A Comparative Study of Different Neural Network Models for Disaster-Related Tweet Classification

Jaydeep Dharamsey[#1], Raj Manoj Dedhia[#2]

School of Computer Science,

University of Windsor, Ontario, Canada

[1]dharamsj@uwindsor.ca

[2]dedhiar@uwindsor.ca

Abstract- In this paper, we compare the performance of several neural network architectures on a dataset of Twitter messages related to natural disasters. We focus on four architectures: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Bidirectional Gated Recurrent Unit (BiGRU), and Bidirectional Long Short-Term Memory (BiLSTM) networks. We use a range of performance metrics to evaluate the algorithms, including precision, recall, and F1 score. Our results show that BiGRU networks outperform the other architectures on this task, with a precision score of 0.85 and F1 score of 0.75 on average.

INTRODUCTION

Natural disasters often result in a large amount of damage and loss of life. In recent years, social media platforms such as Twitter have become important channels for people to share information about disasters and their effects. This has led to the development of machine learning algorithms for analyzing Twitter data to better understand and respond to disasters.

One class of algorithms that has been particularly successful in this domain is neural networks, which are a type of machine learning algorithm that is inspired by the structure and function of the human brain. Neural networks can learn to extract key features from data and make predictions based on those features.

We evaluate the performance of four neural network architectures on this task: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Bidirectional Gated Recurrent Unit (BiGRU), and Bidirectional Long Short-Term Memory (BiLSTM). We use the following performance metrics to evaluate the algorithms:

Precision: the proportion of predicted positive instances that are actually positive

Recall: the proportion of actual positive instances that are predicted to be positive

F1 score: the harmonic mean of precision and recall

Accuracy: the proportion of correctly classified instances

RELATED WORK

There has been a significant amount of research on the use of neural networks for disaster analysis on Twitter. Some previous studies have compared different architectures

on this task, but they often focus on a small number of algorithms and do not always use the same performance metrics.

In [1], the authors compare the performance of feedforward networks, CNNs, and RNNs on a dataset of Twitter messages related to earthquakes. They evaluate the algorithms using accuracy and F1 score and find that the feedforward network outperforms the other architectures.

In [2], the authors compare the performance of multiple architectures, including LSTMs (Long Short-Term Memory), on a dataset of Twitter messages related to hurricanes. They use precision, recall, and F1 score as their evaluation metrics, and find that the LSTM (Long Short-Term Memory) outperforms the other architectures.

Our study extends this previous work by comparing a broader range of architectures on a larger dataset, using a range of performance metrics to evaluate the algorithms.

METHODOLOGY

A. Data Description

The dataset is taken from kaggle.com Competition titled “Natural Language Processing with Disaster Tweets” which was originally created by Appen.com. The "Natural Language Processing with Disaster Tweets" dataset is a collection of tweets related to natural disasters, such as earthquakes and hurricanes. The messages have been annotated with labels indicating whether they are relevant to a disaster (i.e., whether they contain information about a disaster or its effects). This dataset is commonly used for research in natural language processing, specifically for developing and evaluating machine learning models that can automatically classify tweets as relevant to a disaster.

The dataset includes 7613 tweets, with 3271 labeled as relevant and 4342 labeled as irrelevant. The tweets are in English and contains URLs, hashtags, and user mentions. The dataset also includes additional information about the tweets, such as the date and time they were posted and the location from which they were sent.

Researchers can use this dataset to train and evaluate machine learning models that can automatically classify disaster-related tweets. This can help emergency responders and organizations quickly identify and access relevant information during a disaster, improving response time and efficiency.

