Disaster Tweet Classification

Bhuvan Kumar
Undergraduate, CSE Dept
PES University
Bangalore, India
bhuvarsenal@gmail.com

Prajwal Kamath K
Undergraduate, CSE Dept
PES University
Bangalore, India
kamathprajwal12@gmail.com

Pranav Rajnish
Undergraduate, CSE Dept
PES University
Bangalore, India
pranavrajnish2312@gmail.com

***Abstract*—Social media and twitter in particular has evolved into a sort of public forum where massive amounts of information is shared. In todays digital world, news and live events are broadcast all over social media in real time. Twitter has become an important communication channel in times of emergency. The ubiquity of smartphones enables people to announce an emergency they're observing in real-time. Because of this, more and more agencies are interested in programatically monitoring Twitter (i.e. disaster relief organizations and news agencies).**

The first-responders play an important role in any crisis and with their help we can quickly send aid where its needed.

In this project we have compared various methods of classifying disaster tweets namely, BERT transformer model, LSTM, Logistic Regression, Naive Bayes and Random Forest model. We find that the deep learning models perform better than the machine learning models and BERT is found to perform the best.

By accurately and quickly predicting if a tweet is related to a disaster or not we can send aid and disaster relief to the affected site.

***Index Terms*—Disaster tweet classification, BERT, tweet**

I. Introduction

Social media sites provide an important and quick understanding into a situation as it unfolds in real time. Research has found that the general public use SM applications during disasters to communicate information regarding urgent needs, infrastructure damage, injured or dead people, volunteering or donation efforts, and situational updates (Kumar et al. 2019; Madichetty and Sridevi 2019; O'Keefe and Alrashdi 2018; Alam, Joty, et al. 2018). Timely access to SM data can be leveraged for emergency response in the first few hours to significantly reduce both human loss and economic damage (Alam, Joty, et al. 2018).

The main challenge is to detect the relevant messages in a sea of data. The vast amounts of tweets make it hard to filter out the useful tweets that we are after, especially in ambiguous cases. As a result, a significant portion of the collected tweets can be irrelevant. Therefore, detecting disaster-related tweets is commonly modelled as an automatic classification task and tackled with machine learning algorithms and more recently with deep learning algorithms.

Our project aims to perform a binary classification of whether a tweet is related to some disaster or not. The disaster could be a natural disaster like a landslide or flood, or it could be a man-made disaster like a fire or terror attack.

The dataset we have used is a publicly available kaggle dataset consisting of around 11370 labelled tweets. The main classifier we are exploring is the BERT classifier. BERT is an open source machine learning framework for natural language processing (NLP). BERT is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context. Another deep learning model we have used is LSTM, which is commonly used for NLP task as it is able to remember key words that came before to establish context.

Some of the machine learning models used are logistic regression, naive bayes and random forest model. These serve as a good comparison point between the deep learning and machine learning models.

The rest of the paper is structured as follows: The Related Work section covers a literature survey of current methods of classifying disaster tweets. The Proposed Methodology goes over the data collection and pre-proccessing phase. It also covers our solution to this problem and details the steps taken achieve this solution. The Experiment Results section goes over the results obtained from implementing this solution. The Conclusion section summarizes the paper and has the closing thoughts. The References section highlights all the papers and sources we haver referenced in this paper.

II. Related Work

[1] aims to aims to classify tweets that contain disaster-related information using the BERT model and BERT-based LSTM. BERT is an open source machine learning framework for natural language processing (NLP). BERT is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context. BERT, which stands for Bidirectional Encoder Representations from Transformers, is based on Transformers, a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection. The Text is converted to lower case and also non ASCII characters are removed. Texts with length less than 4 are not used. For BERT, an additional [CLS] token is inserted to the beginning of each text. This paper compares the results of 5 different models namely: 1) Baseline Model (LSTM) 2) Default BERT 3) BERT + Non-linear layers 4) BERT + LSTM 5) BERT + CNN The Dataset used to train this model is the CrisisLexT26 as well as CrisisNLP. 5 metrics which are accuracy, Matthews correlation coefficient, precision, recall, F1-score, are considered when

evaluating a model. The Batch size is 32. Initial learning rate is set to 0.001 for non-BERT parameters and 0.00002 for BERT par'meters. Patience is 5, after reaching which learning rate decay by 50 percent. The maximum number of learning rate decay trials is 5 before early stopping. Adam optimizer is used for all models' training. Cross entropy loss is the loss function. Some of the key takeaways from this paper is that The BERT + NL, BERT + LSTM and BERT + CNN models have higher score than the baseline model. The default BERT model has higher recall than all of the models. The paper reports that in some cases, the model outperforms human classification. A drawback of this paper is that the model seems to struggle in cases of ambiguous tweets, which results in a lower score.

