JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

REAL TWEET OPINION MINING BASED ON DistilBERT

[1]Thanasekhar B, [2]Arthi Ramachandran, [3]Christy KT

[1] Associate Professor, Department of Computer Technology, Anna University, MIT Campus, Chennai, India
[2]PG Scholar, Department of Computer Technology, Anna University, MIT Campus, Chennai, India
[3] Teaching Fellow, Department of Computer Technology, Anna University, MIT Campus, Chennai, India

Abstract : Sentiment analysis or opinion mining is the computational study of people's opinions, sentiments, attitudes, and emotions expressed by user-generated content. The essence of sentiment analysis is the text classification task, and different words have different contextual information for classification. In the current sentiment analysis studies, word representation is mostly used. However, the semantic information of a word alone is considered in word representations but ignores the sentiment information of the word. In the proposed sentiment analysis method, sentence representation is used. Two approaches are proposed for sentiment analysis of the full context information which describes the real disaster is input into DistilBERT sentiment classifier and BiLSTM sentiment analysis method to capture the sentiment scores of the context effectively. In the case of large text, using only words would be very tedious and we would be limited by the information extracted from the word embeddings. Hence a word embedding technique in RNN machine learning model BiLSTM is compared with the sentence embedding technique DistilBERT. The evaluation metrics used in the comparison of performance is F1 score. The model acquires accuracy of 84% for BERT along with BiLSTM and 90% for BERT along with DistilBERT. Thus BERT with DistilBERT gives higher performance than BERT with BiLSTM word embedding method.

Ketwords : Opinion Mining, Sentiment Analysis, DistilBert, Word Embedding, Tweet Classifier

1. INTRODUCTION

Sentiment analysis has been widely researched in the domain of marketing with the aim of generating summarized opinions of users about different aspects of products. However, there has been little work focusing on identifying the polarity of sentiments expressed by users during disaster events to contribute the situation awareness and better understanding. Identifying such sentiments from micro blogs can help emergency responders understand the dynamics of the network and crisis management. BERT (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. Utilizing this bidirectional capability, BERT is related to Natural Language Processing (NLP) tasks: Masked Language Modeling and Next Sentence Prediction (NSP). The significant aim of Masked Language Model (MLM) training is to hide a word in a sentence and then have the program predict what word has been hidden (masked) based on the hidden word's context. The aim of Next Sentence Prediction training is to have the program predict whether two given sentences have a logical, sequential connection or whether their relationship is basically irregular. DistilBERT is a deep learning model, lighter, cheaper, and quicker than BERT, which was pre-trained on the same corpus in a self-supervised fashion. DistilBERT has about half the total number of parameters of the BERT base and retains 95% of BERT's performances. This means it was pre-trained on the raw texts only, with no humans labelling them in any way with an automatic process to generate inputs and labels from those texts using the BERT base model. The challenge prevails with the large models are operating in on-the-edge and/or under constrained computational training. These challenges leverage to DistilBERT model being 60% faster than BERT. A Bidirectional LSTM, is a type of RNN (Recurrent Neural Network) sequence processing model that consists of two LSTMs: one takes the input in a forward direction, and the other in a reverse direction. BiLSTMs effectively increase the amount of information available to the neural network, progressively the context is (i.e. word representation to sentence representation) available to the machine learning algorithm (predicting the words immediately follow and precede a word in a sentence).

2. RELATED WORK

Xu et.al. (2019) proposed a new representation method of word vector based on the improved term weight computation by integrating the sentiment information contribution degree into the TF-IDF algorithm. This method was proposed to overcome the word deficiency representation and is compared with the previous works of sentiment analysis using RNN, CNN, LSTM, and NB. A word embedding method using latent contextual semantic relationships and co-occurrence statistical characteristics between words in tweets is proposed by Jianqiang et. al. (2018). This embeddings are combined with n-grams features and word sentiment polarity score features to form a sentiment feature set of tweets. The feature set is integrated into a deep convolution neural network for training and predicting sentiment classification labels. Pan et. al. (2020) proposed an approach for leveraging contextual features from unlabelled movie and restaurant reviews with a neural-network-based learning model called Ladder network. The experimental results by using two benchmark datasets, IMDb and YelpNYC, show that the model outperforms the baseline models including

LSTM and SVM. Kaliyar et. al. (2021) experimented a bidirectional training approach which is a priority for modelling the relevant information of fake news that is capable of improving the classification performance with the ability to capture semantic and long-distance dependencies in sentences. This framework consists of a BERT-based deep learning approach, called FakeBERT by combining different parallel blocks of the single-layer deep Convolutional Neural Network (CNN) having different kernel sizes and filters with the BERT. Such a combination is useful to handle ambiguity, which is the greatest challenge to natural language understanding. Yang et. al. (2019) created a new emotion-semantic-enhanced convolutional neural network (ECNN) to improve the performance of sentimental analysis by projecting emoticons and words into an emoticon space. This helps to identify subjectivity, polarity, and emotion in microblog environments. Basiri et. al. (2019) introduced five policies namely, most occurring first (MOF), most general first (MGF), most specific first (MSF), first occurring first (FOF), and last occurring first (LOF). Finally, using the part-of-speech (POS) tags, potential terms in the sentences are specified and a comprehensive sentiment lexicon is employed to compute the polarity of the sentences. In this work, the MGF policy achieves the best performance in finding the main target of reviews, while for finding the ultimate polarity of reviews, the MOF outperforms other policies. Imran et. al. (2020) proposed to analyse the reaction of citizens from different cultures to the novel Coronavirus and people's sentiment about subsequent actions taken by different countries by using a deep learning long short-term memory (LSTM) model for estimating the sentiment polarity and emotions from extracted tweets.

Jianqiang et. al. (2017) proposed the text pre-processing method on sentiment classification performance in two types of classification tasks and summed up the classification performances of six pre-processing methods using two feature models and four classifiers on five Twitter datasets. The experiments show that the accuracy and F1-measure of Twitter sentiment classification classifier are improved when using the pre-processing methods of expanding acronyms and replacing negation but barely changes when removing URLs, removing numbers, or stop words. Gao.et. al. (2019), implemented three target-dependent variations of the BERTbase model, with positioned output at the target terms and an optional sentence with the target built in. TD-BERT model achieves new state-of-the-art performance, in comparison to traditional feature engineering methods, embedding-based models and earlier applications of BERT. Alsaeedi et. al. (2019) proposed a sentiment analysis technique of Twitter data. The basic idea of sentiment investigation is to detect the polarity of text documents or short sentences and classify them on this premise. The main objective is to study the existing sentiment analysis methods of Twitter data and gain a theoretical comparison of those approaches. Various sentiment-analysis approaches used for Twitter are described including supervised, unsupervised, lexicon, and hybrid approached. Supervised ML method employing an algorithm of NB, MaxEnt, and SVM with the feature selection method of unigrams, bigrams, POS using Tweets collected using Twitter API achieves an accuracy of 82.7%. Unsupervised methods using Exploring slag sentiment words in sentiment analysis (ESSA) and a feature selection method of Unigrams on STS and OMD datasets achieves classification accuracies of 0.726 for the STS dataset and 0.692 for the OMD dataset. The outcome demonstrates the machine learning technique SVM classifier may be considered as standard learning strategies.

