Breaking News or Noise? Unveiling the power of BERT, RoBERTa, DistilBERT and other Deep Learning Models for Disaster Tweet Classification

Siyam Sajnan Chowdhury
School of Computer Science
University of Windsor
Windsor, Canada
chowdh1b@uwindsor.ca

Kimia Tahayori
School of Computer Science
University of Windsor
Windsor, Canada
tahayor@uwindsor.ca

***Abstract*—Tweets pertaining to disasters are tweeted by various users on Twitter, the popular social media platform, currently rebranded to X. These tweets could potentially contain crucial information regarding people who are hurt or injured and also very important information such as location, infrastructural damage, and extent of injuries. Government agencies and humanitarian aid groups could use this information to arrange operations to rescue and save lives. The number of these tweets are very large and to effectively use the information from tweets, a filtering process has to be employed to filter out the junk information and utilize the information from the real disaster tweets. In this paper, we have used and compared various conventional deep learning algorithms and transformer-based deep learning algorithm to classify disaster-tweets into two classes, true disaster-related tweets, and false disaster-related tweets. The transformer-based deep learning models achieved accuracies of around 82-84% compared to the conventional deep learning models accuracies of 57-72%. This illustrates the superior performance of these transformer-based models compared to conventional deep learning models.**

***Keywords*—BERT, Disaster, Tweets, Deep Learning, RoBERTa, DistilBERT.**

I. Introduction

In these times marked by unmatched and unparalleled connectivity by means of social media platforms, they have also surfaced as an essential mechanism to spread real-time information about disasters and crises – be it natural or man-made. These mediums play a crucial role in disaster response and rescue because, when emergency hotlines are overloaded with frantic callers pleading for help, social media remains the only available option for stranded people to express their struggling state and share their location and needs [1] – [4], with tweeting out their locations and needs on Twitter being one of the ways to send distress signals via social media [5]. The short and concise nature of tweets make it especially suitable to communicate about their perils efficiently and the immediacy of tweets enables it to go beyond the geographic boundaries and has the potential to reach thousands of people and even more. During these distressing times, people have been rescued from the most extraordinary places based on just their tweets such as - from tweeting out from under the rubbles of a building collapsed in an earthquake [6] or tweeting out about being stuck in a flood affected area after being struck by a hurricane [7]. There have been numerous instances such as these in the past decade where people have been rescued from these terrible situations after being struck by some disasters where it would, otherwise, have been almost impossible to track down their location and save them on time. Twitter also helps amplify the pleads and cries for help manifold by its sharing, resharing and hashtag features thus helping it reach to the necessary authorities.

Twitter helps stranded users provide dynamic information in real-time such as their location, updates on their situation and specific requests for aid be it food, water, or rescue. They could also provide potential rescuers and aid providers with information regarding their immediate surrounding as well as any potential barriers to reaching that location.

These information on Twitter are invaluable to government rescue agencies and humanitarian and social service groups as to how to organize rescue missions and how to effectively prioritize the rescue operations. However, the sheer volume of these tweets makes it difficult for these organizations and agencies to manually sort through the important ones that are genuine distress signals, especially the ones with important information that is crucial to ones' rescue from other random tweets that are merely irrelevant or even detract others with misinformation about said rescue. Some examples of informative and authentic disaster tweet and non-informative tweets are presented in Table 1.

This is where intelligent systems can be used in order to classify, within seconds, in real-time, the authentic disaster tweets from the irrelevant and unauthentic ones. Various machine learning and deep learning models have been employed in this domain for these type of classification tasks with good success.

In this work, we have conducted a comparative analysis of different conventional deep learning models and more recent transformer-based deep learning models. We have used 4 conventional deep learning models - Convolutional Neural Network (CNN) [8], Long Short-Term Memory (LSTM) [9], Bidirectional LSTM (Bi-LSTM) [10], and Gated Recurrent Unit (GRU) [11] and 3 transformer-based deep learning models - Bidirectional Encoder Representation from Transformers (BERT) [12], Distilled BERT (DistilBERT) [13], and Robust BERT Approach (RoBERTa) [14] on disaster classification. The models were trained and tested on a dataset [15] containing information about different types of

tweets about different types of disasters with the keywords specified in numerous instances.

The contributions of this paper are summarized as follows:

- Implemented 7 different deep learning models with 3 of them being more recent and powerful transformer-based model.
- Compared the performance of the conventional deep learning models with the transformer-based deep learning models.

The rest of the paper is structured as follows: Section II discusses the problem statement, Section III discusses the relevant literature, Section IV explains the detailed methodology, Section V states the experimental results and discusses the analysis, Section VI provides the conclusion with future research possibilities in this domain

TABLE 1. INFORMATIVE TWEETS AND NON INFORMATIVE TWEETS

Tweet Number	Informative Disaster Tweets
1	Emergency crews respond to chemical spill downtown beaumont #benews http://t.co/PME0HOJVYA
2	USA: BREAKING NEWS: CHEMICAL SPILL/EVACUATIONS/RED CROSS EMERGENCY http://t.co/007Npen6LG
3	Emergency 4 Harbor City Mod v4.5.2 #6 Chemical Fire in Residential Area!: http://t.co/uLuPxYzJwV via @YouTube
4	RP said they can see smoke coming from the silo on 260th Street in Hartford but no flames.
	Non-informative Disaster Tweets
5	Nueva favorita: EmergeNCY feat. The Chemical Brothers / My Bits http://t.co/MET4YtZMFB @DeezerColombia
6	@Chemical_Babe its a family emergency so I can't make it unless I have a chance to use by phone for stream.
7	I rated Catastrophe (2015) 8/10 #IMDb - hilarious! http://t.co/cjrSSRY1RT
8	failure is a misfortunebut regret is a catastrophe

II. RESEARCH QUESTIONS

In this section, we will state and define the research questions that we seek the answers to. Our investigation into disaster tweet prediction using different deep learning models is guided by a set of comprehensive questions that we wanted to find the answers to with our research.

