Journal of Theoretical and Applied Information Technology
30th April 2021. Vol.99. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SELF-ATTENTION LAYER MECHANISM BASED MODIFIED BI-DIRECTIONAL LONG SHORT TERM MEMORY FOR TWITTER SENTIMENT CLASSIFICATION

[1] D. RAMANA KUMAR, [2] S. KRISHNA MOHAN RAO

[1] Research Scholar, Department of Computer Science and Engineering, JNTUH, Hyderabad 500085, India.

[2] Principal, GIET, Bhubaneswar 751001, Odisha, India.

E-Mail: [1]ramanad74@gmail.com, [2]krishnamohan6@yahoo.com

ABSTRACT

The rapid growth of social networking services has resulted in the creation of a large amount of explicit information in the form of electronic text. Studies on the emotional analysis of texts have therefore received great interest. Semantic analysis (SA) on social media like Twitter has become a very important and challenging task. Characteristics of such data include the length of the tweet, misspellings, abbreviations, and special characters, where analysing emotions requires an unconventional approach. Additionally, tweets often contain sounds in the form of abbreviations, incorrect grammar, freestyle and grammatical errors. SA aims to estimate real emotions based on the raw human-expressed text in the area of natural language processing (NLP). The main goal of this work is to identify the polarity of the tweets by using raw input data. In this research study, a Modified Bi-directional Long Short Term Memory (MBLSTM) is developed for predicting the sentiments. Initially, raw data is given as input to pre-processing techniques and extracted the important features using two feature extraction techniques. Text blogs are used to identify the polarity of tweets and learning rate of MBLSTM is improved by Self-Attention mechanism. The experiments are conducted on Twitter dataset in terms of accuracy, precision, recall and f-score. The validated results proved that the proposed MBLSTM achieved 93.66% of accuracy and 93% of precision, where traditional BLSTM achieved 90.05% of accuracy and 88.12% of precision.

Keywords: *Bi-directional Long Short Term Memory; Emotions; Polarity; Self-Attention; Sentiment Analysis; Twitter data*

1. INTRODUCTION

Today technology plays an important part in the growth of real business, entertaining industry, and economy. People comment on products / guidelines on social media platforms namely Twitter, Facebook, Instagram, etc. [1] In order to expand their services, they need accurate and consistent feedback from their clients. Typically, the post composition received from these websites is a short text stated in a semi-structured and unstructured form [2]. The generalization of unstructured data can improve the learning process and lead to an efficient polarity determination. However, it is more difficult to generalize Unstructured and semi-structured text because information in different languages is communicated in the form of emotions, unusual words, inappropriate grammar (free text), and small forms around the world. Twitter is the top most of 140-200 character blogs [3,4]. With over a million active users, Twitter has become a goldmine for organizations and individuals with strong political, social and economic interests to maintain and improve their reputations and reputations. Sentiment analysis gives these organizations the ability to review real social network sites. [5].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Sentiment analysis is an automated process to determine whether a text section contains objective or opinion-based content and can also determine the emotional polarity of the text [6]. The purpose of categorizing Twitter's emotions is to automatically determine whether Twitter's emotional data are negative or positive. Text generalization and emotional analysis helps to determine buyers' general opinion about their products [7]. While extracting the user's opinion, the sentiment formulation from the noisy data are not clear. Text mining is conducted out by using lexicon-based and machine learning techniques. Lexicon-based learning needs a large vocabulary with semantic power that is used to evaluate emotions [8,9]. The biggest problem with emotion analysis in NLP is to handle the poor labeled data. To solve this problematic, deep learning has been combined with emotion study to reveal the polarization of unstructured data received from social platforms [10]. This type of combination achieves highly precision emotions through learning algorithms and also works well for determining the polarity of a given file based on trained tagged data. In this research study, a deep learning technique called MBLSTM with SA is developed for twitter sentiment analysis. The features are extracted using two feature extraction techniques and polarities of tweets are identified by using textblog. Accelerated gradient on Rectified Adam Optimizer is included in the MBLSTM to improve the performance of sentiment analysis and SA is used to improve the learning rate of data. At last, the polarity of tweets are classified into positive, negative and neutral by using MBLSTM. The experiments are carried out on Sanders dataset in terms of f-score, error rate, precision, recall and classification accuracy.

The organization of the research work is defined as: The study of existing techniques on twitter sentiment analysis is presented in Section 2. The explanation of proposed MBLSTM is described in Section 3. The validation of proposed method with existing techniques on Sanders dataset is provided in Section 4. Finally, the conclusion of the research work with its future work is illustrated in Section 5.

2. LITERATURE REVIEW

This subsection discusses the recent emotion analysis techniques for determining people's opinions on specific topics. Moreover, the benefits and limitations of existing techniques are discussed.

A. P. Rodrigues, and N. N. Chiplunkar, [11] established Hybrid Lexicon-Naive Bayesian Classifier (HL-NBC) methodology for sentimental analysis. HL-NBC's methods worked with tagged records in real time and filtered out unrelated tweets using a vocabulary model. The HL-NBC method works mainly with large amounts of data. The experimental simulations of the Twitter dataset were performed with precision, precision, recall, f-dimension, and runtime that were used to test HB-NBC method. However, sarcastic tweets were misleading emotion analysis activities and were misclassified. This method doesn't focus on how ironic emotions were filtered out, which lead high error rate.

S.M. Nagarajan, and Usha Devi Gandhi [12] established a hybrid system using particle swarm optimization and genetic algorithm, which were used for sentiment classification. Using a machine learning classifier (Decision Tree (DT)), these methods gave 90% accuracy in classifying total tweets into negative, positive, and neutral. Compared to traditional methods like SVM, DT, hybrid SVM and KNN method, the established approach provides better performance in terms of accuracy, precision, F-score and recall. Due to its immediate nature, spelling errors and high levels of feature space reduced classification accuracy.

