



A stacked convolutional neural network for detecting the resource tweets during a disaster

Sreenivasulu Madichetty¹ · Sridevi M.¹

Received: 12 October 2019 / Revised: 7 August 2020 / Accepted: 15 September 2020 /
Published online: 25 September 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Social media platform like Twitter is one of the primary sources for sharing real-time information at the time of events such as disasters, political events, etc. Detecting the resource tweets during a disaster is an essential task because tweets contain different types of information such as infrastructure damage, resources, opinions and sympathies of disaster events, etc. Tweets are posted related to Need and Availability of Resources (NAR) by humanitarian organizations and victims. Hence, reliable methodologies are required for detecting the NAR tweets during a disaster. The existing works don't focus well on NAR tweets detection and also had poor performance. Hence, this paper focus on detection of NAR tweets during a disaster. Existing works often use features and appropriate machine learning algorithms on several Natural Language Processing (NLP) tasks. Recently, there is a wide use of Convolutional Neural Networks (CNN) in text classification problems. However, it requires a large amount of manual labeled data. There is no such large labeled data is available for NAR tweets during a disaster. To overcome this problem, stacking of Convolutional Neural Networks with traditional feature based classifiers is proposed for detecting the NAR tweets. In our approach, we propose several informative features such as aid, need, food, packets, earthquake, etc. are used in the classifier and CNN. The learned features (output of CNN and classifier with informative features) are utilized in another classifier (meta-classifier) for detection of NAR tweets. The classifiers such as SVM, KNN, Decision tree, and Naive Bayes are used in the proposed model. From the experiments, we found that the usage of KNN (base classifier) and SVM (meta classifier) with the combination of CNN in the proposed model outperform the other algorithms. This paper uses 2015 and 2016 Nepal and Italy earthquake datasets for experimentation. The experimental results proved that the proposed model achieves the best accuracy compared to baseline methods.

Keywords NAR · CNN · Disaster · Stacking

✉ Sreenivasulu Madichetty
sreea568@gmail.com

Sridevi M.
msridevi@nitt.edu

¹ Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, India

1 Introduction

Micro-blogging [10, 14, 36, 40] sites like Twitter, Facebook, Instagram, etc. are helpful for collecting situational information [13] during a disaster like an earthquake, floods, disease outbreaks [25], etc. During these events, minor tweets are posted relevant to the specific classes such as infrastructure damage, resources [6, 33], service requests [24], etc., and also spam tweets, communal tweets and emotion information are posted [8, 16, 17, 19, 31, 38]. Therefore, it is required to design the powerful methodologies for the detection of specific class tweets (like Need, Availability of resources, etc.), so that relevant tweets can be automatically detected from the large set of tweets. The detection of specific class tweets [1, 11, 21, 35] has received much attention in the last two years. In the next few years, the detection of specific class tweets is likely to become more important in social media. Specifically, the detection of two types of tweets contains information related to Need and Availability of resources is a challenging task. During the disaster, victims post tweets with information such as where essential resources such as food, water, medical aid, shelter, etc. are needed or required. Similarly, humanitarian organizations post tweets with information such as where specific resources such as medical resources, food, water packets, etc., are available in the affected area. Examples of Need and Availability of Resource tweets are shown in Table 1. The first four tweets represent the need for resources such as mobile hospitals, password-free Wi-Fi, blood and ambulances. The next four tweets reflect the availability of information on resources such as the Italian Army to provide services to earthquake victims, the availability of shelter tents, money and ambulances. However, detection of Need and Availability of Resource tweets is very beneficial for both humanitarian organizations and victims during the disaster.

1.1 Objectives of this study

The main objective of this work is to assist the victims and humanitarian organizations in the event of a disaster by designing a method for automatic identification of Need and Availability of Resource tweets (NAR) from Twitter. The problem of detecting NAR tweets can be treated as a multi-classification problem. The classes are (i) Need of resource tweet (ii) Availability of resource tweet and (iii). None of both.

1.2 Prior work with limitations

Only a few existing works [1, 3, 11] are only focused on extracting the need and availability of resource tweets during the disaster. Among them, most of the works used information-retrieval methodologies such as word2vec, a combination of word embeddings and character embeddings, etc. Specifically, the authors in [3] used both information-retrieval methodologies and classification methodologies (CNN with crisis word embeddings) to extract the Need and Availability of Resource tweets during the disaster. The main drawback of CNN with crisis embeddings is that it does not work well if the number of training tweets is small and, in the case of information retrieval methodologies, keywords must be given manually to identify the need and availability of resource tweets during the disaster.

To overcome the above-mentioned issues, a novel method is proposed by using the stacking mechanism [44] to identify NAR tweets during the disaster. The stacking mechanism uses a two-level classifiers. The first level uses multiple classifiers and the classifier output is used as the second level classifier input, while the second level uses only one classifier.

Table 1 Examples of need and availability of resource tweet

Tweet no	Need of Resource tweets
1.	Italy Earthquake Emphasizes The Need For Mobile Hospitals [URL]
2.	New post: People urged to remove password from WiFi to help Italian earthquake relief [URL]
3.	#Terremoto Numeri Protezione Civile: 840840840/803555 Serve sangue.Civil Protection Nos. Urgent need for blood donation #Italy #earthquake”.
4.	Italy needs help! The earthquake left many people without home, hospitals need ambulances [URL]
	Availability of Resource tweets
5.	Death toll rises to 84 in Italy - The Italian army has been mobilized to aid earthquake-affected areas [URL].
6.	2 tent cities totaling about 50 tents being set up for quake victims in Italy’s central Marche region - ANSA
7.	I just donated some \$ to the Italian Red Cross Italy’s Earthquake Victims Need Aid Support[URL] via @bustle”.
8.	Search and rescue dogs
	20 ambulances on the ground in #Perugia following #earthquake volunteers from @crocerossa on the scene.

The stacking method does not produce improved results if the models used in the stacking method are stable. Therefore, different models such as CNN and KNN classifiers with domain-specific features are used in this work. CNN is used to capture the semantic similarity between words, and even vocabulary words are different in the testing phase. In order to overcome the problem of a lower number of training tweets, new features are proposed and used in the KNN classifier to detect NAR tweets. The two models (CNN and KNN classifiers with proposed features) have different functionality for the detection of tweets. The output of these two models is given as input to the SVM (second level) classifier. The SVM classifier is trained to determine the relationship between the output of the two CNN and KNN classifier models. It gives the final prediction of tweets whether a tweet label is a resource need or a resource availability or none. The efficacy of the final prediction depends on the classifiers used in level-1 and level-2. The reason for selecting the KNN and SVM classifiers as first and second level classifiers is clearly explained in Sections 4.4.2 and 4.5.2.