3. PROBLEM DEFINITION

People express their opinions through social network applications, like Twitter, Instagram and Facebook. With the rapid increase of social media sites, people are using these platforms to voice their opinions about daily issues. In recent years, we have been faced with a series of natural disasters causing a tremendous amount of financial, environmental and human losses. The unpredictable nature of natural disasters behaviour makes it hard to have a comprehensive situational awareness to support disaster management. Using opinion surveys is a traditional approach to analyse public concerns during natural disasters is expensive and time-consuming. So, to overcome such problem the advent of social media has provided an alternative means of analysing public concerns. A new form of communication, such as micro blogging and text messaging has emerged and become pervasive. While there is a freedom to share opinions and convey information through tweets, often these short messages are used to analyse the sentiments of people and about what is going on in the world around them. The objective of the work is as follows

- To identify the tweets that describes relevantly about the disaster.
- Evaluate the sentiment scores and results of the tweets regarding disaster using word embedding and sentence embedding methods.
- To reduce the negative impacts and the number of casualties caused by natural disaster.

4. REAL TWEET OPINION MINING

In emergencies and disaster, people are likely to utilize social media to communicate their hindrances. Opinion investigation is regularly performed on literary information to help organizations screen brand and product; however, its utilization is high for marketing. The main focus of this work is to effectively analyse the sentiments of the tweets that genuinely describes about the disaster. Sentiment analysis (or opinion mining) is a technique used to determine whether data is positive, negative, or neutral. In particular, addresses the identification of the real tweets and exact sentiment conveyed by the user rather than the overall sentiment polarity of his text message or post. The challenges faced during sentiment analyses are subjectivity and tone, context and polarity, irony and sarcasm, comparison, and emoji. The objective text does not speak explicitly about the sentiment whereas the subjective text does. Capturing a bit of context, the way the tweet was posted will help to analyse sentiment that is not straightforward. Tweets might be expressing negative sentiments using positive words, which can be difficult to detect without having a thorough understanding of the context of the situation in which a feeling was expressed. Special attention has to be paid at character-level, word-level and sentence-level while performing sentiment analysis on tweets.

Tweets from the Kaggle dataset are collected which has more than 10,000 tuples to train and test. The tweets are preprocessed using a tweet preprocessor. The clean and fine-tuned tweets are given as input to the fake tweet classifier model. Disaster related tweets are collected as input to the next modules. Again, based on the sentiment analysis classifier the tweets are classified as positive or negative scores. Figure 1 illustrates the architecture of real tweet detection and sentiment analysis.

Here, the tweets are posts made on the social media application Twitter. Tweets are exceptionally casual with inventive spelling and accentuation, incorrect spellings, slang, new words, URLs, and class explicit phrasing and shortened forms, for example, RT for "re-tweet" and # hashtags, which are a sort of labelling for Twitter messages. Fake tweet analysis and opinion mining to be carried out purely based on the text the data has to be preprocessed to remove links, smileys, etc. All the unwanted data in the tweets are removed completely. Figure 2 illustrate the preprocessing steps of the data. The algorithm1 fine tuning the data, preprocess the tweets by removing link, url and so on.

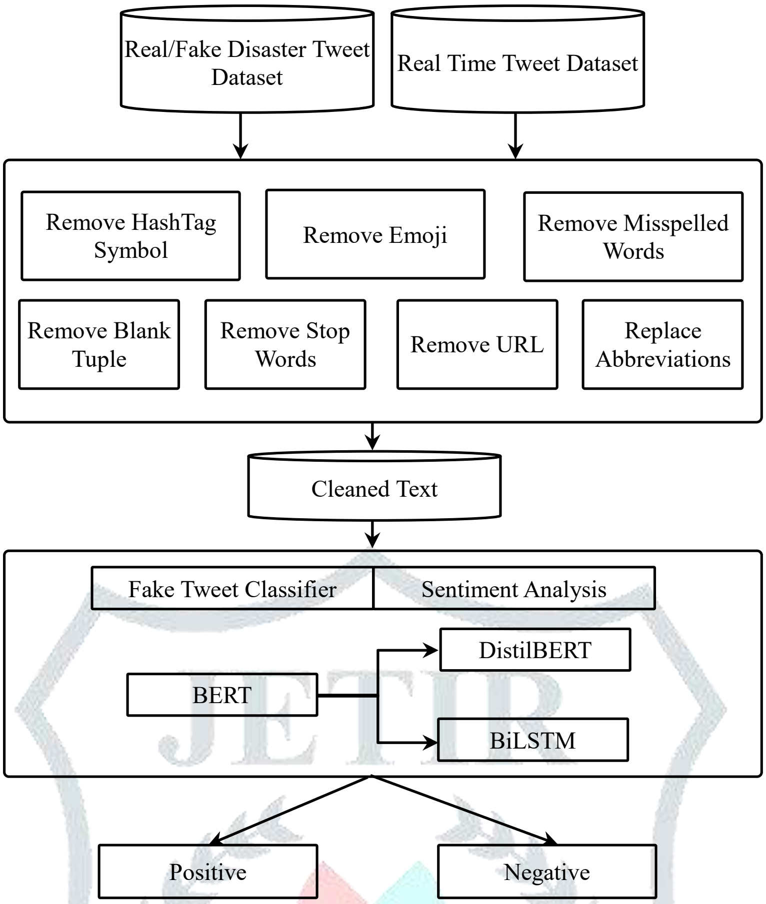

Figure 1 System architecture of proposed disaster related real tweet sentiment analysis

To perform classification of the collected fine-tuned tweets a machine learning technique BERT is used. Machine Learning is an area of Artificial Intelligence that allows machines to learn without being explicitly programmed. These Machine Learning algorithms are trained and tested with a data sample of 10800 tweets, of which a ratio of 80:20 has been used. There are 6203 of not about disaster and 4673 tweets of real disaster. Fine-tuning a BERT for fake tweet classifier with the help of ktrain, a lightweight wrapper for the deep learning library TensorFlow keras, has been programmed in python for text classification. The Python package Scikit-learn was used for splitting the dataset. Scikit-learn contains efficient and straightforward tools for data mining and data analysis and is open source. The input for the BERT model is a sequence of words and the outputs are the encoded-word representations (vectors). For single-sentence classification, the vector representation of each word as the input is used to classify. The tweet which is the sequence of words is tokenized to a sentence and given as the input to the BERT model. Since the tweets are of different length sequences, padding of tweets with the same sequence length is performed during tokenizing. To attain the contextual relation between the words BERT uses techniques like the Masked Language model(ML) and Next Sentence Prediction.