The first research question that we asked was how conventional deep learning models' performance vary across the different models CNN, LSTM, BiLSTM and GRU with regards to the metrics precision, recall, F1-score, Accuracy and Area under Receiver-Operator Characteristics curve (AUC-ROC) in the domain of disaster related tweets. These sequential models have all been used for disaster tweet classification but their performances were yet to be compared relative to each other so we looked into the performances of these models.

The second research question that we looked to answer was how much transformer-based deep learning models can improve the performance from traditional deep learning-based models in the domain of classification of disaster-related tweets. We also wanted to find out whether certain models were more robust in ascertaining and figuring out the nuanced patterns of texts pertaining to disasters.

Building on the second question, we wanted to find out the efficacy of transfer learning in enhancing the performance of the pre-trained models. We pursued to answer the extent to which transfer learning could be utilized in the classification of disasters.

Finally, we wanted to find out which of these deep learning models would show the highest performance regardless of the type of deep learning model.

III. RELATED WORKS

In this section, we briefly list down the relevant papers that we considered to gain further insights into this domain and how this domain progressed over the years.

Vieweg et al. [16] was one of the initial papers that provided with an ontology for catastrophes and proposed a robust methodology to categorize and identify tweets and cluster them into categories that would be used to format the information that would aid large humanitarian organizations.

Purohit et al. [17] designed Twitris v3 which focuses on the geotagging feature of tweets as its key base from which the algorithm could carry out classification utilizing semantic enrichment.

Kongthon et al. [18] used tweets in order to analyze and interpret useful tweets from the 2011 flood that ravished the shores of Thailand.

Caragea et al. [19] used CNN to distinguish useful informative tweets from uninformative ones in the domain of disaster detection.

Nguyen et al. [20] went further than Caragea et al. [19] and proposed a methodology that utilizes MLP and CNN.

Aipe et al. [21] developed a deep CNN network for multi-label classification in this domain. Their model segregated the tweets that were specific to some crisis from the ones that are not by augmenting the model with features that were relevant and specific to the particular crisis domain.

Mendoza et al. [22] looked into whether information obtained from social media regarding natural disasters could actually be trusted. They looked into the dissemination of false and fake news on social media during an earthquake that shook Chile. The situational tweets were identified that contained information regarding the issue while the others were not situational.

Caragea et al. [23] utilized the geo-tagged feature and map feature of twitter in order to map out a model that could classify twitter posts into multiple categories such as information of the affected people, damage to infrastructure as whole and other information that could be used be potential rescuers.

Madichetty et al. [24] used a hybrid CNN-ANN architecture to classify informative tweets from non-informative ones from a dataset based on Hurricane Harvey tweets where they used CNN for extracting the features from the disaster dataset and used ANN for classification.

Ningsih et al. [25] looked into disaster tweet classification using pretrained BERT model obtaining a 79% accuracy.

Madichetty et al. [26] also looked into disaster tweet classification with tweets containing both text and image data where they used pre-trained RoBERTA model for processing

the text part of the tweet and generating a probability vector and VGG-16 CNN model for generating a feature vector from the images. They then used these two and combined them using a multiplicative fusion technique and they used this to give the final classification prediction.

IV. Methodology

This section focuses on explaining the dataset used as well as the methodology that has been employed in order to classify the disaster tweets. We have used 4 conventional deep learning models for the classification of disaster tweets: (i) Convolutional Neural Network (CNN), (ii) Gated Recurrent Unit (GRU), (iii) Long Short-Term Memory (LSTM), (iv) Bi-Directional LSTM. We have also used 3 transformer-based deep learning models for the classification task: (i) Bidirectional Encoder Representation from Transformers (BERT), (ii) Distil BERT (Distilled BERT), (iii) Robustly Optimized BERT Approach (RoBERTa).

A. Dataset Description

The dataset is created by web scraping from twitter and created by the company figure-eight [15]. The dataset comes split into test and training data. 10% of the train dataset is then split into the validation set. The training dataset contains 5 columns – id, keyword, location, text, and label. The id column contains the unique identifiers. The keyword columns contain certain keywords contained in the tweet that has the most impact on the target. The location column contains the location information which came with the tweet. The text column contains the actual tweet exactly as it was posted. Finally, the label column contains the labels 1 or 0 – 1 indicating that the tweet is, in fact, a disaster-related tweet and 0 meaning that, the tweet is not a disaster-related tweet.

B. Data Preprocessing

The data preprocessing applied is uniform across all the models in order to maintain a consistent and uniform base to compare the different models.

We created a common function to run all the basic preprocessing steps. The first step in the preprocessing is to convert the tweet into all lowercase. We then used regular expression for the following preprocessing steps. Then we substituted all the URLs in the tweet with a space. We replaced all ticker symbols with a space too. We removed all the characters that were not either a letter or an apostrophe with a space. After that, we removed all the single letter words. We then removed multiple spaces in a row to just a single space. We then used the string strip method to remove any leading or trailing whitespace characters.