1.3 Contributions of this work

The main contributions are summarized as:

1. A Stacked Convolutional Neural Network is proposed to automatically identify the need and availability of resources tweets during the disaster.
2. Crisis word embedding is used in a deep learning model and domain-specific features are used in a feature-based classification method. Various classification algorithms such as SVM, Bagging, gradient boosting, random forest, KNN, Decision tree and Naive Bayes classification are also used.
3. Extensive experiments are carried out on real-time Twitter datasets such as the Nepal and Italy earthquakes in 2015 and 2016.
4. The proposed model is compared to the existing methodologies by using different parameters. In addition, statistical validation is performed to compare the methods using the MCNemar test.

This paper is organized as follows. The second section examines the related work. The proposed approach for the detection of NAR tweets during a disaster is described in the third section. Experimental results and analysis are discussed in the fourth section. The last section is the conclusion of the paper.

2 Related work

Many studies [22, 28, 32, 41] focused on the detection of the tweets related to a disaster. Preliminary work [41] focused mainly on extracting the features such as uni-gram and bi-gram frequency, Parts-Of-Speech (POS), objective or subjective, personal or impersonal and formal or informal from tweets and used the classifiers for classifying the tweets based on the relevancy. Classifiers such as Naive Bayes and Max entropy classifiers are used for detection of the situational tweets related to the disaster. The authors explained that their work depends on the vocabulary of a specific event. In [32], the authors investigated and developed an application for detecting the earthquake based on the features such as context words, keyword position, content words and length of the tweets. It is applicable only for Japanese tweets. To overcome the problem domain dependent, the authors in [28] proposed a novel framework for classifying the situational and non-situational information based on the low-level lexical and syntactical features. After classification, the tweets are summarized

based on the content words and also concluded that it works on cross-domain (domain independent). However, all the methods are focused only on situational tweets related to disaster but they failed to address specific class tweets.

In recent years, more researchers focused on the detection of user-defined class tweets during a disaster. Several studies, for instance [2, 11, 21, 29] have been proposed on different specific classes. The authors in [21], suggests that decision tree with context and content features give the best results for recall and F1-measure parameters among the classifiers such as SVM, AdaBoost and random forest. However, it does not focus on NAR tweets.

In recent literature, the authors in [35] developed a method by extracting the features by applying maximum frequency of words from the tweets to detect resource tweets during a disaster. Resources include both availability and need of the resources. However, it's not focused alone on the availability and need of the resources tweets during a disaster. The authors in [9] designed Artificial Intelligence Disaster Response (AIDR) system for classifying the tweets into user-defined categories for detecting the tweets related to the disaster. In AIDR, the uni-gram and bi-gram features are used for detecting the tweets related to the user-defined categories. These features are applied for detecting any user-defined classes during a disaster. In [2], the authors manually analyzed WhatsApp messages for the requirement of medical, human, infrastructural resources during a disaster by considering the case study of Nepal earthquake dataset 2015. However, they have not proposed an automatic method for identifying the resources. In [11], the authors found that neural network retrieved models by integrating the character-level and word-level embeddings with pattern recognition techniques perform well than state-of-art models. The authors applied information retrieval techniques for detecting the NAR tweets. In [7], the authors used a novel vector training approach for clustering the tweets about the emergency situations and compared their method with Bag-Of-Words (BOW), word2vec-sum and doc2vec. And described that clustering of tweets will be helpful further for identifying the different aspects of topic in emergency situations. However, they are not proposed a method for identifying the NAR tweets during a disaster.

3 Stacked convolutional neural network

The problem can be defined as follows: Given a 'N' number of tweets $X = \{x_1, x_2, x_3, x_4, \dots, x_N\}$, identify the tweets which are related to the three classes such as 1). Need of the resource 2). Availability of the resource and 3). None of the above. This section describes the stacked convolutional neural network for identifying the NAR tweets during a crisis. The overview of the proposed stacked convolutional neural network is shown in Fig. 1. The stacking mechanism [44] combines the predictions of diverse classifiers in the best way by learning the relationship between the models. Different classifiers vary in prediction errors from the data. For instance, some classifiers mispredict the data, while some other classifiers predict the same data correctly. It increases the generalization ability of the model and reduces the misclassification rate, bias and variance of the model. The stacking based classifiers give a high performance than the individual classifier models due to its generalization ability [42]. However, most of the resource detection systems focus on the individual classifier models rather than the ensemble methods (a combination of diverse classifiers). In this work, stacked convolutional neural network is proposed for detecting the resource tweets from social media during the disaster.

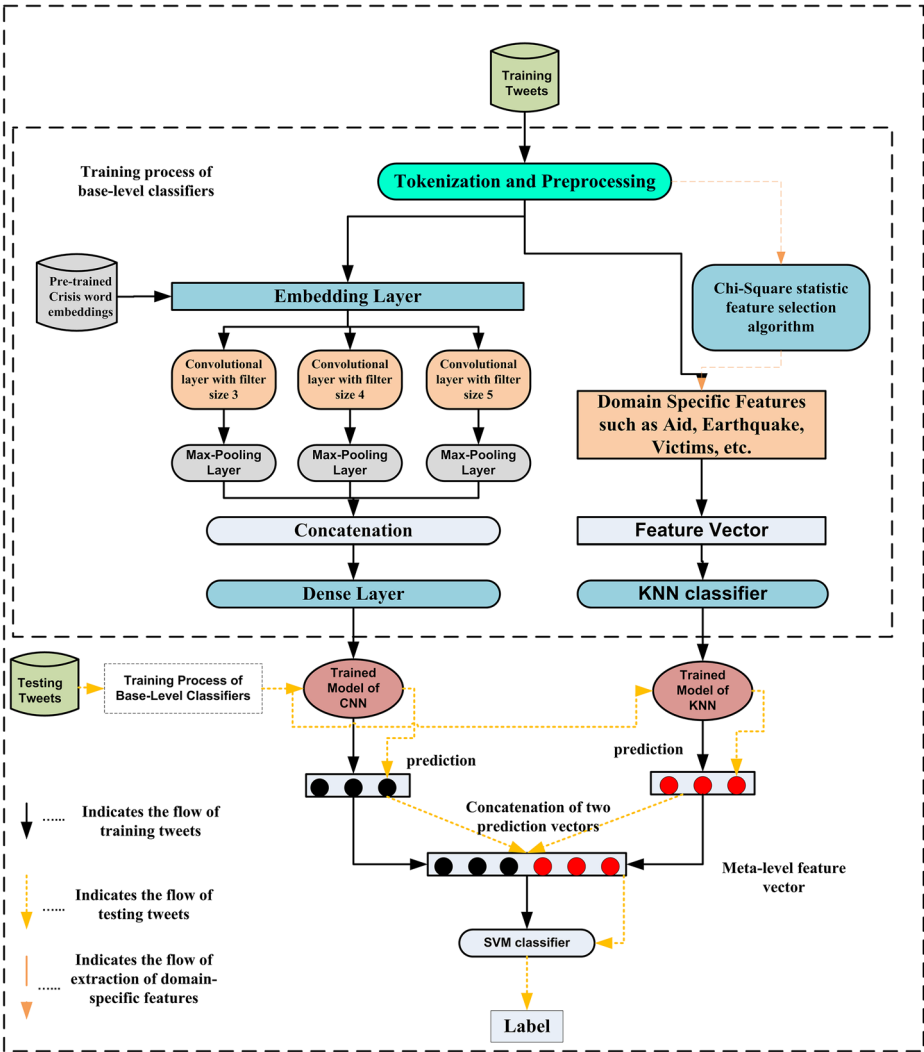


Fig. 1 The overview of the proposed stacked convolutional neural network

It consists of two phases of the classifier. In the first phase, the Convolutional Neural Network and the KNN classifiers are used and referred to as base-level classifiers. The SVM classifier is used as a meta-level classifier in the second phase. Before the tweets are given as inputs to the base-level classifiers, the following pre-processing and extraction steps are performed, such as:

3.1 Tokenization and pre-processing

- All tweets are changing to lower case letters to avoid the multiple copies of same words.
- These are divided into words and it referred as tokens
- The user mentions (@users), hash-tags (#) and URL’s are removed from the tweets.
- Similarly, stop-words, numerical and unknown symbols are omitted from tweets.

