Using Deep learning and Natural Language Processing approach to predict real disasters from tweets

Raj Kupekar
School of Computing
National College of Ireland
Dublin, Ireland
X18186432
kupekarraj@gmail.com

***Abstract*—The purpose of this research work is to use the combination of Natural Language processing (NLP) and deep learning to predict the occurrence of natural disaster from the data generated from social media platform Twitter by classifying the real and fake tweets. The deep learning model is formed from the combination of Bidirectional Long Short-Term Memory (Bi-LSTM) layer and Embedding layer which implements GloVe vector. To obtain the better results data has been cleaned where non-significant words and special characters are removed. Tokenizer module is used to convert the text into tokens in Keras library. Evaluation of the model has been done using accuracy and confusion matrix as performance metrics. Thus, the model attains the validation accuracy of .80 percent with a minimum number of misclassified records.**

Keywords— natural disaster, NLP, Bi-LSTM, tweets, GloVe, Keras, deep learning.

I. INTRODUCTION

In the recent year's social media platform Facebook and Twitter has shown tremendous growth due to human acceptance in day to day life. It has been seen that these platforms share massive information not only on health, disease, politics, science and more but also on natural disaster. Disasters creates a situation of panic among people which made them to share information on social media for emergency help, awareness, news and updates on the disaster management. Natural Disaster causes damage in the infrastructure which causes the flow of information and made a condition for the sudden help. People ask for quick help and became active on social media so that their information can spread to larger masses. These information from social media generally contains where the disaster happened, people asking for help such as food and shelter, damage on the infrastructure. Through this huge amount of data has been generated and could help in natural disaster management.

Many times, it has also been seen that information circulating on social platform like Twitter has been proved to be fake. So, to meet this requirement in our research work NLP along with deep learning has been used to predict the difference of real and fake tweets related to disaster. The aim of this research work is to predict the occurrence of natural disaster and can reach to the optimum level of accuracy to correctly classify the tweets. NLP is used to extract meaningful insights from text or speech. Similarly, NLP is used to analyse the analyse tweets that are related to describes disaster. For this research work a package of natural language tool kit (NLTK) were imported which are: stopwords . These packages are helped to remove inflectional morphological words and non-essential commonly used words such as 'is', 'the', 'and', 'an', 'a' and more. On the other hand, deep learning method Bi-LSTM has been used to maintain long term dependencies in textual data. For the architecture of deep learning model, embedding layer is implemented which used GloVe vector for preserving the semantic and syntactic relation between the words. This model has been evaluated using accuracy and confusion metric using performance metrics.

This research report is further divided into four sections which are: Literature Review- which describes the working and results received using NLP and deep learning method, Methodology- it describes methodological approach for its implementation. This section is further sub-divided into business understanding, data understand and data modelling, Evaluation- it discusses about the performance of the deep learning model and Conclusion and Future Work- research work is concluded here.

II. LITERATURE REVIEW

In today's world, social media is grabbing a lot of attention as millions of people express their thought, opinion as well as emotions which are prominently the important source of information. Considering this fact, a study in paper [1] shows twitter emotion analysis using a Turkish tweets repository. Here, three different deep neural networks are been implemented and examined. Initially, the raw tweets are pre-processed where all the irrelevant words, symbol and links are discarded from the textual data. Fixed length (F-5) and snowball steamers are used for stemming the tokenized word to their root words. Followingly, stop-words are removed from the transformed text using a Turkish stop words in NLTK toolkit. Furthermore, three different neural networks namely ANN, CNN and LSTM were built on this feature extracted text. Also, the experimental shows that LSTM network outperformed as compared to other networks where six different emotions were annotated using a lexicon based annotated approach.

Another study linking with tweeter analysis is carried out in paper [2]. Here, classification of drunk tweets is evaluated maintaining the semantic relationship between words. The data is gathered using an Amazon Mechanical Task with a help of three conditional questionnaire. During, pre-processing the posting times of tweets are extracted and accordingly, the tweets are categorized into three categories viz. morning, afternoon and night. Further, error handling tools were incorporated for eliminating the misspelled tweets, social media slangs and other irrelevant information. The natural language understanding and linked open data are used for capturing the contextual and semantic features in the textual data. Finally, the pre-processed text is pruned using a PageRank algorithm for further classification of drunk tweets. Machine learning algorithms like Naïve Bayes and SVM is implemented for decision making.

Based on a similar emotion recognition study, paper [3] focuses on maintaining the semantic and syntactic relationship between the uttered text sequences. In this paper, a Semantic-Emotion Neural Network (SENN) architecture is designed where two separate networks are trained for capturing the contextual and semantic information. As user generated textual data is used for analysis, pre-processing of data is carried out where, the redundant texts are eliminated. Similarly, a word2vec embedding vector is used for extracting the textual features which are represented using a one hot vector of n-dimensions. A Bi-LSTM network is implemented to process these extracted features which captures the contextual information whereas, a CNN network is used for extracting the semantic information. Subsequently, these unimodal networks are fused together using a late fusion technique for classifying the final emotion. The model produced an overall accuracy of 89% which is relatively higher when compared with other baseline machine learning models.