We then use the Keras preprocessing module to perform tokenization breaking down the strings into smaller tokens, followed by text to sequence conversion converting the sequence of words into numerical representation. The sequence is then padded to ensure uniformity in the length of the sequences with the maximum length set to 128.

C. Conventional Deep Learning Models

Conventional or traditional deep learning models encompass the models with neural network architecture that predates more recent architectures such as transformer-based ones. Notable among these are Feedforward Neural Networks, CNN, Recurrent Neural Networks, LSTM, GRU, Restricted Boltzmann Machines and others. In this paper, we employed 4 of these conventional models – CNN, LSTM, BiLSTM and GRU. These models are briefly explained:

1) CNN

CNNs are commonly used for image classification but it has been slightly adapted to be used for text classification tasks. The input text data is represented as a 2d matrix representation which is then converted into dense vectors using an embedding layer by capturing the semantic relationship between different words. This is the alteration done to adapt it to natural language processing (NLP) tasks. This step is followed by a convolutional layer where filters are convolved about the whole vectors to capture patterns and features. This is followed by activation function, generally Rectified Linear Unit (ReLU) to introduce non-linearity, enabling learning of complex patterns by the model. The equation of ReLU is defined in equation 1.This is followed by pooling layer to reduce the number of dimensions of the maps and choosing the most important features. Then there is the flattened layer converting it into a 1-d vector after which the fully connected layer is applied and then the output is produced via softmax activation function which is defined in equation 2. The architecture of CNN is illustrated in figure 1.

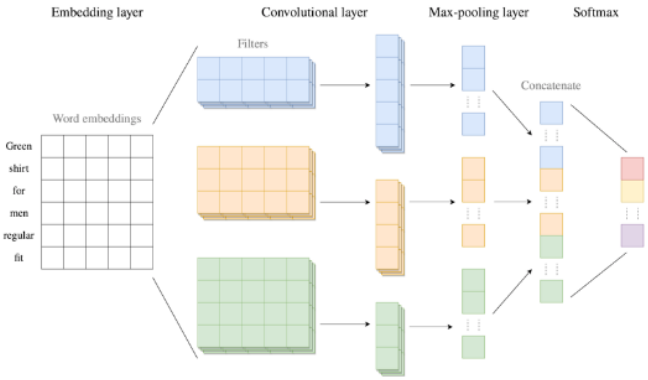

Figure 1: CNN architecture for text classification [29]

$$\text{ReLU} = max(0, x_j) \quad (1)$$

Equation 1 stipulates that ReLU outputs 0 for a negative value of x_j whereas for any positive value x_j, it returns the value x_j.

$$\text{Sigmoid, } s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \quad (2)$$

Equation 2 stipulates that softmax outputs a probability distribution over all the classes n and the sum of the resulting vector is 1. Each element represents individual probability of that specific input belonging to a specific class.

2) LSTM

LSTM is built on the idea of Recurrent Neural Networks (RNN) with the goal of reducing the vanishing gradient problem and also capture long-term dependencies in sequential data. LSTM's main architectural component that gives it an edge over RNN are the memory cells with memory states, input gates that control the input and forget gates that control which of the previous state information should be forgotten and which ones to not. The output gate controls the output based on the training and the states. The movement of values through this gate is controlled by the sigmoid function,

defined in equation 3, and the values that go through are assigned a weight by the tanh activation function, defined in equation 4. The architecture of LSTM is illustrated in figure 2.

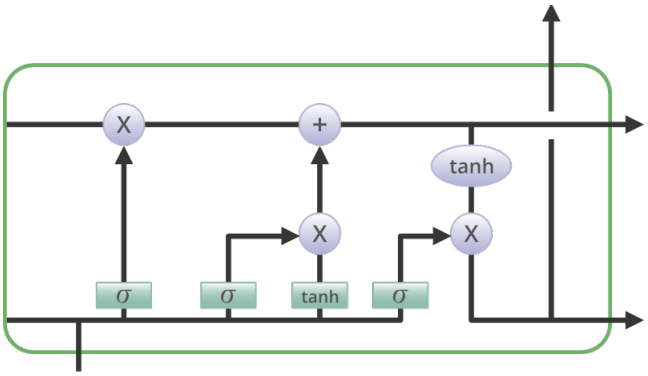

Figure 2: LSTM architecture [30]

$$\text{sigmoid}, \sigma(x) = \frac{1}{1+e^{-x}} \quad (3)$$

Equation 3 explains the sigmoid function where x is the input. Sigmoid function squashes the input values to values between 0 and 1.

$$\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1} \quad (4)$$

Equation 4 explains the tanh function where the input is squashed between -1 and 1 which is essentially a rescaled version of the sigmoid function but instead of squashing the negatives to 0, it is squashed to -1. The 0 being it the centered makes it really useful for updating weights during backpropagation.

3) Bi-directional LSTM

Bi-directional LSTM is very similar to LSTM but instead of unidirectional LSTM approach, it processes the inputs in both forward and backward directions. It has the same exact processes as the LSTM architecture but instead of just one LSTM layer, there are two, the second one processing the input in the backward direction enabling the model to capture dependencies from both directions, both past dependencies and future dependencies. After obtaining the outputs from both the layers, they are generally concatenated. The architecture of Bi-Directional LSTM is illustrated in figure 3 where the x_n are the inputs, y_n are the outputs, A is the LSTM layer that processes data in the forward direction and A' is the LSTM layer processing data in the backward direction.