3.2 Feature extraction

For each tweet, two types of feature representation, and the following techniques are used to generate a feature representation from tweets, such as:

3.2.1 Word embeddings

We used pre-trained crisis word embeddings to represent the 300-dimensional vectors for each word in a tweet. It is mainly based on 52 million crisis-related tweets collected during 19 crisis events and used word2vec tool for training the word embeddings. It uses the Continuous Bag Of Words Model (CBOW) architecture with negative sampling to generate word embeddings.

3.2.2 Domain-specific features

χ^2 – static feature selection algorithm is used [45] to extract the top-most informative words from tweets because it has already been shown to be one of the most efficient feature selection algorithm for text categorization. The SVM classifier is used for the χ^2 – static feature selection algorithm because the authors in [20] concluded that the SVM with χ^2 statistic feature selection performed well than other traditional methods. The extracted domain-specific features are shown in Table 2. The first, second, and third columns are the serial number, features and information category, respectively.

3.3 Base-level classifiers

The above two methods provide two feature vector representations for each tweet that are given as input to base-level classifiers such as CNN and KNN Classifiers.

3.3.1 Convolutional neural network

CNN is suitable to elicit local and deep features from natural language. The authors [12] have shown that CNN has had better results in sentence classification. The authors in [34] have extended a convolutional-recursive deep model for 3D object classification that employs a combination of Convolutional and Recursive Neural Networks (RNN) cooperatively. The CNN layer discovers the low-level translation stable features that are feed into multiple, fixed-tree RNNs to formulate higher-order features. In [27], the authors have shown that CNN outperforms many traditional methods in biomedical text classification,

Table 2 Proposed domain-specific features for identifying the NAR tweets

S.No.	Features (Informative words)	Information category
1).	Earthquake	Calamity
2).	Food, Meals, Packets and Water	Nutritment
3).	Urgent and Urgently	Emergency
4).	Victims	Fatality
5).	Aid, Need, Needed, Needs Tents, Medical, Help and Shelter	Availability and Need
6).	Delhi and Temple	Spot

especially for assigning subjective medical headings to biomedical articles. CNN contains the following layers, such as the Embedding layer, Convolutional Layer, Pooling Layer and Dense layer.

Embedding Layer It is the very first layer of CNN. It takes a fixed number of words from the tweets as input and converts into a corresponding 300-dimensional crisis word vector. The 300-dimensional tweet vector is passed into a series of convolution and pooling operations to understand high-level feature representations.

Convolution and Pooling Layer In the convolution layer, the new features ‘ F ’ are generated by using convolution kernel ‘ $U \in R^{g \times d}$ ’ to a window of g words (filter size) as shown in (1).

$$(i.e) F_j = f(U \cdot x_{j:j+g-1} + b) \tag{1}$$

Where ‘ $x_{j:j+g-1}$ ’ is the concatenation of input vectors ‘ $(x_j, x_{j+1} \dots x_{j+g-1})$ ’, ‘ b ’ is a bias term and ‘ f ’ is a non-linear activation function like ‘sig’, ‘tanh’, etc. The filter is used to the window of ‘ g ’ words for getting the feature map with ‘ $F \in R^{n-g+1}$ ’, which is shown in (2). Different ‘ g ’ values (3,4,5) are used to capture the different n-gram features from the tweet.

$$F_i = f_1, f_2 \dots \dots \dots f_{n-h+1}. \tag{2}$$

This process is repeated for 100 times (100 filters) to produce the 100 feature maps to learn the complementary features of the same filter size. After getting the feature map, maximum pooling is applied to each feature map.

$$m = [\mu_q(F_1), \mu_q(F_2), \mu_q(F_3) \dots \dots \dots \mu_q(F_N)] \tag{3}$$

where ‘ $\mu_q(F_i)$ ’ refers to the maximum pooling operation [4] used to the each window of ‘ q ’ features in the feature map ‘ F_i ’. The output dimension is reduced by the max-pooling while keeping important features from each feature map.

After the maximum pooling operation, different feature vectors are generated from the convolution layer with filter sizes (3, 4, 5). Then, the concatenation operation is applied to the different feature vectors to become a single block.

Dense layer The dense layer with the softmax activation function is used on the top of the pooling layer to keep the features generated from the pooling layer. It is shown in the (4).

$$z = f(Wm + b_e) \tag{4}$$

Where ‘ W ’ is a weight matrix, ‘ b_e ’ is a bias vector and ‘ e ’ is a non-linear activation function. The input of dense layer may be variable length, which produces fixed output ‘ z ’, and it is given as input for classification.

The output layer defines the probability distribution and uses a softmax function. The probability of the ‘ t ’ label output is given by (5).

$$P(y = t / TW, \theta) = \frac{\exp(W_t^T z + b_t)}{\sum_{i=1}^t \exp(W_i^T z + b_i)} \tag{5}$$

Where ‘ W_t ’ is the weights associated with the class ‘ t ’ labels in the output layer.

3.4 KNN classifier

We adopted the K-Nearest Neighbour as a base-level classifier in the proposed model to get the feature vector of the tweet to the meta-level (second-level) classifier. It acts as a first-level classifier for getting better performance than other classifiers (Decision tree, Naive Bayes classifier), and a detailed explanation is shown in Sections 4.4 and 4.5.2. It accepts domain-specific features such as aid, needs, etc., as an input feature vector of the tweets. The KNN classifier gives the scores to the tweet neighbors among the training tweets and uses the class labels of ‘k’ most similarity neighbors to predict the probability vector of the tweet. We use the Euclidean distance ‘ $E(Tw, Tw^1)$ ’ to measure the similarity between the tweets ‘ Tw ’ and ‘ Tw^1 ’ that is shown in (6)

$$E(Tw, Tw^1) = \sqrt{\sum_i^N (Tw_i - Tw_i^1)^2} \quad (6)$$

Where ‘N’ is dimension size of the tweet vectors ‘ Tw ’ and ‘ Tw^1 ’. The classes of these neighbors are weighted using the similarity of each neighbor to Tw_0 as follows:

$$score(Tw_0, C_i) = \sum_{Tw_j \in KNN(Tw_0)} Sim(Tw_0, Tw_j) \delta(Tw_j, C_i) \quad (7)$$

where ‘ $KNN(Tw)$ ’ indicates the set of K-nearest neighbors of tweet Tw . $\delta(Tw_j, C_i)$ represents the probability of Tw_j with respect to the class C_i and $i=3$ represents the number of classes are three such as Need of resource, Availability of resource and None of the both.