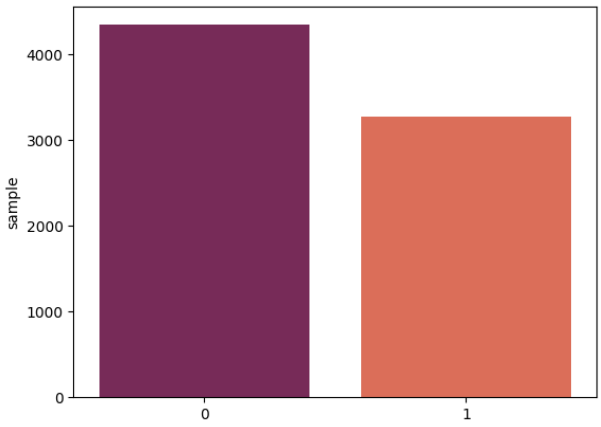


Fig 1- Count of disaster and non-disaster related tweets

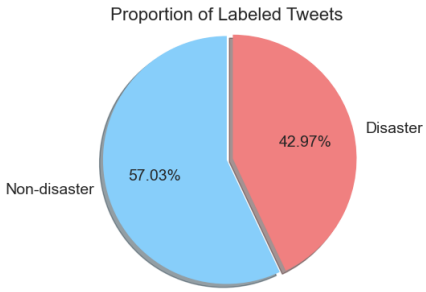


Fig 2- Proportion of disaster and non-disaster tweets

B. Exploratory Data Analysis and Pre-processing

We discovered 57.03% of non-disaster tweets and 42.97% of catastrophe tweets after analyzing the dataset. Even yet, the data was not in a pure format because the tweets had a lot of hashtags and URLs.

First, stop words were removed from the data, and slang terms were replaced with more formal terminology. Nouns and pronouns were other elements that had no impact on our findings when the en_core_web_sm dataset from spacy library was used. Everything was converted to lowercase to get the final dataset, and the Keras tokenizer was used to tokenize the data.

A Word cloud of disaster and non-disaster tweets is generated to understand the frequently used words in the tweets.

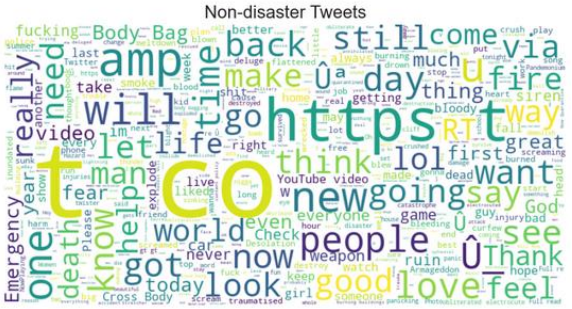


Fig 3- Word Cloud of Non-disaster tweets

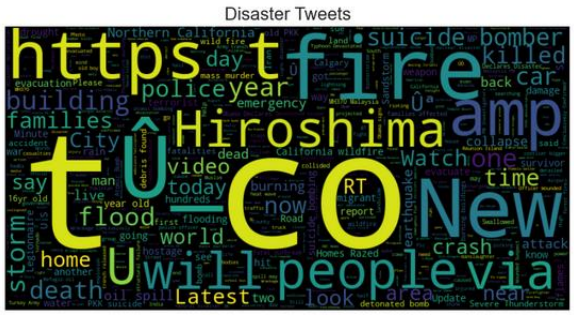


Fig 4- Word Cloud of disaster tweets

As we can see, a lot of words are not related to any disaster like t.co, people, will etc. These stop words are unimportant for the training of the model and need to be removed. nltk(Natural Language Toolkit) is a collection of word libraries like stop words, punctuations, etc. After the removal of stop words, we can see that the most frequently used words are fatalities, deluge, armageddon and sinking. These words are more related to disasters.

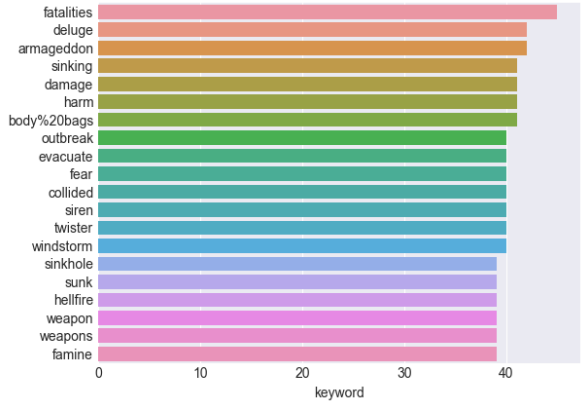


Fig 5- Most frequently used words in disaster tweets

Generating the probability density of disaster tweets vs non-disaster tweets rates interesting patterns of data such as the average word length of disaster tweets being higher than non-disaster tweets. Also, the punctuation percentage is more centrally concentrated in disaster tweets compared to non-disaster tweets. Disaster tweets also have a larger noun count compared to non-disaster tweets. Disaster tweets also have higher tweet character count compared to non-disaster tweets.