[2]In this work, they have collected 192, 948 tweets by combining a number of such datasets, preprocessed, filtered and duplicate removed, which resulted in 117, 954 tweets. Then evaluated the performance of multiple ML algorithms such as Logistic Regression (LR), Decision Tree (DT), SVM, NB, ANN and RF,Gradient Boosting Classifier (GB), Ridge-Classifier, AdaBoost, k-Nearest Neighbors (KNN), xgboost, and catboost and dl algorithms such as CNN and Bi-directional LSTM (Bi-LSTM). in classifying disaster-related tweets in three settings, namely "in-disaster", "out-disaster" and "cross-disaster".

They conduct experiments under three settings to evaluate twelve ML models and two DL models with three different word embeddings for the disaster-related tweet classification task. Three Settings are:- • In-disaster: training and test data belong to the same disaster type. • Out-disaster: training and test data belong to different disaster types. • Cross-disaster: training set consists of tweets of various disaster types.

We extracted tweets from Disaster Data Corpus 2020 created by Wiegmann et al. 2020, that includes data from seven repositories, namely, CrisisLex T26, CrisisLex T6 , CrisisNLP - RESOURCE 1, CrisisNLP - RESOURCE , CrisisNLP - RESOURCE 5 , Epic Annotations, and the dataset collected by (McMinn et al. 2013).

The performance of algorithms were measured using average F1-score. To reduce the number of training and evaluations, they selected the top three ML models and best DL model based on average F1-score for the out-disaster and cross-disaster experiments.

We notice F1-score for ML and DL algorithms for in-disaster experiments range from 0.49 to 0.92. DL models outperform the traditional ML algorithms. This finding supports previous research by (D. Nguyen et al. 2017) and (Burel and Alani 2018) who showed that DL classifiers performed better than all non-DL classifiers. It seems possible that word embedding performs better than the BOW and TF-IDF representations. Among the DL models, the Bi-LSTM model with Word2Vec features has performed the best across all three experimental settings. Another important finding is that with the default parameters, the KNN algorithm has performed the worst for all three experiments This is the largest study so far ever conducted, evaluating around 0.2 million labelled tweet dataset. The evidence from this study suggests that classifiers can be trained to identify disaster-related tweets in all three categories. A few Drawbacks from this paper include, the training time for the DL algorithms were higher than the classical ML models. These findings are limited by using default parameters for the ML algorithms and considering only English tweets

[3] In this paper,they have Implemented seven different conventional machine learning and five different deep learning models and training and testing them with four different disaster related datasets.They have also investigated the role of the TF-IDF feature and two different word embedding vectors in the classification task.Compared the performance of conventional machine learning and deep learning in case of imbalanced data distribution.

