Disaster Classification of Tweets

Reuben Rouse
Mukesh Patel School of Technology Management and Engineering
Mumbai, India
reubenjrouse@gmail.com

Karan Shah
Mukesh Patel School of Technology Management and Engineering
Mumbai, India
karan@gmail.com

Lakshya Shetty
Mukesh Patel School of Technology Management and Engineering
Mumbai, India
lakshya@gmail.com

***Abstract*—The purpose of this project is to use social media data, like tweets, to explore and create a reliable disaster classification model. Natural disasters or phenomenon like tsunamis, hurricanes, wildfires, and earthquakes frequently produce a lot of user-generated material for websites like Twitter. These tweets are a useful tool for disaster response and management because they can offer insightful information. To effectively use this information, we use a multi-stage approach that includes text preprocessing with lemmatization, feature extraction using the BERT model, and classification through a Long Short-Term Memory (LSTM) neural network.**

Keywords—BERT, LSTM, Disaster Classification, NLP

I. Overview

In today's digital age, people use social media apps like twitter to spread reliable information like disasters in real time and access to this information is crucial, but unfortunately people spread a lot of misinformation and if not moderated, can be extremely dangerous.

This paper will delve into how twitters crisis misinformation policy's algorithm can be strengthened.

Natural catastrophes have a terrible effect on infrastructure, property, and human life. Minimizing these events' effects requires the ability to quickly identify, categorize, and react to them. During such disasters, social media sites, especially Twitter, have become invaluable sources of up-to-date information. However, automatic catastrophe classification has difficulties due to the unstructured character of tweets and their vast number.

A. Primary objectives

1) Develop an automated disaster classification model using tweets

2) Understand how BERT is revolutionalizing NLP

3) How to pair BERT with LSTM to do text classification

Several studies have explored disaster classification using social media data. Recent advancements in Natural Language Processing (NLP) techniques, such as BERT, have significantly improved classification accuracy. Additionally, LSTM, a type of recurrent neural network, has been used for sequence modeling and classification tasks.

II. Methodology

A. Source of data

The data used for this in this research paper has been retrieved from Kaggle. The dataset contains 2 files

1) train.csv–Training dataset

a) id: unique identifier of the tweet

b) text: The tweet itself

c) keyword: Keyword of the tweet (can be empty)

d) location: Location from where it has been tweeted (can be empty)

e) target: 1 if it's a disaster, 0 if its not a disaster

2) test..csv–Testing dataset

a) id: unique identifier of the tweet

b) text: The tweet itself

c) keyword: Keyword of the tweet (can be empty)

d) location: Location from where it has been tweeted (can be empty)

The training dataset has 7613 tweets and the testing dataset has 3263 tweets. The training dataset is almost balanced and since there is less number of training tweets we wont be dropping the "*non disaster tweets*".

B. Data preprocessing

The steps we have followed to preprocess the training and testing datasets in this particular order are

1) Converting to lower case: converting all text to lower case makes it more convenient and reduces complexity so the model is not case senesitive.

2) Replacing abbreviations: Reducing abbreviations is extremely important when dealing with text taken from social media apps, containing slang.

3) Stopwords removal and lemmatization: Unecessary words are removed and non-stop words are reduced to their lemma form

4) Remove HTML tags

5) Replacing URLs: Since we have seen that although the number of disaster tweets is less the number of disaster tweets containing URLs are more than non disaster tweets, hence can be used a feature to distinguish between both.

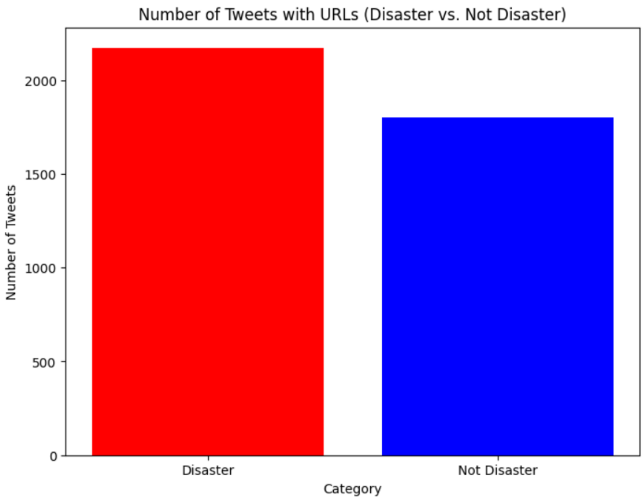

Fig. 1. This figure shows the number of tweets containing URLs.

