Sādhanā (2020)45:270
https://doi.org/10.1007/s12046-020-01504-0

Detection of situational information from Twitter during disaster using deep learning models

SREENIVASULU MADICHETTY* and SRIDEVI MUTHUKUMARASAMY

Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, India
e-mail: sreea568@gmail.com; msridevi@nitt.edu

MS received 5 December 2019; revised 21 June 2020; accepted 30 August 2020

Abstract. Twitter is an excellent resource for communicating between the victims and organizations during a disaster. People share opinions, sympathies, situational information, etc., in the form of tweets during a disaster. Detecting the situational tweets is a challenging task, which is very helpful to both humanitarian organizations and victims. There is a chance that both situational and non-situational information may be present in a tweet. Most of the existing works focused on identifying single-information-type tweets like situational information, actionable information, useful information, etc. Detecting the mixture of situational and non-situational information tweets remains a challenging task. Although existing works designed an SVM classifier using low-level lexical and syntactic features for classifying situational and non-situational tweets, their method does not work well for a mixture of situational and non-situational information tweets. This paper addresses the problem of detecting the situational tweets using different deep learning architectures such as Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Bi-directional Long Short-Term Memory (BLSTM) and Bi-directional Long Short-Term Memory with attention (BLSTM attention). Moreover, deep learning models are applied to Hindi language tweets besides English language tweets for identifying the situational information during a disaster. Some of the tweets are posted in the Hindi language, where the information is not available in the English language in countries like India during the disaster. Experiments are performed on various disaster datasets such as Hagupit cyclone, Hyderabad bomb blast, Sandhy shooting, Nepal Earthquake and Harda rail accident in both in-domain and cross-domain. The results of deep learning models demonstrate that it outperforms the existing traditional approach, such as the SVM classifier with low-level lexical and syntactic features for detecting the situational tweets during the disaster. Additionally, to our best knowledge, this is the first attempt in applying the deep learning models to identify the Hindi language situational tweets during the disaster.

Keywords. Deep learning model; disaster; tweets.

1. Introduction

Numerous studies [1–11] have shown that people use social media to understand the situational awareness during disasters (natural/human-made disasters). Different users post different categories of tweets on Twitter related to the disaster. Different categories of tweets can be as defined as casualties, infrastructure damage [12], resource requests [13], availability of resources [14–18], sympathy for victims, prayers to the people, emotion tweets [19, 20], useful information [21], spam tweets [22, 23], etc. Among them, only some categories of tweets are important to humanitarian organizations/government organizations during a disaster. Therefore, in [24], the authors categorized the tweets into situational and non-situational tweets where situational tweets contain information related to the casualties, infrastructure damage, etc., and non-situational tweets contain information related to the sympathizing, communal, opinions, etc. The authors used machine learning classifiers based on the Bag-Of-Words (BOW) model for detecting the situational information during the disaster. In contrast the authors of [25] worked on characterizing the communal and non-communal tweets, which are a subset of non-situational tweets. Recently, the authors in [26] designed a method based on the low-level lexical and syntactic features using a machine learning classifier for classifying situational and non-situational tweets and summarization of situational tweets during the disaster. Similarly, the authors [26] developed a method based on the low-level lexical and syntactic features for Hindi tweets. However, all the works focused on machine learning classifiers based on the hand-crafted features for classifying situational and non-situational tweets during the disaster.

*For correspondence

Published online: 30 October 2020

In a practical scenario, multiple information such as situational and non-situational information rather than just situational information can be present in a single situational tweet. These types of tweets are considered as raw tweets in this work. Examples of a raw tweet for both situational and non-situational tweets are shown in Table 1. However, in the existing works, the authors of [26] showed that most of the tweets contain a mixture of situational and non-situational information during the disaster. Therefore, they divide the tweets into multiple fragments using the end markers (‘!’, ‘.’ , ‘?’) and then apply the classifier with the use of low-level lexical and syntactic features for classifying the situational and non-situational tweets. Examples of fragment tweets are shown in Table 2. However, the authors mentioned that their method does not perform well for a mixture of situational and non-situational information tweets compared with the situational tweets (fragment) only. Moreover they neglect the fragment tweets that are less than five words, and there is a chance to lose the situational information from the tweets. Furthermore, their methods do not give much accuracy on both fragment and raw tweets posted during the disaster.

Therefore, this paper addresses the classification of situational and non-situational tweets where situational tweet contains both situational and non-situational information by applying different deep learning methods. Most importantly we used two different language tweets, both English and Hindi, for applying deep learning models.

The overall contributions of this paper are as follows:

1. Compare the different word embeddings such as Word2vec, Glove, Bi-directional Encoder Representations from Transformers (BERT) and crisis word embeddings for finding suitable word embeddings to detect the situational tweets during the disaster. It is found that crisis embeddings are more suitable and allow independent vocabulary features for cross-domain (training and testing the model with past and future event datasets, respectively).
2. Perform different experiments using different deep learning architectures such as Convolution Neural Network (CNN), Long Short-Term Memory (LSTM), Bi-directional Long Short-Term Memory (BLSTM) and Bi-directional Long Short-Term Memory with attention (BLSTM attention). It is found that BLSTM attention mechanism based on the crisis embeddings gives the best performance for detecting the situational tweets where it includes both situational and non-situational information during the disaster. And also, for Hindi tweets, CNN gives better performance than the other deep learning architectures.
3. BLSTM attention with crisis embeddings is compared to an existing approach such as an SVM classifier based on the low-level lexical and syntactic features across various diverse datasets on both English and Hindi language tweets in both in-domain and cross-domain.

The paper is organized as follows. Section 2 explains the related works. Section 3 describes the different deep learning models used for detecting situational tweets during the disaster. Experiment results are discussed in Section 4. Section 5 concludes the paper.