They have used seven different machine learning classifiers and five different deep neural networks based models for system development. In case of machine learning the following classifies are used: (i)Support Vector Machine (SVM), (ii) Random Forest (RF), (iii)Logistic Regression (iv) K-Nearest Neighbors (KNN),(v) Naive Bayes (NB), (vi) Gradient Boosting (GB), (vii)Decision Tree (DT). We have used following models for deep learning-based classifiers: (i) Convolution Neural Network(CNN), (ii) Long-Short-Term-Memory (LSTM), (iii) Gated Recurrent Unit (GRU), (iv) Bi-directional Gated Recurrent Unit (Bi-GRU), (v) Gated Recurrent Unit-Convolution Neural Network (GRU-CNN) The dataset used in this study includes tweets related to seven different disasters: (i) Maria Hurricane , (ii) Harvey Hurricane, (iii) Irma Hurricane, (iv) Iran-Iraq earthquake, (v) Mexico earthquake, (vi) Sri Lanka flood (vii) California wildfire The dataset is collected in such a manner that, if the tweet has more than one image URLs, then those images were saved with the same tweet text, so the published dataset contains several duplicate tweet text. Duplicate tweet text is then removed from the dataset. Finally, they merged eventspecific datasets into one, which means all hurricane datasets are merged into one. All the punctuation marks were removed. In the case of conventional machine learning algorithms Term Frequency-Inverse Document Frequency (TF-IDF) vector is used as the input to the classifiers. All the possible combinations of 1-gram, 2-gram, and 3-gram TF-IDF features were extracted to experiment with all classifiers. In the case of the deep neural network, two different pretrained word vectors GloVe [17] and Crisis [18] are used. In the case of GloVe, 100-dimensional word vector embedding whereas in case of Crisis embedding 300-dimensional word vector embedding is used. The categorical cross-entropy and Adam is used as the loss function and optimizer respectively for each of the neural network models. For all the hidden layer, they have used ReLU activation function whereas, at the output layer, they have used the softmax activation function in each of the neural networks. The performance of the models is measured in terms of Precision (P), Recall (R) and F1-score (F1). Among seven different classifiers: SVM, RF, LR, KNN, NB, GB, and DT, Gradient Boosting (GB) classifier performed best in the case of all the disaster-related events whereas SVM performed worst. The Gradient Boosting classifier achieved an F1-score of 0.79, 0.80, 0.70, and 0.67 for Hurricane, Earthquake, Flood, and wildfire events respectively. For the experiments performed with deep neural networks, GloVe embedding performed best in case of wildfire. In the case of the hurricane, both GloVe and Crisis gave comparable results whereas in case of floodboth the embedding techniques gave a mixed performance. Deep neural network-based models outperformed the conventional machine learning techniques. In the case of Hurricane, Bi-directional Gated Recurrent Unit (Bi-GRU) performed best. In the case of Earthquake, the combination of the Gated Recurrent Unit and Convolution Neural Network (GRU-CNN) performed best. In the case of Flood, Long-Short-Term-Memory (LSTM) and Gated Recurrent Unit (GRU) both equally performed best,whereas, in the case of Wildfire, Gated Recurrent Unit with Convolution Neural Network (GRU-CNN) performed best. Some of the Key takeaways include, it compares the effectiveness of conventional machine learning with deep learning networks to see how classifiers perform across each class. They use a pre-trained word vector that reduces the computational overhead of the model. A few drawbacks include, for all the deep neural network-based model, the experiments were performed with fixed batch size, learning rate and optimizer. Only English tweets were used for the classification.

In [4] Their system uses Twitter data and performs parsing, domain specific classification, and sentiment analysis. The proposed system has also found overlap of information in short text by using precise filtering on tweets. To extract tweets, we use Archivist, a service that uses Twitter Search API to find and archive tweets. For extraction of keywords, entities and sentiments we used Alchamy API. Alchemy API utilizes natural language processing technology and machine learning algorithms to analyze content. It can extract keyphrases, named entity, and topic level sentiments. Keyphrases are actually metadata of text returned by Alchemy API. Alchemy API is capable of identifying people, companies, organizations, cities, geographic features, and other typed entities within text. It can extract 28 types of entities from text which contains hundred of further subcategories. A series of statistical algorithms are combined with a huge data-set describing the world's objects, individuals, and locations. It also combine subtypes of entity which provides detailed ontological mappings for an entity, for instance identifying a Person as a Politician or Athlete. The details on component wise processing and analysis of tweet data are explained in the following subsections.

The objective is to extract valuable information from tweets and classify the tweets into different categories based on the knowledge contained in them. The collected tweets are given to Alchemy API. It accepts unstructured text, processes it using natural language processing and machine learning techniques, and returns keywords and sentiments of users about keywords. The proposed system extracts participating keywords and their associated sentiments using Alchemy API which is able to extract sentiments at topic level.

The proposed system has processed 40,000 tweets of different categories for testing and verification. By considering the keywords returned by Alchemy API, 3874 diabetic tweets were classified from all categories. However, when the proposed knowledge enhancer and synonym binder in addition to knowledge generator are applied then the proposed system has classified 8636 diabetic tweets from all categories because knowledge enhancer extract entities from tweets which are important for categorization. Its accuracy has increased in a range of 0.1 percent to 55 percent for different categories as shown in Table IV. Information gain is due to verb, entities and their synonyms in tweets which are extracted by the proposed system; however, missed by Alchemy API as keywords. To increase this information gain, our system has collected verb and entities from tweets, bind synonyms with entities and keywords and have applied filtering by addition of verb and entities with keywords and their synonyms.

Hence,finally they have demonstrated a system to extract knowledge from tweets and then classify tweets based 6 on the semantics of knowledge contained in them. For avoiding information loss, knowledge enhancer is applied that enhances the knowledge extraction process from the collected tweets. The maturity of knowledge gained using knowledge enhancer module has helped to filter tweet more precisely avoiding information loss. They have also measured missing information during specific keyword-based search and then proposed a method to collect more precise information about specific topic or domain. Sentiment analysis shows people attitude towards different topics. This data can also help to generate richer user profile and generate valuable recommendations. In future they are planing to integrate the proposed system with personalized profile management, sentiment analysis, and recommender system.

