Deep Learning approach to predict disaster tweets using DistilBERT

Hridoy Pal, Subhram Satyajeet, Dhruvi Mukeshbhai Kharadi

pal61@uwindsor.ca, satyaje@uwindsor.ca, kharadi1@uwindsor.ca

School of Computer Science

University of Windsor

Abstract

Social media platforms are becoming increasingly popular. Social media platforms are working hard to make sure that people get every information on their platform including real-time information such as disasters. With the level of increasing demand for social media like X (formally known as Twitter), there is a growing interest in using deep learning approaches to classify tweets related to disasters. This research paper proposes a deep learning approach using the DistilBERT model to classify disaster tweets. DistilBERT is a lightweight version of BERT. The study explores some of the areas of Natural Language Processing techniques to train the DistilBERT model on a large-scale labeled dataset on disaster tweets and was able to achieve an accuracy of 84.84%. The results show that the DistilBERT model proves a satisfactory performance in predicting disaster tweets.

Keywords: Deep Learning, Natural Language Processing, Artificial Intelligence, LLM (Large Language Models), BERT

1. Introduction

As of today, social media has become part of people's everyday life. Social media play a significant role in times of emergency, like X (formally known as Twitter), Facebook, and so on. Twitter has become a useful tool to communicate in time of emergency. Twitter has over 528 million active users in 2023, and that will increase to over 625 million by 2028 [6]. We have seen that people post things on Twitter that are not entirely true or provide misleading information about anything. Because there is no method to verify what is being posted, anyone can post anything. Additionally, people tend to believe the information they see online, especially Twitter. Misleading information about any accidental disasters or natural disasters can cause chaos in society. Therefore, that is important to predict disaster tweets to supply meaningful information to people so that they have the information to make them safe, and it is not always clear whether a person meant disaster or not.

The text classification research domain has become one of the interesting and challenging domain of NLP (Natural Language Processing). This is also important to establish efficient algorithms with deep learning approaches. Contextual embedding methods have advantage to predict the text based on the context of the text. Therefore, they have proven better performance for text classification tasks.

Finally, disasters are happening continuously all over the world, and people are sharing information on twitter in real time. This data can be used to build a disaster monitoring system to respond quickly and provide step by step to people, and conduction rescue missions, etc.

1.2 Background Study

Le et al. [1], this study aim to successfully classify disaster tweets and develop a machine learning model to predict if a person is in danger or not. The experiment was performed on a dataset from the Natural Language Processing with Disaster Tweets competition. The dataset has a total of 10876 raw samples and is labeled as two classes: Not Disaster (0), and Disaster (1). Additionally, the paper proposed a Bert-based network for the classification task of disaster tweets. The accuracy of the model is 0.8867. They also used some machine learning models to show a comparison between ML models (MNB, LogisticRegression, RandomForest) and Bert fine-tuning. As their experiment shows, Bert's fine-tuning model outperformed the ML models for the text classification task.

Fontalis et al. [2], this paper shows the recent state-of-the-art developments of the deep learning approaches in disaster tweets by comparing state-of-the-art models and different datasets. This paper is focused on two different tasks, the first is to classify binary-class disaster tweets and multi-class disaster tweets. The overall results are achieved by DeBERTa for the first task with 0.8404 accuracy, and RoBERTa base for the second task with 0.7067 accuracy.

Danday et al. [3], the study proposed a model using DistilBERT to generate word embedding vectors and similarity matrix, to build a Graph-based Convolution Network to classify disaster tweets in crisis management. GCM models are being used in NLP because the GCM model outperforms the other machine learning models and deep learning methods. The accuracy of the proposed model at 100 epochs is around 76%. The study clearly shows that the proposed model outperforms this text classification approach as a tweet is a disaster or not.

1.3 Research Questions

What is disaster tweets classification?

Why disaster tweet classification is important?

What are some state-of-the art algorithms?

