Ngozi Nsofor _ 202219469

CLASSIFICATION OF DISASTER TWEETS

Abstract

Natural and man-made disasters occur frequently and have dire environmental and economic consequences. Artificial Intelligence (AI) researchers have tested Natural Language Processing (NLP) techniques on disaster-related Twitter data to aid first responders in providing prompt disaster response. This research investigates the effectiveness of combining word vectorizers with Multinomial Naïve Bayes (MNB) over word embedders with Bilateral Long Short-Term Memory (BiLSTM) Recurrent Neural Network (RNN) to classify tweets obtained from around the world as disaster-related or not. Using accuracy, precision, recall and f1-score as metrics, the GloVe and BiLSTM combination outperformed other models and baselines.

1.0 Introduction

The United Nations Office for Disaster Risk Response (2023), in the Sendai Framework, highlighted the importance of efficient disaster risk management, as disasters could cause adverse economic and environmental consequences.

Natural disasters have accounted for 90% of disaster occurrences worldwide (Wannous and Velasquez, 2017). According to Guha-Sapir, Hoyois and Below (2016), an average of 380 natural disasters occurred annually between 2005 and 2015 resulting in 190 million victims and over US $150 billion in economic damages. Considering the severity of these consequences, the Sendai Framework emphasizes the importance of combining science, technology, and policies to reduce the risk of disasters and provide solutions to relief efforts (Wannous and Velasquez, 2017).

Social media, notably Twitter, provides an avenue to obtain online real-time user-generated information on disasters, which can be useful to first responders in providing effective disaster response to save lives and property (Khun et al., 2019). However, this becomes challenging as it is necessary to first determine which tweets are about real disasters, and which are unrelated (Gao et al., 2020).

Research in the field of AI has seen an increase in the application of NLP techniques like sentiment analysis and text classification of Twitter posts to aid in disaster response. By identifying and classifying tweets about real disasters, information can be extracted on situational awareness and disaster locations for resource allocation and rescue efforts (Asinthara, Jayan and Jacob, 2022).

To achieve this, numerous studies have focused on the use of Machine Learning (ML) algorithms like Support Vector Machines (SVM), more advanced techniques like Recurrent Neural Networks (RNNs) and state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019).

This research aims to derive insights into the effectiveness of combining word vectorizers with a traditional ML model, and word embedders with a RNN for classifying disaster-related tweets. Using BiLSTM which is less computationally expensive than more advanced methods and has also shown great success in this field (Asinthara, Jayan and Jacob, 2022), the goal is to contribute to the development of simpler and efficient methods for tweet classification in this domain.

2.0 Background

Twitter is used by millions of people worldwide. It is therefore no surprise that it has gained popularity among AI researchers as a platform for accessing data to test NLP techniques. It has also been recognized as a source of information for situational awareness on disasters resulting in a rise in the application of NLP techniques on twitter data for disaster response (Beigi et al., 2016).

2.1 Related Work

Some researchers have focused on comparing the performance of different algorithms for the classification of disaster tweets and how different factors could contribute positively or adversely to the results.

Roy, Mishra and Matam (2020) extracted tweets related to the Fani Cyclone that occurred in 2019 using the keyword “Fani Cyclone”. SVM was used for the binary classification of tweets as Informative and Non-informative. An accuracy of 74.268% was obtained on the preprocessed twitter data as opposed to 69.426% without preprocessing, thus concluding that data preprocessing is important for the classification task. A data summarization algorithm was also included in the experimental set up to provide information on the number of victims and situational awareness.

Parilla-Ferrer, Fernandez and Ballena (2015) compared the performance of SVM with Naïve Bayes using manually labelled data from the Habagat flooding of Metro Manilla in 2012. SVM outperformed Naïve Bayes in accuracy, recall, and f1-score; however Naïve Bayes gave a better precision (84%) than SVM (79.4%). It is important to state that the data was very imbalanced containing 65% irrelevant tweets and 35% relevant tweets.

Kabir and Madria (2019) collected Twitter data from Hurricanes Harvey and Irma for a multiclass classification of tweets to determine rescue situation and resources needed. Using a combination of BiLSTM with an additional Conv1D layer from a Convolutional Neural Network (CNN) and GloVe word embeddings, classification was done on 4,900 manually labelled tweets. Their set up outperformed the Logistic

Regression, SVM and CNN baselines. Results obtained for their model was 93.7% accuracy, 81.7% precision, 93.4% recall and 87.2% f1-score.

Alrashdi and O'Keefe (2019) compared the performance of BiLSTM and CNN models using GloVe embedding and Crisis (a domain-specific pretrained embedding technique) on NLP Crisis dataset. BiLSTM with GloVe gave the highest f1-score (62.04%) over the other combinations. They concluded that GloVe embedding is as effective as domain-specific embedding techniques for performing classification tasks.

2.2 Aims and Objectives

The above researchers have used NLP techniques on data relating to specific disasters in one or more locations. However, this research would compare the performance of a ML model (MNB) and a Deep Learning (DL) model (BiLSTM) using word vectorizers and word embedders respectively in classifying data from different parts of the world that report on various disasters.