2. Related work

Most of the existing methodologies [24, 26, 27] have been tried to detect the situational information during the disaster. In [28], the authors used uni-gram and bi-gram features for classifying the tweets into user-defined categories

Table 1. Examples of the raw tweet where the situational tweet contains a mixture of situational and non-situational information.

Tweet no.	Situational information
1.	RT @KiranKS: #Hyderabad blasts: Blast 1 (Konark) at 7.01 PM, Blast 2 (Venkatadri) at 7.06 PM. And I had this tweet at 7.07 PM !!! [URL]
2.	RT @scott_eff: Oh, mercy. 27 dead in the Connecticut school shooting. I'm not sad, I'm just angry. F*** your second amendment that lets ...1
3.	@BDUTT: two blasts in Dilsukh Nagar in Hyderabad, several casualties, 50 reported injured. Oh god! What is going on!!!
4.	Oh my god: (shooting at an elementary school and 19 kids and 9 adults died. What the hell is wrong with people
	Non-situational information
5.	22W got its name...#HAGUPIT...but it won't stop here and it will go on intensifying...(image from JMA) [URL]
6.	I am here to share your pain, grief, PM to Hyderabad blast victims hindustan times, Hindustan Times I am here
7.	My prayers go out to all of the families affected in the Connecticut elementary school shooting
8.	Please #Hagupit, keep away from the Philippines this Christmas season

Table 2. Examples of the fragment tweet for situational and non-situational information.

Tweet no.	Situational information
1.	Philippine storm kills 27, but no damage to Manila
2.	Alert list, 56 critical areas due to Manila, Philippines the government on
3.	DSWD says ready with food packs, shelters for Hagupit
4.	Update, joint typhoon warning center's forecast track for typhoon as of 11 am
	Non-situational information
5.	I never thought I had a trauma until I heard of rubyph spare the Philippines!
6.	We caught 3 days of typhoon Hagupit's motion over Philippines
7.	Football, Philippine–Thailand match may be postponed due to Hagupit sportstapapp
8.	Pray for protection, not for suspension

during the disaster. The authors of [24] used the machine learning classifier based on BOW model features for classifying the situational and non-situational tweets during the disaster. However, it depends on the vocabulary present in the training tweets and it does not give better performance for the unseen vocabulary that is present only in the testing tweets but not in training tweets. It works well for the same disaster event where training and testing are done on the same disaster dataset only but not helpful for other disaster events. Subsequently, the authors [29] used the SVM classifier based on the low-level lexical and syntactic features for classifying the situational and non-situational tweets during the disaster. Later, the authors of [27] added some more features to the low-level lexical and syntactic features for improving the performance in classifying situational and non-situational tweets during the disaster. However, they are independent of vocabulary present in the training tweets and it gives a good performance even when training and testing are done for different disaster events. Moreover, they developed a technique for summarizing the situational information tweets to understand the situation during the disaster after classifying the situational and non-situational tweets. Later, the authors [26] improved the model by adding two more features and also developed low lexical features for detecting the resource-poor Hindi language tweets during the disaster. The authors show that Hindi tweets are useful compared with English tweets because Hindi tweets give information earlier than the English tweets, and they cover the information that is not available in English tweets during the disaster. They also explained that the same tweet contains situational and non-situational information and showed that performance is improved after the fragmentation of tweets. However, the authors indicate that their methods do not give better performance for the tweet that contains situational and non-situational information during the disaster. However, it cannot decide whether tweets posted during the disaster have either situational information or non-situational information or both situational information and non-situational information. It may include all categories of information during the disaster. However, it is essential to detect the situational information during the disaster for the humanitarian organizations and victims. Moreover, traditional methods do not differentiate the situational and non-situational informative tweets accurately due to the presence of the non-situational information present in both situational and non-situational tweets. However, all the existing methods used handcrafted features for detecting the situational tweets during the disaster.

The authors in [30] present a system named as CrisMap, which is used for quickly collecting and analysing the messages from Twitter during the disaster. After analysing the information from the tweets, it projects into the maps where the severe damage occurs and situational awareness. The system is developed by exploiting the word embeddings and big data technologies. They validated the model using two Italian natural disaster datasets and performed a case study on the latest earthquake that occurred in central Italy. However, they are not focused on English tweets. In [31], the authors developed a method based on neural networks without feature engineering for classifying crisis-related information during the disaster. They performed experiments in event data, out-of-event data and a combination of both. The authors show that CNN with crisis embeddings works better than the CNN with other embeddings. CNN with crisis word embeddings performs better than the non-neural models such as SVM, Logistic Regression and Random forest. The features used for non-neural models are uni-gram, bi-gram and trigram features and they convert features into TF-IDF vectors; χ^2 feature selection algorithm is used for selecting the features in the SVM classifier. In [32], the authors use deep learning models based on the Word2vec and Glove vector representation of Facebook posts for detecting domestic violence posts. However, their work did not focus on either natural or human-made disaster tweets from Twitter and did not focus on different languages. Also, they do not generalize on different disaster event datasets. In [33], the authors develop a graph-based semi-supervised learning method with CNN for classifying crisis-related tweets during the disaster. It utilizes the unlabelled data of the Nepal Earthquake and Queensland floods for experimenting during the early stages of the disaster and also compares without adding unlabelled data. However, all the works focused on the single information type on the tweet during a disaster. The common limitation of the afore-mentioned works, which are related to the deep learning methods, is that it does not focus and discuss the use of deep learning methods for detecting the mixture of situational and non-situational information tweets during the disaster. Moreover, they did not generalize on different disaster event datasets for classifying situational and non-situational tweets during the disaster.

To overcome all these limitations, we propose to use different deep learning models with suitable word embeddings for detecting the situational tweets where they contain a mixture of situational and non-situational information during the disaster and experimented on diverse disaster datasets.