2. Methodology

2.1 Data Pre-processing

For data preprocessing, the following steps were executed:

1. A new column named “length” was added to the training dataset(‘df_train’) where each entry reflects the length of the corresponding “text” column.
2. A similar “length” column was added to the test_dataset (“df_test”).
3. Descriptive statistics are printed by the "length" column in the training and testing dataset, including count, mean, standard deviation, minimum, 25th percentile, median.

These statistics helped provide insights into the distribution of text lengths in the datasets, which further helped for understanding the data and making decisions about the preprocessing stage.

Post addition of the length column, several parameters were defined. Those are as follows:

1. **BATCH_SIZE:** For specification of number of samples in each mini batch during training.
2. **NUM_TRAINING_EXAMPLES:** Representation of the total number of training examples in the “df_train” dataset.
3. **TRAIN_SPLIT and VAL_SPLIT:** Indication of the fraction of data to be used for training and validation, respectively.
4. **STEPS_PER_EPOCH:** Calculation of the number of steps(batches) per epoch during training. It is derived based on the total number of training examples, training split and batch size.
5. **EPOCHS:** Specification of the number of training epochs, i.e., the number of times the entire dataset will be passed forward and backward through the neural network.
6. **AUTO:** Configuring TensorFlow to automatically choose the number of parallel calls during data loading for optimal performance.

Finally, after the definition of parameters, the training data was split into training and validation sets using the “train_test_split” function from scikit-learn.

2.1. Proposed Model

Steps Followed:

1. In the first step, the DistilBERT model was loaded and configured to have preset value as “distil_bert_base_en_uncased” which presets the model with a base architecture trained on English text in a case-insensitive manner.
2. A DistilBERT preprocessor is created from the loaded preset. The sequence length has been set to 160 tokens and the name “preprocessor_4_tweets” is assigned to this preprocessor. This sequence length represents the maximum number of tokens the model will process in a single input.
3. Then a DistilBERT classifier is created by specifying the preprocessor created in the previous step and the number of classification classes is set to 2 since our model is focussed on binary classification.

The steps followed initialized a DistilBERT model, adjusted the sequence lengths to have input sequences of up to 160 tokens and created a binary classifier using Keras NLP library.

DistilBERT is a lightweight version of BERT, it is faster and gives better performance compared to BERT. DistilBERT analyse the text and learns from it then transfers information to the next model [4]. DistilBERT was pre-trained on raw texts, with no human labelling [3], the number of pooling layers and embedding is 6 and output vector dimension of 786, shown at Fig 2.1 below.

Tokenizer (type)	Vocab #
distil_bert_tokenizer (DistilBertTokenizer)	30,522

Model: "distil_bert_classifier"

Layer (type)	Output Shape	Param #
padding_mask (InputLayer)	(None, None)	0
token_ids (InputLayer)	(None, None)	0
distil_bert_backbone (DistilBertBackbone)	(None, None, 768)	66,362,880
tf.__operators__.getitem (SlicingOpLambda)	(None, 768)	0
pooled_dense (Dense)	(None, 768)	590,592
classifier_dropout (Dropout)	(None, 768)	0
logits (Dense)	(None, 2)	1,538

```
Total params: 66,955,010 (255.41 MB)
Trainable params: 66,955,010 (255.41 MB)
Non-trainable params: 0 (0.00 B)
```

Fig 2.1: DistilBERT fine-tuning model, based on BERT Network

Post Model Creation steps:

The classifier model created is compiled using the following parameters:

- Loss Function: The loss function used here is SparseCategoricalCrossentropy.This is one of the popular choices for classification problems.
- Optimizer: The Adam optimizer is used with a learning rate of 1e-5.
- Metrics: The model's performance will be evaluated using accuracy during training.

Finally, the model is trained using the fit method, and the parameters configured are:

- Training Data: The model is trained using the dataset X_train and Y_train.
- Batch_size: This parameter uses the preset variable “BATCH_SIZE” that helps to efficiently use the computational resources.
- Epochs: This parameter is also set using the pre-configured “EPOCHS” variable in the preprocessing step that represents one iteration through the complete dataset.
- Validation Data: The second part of the data obtained after splitting the training data, i.e., “X_val”, “y_val” has been used in this parameter to evaluate the performance on the data after each epoch.