To achieve this, I will build four (4) different models. Two models will be built using Count Vectorizer and TF-IDF Vectorizer with MNB, and the other two using Word2vec and GloVe word embeddings with the BiLSTM RNN.

The performance of all the models would be evaluated to determine which combination performs best for this task. The best performing RNN model combination would be optimized and compared with other models obtained from previous academic work as baselines.

3.0 Methodology

This section outlines the process of achieving the aims of this research.

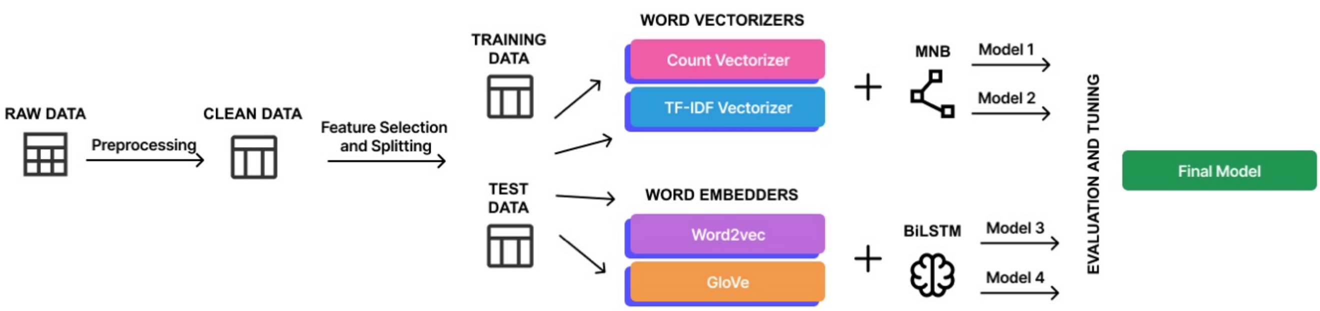

Figure 1: Methodology flow diagram

3.1 The Data

The data used for this research was obtained from Kaggle (Howard et al., 2019). It is a compilation of 7,613 manually labelled tweets from various locations around the world compiled using disaster-related keywords like aftershock, fire, accident, etc. The metaphorical use of the keywords can be easily distinguished by humans, for example ***"The bass guitarist at that concert was on fire!"***, but may prove difficult for a ML model to deduce.

id	keyword	location	text	target
93	ablaze	Birmingham	@nxwestmidlands huge fire at Wholesale markets ablaze http://t.co/rwzbFVNXER	1
54	ablaze	Pretoria	@PhDSquares #mufc they've built so much hype around new acquisitions but I doubt they will set the EPL ablaze this season.	0

Table 1: Raw dataset sample (Howard et al., 2019)

3.2 Text Data Preprocessing

Preprocessing is important to provide structure to the data, improve computation time and accuracy of the classification task (Pradha, Halgamuge and Tran Quoc Vinh, 2019).

As this research was entirely focused on the binary classification of tweets, the first step towards preprocessing was the removal of the unnecessary columns in the dataset – the ***"id"***, ***"keyword"*** and ***"location"*** columns.

Using a function, the data was cleaned with the steps outlined below:

- Expansion of contractions: Contractions in the text were expanded. For example, words like ***"I'm"*** were changed to ***"I am"***.

- Removal of mentions (@twitteruser): Mentions are used by Twitter users to tag other users to a tweet. Mentions are not relevant in obtaining the context of a tweet.

- Removal of URLs and HTML tags: URLs and HTML tags were removed as they do not contain relevant information.

- Removal of Numbers and Punctuations: Numbers and punctuations were removed from the text as they are irrelevant to the task.

- Conversion of text to lower case: This is important to avoid word duplication. For example, ***"Lower"*** and ***"lower"*** would be treated as separate words when in fact they are the same.

- Tokenization: This involves breaking the text into individual words to facilitate lemmatization and stop words removal.

- Lemmatization: This reduces words to their base form based on their dictionary meanings.

- Removal of stop words and non-ascii characters: Stops words are words frequently used in a sentence that add no value to the context like ***"a"***, ***"an"***, ***"the"***. This is important to this task as word vectorizers use word frequency to determine importance during feature extraction. Non-ascii words were removed to ensure only words frequently used in English are retained in the data.

The data was inspected for duplicated rows and null values. Ninety-two (92) duplicated rows were found and removed, and no null values were present in the data. The final dataset contained 7521 rows of cleaned text data and 2 columns – ***"target"*** and ***"clean_text"***.

text	**target**	**clean_text**
Our Deeds are the Reason of this #earthquake May ALLAH Forgive us all	1	deed reason earthquake may allah forgive u
'Four hundred wrecked cars (costing $100 apiece) were purchased for the making of this 1986 film - http://t.co/DTdidinQyF'	0	four hundred wrecked car costing apiece purchased making film

Table 2: Data comparison showing before and after cleaning

3.3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) was performed on the data.

From the visualization below, it can be observed that the data, though not completely balanced, has an adequate amount of information for the target class (1) that is sufficient to build the model and avoid bias. It also shows that the accuracy of the final models could be relied upon to an extent in evaluating their performance (Paduraru and Breaban, 2019).