Finally, it produces the three-dimensional probability vector for each tweet in testing data. Results indicate that the KNN classifier also plays a significant role in the proposed model for detecting the NAR tweets.

3.5 Meta-level classifier

In this work, we have adopted the SVM classifier [39] and it is one of the traditional machine learning algorithms in the proposed model. SVM is used as a meta-level classifier for getting better performance than other classifiers (Decision tree, Naive Bayes classifier) and a detailed explanation is shown in Sections 4.4 and 4.5.2. It accepts the concatenation of the predicted outputs of the CNN and KNN classifiers as input features. The size of the input vector is six-dimensional. We used the Radial Basis Function (RBF) kernel in the SVM classifier for transforming the data into a higher dimensional feature space. Given a set of testing tweets to the base-level classifiers and it produces the output of six-dimensional vectors. The results are sent as input features to the meta-level classifier (SVM classifier). The output of the SVM (second level classifier) is used as a final tweet prediction. Later, the learned model will be used to detect NAR tweets during a disaster.

The main advantage of the proposed stacked convolutional neural network for detecting NAR tweets during a disaster is that it works effectively, even for small datasets, due to the use of domain-specific features. And also, even though the words are different in both training and testing tweets using the CNN model. The summarization of the proposed method is shown in algorithm 1.

Algorithm 1 The summarization of the proposed method.

Input:

CNN and KNN with proposed features

1: It represents tweet related to the availability of resources

0: It represents tweet non-related to the need and availability resources

2: It represents tweet related to the need resources

Steps:

1. The tweets are preprocessed by applying the following techniques.
 - Removal of stop-words, numerical and unknown symbols.
 - Changing to lower case letters.
2. Word vectors are generated for all tweet words using word embeddings and pass through the CNN.
3. Proposed features such as terms related to Calamity, Nutriment, etc., and are selected by using χ^2 - statistic feature selection algorithm and features are sent to the KNN.
4. The output of the CNN and KNN is concatenated and considered as features.
5. The features are sent to the second level (SVM) classifier.
6. The output of the classifier (SVM) is predicted as a class label.

Output:

$Y_D = \{0, 1, 2\}$

4 Experimental results and analysis

In this section, we first introduce the datasets, parameters details of the model and metrics used for performance evaluation. Subsequently, the experimental results include the results of the preliminary experiments, the classifier selection experiments in the proposed model and the ablation experiments. Furthermore, a comparison is made between the proposed approach and existing approaches.

4.1 Datasets

The data are collected from Nepal and Italy earthquakes that occurred during 2015 and 2016, respectively. Tweets are crawled from the tweet-id's through the Twitter API the tweet-id's are obtained from the authors [11]. Out of the total tweets, 80% and 20% of tweets are used for training and testing the proposed model, respectively. The details of disaster datasets are given in Table 3. The code is made available to the public ¹.

4.2 Parameter details

Training the CNN model by optimizing the sparse-cross entropy of (5) using the ADADELTA [46] algorithm. The maximum epoch number is set at 50. The mini-batch

¹<https://github.com/sreenivasuluMadichetty/StackedConv>

Table 3 Details of Nepal and Italy earthquake datasets

Dataset	Total no. of tweets	Distinct tweets	Availability tweets	Need tweets
Nepal Earthquake	100k	50,018	1,333	495
Italy Earthquake	180k	70,487	233	177

sizes of 32, 64, 128 are used. The mini-batch size is 64, which gives better results compared to other batch sizes and is tabulated in Table 6 and filter sizes of 3, 4, 5 are used. To avoid the over-fitting, 0.5 dropout [37] and early stopping criteria based on the loss of the validation data are used. All the experiments are performed using the python language scikit [23] package. Table 4 gives the inscription of the various methods. The first column, second column and third column indicate the serial number, method name and abbreviation, respectively. In the abbreviation, the methods before and after ‘+’ symbol are the base-level classifiers (first level classifiers), ‘+’ indicates the concatenation of predicted output of the base-level classifiers (first level classifiers) and ‘→’ symbol indicates the flow of predicted output of the base-level classifiers as input to the meta-classifier. The method after ‘→’ symbol indicates the meta-level classifier (second level classifier).

4.3 Metrics for performance evaluation

The performance of the proposed models is assessed based on the standard measures such as accuracy, precision, recall and f1-score are calculated using Eqs. 8 to 11, respectively.

Table 4 Inscriptions

S.No	Methods	Abbreviations
1.	CSS	CNN+SVM → SVM
2.	CSK	CNN+SVM → KNN
3.	CSNB	CNN+SVM → Naive Bayes
4.	CSD	CNN+SVM → Decision tree
5.	CDS	CNN+Decision tree → SVM
6.	CDK	CNN+Decision tree→ KNN
7.	CDN	CNN+Decision tree → Naive Bayes
8.	CDD	CNN+Decision tree → Decision tree
9.	CNBS	CNN+ Naive bayes classifier → SVM
10.	CNBK	CNN+ Naive bayes → KNN
11.	CNBNB	CNN+ Naive bayes → Naive Bayes
12.	CNBD	CNN+ Naive bayes → Decision tree
13.	CKS	CNN+ KNN → SVM
14.	CKK	CNN+ KNN→ KNN
15.	CKNB	CNN+KNN → Naive Bayes
16.	CKD	CNN+ KNN→ Decision tree
17.	SPROP	SVM with proposed features
18.	Baseline	SVM with BoW model

$$Accuracy = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \quad (8)$$

$$Precision = \frac{TP_i}{TP_i + FP_i} \quad (9)$$

$$Recall = \frac{TP_i}{TP_i + FN_i} \quad (10)$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (11)$$

where TP_i = Total No. of positive tweets detected correctly as positive.

TN_i = Total No. of negative tweets detected correctly as negative.

FP_i = Total No. of negative tweets wrongly detected as positive.

FN_i = Total No. of positive tweets wrongly detected as negative.

i = No. of classes.

The accuracy of CNN is shown in Table 6 for various batch sizes. However, the batch size of 64 got the best accuracy compared to the batch sizes of 32 and 128. Therefore, for further experiments batch size of CNN, 64 is considered.

4.4 Experimental results

This section explains the results of the preliminary experiments, the classifier selection experiments in the proposed model, and the ablation experiments.