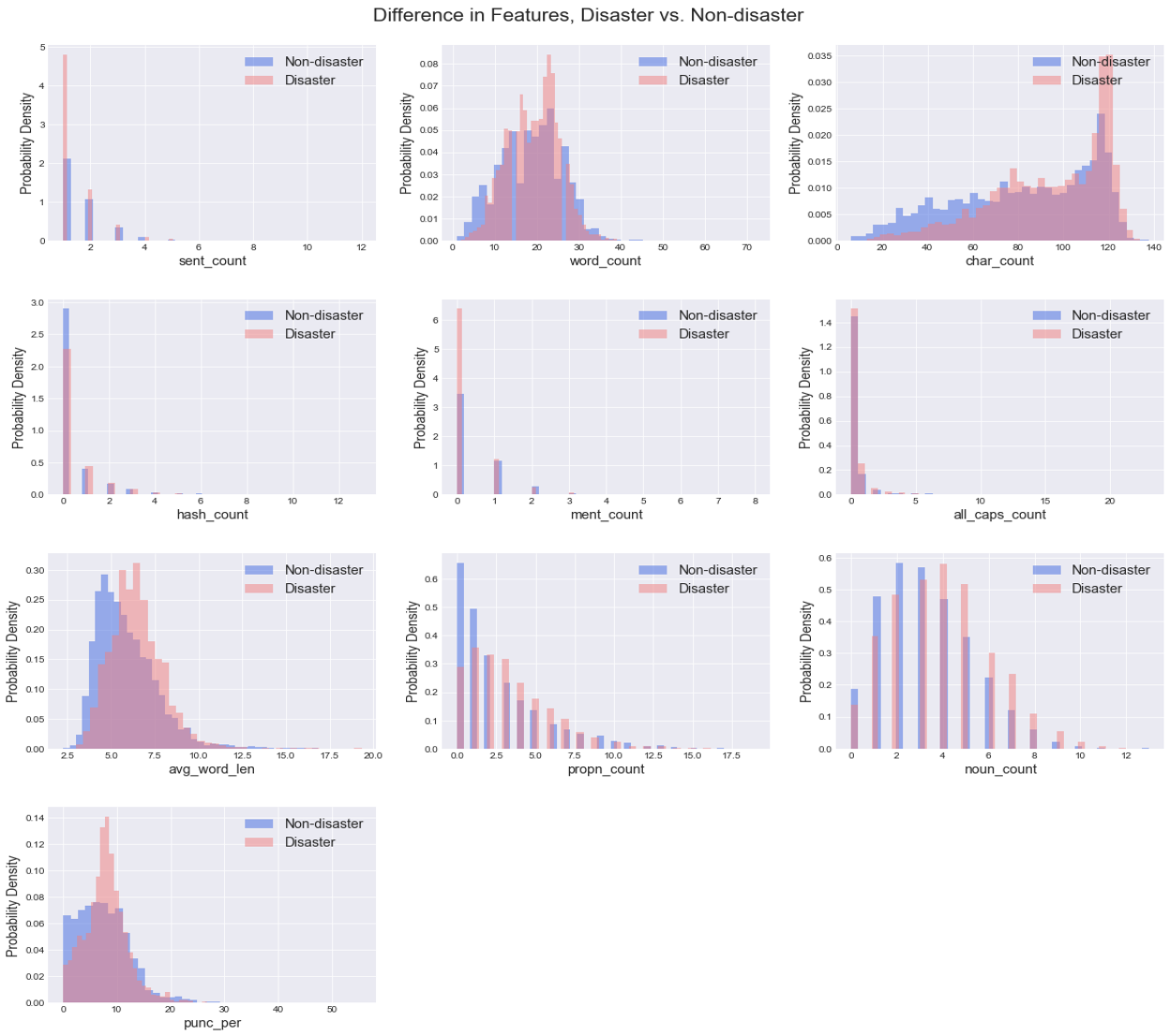


Fig 6- Distinctive features of disaster tweets vs non-disaster tweets

C. Global Vector Generation (GloVe)

Word Representation Using Global Vectors GloVe is a technique for producing vector representations of words using unsupervised learning. Training is done using corpus-based global word-word co-occurrence statistics, and the resultant representations show off some of the word vector space's fascinating linear substructures. GloVe vectors are produced using a context window and a word co-occurrence matrix. In GloVe, word vectors are a type of word representation that connects a machine's knowledge of language to that of a person. They have mastered text representations in an n-dimensional space, where words with the same meaning are represented similarly. This means that two related words are represented by very closely spaced, identical vectors.

