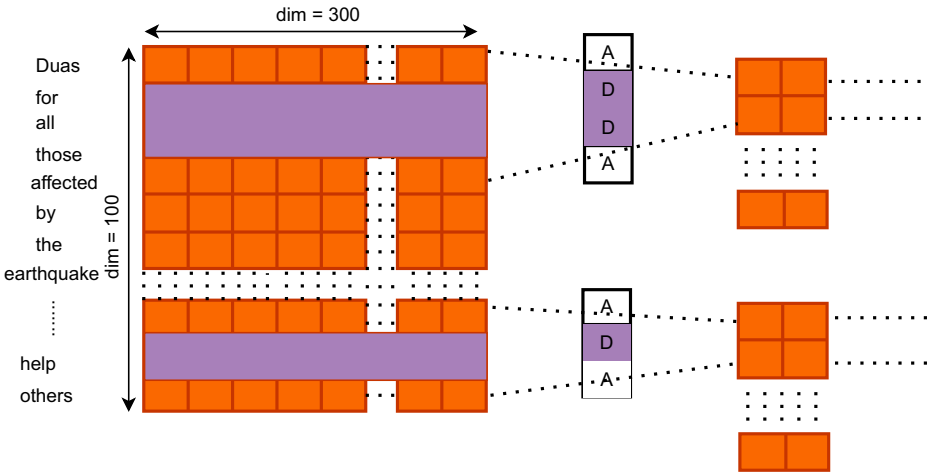

Fig. 4 Example of a window size 4 with gap 2 and a window size 3 with gap 1. The 'D' represents that for the corresponding window position, the input is ignored

Algorithm 1 Creation of skipped windows with r gapped size c. One possibility of a 1 gapped size 4 is [A, D, A, A], where 'A' present in the position represents activated and 'D' present in the position represents deactivated.

Input: $r : 0 < r < c, c : c > 0, w \leftarrow [l_1, ..., l_j]$
Output: WIN is a set containing window shapes of size c
Set $WIN \leftarrow \emptyset$
for all $n \in [2, c)$ and $n \in \mathbb{N}_+$ **do**
    Set $l_1 \leftarrow$ A in w
    Set $l_c \leftarrow$ A in w
    for all $x \in [n, n + r]$ and $x \in \mathbb{N}_+$ **do**
        Set $l_x \leftarrow$ D
        for all $y \in [n + r + 1, c)$ and $y \in \mathbb{N}_+$ **do**
            Set $l_y \leftarrow$ A in w
        end for
        $WIN \leftarrow WIN \cup \{w\}$
    end for
end for

For example, on a window size of 4, if we apply 1-skip, it will produce two shapes as [A, D, A, A], [A, A, D, A] where 'A' indicates 'Activated' and 'D' indicates 'De-activated' positions in a window. But on a window size of 4, if we apply a 2-skip, it will produce one shape as [A, D, D, A].

The traditional CNN model is extended by adding three other SkipCNNs of 1-skipped size 3 windows, 1-skipped size 4 windows, and 2-skipped size 4 windows where 1-skipped size 3 windows produce one shape as [A, D, A], 1-skipped size 4 windows produces two shapes as [A, D, A, A], [A, A, D, A], and 2-skipped size 4 windows produce one shape as [A, D, D, A]. Then, after each of these extra CNN layers, a max-pooling layer with pool size 4 and stride 4 is added. Other parts of the model are the same as the base CNN model. The resulted model is given in Fig. 5.

Our intuition is that the 'SkipCNNs' can extract 'Skip-gram' like features. For example, it can extract useful features such as ('affected by', 'earthquake'), ('affected', 'India'), ('affected', 'Nepal'), ('earthquake', 'Nepal'), ('earthquake', 'Stay safe') from sample tweet presented earlier. For both CNN-GRU and CNN-SkipCNN, a dropout layer with the dropout rate of 0.2 is added before each convolutional layer for regularizing the inputs.

5 Experimental setup

The experimental setup utilized to assess our event detection models is described in this section. The proposed models are applied and tested on the tasks described in Section 3. We must choose a dataset for this experiment, and the evaluation will be performed on that dataset. The working of the framework with example is also illustrated.