6) Removing special characters: Punctuation marks and special characters removed.

7) Removal of common words: Observed most common words in disaster tweets and non-disaster tweets using WordCloud ans removed all the commonly occurcing words

Fig. 2. The most common words in disaster(left) and non-disaster(right) tweets

C. Feature extraction using BERT

BERT or Bidirectional Encoder Representations from Transformers is a bidirectional transformer which is pretrained on a large corpus of text with MLM (Masked Language Model) and NSP (Next Sentence Predictions) objectives. It is preferred for text classification but not text generation.

The name of our BERT model is “bert-base-multilingual-uncased”. It is trained on the top 102 languages with the largest Wikipedia. This model has only been trained on MLM objective and hence has a fair idea of how words work together.

We have set the maximum length of the input ids to 140, and anything more than 140 tokens will be truncated. After getting the initial embeddings of our tweets we will later fine tune them to our specific tasks in our classification model ahead.

Since the BERT model takes into account the entire context of the sentence it is very useful for classification as it understands the relationship between words, and later how they relate to each other in our given context and accordingly fine-tune themselves.

These continuous word embeddings are not only useful for text classification but also for sentiment analysis, machine translation etc. BERT significantly boosts performance for Natural Language Processing (NLP) tasks.

a) Limitations of BERT: While BERT is widely credited with revolutionizing NLP, the architecture has a number of drawbacks. Since BERT is bidirectional and adding the target during training would cause a target leakage, there is typically no apparent way to train it for auto-regressive tasks (predicting one token at a time during inference). To be precise, workarounds have been developed in some studies, but BERT designs are not SOTA in any of these tasks. BERT-like models remain some of the state-of-the-art models available today for the fine-tuning tasks stated above as well as other NLP categorization or understanding tasks.

D. Model training

We will be combining BERT based encoding along with an LSTM layer in our model for our binary classification model.

1) Input layer: This is the input data of our neural network. It has to be a sequence of 140 32-bit signed integers. These input ids are the initial word embeddings that we had generated earlier during feature extraction.

2) BERT encoder: This is the last layer of our BERT model which is attached to our input layer, this contains the contextual embeddings of the words.

3) LSTM layer: We are incorporating a Reccuring Neural Network (RNN) architecture along with BERT to further the contextual embeddings.

As we have seen earlier, BERT is amazing at capturing the contexual information about the corpous, but it lacks in capturing the sequential patterns of the data. LSTM is known for its capacity to remember the long term dependencies and is useful in NLP tasks where the order of the text is important.

Hence we’re using the contextual information provided by BERT along with the sequential dependencies modelling provided by LSTM.

4) Dense layer: The last layer is a dense layer with a single unit and an activation function of sigmoid. This layer is commonly seen in binary classification tasks as sigmoid is preferred for binary classification as it gives a output range of (0,1) with a default threshold of 0.5. This resembles a probability distribution function.

```
 Layer (type)                Output Shape              Param #
=================================================================
 input_ids (InputLayer)      [(None, 140)]             0

 tf_bert_model_10 (TFBertMod  TFBaseModelOutputWithPoo  167356416
 el)                         lingAndCrossAttentions(l
                             ast_hidden_state=(None,
                             140, 768),
                              pooler_output=(None, 76
                             8),
                              past_key_values=None, h
                             idden_states=None, atten
                             tions=None, cross_attent
                             ions=None)

 lstm_8 (LSTM)               (None, 100)               347600

 dense_8 (Dense)             (None, 1)                 101

=================================================================
Total params: 167,704,117
Trainable params: 167,704,117
Non-trainable params: 0
```

Fig. 3. Model's summary

5) Binary crossentropy: Binary crossentropy is a commonly used loss function where the output is binary. In our dataset our output 'target' is binary-1 if it's a disaster,0 if its not a disaster

6) Adam optimizer: Adam optimizer combines the best qualities of ADAGrad and RMSProp algorithms and works the best out of all other algorithms and is designed specifically for neural networks.