3. Deep learning models

Deep learning models such as CNN, LSTM, BLSTM and BLSTM with attention mechanism are used for detecting the situational tweets in which situational tweet contains both situational and non-situational information during the disaster. The general architecture of the deep learning models is shown in Figure 1. It contains embedding layer, neural architecture, drop-out layer, fully connected layer and soft-max layer, which are shown as follows.

- *Embedding layer*: Tweets are given as an input to the embedding layer. Each tweet is divided into tokens. Each token is converted into a fixed-sized word vector, also known as word embedding. Pre-trained word embeddings such as GloVe, Word2vec, Crisis word embedding, BERT embeddings, etc. are used for generating the word vector for each token. Among the pre-trained word embeddings, crisis word embeddings are developed using 52 million crisis-related tweets using the Word2vec algorithm. The authors of [34] suggest that crisis word embeddings are useful for classification tasks on crisis-related tweets and also prove experimentally for our tasks that are shown in Section 4.2. General embeddings like a Glove, Word2vec, BERT embeddings, etc. do not have much impact on disaster-related tweets because the terms used in the disaster may come in general scenarios also. The output of the embedding layer is fixed-sized word vectors that are given as an input to the deep learning models.
- *Dropout layer*: Dropout is a technique used to ignore the randomly selected neurons from deep learning models and it reduces over-fitting. Therefore, it is used before and after the deep learning model at a rate of 0.25 and 0.50, respectively.
- *Fully connected layer*: The output of a deep learning model is given as an input to the fully connected layer. The fully connected layer is a dense output layer with a number of neurons that is equal to the number of classes.
- *Softmax layer*: Softmax layer uses the softmax activation function for assigning the decimal values of class labels to the output of the fully connected layer.
- *Neural architecture*: Deep learning models are differentiated by various neural architectures while the remaining is the same. CNN plays a vital role in text classification tasks for extracting the most informative *n*-grams; recently it was used for sentiment analysis [35]. CNN identifies the local predictors of the text, which gives information for text classification tasks irrespective of word order in the text. Later, it combines all the local predictors for determining the class label. The output of CNN is a fixed-size vector. CNN captures the informative words (situational words) in a tweet using a max-pooling layer for classifying situational tweet and non-situational information. However CNN does not consider word order in a sentence for text classification tasks, whereas LSTM is a popular Recurrent Neural Network (RNN)

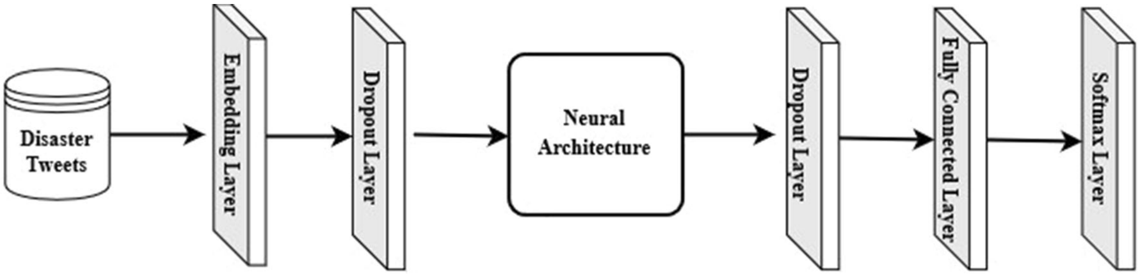

Figure 1. General architecture of the deep learning models.

architecture. LSTM uses three cell states: input gate, forget gate and output gate cell to overcome the problem of vanishing gradient in a standard RNN. The input gate (remember gate) cell remembers the context of input for learning the long-term dependences in a tweet. Forget gate cell is used to erase the unimportant information in a tweet and the output cell passes the information to the next state. Therefore, LSTM gives State-of-the-art results for many text sequence processing tasks [36, 37] and text classification tasks [38]. LSTM considers the order of words in a sentence while keeping the order in the memory cell. BLSTM [39] is a Bi-directional LSTM, and it captures information from a tweet in both forward and backward directions. The attention mechanism is a memory access mechanism that allows the model to learn the more important word vectors in a sequence of word vectors in forward and backward directions from the tweet. BLSTM with attention mechanism remembers the long-term dependences in both directions of a tweet and gives weight to the important context for classifying tweets. For example, in situational tweets, it captures the situational contexts in a tweet for identifying the situational tweet during a disaster.

The deep learning models are a clear advance on current methods because there is no requirement of hand-crafted features, lexicons, etc. It is very difficult to develop robust hand-crafted features for detecting the situational tweets during the disaster for avoiding the use of deep learning models.

4. Experiment results

Existing works [24–26] heavily depend on traditional machine learning algorithms and hand-crafted features. However, they do not improve the performance for classifying situational and non-situational tweets much during the disaster. Existing work [26] shows that the SVM classifier based on low-level lexical and syntactic features gives higher performance than the BOW model. However, they do not study their models on large datasets and unbalanced datasets. In fact, for robust models, there is a need to test on large datasets. Moreover, they did not focus on the mixture of situational and non-situational information tweets. Therefore, the SVM classifier with low-level lexical and syntactic features has been experimented similarly to that in [26] for comparing the deep learning models. It uses 10-fold cross-validation for training and testing datasets similar to [26]. Deep learning models are implemented using Keras [40]. It also uses 10-fold cross-validation for comparing the existing method and also uses the parameters that are shown in Table 3.

4.1 *Datasets*

Each disaster dataset contains unbalanced raw tweets (numbers of situational and non-situational information tweets are unequal), balanced raw tweets (numbers of situational and non-situational information tweets are equal) and fragment tweets shown in Table 4. The authors [26] in the existing work annotate, where the fragments are significantly less. However, they have a large number of annotated raw tweets. They provide annotated unbalanced and balanced raw tweets for experiments. However, the authors [26] focused only on fragment tweets but not raw tweets. It is very important to detect the raw tweets for summarization. The different types of disaster events used for experimenting with the models are shown here.