In the Masked Language model, 15% of the words in each sequence are replaced with mask tokens. The model then attempts to predict the real value of masked words based on the contextual meaning of other words in the sequence. In the Next Sentence Prediction BERT technique, the model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the real document. Figure 3 gives a vivid picture of how the BERT model predicts the tweets. Text.texts_from_df() method split the dataset into train and test set, the maximum length of words is set in between the range of 400 to 500 and the column of the data frame to be considered for training and testing. Functions such as learner.find() and learner.plot() to find the best learner rate for the given batch size.

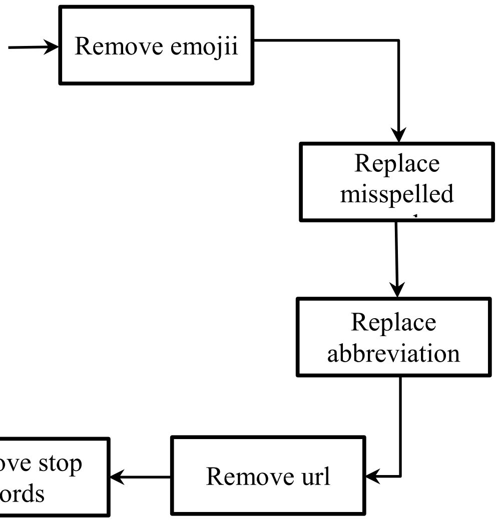

Figure 2 Preprocessing the dataset

ALGORITHM 1: FINE TUNING THE DATA

Data: dataset:= corpus of real time streaming tweets
Result: preprocessed tweets
Real time testing set $\{T_i \in t_1, t_2, \ldots. t_n\}$
begin
    for each attribute T_i
    // check for missing values
    if T_i == NULL then remove tuple of T_i endif
    // check for hashtag or emoji
    if T_i == # hashtag || emoji || url then remove #hashtag || emoji || url endif
    endfor
end

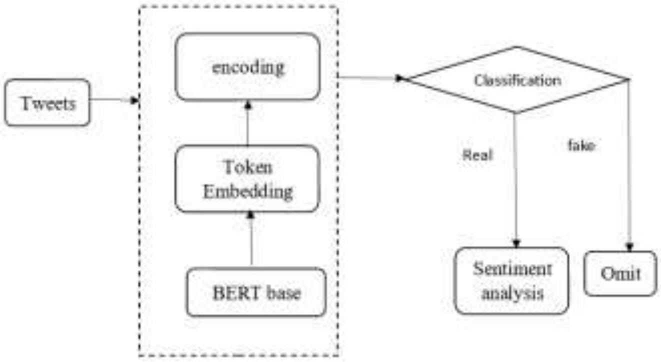

Figure 3 Fake and real tweet prediction using BERT

5. SENTIMENT ANALYSIS CLASSIFIER

Sentiment analysis of a word or sentence or tweet by a computer is really difficult, and human intervention is required to identify the polarity between the words and analyze the sentiment. Sentiment analysis is called opinion mining because it is meant to analyze the feelings, opinions, and emotions of the people. It is been performed by analyzing the sentiment of each word and classifying them as positive, negative and neutral. There are also many different ways the sentiment analysis can be performed, based on the lexicon approach, machine learning, deep learning, or a combination of both machine learning and lexicon-based approaches. Sentiment analysis based on the machine learning approach proves to be more accurate in recent trends. Tweets in the text file are first to split into tokens. Replace each token with the id from the embedding table of the DistilBERT model. The input vector through DistilBERT works just like BERT. The output would be a vector for each input token.

The sentence embedding value of the DistilBERT is given as input to the Logistic Regression of the scikit learning model for categorical classification of positive or negative. Figure 4 illustrates the workflow process of DistilBERT. The dataset used for training and testing is sentiment140 consist of 1.6 million tuples with positive and negative scores. The dataset is split in the ratio of 80:20. Algorithm 2 sentiment analysis classifier, distinguish the tweets given as either real or irrelevant about the disaster**.**

ALGORITHM 2: SENTIMENT ANALYSIS CLASSIFIER

Data: dataset:= Processed tweets T1,T2,T3
Return: classification:= Positive or Negative
Notation: TE: Token Embedding, IE: Input Embedding, MLM- MAskedLM, SE: Sentence Embedding Layer, Hidden_nodes, Attention_heads, Parameters set the default parameters

```
begin
        IE ← TE
        IE ← MASK 80% of input
        IE ← random token10%
        IE ← unchanged 10%
        Learning_rate, Max_length, Batch_size, Epoch ← Set the hyper parameter values return real or fake
        IE ← SE
        Learning_rate, Max_length, Batch_size, Epoch ← Set the hyper parameter values return positive or negative
        Real || fake ← T3
        if T3 == Real then return positive || negative ← T3 endif
end
```

The tweets from Twitter are collected based on the user name specified. Using twitter API, it is achievable to collect the twitter data. Library named RAuth is used in performing authentication by giving in the keys. Consumer Key, Consumer Secret, Access Token and Access Token Secret for twitter application and perform Handshake protocol. After which, certificate is downloaded and PIN is generated for the application to access tweets. Tweets are saved in csv file. Almost 100 tweets are saved from recent ones. preprocess the collected tweets. Using TextBlob a python library for processing textual data sentiment scores of the tweets are generated. TextBlob provides a simple API for tasks such as parts-of speech tagging, noun phrase extraction, sentiment analysis, classification, translation and more.