III. Results

A. Evalutation Metrics

1) Accuracy

This evaluation metric simply just tells us how many correct predictions our model has made. This is an easy to interpret evaluation metric and is hence chosen for its convenience.

Accuracy has its own limitations, which is that it is not very optimal for imbalanced datasets and as we have seen the slight imbalance in our dataset earlier, if we have more non disaster tweets and if our model keeps predicting that its not a disaster, our accuracy would be high. In such cases, we should consider other evaluation metrics such as F1, Precision and Recall.

2) F1-Score

This metric is used when we want to optimize precision and recall simultaneously. In a dataset like ours where there is an imbalance i.e. there are more number of non disaster tweets and predicting a non disaster tweet as a disaster can lead to misinformation, an evaluation metric like F1-Score mitigates the impact of false positives and is more sensitive to the minority class. This is advantageous as predicting the minority class (disaster tweets) is more crucial.

Predicting a non-disaster tweet as a disaster which is a FP is very dangerous as this leads to spread of misinformation.

B. Final Results

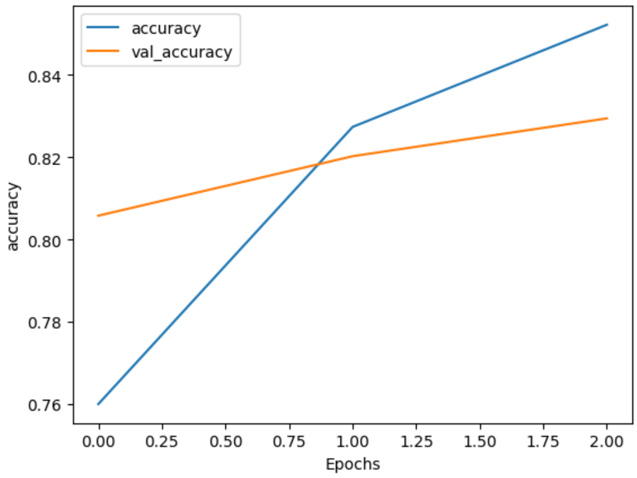

Fig. 4. Comparing validation accuracy and training accuracy

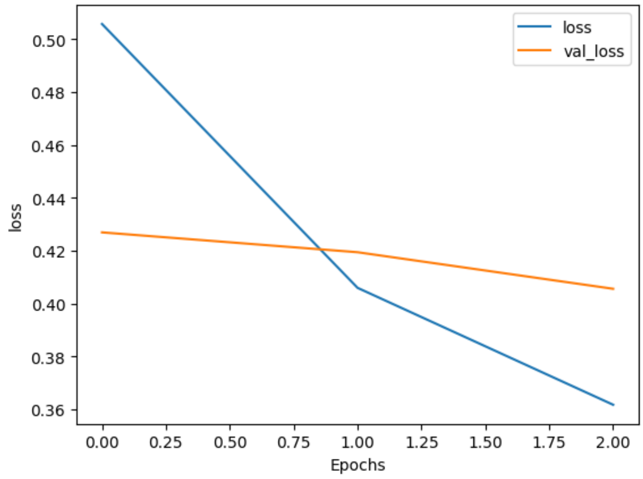

Fig. 5. Comparing validation loss and training aloss

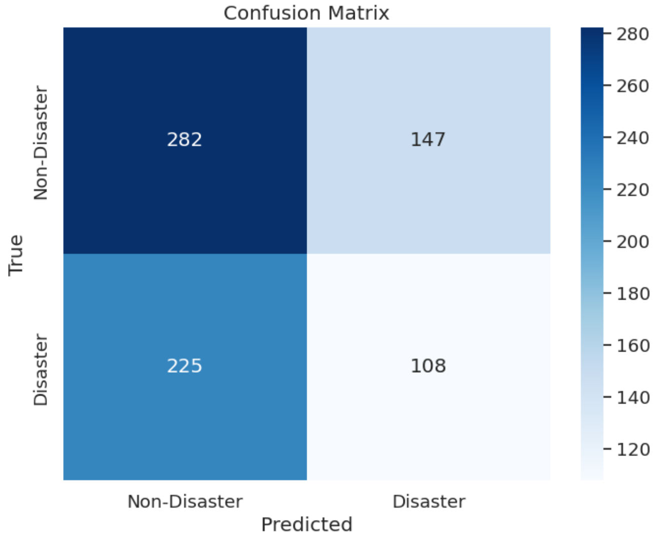

Fig. 6. Confusion Matrix

```
Classification Report:
              precision    recall  f1-score   support

           0       0.56      0.66      0.61       429
           1       0.43      0.33      0.37       333

    accuracy                           0.52       762
   macro avg       0.50      0.50      0.49       762
weighted avg       0.50      0.52      0.51       762
```

Fig. 7. Classification report of model

Hyperparameters

a)

IV. Conclusion

In this study, we have explored the potential of using Twitter data for disaster classification, an increasingly relevant and valuable resource for disaster management and early warning systems. By using various machine learning models and pre trained models like BERT, we established that they are highly classified in text classification.

Our results indicate that Twitter data can be successfully classified to identify relevant disaster-related content with a high degree of accuracy. We have demonstrated that Twitter can serve as a real-time information source during disaster events, providing early warnings and valuable insights.