1. *HydBlast (Hyderabad blast)*: Two bomb blasts in Hyderabad, 2013.
2. *SanHShoot (Sandy Hook shooting)*: A 20-year-old attacker shot and killed 20 children and 6 adult staff members on December 14, 2012, in Newtown, Connecticut, USA.
3. *TypHagupit (Typhoon Hagupit)*: A strong cyclone occurred in the Philippines in 2014, and it was named as Typhoon Hagupit.
4. *NepEquake (Nepal Earthquake)*: A severe earthquake in Nepal on 2015.
5. *HarDerail (Rail accident)*: Two passenger trains derailed near Harda, Madhya Pradesh, in India, August 2015.

Different disaster event tweets across different regions of the world are used to ensure the diversity of the tweets in the model. Among the datasets, the first three disaster event datasets use English tweets that are categorized into raw tweets and fragment tweets. The next two datasets are Hindi tweets provided in [26]. Hyderabad bomb blasts, Sandy shooting and Harda rail accidents are related to human-made disasters. Nepal Earthquake and Typhoon Hagupit are the events related to the natural disaster.

4.2 *Raw tweets*

In [26], the authors explain that their method does not give good performance on the raw tweets. However, it is essential to detect these raw tweets during the disaster. Different experiments are performed using deep learning models on these raw datasets for detecting situational raw tweets during the disaster.

The performance of any deep learning model mainly depends on the word embeddings used in the model. Experiments are performed using different pre-trained word embeddings such as Word2vec, Glove, BERT-Base, BERT-Large and Crisis word embeddings on raw tweets to select a suitable pre-trained word embedding model. CNN (deep learning model) is used for choosing the best word embedding model due to its simple architecture compared

Table 3. Parameter details of the different deep learning models.

Sl. no.	Deep learning model	Parameter
1	CNN	• Total number of layers = 8 • Number of embedding layers = 1 • Number of dropout layers = 2 • Number of convolutional layers = 1 • Number of pooling layers = 1 • Number of flatten layers = 1 • Number of fully connected layers = 1 • Number of soft-max layers = 1 • Loss function = "Binary cross entropy" • Filter sizes = {3,4,5} • Number of filters for each size = 100 • Learning rate = 0.01 • Optimizer = "Adam" • Dropout ratio = 0.25, 0.50 • Mini batch size = 128 • Maximum number of epochs = 15
2	LSTM	• Total number of layers = 5 • Number of embedding layers = 1 • Number of dropout layers = 2 • Number of LSTM layers = 1 • Number of soft max layers = 1 • Maximum number of epochs = 15 • Mini batch size = 128 • Dropout ratio = 0.25, 0.50 • Learning rate = 0.01
3	BLSTM	• Total number of layers = 5 • Number of embedding layers = 1 • Number of dropout layers = 2 • Number of Bi-directional LSTM layers = 1 • Number of soft max layers = 1 • Maximum number of epochs = 15 • Mini batch size = 128 • Dropout ratio = 0.25, 0.50 • Learning rate = 0.01
4	BLSTM with attention	• Total number of layers = 6 • Number of embedding layers = 1 • Number of dropout layers = 2 • Number of Bi-directional LSTM layers = 1 • Number of Attention layers = 1 • Number of soft max layers = 1 • Maximum number of epochs = 15 • Mini batch size = 128 • Dropout ratio = 0.25, 0.50 • Learning rate = 0.01

with the other deep learning architectures like LSTM. From Table 5, it is evident that the BERT embedding gives poor performance than the others in the classification of crisis-related tweets even though BERT embedding is more advanced (capturing both positional and contextual information) than the other pre-training models. Glove embedding got a slight improvement in the parameters of F1-score and accuracy as well as Word2vec on the precision parameter in the Sandy shooting dataset for raw tweets (a mixture of situational and non-situational information tweet). However, crisis embedding outperforms the other pre-trained embeddings in all datasets. Therefore, crisis

Table 4. Different types of tweets in each disaster dataset.

Types of tweets	HydBlast	SanHShoot	TypHagupit	NepEquake	HarDerail
Unbalanced raw tweets	4,930	4,998	4,996	–	–
Balanced raw tweets	3136	3022	3128	–	–
Fragment tweets	832	864	906	562	240

Table 5. Comparison of different word embeddings using CNN on raw tweets.

Embedding type	Precision	Recall	$F1$-score	Accuracy
TypHagupit dataset				
Word2Vec embeddings	97.73	98.30	98.01	97.27
Glove embeddings	97.63	98.45	98.03	97.29
BERT-Base	95.03	98.80	96.87	95.61
BERT-Large	95.64	98.95	97.26	96.17
Crisis embeddings	*98.38*	*99.21*	*98.79*	*98.33*
HydBlast dataset				
Word2Vec embeddings	97.09	98.43	97.75	96.91
Glove embeddings	97.07	98.47	97.76	96.93
BERT-Base	94.57	98.72	96.59	95.25
BERT-Large	94.71	98.63	96.62	95.29
Crisis embeddings	*97.9*	*98.84*	*98.39*	*97.80*
SanHShoot Dataset				
Word2Vec embeddings	*98.54*	97.58	98.05	97.31
Glove embeddings	98.32	98.02	*98.17*	*97.45*
BERT-base	97.53	95.70	96.59	95.30
BERT-large	97.82	95.73	96.75	95.52
Crisis embeddings	98.23	*97.96*	98.09	97.35

Best performed values are in italics

embedding is selected and used for finding suitable deep learning models for classifying situational and non-situational tweets where situational tweet contains a mixture of situational and non-situational information during the disaster. Different experiments are performed using different deep learning architectures on various disaster datasets such as unbalanced and balanced raw tweets. It is found that BLSTM attention mechanism based on the crisis embeddings gives the best performance for large size datasets and the results are shown in Table 6 across different unbalanced datasets. Therefore, BLSTM with attention mechanism based on crisis embeddings is considered as a default deep learning model for detecting situational tweets during the disaster.