5.1 Twitter dataset

To assess the performance of these models, we require labelled datasets in which every tweet is labelled with whether it is associated with a crisis occurrence or not, and also labelled with what type of humanitarian information it conveys, out of six distinct types of information. For this experiment and assessment, we used the CrisisNLP dataset [20]. It includes tweets

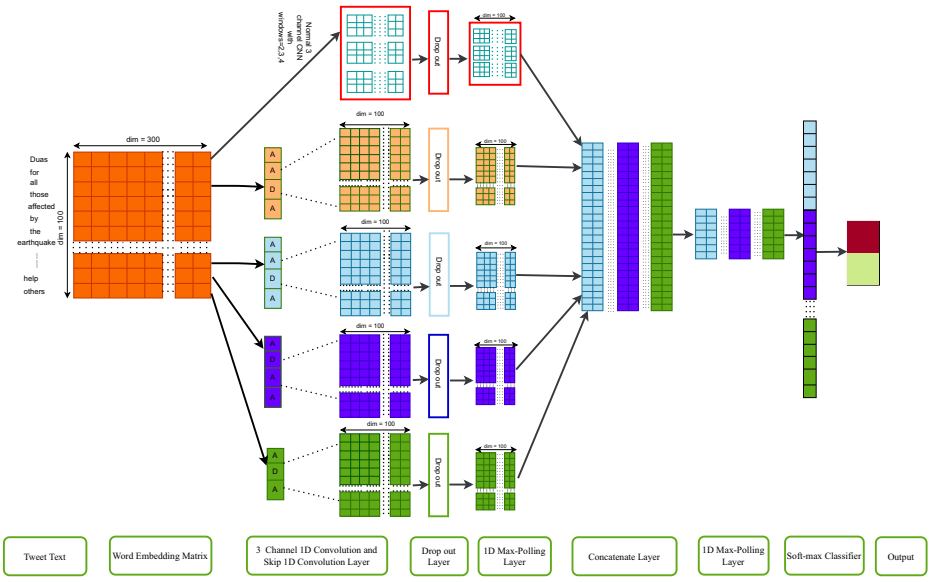

Fig. 5 The CNN-SkipCNN model

posted during various crisis events. It contains tweets for 10 crisis events such as earthquake, typhoon, land slides, cyclone etc. There are 13 datasets related to these 10 events. These datasets include tweets, some of which are linked to the crisis and others which are not. Out of these 10 event-related datasets, 10 are in English and the rest 3 are in other languages. We choose 4 datasets in English related to 4 events namely 'Nepal Earthquake', 'Cyclone PAM', 'California Earthquake', and 'Typhoon Hagupit'. All the tweets of these datasets are labelled into different classes according to the type of information they convey. Some of the class labels are 'donation and volunteering', 'immediate needs', 'damage to infrastructure', 'deceased or wounded people' and one 'unrelated or irrelevant class'. Table 1 provides all

Table 1 Class label names with small description in the datasets. Column name 'Labels' indicates the total number of annotations for each class

Class	Labels	Descriptions
Affected people	1416	Tweets with reporting of deaths, injuries, missing, found, or displaced people
Donations and volunteering	1979	Tweets containing donations information like food, water, shelter, services etc. or offering volunteer work
Infrastructure and utilities	1299	Messages reporting the damage of infrastructure and utilities
Sympathy and support	2139	Messages conveying sympathy and emotional support
Other useful information	7309	Messages containing useful information that does not fit in one of the above classes
Not related or irrelevant	14505	Irrelevant or not informative, or not useful for crisis response

the class labels along with their small description and also provides the total number of tweets for each class label from all 4 datasets. The most common classes of the datasets are 'other useful information' and 'not related'. The detail statistics about the events, we have used in our experiment are listed in Table 2.

D.T. Nguyen et al. in their work [31] have used the same CrisisNLP dataset which we have used in this work and they have calculated the inter-annotator agreement (IAA) scores. We have used their IAA scores of the datasets for accessing the difficulty of the classification task. The highest IAA reported is 0.85 for the California earthquake and the lowest IAA reported is 0.70 for Typhoon Hagupit. The IAA reported is around 0.75 for the other two events.