D. Models

1. **CNN-**

A typical deep learning neural network type for applications like sentiment analysis and text categorization is the convolutional neural network (CNN). To identify certain patterns or features in the input data, the

convolutional layer employs several filters. By using a summarizing function, the pooling layer decreases the dimensionality of the feature maps. Convolutional and pooling layers' learnt features are combined in the fully connected layer, which uses the input data to produce predictions.

This model is a deep learning neural network for natural language processing tasks. It consists of multiple layers, including an embedding layer, a global max pooling layer, a batch normalization layer, and multiple dense layers. The input to the model is a sequence of words or tokens, which are embedded into a high-dimensional space by the embedding layer. The bidirectional recurrent layer processes the input data in both forward and backward directions, allowing the model to capture long-range dependencies in the input data. The global max pooling layer and the batch normalization layer help to reduce the dimensionality of the data and improve the model's generalization ability. The dense layers make predictions based on the input data. This model has 2,311,357 trainable and non-trainable parameters.

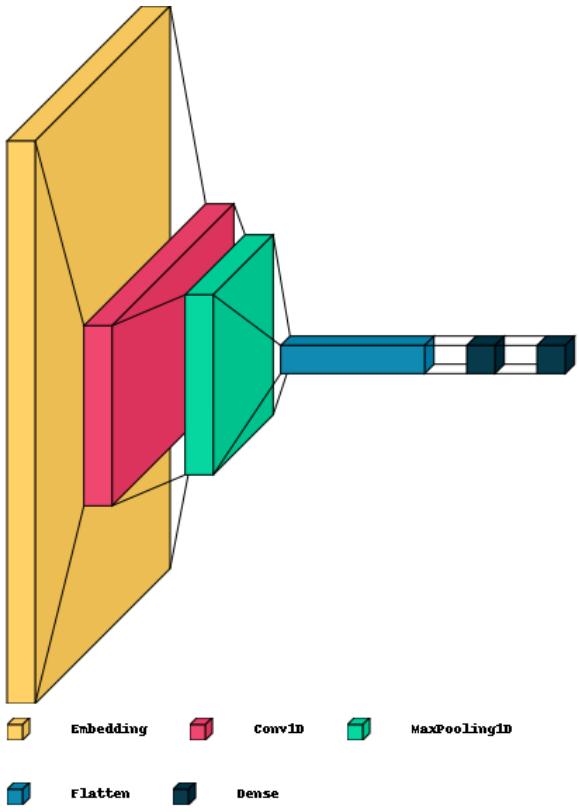


Fig 7- Keras visualization for CNN model used for evaluation

2. **RNN** -

An example of a deep learning neural network is a recurrent neural network (RNN), which is frequently used for text creation and other natural language processing applications. RNNs process the input data in a sequence as opposed to standard feedforward neural networks, which do it in a single pass, enabling them to simulate the temporal connections between the input data.

For applications like language modelling, the RNN's output can either be a vector representation of the input sequence or a series of predicted words (e.g., for text generation tasks). The kind of activation function utilized in the recurrent units affects the output. For instance, the output of a SoftMax activation function may be used to

sample the following word in a sequence since it produces a probability distribution across the vocabulary terms.

Deep learning neural networks are used in this approach to analyze natural language. It is made up of several layers, such as an embedding layer, a bidirectional recurrent layer, a global max pooling layer, a batch normalization layer, and several dense layers. A list of words or tokens that are embedded by the embedding layer into a high-dimensional space serves as the model's input. The model can recognize long-range relationships in the input data because the bidirectional recurrent layer analyses the input data in both forward and backward directions. The batch normalization layer and the global max pooling layer contribute to reducing the dimensionality of the data and enhancing the generalizability of the model. Using the incoming data as a foundation, the thick layers produce predictions. There are 2,361,613 trainable and untrainable parameters in this model.