In this technological advancement, getting user feedback is an effective mechanism for improving the launched product or applications to their best. Thus, a similar study [4] of text classification is conducted on the amendment policies imposed by government for public welfares. Here, a word vectorization is used for extracting the textual features from the raw text. Prior to this, tokenization and removal of unnecessary symbols, words and links are been carried out. Accordingly, a deep CNN layer is designed for processing the extracted features and for classifying the class of the given text. In this study, two CNN layers having a filter size of 256 and kernel size of 5*5 are been designed followed by a max-pooling layer of 2*2. The model is evaluated using accuracy as a performance matrix, where the experimental results shows that the model produced an accuracy of 81% which outperformed other baseline machine learning models.

In previous study [4], only a CNN layer was used for text classification. However, similar word embedding technique is used in paper [5] for emotion classification using a combination of CNN and LSTM layer. Unlike [4], a GloVe word

embedding is used for feature extraction which converts the given text into a one hot vector of N-dimensional. Importantly, a CNN layer without any activation layer is implemented on top of these extracted features. This layer is built using two 2D CNN layer having kernel size of 5*5. Furthermore, a stacked LSTM layer is built connecting this CNN layer for maintaining the long dependencies in textual data. Here, a standard variant is used in building the LSTM model and SoftMax as an activation layer. Also, precision, recall and F-1 score are used as a performance metrics for evaluating the model. Accordingly, the CNN-LSTM model performed better where the experimental results produced the following scores of 98%, 99% and 99% respectively.

In text classification, in many instances not all the information contributes in decision making. Thus, it is a good practice to eliminate this information before giving as an input to the final layer. With this conceptual theory, paper [6] make use of an attention layer which processes only the relevant information and discards the other. Word2Vec embedding vector is used for extracting the features, where a dimension size of 32 is used for differentiating between different words. Eventually, a hybrid layer of neural network is used for getting the final classification of textual data. Initially, the vector representation is given as input to the Bi-LSTM network to maintain the long-term dependencies, followed by a CNN layer for capturing the local features in the sentences. Furthermore, the output from the CNN layer is given as an input to the attention layer. Accordingly, the model produced an overall accuracy of 87% where a drop layer with drop rate of 0.5 is included in the system architecture.

Another approach to process only the relevant information, a study in [7]make use of an attention layer which captures only the important features required in decision making. This study is carried out on several datasets having multiclass labels. A fast text label embedding technique is used for obtaining the vector representation of the textual data. These features are given as an input to a Bi-LSTM layer followed by an attention layer. subsequently, another network is trained using a graphical attention network which, are then concatenated using a dot product for further classification. Thus, this approach yielding an overall accuracy of 91%.

As neural networks exhibit some exceptionally good properties while dealing with sequential data, a study in [8] presents a new framework comprising of Tensor graph convolutional networks for text classification. After extracting the textual features, two different types of aggregating methods are adopted. Firstly, an intra-graph propagation is used for aggregating the information from several nodes into a single node. On the other hand, an inter-graph propagation is used for harmonizing heterogenous information between different graphs. Accordingly, the model produced an accuracy of 87% where accuracy is used as a performance metrics. Natural disaster can cause significant cause to the environment and can threatens human life. The happening of the disaster can cause unprecedented change in the human life but prediction of this disaster with the help of advancement of technology can minimize the damage. Similarly, in the study [9] has developed a hybrid model for the prediction of disasters by using machine learning techniques and data mining approach. This research work has used weather data for building flood type disaster occurrence model. The model used Time Series for predicting attributes and RMSE is used for predicting the measures.

In year 2015 Tamil Nadu has faced floods [10] a hash tag trend has been followed on Twitter. Based on these activities on social media author has implemented Naïve Bayesian and SSVM classification to identify the severity of disaster. This method also helps in to detect the affected area. The approach generates an accuracy of 89 percent on real time geo-parsed tweets. In another paper, [11] machine learning and NLP has been used to monitor the tweets which were updated during natural disaster. The model implemented LibLinear classifier to extract actionable tweets from large amount of raw tweet. This model has been tested on Myanmar_Earthquake_2016 dataset which attains an accuracy of 75 percent. Traffic incidents causes traffic congestion and by predicting these incidents can help travellers to plan their trip with more ease. In the recent study [12], Twitter data has been used to predict the accidents, traffic, natural disaster. The approach classifies tweets based on geolocation and uses NER and entity disambiguation and the experiment were tested on West Midlands are, United Kingdom for the real time data generation.