In conclusion, this research underscores the value of social media like Twitter, in disaster classification. It offers a unique approach that has the potential to improve our ability to respond effectively to disasters, which adds to the safety and well-being of communities worldwide. By responsibly using the power of social media, we can harness the collective voice of the public, turning it into a valuable asset for disaster management and response.

References

[1] Huq, M R., Mosharraf, A., & Rahman, K. (2017, June 30). Data Analysis and Phase Detection During Natural Disaster Based on Social Data.

[2] Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia. (2017). Attention Is All You Need..

[3] I. Boglaev, “A numerical method for solving nonlinear integro-differential equations of Fredholm type,” *J. Comput. Math.*, vol. 34, no. 3, pp. doi: 10.4208/jcm.1512-m2015-0241.

[4] Zhang, Zhengyan & Han, Xu & Liu, Zhiyuan & Jiang, Xin & Sun, Maosong & Liu, Qun. (2019). ERNIE: Enhanced Language Representation with Informative Entities.

[5] R. K. Kaliyar, "A Multi-layer Bidirectional Transformer Encoder for Pre-trained Word Embedding: A Survey of BERT," *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2020, pp. 336-340, doi: 10.1109/Confluence47617.2020.9058044.

[6] P. Duraisamy, M. Duraisamy, M. Periyanayaki and Y. Natarajan, "Predicting Disaster Tweets using Enhanced BERT Model," *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2023, pp. 1745-1749, doi: 10.1109/ICICCS56967.2023.10142660.

[7] Y. Zhang, "Research on Text Classification Method Based on LSTM Neural Network Model," *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, Dalian, China, 2021, pp. 1019-1022, doi: 10.1109/IPEC51340.2021.9421225.

[8] X. She and D. Zhang, "Text Classification Based on Hybrid CNN-LSTM Hybrid Model," *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, 2018, pp. 185-189, doi: 10.1109/ISCID.2018.10144.

[9] J. Rathod, G. Rathod, P. Upadhyay and P. Vakhare, "Disaster Tweet Classification using ML," 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2022, pp. 523-527, doi: 10.1109/ICAAIC53929.2022.9792836.

[10] R. K. Kaliyar, "A Multi-layer Bidirectional Transformer Encoder for Pre-trained Word Embedding: A Survey of BERT," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 336-340, doi: 10.1109/Confluence47617.2020.9058044.