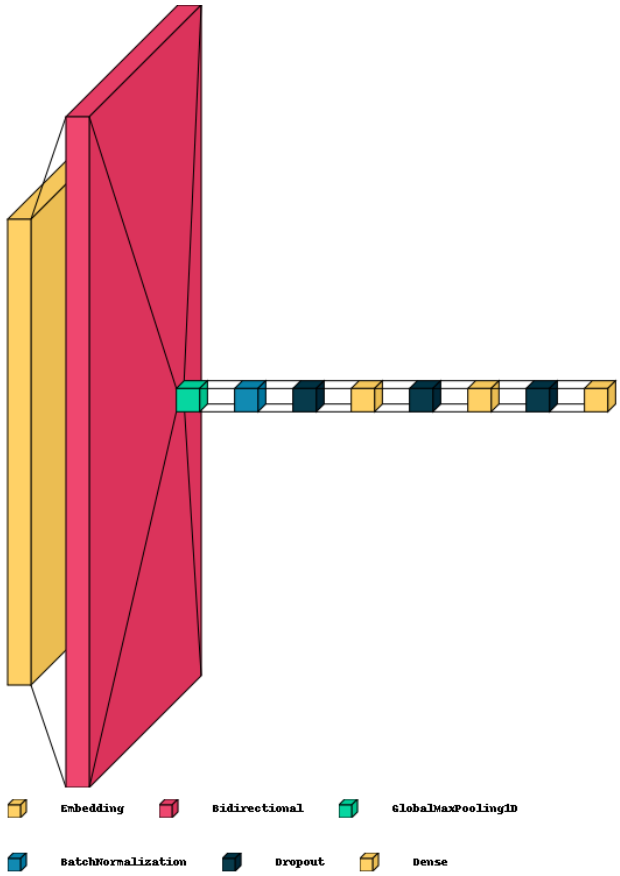


Fig 8- Keras visualization for RNN model used for evaluation

3. Bidirectional GRU (Gated Recurrent Unit)

Recurrent neural networks (RNNs) of the Bidirectional Gated Recurrent Unit (GRU) variety are frequently employed for natural language processing tasks including language modelling and text categorization. Bidirectional GRUs (Gated Recurrent Unit) analyze the input data in both forward and backward directions, in contrast to conventional RNNs, which only process the input data in one direction. This enables them to recognize long-range relationships in the input data.

For language modelling tasks, for example, the output of the bidirectional GRU can be a vector representation of the input sequence,

or it can be a prediction based on the input sequence (e.g., for text classification tasks). The network's output layer and the kind of activation function employed in recurrent units define the output. A probability distribution across the classes in the classification job will be the output, for instance, if a SoftMax activation function is employed in the output layer. This probability distribution may then be utilized to create a prediction.

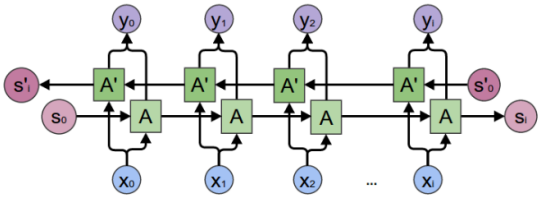


Fig 9- Basic Structure of BiGRU model

4. Bidirectional LSTM -

When it comes to natural language processing tasks like language modelling and text classification, recurrent neural networks (RNNs) of the Bidirectional Long Short-Term Memory (LSTM) network type are frequently employed. Bidirectional LSTMs analyze the input data in both forward and backward directions, in contrast to conventional RNNs, which only process the input data in one direction. This enables them to capture long-range relationships in the input data.

As a result of its capacity to identify long-range relationships in the input data and derive predictions based on those correlations, bidirectional LSTMs are a potent tool for natural language processing jobs. They have been effectively used for various applications, including named entity recognition, text categorization, and language modelling.

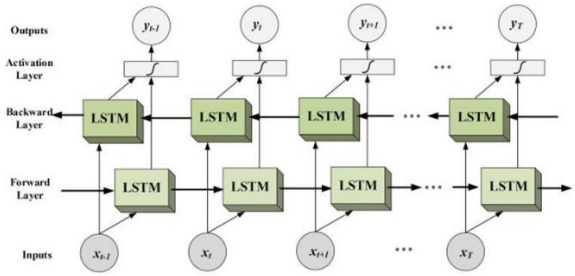


Fig 10- Basic Structure of BiLSTM model

E. Training and Tuning

The process of choosing a neural network's ideal collection of hyperparameters is known as "hyperparameter tuning." The learning rate, the quantity of hidden units, and the regularization strength are examples of hyperparameters—values that are not learnt during training. Finding the ideal parameters for a given job is crucial for getting accurate results since these variables have a substantial influence on how well the network performs.