M. Z. Asghar, [13] offered a scheme to classify the tweets using hybrid scheme such as slang classifier, an emoticon classifier, SWN classifier and an improved domain-specific classifier as T-SAF, was used to solve the problem of misclassification of emotions. T-SAF technique helps classify errors in tweets, detect and categorize emoji using an expanded emoji dictionary, and improve classification efficiency. The hybrid method's performance results were better in terms of accuracy, precision, F-measurement, and recall compared to some basic methods like RF, SVM, and Artificial Neural Network. Without SWM search operations, one limitation of the methodology was the absence of automatic evaluation of domain-specific words.

I. Alsmadi, and Gan Keng Hoon, [14] examined the characteristics of short texts and presented a supervised term weighting approach (SW) in multi-dimensional vector space. The document strength factor was measured by how the SW dealt with the sparse and shortness of short texts. Experimentations were done with two datasets like Sanders and a collected dataset to test the performance of SW. Associated to unsupervised approaches, the SW technique showed generally better properties. The drawback of this SW method was short texts on social networks that contained wrong spelling, and a high dimensionality of the feature space, resulting in reduced classification accuracy performance.

M. Wang, and G. Hu, [15] proposed the Attentional-graph Neural Network based Twitter Sentiment Analyzer (AGN-TSA). AGN-TSA has combined the text information of the tweet and the connection information of the user through a three-level neural structure comprising the word embedding layer, the user embedding layer and the network of attention graph layer. The parameters such as accuracy, recall, and F1 score were used to check the performance of the AGN-TSA. The method performs poor due to the presence of imbalance data and irrelevant data in tweets.

S. S. Shekhawat, et al., [16] proposed a hybrid strategy called Spider-Monkey Optimization with Clustering to extract emotions from tweets posted on Twitter by finding the optimal cluster header of the dataset. The metaheuristic-based clustering technique was superior to traditional methods due to the subjective behavior of the tweet. Experimental results proved that the method developed in terms of average accuracy and average measuring time that was better than conventional methods. However, this method was not suitable for sarcastic and contradictory tweets. To overcome the above problems, the research study proposed the MBLSTM with the SA system to express user opinions on specific topics. Furthermore, the proposed method determined the polarity of the tweet before removing the tags from the tweet. A brief explanation of the proposed method is provided below.

3. PROPOSED METHODOLOGY

A SA Extraction of opinions from unstructured data is important to get an opinion on an input tweets. Without normalization, SA makes several problems, so pre-processing is performed to learn the sentiment labels. Therefore, tweets are first processed to remove noisy features such as misspellings, abbreviations, and incorrect grammar with an arranged form of information. Figure. 1 shows the process diagram of the proposed method.

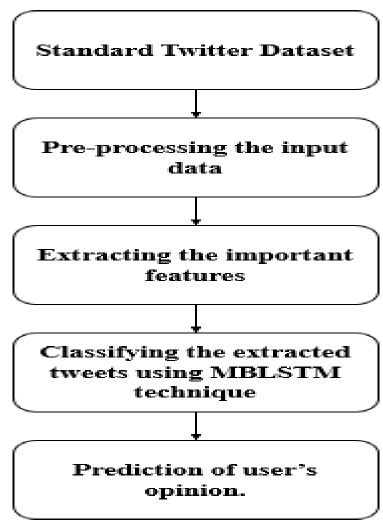

Figure 1: Workflow of proposed method

At first, the raw data are pre-processed into a normalized form. Since noisy Twitter data is used as input for SA, pre-processing is a vital phase to predict polarity accurately. In this way, unstructured data is normalized and then sentiment is extracted using feature extraction techniques. The pre-processing phase is separated into two phases: (a) tokenization and (b) lemmatization and stemming. Latent dirichlet allocation (LDA) is used in initialization process,

Journal of Theoretical and Applied Information Technology
30th April 2021. Vol.99. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

where final classification is carried out using MBLSTM technique.

3.1. Data Collection

The data is captured with a URL using the Twitter Streaming API Real-time access is given by the public streaming API with 1% of all public tweets, but direct messages is not provided. Since tweets data are in JSON format, each line can easily be parsed as objects. This paper only focuses on mood classification using the MBLSTM method, which only receives a text attribute to determine polarity. For experimental purpose the STC dataset is used to analysis in MBLSTM, with this dataset containing a total of 5513 tweets for SA [17].

There are totally four categories in STC such as Google, Apple, Microsoft, and Twitter, where these groups are available in the following link: http://www.sananalytics.com/lab/twittersentiment/. There are several tweets in each category to help you get people's opinion on the products. The proposed MBLSTM defines three polarity classes as positive, negative, and neutral. These recognised polarities are then categorized to the "1" for positive class, "0" for the neutral class and "-1" for the negative class. The next step is pre-processing the raw input data, which is described as follows:

3.2. Pre-Processing

In pre-processing steps, tokenization, lemmatization and stemming are used to remove the irrelevant or any unwanted data from the raw input.

Tokenization: The original input is taken from the Twitter record, which have more number of unneeded characters such as comma, punctuation, period, semicolon, exclamation mark, hyphen, curly bracket, curly bracket, question mark, apostrophe and quotation marks. For instance, consider the subsequent example such as "HELLO world how was your day? "The result of the text after tokenization is <HELLO> <World> <How> <is> <the> <day> <for> <you>. Here the (?) is discarded and the residual list the token is directed for further processing.

Lemmatization: It is performed to convert various word forms into a basic word. Part of Speech (POS) tagging is done in advance on the input record. Here the sentence having the POS is compared with the given text. Then lemmatization is performed to group the same words into one element. This will transform the informal word into its root word. For sample, the words "best", "best", "good" are transformed to "good" because all the words have the similar meaning.