Predicting the geographical location of the disaster is an important factor for disaster prevention and disaster monitoring. In one such study [13], deep learning landslide recognition method has been proposed based on optical remote sensing images.

The model has multiple hidden layers and SoftMax classifier is used for class prediction. This deep learning model achieves state-of-the-art result in efficiency and accuracy. In another study [14] an algorithm has been proposed for flood disaster management. The algorithm categorizes tweets from high to low priority based on used location mentioned in the tweet by using Markov model. The system attains a classification accuracy of 81 percent and prediction accuracy of 87 percent.

III. DATA MINING METHODOLOGY

The [1] aim of this study is to do predictions and not to find any patterns, that is the key reason CRISP-DM methodology is considered for this study [15] which is very apt if the goal of the project is to perform predictions. The entire process is organized in the 6 phases as seen in the below figure.

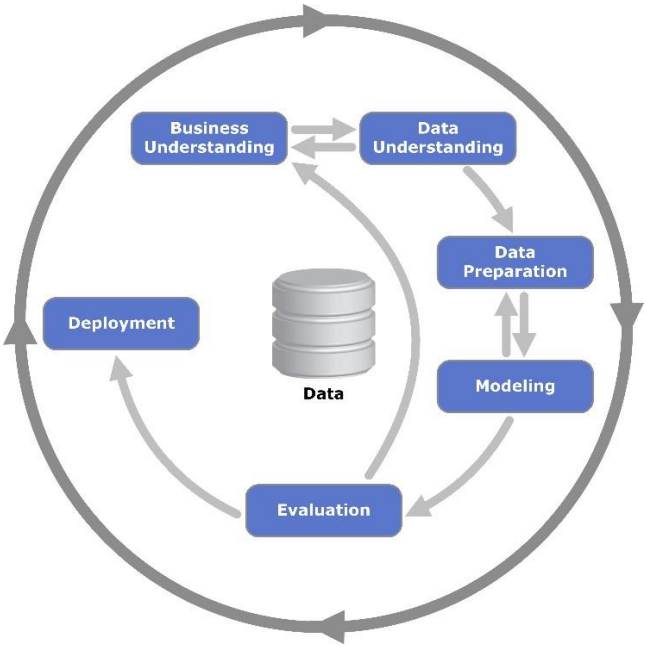

Fig. 1: CRISP-DM

1. Business understanding:

This is the primary phase that focuses on the business objective of the project. A good domain expertise along with the knowledge of previous research done in this domain is needed to come up with the correct business objective and then convert it in the data mining problem. This can be broken into 2 subcategories'

1.1. Objective setting

In this study one of the main business objective is to achieve the optimum level of accuracy in predicting the disaster from tweets.

1.2. Generate Project Plan

This stage focuses on the plan for achieving the business objectives and describes the selection of data mining tools and techniques to be used in the project. The model in this study is built in python using the Keras deep learning library with TensorFlow at the backend. The model development, training and testing is done on open source free cloud service Colab offered by Google. Seaborn, matplotlib and wordcloud packages are used to plot the graphs that can help in understanding data. The nltk is used to remove the stop words from the text.

2. Data understanding:

The Dataset used in this study is downloaded from the figure-eight website [16]. It contains 10,000 tweets already split into 7613 as training set and 2387 as test set with information such as keyword, location, text, id and target variable. Out of the total tweets, only 7613 observations are used as records in test.csv are not labelled with binary values 0, depicting non-disaster and 1, indicating disaster. In the current study the dataset (only the labelled records 7613 in training set i.e. train.csv) is split into the ratio of 80:20. The training set now contains 6090 observations and testing set consists of 1523 observations. The keyword column and location column have 61 and 2533 null values respectively. As missing values in both columns cannot be filled with any alternate values they are left as is.

The Figure 2 shows the distribution target values across the dataset. The target label value of 0, which means presence of disaster, accounts for 57% of the overall data whereas label value of 0, which means absence of disaster, accounts for the 43% of the data. It is evident for the bar plot that output classes are

well balanced for further analysis and data modelling.

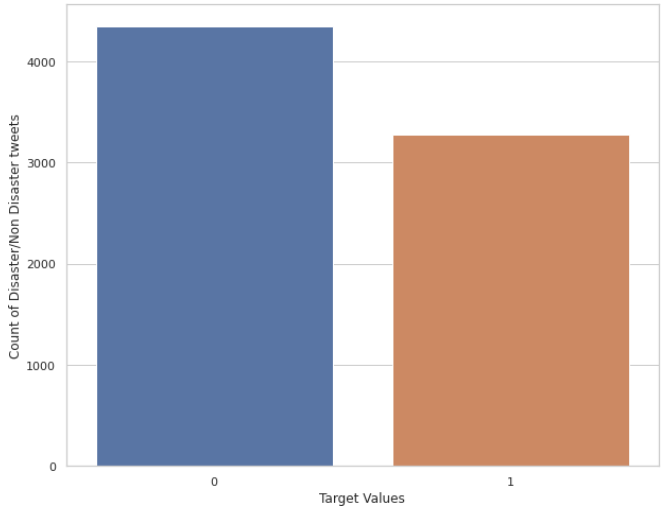

Fig. 2 : Output category Vs Frequency

The distribution plot in Figure 3 and Figure 4 shows the word length in each of the tweets that are related to disaster and non-disaster respectively. Based on these plots maximum length parameter that is passed for padding sequence method is decided.