5.2 Model parameter

To train every model, we use the categorical cross-entropy as loss function because this loss function is described as more efficient for classification tasks than other available loss functions including mean squared error and classification error [29]. For training every model, we also use the Adam optimizer algorithm instead of the classical stochastic gradient descent (SGD) algorithm to update network weight based on training data. Because SGD uses the same learning rate for all weight updates and does not vary the learning rate throughout training. Adam, on the other hand, is an adaptive learning rate technique that calculates individual learning rates for various parameters. Adam is a popular deep learning algorithm because it combines the benefits of two popular extensions of SGD, one of which is the adaptive gradient algorithm (AdaGrad), which improves performance on problems with sparse gradients by maintaining a per-parameter learning rate by incorporating knowledge of past observations. The second method is the root mean square propagation (RMSProp), which adapts per-parameter learning rates based on the average of recent gradient magnitudes for the weight.

We chose the above parameters based on empirical findings which are previously reported. The settings of these parameters may not be the best for optimal results but in the next section, we demonstrate that without any data-driven parameter tuning, the models produce promising results. For implementing the proposed models in this work, we have used python, Keras with TensorFlow backend, and the scikit-learn module. We have fixed the size of each epoch to 10 and the size of mini-batch to 100 for each model on all datasets. These parameters are selected randomly and fixed for consistency.

Table 2 Class distribution of events under consideration

Class	EVENT			
	Nepal Earthquake	Typhoon Hagupit	California Earthquake	Cyclone PAM
Affected people	752	204	207	253
Donations and volunteering	1073	436	82	388
Infrastructure and utilities	348	352	366	233
Sympathy and support	1021	638	159	321
Other useful information	2585	3081	951	692
Not related or irrelevant	6695	6974	119	717
Grand Total	**12474**	**11685**	**1884**	**2604**

The bold values represents the total number of tweets for each crisis event dataset used for the experiment

5.3 Illustration of framework

The illustration of the framework depicted in Fig. 1 is explained with the follwing example, let us consider a tweet "RT @cctvnews Hundreds feared trapped in Kathmandu crumbled #Dharara Tower: local media http://t.co/TQXr08dBih". This tweet is given as input to the preprocessing phase and the output of the preprocessing phase is "hundred feared trapped kathmandu crumbled dharara tower local medium". This preprocessed tweet is fed to the word vector initialization phase which converts this into 100×300 size word embedding matrix. This word embedding matrix is given as input to the proposed models.

For base CNN model, the embedding matrix is given as input to the three CNN layers through three different channels. The output of convolution operation from each channel is passed through three different dropout layers and max-pooling layers. Then all three outputs of the max-pooling layer are concatenated and fed to another max-pooling layer and finally pass through the softmax layer which gives as "related" as output for the binary classification task. similarly for the multi-class classification task, the softmax layer produces "Affected People" as output.

For CNN-GRU model, the embedding matrix obtained for the same tweet from phase 2 is pass through the three CNN layers, dropout layers, and max-pooling layers. Then the output of max-pooling layers are concatenated by the concatenation layer and pass through the GRU layer, global max-pooling layer, finally, the output of the global max-pooling layer is gone through the softmax layer which gives output as "related" for binary classification task and "Affected People" for the multi-class classification task.

For CNN-SkipCNN model, the 100×300 size embedding matrix obtained for the above mentioned tweet is given as input to seven CNN layers through seven channels, out of which three CNN layers are similar to the base CNN model and the rest four CNN layers used as SkipCNN. The output of these convolution layers passes through respective dropout layers and max-pooling layers. The outputs of max-pooling layers are concatenated by the concatenation layer and again another max-pooling layer is used. Then it is taken by softmax layer which produces output as "Related" for binary classification problem and "Affected People" for multiclass classification problem.

6 Result analysis and discussion

For evaluating the effectiveness of our proposed models, we are comparing our model's performance with the models proposed by D.T.Nguyen's in their work [36]. There are 3 reasons for choosing D.T. Nguyen's model. The first reason is that the tasks under consideration for our work are the same as the task of D.T. Nguyen et al.'s work. The second reason is that the dataset used for model evaluation is the same. As described in Section 5.1, we have used a dataset for 4 events. They also used the same 4 datasets along with other datasets. They use three data settings in their work. They are event-only, out-of-event only, and a combination of both. We choose results of event-only data settings because we did not consider the other two data settings in our work. The third reason is that in both works, CNN-based models are used. So the comparisons will be more straight forward and accurate. The CNN based models of D. T. Nguyen et al. are CNN_I, CNN_{II} for binary classification problem and CNN_I, MLP-CNN_I for multi-class classification problem. CNN_I is a CNN model which uses crisis

embedding for its initialization and CNN_{II} is the same CNN model but uses google embedding for its initialization. MLP-CNN_I is obtained by adding a multi-layer perceptron to the CNN_I model. For the purpose of comparison, we have directly used the results reported for CNN_I, CNN_{II} along with the results reported for traditional approaches which include support vector machine (SVM), Logistic Regression (LR), and Random Forest (RF) for binary classification and used the scores of CNN_I, MLP-CNN_I for multi-class classification.