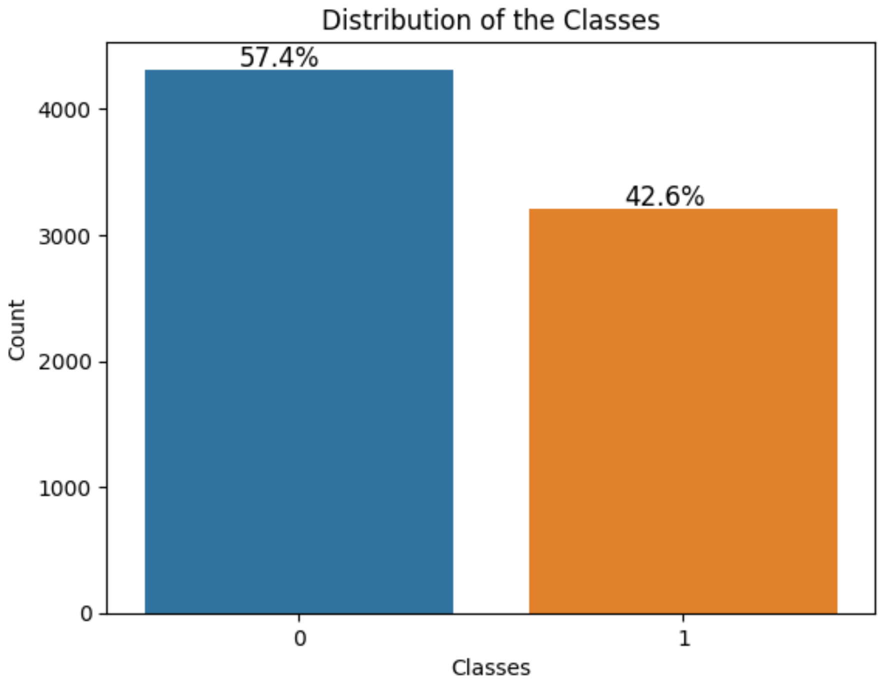

Figure 2: Distribution of the classes

3.3.1 WordCloud

The WordCloud visualization technique was used to visualize the frequency of words in the classes. This technique uses the size of words to represent their frequency in the data (Hossain et al., 2021).

The visualization shows a distinction between tweets in the two classes. Words like fire, flood, death, suicide bomber, storm, etc. which are disaster related can be seen in the disaster tweet visualization. Conversely, the irrelevant tweets visualization shows everyday words, and the disaster-related words used metaphorically are smaller.

This distinction is important for the models’ performance. It shows potential towards effective classification as the model can distinguish the data more accurately.

(3a) Word frequency of disaster tweets

(3b) word frequency of irrelevant tweets

Figure 3: WordCloud visualization of word frequency in disaster and irrelevant tweets.

3.4 Word Vectorizers

Word vectorization involves converting textual data to numerical data (vectors) which can be fed to the ML algorithm (Singh and Shashi, 2019). The Count Vectorizer and TF-IDF Vectorizer are used in this research. The difference between these two techniques is that Count Vectorizer gives importance to frequently occurring words while the opposite is the case for TF–IDF Vectorizer which assigns lower weights to frequent words and higher weights to fewer occurring words.

Count Vectorizer is a technique in Python's scikit-learn library which uses the frequency (count) of each word in a text to create vectors (Assery et al., 2019).

The Term Frequency – Inverse Document Frequency (TF-IDF) Vectorizer assigns weights to words based on their frequency in the text and their importance in the document (Assery et al., 2019) calculated as TF * IDF where:

$$\boldsymbol{TF} = \frac{\text{frequency } of\ word\ (w)\ in\ a\ tweet}{\text{total number of words in the tweet}} \tag{3.1}$$

$$\boldsymbol{IDF} = \log\left(\frac{\text{total number of tweets}}{\text{number of tweets that contain word (w)}}\right) \tag{3.2}$$

3.5 Word Embedders

Word embedding is an enhancement from word vectorization. Each word is represented by a high-dimensional vector such that the semantic meaning of the word is retained. These vectors are positioned in a vector space and similar words are clustered together in the space (Curto et al., 2022).

The embedding techniques used in this report are ***Word2vec*** (Mikolov et al., 2013) and ***Twitter GloVe*** (Pennington, Socher and Manning, 2014).

3.6 Model Selection

Multinomial Naïve Bayes (MNB) works by predicting the probability of a text belonging to each classification class based on the frequency of the words in the text (Singh et al., 2019). This model was chosen based on its success in text classification, its simplicity and computational efficiency.

Bilateral Long Short-Term Memory (BiLSTM) model consists of two LSTM layers, where one layer reads the input sequence using forward propagation and the other using backward propagation to capture patterns in sequential data. It is a type of RNN that can process sequential data, making it useful for NLP techniques. Its design is such that it can remember previous inputs which it uses to predict future outputs (Graves and Schmidhuber, 2005).