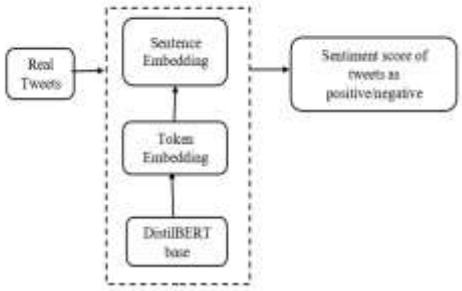

Figure 4 Sentiment analysis classifier

ALGORITHM 3: REAL TIME TWEET ANALYSIS

INPUT: #hashtag keyword to search and collect the tweet
OUTPUT: Sentiment scores for the tweets
//Collection of tweets using #hashtag
//Create a Twitter Developer account to obtain the credentials

```
begin
        consumer_key      ← consumer key to twitter
        consumer_secret   ← consumer secret to twitter
        access_token      ← access token to twitter
        access_secret     ← access secret to twitter
        auth              ←     authenticate using credentials
        df                ← creation of dataframe to collect tweets
        keyword           ←  hashtag to begin keyword search
        for count != 0
            tweets        ← collected tweets using API
        end for
        filename          ← csv file to save the collected tweets
        for tweets in df
            score         ←     sentiment value of the tweets
            quantify      ←  positive || negative|| neutral with score value
        end for
end
```

Polarity of the tweets are generated with which the tweets can be quantify as positive, negative or neutral tweets. Polarity value lies between -1.0 to +1.0. These tweets can be used for evaluation of the previous model sentiment analysis classifier. Figure 3.5 illustrates the analysis process of tweets using TextBlob. Algorithm 3 real tweet analysis describes generating sentiment scores for real tweet collected using hashtag disaster.

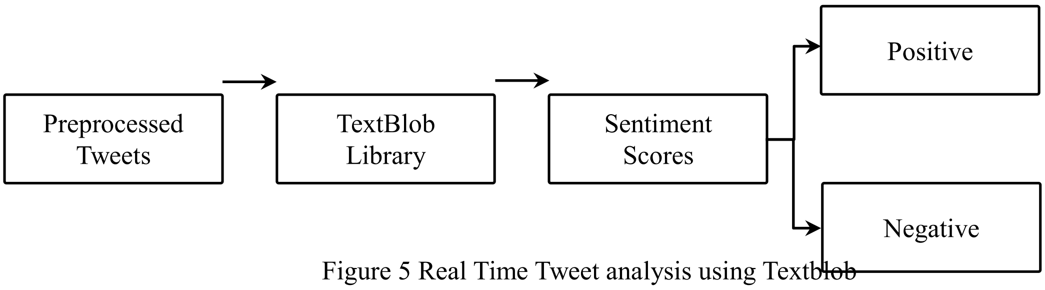

Figure 5 Real Time Tweet analysis using Textblob

6. BILSTM SENTIMENT CLASSIFIER

Long Short-Term Memory (LSTM) networks is a kind of RNN model that deals with the vanishing gradient problem. It learns to keep the relevant content of the sentence and forget the non-relevant ones based on training. This model preserves gradients over time using dynamic gates that are called memory cells. At each input state, a gate can erase, write and read information from the memory cell. Gate values are computed based on linear combinations of the current input and the previous state.

BiLSTM allows us to look ahead by employing a forward LSTM, which processes the sequence in chronological order, and a backward LSTM, which handles the sequence in reverse order. The output is then the concatenation of the corresponding states of the forward and backward LSTM. The idea behind bi-directional network is to capture information of surrounding inputs. It usually learns faster than one-directional approach although it depends on the task.

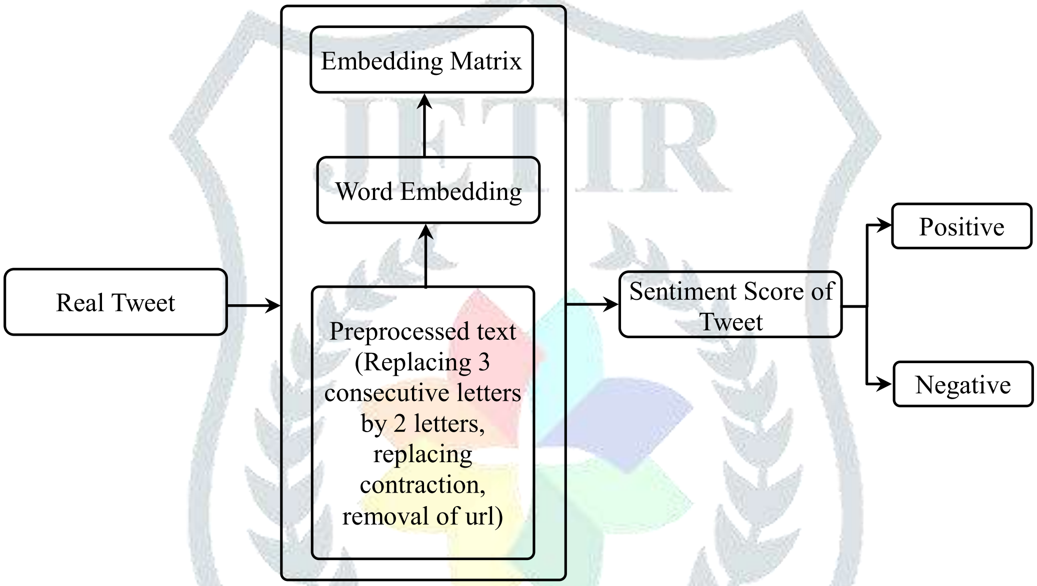

Figure 6 Prediction of tweet score by BiLSTM

A Sequence model deploys Recurrent Neural Network type BiLSTM architecture to perform Sentiment Analysis to categorize a positive or negative tweet. The dataset being used is the sentiment140 dataset. It contains 1,600,000 tweets extracted using the Twitter API. The tweets have been annotated (0 = Negative, 1 = Positive) and they can be used to detect sentiment. The dataset distribution of positive and negative tweets is verified to get a balanced class distribution. Text pre-processing transforms text into a more digestible form so that deep learning algorithms can perform better.

Pre-processing takes place as removal of url, changing to lower cases, replacing three consecutive letters by two letters, replacing contractions with the help of contraction list. The dataset is split into 95% as training set and 5% as test set. Create word embedding using Word2Vec model, which represent document vocabulary in a vector representation. Tokenization is a way of separating a piece of text into smaller units called tokens. Here the token is word from entire text. Padding is the process by which padding tokens are added at the start or end of a sentence to increase its length up to the required size.

Embedding Matrix is a matrix of all words and their corresponding embeddings. Embedding matrix is used in an Embedding layer in the model to embedded a token into it's vector representation, that contains information regarding that token or word. The embedding vocabulary from the tokenizer and the corresponding vectors from the Embedding Model is represented by the Word2Vec model.