4.4.1 Preliminary experiments

Initially, the experiment is performed on the SVM classifier based on the proposed domain-specific features for the identification of NAR tweets and compared to the BoW model shown in Table 5. It highlighted the impact of the proposed domain-specific features compared with the BoW model for the proposed solution. It is beneficial for the proposed solution to identify tweets, especially for smaller datasets. Later, various experiments are performed using the CNN model to determine the best batch size. The batch sizes such as 16, 32 and 64 are used. Results of the CNN model using the accuracy parameter is shown in Table 6 by varying the batch sizes. The results show that the CNN model provides the best outcome for the batch size of 64 compared to others, such as 32 and 128. Therefore, for additional experiments, 64 batch size is considered. It is noted that the values reported in all tables are based on the average Need and Availability of resource classes.

4.4.2 Classifier selection in the proposed method

The following four different experiments are performed for the proposed method to choose the best appropriate classifier for base-level and meta-level classifiers.

Table 5 Comparison of SPROP with baseline model

Dataset	Precision		F1-Score		Accuracy	
	Baseline	SPROP	Baseline	SPROP	Baseline	SPROP
Nepal Earthquake	26.5	71.5	34.5	63.5	59.5	67.3

Table 6 Accuracy of CNN by varying the batch sizes

S.No	Mini-batch size	Accuracy
1.	32	75.89
2.	64	76.77
3.	128	68.79

1. In the first experiment, the output of CNN and SVM (base-level classifiers) are given as features to the meta-level classifier. By varying the meta-level classifiers (SVM, KNN, Decision tree and Naive Bayes), the results are reported in Table 7. KNN gives the best performance than other classifiers for the Nepal earthquake dataset. But in the case of the Italy earthquake dataset, SVM gives the best performance than the other classifiers.
2. In the second experiment, the CNN output and the decision tree (base-level classifiers) are given as features to the meta-level classifier. The models used in the second experiment by different meta-level classifiers are CDS, CDK, CDN and CDD, and the results are reported in Table 8. Among the other models, CDK gives the best accuracy for the Nepal earthquake dataset and Italy earthquake dataset. CDN also provides the same accuracy as CDK in the case of the Italy Earthquake dataset.
3. In the third experiment, the output of the CNN and Naive Bayes classifiers (base-level classifiers) is given as a feature to the meta-level classifier. The models used in the third experiment to vary the meta-level classifiers are CNBS, CNBK, CNBNB and CNBD, and the results are reported in Table 9. CNBNB has the best accuracy among the models for both disaster datasets. CNBS gives the same accuracy as the CNBNB in the case of the Italy earthquake dataset.
4. Finally, in the fourth experiment, the output of the CNN and KNN classifiers (base-level classifiers) is given as input to the meta- classifier. The models used in the fourth experiment to vary the meta- classifiers are CKS, CKK, CKNB and CKD, and the results are tabulated in Table 10. CKS achieves the highest accuracy among the models for both disaster models.

Table 7 Comparison of proposed models (CNN+SVM → classifier) on Nepal and Italy earthquake datasets

Dataset	Model	Precision	F1-score	Accuracy	Execution time (ms)
Nepal Earthquake	CSS	73.50	75.50	77.48	1.84
	CSK	73.50	76.00	77.65	1.91
	CSNB	76.00	74.00	77.30	1.86
	CSD	72.00	70.50	73.40	1.79
Italy Earthquake	CSS	77.50	74.00	74.34	1.77
	CSK	73.00	70.00	72.57	2.39
	CSNB	79.50	71.50	73.45	1.59
	CSD	66.50	66.50	69.91	1.59
Average of Nepal and Italy Earthquake	CSS	75.50	74.75	75.91	1.80
	CSK	73.25	73.00	75.11	2.15
	CSNB	77.75	72.75	75.37	1.73
	CSD	69.25	68.50	71.65	1.69

Table 8 Comparison of proposed models (CNN+ Decision tree → classifier) on Nepal and Italy earthquake datasets

Dataset	Model	Precision	F1-score	Accuracy	Execution time (ms)
Nepal Earthquake	CDS	72.50	74.00	76.50	1.72
	CDK	72.50	74.50	76.70	1.81
	CDNB	73.00	72.50	76.40	1.83
	CDD	72.50	73.00	74.80	1.84
Italy Earthquake	CDS	80.00	73.50	73.45	1.59
	CDK	77.50	74.00	75.22	2.04
	CDNB	79.50	72.00	75.22	1.59
	CDD	68.50	68.00	69.03	2.04
Average of Nepal and Italy Earthquake	CDS	76.25	73.75	74.97	1.65
	CDK	75.00	74.25	75.96	1.93
	CDNB	76.25	72.25	75.81	1.71
	CDD	70.50	70.50	71.91	1.94

After performing four different experiments, the best f1-score models (models that achieve the best f1-score) are selected from the four various experiments of models such as CDK, CKS / CKK, CNBS and CSK for both disaster datasets. In the same way, the best precision models (models that achieve the highest precision) such as CKNB, CDNB, CNBB / CNBD and CSNB on the Nepal earthquake dataset are selected. Similarly, CSNB, CDS, CNBNB and CKS models achieve the best precision for the Italy earthquake dataset. In the case of the execution time, CDS runs very fastly on average of both disaster datasets. However, it does not give the best results compare to other models.

Finally, all models are compared and selected as the CSK model that achieves the best f1-score for the Nepal earthquake dataset. In the case of an accuracy parameter, the CSK model gives the best performance for the Nepal earthquake dataset but not provide for the

Table 9 Comparison of proposed models (CNN+ Naive Bayes classifier → classifier) with variations on Nepal and Italy earthquake datasets

Dataset	Model	Precision	F1-score	Accuracy	Execution time (ms)
Nepal Earthquake	CNBS	72.00	74.00	76.50	1.75
	CNBK	71.50	73.50	76.20	1.73
	CNBNB	74.50	72.50	76.95	1.80
	CNBD	74.50	71.00	74.60	1.77
Italy Earthquake	CNBS	73.00	71.00	72.57	1.68
	CNBK	73.00	66.50	70.80	2.12
	CNBNB	79.50	69.50	72.57	2.12
	CNBD	70.50	67.50	71.68	1.95
Average of Nepal and Italy Earthquake	CNBS	72.50	72.50	74.53	1.71
	CNBK	72.25	70.00	73.50	1.92
	CNBNB	77.00	71.00	74.76	1.96
	CNBD	72.50	69.25	73.14	1.86

Table 10 Comparison of proposed models (CNN+ KNN → classifier) with variations on Nepal and Italy earthquake datasets

Dataset	Model	Precision	F1-score	Accuracy	Execution time (sec)
Nepal Earthquake	CKS	73.50	75.50	77.50	1.79
	CKK	74.00	75.50	76.95	1.79
	CKNB	77.00	72.50	76.77	1.80
	CKD	72.50	71.00	73.93	1.95
Italy Earthquake	CKS	82.50	76.50	76.99	1.68
	CKK	76.50	73.00	75.22	1.68
	CKNB	80.50	72.00	75.22	1.76
	CKD	69.00	69.00	69.91	2.47
Average of Nepal and Italy Earthquake	CKS	78.00	76.00	77.25	1.73
	CKK	75.25	74.25	76.09	1.73
	CKNB	78.75	72.25	75.99	1.78
	CKD	70.75	70.00	71.92	2.21

Italy earthquake dataset. Overall comparison of all the models, CKS performs well than the other models on both disaster datasets. Therefore, CKS is selected to identify NAR tweets during the disaster.