Upon the completion of training, the fit method returns an object that consists of all the details of the training process and progress of each epoch. It can be used to check the performance of the model through the progress of epochs by monitoring the loss and accuracy values.

Finally, to check the metrics of performance by the model, we use a confusion matrix that takes true labels and predicted probabilities of the classification and helps to visualize the accuracy along with some additional metrics.

Confusion Matrix: In the AI (Artificial Intelligence) models, confusion matrix displays the accuracy metrics of the models using true labels and predicted labels. The resulting values are then unpacked into variables as true negative, false positive, false negative and true positive. It provides insights into the model's ability to correctly classify instances.

2.3 Dataset

The dataset we used is from a Kaggle.com competition named Natural Language Processing with disaster tweets [5], and the data were collected from twitter. The dataset contains total of 10876 raw samples, with 7613 as train data and 3263 as test data.

Train data, shape = (7613, 5)

Test data, shape = (3263, 4)

```
Training Set Shape = (7613, 5)
Training Set Memory Usage = 0.29 MB
Test Set Shape = (3263, 4)
Test Set Memory Usage = 0.10 MB
```

Fig 2.3: Contents of Dataset

Additionally, the classification values were restricted to 0 and 1 where the value 1 represented if the tweet was based on a real disaster and the value 0 represented otherwise.

2.4 Evaluation Matrix

Accuracy is calculated as follows:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

F1 score is calculated as follows:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

and,

True Positive [TP] = your prediction is 1, and the ground truth is also 1 - you predicted a *positive* and that is *true*!
False Positive [FP] = your prediction is 1, and the ground truth is 0 - you predicted a *positive*, and that is *false*.
False Negative [FN] = your prediction is 0, and the ground truth is 1 - you predicted a *negative*, and that is *false*.

3. Results and discussion:

The accuracy of the model is 84.83%.

Model	Accuracy	F1 Score	Logloss
DistilBERT	0.8483	0.81	-
RandomForest	0.7898	-	-
LogisticRegression	-	-	0.486