Neural networks' effectiveness in natural language processing can be influenced by a variety of hyperparameters. The quantity of the vocabulary, the dimensionality of the word embeddings, the kind of recurrent unit, and the kind of activation function are a few typical hyperparameters. It can be difficult to tune these hyperparameters since there are many value combinations to test and because the best values rely on the job and the data.

To calculate, we utilized settings from the Keras callback functions, including ModelCheckpoint, ReduceLROnPleatue, and EarlySopping, to get perfectly fitted model. Batch size was set to 32 with 100 epochs. These are used as callback functions, and we halt model training after the loss rate drops for any consecutive iteration. Choosing the optimal loss function also affected the accuracy of the model by 1-3%. Categorical

cross-entropy and binary cross entropy were used but binary cross entropy was chosen as a perfect loss function for the model.

Impact of some hyperparameter tunings on the accuracy of models in general. Epoch size had a major impact on accuracy of the model. Selecting an epoch size of 100 has an average accuracy improvement of 4% in all the models but this led to a high value of validation loss and was resulting in overfitting of the model. Allowing the model to stop when there is no improvement in validation loss from previous epoch results in an optimal model which fits well with training as well as testing data.

Choosing an optimal optimization algorithm was also vital for the models. Gradient descent and adagrad were initially used but Adam provided the best result with about 1-2% improvement in overall accuracy in all the models.

Overall, hyperparameter tuning is a crucial step in the development of neural networks for natural language processing tasks. By carefully selecting the optimal values for the hyperparameters, it is possible to improve the performance of the network and achieve better results.

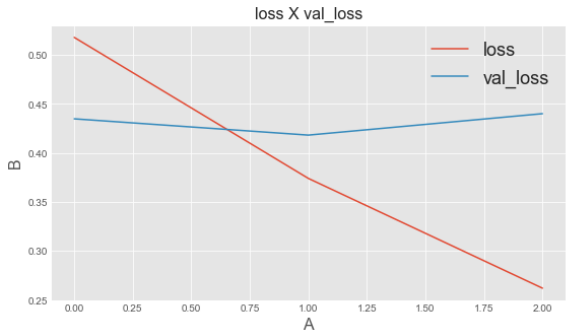
EXPERIMENTAL RESULT ANALYSIS

The overall results show that all of them have performed and produced results that are similar, but BiGRU is superior to other models in terms of precision and accuracy, scoring 89% and 81%, respectively. However, its average F1-Score value declines when compared to other models, and time consumption is at its highest.

	CNN	RNN	BiGRU	BiLSTM
F1-Score	0.73	0.73	0.63	0.74
Precision	0.88	0.84	0.89	0.85
Recall	0.63	0.65	0.64	0.66
Accuracy	0.8	0.79	0.81	0.8
Training Time(seconds)	31.7	108.6	292.7	202.4
Epochs Executed (Out of 100)	3	6	6	4

Table 1- Comparative analysis parameters for all models

The CNN model is a straightforward feed-forward network, but it needs a lot of data to train, hence it has the fewest execution epochs among the other three models since it overfitted the earliest. Although both non-bilateral models produce outcomes that are inferior to those of bilateral models, they both need less calculation time each epoch. For all algorithms, the recall value is the same.



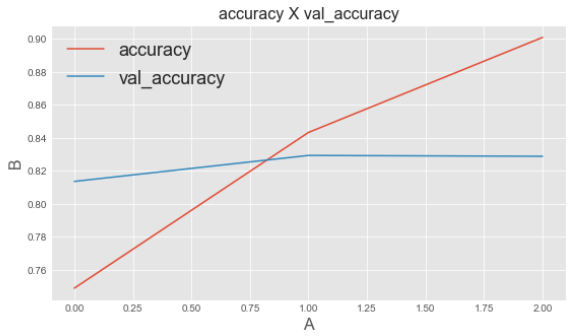


Fig 11- Line graph of loss vs validation loss(right) and accuracy vs validation accuracy(left) to train CNN model