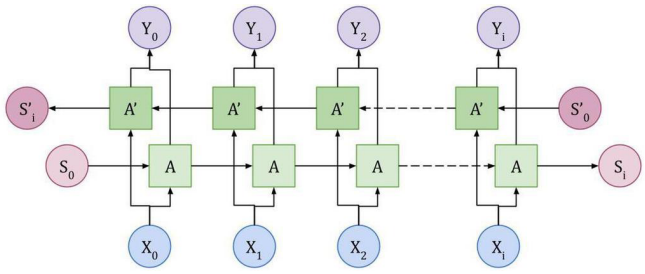

Figure 3: Bidirectional LSTM architecture [31]

4) GRU

GRU is also based on the RNN architecture designed to address the same limitations of RNN as LSTM. However, GRUs have a more streamlined structure than LSTM where there are reset and update gate instead of hidden state. The reset gate controls the extent of information from previous states to be forgotten and the update gate controls the extent of information from the new state to be stored. GRUs are more computationally efficient compared to LSTMs. The architecture of GRUs are illustrated in figure 4.

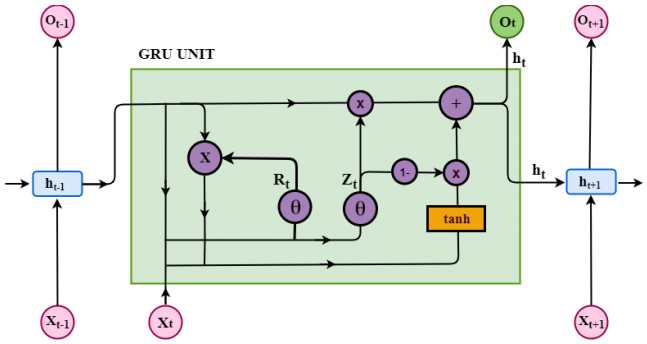

Figure 4: GRU architecture [32]

D. Implementation of the conventional models

Similar model architecture and settings has been used by all of the four models. The details of these models are explained as follows:

The architecture that we used contains an input layer with an input size of 128.

This is followed by an embedding layer that converts the input sequence into dense vectors. The input dimension is set to the size of the vocabulary while the output dimension is set to represent the dense vectors as a vector of size 50.

The next layer is the only layer in the architecture where the different models use different layers.

The convolutional layer is followed by a pooling layer to extract the most prominent and significantly relevant features.

In LSTM, an LSTM layer with 64 units is used to capture long range dependencies.

In Bidirectional LSTM, a Bidirectional LSTM layer is used with 64 units to capture contextual information from both directions.

In GRU, a GRU layer is used with 64 units. It is a class of Recurrent Neural Network that does very well with capturing contexts in long range dependencies.

In CNN, a convolutional layer is then used with a filter size set to 64 which ascertains the features to be extracted. The size of the kernel is set to 3. The activation function used is ReLU.

We added a dropout layer to limit overfitting with a dropout rate of 30%. Finally, the output layer has a sigmoid activation function to classify into the two classes. The sigmoid activation function is defined in equation 3.

The compilation of the model was done using the Adam optimizer with learning rate set to 0.00005 with the loss set to binary cross entropy which is very well suited to binary classification and the evaluation metric set to accuracy.

We trained the model using the 128 sized padded data ran over 3 epochs with batch sizes of 16. We evaluated the performance of it on a validation set.

E. Transformer-Based Deep Learning Models

1) Transformers

After the introduction of the transformer architecture in 2017 by Vaswani et al. [27], the field of natural language processing (NLP) has been revolutionized. Its use was not only limited to NLP but has since expanded into other domains as well.

Transformers have proven to be highly efficient and successful in processing sequential data and capturing long-range dependencies. This has proven to be very useful in the domain of language processing such as translation, sentiment extraction and analysis, text-to-text generation etc.

The standard transformer architecture consists of two key components - an encoder and a decoder. The transformer architecture is illustrated in figure 5.

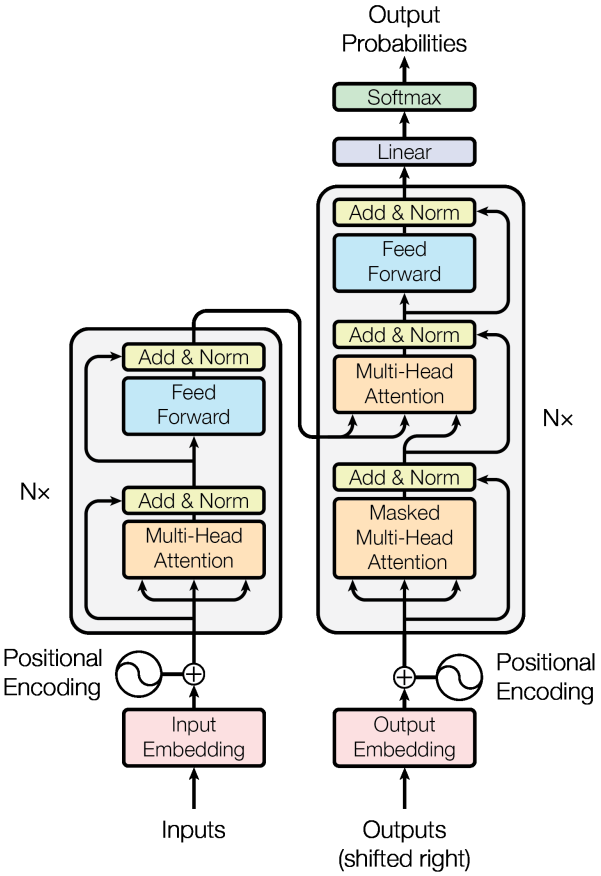

Figure 5: Transformer architecture [27]