Table 1: Comparison of ML Models and DistilBERT

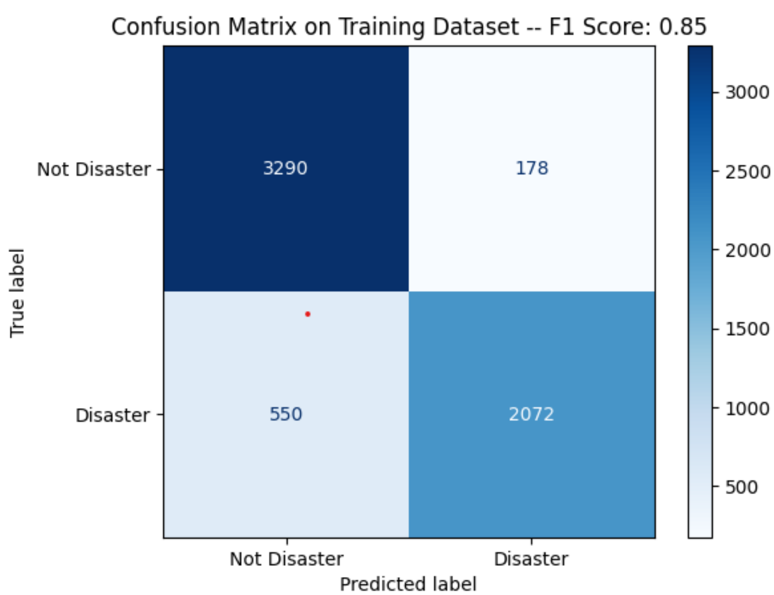

Fig 3: Confusion Matrix of Training data and F1 score

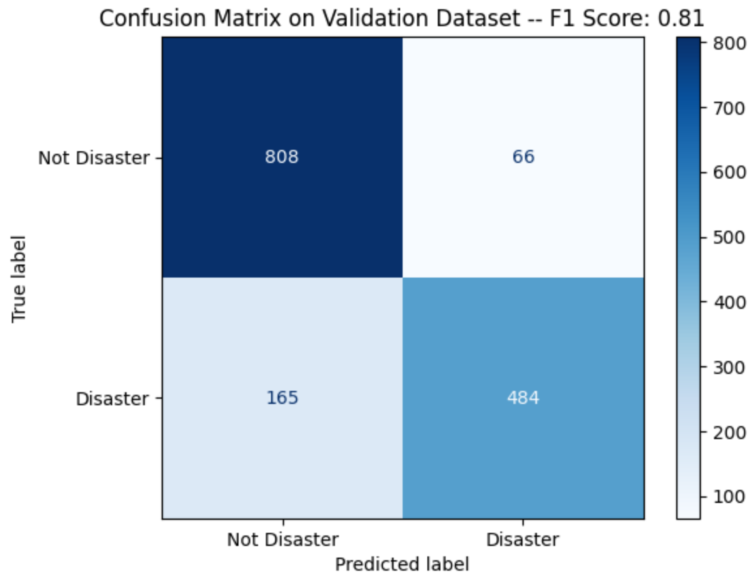

Fig 3.1: Confusion Matrix of Validation data and F1 score

We experimented with ML models, but the performance differences are visible. DistilBERT outperformed the existing ML models such as RandomForest and LogisticRegression, as shown in Table 1 above.

4. Conclusion

In conclusion, the deep learning approach to predict the disaster tweets successfully employed DistilBERT techniques to classify tweets as real disasters or not. With careful data preparation and model evaluation, the research was able to identify actual tweets related to disasters with impressive accuracy. The advanced features and functionalities offered by models like DistilBERT imply the rapid advancements taking place in NLP technology. The project highlights the potential of NLP in facilitating real-time disaster response by promptly interpreting the relevant information from social media. Continuous monitoring and ethical considerations remain crucial to ensure responsible deployment, fostering a reliable tool for crisis management.

5. Future work

The project can have several potential applications in the disaster-management domain. The project can be extended to incorporate cross-lingual analysis to interpret tweets in multiple languages that will enable global application in disaster response. Another extension can include the integration of analysis of images that are part of some tweets, which might help in better understanding of the disaster-based information.

References

[1] Le, Anh Duc. "Disaster Tweets Classification using BERT-Based Language Model." *arXiv preprint arXiv:2202.00795* (2022).

[2] Fontalis, Spyros & Zamichos, Alexandros & Tsourma, Maria & Drosou, Anastasis & Tzovaras, Dimitrios. (2023). A Comparative Study of Deep Learning Methods for the Detection and Classification of Natural Disasters from Social Media. 320-327. 10.5220/0011666500003411.

[3] Danday, Anusha & Murthy, T. (2022). Twitter Data Analysis using Distill BERT and Graph Based Convolution Neural Network during Disaster. 10.21203/rs.3.rs-2041154/v1.

[4] Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." ArXiv abs/1910.01108 (2019): n. pag.

[5] Addison Howard, devrishi, Phil Culliton, Yufeng Guo. (2019). Natural Language Processing with Disaster Tweets. Kaggle. https://kaggle.com/competitions/nlp-getting-started

[6] Rohit Shewale, (2023). Ddemandsage. https://www.demandsage.com/twitter-statistics/

[7] Devlin, Jacob & Chang, Ming-Wei & Lee, Kenton & Toutanova, Kristina. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[8] Koshy, R., Elango, S. Multimodal tweet classification in disaster response systems using transformer-based bidirectional attention model. Neural Comput & Applic 35, 1607–1627 (2023). https://doi.org/10.1007/s00521-022-07790-5

Team's Contribution:

Each member of the team equally contributed to the project and the report. Members are as follows:

Hridoy Pal - 110146466

Subhram Satyajeet - 110127932

Dhruvi Mukeshbhai Kharadi - 110127398