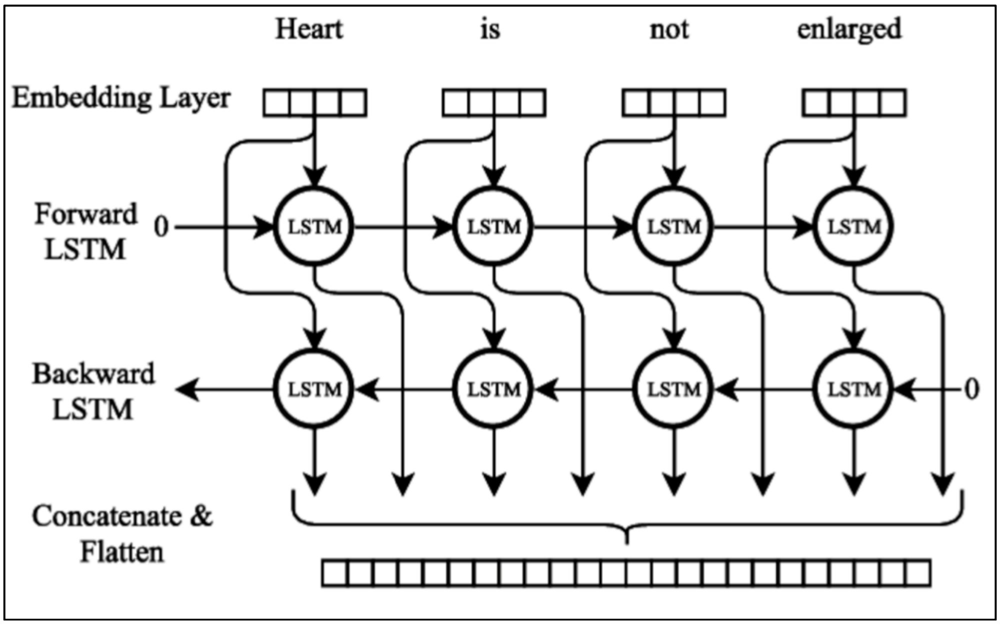

Fig 4: Architecture of a BiLSTM model (Tam, Said and Tanriover, 2021)

4.0 Experiments

This section explains the process of combining simpler vectorizers with the MNB algorithm and more complex vectorizers with BiLSTM to achieve the classification task on the dataset.

4.1 The Dataset

As previously described, the dataset was obtained from Kaggle, a compilation of 7,613 tweets from around the world containing both disaster-related and irrelevant tweets. Preprocessing was done using the Natural Language Tool Kit (NLTK) developed by Bird, Klein and Loper (2009) and RegEx python libraries (Python, 2009). NLTK was used to perform tokenization, lemmatization and removal of stop words from the text, while RegEx was used to remove mentions, URLs, HTML tags and numbers.

4.2 Data Splitting

The data was split into training (80%) and testing (20%) sets and the stratify hyperparameter was set to the target (y) to ensure the percentage of the classes in the data is maintained during the split. Splitting is done to avoid overfitting which happens when a model completely learns the data used to train it and will not generalize well to new data.

4.3 Feature Extraction and Multinomial Naïve Bayes

This is the process of converting text data to vectors using the count and frequency of words to assign numbers or weights to the data. The Count Vectorizer and TF–IDF Vectorizer were used. To avoid sparsity in the data that could affect model performance, the ***max_features*** hyperparameter of both vectorizers was set to five hundred (500), ensuring the top 500 words are selected as features for the MNB algorithm.

Both methods were applied separately to the training and testing data and two models were built using each vectorization technique.

4.4 Word Embeddings and BiLSTM

This process involves converting words to their numerical embeddings while retaining their semantic meaning. Word2vec and GloVe embeddings were used.

To create the embeddings, the text was tokenized, using the ***word_tokenize*** function, and passed into the embedding model. An embedding dictionary was created to map each word to its corresponding vector value using the words as keys and the vectors as the values.

A sparse matrix with a dimension of 100 was created to house the embedding values of the words in the dictionary, and a random embedding is generated if the word is not found in the dictionary.

Numerical sequences of each row in the data were created by mapping each word to its corresponding vector value and finally the sequences were padded with zeros to ensure equal lengths of the sequences. These sequences are fed into the BiLSTM Neural Network.

4.5 BiLSTM Architechture and Hyperparameters

The BiLSTM architecture comprised of an embedding layer which the pre-trained embedding models are passed into; three hidden layers; a dropout layer was added to reduce the model's complexity and avoid overfitting; the output dense layer with units set to the number of classes and the sigmoid activation function used for binary classification.

Hyperparameter	**Value**
BiLSTM_units	64, 128, 256
Dropout	0.3
Activation	Sigmoid
Optimizer	Adam
Learning_rate	0.0003
Loss_function	binary_crossentropy
Metrics	accuracy
Epochs	20
Batch_size	32
EarlyStopping_monitor	val_accuracy
Min_delta	0.0001

Table 3: BiLSTM model hyperparameters

4.6 Evaluation Metrics

The models were evaluated using the accuracy, precision, recall and f1-score.