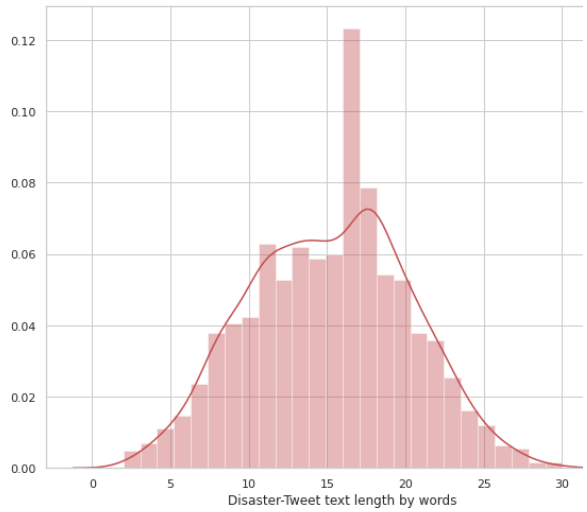

Fig. 3 : Word length in Disaster tweets

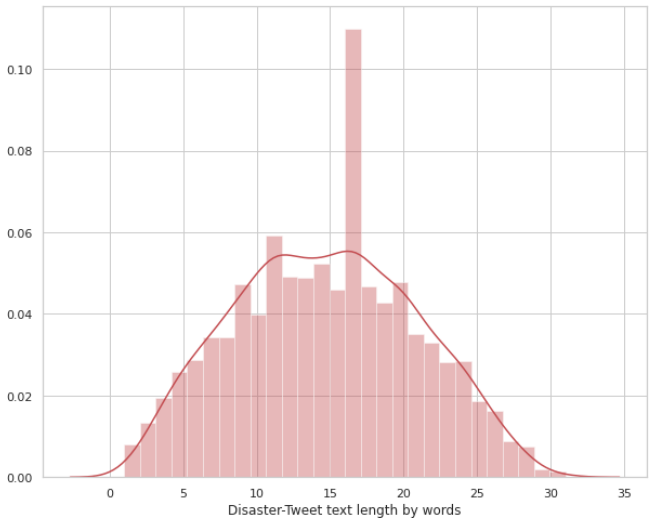

Fig. 4 : Word length in non-Disaster tweets

To understand the important keywords, basically those words that are used most frequently or repeated several times in the keyword feature, barplot is generated. The barplot in the Figure 5 shows the frequency of the top 30 keyword in the dataset.

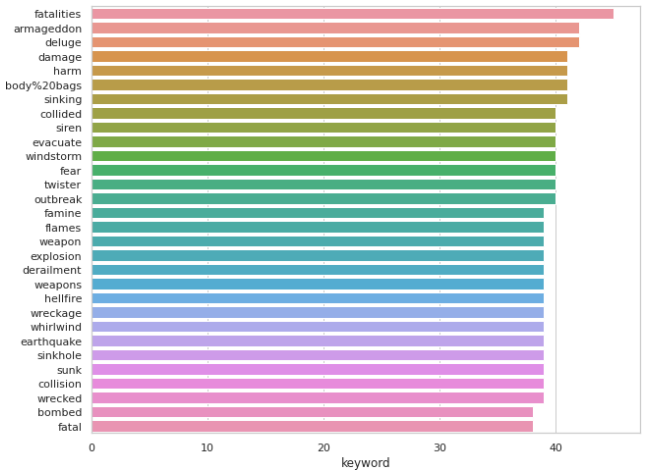

Fig. 5 : Frequency of Keywords

3. Data Preparation:

This stage of the project is very important as here it is decided how to convert raw data into the proper form that can be processed by the model resulting in the optimum level of accuracy. The data used in this study is in the raw text form extracted from twitter and need lots of cleaning and processing before it is fed to algorithm.