III. Proposed Methodology

We have decided to use three Machine learning models and two Deep learning models.In Machine learning we have used Naive Bayes,Logistic Regression and Random Forest models.In Deep learning we have used LSTM and Bert models.

A. Dataset

The Dataset[5] was collected from Kaggle which was inherited from Disasters on social media.It contains 11370 tweets associated with disaster keywords like "crash", "quarantine", and "bush fires" as well as the location and keyword itself.Of the above tweets 9256 tweets are found to be non disastrous tweets and 2114 tweets are found to be disastrous tweets.The top 10 Keywords are found to be as follows ws

B. Preprocessing

We started by converting all the words into lowercase characters. Then Tokenization is performed which is the process by which a large quantity of text is divided into smaller parts called tokens. Stopwords are then removed with the help of nltk package. Part of Speech tagging is performed defined as the process of assigning one of the parts of speech to the given word. Finally Lemmatization is performed which reduces the

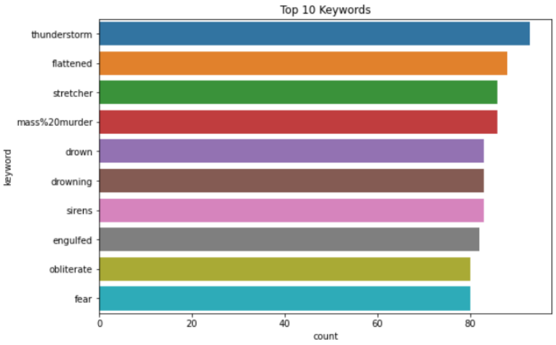


Fig. 1. Top 10 Keywords

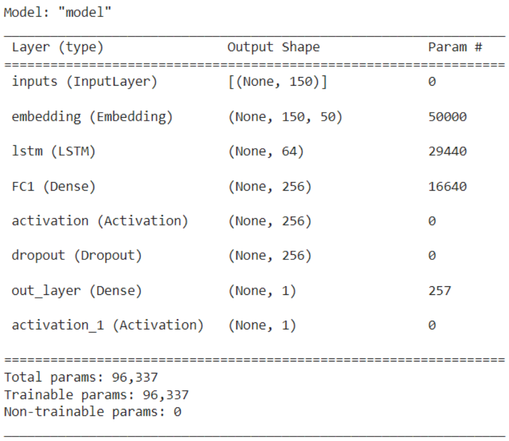
inflected words properly ensuring that the root word belongs to the language and gives its dictionary word

C. Machine Learning Models

We used Naive Bayes Classifier as it doesn't require much training data and handles both continuous and discrete data. It is highly scalable with the number of predictors and data points.Term Frequency-Inverse Document Frequency (TF-IDF) vector is used as the input to the Multinomial Naive Bayes classifier. Then we used Random Forest as it provides a higher level of accuracy in predicting outcomes.Finally we used Logistic Regression as it helps to predict binary outcome based on prior observations of a data set.Term Frequency-Inverse Document Frequency vector is passed as input to the above two models.

D. Deep Learning Models

1) LSTM: We implemented a simple LSTM model to contrast with the machine learning models. For the pre-processing stage, we have used the tokenizer class from keras to convert the list of words to a set of sequences. These sequences are padded and the sent to the model.

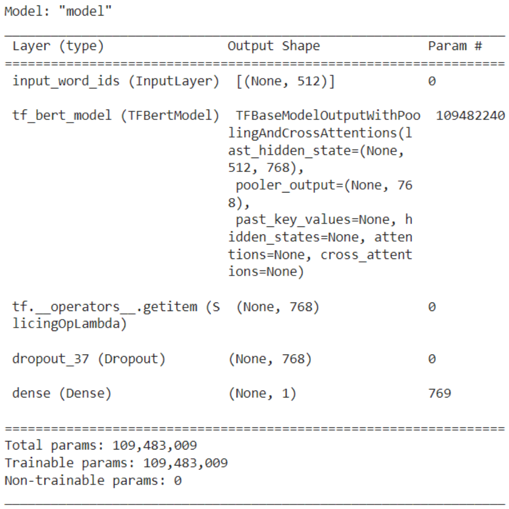


```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 inputs (InputLayer)         [(None, 150)]             0         
                                                                 
 embedding (Embedding)       (None, 150, 50)           50000     
                                                                 
 lstm (LSTM)                 (None, 64)                29440     
                                                                 
 FC1 (Dense)                 (None, 256)               16640     
                                                                 
 activation (Activation)     (None, 256)               0         
                                                                 
 dropout (Dropout)           (None, 256)               0         
                                                                 
 out_layer (Dense)           (None, 1)                 257       
                                                                 
 activation_1 (Activation)   (None, 1)                 0         
                                                                 
=================================================================
Total params: 96,337
Trainable params: 96,337
Non-trainable params: 0
_________________________________________________________________
```