Binary classification (Task 1) Table 3 presents the results of binary classification comparing different classifiers using the AUC(Area Under Curve) score. The area under the curve (AUC) is a performance measurement for the classification problems at various threshold settings. The AUC is a metric that indicates the degree of separability. It indicates how well the model can discriminate between classes. The larger the AUC, the better the model predicts or distinguishes between classes. We have used the AUC score of classifiers CNN_I, CNN_{II}, SVM, RF and LR reported by D.T. Nguyen et al. in their work [36]. It is clear from the table that all neural network-based models outperformed the non-neural based models. Both of our CNN-GRU and CNN-SkipCNN models consistently obtain the best results compared to other models. Even our Base CNN model also outperforms the CNN models reported in [36]. Between CNN-SkipCNN and CNN-GRU, the CNN-SkipCNN performed better than CNN-GRU, but the improvement is rather incremental. If we compare our best performing model 'CNN-SkipCNN' with the best model reported in [36], the improvement of our model is 9.52 absolute points in the 'Nepal Earthquake' dataset, 16.55 absolute points in 'California Earthquake', 4.77 absolute points in 'Typhoon Hagupit' and 5.21 absolute point in 'Cyclone PAM' dataset.

Multi-class classification (Task 2) Table 4 presents the results of multi-class classification comparing different classifiers using Macro f1 and accuracy as measures. The Macro f1-score (macro-averaged f1-score) is defined as the arithmetic means of the per-class f1-scores, where f1-score is defined as the harmonic mean of the model's precision and recall. The quality of models with multiple classes can be evaluated by using this macro f1-score. Each class will be given as same weight in this macro f1-measure. Models will get a low macro f1-score that only performs well on common classes while doing badly on uncommon classes. Both models outperformed the state-of-the-art SVM model and two other DNN models. Among the two proposed models, the CNN-SkipCNN is achieving a better result than the other models. If we take macro f1 as a measure to compare the performance of CNN-SkipCNN with the best state of the art model presented in Table 4, the improvement is 27.96 in absolute points in 'Nepal Earthquake', 19.23 in absolute points in 'California Earthquake', 23.84 in absolute points in 'Cyclone PAM' but the only exception is for 'Typhoon Hagupit', where MLP-CNN_I performs better. Similarly, for accuracy

Table 3 The AUC scores of non-neural and neural network-based classifiers

EVENTS	AUC Score							
	Base CNN	CNN-GRU	CNN-SkipCNN	CNN_I	CNN_{II}	SVM	RF	LR
Nepal Earthquake	96.09	96.24	96.41	86.89	85.71	85.34	82.70	85.47
California Earthquake	96.41	97.22	97.76	81.21	78.82	78.95	75.64	79.57
Typhoon Hagupit	94.24	94.34	94.94	87.83	90.17	78.08	82.05	82.36
Cyclone PAM	99.06	99.26	99.38	94.17	93.11	90.82	90.26	90.64

Table 4 The accuracy and Macro f1 scores of all classifiers

Datasets	Measures	Base CNN	CNN-GRU	CNN_SkipCNN	CNN_I	MLP-CNN_I	SVM
Nepal Earthquake	Macro f1	81.51	84.12	84.96	57.00	57.00	48.00
	Accuracy	87.32	88.16	88.45	72.98	73.19	70.45
California Earthquake	Macro f1	81.96	88.60	89.23	70.00	70.00	65.00
	Accuracy	90.19	92.37	93.32	77.80	76.85	75.66
Typhoon Hagupit	Macro f1	65.28	73.41	74.00	76.00	77.00	70.00
	Accuracy	83.80	84.58	85.53	81.82	82.12	75.45
Cyclone PAM	Macro f1	87.80	91.75	92.84	67.00	69.00	65.00
	Accuracy	90.87	92.56	93.40	70.45	71.69	68.59

measures, both CNN-GRU and CNN-SkipCNN outperformed all other models and CNN-SkipCNN is the best model. It is clear from the results that, the CNN-SkipCNN model has an improvement of 15.26 in absolute points, 15.52 in absolute points, 3.41 in absolute points, and 21.71 in absolute points in 'Nepal Earthquake', 'California Earthquake', 'Typhoon Hagupit', and 'Cyclone PAM' databases respectively than the best state of the art model reported in Table 4.