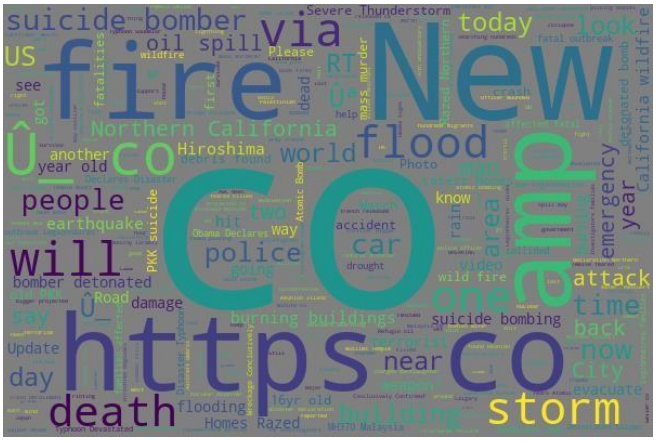

Fig.6 : Word cloud for Disaster tweets

The Figure 6 and Figure 7 shows the word cloud for tweets that are related to disaster and non-disaster respectively. From both the word clouds it is evident that there are several tweets beginning with word "***https***" but of less significance to be used in training the model. Also, there is another word ***"amp",*** of no importance . Thus, it is crucial to remove both these words from the text feature variable that contains the tweet data.

Fig.7 : Word cloud for non-Disaster tweets

In addition, there are some special characters like @,&,$, ! etc. and numbers as a part of the address in tweet, need to be cleaned from the text. All the data cleaning is done using regular expression library in python.

In the next step the sentences in the text column are converted into the tokens using Tokenizer module from Keras library. The *fit on texts* method of the tokenizer instance takes in the data and encodes it.

In this study, since there are 16,070 distinct words in text feature, tokenizer instance is initialized with the hyperparameter num_words as 16,070 to tokenize these many words. A word index property of the of the tokenizer returns a dictionary containing key value pairs for 16070 words where key is the word and value is the token for that word. Furthermore, sentences in the tweets are converted into the sequence of integers using this word-token dictionary map, replacing the words with token. For e.g. sentence "forest fire near la ronge sask canada" is encoded as [73, 3, 120, 571, 5475, 5476, 1181]. As in the Figure 3 and 4 length of the tweets vary due to which resulting sequence of integers have different length. It is essential to covert each of the sequences into fixed length else it becomes hard to train a neural network. This is achieved by padding zeroes to the tweets that are less than the maximum tweet length which is set as 25 in this case based on the distribution plots.

4. Data Modelling:

This stage is key to the design, development, hyperparameter tuning, training and testing of the model. The motivation to use Neural Network + words embeddings in this study is gained from [17, 18]. According to authors of the study, the ability to integrate with ease the pre-trained word embeddings and the inherent non-linearity of the network both lead to superior classification accuracy.

- **Embedding Layer**

This study implements GloVe embeddings, pre-trained word embeddings into the Keras Embedding layer and uses to train the classifier on the tweet's dataset. GloVe embedding is sourced from [19] and the 100-dimensional version is used in this paper, therefore the output dimension is set to 100. Keras Embedding layer is seeded with GloVe word embedding weights. The initial implementation of the model used Keras Embedding layer API to learn the embedding, but the new transfer learning approach significantly improved the validation accuracy by 8% which will be discussed in evaluation section in detail. Figure 8 shows the architecture of the deep neural network used in this study for disaster prediction.