Fig. 2. LSTM Model

The model consists of: Input layer, Embedding layer, LSTM layer, Dense fully connected layer 1, Activation function relu, Dropout layer to decrease overfitting, Dense fully connected layer 2, Activation later sigmoid. The model has a total of 96,337 parameters, all of which are trainable. The loss function used is binary cross entropy and the optimizer is RMSprop. We then fit it onto our sequences matrix, with a batch size of 128, 20 epochs and Early Stopping callback to decrease overfitting.

2) BERT: It is a really powerful language representation model.We are using the Bert-based-uncased pretrained model.The Hugging Face transformers library provides an easy integration of Bert pre-trained model.First we tokenize our Tweet's and encode our input using fast-tokenizer.Then using fast-encode we encode the text to numbers so that it can be used for model training.We then prepare the dataset for training by creating batches of data with Batch-size as 64 which we pass to our model.



```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 input_word_ids (InputLayer)  [(None, 512)]            0         
                                                                 
 tf_bert_model (TFBertModel)  TFBaseModelOutputWithPoo  109482240
                             lingAndCrossAttentions(l            
                             ast_hidden_state=(None,             
                             512, 768),                          
                              pooler_output=(None, 76            
                             8),                                 
                              past_key_values=None, h            
                             idden_states=None, atten            
                             tions=None, cross_attent            
                             ions=None)                          
                                                                 
 tf.__operators__.getitem (S  (None, 768)              0         
 licingOpLambda)                                                 
                                                                 
 dropout_37 (Dropout)        (None, 768)               0         
                                                                 
 dense (Dense)               (None, 1)                 769       
                                                                 
=================================================================
Total params: 109,483,009
Trainable params: 109,483,009
Non-trainable params: 0
_________________________________________________________________
```

Fig. 3. BERT Model

In BERT model the first layer will be the pre-trained BERT model followed by our own network layers.Additional CLS token is added to the beginning of each text.We Then the transformer output is passed through a dropout to take care of overfitting, if any. We then pass the output through a dense layer with sigmoid activation. We create a model using the Model function with the appropriate parameter values for inputs and outputs as defined above. We compile the model with Adam optimizer and binary cross entropy loss function.

E. Live Tweet Analysis

An interesting aspect of out project is that we used snscrape to scrape the live tweets from Twitter.We scraped around 200 tweets and used the above BERT model to predict whether the tweet is disastrous or not instantaneously.

IV. Experiment Results

The performance of our model is measured in terms of Accuracy Score.We observe that Deep learning models perform better than machine learning models.BERT performs the best with accuracy of 91.8.LSTM performs next best with accuracy of 87.8.Machine learning models namely Naive Bayes gives us accuracy of 86.6 ,Random Forest 87.5 and Logistic Regression with accuracy of 87.We managed to classify live tweets accurately using which aid can be sent instantaneously.

V. Conclusion

This study investigated the performance of different machine learning and deep learning models to classify whether certain tweets fall under disaster or non-disaster. The research suggests that the BERT model performed the best, achieving an accuracy of 91.82 percentage. We further extended our code to perform live tweet analysis with the BERT model.

References

[1] Guoqin Ma,"Tweets Classification with BERT in the Field of Disaster Management"

[2] Nilani Agriyage, Raj Prasanna, Emma E H Doyle " Identifying Disaster Related Tweets: A Large Scale Detection Model Comparison" - July 2021

[3] Abhinav Kumar,Jyoti Prakash Singh,Sunil Saumya 'A Comparative Analysis of Machine Learning Techniques for Disaster-Related Tweet Classification',2019 IEEE Region 10 Humanitarian Technology Conference Depok, Indonesia

[4] Rabia Batool, Asad Masood Khattak,Jahanzeb Maqbool ”Precise Tweet Classification and Sentiment Analysis"