Model architecture consist of a embedding layer, which is responsible for converting the tokens into vector representation, two BiLSTM layer to get the contextual meaning of the input, and a Dense layer with sigmoid activation function to get the predicted output. Evaluating the model with the metrics precision, recall and f1-score. Figure 3.6 illustrate the real time tweet prediction sentiment score by BiLSTM model. Algorithm 4 BiLSTM sentiment classifier describes generating sentiment scores for the disaster related tweets.

ALGORITHM 4: BILSTM SENTIMENT CLASSIFIER

INPUT: Dataset consisting of tweets both positive and negative(sentiment140)
OUTPUT: Sentiment scores for the tweets
//Collection of tweets from sentiment 140 dataset

```
//Create a sequential model to classify tweets as positive or negative
Begin
    df  ← tweet dataset
    df2  ← preprocessed tweets of df
    train_data, test_data  ← split the data in 95:5 proportion for training and validating
    embedding_matrix  ← embedded vocabulary in vector representation of train_data
    Sequential_model  ← BiLSTM model architecture to train
    //Prediction using hybrid approach
    Real_tweet  ← tweet classified as real by fake tweet classifier for tweets in Real_tweet
    score_value  ← sentiment value of the Real_tweet predicted using BILSTM  sentiment classifier
    quantify  ← positive || negative with score_value
end
```

7. EXPERIMENTAL SETUP AND RESULT ANALYSIS

7.1. DATASET DESCRIPTION

Collecting real-time tweets using the popular disaster hashtag, a developer account has to be created in the Twitter micro blog. Keys such as access token, access secret key, consumer API key, and consumer API secret key are created for each account to collect the tweets in real-time. Using Twitter API, the details of the user along with the tweets count of more than 100 can be collected. The collected data frame can be saved as a CSV file. For training, the Fake Tweet Classifier(FTC), disaster tweets with two classes real and fake is used which consist of more than 10000 tuples. The Sentiment Analysis Classifier (SAC) is trained with a benchmark dataset of sentiment140 which consists of two classes positive and negative. The SAC and FTC model experimental dataset is shown in Table 1 and Table 2. Figure 1 shows the word cloud representation of positive tweets and figure 2 shows word cloud representation of negative tweets in sentiment140 dataset. Figure 3 shows word cloud representation of real tweet and figure 4 shows word cloud representation of fake tweet in disaster tweet dataset.

Table 1 Distribution of binary class tweets in SAC dataset

DATASET	TOTAL TWEET COUNT	SPLIT	POSITIVE COUNT	NEGATIVE COUNT	FEATURE COUNT
Sentiment140		Train	700000	700000	6
		Test	100000	100000	
Total	1600000		800000	800000	

Table 2 Description of FTC experimental dataset

DATASET	TOTAL TWEET	SPLIT	REAL COUNT	FAKE COUNT	FEATURE
Disaster tweet		Train	3673	5000	4
		Test	1000	1203	
Total	10876		4673	6203	

The Colab notebooks that allows us to combine executable code, rich text and dataset files in a single document is used for implementation of real tweet opinion mining based on DistilBERT. Further the deployment of the classifier is done locally in local machine with the RAM of 8 gb and NVIDIA mx250. Ktrain library from keras is utilized for model implementation. Specifically, the uncased version of BERTBASE with 768 of embedding dimension is chosen for FTC. DistilBERT version with 768 hidden layer and 66M parameters is utilized for SAC model to classify the real tweets as positive or negative sentiment scores.

Figure 1 Word cloud representation of positive tweets in sentiment140 datasets

Figure 2 Word cloud representation of negative tweets in sentiment140 datasets

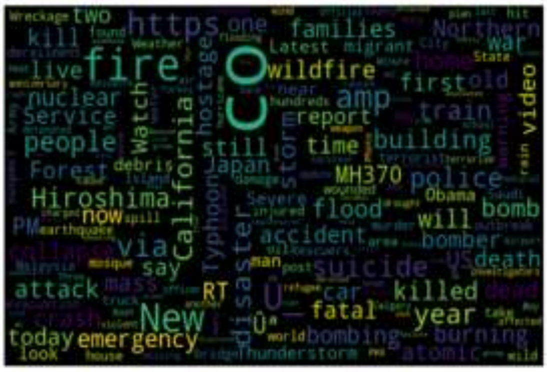

Figure 3 Word cloud representation of real tweets in disaster datasets

Figure 4 Word cloud representation of fake tweets in disaster datasets

7.2. PERFORMANCE METRICS

Accuracy and F1-measure are used as evaluation metrics as they are reliable measures of performance. The following parameters are used in computing the performance metrics:

1. TP: the count of positive tweets/tweets that are labelled as positive.
2. FP: the count of negative tweets/tweets that are labelled as positive.
3. TN: the count of negative tweets/tweets that are labelled as a negative.
4. FN: the count of positive tweets/tweets that are labelled as negative.
5. Accuracy: the number of correctly classified tweets /tweets divided by the overall number of tweets/tweets.
6. Precision: the percentage of correctly predicted positive tweets/tweets to the total positive tweets/tweets.
7. Recall: the percentage of correctly predicted positive tweets/tweets to all tweets/tweets in the actual class.
8. F1-measure: the weighted average of precision and recall.

7.3. RESULTS

7.3.1. Fake Tweet Classifier Results

Machine learning technique BERT is used for classification of real and fake tweets. The model is fine tuned with disaster dataset. Disaster dataset consist of more than 10,000 tuples with 6203 of not about disaster and 4673 tweets of real disaster. The training and testing set is in the ratio of 8:2. Prediction of real or fake tweets of disaster using fake tweet classifier using real-time tweets is illustrated in the below figure 5. The data array consists of tweets and the second element describes about the injury happened to a person hence its considered as real tweet.

Figure 6 shows the training curve slopes down gently and the validation curve tries to become stable after the first epoch. The table 3 shows the performance metrics F1 score and accuracy of the fake tweet classifier. Since the F1 score is more than 80% means that low false positive and low false negative so that the FTC almost correctly identify the real and fake tweets that describes about the disaster.