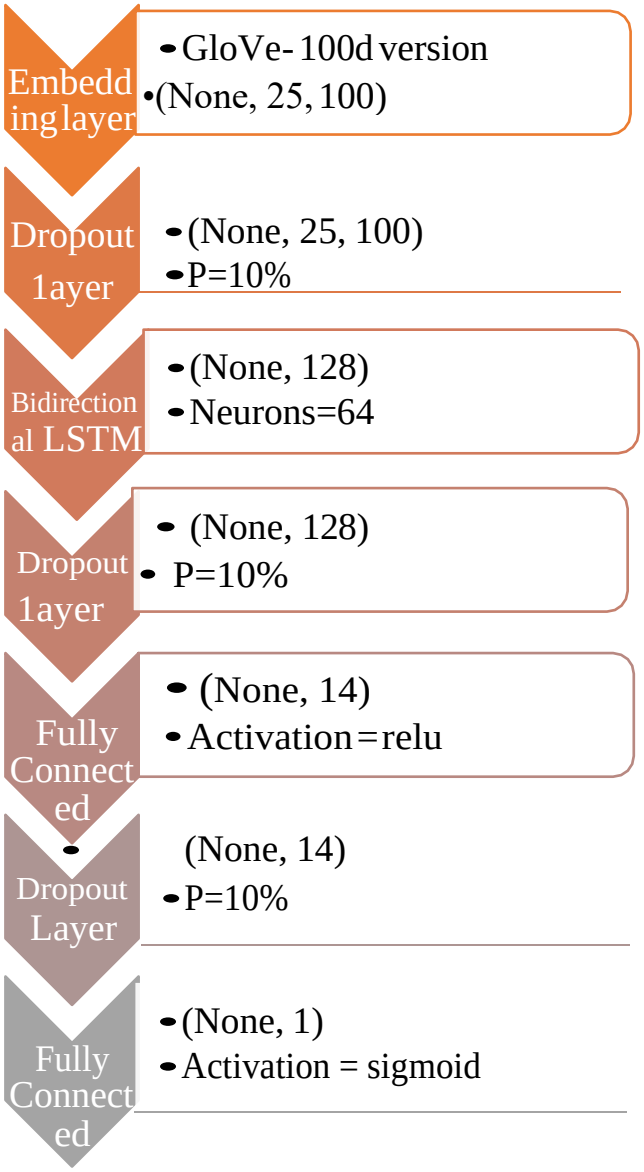

Fig. 8 : Architecture of Disaster prediction deep learning model

- **Bidirectional Long Short-Term Memory (LSTM) layer**

Bidirectional LSTM layer forms the core of the architecture in this study. Bidirectional LSTMs are more effective than LSTMs [20]. Using Bidirectional LSTMs (over LSTMs) have the effect of allowing LSTMs to learn the problem faster [21]. Bidirectional LSTM have been proposed for text classification, as acknowledge in several works [22, 23]. In the BLSTM layer 64 hidden neurons are used selected based trial and error method as it achieved good accuracy and validation accuracy.

- **Dropout layer**

To deal with the overfitting problem dropout technique is used [24]. To avoid overfitting problem dropout layer is added before and after the Bidirectional layer and not between recurrent connections. Neurons should be randomly dropped out after or before the BLSTM and not inter-BLSTM [25]. One dropout layer is added after and before BLSTM layer and after fully connected layer. The dropout rate in each of the dropout layer is set to 10%, meaning, 1 in 10 units are excluded randomly from each weight update cycle.

- **Fully Connected layer**

Two Dense layers are added at the end of the model in order to introduce non-linearity in the model. First dense layer accepts 14 units as output shape with ReLU as activation function. The hyperbolic tangent and sigmoid functions both suffer from the vanishing gradient problem and slows down the learning resulting in slow convergence [26] [27]. Whereas, ReLU activation function is 9 times faster than hyperbolic tangent and suffer vanishing less from gradient problem.
Lastly, since this is a classification problem, second and last layer is Dense output layer with one neuron and an activation function used is sigmoid to make either 0 or 1 predictions for the two classes (disaster or no disaster) in the problem.

- **Loss Function**

Since the classification problem in the study is a binary classification task, log loss is used as the cost function (binary cross entropy in Keras library) [28, 29] .

IV. EVALUATION

In this section, model evaluation using various performance metrics is carried out. As GloVe embedding vector is used over Word2vec the local statistics as well as global statistics information are maintained while generating the word vectors. The model is evaluated using accuracy and confusion matrix as performance metrics. The model is trained on training data while, evaluated using a validation dataset using an epoch size of 100. The following figures depicts the model results for the last five epochs and the graphical view of accuracy and validation accuracy.

```
loss: 0.0559 - accuracy: 0.9764 - val_loss: 1.5531 - val_accuracy: 0.7925
loss: 0.0515 - accuracy: 0.9770 - val_loss: 1.5674 - val_accuracy: 0.7807
loss: 0.0438 - accuracy: 0.9785 - val_loss: 1.4924 - val_accuracy: 0.7853
loss: 0.0460 - accuracy: 0.9788 - val_loss: 1.5628 - val_accuracy: 0.7905
loss: 0.0404 - accuracy: 0.9796 - val_loss: 1.7647 - val_accuracy: 0.7820
```

Fig. 9 : Last five epochs featuring accuracy and validation accuracy

The model is trained using two different methods, where initially the model is trained without any embedding layer and subsequently trained using a GloVe embedding layer such that the effect of these models can be compared. Accordingly, the model is evaluated using confusion matrix and accuracy.
Primarily, the model without any embedding layer produced an overall validation accuracy of approximately 0.72 percent. Also, it is observed that the actual model accuracy tends to be constant after certain epochs. Thus, this accuracy is boosted by including an embedding layer. The following figure shows the accuracy graph for model without any embedding layer.