The first analysis highlighting the impact of deep learning models compared with the existing work for detecting the situational tweets is shown in Table 6. The reason for the poorer performance of the existing method on raw tweets is that the method captures the information mostly present in both situational and non-situational tweets due to the chance of some non-situational fragments present in the situational raw tweets. Therefore, the existing method cannot differentiate the situational and non-situational raw tweets during the disaster. Both situational and non-situational tweets are equally selected and compared with the current method to check the effect of the balanced dataset on detecting the situational tweets, shown in Table 7.

However, the performance of deep learning models does not show a significant difference between balanced and unbalanced datasets. The most striking result of the data is that deep learning models give the best result above 95% for all parameters in in-domain. Also much higher values for classifying situational tweets from non-situational tweets with respect to those reported by the authors [26] in case of large datasets are found, especially for raw tweets. However, at the time of the disaster, tweets contain both situational and non-situational information.

4.3 *Fragment tweets*

The authors in [26] provided the annotations of fragment tweets for experimenting with the deep learning models. They showed that low-level lexical and syntactic features improve the performance by fragmenting the tweets, and the quality of tweets also improved for summarization of the tweets. The results of the different pre-trained word

Table 6. Comparison of deep learning models with existing model [26] on unbalanced raw tweets.

Model	Precision	Recall	$F1$-score	Accuracy
TypHagupit dataset				
Existing model	64.78	96.28	76.85	62.84
BLSTM with attention	*99.20*	*99.70*	*99.44*	*97.89*
BLSTM	98.20	98.80	98.5	97.97
LSTM	98.5	98.6	98.6	98.00
CNN	98.38	99.21	98.79	98.33
HydBlast dataset				
Existing model	76.66	95.45	84.97	76.19
BLSTM with attention	*98.6*	*99.04*	*98.81*	97.70
BLSTM	97.9	98.78	98.37	97.76
LSTM	96.87	99.13	97.98	97.22
CNN	97.9	98.84	98.39	*97.80*
SanHShoot dataset				
Existing model	75.73	96.47	84.77	75.79
BLSTM with attention	*99.07*	*98.07*	*98.56*	97.31
BLSTM	98.15	97.63	97.89	97.06
LSTM	97.68	97.83	97.75	96.86
CNN	98.23	97.96	98.09	*97.35*

Best performed values are in italics

Table 7. Comparison of deep learning model with existing work [26] on balanced raw tweets using different metrics.

	Recall		$F1$-score		Accuracy	
Dataset	Existing method [26]	Deep model	Existing method [26]	Deep model	Existing method [26]	Deep model
TypHagupit	60	*97.63*	59.08	*97.20*	65.57	*97.18*
HydBlast	63.75	*99.36*	61.30	*96.87*	59.85	*96.78*
SanHShoot	85	*96.09*	70.35	*96.31*	66.68	*96.32*

Note that ‘deep model’ indicates the deep learning model (Bi-directional LSTM with attention mechanism).

Best performed values are in italics

embedding models on fragment tweets are shown in Table 8.

Results on different disaster datasets have proved that no specific pre-training word embedding model gives the best performance on fragment tweets. Although BERT embeddings (BERT-Base and BERT Large) capture the position of a word, contextual information from the tweet does not give any significant improvement on the disaster fragment tweets. Moreover, it provides less performance compared with the other pre-trained word embeddings. The results of deep learning models across different datasets are presented in Tables 9, 10 and 11.

These results reveal that LSTM does not give good performance in almost all the datasets to detect situational tweets due to either small datasets or fragment tweets. From the observation, it is found that the performance of the LSTM on fragmented tweets is lower due to information lost during the fragmentation of tweets. There is no significant variation among CNN, BLSTM, BLSTM attention models, but BLSTM attention gives better results than the other models in case of recall parameter. However, the recall parameter is crucial for detecting situational tweets during the disaster because if it detects the situational tweets as non-situational tweets then there is a chance of human loss. Therefore, for further comparison with the actual work, BLSTM attention is used as a default deep learning model in this paper. For other parameters such as precision, $F1$-score and accuracy, BLSTM performs best for the Sandy Hook shooting, CNN shows best for the Typhoon Hagupit dataset and BLSTM attention for the Hyderabad bomb blast (except precision parameter). The reason for these variations might be the very small dataset. Deep learning models give the best performance compared with the current work based on fragmented tweets across different parameters. The current work does not correctly detect some tweets that do not have low-lexical and syntactic features within them. The authors of [26] mentioned the type of tweets in which their method does not work correctly for classifying situational and non-situational tweets. However, deep learning models are proposed to use to overcome this limitation for detecting the situational tweets during the disaster.

Table 8. Comparison of different word embeddings using CNN on fragment tweets.

Embedding type	Precision	Recall	$F1$-score	Accuracy
TypHagupit dataset				
Word2Vec embeddings	89.30	*90.63*	*89.93*	*89.95*
Glove embeddings	*90.02*	89.49	89.61	89.74
BERT-base	85.53	86.34	85.87	85.86
BERT-large	86.47	89	87.58	87.41
Crisis embeddings	88.8	88.16	88.71	88.63
HydBlast dataset				
Word2Vec embeddings	*90.20*	85.55	87.69	88.22
Glove embeddings	89.26	87.16	*88.10*	*88.58*
BERT-base	86.21	86.20	85.96	85.96
BERT-large	89.71	81.93	85.42	86.07
Crisis embeddings	87.16	*87.37*	85.16	86.21
SanHShoot dataset				
Word2Vec embeddings	94.66	*93.93*	94.21	*94.21*
Glove embeddings	*94.71*	93.90	*94.22*	*94.21*
BERT-base	93.89	91.91	92.77	92.95
BERT-large	94.15	92.15	93.03	93.18
Crisis embeddings	93.55	91.76	92.58	92.82

Best performed values are in italics

Table 9. Comparison of deep learning models on fragment tweets.