Stemming: Here, a stem extraction is achieved to find the term in the Twitter sequence. The lemmatization minimized the number of words in a dictionary sentence, but stemming lessens the word into a root word. This is done using Porter's steamer algorithm, which lessens the word to its original form. It does this by breaking the prefix or suffix and replacing a word that has no significance with the English dictionary.After that pre-processing, the following stage is to determine the key characteristics of the pre-processed data. The final result is a sequence of strings provided as input to retrieve the information. However, before extracting useful information, determine the polarity of the tweet. Existing approaches recognize the tweets polarity after removing features that lead to poor ranking functionality. In this work, a text blog function [17] is used to determine the polarity of the text and improve the accuracy of the classification. Typically in traditional schemes, sentences are used entirely to determine polarity. In this work, the sentences are divided into each letter, then the polarity is determined using a text blog. This process speeds up the MBLSTM for tweet classification.

3.3. Feature Extraction

After determining the polarity, the next step is to derive the properties from the pre-processed tweet. In this work, feature information contains analysis of the frequency of negative and positive words, number of tags in the text, rating of negative and positive words, and complete

Journal of Theoretical and Applied Information Technology
30th April 2021. Vol.99. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

rating of sentence. This study uses new feature information systems to extract features from tweet data by using the two methods such as counter vectorization and term frequency-inverse document frequency (TF-IDF).

Counter vectorization

This is the principle for introducing or removing a reference word from a large group of text. Each word is assigned a series of mutual difficulties to determine the similarity among the obtainable words. Counter vectorization is usually done in two conducts; (i) Practices are used to roughly estimate the substitutability of contextual use of words. (ii) Computational descriptions and progressions are used to highlight the aggregation of aggregated text data.

TF-IDF

TF-IDF is used to extract the features from text data. The frequent appears of term in a document are recognised by using TF-IDF, which is scientifically show in the equations (1) and (2).

$$Term\ Frequency = \frac{No.of\ times\ term\ (t)\ appears\ in\ document}{Total\ no.of\ terms\ in\ a\ document} \quad (1)$$

$$Inverse\ document\ frequency = \log\frac{Total\ no.of\ documents}{No.of\ documents\ with\ term(t)} \quad (2)$$

Two methods (TF-IDF and counter vectorization) are used to extract useful information from tweets. In later phases, the LDA methodology

is initialized to classify tweets into three types. This process collected a total of 19 features from all tweets, which are divided into three classes.

3.4. Latent dirichlet allocation

Afterward the information is extracting from the composed data, LDA is used to divide the tweets into three categories [19]. Initially, the hidden topics are called distributed sentences. Using observable data, hidden subjects are easily identifiable. Each document creates a two-step process in which the topics distributed for each document are selected in the first step. When building the corpus, the study uses the two parameters π and μ for the three-layer representation in LDA. The object documents are then verified for each and every document. In LDA, all words in the document are examined to create a word-level variable that is seen as a joint allocation that generates an LDA process. Equation (3) is used to determine the random variable for the K-dimensional dirichlet potential density function. The combined distribution of the topic mixture and the potential of the corpus are measured by equations (4) and (5).

$$p(\aleph|\pi) = \frac{\Gamma\left(\sum_{i=1}^{k}\pi_i\right)}{\prod_{i=1}^{k}\Gamma(\pi_i)}\aleph_1^{\pi_1-1}\ldots\ldots\aleph_k^{\pi_k-1} \quad (3)$$

$$p(\aleph, x, y|\pi, \mu) = p(\aleph|\pi)\prod_{n=1}^{N} p(x_n|\aleph)\, p(y_n|x_n, \beta) \quad (4)$$

$$(D|\pi, \mu) = \prod_{d=1}^{M}\int p(\aleph_d|\pi) \times \left(\prod_{n=1}^{N_d}\sum_{x_{dn}} p(x_{dn}|\aleph_d)p(y_{dn}|x_{dn}, \mu)\right)d\aleph_d \quad (5)$$

Where, $\aleph$ is signified as document-level topic variables, π is stated as dirichlet parameter, document is labelled as M, sum of words is accessible as N, μ defines the topics, x is designated as per-word topic task, and y is specified as observed word.

An important function of the LDA is to measure the estimation of the posterior distribution of hidden variables in a document, known as an interactive problem. The values of the separate weights of the three classes are kept in a dictionary, in which the weighted class values are determined in the dictionary. After receiving neutral, negative and positive weights, the learning rate is processed using the SA layer in MBLSTM to overcome the information loss' limitation.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.5. Classification

The classification process is performed using this algorithm, after obtaining neutral, negative and positive weights. First, BLSTM uses two LSTMs to procedure sequences in two directions includes forward and backward [20]. Thus, forward and backward events are deliberated at the same time. The calculation of Bi-LSTMs is formulated as follows in Eq. (6-7):

$$\overrightarrow{h_t} = lstm\left(\overrightarrow{h_{t-1}}, e(w_t)\right) \quad (6)$$

$$\overrightarrow{h_t} = lstm\left(\overline{h_{t-1}}, e(w_t)\right) \quad (7)$$

Then, the order of forward and backward hidden states are considered as the sign of each word w_t, and the demonstration is specified as $h_t = [\overrightarrow{h_t}; \overline{h_t}]$. BLSTM is an adaptive technique that determines the degree of previous conditions and also processes the characteristics extracted from the input. In some cases, the traditional BLSTM ideal has long sentences as input data. To solve this problematic, MBLSTM is used to encode sentences and get an implicit vector for every time. This MBLSTM establishes an adaptive mechanism that can determine the extent to which the previous position is maintained and preserve the detected properties of the existing input. Long sentences are encoded as hidden vectors of a certain length so that information is compiled for long sentences. Thus, the output vector of the last time stage can accurately expose the meaning of long sentences. For this process, the accelerated gradients on Rectified Adam Optimizer is used in this research study.