A brief understanding of the encoder block of the transformer architecture is imperative to gain an understanding of the working principles of the transformer-based deep learning models such as BERT, DistilBERT and RoBERTa used in this paper, as all these utilize the encoder block of the transformer.

The encoder takes in the input sequence and converts them into embeddings that convert each word into vectors to be represented in an embedding space. Positional encoding is then used by the transformers' encoder module as it does not understand the order that it is fed with, and it helps it understand the sequence of the input. This is followed by the application of multiple iterations or layers of self-attention mechanism. Self-attention mechanism is a process that transformer models use to assign different weights to different words in a sequence and this specific focus is varied in different iterations of it. It then computes the scores based on the attention for each individual word in the input sequence with regards to each words' relationship with other words in the sentence. This enables the capturing of dependencies irrespective of how far a word is from another word in the input sequence. This attention is multi-headed meaning that numerous attention heads work in parallel, each learning different representations of the input sequence enabling the model to understand context from different perspectives and also understand different dependency types. The outputs from all these multiple attention heads are then combined linearly and layer normalization is applied reducing the chances of any issues arising during training. After this step, vectors from each attention head are fed to position-wise feed forward neural networks to add non-linearity and capture the different patterns of the input sequence. This is followed by another layer normalization step with a residual connection being added. This gives an output which contains a highly contextual representation of the sequence that first came into the transformer encoder block.

2) BERT

BERT is one of the most powerful language model that is based on the transformer architecture – specifically the encoder block. BERT processes text from both directions instead of just from left to right or vice versa. This makes it order-independent unlike traditional model where it mostly works in a given order. BERT employs the encoder block and stack multiple encoders in an end-to-end fashion as its key architectural state. There are two commonly available versions of BERT – BERT Base and BERT Large. BERT base is the smaller of the 2 versions and consists of 12 encoder blocks stacked together while the BERT Large consists of 24 of the encoder blocks, having more parameters and a deeper architecture. It is illustrated in Figure 6.

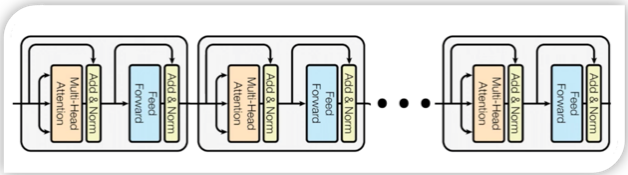

Figure 6: BERT architecture [12]

BERT implements a masked language model as its goal during training where a certain number of words are replaced with masked token which the model tries to predict with the knowledge of other unmasked words. It is, in essence, a type of fill-in-the-blank exercise. This is done by a classification layer that is meticulously placed after a single encoder block's output. This classification is doen by embedding the output obtained as a vocabulary dimension and then predicting the probability for each and evert word that appears in a vocabulary by using the softmax function. The softmax function is illustrated in equation 3. The time for convergence is a bit slow due to BERT only focusing on the correctness of the masked values [28].

3) Distilled BERT

Distilled BERT, also known as DistilBERT is a smaller, lightweight, and compressed version of BERT, the goal of which is to retain as much of the performance of BERT as possible while coming in a more compact form in terms of size and computation. The name DistilBERT comes from the concept that the knowledge from the bigger pre-trained BERT model is distilled down to a smaller, more compact form with its knowledge being transferred.

The architecture of DistilBERT is identical to BERT but the number of blocks in it is exactly half that of BERT Base – 6 instead of 12. This means that DistilBERT has less number of parameters to work with. The idea is for DistilBERT to try and copy or mimic the behavioral patterns of the BERT model thus making it more efficient as it comes in a smaller form while maintaining the knowledge transfer as much as possible.

4) RoBERTa

Robust BERT Approach, or RoBERTa, is an enhancement of the BERT architecture with several different modification in the working principle of the BERT model. It employs dynamic masking during the pre-training stages. In the different batches, the individual tokens that are contained are swapped at random. The next sentence prediction goal of the BERT model is also eliminated making it faster and more efficient. This means that it relies solely on the masked language modelling, or the idea of fill-in-the-blanks as described in BERT. RoBERTa also utilizes far larger batch sizes making it more efficient and improves its learning capabilities.

F. Implementation of the transformer-based models

All three of the transformer-based deep learning models are implemented with similar settings. The detail of the implementation are as follows:

For the first step, the different tokenizers for each of the models are loaded – BERT Tokenizer for BERT, DistilBERT Tokenizer for DistilBERT and RoBERTa Tokenizer for RoBERTa. The tokenizer will be used to convert the raw text data into a form that is compatible with these models.

This is followed by loading the pre-trained models for the respective models from the Hugging Face Transformers library [33]. It is to be noted that the base version of BERT and DistilBERT is used.

With the pre-trained models loaded, regular preprocessing is done to the raw texts as described in the preprocessing section above.