Accuracy is the ratio of the total number of correct predictions in the classes to the total number of predictions

$$\boldsymbol{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{4.1}$$

Precision is the percentage of correct predictions made for the target class out of all predictions made for the target class

$$\boldsymbol{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4.2}$$

Recall is the percentage of correct predictions made for the target class out of the total number of the target class in the data

$$\boldsymbol{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4.3}$$

F1-score is calculated as the harmonic mean of the precision and the recall. It aims to find a balance between the two metrics.

$$\boldsymbol{F1 - score} = 2 * \left(\frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}\right) \tag{4.4}$$

Where TP, TN, FP, and FN = True Positives, True Negatives, False Positives and False Negatives respectively.

4.7 Hyperparameter Tuning

The RNN model with the highest accuracy was tuned to try for a higher score. Tuning was done using the RandomSearch tuner in the kerastuner library. Tuned hyperparameters included a range for the number of units in the hidden layers, and a random choice for the learning rate, and validation accuracy was monitored.

Hyperparameter	**Value**
Learning_rate	1e-2, 1e-3, 1e-4, 5e-4
BiLSTM_units	min_value=32, max_value=256

Table 4: Hyperparameter tuning values

4.8 Baselines

Deb and Chanda (2022) used similar data obtained from Kaggle to compare the efficacy of different word vectorization and embedding techniques, including BERT. Their combination of Bag of Words (BOW) representations with Logistic Regression (LR), Random Forest (RF) and Decision Tree (DT) would be used as baselines with focus on accuracy and f1-score.

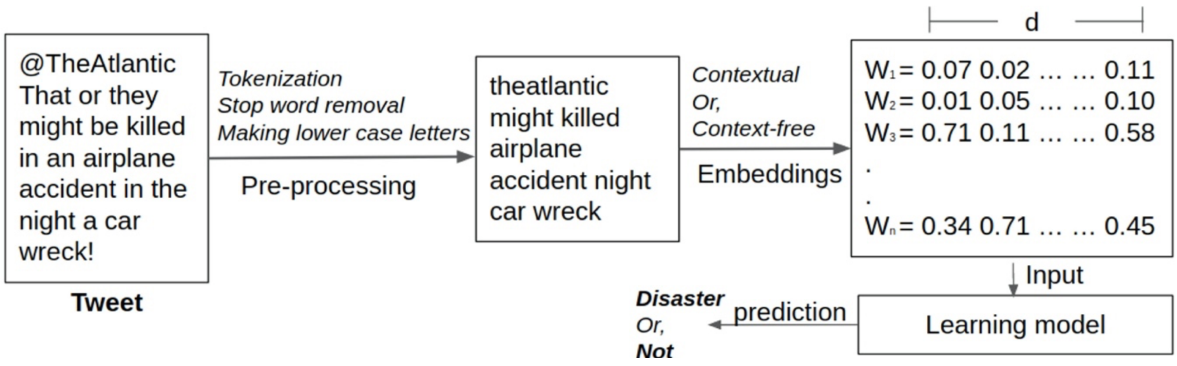

Figure 5: Graphical representation of the prediction model (Deb and Chanda, 2022)

5.0 Results

The GloVe and BiLSTM model combination outperformed other models in all metrics except precision where the Word2vec and BiLSTM combination showed dominance (96%).

Interestingly, the MNB models outperformed the Word2vec and BiLSTM model. This could indicate the need for more tuning of the Word2vec model's hyperparameters. The results and confusion matrices are seen below.

Metric	**CountVectorizer and MNB**	**TF-IDF and MNB**	**Word2vec and BiLSTM**	**GloVe and BiLSTM**
Accuracy	0.76	0.77	0.64	0.80
Precision	0.76	0.78	0.96	0.77
Recall	0.65	0.63	0.17	0.75
F1-score	0.70	0.70	0.29	0.76