Rectified Adam Optimizer:

Adam is another method that calculates the adaptive learning rate of each parameter [19]. As Adamdelta and RMS prop, the rear quadrilateral gradient not only maintains the velocity decreasing velocity of v_t, but also maintains the velocity decreasing average of the previous gradient as well as the speed. Getting gradients on time compared to a stochastic object at timestep t.

$$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1}) \quad (8)$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (9)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \quad (10)$$

m_t and v_t are approximately for the first moment (center point) and the second moment (center variation), respectively. Adams authors, represented as vectors m_t and v_t, find that they tend to be biased towards zero, especially in the early stages of the period and particularly when the decay rate is low (i.e. β_1 and β_2 are close to 1). Compute the bias-corrected first and second moment to counteract these biases, which is shown in Eq. (11-12):

$$m_t^{new} = \frac{m_t}{1-\beta_1^t} \quad (11)$$

$$v_t^{new} = \frac{v_t}{1-\beta_2^t} \quad (12)$$

The exponential moving average (EMA) can be understood as an estimate to the simple moving average (SMA) [22].

$$\rho\left(\frac{(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}g_i^2}{1-\beta_2^t}\right) \approx \rho\left(\frac{\sum_{i=1}^{f(t,\beta_2)} g_{t+1-i}^2}{f(t,\beta_2)}\right) \quad (13)$$

Where ρ is the degree of freedom and approximation of ρ based on t to conduct a quantitative analysis. The algorithm of Rectified Adam Optimizer is given as follows:

Algorithm 1: Rectified Adam Optimizer
Input: $\{\alpha_t\}_{t=1}^{T}$: $step\ size$, $\{\beta_1, \beta_2\}$: decay rate to calculate moving average and moving 2nd moment, θ_0: initial parameter, $f_t(\theta)$: stochastic objective function. **Output**: θ_t: resulting parameters 1. $m_0, v_0 \leftarrow 0, 0$ (Initialize moving 1st and 2nd moment) 2. $\rho_\infty \leftarrow 2/(1-\beta_2) - 1$ (compute the maximum length of the approximated SMA) 3. while $t = \{1, \dots, T\}$ do

a. $g_t \leftarrow \Delta_\theta f_t(\theta_{t-1})$ (Calculate gradients w.r.t stochastic objective at timestep t)
b. $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$ (Update exponential moving 2nd moment)
c. $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$ (Update exponential moving 1st moment)
d. $\widehat{m}_t \leftarrow m_t/(1-\beta_1^t)$ (Compute bias-corrected moving average)
e. $\rho_t \leftarrow \rho_\infty - 2t\beta_2^t/(1-\beta_2^t)$ (Compute the length of the approximated SMA)
f. if the variance is tractable, i.e., $\rho_t > 4$ then
   i. $\widehat{v}_t \leftarrow \sqrt{\frac{v_t}{1-\beta_2^t}}$ (Compute bias-corrected moving 2nd moment)
   ii. $r_t \leftarrow \sqrt{\frac{((\rho_{(t-4)})(\rho_t-2)\rho_\infty)}{(\rho_\infty-4)(\rho_\infty-2)\rho_t}}$
   iii. $\theta_t \leftarrow \theta_{t-1} - \alpha_t r_t \widehat{m}_t/\widehat{v}_t$ (Update parameters with adaptive momentum)
g. else
h. $\theta_t \leftarrow \theta_{t-1} - \alpha_t \widehat{m}_t$ (Update parameters with un-adapted momentum)

4. return θ_t

Then, they use them to update the parameters, as we seen in Adadelta and RMSprop, which create the Adam update rule as in Eq. (14):

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}^{new}_t}+\epsilon} m_t^{new} \tag{14}$$

The authors propose standard values of 0.9 for β_1, 0.999 for β_2, and 10^{-8} for ϵ. In addition, this study uses a self-attention model that uses Veritable Length Weighted Context Structure (VLWCS) and improves classification accuracy. From these classification results, the emotional analysis of the tweet can be determined.

Consider the sequence $S = \{x_1, x_2, \dots, x_l\}$, a, where l is described as the length of the sequence and the MBLSTM literally trains the model word. Each time for t, the state of the memory cell m_t and the hidden state h_t are restructured, as shown arithmetically in the equations (15), (16), and (17).

$$\begin{pmatrix} In_t \\ Fo_t \\ Out_t \\ \hat{c}_t \end{pmatrix} = \begin{pmatrix} S \\ S \\ S \\ T \end{pmatrix} W.[h_{t-1}, i_t] \tag{15}$$

$$m_t = Fo_t \odot m_{t-1} + In_t \odot \hat{c}_t \tag{16}$$

$$h_t = Out_t \odot T(m_t) \tag{17}$$

Where, input is defined as i_t and memory cell's current state is illustrated as $\hat{c}_t$. The In_t, FO_t *and* Out_t are specified as the results of the enter, forget and exit function. In addition, S is called the logistic sigmoid function and T is called the $tanh$ function. The MBLSTM is efficient for extended sentences and also uses the SA model utilizing VLWCS. SA is described below.