4.4.3 Ablation experiments

Various experiments are conducted to assess the effectiveness of the individual component in the proposed model (CKS) on two datasets, such as Nepal and Italy earthquake. The proposed model is initially evaluated and the results for two datasets are tabulated in Table 11. Later, the experiments are performed by excluding informative (domain-specific) features and CNN individually in the proposed model and the results are reported in Table 11. The informative features play a crucial role in the proposed method for Italy's earthquake dataset, which reduces the performance of the proposed model by almost 5.31% accuracy. In the case of the Nepal Earthquake, the performance is reduced by approximately 0.90% accuracy. By removing the CNN model, the performance of both datasets is drastically reduced by almost 25% and 15% for the Nepal and Italy earthquake datasets, respectively. It indicates that CNN plays a significant role in both disaster datasets. By removing both CNN and SVM classifiers from the proposed model, the performance reduction is the same as when CNN is removed. It indicates that the SVM classifier alone

Table 11 Accuracy of ablation experiments on Nepal and Italy earthquakes

Classifier	Nepal Earthquake (Execution time)	Italy Earthquake (Execution time)
Proposed method (CKS)	77.50 (1.79ms)	76.99 (1.64ms)
CNN→SVM	76.60 (1.76ms)	71.68 (1.54ms)
KNN→SVM	50.89 (1.78ms)	56.64 (1.60ms)
KNN	50.88 (0.47ms)	56.63(0.294ms)
CNN	76.77 (1.75ms)	70.79 (1.57ms)

does not have much impact on the performance of the model. However, the proposed method (CKS) provides the best accuracy than any of the components used to identify NAR tweets during the disaster. It is also proved by using statistical validation and it is given in Section 4.5.2.

4.5 Comparison of the proposed approach with the existing approaches

This section provides a brief explanation of the methods that are compared with the proposed model. It can be categorized into two subsections based on the methods. 1. Classification Methodologies. 2. Statistical validation of the classifier models.

4.5.1 Classification methodologies

This section describes the comparison of the proposed model with the existing classification methodologies [9, 12, 30, 35]. In [9], the authors presented an AIDR platform for automatic classification of tweets into user-defined categories with the use of uni-gram and bi-gram features. Similarly, in this paper, the SVM classifier with features such as uni-gram and bi-gram used as a baseline, and experiments are performed. In [35], the authors used features such as location, infrastructure damage, communication, etc., for identifying the resources during a disaster and SVM classifier is used for classification. The authors [12] used CNN for sentence classification by hyper-tuning the parameters. Similar to this, CNN is experimented and compared with the proposed model. In [30], the authors used the low-level lexical and syntactical features for identifying the situational information during a disaster. The proposed CKS model achieves the best accuracy compared to existing methods on the Nepal and Italy earthquake dataset and the results are reported in Table 12. However, the proposed model outperforms existing methods on both Nepal and Italy earthquake datasets for identifying the NAR tweets. Better accuracy is achieved for the proposed model when compared to the existing method due to the use of informative features and traditional classifiers, which enhanced the diversity of the model for identifying the NAR tweets. In general, stacking models give better accuracy than individual models when the models have diversity. And also, it is observed that from Table 12, for Italy earthquake dataset has a huge impact on the proposed method compared to the Nepal earthquake dataset due to the small dataset. In case of the execution time, Rudra model [30] runs very fastly and BoW model [9] runs very slowly compared to other models. However, it does not give the best result for detecting the NAR tweets during the disaster.

Table 12 Comparison of proposed model with existing methods on average of Nepal and Italy earthquake datasets using Accuracy parameter

Models	Nepal Earthquake (Execution time in ms)	Italy Earthquake (Execution time in ms)	Average of both Nepal and Italy Earthquake	Year
Baseline [9]	59.50 (13.67)	67.20 (1.23)	63.35	2014
Kims et al. [12]	76.77 (1.75)	70.79 (1.59)	73.88	2014
Koustav et al. [30]	47.10 (0.072)	48.83 (0.007)	47.96	2018
Sreenivasulu M et al. [35]	53.36 (0.40)	67.25 (0.35)	60.30	2017
Proposed method	77.12 (1.91)	76.99 (1.68)	77.05	Present

4.5.2 Statistical Validation for comparison of the various classifier models

In this section, we have investigated the statistical significance of the different classification models. The authors in [5] suggest that the use of the MCNemar statistical test for the deep learning models. Therefore, we have used the MCNemar statistical methods [5] to study the efficacy of statistical significance for classification methods. The contingency table of the MCNemar test is shown in Table 13.

Here ' N_{01} ' represents the number of tweets corrected detected by Model A and Model B. ' N_{02} ' represents the number of tweets corrected detected by Model B and wrongly detected by Model A. ' N_{11} ' represents the number of tweets corrected detected by Model A and wrongly detected by Model B. ' N_{12} ' represents the number of tweets wrongly detected by Model A and Model B

The chi-squared (χ^2) can be defined as follows:

$$\chi^2 = \frac{(|N_{02} - N_{11}| - 1)^2}{N_{02} + N_{11}} \quad (12)$$

The hypothesis is:

1. Null hypothesis (N0): There exists no significant difference between the performances of the classifier model.
2. Alternate hypothesis (N1): It can be defined as the existence of a significant difference between the performances of the classifier model.

If N0 is accepted, then the probability (p) value is greater than 0.05. If N1 is accepted, then the probability (p) value is less than 0.05.

Tables 14 and 15 show the results of the MCNemar statistical test of the performance of the various proposed methods and the comparison with the existing methods. In tables, the ' $\uparrow\uparrow$ ' indicates that the strong evidence of the proposed method is statistically significant compared to the other method and that the probability value is less than 0.01 ($p < 0.01$). It represents the confidence level of 99.99% of the proposed method. ' \uparrow ' indicates that the weak evidence of the proposed method is statistically significant compared to the other method and the probability value is between 0.01 and 0.05 ($0.01 < p < 0.05$). ' \sim ' indicates that there is no statistical significance between the two methods of the same classification performance. Subsequently, the methods in the first column of the Tables 14 and 15 are statistically significant compared to the other methods in the row. From Table 14, we can describe it in the following ways:

1. There is strong evidence that the CSS is statistically significant compared to other methods such as CSD and failed to reject the N1 hypothesis. It is the weak evidence that the CSS is statistically significant than the CSK and was unable to accept the N0 hypothesis. There is no significant difference between the CSS and CSNB, and failed to reject the N0 hypothesis.