Model	Precision	Recall	$F1$-score	Accuracy
TypHagupit dataset				
BLSTM with attention	*88.49*	*94.13*	87.98	87.96
BLSTM	88.32	86.76	87.43	87.51
LSTM	30.05	60.00	39.96	50.33
CNN	88.8	88.16	*88.71*	*88.63*
HydBlast dataset				
BLSTM with attention	84.38	*90.59*	*87.25*	*86.77*
BLSTM	85.80	87.20	86.27	86.18
LSTM	30	60.00	39.96	50.12
CNN	*87.16*	87.37	85.16	86.21
SanHShoot dataset				
BLSTM with attention	92.09	*92.12*	91.95	92.01
BLSTM	*94.04*	92.11	*92.95*	*93.04*
LSTM	33	70	45.58	47.57
CNN	93.55	91.76	92.58	92.82

Best performed values are in italics

Table 10. Comparison of deep learning model with existing work [26] on fragment tweets.

	Recall		$F1$-score		Accuracy	
Dataset	Existing method [26]	Deep model	Existing method [26]	Deep model	Existing method [26]	Deep model
TypHagupit	94	*94.13*	87	*87.98*	85.86	*87.96*
HydBlast	85	*90.59*	84	*87.25*	84.26	*86.77*
SanHShoot	87	*92.11*	89	*91.95*	90.04	*92.01*

Note that ’deep model’ indicates the deep learning model (Bi-directional LSTM with attention mechanism).

Best performed values are in italics

Table 11. Comparison of deep learning models on Hindi tweets.

Model	Precision	Recall	$F1$-score	Accuracy
NepEquake dataset				
BLSTM with Attention	68.41	*66.80*	*67.46*	68.27
BLSTM	58.39	62.76	60.32	*72.76*
LSTM	57.77	56.56	57.10	59.02
CNN	*94.67*	39.77	55.98	68.65
HarDerail dataset				
BLSTM with Attention	71	48.48	57.61	45.00
BLSTM	53	47.02	49.83	58.55
LSTM	69.66	*60.35*	*64.67*	54.49
CNN	*89.40*	38.21	53.53	*66.71*

Best performed values are in italics

4.4 *Hindi tweets*

The authors in [26] showed that Hindi tweets give information faster than the English tweets because local people post the tweets in their language immediately during the disaster. Therefore, detecting the Hindi tweets related to situational information is also essential during the disaster. It uses low-level lexical and syntactic features for classifying situational and non-situational tweets. They prepared custom lexicons for Hindi tweets. Human resources are required to prepare the Hindi lexicons. The main drawback of this method is that it cannot work correctly for the new words other than lexicon words that are present in the Hindi tweets. To overcome these limitations, deep learning models are proposed to use for detecting situational information on Hindi tweets.

The embedding layer is similar to English tweets in the deep learning models. However, the word embeddings are different from the English tweets. The authors of [41, 42] used the skip-gram model and Continuous Bag-Of-Words (CBOW) model for generating the word embeddings from the large Wikipedia Hindi text. The effects of two different Hindi word embeddings on deep learning models for detecting situational tweets in-domain are provided in Table 12. Among the two word embeddings (skip-gram and CBOW model), word embeddings generated using the skip-gram model give better performance than the CBOW model. Therefore, it is suggested that skip-gram word embeddings are useful for the detection of situational tweets and also used for further analysis. The rest of the layers used for Hindi tweets are similar to the model used for the English tweets.

Table 12. Comparison of word embedding on Hindi tweets.

Model	Precision	Recall	$F1$-score	Accuracy
NepEquake dataset				
Skip_gram	*94.67*	*39.77*	*55.98*	*68.65*
CBOW	87.40	38.06	52.42	66.32
Harderail dataset				
Skip_gram	*89.40*	38.21	52.21	*66.71*
CBOW	79.40	*39.05*	*52.35*	65.26

Best performed values are in italics

Similar to English tweets, the architectures of deep learning models such as CNN, LSTM, BLSTM and BLSTM attention are used for experiments. There was a significant positive correlation between the datasets and deep learning models. LSTM, BLSTM and BLSTM attention gives the worst performance for very small datasets. However CNN gives a better performance compared with others on precision value, which is shown in Table 11. For Hindi tweets, CNN gives better precision value than the existing works and the comparison of deep learning models (CNN) with the existing methods on Hindi tweets is shown in Table 13. Deep learning models do not give much value for accuracy parameters compared with the existing works. However, precision is also an essential parameter for detecting situational tweets during the disaster.

4.5 *Cross-domain*

The authors in [43] developed a method based on the advanced linguistic features for the detection of damage assessment of Italian tweets, especially for cross-event and out-domain. However, they are not focused on English tweets and also detection of situational tweets. In this work, cross-domain can be defined as training and testing the model with the past and future event datasets, respectively. The past event datasets such as Hyderabad bomb blasts and Sandy shooting are used for training the model, and future event datasets such as Hyderabad bomb blasts, Sandy shooting and Typhoon Hagupit are used for testing the model. In a real-time scenario, at the time of the disaster, tweets may contain both situational and non-situational information. Therefore, raw tweets are considered for cross-domain experiments on English tweets. The impact of word embeddings will be especially noticeable in cross-domain only. Therefore, deep learning models (which use word embeddings) are compared to the existing method in

Table 13. Comparison of deep learning model with existing work [26] on Hindi tweets.