This is followed by passing the list of texts, the respective tokenizers, and the maximum sequence length of 128 into the encoding function of the respective models which returns a numpy array of the tokens, masks, and segments.

Then we constructed a model using the Tensorflow Keras library [34] that takes as input the 3 NumPy [35] arrays from the encoder function. After processing of these by the models, the output is pooled and a dropout of 30% used to regularize the output. This is then passed through a dense function with the sigmoid activation function to generate an output.

The model is compiled in the same way as the conventional models with identical settings - Adam optimizer with learning rate set to 0.00005 with the loss set to binary cross entropy.

We kept the implementation parameters very similar in order to make sure that the grounds for comparison are the same.

V. Experimental Results & Analysis

In this section, we shall state the experimental evaluation metrics that we have used, list down the results that we have obtained for the 7 models and go into analyzing the results that we have obtained.

1) Evaluation Metrics

The evaluation metrics that we used to compare our models' performances are Precision, Recall, F1-Score, Accuracy, and AUC-ROC. The equations of these are defined in equation 5 to equation 10.

$$Precision = \frac{t_p}{t_p+f_p} \tag{5}$$

Here, t_p is the instances of true positives and f_p the instances of false positives. The precision is a classifier's ability to label samples as positive when it was actually positive.

$$Recall = \frac{t_p}{t_p+f_n} \tag{6}$$

Here t_p is the instance of true positives and f_n are the instance of false negatives. The recall is the classifier's ability to correctly predict all the positive samples. Recall is also known as the True Positive Rate (TPR) or Sensitivity.

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{7}$$

F1-Score is the harmonic mean of precision and recall and it shows the model's capability to balance precision and recall.

$$Accuracy = \frac{t_p+t_n}{t_p+t_n+f_p+f_n} \tag{8}$$

Accuracy is the percentage of correctly classified points in the whole dataset.

$$False\ Positive\ Rate\ (FPR) = \frac{f_p}{f_p+t_n} \tag{9}$$

The ROC curve is plotted with the TPR on the y axis and the FPR on the x axis.

$$AUC-ROC = \int_0^1 TPR.dFPR \tag{10}$$

AUC-ROC is calculated by the area under the ROC curve. It is a really useful metric to evaluate a binary classifier's performance. It tells us the discriminative capacity of the model to distinguish between the positive and negative classes.

2) Implementation Details

We implemented the aforementioned models on a Lenovo ThinkStation P720 desktop computer running on Linux Operation System with dual Intel Xeon Processors, 64GB of DDR4 RAM and NVIDIA Quadro Graphics.

We used Visual Studio Code as our text editor to run our models with the programming language python.

3) Experimental Results & Analysis

The experimental results of the different models' performance by the metrics precision, recall, f1-score, accuracy, and AUC-ROC are illustrated in Table 2, 3, 4, 5 and 6.

TABLE 2. PRECISION OF MODELS

Model	Precision
CNN	0.756
LSTM	0.785
Bi-LSTM	0.759
GRU	0.782
BERT	**0.843**
DistilBERT	0.826
RoBERTa	0.832

Precision is a very crucial metric in regard to this dataset as it indicates what proportion of the tweets that were predicted to be a disaster tweet by the model was actually about a real disaster.

It can be seen that the transformer-based models outperform the conventional deep learning models. BERT outperforms all the other models in terms of precision with 0.843 compared to a very close second by RoBERTa. This could be due to the difference in training architectures between them or the fact that RoBERTa works better on longer inputs while the input size was kept the same here. DistilBERT also performed very well in comparison to BERT so despite the smaller and compact form, the difference in performance was quite minimal. The high precision means that false positives can be minimizes thus ensuring that crucial resources are not wasted on incorrectly flagged irrelevant tweet as a disaster tweet.

The conventional models performed similarly with GRU and LSTM with precisions of 0.782 and 0.785 respectfully, having very little difference. Bi-LSTM performed poorer than LSTM at 0.759, and CNN performed the worst out of all the models.

TABLE 3. RECALL OF MODELS

Model	Recall
CNN	0.575
LSTM	0.727
Bi-LSTM	0.686
GRU	0.722
BERT	**0.843**
DistilBERT	0.826
RoBERTa	0.824

Recall, or sensitivity is also a very important metric for our classification task. It indicates how many of the tweets that were indeed about a real disaster were correctly classified by the model as such.

BERT outperformed every other model in terms of this metric as well with a recall of 0.843. RoBERTa and DistilBERT also performed very similarly to each other, and they were not that far behind from BERT too. Good recall values indicate that the model would be more likely to be able to classify tweets about real disasters more effectively thus being able to be used for real-time disaster monitoring as well as scheduling and dispatching emergency response based on an automated system.

However, it is to be noted that the recall for the conventional deep learning models were considerably worse than the transformer-based model with CNN performing very poorly with a recall of only 0.575. Just like precision, LSTM and GRU performed very similarly with Bi-directional LSTM lagging behind somewhat.

TABLE 4. F1-SCORE OF MODELS

Model	F1-score
CNN	0.422
LSTM	0.699
Bi-LSTM	0.629
GRU	0.683
BERT	**0.841**
DistilBERT	0.826
RoBERTa	0.821

F1-score is a metric that we have used to see how balanced the precision and recall values are, thus penalizing values where precision is high, and recall is low or vice versa. It is an indicator of balanced performance indicating that both false positives are reduced as well as relevant tweets are captured as best as possible.