Consider a context vector $C = \{c_1, c_2, \dots, c_n\}$ for MBLSTM, where V is signified as encoder output, and W is designated as linear transformation coefficients for V, as shown in equation (18).

$$V' = L(V) = W \times V \tag{18}$$

A vector of variable length a_n is given as weighting and C_w as weighted context. The location-based functionality is then estimated using the VLWCS, which is scientifically given in the equations (19), (20), and (21).

$$a_n = softmax([c_1^T * V', c_2^T * V', \dots, c_l^T * V']) \tag{19}$$

$$C_w = (c_1 * a_n + c_2 * a_n +, \dots, + c_n * a_n) \tag{20}$$

Journal of Theoretical and Applied Information Technology
30th April 2021. Vol.99. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\hat{C}_n = [C, C_W] \quad (21)$$

At last, the final probability $p(y|x) = softmax(MLP([\hat{C}_n, h_n]))$ are obtained, where the SA process improves the performance of MBLSTM for classifying the context. The working procedure of MBLSTM-SA is explained as pseudo code.

Pseudo Code: Algorithm: MBLSTM with SA Layer

1. **Procedure**: MBLSTM
2. **Class** MBLSTM[tfidf_features,Tweets_SA,Twitter_text]
3. $Tfidf \leftarrow$ tfidf_vectorizer.fit_transform(tweetsText) // Twitter features Extracting using TFIDF
4. $SentiAnalysis \leftarrow$ Analysis(tweetsText) // Calculate sentiment analysis
   (Tweet_trainX, Tweet_testX, Tweet_trainY, Tweet_testY) ← train_test_split
   (Twitter_features, Tweets_SA, test_size = 0.3)
   // spitting the data into training and testing for the MBLSTM model
5. Tweet_trainX = sequence.pad_sequences(Tweet_trainX, maxlength = 1000)
   Tweet_testX = sequence.pad_sequences(Tweet_testX, maxlength = 1000)
6. **def** model:

//MBLstm Model Initialization

```
Model ← Sequential()
Adding SelfAttention Layer ← SeqSelfAttention(attention_activation ='relu')
opt = RAdam(total_steps = 5000, warmup_proportion = 0.1, min_lr = 1e-5)
Compile ← model.compile(optimizer = opt, 'sparse_categorical_crossentropy',
              metrics = ['accuracy','mae','mse','rmse'])
Model_Fit ← model.fit(Tweets_trainX, Tweets_trainY, batch_size = BatchSize, epochs = 50,
              validation_data = [Tweets_testX, Tweets_testY])
Model_Prediction ← model.predict_classes(Tweets_testY)
```

7. **End function**
8. **def** Senti_Analysis:

```
Senti_analysis = Text Blob (Tweets_text)  // Text Blob utility function
                                             for sentiment polarity
If Senti_analysis.senti.polarity > 0:
    return 1
elif Senti_analysis.senti.polarity == 0:
    return 0
else:
    return -1
```

9. **End function**
10. **End Class**

Journal of Theoretical and Applied Information Technology
30th April 2021. Vol.99. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The validations of MBLSTM with SA with existing techniques are described as follows:

4. RESULTS AND DISCUSSION

In this section, the experimental results of proposed MBLSTM are described, where the simulations are carried out in the system of Intel Core i5, 8GB RAM with 2.2 GHz using Python language of version 3.7.3. The important parameters namely precision, recall, accuracy, f-score, Mean Absolute Error (MAE), Mean Square Error (MSE) and Root of MSE (RMSE) are used

for validation of proposed MBLSTM. The following section describes the quantitative and qualitative analysis of BLSTM-SA along with parameter evaluation.

4.1. Parameter Evaluation

Precision is defined as the documents, which are truly positive and the mathematical expression is explained in Eq. (22). A portion of positive documents that are categorized as positive for a given number of documents is defined as recall and mathematical expression is illustrated in Eq. (23).

To determine the twitter sentiment analysis, the measurement of random errors and statistical variability is defined as overall accuracy, which is shown in Eq. (24). F-measure is defined as the calculation of score by considering the precision and recall and shown in Eq. (25).

The other parameters are described in Eq. (26-27):

$$Precision = \frac{TP}{TP+FP} \quad (22)$$

$$Recall = \frac{TP}{TP+FN} \quad (23)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (24)$$

$$F-score = \frac{2\,TP}{2TP+FP+FN} \quad (25)$$

$$MAE = 1/N \sum_{j=1}^{N} |y_n - x_n| \quad (26)$$

$$MSE = 1/n \sum_{j=1}^{n} |\hat{y}_i - y_i|^2 \quad (27)$$

Where, TP is denoted as true positive, TN is indicated as true negative, FP is represented as false positive, and FN is denoted as false negative.

4.2. Classification of Ternary

In order to classify the tweets polarity, the proposed MBLSTM is used. The input tweets are categorized into three classes includes ‘‘positive as P,’’ ‘‘neutral as Neu,’’ and ‘‘negative as N’’. In this validation, the proposed MBLSTM is compared with existing techniques: SVM, RF [12], standard BLSTM and BLSTM-SA. The classification results for proposed MBLSTM with existing techniques are presented in Table 1.

Journal of Theoretical and Applied Information Technology
30th April 2021. Vol.99. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Classification Results of Proposed MBLSTM for three categories

Methods	*Precision*			*Recall*			*F-score*		
	P	**N**	**Neu**	**P**	**N**	**Neu**	**P**	**N**	**Neu**
SVM [12]	0.3	0.10	0.56	0.54	0.08	0.34	0.38	0.09	0.42
RF [12]	0.54	0.39	0.67	0.41	0.05	0.85	0.47	0.09	0.75
BLSTM	0.92	0.46	0.86	0.74	0.46	0.95	0.82	0.46	0.91
BLSTM-SA	0.92	0.74	0.95	0.91	0.68	0.95	0.91	0.71	0.95
Proposed MBLSTM	**0.96**	**0.89**	**0.93**	**0.93**	**0.85**	**0.94**	**0.95**	**0.85**	**0.94**

The proposed MBLSTM achieved higher precision, recall and f-score than all existing techniques on positive and negative classes. For instance, the MBLSTM achieved nearly 93% to 96% of precision, recall and f-score on positive class and also achieved nearly 85% to 89% of precision, recall and f-score on positive class. However, MBLSTM achieved less performance than existing

BLSTM-SA on neutral class, this is because the distributions of tweets are high only on positive and negative classes. For example, proposed MBLSTM achieved 93% of precision, 94% of recall and f-score on neutral class, where BLSTM-SA achieved 95% of precision, recall and f-score on neutral class. The existing SVM and RF achieved very poor performance on all classes, when compared with BLSTM and BLSTM-SA. The reason is the inclusion of SA mechanism in the deep learning for predicting the learning rate. In addition, the comparative analysis of proposed MBLSTM with traditional methodologies by means of recall,

f-score and precision for overall tweets in the following Table 2 and graphical representation is illustrated in Figure 2.