	Precision		Accuracy	
Dataset	Existing method [26]	Deep model	Existing method [26]	Deep model
NepEquake	85.87	*94.67*	*81.3*	68.65
HarDerail	68.70	*89.40*	*77.93*	66.71

Best performed values are in italics

Table 14. Comparison of deep learning model with existing work [26] on training Sandy (past) dataset and testing different dataset (future) tweets.

	Recall		$F1$-score		Accuracy	
Testing dataset	Existing method [26]	Deep model	Existing method [26]	Deep model	Existing method [26]	Deep model
TypHagupit	29.46	*64.64*	43.13	*70.82*	57.56	*73.36*
HydBlast	47.50	*80.16*	54.67	*81.14*	24.20	*81.30*

Best performed values are in italics

Table 15. Comparison of deep learning model with existing work [26] on training Hyderabad (past) dataset and testing different dataset (future) tweets.

	Recall		$F1$-score		Accuracy	
Testing dataset	Existing method [26]	Deep model	Existing method [26]	Deep model	Existing method [26]	Deep model
TypHagupit	49.10	*98.72*	58.82	*74.37*	26.82	*65.98*
SanHShoot	87.87	*95.56*	69.87	*79.53*	41.13	*75.41*

Note that 'deep model' indicates the deep learning model (Bi-directional LSTM with attention mechanism).

Best performed values are in italics

cross-domain. The comparisons of deep learning model with existing method on English tweets are shown in Tables 14 and 15. In the case of a cross-domain scenario, as expected, our experiments showed better performance than the existing method [26]. The authors of [26] explain that their features are vocabulary independent and give better results, especially in cross-domain. However, in the case of raw tweets, it does not give much performance because a situational raw tweet contains non-situational information also. Therefore their method fails to differentiate these types of tweets, which has both situational and non-situational information. However, if a tweet has minimal situational information, it is useful to the humanitarian organizations. Further, the performance of cross-domain compared with the in-domain case is less. Therefore, at the initial stage of disaster, use of the past data (cross-domain) for training the datasets is suggested. At later stages of a disaster, after collecting the data, it is better to use the present data (in-domain) for training the dataset in the deep learning models.

It is checked in cross-domain also for training and testing the past and future event dataset on Hindi tweets, respectively. The precision value of the deep learning model and the existing method is 70% and 65.94%, respectively. Our results are below expectations for Hindi tweets as the size of the dataset is very low, and also the word embeddings are not domain-specific. Further, data collection would be needed to determine precisely how deep learning models work for Hindi tweets. There is a need to develop domain-specific (crisis-related) word embeddings for the Hindi tweets. However, CNN gives better performance than the existing method for the precision parameter.

5. Conclusion

The evidence from this study suggests that BLSTM with attention mechanism based on the crisis word embeddings performs better among the deep learning techniques described in Section 4.2 for detecting situational tweets where it contains a mixture of situational and non-situational information during the disaster. Even though BERT embeddings capture the position of the word and contextual information, they are not suitable for detecting the situational tweets. From the Hindi tweets results, it is found that CNN gives better performance than other deep learning models for identifying the situational tweets. We showed the importance of deep learning models for identifying the situational tweets during the disaster on various datasets with different sizes in different disaster events for English and Hindi languages. It is confirmed that deep learning models will also be useful in cross-domain for detecting the situational tweets. Therefore, it is suggested that BLSTM attention based on the crisis embedding trained with the past event data is beneficial in the initial stage of a disaster as we do not have the training data. It is also helpful for later stages of a disaster after getting the training data of the event. The limitation is that deep learning models do not give much performance on Hindi tweets due to the insufficient number of tweets available. However, these models provide excellent performance compared with traditional approaches. In the future, we plan to apply these deep learning models to some other local Indian languages like Telugu, Tamil, Kannada, etc.

References

[1] Imran M, Castillo C, Diaz F and Vieweg S 2015 Processing social media messages in mass emergency: a survey. *ACM Comput. Surv.* 47: 1–38

[2] Madichetty S and Sridevi M 2020 Improved classification of crisis-related data on Twitter using contextual representations. *Proc. Comput. Sci.* 167: 962–968

[3] Madichetty S and Sridevi M 2018 A survey on event detection methods on various social media. In: *Recent Findings in Intelligent Computing Techniques*, pp. 87–93

[4] Varga I, Sano M, Torisawa K, Hashimoto C, Ohtake K, Kawai T, Oh J-H and De Saeger S 2013 *Aid is out there: looking for help from Tweets during a large scale Disaster*. ACL, vol. 1, pp. 1619–1629

[5] Sreenivasulu M and Sridevi M 2020 Comparative study of statistical features to detect the target event during disaster. *Big Data Min. Anal.* 3(2): 121–130

[6] Rudra K, Goyal P, Ganguly N, Imran M and Mitra P 2019 Summarizing situational Tweets in crisis scenarios: an extractive–abstractive approach. *IEEE Trans. Comput. Soc. Syst.* 6: 981–993

[7] Madichetty S and Sridevi M 2020. Detecting informative tweets during disaster using deep neural networks. In: *Proceedings of the 2019 11th International Conference on Communication Systems & Networks (COMSNETS)*. IEEE, pp. 709–713

[8] Rudra K, Goyal P, Ganguly N, Mitra P and Imran M 2018 Identifying sub-events and summarizing disaster-related information from microblogs. In: *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 265–274

[9] Rudra K, Sharma A, Ganguly N and Ghosh S 2018 Characterizing and countering communal microblogs during disaster events. *IEEE Trans. Comput. Soc. Syst.* 5: 403–417

[10] Saroj A and Pal S 2020 Use of social media in crisis management: a survey. *Int. J. Disaster Risk Reduct.* 48: 1–19

[11] Madichetty S and Sridevi M 2020 Classifying informative and non-informative tweets from the twitter by adapting image features during disaster. *Multimed. Tools Appl.*, 79:1–23