So clearly CNN-SkipCNN performs better than other methods which suggest that skipCNN's may be a more effective feature extractor than GRU for both the tasks under consideration in very short texts such as tweets.

To further validate our results statistically, we have applied the Friedman test developed by Milton Friedman [15, 16], which is a popular non-parametric statistical test for the comparison of multiple classifier [12, 13]. The Friedman test is performed on AUC scores of Table 3 for binary classification task and on macro f1-score of Table 4 for the multi-class classification task. The formula used to do the Friedman test is given below.

$$F_{\mathrm{r}} = \left(\frac{12}{N * k * (k+1)}\right) * \left(\sum_{i=1}^{k} T_i^2\right) - 3 * N * (k+1) \tag{1}$$

N = Number of datasets
k = Number of classifiers
T_i = Sum of ranks for i^{th} dataset

For binary classification, we are comparing 8 classifiers on 4 datasets, the F_r value calculated as 26.0. The p-value is obtained as 0.000503. We have used the significance level (α) as 0.05. As the p-value is less than the significance level α, so the null hypothesis is rejected. Hence from the alternate hypothesis, it is observed that the results are significant.

Similarly, for the multi-class classification task, we are comparing 6 classifiers on 4 datasets. The F_r value is calculated as 11.857. The p-value is found as 0.0367. Here also we have chosen the significance level (α) as 0.05. As the p-value is less than the significance level α, so the null hypothesis is rejected and from the alternate hypothesis, it is observed that the results are significant.

Discussion During a time of crisis, lots of messages are posted online. Analysing these messages require real-time processing capabilities. Among these messages, some are related to crisis situations and some are not. First, we have filtered out those messages which are related to crisis situations by using a binary classifier and then use the crisis-related messages for humanitarian purposes.

Table 5 The class wise f1-score of CNN-SkipCNN classifier for each datasets

Class	EVENT			
	Nepal Earthquake	Typhoon Hagupit	California Earthquake	Cyclone PAM
Affected people	88.97	62.05	95.62	94.49
Donations and volunteering	86.99	74.04	89.23	95.14
Infrastructure and utilities	78.95	70.11	92.20	93.48
Sympathy and support	80.03	67.66	86.69	84.12
Other useful information	82.54	79.90	92.27	92.57
Not related or irrelevant	92.28	90.24	77.32	97.51

Humanitarian respondents need to identify which type of help people need from these messages. To identify different information types for humanitarian aid, we have presented the results of the multi-class classification problem. For this problem, we have used six different types of humanitarian assistance.

In Table 5 we have presented the class-wise f1-score for the CNN-SkipCNN model because from the result analysis in Section 4 it is clear that CNN-SkipCNN outperforms all other models. Figure 6 shows the class-wise distribution of the training data used in all these models. It seems from Table 5 that the Nepal earthquake, California earthquake, and Cyclone PAM datasets are easier to classify than Typhoon Hagupit because the f1-score is larger for former datasets than in Typhoon Hagupit across different classes. The f1-score for all classes of the Cyclone PAM dataset is greater than all other datasets, which means