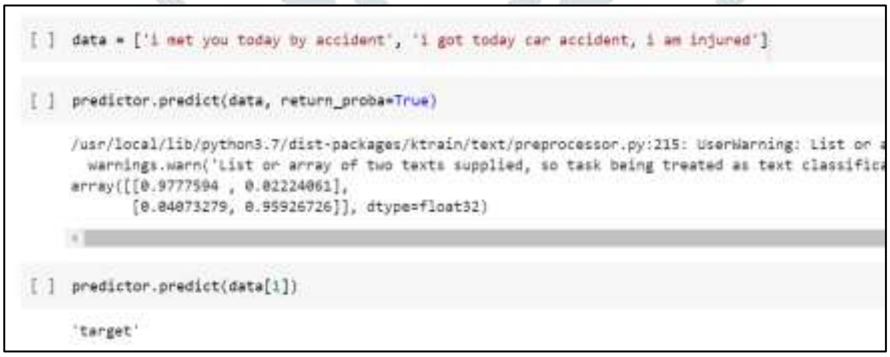

Figure 5 Prediction of fake or real tweets

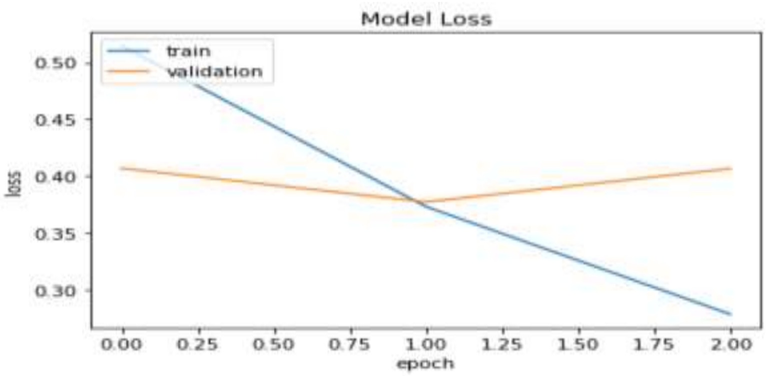

Figure 6 Fake tweet classifier model loss graph

Table 3 Performance metrics of FTC Recall

Aspect	Precision	Recall
Fake	0.86	0.86
Real	0.82	0.82
Accuracy	0.84	

Table 4 Description of DistilBERT hyper parameter

Hyper Parameter	Values Count
Train_Batch_Size	64
Predict_Batch_Size	64
Learning_Rate	2e-6
Num_Train_Epochs	3
Max_Seq_Length	32
Warmup_Proportion	0.1

The recall ratio of correctly predicted fake observations to the all observations in actual class as fake is 0.86 and for real observation class as real is 0.82. Thus the accuracy of the FTC model is 84%.

7.3.2. Sentiment Analysis Classifier

Analyzing the sentiment of a word or sentence or tweet by a computer is really difficult, and the human intervention is required to identify the polarity between the words and analyze the sentiment. DistilBERT model is deployed for sentence embedding along with logistic regression for classification. The hyper parameter of DistilBERT model is described in the following table 4.

```
[ ] data = ["damn you north korea", "negative thoughts leads to negative events"]

[ ] predictor.predict(data)

    /usr/local/lib/python3.7/dist-packages/ktrain/text/preprocessor.py:215: UserWarnin
      warnings.warn('List or array of two texts supplied, so task being treated as tex
    ['0', '0']

[ ] predictor.predict(data, return_proba=True)

    /usr/local/lib/python3.7/dist-packages/ktrain/text/preprocessor.py:215: UserWarnin
      warnings.warn('List or array of two texts supplied, so task being treated as tex
    array([[0.95323986, 0.04676017],
           [0.8406879 , 0.15931216]], dtype=float32)
```

Figure 7 Sentiment score and probability matrix of the Distilbert model

Prediction of sentiment scores of the tweet data array is illustrated in figure 7. Negative sentence array is predicted by using DistilBert model as negative tweets and its probability matrix of identifying the tweets as positive or negative is also drawn. During the training of a machine learning model, the current state of the model at each step of the training algorithm can be evaluated. It can be evaluated on the training dataset to give an idea of how well the model is "learning." It can also be evaluated on a hold-out validation dataset that is not part of the training dataset. Evaluation on the validation dataset gives an idea of how well the model is "generalizing." Train Learning Curve: Learning curve calculated from the training dataset that gives an idea of how well the model is learning. Validation Learning Curve: Learning curve calculated from a hold-out validation dataset that gives an idea of how well the model is generalizing. The below graph figure 4.8 illustrate the training and validation loss of the sentiment 140 dataset. The graph in figure 8 proves as a good fit. The training loss curve slopes down to decrease as the model learns well. The validation curve also slopes well to become stable after 8 epoch at the loss rate of 33%.

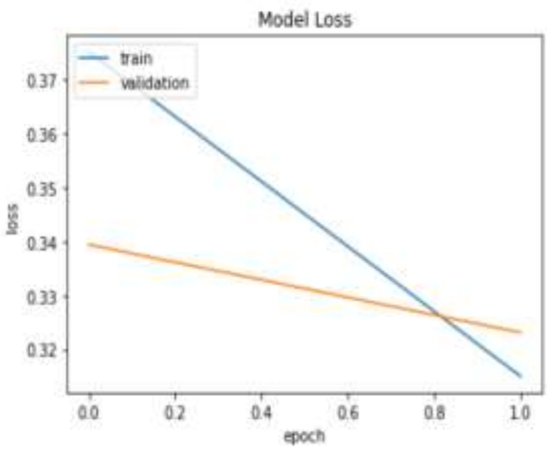

Figure 8 Loss graph illustration of the Sentiment Analysis Classifier

The Performance metrics of the SAC is illustrated in the table 5.Since the F1 score is more than 90% means that low false positive and low false negative so that the SAC almost correctly identify the positive and negative tweets correctly. The recall ratio of correctly predicted positive observations to the all observations in actual class as positive is 0.91 and for negative observation class as negative is 0.92. Precision is a good measure to determine, when the costs of False Positive is high. In detecting the sentiment value of real tweet describing about disaster, a false positive means that a tweet is positive (actual negative) has been identified as negative. The user might be misled if the precision is not high for the SAC model. Thus the accuracy of the SAC model is 91 percentage.

Table 5 Performance metrics of SAC

Aspect	Precision	Recall	F1-score
Positive	0.92	0.91	0.91
Negative	0.91	0.92	0.91
Accuracy		0.91	

7.3.3. Bi-lstm Sentiment Classifier

A hybrid approach of identification of tweet as real or fake by BERT model, collecting the real tweets to do sentiment analysis using BILSTM is deployed and its result is analyzed as follow. Figure 4.9 illustrates the accuracy graph obtained for bidirectional LSTM model the training and validation curve of the model increases at the beginning and is stable after 10 epochs.