Table 2: Performance Analysis of MBLSTM with existing techniques

Methodology	**Precision**	**Recall**	**F-score**
Binary SVM [14]	58.13	47.51	47.49
TFIDF-SVM [14]	65.82	65.24	65.17
SW-SVM [14]	81.16	80.85	80.83
PCA-RF [13]	74.74	74.20	72
BLSTM	88.12	92.31	90.17
BLSTM-SA	91	93	91
Proposed MBLSTM	93.0	91.0	91.0

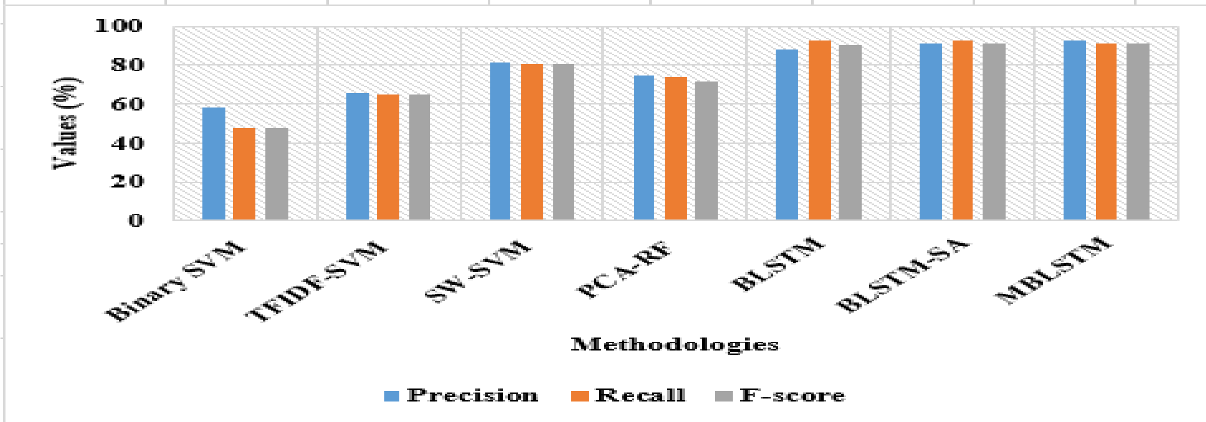

Figure 2: Graphical Representation of proposed MBLSTM in terms of precision, recall and F-score

The existing technique SVM is implemented with feature extractions namely Binary, SentiWord (SW) and TF-IDF achieved less precision, recall and f-score. For instance, TF-IDF- and Binary-SVM achieved nearly 50% to 65% of precision, recall and f-score, where SW-SVM achieved nearly 80% of precision, recall and f-score. The reason is that the SW extracted the related features of positive, negative and neutral classes by using SVM classifiers. While comparing with TF-IFD, PCA-RF achieved high performance (i.e. nearly 74%) on overall tweets, this is because RF overcomes the limitation of overfitting on tweets. While implementing the deep learning on sentiment analysis, BLSTM achieved nearly 89% to 93% of precision, recall and f-score. The existing BLSTM-SA achieved higher performance than BLSTM due to inclusion of SA mechanism. However, the BLSTM-SA achieved poor performance, when comparing with proposed MBLSTM. For instance, BLSTM-SA achieved 91% of precision and MBLTSM achieved 93% of precision due to accelerated gradients on Rectified Adam optimizer.

4.3. Quantitative Analysis of Proposed MBLSTM in Terms of Accuracy

In this section, the performance of proposed MBLSTM is validated with different existing techniques that is shown in Table 3 and Figure 3.

Table 3: Comparative Analysis of Proposed MBLSTM in terms of Classification Accuracy

Methodology	Accuracy (%)
SVM [12]	37.06
RF [12]	63.03
HL-NBC [11]	82
Hybrid optimization with Decision Tree [12]	90
T-SAF [13]	85
BLSTM	90.05
BLSTM-SA	91.41

Journal of Theoretical and Applied Information Technology
30th April 2021. Vol.99. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Proposed MBLSTM	**93.66**