Table 13 Contingency table

Number of categories	Correct by model A	Incorrect by model A
Correct by Model B	N_{01}	N_{02}
Incorrect by Model B	N_{11}	N_{12}

Table 14 Results of the McNemar-tests on variants of the proposed methods for both Nepal and Italy Earthquake datasets

Methods	Method-1	Method-2	Method-3
CSS	CSK (↑)	CSNB(∼)	CSD (↑↑)
CDK	CDS (↑)	CDNB (∼)	CDD (↑↑)
CNBNB	CNBS (∼)	CNBK (↑↑)	CNBD (↑↑)
CKS	CKK (↑↑)	CKNB (↑↑)	CKD (↑↑)
CKS	CSS (↑↑)	CDK (↑↑)	CNBNB (↑↑)

2. There is strong evidence that the CDK is statistically significant as the CDD method and failed to reject the N1 hypothesis. It is weak evidence that the CDK is statistically significant than the CSK and failed to accept the N0 hypothesis. However, there is no statistically significant difference between the CDK and the CDNB, and accept the N0 hypothesis.
3. There is no statistically significant difference between the CNBNB and CNBS, and accept the N0 hypothesis. But there is strong evidence that CNBNB is statistically significant than the CNBK and CNBD, and failed to reject the N1 hypothesis.
4. It is strong evidence that the CKS is statistically significant than the other methods such as CKK, CKNB and CKD, and accept the N1 hypothesis.
5. Finally, we have shown strong evidence that CKS is statistically significant than the CNBNB, CDK and CSS, and accepts the N1 hypothesis.

Similarly, from Table 15, we can explain as follows:

1. The first row shows a comparison of the ablation experiments, and the second row represents a comparison of the proposed method with the existing methods.
2. The results show that the strong evidence of the proposed method is statistically significant compared to the existing methods and leads to significant improvement by adding the proposed features to the model. And it accepts the N1 hypothesis.

4.6 Discussion

This paper proposes a method named as CKS (CNN and KNN are used as base-level classifiers, and SVM is used as a Meta-level classifier) for identifying tweets related to the Need and Availability of Resources during the disaster. It intends to assist victims and humanitarian organizations for identifying where the resources are available and where the resources are needed or required using social media in the event of a disaster. It also helps service providers to collect the necessary resources, transport, etc., to provide the victims with the resources that they need. For example, we can automatically make a different mark on the map to help local volunteers and victims that specifies where a large number of resources are needed or available. The performance of the proposed method has been demonstrated in both the Nepal and Italy Earthquake datasets. The research implication to the COVID-19 is discussed in the subsequent section.

4.6.1 Research implication to COVID-19

Our study has a practical implication to COVID-19 for resource management. Developing countries like India are suffering from financial resources, particularly when the government

Table 15 Results of the McNemar-tests for ablation methods and the existing methods for both Nepal and Italy Earthquake datasets

Proposed methods	Method-1	Method-2	Method-3	Method-4
CKS	CNN → SVM (↑↑)	KNN → SVM (↑↑)	KNN (↑↑)	CNN (↑↑)
Proposed Method (CKS)	Baseline [9] (↑↑)	Kims et al. [12] (↑↑)	Koustav et al. [30] (↑↑)	Sreenivasulu M et al. [35] (↑↑)

has no choice but to close the business and lock it up during COVID-19. These can affect their daily consumption of food, lack of nutrition, medical resources (such as ventilators), masks and other urgent needs. These types of resources are automatically identified by using social media. Some organizations post this type of resource information on social media, where resources are available. In the same way, victims post information where resources are needed. However, the proposed model can be used for the identification of these types of resource tweets during COVID-19. It may help both the people and humanitarian or government organizations to save the time and effective utilization of the resources.

There is a chance of increasing the number of COVID cases in the future. It may have an impact on the lack of medical resources, such as ventilators, hospital services, victim quarantine, masks, etc. Identifying this type of resource where it is available and needed is very important for the effective use of resources. This allows people to save their lives and avoid the spread of the disease from one person to another.

In the future, work can be extended to extract a specific type of resource information where it is required and available along with a priority-based geo-location of tweets. For example, the highest priority is given for the tweets where it contain information related to very urgent needs such as ventilators and masks, etc., and have geo-location information (where resource needs or availability exist). If tweets contain information on resources such as the need or availability of food, donation of money or services is the next priority. If the tweets include some other type of resource information without Geo-location shall be given as a minimum priority. And also, the automatic matching of the Needs and Availability of Resource tweets during COVID-19.

The pre-requirements of the model to be working in the real-world for deployment can be described is as follows:

1. GPU server.
2. Finding the relevant hash-tags and keywords.
3. Filtering the Fake and Spam tweets.

GPU server GPU Server is needed to store and processing the tweets from the twitter during COVID-19.

Finding the relevant hashtags and keywords Finding the relevant hashtags and keywords is one of the important modules for deploying the model in the real-world. The relevant hashtags and keywords can be used for filtering the tweets related to the COVID-19. During the COVID-19, users post millions of tweets on twitter by using different hashtags and keywords like #Covid-19, #CORONA, COVID, etc. Therefore, finding the relevant hashtags is an important task during the COVID-19. Various methods [15, 18, 43] are available in the existing literature for finding the relevant hashtags to the COVID-19. Most of the methods, there is a need to give some seed keywords manually for finding the relevant hashtags.

Filtering the Fake and Spam tweets After extracting the tweets related to the COVID-19, there is a need to filter the Fake and Spam tweets from the extracted relevant tweets to the COVID-19. Fake tweets [26] can be defined as if it contains the incorrect time or location related to the need and availability of resources or link to the misleading information, etc., is called fake tweets. Spam tweets [8, 19, 26] can be defined as if it contains links to the advertisements or loans or some other irrelevant content, etc., is called spam tweet.

After removing the fake and spam tweets from the relevant tweets, the resultant tweets are passed to the model for identifying the resource tweets during the disaster.

5 Conclusion

Detection of the NAR tweets during a disaster is a difficult task due to different kinds of tweets are posted related to the disaster. A model is proposed for identifying the NAR tweets during a disaster. The results suggest the idea that the stacking of a convolutional neural network with traditional feature-based classifiers is useful for detecting the NAR tweets. The results also suggest that the combination of CNN, KNN and SVM (CKS) with domain-specific features outperform the other combinations. Also, the proposed method outperforms the existing methods on Nepal and Italy earthquake datasets. Furthermore, we discussed the application of the proposed method for real-time scenarios like the COVID-19 outbreak. In future work, the accuracy of the model is improved by using other deep learning models to detect NAR tweets during a disaster.