BERT outperformed all the other models yet again as it had the highest precision and recall. DistilBERT and RoBERTa were close behind as they too had quite good recall and precision.

CNN had terrible f1-score as its recall was very poor, so it was reflected here. LSTM and GRU had similar yet slightly lower f1-score of around 0.68 and 0.69 as their precision were higher than recall hence the penalty. Bi-LSTM did not perform that well with an f1-score of 0.629.

TABLE 5. ACCURACY OF MODELS

Model	Accuracy
CNN	57.48%
LSTM	72.70%
Bi-LSTM	68.64%
GRU	72.18%
BERT	**84.25%**
DistilBERT	82.55%
RoBERTa	82.41%

Accuracy is a metric that illustrates the general predictive performance of the model – what proportion of the predictions made by the model are accurate.

BERT outperformed all the other models with the highest accuracy of 84.25%. DistilBERT and RoBERTa were also pretty close with a difference of around 2%. These transformer-based models had a minimum of almost 10% accuracy improvement over the best performing conventional deep learning model – LSTM with 72.70%.

Among the conventional models, CNN performed very poorly with an accuracy of just 57.48%, which could be said to be only merely better than random chance. LSTM and GRU performed really similarly with around 72% accuracy and BiLSTM having 68%.

TABLE 6. AUC-ROC OF MODELS

Model	AUC-ROC score
CNN	0.50
LSTM	0.68
Bi-LSTM	0.62
GRU	0.66
BERT	**0.83**
DistilBERT	0.82
RoBERTa	0.82

AUC-ROC is a powerful metric that indicates how well a model can distinguish between positive and negative classes. As it is threshold independent, it provides a comprehensive summary of the performance. A higher AUC-ROC score indicates that the model has better discriminative capacity and is a good means of comparing a models' discriminative capacities.

It is no wonder that BERT has the highest AUC-ROC score of 0.83 as it outperformed the other models in all the other metrics, so it has the best discriminative capacity out of all the other models.

RoBERTa and DistilBERT performed really well too with a very close second score of 0.82. It is to be noted here too that using a more compressed DistilBERT does not cause a massive fall in the AUC-ROC compared to the BERT so if computational time and memory efficiency is an issue, it could be used instead of BERT with almost negligible drop in performance.

The conventional deep-learning models had considerably lower scores compared to the transformer-based models with CNN having the lowest AUC-ROC score of 0.5 which essentially means that its performance is no better than random chance. LSTM had the highest AUC-ROC compared to the other conventional deep learning models with 0.68 with GRU not that far behind.

4) Discussion

These results illustrate the power that transformer-based deep learning models have over the conventional deep learning models with its transfer learning mechanism when it comes to natural language processing tasks, as demonstrated in this paper by disaster tweet classification.

VI. CONCLUSION

A. Summary

We set out to find out how well transformer-based models perform when compared to conventional deep learning models when it came to NLP tasks, specifically in disaster tweet classification.

The three transformer-based deep learning models BERT, DistilBERT and RoBERTa performed considerably well in all regards and across all the metrics when compared to the conventional deep learning models CNN, LSTM, Bi-LSTM, and GRU.

We demonstrated that BERT performs the best among all the models with DistilBERT and RoBERTa performing almost as good, and CNN performs the worst amongst all the models.

B. Future Research

We intend to carry out further research on the same domain using models such as Albert, T5 and GPT-4 and see how much we can push the performance. We also want to apply BERT Large and see whether the larger models bring about any significant improvement in performance. We also want to work on imbalanced data which is more representative of real-world data to see its effect on the predictive performance.

C. Open Problems

Although BERT performed really well in this domain and DistilBERT was not too far behind, both these models would require large processing capabilities when it comes to real-time prediction. Imbalanced data haunts BERT's performance and scaling BERT and DistilBERT to work with large datasets is something to consider.

REFERENCES

[1] A. Kumar and J.P. Singh, "Location reference identification fromtweets during emergencies: A deep learning approach", *International Journal of Disaster Risk Reduction,*vol. 33 pp. 365-375, 2019

[2] J. P. Singh, Y. K. Dwivedi, N. P. Rana, A. Kumar, and K. K. Kapoor, "Event classification and location prediction from tweets during disasters," *Annals of Operations Research*, May 2017. [Online]. Available: https://doi.org/10.1007/s10479-017-2522-3 [Accessed: 10-Dec-2023]

[3] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: A survey," *ACM Computing Surveys(CSUR)*, vol. 47, no. 4, p. 67, 2015.

[4] A. Kumar, J. P. Singh, and N.P. Rana, "Authenticity of geo-location and place name in tweets", *Proceedings of the 23rd americas conference on information systems*, 2017.

[5] L. Palen, & A. L. Hughes, "Social media in disaster communication", *Handbook of disaster research*, pp. 497-518, 2018.