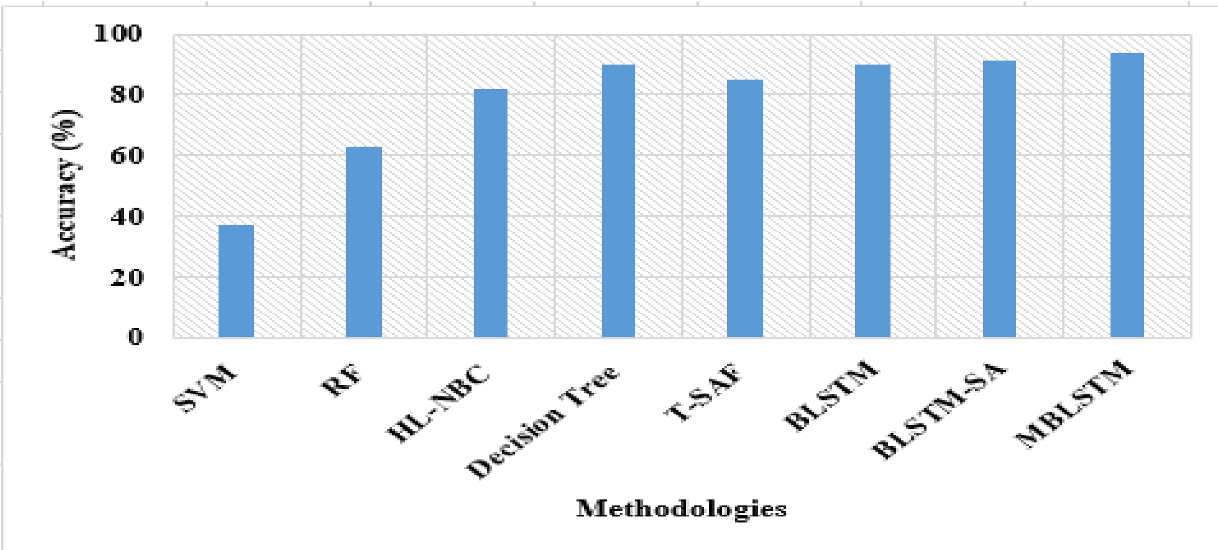

Figure 3: Graphical Representation of Proposed MBLSTM in terms of Accuracy

From the Table 3, it is shown that SVM achieved less classification accuracy (i.e. 37.06%) than other techniques, because SVM is insufficient to handle vast amount of tweets data for processing. The other techniques namely T-SAF, hybrid optimization with decision tree and HL-NBC achieved nearly 83% to 90% of classification accuracy, however deep learning techniques are required for the sentiment analysis. Therefore, BLSTM and BLSTM with SA achieved nearly 91% of accuracy, but these BLSTM are not suitable for long sentences. In order to solve these issue, proposed MBLSTM is developed with rectified adam optimization and also SA is also included in the MBLSTM for speed up the learning rate. Hence, the MBLSTM achieved 93.66% of classification accuracy, which shows better performance than other existing techniques.

4.4. Quantitative Analysis By Means Of Error Rate

The error rate namely MSE, RMSE and MAE are used to validate the performance of proposed MBLSTM with existing techniques, which is provided in Table 4 and Figure 4.

Table 4: Performance Of MBLSTM In Terms Of Error Rate Analysis

Methods	RMSE	MSE	MAE
SVM [12]	0.92	0.85	0.70
RF [12]	0.76	0.58	0.44
BLSTM	0.60	0.36	0.23
BLSTM-SA	0.43	0.19	0.12
Proposed MBLSTM	**0.41**	**0.16**	**0.10**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

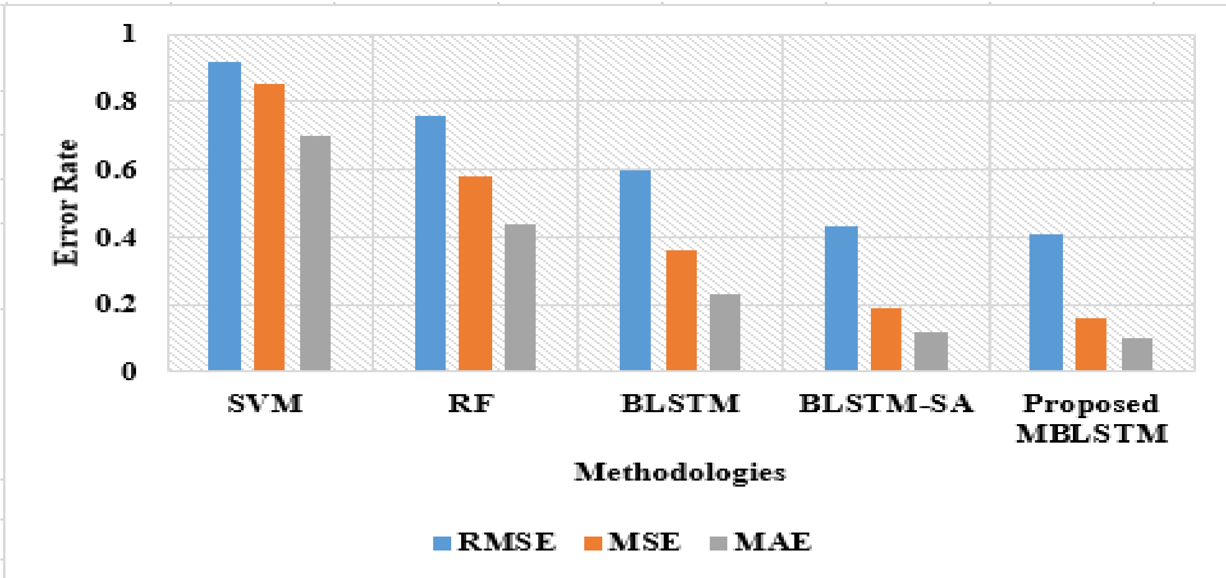

Figure 4: Error Rate Analysis Of Proposed MBLSTM

The SVM has highest RMSE (i.e.0.92) than other existing techniques and proposed MBLSTM. In addition, MSE and MAE are also high in the SVM technique, where RF also has high RMSE (i.e. 0.76), 0.58 of MSE and 0.44 of MAE than BLSTM. While implementing the SA mechanism in BLSTM, it achieved less RMSE (i.e. 0.43), 0.19 of MSE and 0.12 of MAE. However, BLSTM-SA achieved less performance due to long sentence tweets in the input data. Therefore, Rectified Adam Optimizer with BLSTM-SA as MBLSTM is developed in the study and achieved less error rate (i.e. 0.41 of RMSE, 0.16 of MSE and 0.10 of MAE). Therefore, the proposed MBLSTM achieved better performance than existing techniques in terms of accuracy, f-score, error rate, precision and recall.