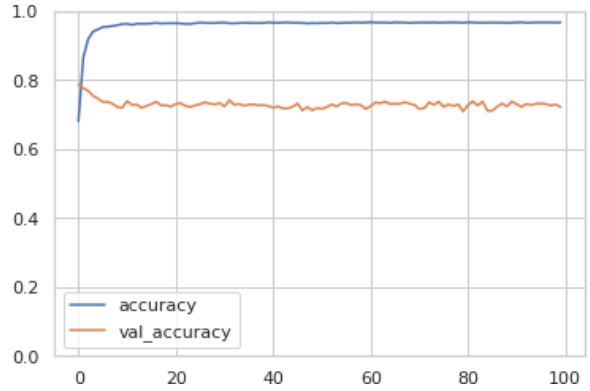

Fig. 10(a) : Accuracy of model without embedding layer

Furthermore, a GloVe embedding layer is inserted while building a neural network. Experimental results show that using an embedding layer, the model accuracy is improved by around 0.8 percent. In figure 3(b), it is seen that the model accuracy remains consistent after a certain number of epochs and it also approaches towards the value 1.0, at the end of the specified number of epochs. On the other hand, the accuracy for validation set remains constant at 0.80 which is comparatively low when compared with the model accuracy on training data. Thus, validation accuracy is considered while evaluating the efficiency of the model.

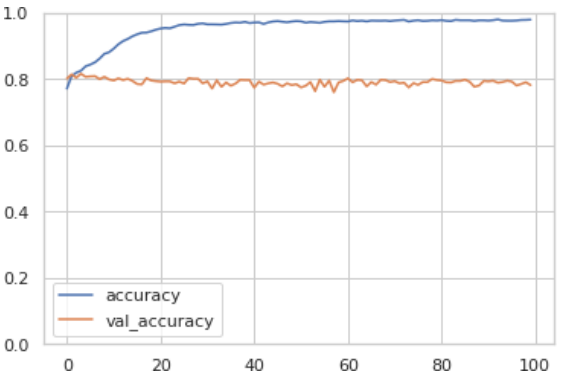

Fig. 10(b) : Accuracy of model using a GloVe embedding

Additionally, the system is also studied with the effect of variations in the epoch sizes for training the model. An epoch size of 50, 100 and 150 was used for training. In case of 50 epoch size, the model yielded a poor accuracy of 0.67 percent. After this, the model was trained using an epoch size of 100 and 150. Accordingly, for both the epoch sizes the model generated almost similar validation accuracy. Thus, to get the optimal model with a minimum time complexity epoch size of 100 is chosen.
Confusion matrix is drawn to get correct prediction score. It is observed that around 69.5 percent true positive class are predicted by the system with approximately, 85.7 percent of the total true negative class. Accordingly, out of the total observations 332 instances are misclassified by the model. Thus, the implemented model produced an optimal accuracy of 0.80 with a minimum number of misclassified records.

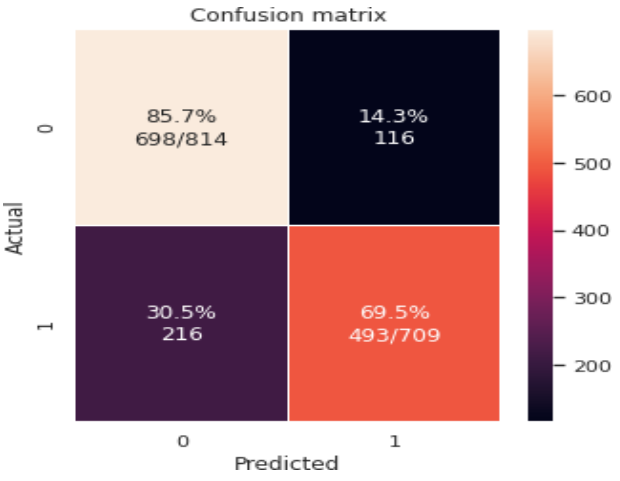

Fig. 10(c) : Confusion Matrix

V. Conclusion and Future Work

In this study, classification of disastrous tweets is been carried out using an LSTM layer. A GloVe embedding vector is used as this embedding technique fits well with the out of vocab words. Similarly, the global and local information are also maintained in the embedded word vector. In this manner, the semantic and syntactic relationship between the words are preserved by the model. In model building, a Bi-LSTM network is implemented over CNN layer, which maintains the long-term dependencies in the textual data. Additionally, a ReLu activation function is used for activating the neural network layer except the output layer which prevents the issue related to the vanishing gradients. Particularly, an Adam optimizer is used, having combined advantages of Adagrad and RMSprop optimizer thus, enabling to deal with problems associated with sparse gradients and on-line settings. Hence, with an objective of correctly classify the disastrous tweets, the implemented model performed well attaining a highest accuracy of 0.80. In near feature, a hybrid combination of Bi-LSTM and CNN layer can be implemented to get a more precise model in predicting the disastrous tweets. Similarly, the model can be evaluated by implementing different word embedding techniques and analyzing their effect to get an optimal model.

References

[1] J. ZHENG and L. ZHEN, "A Hybrid Bidirectional Recurrent Convolutional Neural Network Attention-Based Model for Text Classification," *IEEE,* vol. 6, no. 1, p. 9, 2019.

[2] M. TOCOGLU, O. OZTURKMENOGLU and A. ALPKOCAK, "Emotion Analysis From Turkish Tweets," *IEEE,* vol. 5, p. 6, 2019.