[12] Madichetty S and Sridevi M 2019 Disaster damage assessment from the tweets using the combination of statistical features and informative words. *Soc. Netw. Anal. Min.* 9: 1–11

[13] Purohit H, Castillo C and Pandey R 2020 Ranking and grouping social media requests for emergency services using serviceability model. *Soc. Netw. Anal. Min.* 10: 1–17

[14] Madichetty S and Sridevi M 2017 Mining informative words from the tweets for detecting the resources during disaster. In: *Proceedings of the International Conference on Mining Intelligence and Knowledge Exploration*, pp. 348–358

[15] Purohit H, Castillo C, Diaz F, Sheth A and Meier P 2013 Emergency-relief coordination on social media: automatically matching resource requests and offers. *First Monday* 19: 1–6

[16] Madichetty S and Sridevi M 2020. Identification of medical resource tweets using majority voting-based ensemble during disaster. *Soc. Netw. Anal. Min.* 10: 1–18

[17] Basu M, Ghosh S, Jana A, Bandyopadhyay S and Singh R 2017. Resource mapping during a natural disaster: a case study on the 2015 nepal earthquake. *Int. J. Disaster Risk Reduct.* 24: 24–31

[18] Madichetty S and Sridevi M 2018 Re-ranking feature selection algorithm for detecting the availability and requirement of resources tweets during disaster. *Int. J. Comput. Intell. IoT* 1: 207–211

[19] Madisetty S and Desarkar M S 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

[20] Madisetty S and Desarkar M S 2017 An ensemble based method for predicting emotion intensity of tweets. In: *Proceedings of the International Conference on Mining Intelligence and Knowledge Exploration*. Springer, pp. 359–370

[21] Roy S, Mishra S and Matam R 2020 Classification and summarization for informative Tweets. In: *Proceedings of the 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pp. 1–4

[22] Madisetty S and Desarkar M S 2018 A neural network-based ensemble approach for spam detection in Twitter. *IEEE Trans. Comput. Soc. Syst.* 5(4): 973–984

[23] Gupta H, Jamal M S, Madisetty S and Desarkar M S 2018 A framework for real-time spam detection in Twitter. In: *Proceedings of the 2018 10th International Conference on Communication Systems & Networks (COMSNETS)*. IEEE, pp. 380–383

[24] Verma S, Vieweg S, Corvey W J, Palen L, Martin J H, Palmer M, Schram A and Anderson K M 2011 Natural language processing to the rescue? Extracting "situational

awareness" tweets during mass emergency. In: *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pp. 385–392

[25] Rudra K, Sharma A, Ganguly N and Ghosh S 2016 Characterizing communal microblogs during disaster events. In: *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 96–99

[26] Rudra K, Ganguly N, Goyal P and Ghosh S 2018 Extracting and summarizing situational information from the Twitter social media during disasters. *ACM Trans. Web* 12: 1–35

[27] Rudra K, Ghosh S, Ganguly N, Goyal P and 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*, pp. 583–592

[28] Imran M, Castillo C, Lucas J, Meier P and Vieweg S 2014 AIDR: artificial intelligence for disaster response. In: *Proceedings of the 23rd International Conference on World Wide Web*, pp. 159–162

[29] Sen A, Rudra K and Ghosh S 2015 Extracting situational awareness from microblogs during disaster events. In: *Proceedings of the 2015 7th International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–6

[30] Avvenuti M, Cresci S, Del Vigna F, Fagni T and Tesconi M 2018 rismap: a big data crisis mapping system based on damage detection and geoparsing. *Inf. Syst. Front.* 20: 993–1011

[31] Nguyen D T, Al Mannai K A, Joty S, Sajjad H, Imran M and Mitra P 2017 Robust classification of crisis-related data on social networks using convolutional neural networks. In: *Proceedings of the Eleventh International AAAI Conference on Web and Social Media*

[32] Subramani S, Wang H, Vu H Q and Li G 2018 Domestic violence crisis identification from facebook posts based on deep learning. *IEEE Access* 6: 54075–54085

[33] Alam F, Joty S and Imran M 2018 Graph based semi-supervised learning with convolution neural networks to classify crisis related tweets. In: *Proceedings of the Twelfth International AAAI Conference on Web and Social Media*

[34] Imran M, Mitra P and Castillo C 2016 witter 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 2016)*, Paris, France, May 2016. European Language Resources Association (ELRA)

[35] Kim Y 2014 *Convolutional neural networks for sentence classification*. arXiv preprint arXiv:1408.5882

[36] Palangi H, Deng L, Shen Y, Gao J, He X, Chen J, Song X and Ward R 2016 Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval. *IEEE/ACM Trans. Audio Speech Lang. Process.* 24: 694–707

[37] Sutskever I, Vinyals O and Le Q V 2014 Sequence to sequence learning with neural networks. In: *Advances in Neural Information Processing Systems*, pp. 3104–3112

[38] Johnson R and Zhang T 2016 *Supervised and semi-supervised text categorization using LSTM for region embeddings*. arXiv:preprint arXiv:1602.02373

[39] Zhou P, Qi Z, Zheng S, Xu J, Bao H and Xu B 2016 *Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling*. arXiv preprint arXiv:1611.06639

[40] Chollet F *et al* 2015 Keras. https://github.com/fchollet/keras

[41] Bojanowski P, Grave E, Joulin A and Mikolov T 2016 *Enriching Word Vectors with Subword Information*. arXiv preprint arXiv:1607.04606

[42] Grave E, Bojanowski P, Gupta P, Joulin A and Mikolov T 2018 Learning word vectors for 157 languages. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*

[43] Cresci S, Tesconi M, Cimino A and Dell'Orletta F 2015 A linguistically-driven approach to cross-event damage assessment of natural disasters from social media messages. In: *Proceedings of the 24th International Conference on World Wide Web*, pp. 1195–1200