Table 5: Model performance evaluation

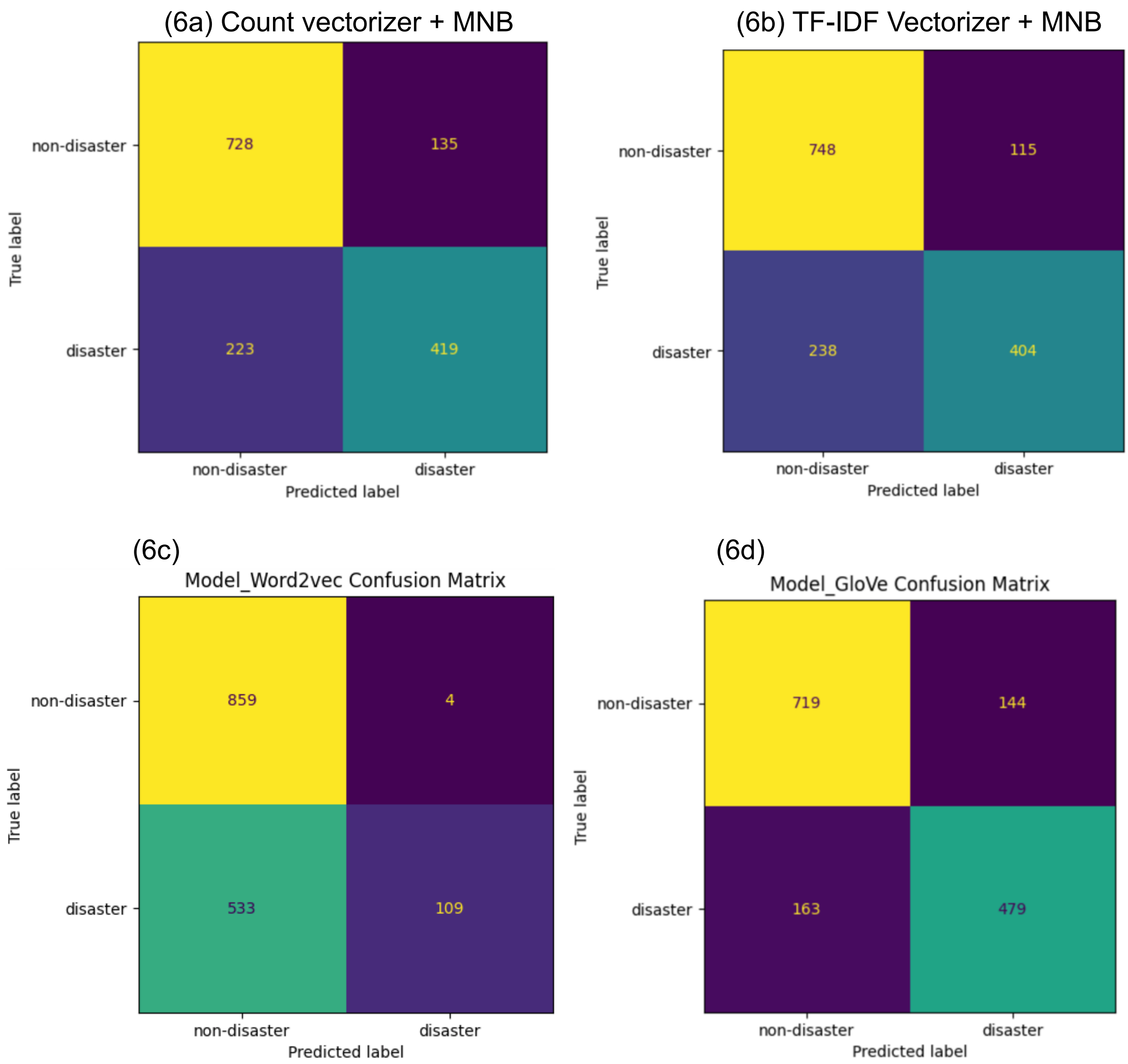

Figure 6: Confusion matrices for the MNB and BiLSTM models

5.1 Training Performance

The models' training performance can be seen in the **Table 6** and **Figure 7**. It shows maximum accuracy and minimum loss obtained for the models.

Model	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Epochs
Word2vec + BiLSTM	0.67	0.68	0.60	0.58	11
GloVe + BiLSTM	0.81	0.80	0.42	0.44	9