[3] M. Grzeça, K. Becker and R. Galante, "Improving the Classification of Drunk Texting in Tweets Using Semantic Enrichment," *International Conference on Web Intelligence,* vol. 1, p. 8, 2018.

[4] E. Batbaatar, M. Li and K. H. Ryu, "Semantic-Emotion Neural Network for Emotion Recognition from Text," *IEEE Access,* vol. 7, pp. 111866-111878, 2019.

[5] S. Zhang, Y. Chen, X. Huang and Y. Cai, "Text Classification of Public Feedbacks using Convolutional Neural Network Based on Differential Evolution Algorithm," *International Journal of Computers Communications & Control,* vol. 14, no. 1, pp. 124-134, 2019.

[6] L. Y and L. S., "Research on Text Classification Based on CNN and LSTM," *IEEE International Conference on Artificial Intelligence and Computer Applications,* pp. 352-355, 2019.

[7] X. Liu, X. You, X. Zhang and J. Wu, "Tensor Graph Convolutional Networks for Text Classification," *IEEE,* 2020.

[8] A. Pal, M. Selvakumar and M. Sankarasubbu, "Multi-Label Text Classification using Attention-based Graph".

[9] H. Thilakarathne, "Developing a Hybrid Model For Disaster Prediction using Machine Learning with Artificial Neural Networks & Data Mining Approach," 2016.

[10] B. Anbalagan and C. Valliyammai, "#ChennaiFloods: Leveraging Human and Machine Learning for Crisis Mapping during Disasters Using Social Media," *2016 IEEE 23rd International Conference on High Performance Computing Workshops (HiPCW),* pp. 50-59, 2016.

[11] S. S. M. Win and T. N. Aung, "Target oriented tweets monitoring system during natural disasters," *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS),* pp. 143-148, 2017.

[12] A. Salas, P. Georgakis, C. Nwagboso, A. Ammari and I. Petalas, "Traffic event detection framework using social media," *2017 IEEE International Conference on Smart Grid and Smart Cities (ICSGSC),* pp. 303-307, 2017.

[13] L. Ying and W. Linzhi, "Geological Disaster Recognition on Optical Remote Sensing Images Using Deep Learning," *Information Technology and Quantitative Management (ITQM 2016)* , p. 566 – 575 , 2016.

[14] J. P. Singh, Y. K. Dwivedi, N. P. Rana, A. Kumar and K. K. Kapoor, "Event classification and location prediction from tweets during disasters," *Annals of Operations Research,* vol. 283, p. 737–757 , 2019.

[15] "CRISP-DM – a Standard Methodology to Ensure a Good Outcome," Datasciencecentral.com, 2020. [Online]. Available: https://www.datasciencecentral.com/profiles/blogs/crisp-dm-a-standard-methodology-to-ensure-a-good-outcome.

[16] "Figure Eight," [Online]. Available: https://www.figure-eight.com/data-for-everyone/.

[17] Y. Goldberg, "A Primer on Neural Network Models for Natural Language Processing," *Journal of Artificial Intelligence Research,* vol. 57, pp. 345-420.

[18] Y. Goldberg and G. Hirst, Neural Network Methods in Natural Language Processing, San Rafael: Morgan & Claypool Publishers, 2017.

[19] "http://nlp.stanford.edu/projects/glove/," [Online]. Available: http://nlp.stanford.edu/data/glove.6B.zip.

[20] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks,* vol. 18, no. 5-6, pp. 602-610, 2005.

[21] "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing,* vol. 45, no. 11, pp. 2673-2681, 1997.

[22] S. Misawa, M. Taniguchi, Y. Miura and T. Ohkuma, "Character-based Bidirectional LSTM-CRF with words and characters for Japanese Named Entity Recognition," *Proceedings of the 1st Workshop on Subword and Character Level Models in NLP,* pp. 97-102, 2018.

[23] B. Lin, F. Xu, Z. Luo and K. Zhu, "Multi-channel BiLSTM-CRF Model for Emerging Named Entity Recognition in Social Media," *Proceedings of the 3rd Workshop on Noisy User-generated Text,* pp. 160-165, 2018.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from," *Journal of Machine Learning Research,* vol. 15, pp. 1929-1958, 2014.

[25] W. Zaremba, I. Sutskever and O. Vinyals, "Recurrent Neural Network Regularization," 2014.

[26] D. Sussillo and L. Abbott, "Random Walk Initialization for Training Very Deep Feedforward Networks," *Neural and Evolutionary Computing,* 2014.

[27] Y. Bengio and X. Glorot, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research,* vol. 9, 2010.

[28] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, Cambridge, Mass: The MIT Press., 2017.

[29] K. Janocha and W. Czarnecki, "On Loss Functions for Deep Neural Networks in Classification," 2017.