References

1. Basu M, Ghosh K, Das S, Dey R, Bandyopadhyay S, Ghosh S (2017) Identifying post-disaster resource needs and availabilities from microblogs. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017. ACM, pp 427–430
2. Basu M, Ghosh S, Jana A, Bandyopadhyay S, Singh R (2017) Resource mapping during a natural disaster: A case study on the 2015 Nepal earthquake. *International Journal of Disaster Risk Reduction*
3. Basu M, Shandilya A, Khosla P, Ghosh K, Ghosh S (2019) Extracting resource needs and availabilities from microblogs for aiding post-disaster relief operations. *IEEE Trans Comput Soc Syst* 6(3):604–618
4. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. *J Mach Learn Res* 12(Aug):2493–2537
5. Dietterich TG (1998) Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput* 10(7):1895–1923
6. Dutt R, Basu M, Ghosh K, Ghosh S (2019) Utilizing microblogs for assisting post-disaster relief operations via matching resource needs and availabilities. *Inf Process Manag* 56(5):1680–1697
7. Ganguly D, Ghosh K (2018) Contextual word embedding: a case study in clustering tweets about emergency situations. In: Companion of the the web conference 2018 on the web conference 2018. International world wide web conferences steering committee, pp 73–74
8. Gupta H, Jamal MS, Madisetty S, Desarkar MS (2018) A framework for real-time spam detection in twitter. In: 2018 10th international conference on communication systems & networks (COMSNETS). IEEE, pp 380–383
9. Imran M, Castillo C, Ji L, Meier P, Vieweg S (2014) Aidr: Artificial intelligence for disaster response. In: Proceedings of the 23rd International Conference on World Wide Web. ACM, pp 159–162
10. Imran M, Castillo C, Diaz F, Vieweg S (2015) Processing social media messages in mass emergency A survey. *ACM Comput Surv (CSUR)* 47(4):67
11. Khosla P, Basu M, Ghosh K, Ghosh S (2017) Microblog retrieval for post-disaster relief: Applying and comparing neural ir models. arXiv:1707.06112
12. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv:1408.5882
13. Li L, Zhang Q, Wang X, Zhang J, Wang T, Gao T-L, Duan W, Tsoi KK-f, Wang F-y (2020) Characterizing the propagation of situational information in social media during covid-19 epidemic A case study on weibo. *IEEE Trans Comput Soc Syst* 7(2):556–562
14. Madichetty S, Sridevi M (2019) Detecting informative tweets during disaster using deep neural networks. In: 2019 11th international conference on communication systems & networks (COMSNETS). IEEE, pp 709–713
15. Madisetty S, Desarkar MS (2017) Identification of relevant hashtags for planned events using learning to rank. In: International joint conference on knowledge discovery, knowledge engineering, and knowledge management. Springer, pp 82–99
16. Madisetty S, Desarkar MS (2017) An ensemble based method for predicting emotion intensity of tweets. In: International conference on mining intelligence and knowledge exploration. Springer, pp 359–370
17. Madisetty S, Desarkar MS (2017) Nsemo at emoint-2017: an ensemble to predict emotion intensity in tweets. In: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp 219–224

18. Madisetty S, Desarkar MS (2017) Exploiting meta attributes for identifying event related hashtags
19. Madisetty S, Desarkar MS (2018) A neural network-based ensemble approach for spam detection in twitter. *IEEE Trans Comput Soc Syst* 5(4):973–984
20. Mesleh AMA (2007) Chi square feature extraction based svms arabic language text categorization system. *J Comput Sci* 3(6):430–435
21. Nazer TH, Morstatter F, Dani H, Liu H (2016) Finding requests in social media for disaster relief. In: 2016 IEEE/ACM international conference on Advances in social networks analysis and mining (ASONAM). IEEE, pp 1410–1413
22. Nguyen DT, Mannai KAA, Shafiq 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
23. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. *J Mach Learn Res* 12:2825–2830
24. Purohit H, Castillo C, Pandey R (2020) Ranking and grouping social media requests for emergency services using serviceability model. *Soc Netw Anal Min* 10(1):1–17
25. Qazi U, Imran M, Ofli F (2020) Geocov19: a dataset of hundreds of millions of multilingual covid-19 tweets with location information. *SIGSPATIAL Spec* 12(1):6–15
26. Rajdev M, Lee K (2015) Fake and spam messages: Detecting misinformation during natural disasters on social media. In: 2015 IEEE/WIC/ACM International conference on web intelligence and intelligent agent technology (WI-IAT), vol 1. IEEE, pp 17–20
27. Rios A, Kavuluru R (2015) Convolutional neural networks for biomedical text classification: application in indexing biomedical articles. In: Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics. ACM, pp 258–267
28. Rudra K, Ghosh S, Ganguly N, Goyal P, Ghosh S (2015) Extracting situational information from microblogs during disaster events: a classification-summarization approach. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, pp 583–592
29. Rudra K, Sharma A, Ganguly N, Ghosh S (2016) Characterizing communal microblogs during disaster events. In: 2016 IEEE/ACM international conference on Advances in social networks analysis and mining (ASONAM). IEEE, pp 96–99
30. Rudra K, Ganguly N, Goyal P, Ghosh S (2018) Extracting and summarizing situational information from the twitter social media during disasters. *ACM Trans Web (TWEB)* 12(3):17
31. Rudra K, Sharma A, Ganguly N, Ghosh S (2018) Characterizing and countering communal microblogs during disaster events. *IEEE Trans Comput Soc Syst* 5(2):403–417
32. Sakaki T, Okazaki M, Matsuo Y (2013) Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Trans Knowl Data Eng* 25(4):919–931
33. Sarkar A, Roy S, Basu M (2019) Curating resource needs and availabilities from microblog during a natural disaster: A case study on the 2015 chennai floods. In: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, pp 338–341
34. Socher R, Huval B, Bath B, Manning CD, Ng AY (2012) Convolutional-recursive deep learning for 3d object classification. In: Advances in neural information processing systems, pp 656–664
35. Sreenivasulu M, Sridevi M (2017) Mining informative words from the tweets for detecting the resources during disaster. In: International conference on mining intelligence and knowledge exploration. Springer, pp 348–358
36. Sreenivasulu M, Sridevi M (2018) A survey on event detection methods on various social media. In: Recent findings in intelligent computing techniques. Springer, pp 87–93
37. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(1):1929–1958
38. Torkildson MK, Starbird K, Aragon C (2014) Analysis and visualization of sentiment and emotion on crisis tweets. In: International conference on cooperative design, visualization and engineering. Springer, pp 64–67
39. Vapnik VN, Vapnik V (1998) Statistical learning theory, vol 1. Wiley, New York
40. Varga I, Sano M, Torisawa K, Hashimoto C, Ohtake K, Kawai T, Jong-Hoon O, De Saeger S (2013) Aid is out there Looking for help from tweets during a large scale disaster. In: *ACL* (1), pp 1619–1629
41. 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. *CiteSeer*, pp 385–392
42. Wang G, Hao J, Ma J, Jiang H (2011) A comparative assessment of ensemble learning for credit scoring. *Expert Syst Appl* 38(1):223–230
43. Wang S, Chen Z, Liu B, Emery S (2016) Identifying search keywords for finding relevant social media posts. In: Thirtieth AAAI conference on artificial intelligence

44. Wolpert DH (1992) Stacked generalization. *Neural Netw* 5(2):241–259
45. Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: *Icml*, vol 97, pp 412–420
46. Zeiler MD (2012) Adadelta: an adaptive learning rate method. arXiv:[1212.5701](https://arxiv.org/abs/1212.5701)

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