5. CONCLUSION

Nowadays, sentiment analysis plays a major role on social media platforms namely YouTube, Twitter and Facebook to identify the opinion of the end users about products or services. In this research study, Twitter dataset is considered, because more than millions of end-users express their opinions in Twitter. However, analysing the vast amount of tweets are difficult for sentiment analysis. Initially, the pre-processing techniques are used to remove the stop words and noisy data from the raw input data. Then, text blogs are used to identify the polarity of the tweets and features are extracted by using counter vectorization and TF-IDF, where LDA is used to initialize the tweets into three classes namely positive, negative and neutral. At last, the classifications of tweets are carried out by using MBLSTM with SA mechanism and Rectified Adam Optimizer is used to improve the performance of MBLSTM. STC dataset is used for validation of the proposed MBLSTM with existing techniques namely BLSTM, RF, SVM and BLSTM with SA. While comparing with the BLSTM with SA, the proposed MBLSTM minimized nearly 2% to 3% of error rate. In addition, the MBLSTM achieved 93.66% of classification accuracy, where BLSTM and BLSTM-SA achieved 91% of classification accuracy on whole tweets. As a future work, an effective optimization techniques are used with deep learning techniques on Twitter dataset for improving the classification accuracy.

REFERENCES

[1] Hanafiah, N., Kevin, A., Sutanto, C., Arifin, Y. and Hartanto, J., 2017. Text normalization algorithm on twitter in complaint category. *Procedia computer science*, *116*, pp.20-26.

[2] Vinodhini, G. and Chandrasekaran, R.M., 2012. Sentiment analysis and opinion mining: a survey. *International Journal*, *2*(6), pp.282-292.

[3] Go, A., Bhayani, R. and Huang, L., 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, *1*(12), p.2009.

[4] Saif, H., He, Y. and Alani, H., 2012, November. Semantic sentiment analysis of twitter. In *International semantic web conference* (pp. 508-524). Springer, Berlin, Heidelberg.

[5] Bansal, B. and Srivastava, S., 2019. Lexicon-based Twitter sentiment analysis for vote share prediction using emoji and N-gram features. *International Journal of Web Based Communities*, *15*(1), pp.85-99.

[6] Alharbi, A.S.M. and de Doncker, E., 2019. Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information. *Cognitive Systems Research*, *54*, pp.50-61.

[7] Arora, M. and Kansal, V., 2019. Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis. *Social Network Analysis and Mining*, *9*(1), p.12.

[8] Wang, Y. and Youn, H.Y., 2019. Feature Weighting Based on Inter-Category and Intra-Category Strength for Twitter Sentiment Analysis. *Applied Sciences*, *9*(1), p.92.

[9] Jianqiang, Z., Xiaolin, G. and Xuejun, Z., 2018. Deep convolution neural networks for twitter sentiment analysis. *IEEE Access*, *6*, pp.23253-23260.

[10] Vateekul, P. and Koomsubha, T., 2016, July. A study of sentiment analysis using deep learning techniques on Thai Twitter data. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 1-6). IEEE.

[11] A. P. Rodrigues, and N. N. Chiplunkar. "A new big data approach for topic classification and sentiment analysis of Twitter data." *Evolutionary Intelligence* (2019): 1-11.

[12] S. M. Nagarajan, and Usha Devi Gandhi. "Classifying streaming of Twitter data based on sentiment analysis using hybridization." *Neural Computing and Applications* 31.5 (2019): 1425-1433.

[13] M. Z. Asghar, F. M. Kundi, S. Ahmad, A. Khan, and F. Khan, (2018). “T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme”. *Expert Systems*, 35(1), e12233.

[14] Alsmadi, Issa, and Gan Keng Hoon. "Term weighting scheme for short-text classification: Twitter corpuses." *Neural Computing and Applications* 31.8 (2019): 3819-3831.

[15] Wang, M. and Hu, G., 2020. A Novel Method for Twitter Sentiment Analysis Based on attentional-Graph Neural-Network. *Information*, *11*(2), p.92.

[16] Shekhawat, S.S., Shringi, S. and Sharma, H., 2020. Twitter sentiment analysis using hybrid Spider Monkey optimization method. *Evolutionary Intelligence*, pp.1-10.

[17] H. Saif, Fernandez, M., He, Y., & Alani, H. “Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold”, 2013

[18] Hasan, A., Moin, S., Karim, A., & Shamshirband, S. (2018). Machine learning-based sentiment analysis for twitter accounts. *Mathematical and Computational Applications*, 23(1), 11.

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

[19] F. Colace, M. De Santo, L. Greco, V. Moscato, and A. Picariello, "Probabilistic approaches for sentiment analysis: Latent dirichlet allocation for ontology building and sentiment extraction", In *Sentiment Analysis and Ontology Engineering*, *Springer, Cham*, pp. 75-91, 201.

[20] Z. Ding, R. Xia, J. Yu, X. Li, and J. Yang, "Densely connected bidirectional lstm with applications to sentence classification". *In CCF International Conference on Natural Language Processing and Chinese Computing* (pp. 278-287). Springer, Cham.

[21] Melinte, D.O. and Vladareanu, L., 2020. Facial Expressions Recognition for Human–Robot Interaction Using Deep Convolutional Neural Networks with Rectified Adam Optimizer. *Sensors*, *20*(8), p.2393.

[22] Jiang, X., Hu, B., Chandra Satapathy, S., Wang, S.H. and Zhang, Y.D., 2020. Fingerspelling Identification for Chinese Sign Language via AlexNet-Based Transfer Learning and Adam Optimizer. *Scientific Programming*, *2020*.