Table 6 Performance metrics of BiLSTM model

Aspect	Precision	Recall	F1-score
Negative	0.83	0.86	0.85
Positive	0.86	0.83	0.84
Accuracy		0.84	

The Performance metrics of the BiLSTM model is illustrated in the table 6. The F1 score, precision and recall values are high which is more than 0.5. Thus the model prediction of positive and negative tweets as exactly belonging to positive and negative class is accurate.

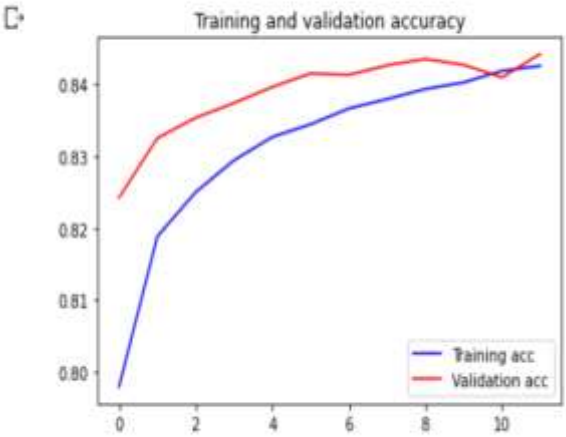

Figure 9 Accuracy graph obtained for BiLSTM model

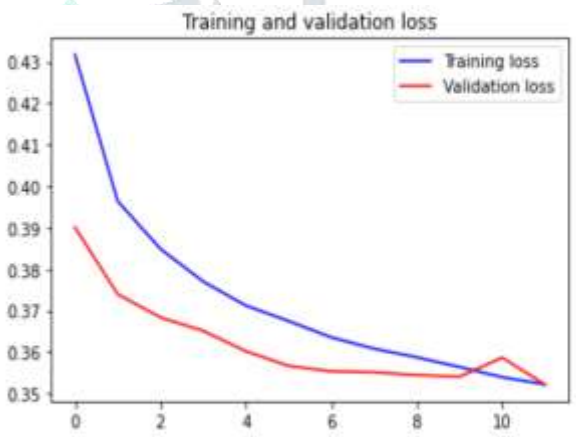

Figure 10 Loss graph obtained for BiLSTM

BiLSTM sentiment analysis is performed by word embedding matrix constructed using word2vec. The training loss curve and validation loss curve tries to decrease as the epoch increases is illustrated in figure 4.10. This proves the model is good fit with less gap between the train and validation curve.

7.3.4. Real Time Tweet Analysis

Polarity scores of the real time tweets collected using Tweet API is generated using TextBlob. Figure 11 describes about unprocessed tweet.

retweetcount	text
1	Natural disasters compilation. \n#climatechang...
1	Natural disasters compilation. \n#climatechang...
0	Some States Still Weigh Mandating Business Int...
3	Hate & racism against the Asian community ...

Figure 11 Unprocessed tweet collected from twitter describing about disaster

text	hashtags	value	senti_score
Natural disasters compilation.	['climatechange', 'disasters', 'weather', 'sto...	0.100000	positive
Natural disasters compilation.	['climatechange', 'disasters', 'weather', 'sto...	0.100000	positive
Some States Still Weigh Mandating Business Int...	['coverage', 'pandemic', 'business', 'interrup...	0.000000	neutral
Hate & racism against the Asian community ...	['COVID19']	-0.291667	negative

Figure 12 Pre-processed tweet with sentiment scores and polarity scores

```
df.senti_score.value_counts()

neutral     46
positive    44
negative    10
```

Figure 13 Value count of total number of positive, negative, neutral tweets for the collected tweets

Real time tweets are quantified with polarity scores using textblob. Based on the polarity scores if the value is positive tweet is quantified as positive, for negative polarity value the tweets are quantified as negative and if the value is zero tweets are quantified as neutral. It is illustrated in figure 4.12 and figure 4.13. Table 4.7 describes the result obtained using proposed work and standard sentiment analysis tool TextBlob for real time collected disaster related tweets. The proposed work DistilBERT and BiLSTM were not trained on neutral disaster tweets hence neutral tweets might either be predicted as positive or negative which is not accounted in the table 7.

Table 7 Polarity of tweets obtained using proposed and current method.

Dataset	Aspect	Positive Count	Negative Count	Neutral Count	Total
Real disaster tweet collected using API	TextBlob	44	10	46	100
	DistilBERT	44	10	-	54
	BiLSTM	35	7	-	42

7.3.5. Comparative Studies

Comparison of real tweet analysis using DistilBERT and BiLSTM is shown in the following table 8 For the binary class problem such as positive or negative performance metrics such as precision, recall and F1 score is used. In the proposed work precision value of DistilBERT model differs from BiLSTM model by 0.08 for Negative class and 0.06 for positive class. The precision value means the model ratio of predicting the tweet as positive/negative tweet from actual positive/negative tweets class to over all tweets.

The recall metrics differ by 0.08 for positive class and 0.06 for negative class means how correctly positive/negative tweets identified among all tweets. The F1 score convey the balance between precision and recall. For the proposed work the difference value of F1 score of negative class is 0.06 and positive class is 0.07.

A. S. Imran [5] acquired F1 score of 80% and accuracy of 80% by implementing sentiment analysis of polarity of the tweet by creating a neural network of embedding layer, max-pool layer and dense layer with the last layer uses sigmoid as an activation function. Word embedding using GloVe pretrained model is employed. Comparing this to the proposed work consider the text of uniform length padding if requires and also tweets that describes about disaster along with the sentence embedding the accuracy and F1 score is 90% for the DistilBERT approach and 84% for BiLSTM word embedding approach. From the table 4, the results obtained using proposed work is same as the tweets polarity scores generated by standard TextBlob.