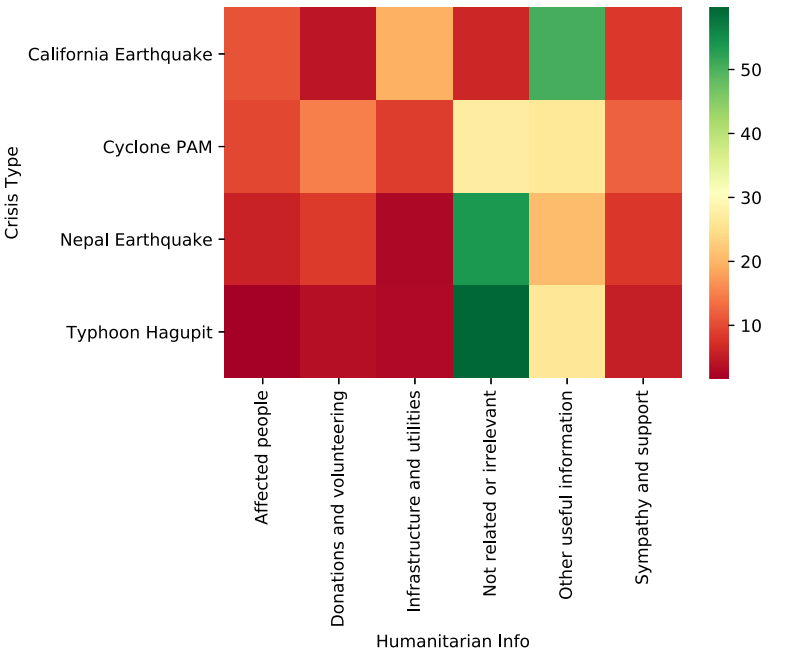

Fig. 6 Heatmap presents the distribution of training data by class for each event used to train the models

among all 4 datasets Cyclone PAM is the easiest dataset for the classification task. For Nepal Earthquake "Not related or irrelevant" class is easier to classify as it has the highest f1-score of 92.28(this class has more than 50% of tweets in this event), but the "Infrastructure and utilities" class is harder to classify as it has lowest f1-score of 78.95(this has less than 5% tweets in this event). For Typhoon Hagupit "Not related or irrelevant" class is easier to classify which has 90.24 as f1-score(more than 50% tweets for this event) and the "Affected people" class is the hardest to classify as it has the lowest f1-score of 62.05(less than 2% tweets for this event). Similar trends are also detected for the other two datasets. For California Earthquake "other useful information" class has the highest f1-score of 94.27 which has more than 50% tweets for that class and the "Not related and irrelevant" class has the lowest f1-score of 77.32 with less than 10% tweets. Cyclone PAM has the highest f1-score of 97.51 for the "Not related or irrelevant" class which has more than 25% of tweets for that class and has the lowest f1-score of 84.12 for the "Sympathy and support" class which has less than 15% tweets for that class.

Generally, the class with the smallest percentage of tweets has the lowest f1-score and seems harder to classify but it is not always true. For example in the Nepal Earthquake dataset, the "other useful information" class has more tweets(about 20%) than the "Affected people" class(about 5%) but the former class has less f1-score (82.54) than the latter class (88.97). We can also observe similar cases in other datasets. The only number of training data is not sufficient to determine that a class is easier or harder to classify. It also includes the inherent hardness related to a class. In this work, we did not consider the semantic relatedness of a tweet for the classification task.

7 Conclusion and future work

In this work, we proposed two deep neural network-based models based on CNN to solve two basic problems during a disaster situation. One is to identify tweets related to disaster and another one is to obtain fine-grained information from these tweets which may be used for different humanitarian activities. This work makes several contributions to this research area. Firstly, our proposed two DNN models can capture implicit features that are potentially useful for classification. Secondly, our models are thoroughly evaluated on 4 twitter datasets related to 4 disaster events and showed that our models outperformed the state of art neural and non-neural models. This proposed work has the following limitations that also point out the future directions of research. The first one is, it remains unknown that how well the model trained on one dataset performs on other disaster datasets and in addition to this, we also expect to obtain a more robust model that is trained across multiple disaster or crisis Tweets datasets. The second one is, we are interested in creating a multilingual disaster detector that can understand and process Tweets in different languages. The third one is we are planning to apply the SkippCNN approach in a semi-supervised setting to identify crisis related tweets along with retrieving information related to humanitarian aid.

References

1. Aivazoglou M, Roussos AO, Margaris D, Vassilakis C, Ioannidis S, Polakis J, Spiliotopoulos D (2020) A fine-grained social network recommender system. Soc Netw Anal Min 10(1):8
2. Alam F, Joty S, Imran M (2018) Graph based semi-supervised learning with convolution neural networks to classify crisis related tweets. arXiv:1805.06289