[6] A. Elci, "Turkey-Syria earthquakes: How Twitter has helped find survivors trapped beneath the rubble", *Euronews*, 10-Feb-2023. [Online]. Available:https://www.euronews.com/next/2023/02/10/how-twitter-helped-find-survivors-trapped-beneath-rubble-after-turkeys-earthquakes#:~:text=Next%20Tech%20News-,Turkey%2DSyria%20earthquakes%3A%20How%20Twitter%20has%20helped%20find,survivors%20trapped%20beneath%20the%20rubble&text=Earthquake%20victims%20took%20to%20Twitter,help%20search%20and%20rescue%20teams. [Accessed: 10-Dec-2023]

[7] Washington Post, "'It was insane': Drone photo on Twitter helps rescue flood victim and dog halfway across country", *National Post*, 11-Oct-2016. Available: https://nationalpost.com/news/world/it-was-insane-

drone-photo-on-twitter-helps-rescue-flood-victim-and-dog-halfway-across-country. [Accessed: 10-Dec-2023]

[8] L. Zewen, F. Liu, W. Yang, S. Peng, and J. Zhou. "A survey of convolutional neural networks: analysis, applications, and prospects." *IEEE transactions on neural networks and learning systems*, 2021.

[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[10] H. Zhiheng, W. Xu, and K. Yu. "Bidirectional LSTM-CRF models for sequence tagging", *arXiv preprint arXiv:1508.01991,* 2015.

[11] C. Junyoung, C. Gulcehre, K. H. Cho, and Y. Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555,* 2014.

[12] J. Devlin, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *arXiv e-prints*, 2018.

[13] S. Victor, L. Debut, J. Chaumond, and T. Wolf. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv preprint arXiv:1910.01108*, 2019.

[14] L. Yinhan et al. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692,* 2019.

[15] A. Howard, Devrishi, P. Culliton, Y. Guo. "Natural Language Processing with Disaster Tweets" *Kaggle*, 2019. Available: https://kaggle.com/competitions/nlp-getting-started

[16] S. Vieweg, C. Castillo, and M. Imran, "Integrating Social Media Communications into the Rapid Assessment of Sudden Onset Disasters", *Social Informatics. Ed. by L. M. Aiello and D. McFarland. Cham: Springer International Publishing*, pp. 444–461, 2014.

[17] H. Purohit, and A. Sheth "Twitris v3: From citizen sensing to analysis, coordination and action", *Proceedings of ISCRAM*, pp. 918–922, 2013.

[18] A. Kongthon, C. Haruechaiyasak, J. Pailai, and S. Kongyoung "The role of Twitter during a natural disaster: Case study of 2011 Thai Flood", *Technology Management for Emerging Technologies (PICMET), 2012 Proceedings of PICMET'12:* IEEE, pp. 2227–2232, 2012.

[19] Caragea et al., "Classifying text messages for the Haiti earthquake". *Proceedings of the 8th International ISCRAM Conference–Lisbon* (Vol. 1), 2018

[20] D. Nguyen et al., "Robust Classification of Crisis-Related Data on Social Networks Using Convolutional Neural Networks.", *ICWSM*, pp. 632–635, 2017.

[21] A. Aipe, N. S. Mukuntha, A. Ekbal, S. Kurohashi, "Deep learning approach towards multi-label classification of crisis related tweets", *Proceedings of the 15th ISCRAM Conference*, 2018.

[22] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: Can we trust what we rt?", *Proceedings of the first workshop on social media analytics*. ACM, pp. 71–79, 2010

[23] C. Caragea, A. C. Squicciarini, S. Stehle, K. Neppalli, and A. H. Tapia, "Mapping moods: Geo-mapped sentiment analysis during hurricane sandy.", *ISCRAM*, 2014.

[24] S. Madichetty and M. Sridevi, "Detecting informative tweets during disaster using deep neural networks", *11th International Conference on Communication Systems & Networks (COMSNETS)* pp. 709-713, 2019.

[25] A. K. Ningsih and A. I. Hadiana. "Disaster tweets classification in disaster response using bidirectional encoder representations from transformer (BERT)." *IOP Conference Series: Materials Science and Engineering*, vol. 1115, no. 1, p. 012032. IOP Publishing, 2021.

[26] S. Madichetty, S. Madisetty. "A RoBERTa based model for identifying the multi-modal informative tweets during disaster.", *Multimedia Tools and Applications* pp 1-19, 2023.

[27] A. Vaswani et al., "Attention Is All You Need", *arXiv e-prints*, 2017.

[28] Rani Horev, "BERT Explained: State of the art language model for NLP", *Towards Data Science*, 10-Nov-2018. [Online]. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270. [Accessed: 10-Dec-2023]

[29] M Sluijmers, "Convolutional neural network text classification with risk assessment", *Squandra Machine Learning Company,* 2018. [Online]. Available: https://machine-learning-company.nl/en/technical/convolutional-neural-network-text-classification-with-risk-assessment-eng/ [Accessed: 10-Dec-2023]

[30] A. Chugh, "Deep Learning | Introduction to Long Short Term Memory", *GeeksForGeeks*, 31-May-2023. [Online] Available: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/. [Accessed: 10-Dec-2023]

[31] A. Taparia, "Bidirectional LSTM in NLP", *Geeksforgeeks,* 2019. [Online]. Available: https://www.geeksforgeeks.org/bidirectional-lstm-in-nlp/ [Accessed: 10-Dec-2023]

[32] Bibi et al., "Dynamic DL-Driven Architecture to Combat Sophisticated Android Malware." *IEEE Access*. PP. 1-1. 10.1109/ACCESS.2020.3009819, 2020.

[33] T. Wolf et al. "Huggingface's transformers: State-of-the-art natural language processing." *arXiv preprint* arXiv:1910.03771, 2019.

[34] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[35] C. R. Harris et al., "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586- 020- 2649- 2