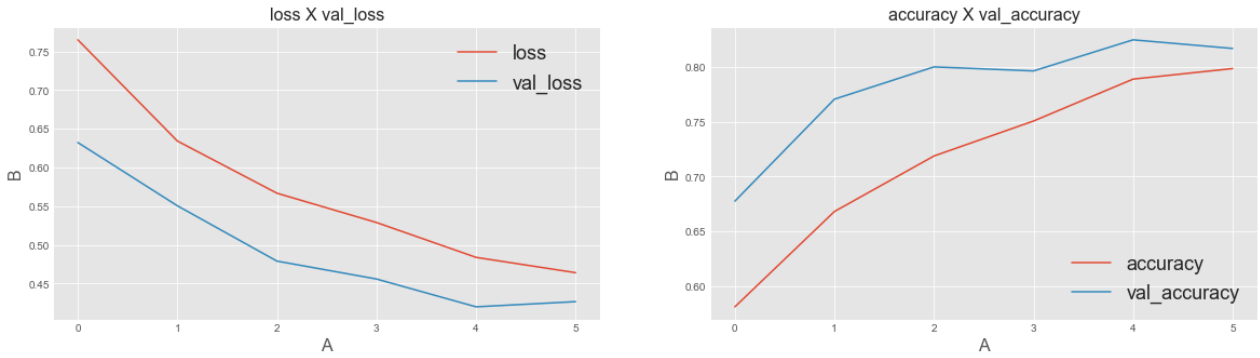


Fig 12- Line graph of loss vs validation loss(right) and accuracy vs validation accuracy(left) to train RNN model

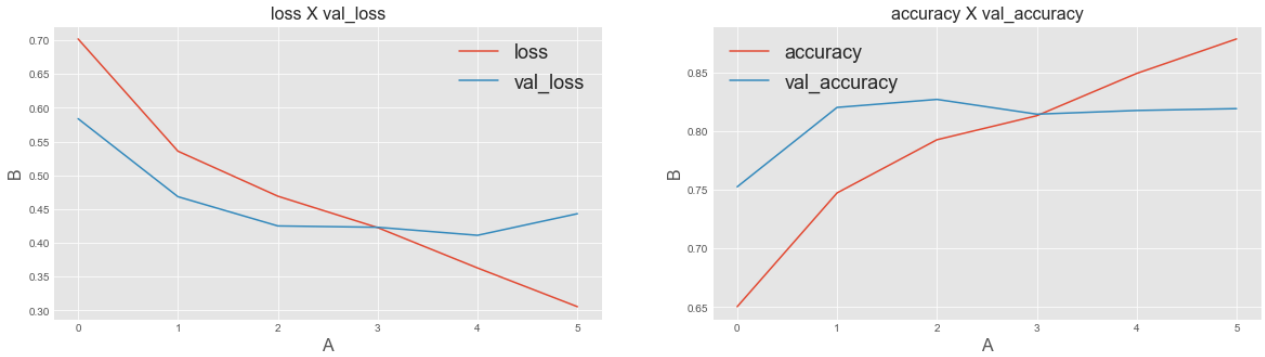


Fig 13- Line graph of loss vs validation loss(right) and accuracy vs validation accuracy(left) to train BiGRU model

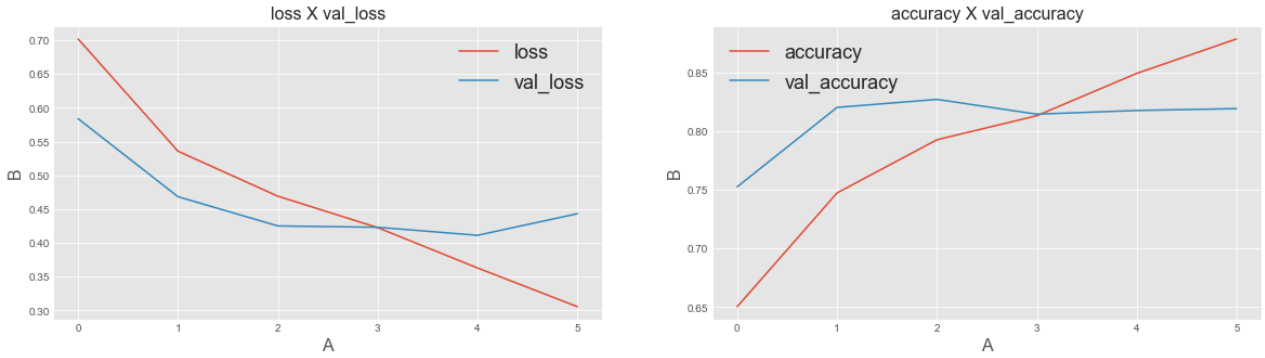


Fig 14- Line graph of loss vs validation loss(right) and accuracy vs validation accuracy(left) to train BiLSTM model

CONCLUSION AND FUTURE WORK

This study examined and contrasted the classification performance of four different Neural Network models. With varying degrees of success, tweets were analyzed for the detection of disaster tweets. The suggested method enables the detection of a disaster tweet with reliable accuracy. The results from the Bilateral Gated Recurrent Network Unit, with an accuracy of 81.81%, show the most promising results.