3. Andrews S, Gibson H, Domdouzis K et al (2016) Creating corroborated crisis reports from social media data through formal concept analysis. J Intell Inf Syst 47:287–312. https://doi.org/10.1007/s10844-016-0404-9
4. Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. J Mach Learn Res 3:1137–1155
5. Bhoi A, Pujari SP, Balabantaray RC (2020) A deep learning-based social media text analysis framework for disaster resource management. Soc Netw Anal Min 10(1):1–14
6. Burel G, Alani H (2018) Crisis event extraction service (crees)-automatic detection and classification of crisis-related content on social media
7. Burel G, Saif H, Alani H (2017) Semantic wide and deep learning for detecting crisis-information categories on social media. In: International semantic web conference. Springer, Cham, pp 138–155
8. Burel G, Saif H, Fernandez M, Alani H (2017) On semantics and deep learning for event detection in crisis situations
9. Caragea C, Silvescu A, Tapia AH (2016) Identifying informative messages in disaster events using convolutional neural networks. In: International conference on information systems for crisis response and management, pp 137–147
10. Cheng W, Sun Y, Li G, Jiang G, Liu H (2019) Jointly network: a network based on CNN and RBM for gesture recognition. Neural Comput and Applic 31(1):309–323
11. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. In: Deep learning and representation learning workshop at the 28th conference on neural information processing systems. Curran Associates, New York
12. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(12/1/2006):1–30
13. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18. ISSN 2210-6502, https://doi.org/10.1016/j.swevo.2011.02.002
14. Dhiman A, Toshniwal D (2020) An approximate model for event detection from Twitter data. In: IEEE Access, vol 8, pp 122168–122184. https://doi.org/10.1109/ACCESS.2020.3007004
15. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32:675–701
16. Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. Ann Math Stat 11:86–92
17. Gao H, Barbier G, Goolsby R (2011) Harnessing the crowdsourcing power of social media for disaster relief. IEEE Intell Syst 26(3):10–14
18. Imran M, Elbassuoni S, Castillo C, Diaz F, Meier P (2013) Extracting information nuggets from disaster-related messages in social media. In: Iscram
19. Imran M, Elbassuoni S, Castillo C, Diaz F, Meier P (2013) Practical extraction of disaster-relevant information from social media. In: Proceedings of the 22nd international conference on World Wide Web, pp 1021–1024
20. Imran M, Mitra P, Castillo C (2016) Twitter as a lifeline: Human-annotated twitter corpora for NLP of crisis-related messages. In: Proceedings of the Tenth international conference on language resources and evaluation (LREC)
21. Imran M, Mitra P, Castillo C (2016) Twitter as a lifeline: human-annotated twitter corpora for NLP of crisis-related messages. arXiv:1605.05894
22. Interdonato R, Guillaume JL, Doucet A (2019) Lightweight and multilingual framework for crisis information extraction from Twitter data. Soc Netw Anal Min 9:65. https://doi.org/10.1007/s13278-019-0608-4
23. Karimi S, Yin J, Paris C (2013) Classifying microblogs for disasters. In: Proceedings of the 18th Australasian document computing symposium, pp 26–33
24. Kaufhold M-A, Bayer M, Reuter C (2020) Rapid relevance classification of social media posts in disasters and emergencies: a system and evaluation featuring active, incremental and online learning. Information Processing and Management, 57, http://www.sciencedirect.com/science/article/pii/S0306457319303152
25. Kejriwal M, Zhou P (2020) On detecting urgency in short crisis messages using minimal supervision and transfer learning. Soc Netw Anal Min 10:58. https://doi.org/10.1007/s13278-020-00670-7
26. Kersten J, Kruspe A, Wiegmann M, Klan F (2019) Robust filtering of crisis-related tweets. In: ISCRAM 2019 Conference proceedings-16th international conference on information systems for crisis response and management
27. Khare P, Fernandez M, Alani H (2017) Statistical semantic classification of crisis information
28. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv:1408.5882

29. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: Proceedings of the 3rd international conference for learning representations
30. Kumar A, Garg G (2019) Sentiment analysis of multimodal twitter data. Multimed Tools Appl 78:24103–24119. https://doi.org/10.1007/s11042-019-7390-1
31. Kumar A, Sangwan SR, Nayyar A (2019) Rumour veracity detection on twitter using particle swarm optimized shallow classifiers. Multimed Tools Appl 78:24083–24101. https://doi.org/10.1007/s11042-019-7398-6
32. Lee H, Ahn Y, Lee H, Ha S, Lee S-g (2016) Quote recommendation in dialogue using deep neural network. In: Proceedings of the SIGIR, pp 957–960. https://doi.org/10.1145/2911451.2914734
33. Li H, Caragea D, Caragea C, Herndon N (2018) Disaster response aided by tweet classification with a domain adaptation approach. J Contingencies Crisis Manag 26(1):16–27
34. Madichetty S, Sridevi M (2019) Disaster damage assessment from the tweets using the combination of statistical features and informative words. Soc Netw Anal Min 9:42. https://doi.org/10.1007/s13278-019-0579-5
35. Mendon S, Dutta P, Behl A et al (2021) A hybrid approach of machine learning and lexicons to sentiment analysis. Enhanced Insights from Twitter Data of Natural Disasters, Inf Syst Front. https://doi.org/10.1007/s10796-021-10107-x
36. Nguyen DT, Al Mannai KA, Joty S, Sajjad H, Imran M, Mitra P (2016) Rapid classification of crisis-related data on social networks using convolutional neural networks. arXiv:1608.03902
37. Olteanu A, Vieweg S, Castillo C (2015) 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
38. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Empirical methods in natural language processing (EMNLP), pp 1532–1543
39. Power R, Robinson B, Colton J, Cameron M (2014) Emergency situation awareness Twitter case studies. Int. Conf. on Info. Systems for Crisis Response and Management in Mediterranean Countries (ISCRAM). Toulouse
40. Qu Y, Huang C, Zhang P, Zhang J (2011) 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
41. Resch B, Usländer F, Havas C (2018) Combining machine-learning topic models and spatiotemporal analysis of social media data for disaster footprint and damage assessment. Cartogr Geogr Inf Sci 45(4):362–376
42. Şahin C, Rokne J, Alhajj R (2019) Emergency detection and evacuation planning using social media. In: Social networks and surveillance for society. Springer, Cham, pp 149–164
43. Sailunaz K, Alhajj R (2019) Emotion and sentiment analysis from Twitter text. Journal of Computational Science, 36
44. Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web, pp 851–860
45. Schempp T, Zhang H, Schmidt A, Hong M, Akerkar R (2019) A framework to integrate social media and authoritative data for disaster relief detection and distribution optimization. Int J Disaster Risk Reduct 39:101143
46. Sinnappan S, Farrell C, Stewart E (2010) Priceless tweets! a study on Twitter messages posted during crisis: Black Saturday. ACIS 2010 Proceedings, p 39
47. Stowe K, Paul M, Palmer M, Palen L, Anderson KM (2016) Identifying and categorizing disaster-related tweets. In: Proceedings of The fourth international workshop on natural language processing for social media, pp 1–6
48. Sultana T, Badugu S (2020) A review on different question answering system approaches. In: Advances in decision sciences, image processing, security and computer vision, pp 579–586. Springer, Cham
49. Thomson R, Ito N, Suda H, Lin F, Liu Y, Hayasaka R, Isochi R, Wang Z (2012) Trusting tweets: the Fukushima disaster and information source credibility on twitter. In: Proceedings of the 9th international ISCRAM conference, pp 1–10
50. Verma S, Vieweg S, Corvey WJ, Palen L, Martin JH, Palmer M, Schram A, Anderson KM (2011) Natural language processing to the rescue? Extracting" situational awareness" tweets during mass emergency. In: ICWSM, pp 385–392
51. Vieweg S, Hughes AL, Starbird K, Palen L (2010) 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

52. Wang Y, Taylor JE (2018) Urban crisis detection technique: a spatial and data driven approach based on latent Dirichlet allocation (LDA) topic modeling. In: Proceedings of the 2018 construction research congress
53. Wang J, Xu W, Fu X, Xu G, Wu Y (2020) ASTRAL: adversarial trained LSTM-CNN for named entity recognition. Knowl-Based Syst, 105842
54. Xia K, Huang J, Wang H (2020) LSTM-CNN architecture for human activity recognition. IEEE Access 8:56855–56866
55. Zhang Z, Luo L (2019) Hate speech detection: a solved problem? The challenging case of long tail on twitter. Semantic Web 10.5:925–945
56. Zhang S, Vucetic S (2016) Semi-supervised discovery of informative tweets during the emerging disasters. arXiv:1610.03750

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Nayan Ranjan Paul[1] · Deepak Sahoo[2] · Rakesh Chandra Balabantaray[1]

Deepak Sahoo
deepsahoo@gmail.com

Rakesh Chandra Balabantaray
rakesh@iiit-bh.ac.in

[1] Department of Computer Science and Engineering, IIIT Bhubaneswar, Bhubaneswar, Odisha, India

[2] Department of Faculty of Emerging Technologies, Sri Sri University, Cuttack, Odisha, India