Table 6: Model training performance

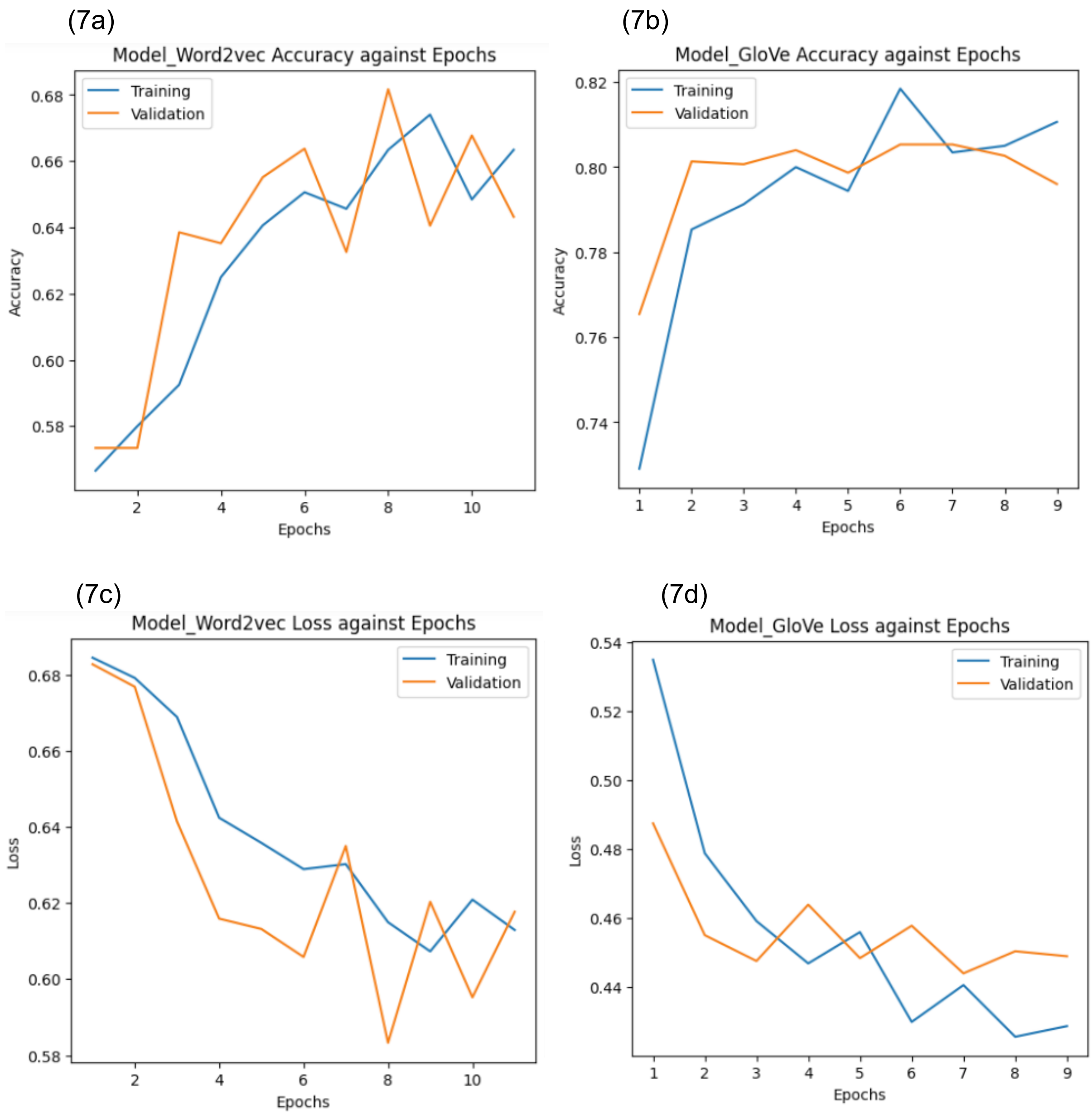

Figure 7: Accuracy and Loss against epochs for the word embedders and BiLSTM

5.2 Hyperparameter Tuning and Baselines

Hyperparameter tuning improved the precision of the best model, however accuracy stayed the same while recall and f1-score reduced. **Table 7** shows the best model compared with the accuracy and f1-score of the baselines and tuned model.

Metric	GloVe and BiLSTM	Tuned GloVe and BiLSTM	Logistic Regression + BOW	Random Forest + BOW	Decision Tree + BOW
Accuracy	0.80	0.80	0.73	0.70	0.64
F1-score	0.76	0.74	0.74	0.73	0.59
Precision	0.77	0.84	-	-	-
Recall	0.75	0.67	-	-	-

Table 7: Best model compared with tuned model and baselines

6.0 Conclusion

This research compared the performance of word vectorizers with a simple classifier (MNB) and word embedders with BiLSTM on twitter data for binary classification. The results showed that the combination of BiLSTM and GloVe embedding outperformed other model combinations including the LR, RF and DT baselines.

From the results, it can be concluded that simple classifiers combined with word vectorizers could be as effective as RNNs built with word embedders for the chosen dataset. The models' performance show that these techniques could be used when resources and computational power for more advanced techniques are not available.

Future research could investigate building the models on a larger and more balanced dataset that is sufficiently preprocessed, combined with advanced techniques like BERT to obtain better results. This could be tested for monitoring disasters on a global scale by concerned organizations like the United Nations and Governments.

BIBLIOGRAPHY

Alrashdi, R. and O'Keefe, S.E.M. (2019). Deep Learning and Word Embeddings for Tweet Classification for Crisis Response. *ArXiv*, [online] abs/1903.11024. Available at: https://arxiv.org/pdf/1903.11024.pdf [Accessed 20 Apr. 2023].

Asinthara, K., Jayan, M. and Jacob, L. (2022). *Classification of Disaster Tweets Using Machine Learning and Deep Learning Techniques*. [online] IEEE Xplore. doi:https://doi.org/10.1109/TQCEBT54229.2022.10041629.

Assery, N., Xiaohong, Y., Almalki, S., Kaushik, R. and Xiuli, Q. (2019). *Comparing Learning-Based Methods for Identifying Disaster-Related Tweets*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICMLA.2019.00295.

Beigi, G., Hu, X., Maciejewski, R. and Liu, H. (2016). An Overview of Sentiment Analysis in Social Media and Its Applications in Disaster Relief. *Sentiment Analysis and Ontology Engineering*, pp.313–340. doi:https://doi.org/10.1007/978-3-319-30319-2_13.

Bird, S., Klein, E. and Loper, E. (2009). *Natural Language Processing with Python*. Beijing : O'reilly.

Curto, G., Jojoa Acosta, M.F., Comim, F. and Garcia-Zapirain, B. (2022). Are AI Systems Biased against the Poor? A Machine Learning Analysis using Word2Vec and GloVe Embeddings. *AI & SOCIETY*. doi:https://doi.org/10.1007/s00146-022-01494-z.