One future work would be to experiment with different numbers of epochs to train the model. The layers of models can also be altered to change the model's behavior and improve the overall accuracy. The experiment can be enhanced further with the usage of other prebuilt models like AlexNet, DenseNet and UNet. Steps like Batch Normalization and Regularization can be added to check if the model's performance can be improved. A faster and more efficient GPU can be used to train the models. Each model taking more than an hour to train proved to be very time-consuming as even making minor changes to the codes resulted in a long waiting time to see the results. Hyperparameters of the models can also be fine-tuned to raise the overall accuracy score.

REFERENCES

[1] K. Bontcheva et al., "Towards real-time classification of Twitter streams for crisis management," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.

[2] T. Miyoshi et al., "Deep learning for earthquake damage assessment from social media," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018.

[3] G. Mohapatra et al., "Twitter as a disaster relief tool," in Proceedings of the 5th International Conference on Information and Communication Technologies and Development, 2011.

[4] Menni C, Valdes AM, Freidin MB, Sudre CH, Nguyen LH, Drew DA, et al. Real-time tracking of self-reported symptoms to predict potential COVID-19.

[5] Smith A, Anderson M. Social media use in 2018. Pew Research Center. 2018 Mar 01. URL: https://www.pewresearch.org/internet/2018/03/01/social-media-use-in-2018/

[6] Sarker A, Lakamana S, Hogg-Bremer W, Xie A, Al-Garadi M, Yang Y. Self-reported COVID-19 symptoms on Twitter: an analysis and a research resource. J Am Med Inform Assoc 2020 Aug 01

[7] Jeon J, Baruah G, Sarabadani S, Palanica A. Identification of Risk Factors and Symptoms of COVID-19: Analysis of Biomedical Literature and Social Media Data. J Med Internet Res 2020

[8] Klein A, Magge A, O'Connor K, Flores Amaro J, Weissenbacher D, Gonzalez Hernandez G Toward Using Twitter for Tracking COVID-19: A Natural Language Processing Pipeline and Exploratory Data Set J Med Internet Res 2021;23(1): e25314

[9] Wang, Jin, et al. "Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification." IJCAI. Vol. 350. No. 10.5555. 2017.

[10] Arora, Monika, and Vineet Kansal. "Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis." Social Network Analysis and Mining 9.1 (2019): 1-14.

[11] Yuan, Ye, and You Zhou. "Twitter sentiment analysis with recursive neural networks." CS224D course projects (2015).

[12] Smith, Leslie N. "A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay." arXiv preprint arXiv:1803.09820 (2018).

[13] Makwe, Aditya, and Abhishek Singh Rathore. "An empirical study of neural network hyperparameters." Evolution in Computational Intelligence. Springer, Singapore, 2021. 371-383.

[14] Haque, Rezaul, et al. "A comparative analysis on suicidal ideation detection using NLP, machine, and deep learning." Technologies 10.3 (2022): 57.

[15] Joshi, Sharmad, et al. "Analysis of Preprocessing Techniques, Keras Tuner, and Transfer Learning on Cloud Street image data." 2021 IEEE International Conference on Big Data (Big Data). IEEE, 2021.

[16] Shubh, Muskan Agarwal, et al. "Handwriting Recognition Using Deep Learning." Emerging Trends in Data Driven Computing and Communications: Proceedings of DDCIoT 2021 (2021): 67.

[17] Sharfuddin, Abdullah Aziz, Md Nafis Tihami, and Md Saiful Islam. "A deep recurrent neural network with bilstm model for sentiment classification." 2018 International conference on Bangla speech and language processing (ICBSLP). IEEE, 2018.

[18] Chauhan, Rahul, Kamal Kumar Ghanshala, and R. C. Joshi. "Convolutional neural network (CNN) for image detection and recognition." 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC). IEEE, 2018.

[19] Lang, Zhe, et al. "PMatch: Semantic-based Patch Detection for Binary Programs." 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC). IEEE, 2021.

[20] Liu, Jin, et al. "Attention-based BiGRU-CNN for Chinese question classification." Journal of Ambient Intelligence and Humanized Computing (2019): 1-12.