Table 8 Comparison metrics of proposed methods

Aspect		precision	recall	F1 -score
Difference in performance metrics	Negative	.08	.06	.06
	Positive	.06	.08	.07

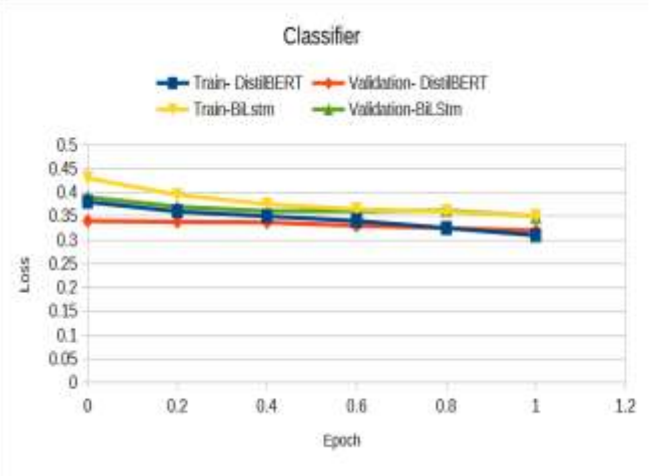

Figure 14 Comparison of loss curve of the DistilBERT and BiLSTM model

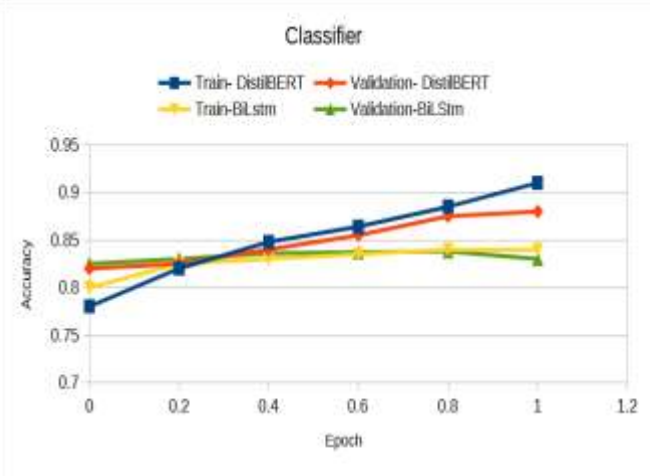

Figure 15 Comparison of accuracy curve of the DistilBERT and BiLSTM model

The loss curve of the both model is decreases and tried to become stable as the epoch increases. The accuracy curve tries to increase and the curve tries to become stable after the Bilstm model reaches 84 % and DistilBERT model reaches 90%.The model tries to reduce the generalization gap between the training curve and validation curve is illustrated in the figure 14 and 15. Hence the over all performance of DistilBERT sentence embedding model for sentiment analysis of disaster tweets is better than BiLSTM word embedding model.

8. CONCLUSION

Evaluating and categorizing a tweet into one that is positive, negative or neutral is an increasingly difficult task. Sentiment analysis of relevant disaster tweets using DistilBERT and BiLSTM is created. For the negative tweets related to disaster the characteristic of the natural disaster must be observed before taking any relief measures. The methods of sentence embedding aim to generate a vector of fixed length from sentences whose lengths may vary. The F1 score of DistilBERT for positive and negative class differ by 0.7 and 0.6 with respect to BiLSTM model. The model acquires accuracy of 84% for BERT along with BiLSTM and 90% for BERT along with DistilBERT. Thus the sentiment scores of sentence embedding is higher than word embedding method. Hence the over all performance of DistilBERT for sentiment analysis of disaster tweets is better than BiLSTM. Since the proposed work was not trained using neutral dataset predicting neutral as either positive or negative is possible. Future scope of the work is to investigate the neutral tweets identified as relevant disaster tweet. The proposed work is compared with the word embedding methods of BiLSTM which is a RNN machine learning model, however it does not assess the performance of other deep neural networks like CNN. Furthermore, to investigate relevant disaster tweet under unsupervised algorithm is also about to be carried out.

REFERENCES

[1] Alsaeedi, A. and Khan, M.Z., "A study on sentiment analysis techniques of Twitter data", International Journal of Advanced Computer Science and Applications, Vol.2, pp.361-374, Feb.2019.

[2] Basiri M. E. et al., "Improving Sentiment Polarity Detection Through Target Identification," IEEE Transactions on Computational Social Systems, Vol. 7, no. 1, pp. 113-128, Feb. 2019.

[3] Gao.Z, Feng.A, Song.X and Wu.X, "Target-Dependent Sentiment Classification With BERT", IEEE Access, Vol. 7, pp. 154290-154299, Oct.2019.

[4] Hassan.A and Mahmood.A, "Convolutional Recurrent Deep Learning Model for Sentence Classification", IEEE Access, Vol. 6, pp. 13949-13957, Mar.2018.

[5] Imran A.S, Daudpota S.M, Kastrati.Z and Batra.R, "Cross-Cultural Polarity and Emotion Detection Using Sentiment Analysis and Deep Learning on COVID-19 Related Tweets", IEEE Access, Vol. 8, pp. 181074-181090, Sep.2020.

[6] Jianqiang.Z, Xiaolin.G and Xuejun.Z, "Deep Convolution Neural Networks for Twitter Sentiment Analysis", IEEE Access, Vol. 6, pp. 23253-23260, Jan.2018.

[7] Jianqiang.Z and Xiaolin.G, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis", IEEE Access, Vol. 5, pp. 2870-2879, Feb.2017.

[8] Kaliyar, R.K., Goswami, A. & Narang, "P. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach", Multimed Tools Appl, Vol.80, pp.11765–11788, Mar.2021.

[9] Krishnamoorthy, S.,"Sentiment analysis of financial news articles using performance indicators", Knowledge and Information Systems, Vol. 56, pp.373-394, Aug.2018.

[10] Pan.Y, Chen.Z, Suzuki.Y, F. Fukumoto and H. Nishizaki, "Sentiment analysis using semi-supervised learning with few labelled data", Cyberworlds (CW), Vol.72, pp. 231-234, Sep.2020.

[11] Prakruthi.V, Sindhu.D and Anupama Kumar D.S, "Real Time Sentiment Analysis Of Twitter Posts", Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Vol.65, pp. 29-34, Dec.2018.

[12] Ramadhani, A.M. and Goo, H.S.,"Twitter sentiment analysis using deep learning methods", IEEE Transactions on Knowledge and Data Engineering, Vol.59, pp. 1-4, Aug.2017.

[13] Saad, Shihab Elbagir, and Jing Yang. "Twitter sentiment analysis based on ordinal regression", IEEE Access, Vol.7, pp.163677-163685, Nov.2019.

[14] Xu.G, Meng.Y, Qiu.X, Yu.Z and Wu.X, "Sentiment Analysis of Comment Texts Based on BiLSTM", IEEE Access, vol. 7, pp. 51522-51532, Apr.2019.

[15] Yang.G, He.H and Chen.Q, "Emotion-Semantic-Enhanced Neural Network," IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 27, no. 3, pp. 531-543, March 2019.

[16] Zhang, D., Xu, H., Su, Z. and Xu, Y., "Chinese comments sentiment classification based on word2vec and SVMperf", Expert Systems with Applications, Vol.42, pp.1857-1863, Mar.2016.