Deb, S. and Chanda, A.K. (2022). Comparative analysis of contextual and context-free embeddings in disaster prediction from Twitter data. *Machine Learning with Applications*, [online] p.100253. doi:https://doi.org/10.1016/j.mlwa.2022.100253.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, [online] 1, pp.4171–4181. doi:https://doi.org/10.18653/v1/n19-1423.

Gao, W., Li, L., Zhu, X. and Wang, Y. (2020). Detecting Disaster-Related Tweets Via Multimodal Adversarial Neural Network. *IEEE MultiMedia*, 27(4), pp.28–37. doi:https://doi.org/10.1109/mmul.2020.3012675.

Graves, A. and Schmidhuber, J. (2005). Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, [online] 18(5-6), pp.602–610. doi:https://doi.org/10.1016/j.neunet.2005.06.042.

Guha-Sapir, D., Hoyois, P. and Below, R. (2016). *Annual Disaster Statistical Review 2015: The Numbers and Trends* . [online] Brussels: Centre for Research on the Epidemiology of Disasters (CRED). Available at: http://www.cred.be/sites/default/files/ADSR_2015.pdf [Accessed 13 Apr. 2023].

Hossain, A., Karimuzzaman, M., Hossain, M.M. and Rahman, A. (2021). Text Mining and Sentiment Analysis of Newspaper Headlines. *Information*, [online] 12(10), p.414. doi:https://doi.org/10.3390/info12100414.

Howard, A., Devrishi, Culliton, P. and Guo, Y. (2019). *Natural Language Processing with Disaster Tweets*. [online] kaggle.com. Available at: https://kaggle.com/competitions/nlp-getting-started [Accessed 28 Feb. 2023].

Kabir, Md.Y. and Madria, S. (2019). A Deep Learning Approach for Tweet Classification and Rescue Scheduling for Effective Disaster Management. In: *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. [online] New York, NY, USA: Association for Computing Machinery, pp.269–278. doi:https://doi.org/10.1145/3347146.3359097.

Khun, N.H., Zin, T.T., Yokota, M. and Thant, H.A. (2019). *Emotion Analysis of Twitter Users on Natural Disasters*. [online] IEEE Xplore. doi:https://doi.org/10.1109/GCCE46687.2019.9015234.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. [online] arXiv.org. Available at: https://arxiv.org/abs/1301.3781 [Accessed 19 Apr. 2023].

Padurariu, C. and Breaban, M.E. (2019). Dealing with Data Imbalance in Text Classification. *Procedia Computer Science*, [online] 159, pp.736–745. doi:https://doi.org/10.1016/j.procs.2019.09.229.

Parilla-Ferrer, B.E., Fernandez, P.L. and Ballena , J.T. (2015). Automatic Classification of Disaster-Related Tweets. *International Conference on Innovative Engineering Technologies (ICIET)*. [online] Available at: http://iieng.org/images/proceedings_pdf/3606E1214072.pdf [Accessed 19 Apr. 2023].

Pennington, J., Socher, R. and Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, [online] pp.1532–1543. doi:https://doi.org/10.3115/v1/d14-1162.

Pradha, S., Halgamuge, M.N. and Tran Quoc Vinh, N. (2019). Effective Text Data Preprocessing Technique for Sentiment Analysis in Social Media Data. *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, pp.1–8. doi:https://doi.org/10.1109/kse.2019.8919368.

Python (2009). *Re — Regular Expression Operations — Python 3.7.2 Documentation*. [online] Python.org. Available at: https://docs.python.org/3/library/re.html [Accessed 22 Apr. 2023].

Roy, S., Mishra, S. and Matam, R. (2020). *Classification and Summarization for Informative Tweets*. [online] IEEE Xplore. doi:https://doi.org/10.1109/SCEECS48394.2020.128.

Singh, A.K. and Shashi, M. (2019). Vectorization of Text Documents for Identifying Unifiable News Articles. *International Journal of Advanced Computer Science and Applications*, [online] 10(7). doi:https://doi.org/10.14569/ijacsa.2019.0100742.

Singh, G., Kumar, B., Gaur, L. and Tyagi, A. (2019). *Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICACTM.2019.8776800.

Tam, S., Said, R.B. and Tanriover, O.O. (2021). A ConvBiLSTM Deep Learning Model-Based Approach for Twitter Sentiment Classification. *IEEE Access*, [online] 9, pp.41283–41293. doi:https://doi.org/10.1109/access.2021.3064830.

United Nations Office for Disaster Risk Response (2023). *The Report of the Midterm Review of the Implementation of the Sendai Framework for Disaster Risk Reduction 2015–2030*. [online] UNDRR: Geneva, Switzerland. Available at: https://www.undrr.org/publication/report-midterm-review-implementation-sendai-framework-disaster-risk-reduction-2015-2030 [Accessed 8 Mar. 2023].

Wannous, C. and Velasquez, G. (2017). United Nations Office for Disaster Risk Reduction (UNISDR)—UNISDR's Contribution to Science and Technology for Disaster Risk Reduction and the Role of the International Consortium on Landslides (ICL). *Advancing Culture of Living with Landslides*, pp.109–115. doi:https://doi.org/10.1007/978-3-319-59469-9_6.