Bi-LSTM Learning Model for The Classification of the Opinion from the Dynamic Natural Text Corpus

Patil N S[1], Poornima B[2], Preethi Basavaraj[3], Vinutha H P[4], Puneeth S P[5]

Associate Professor, Bapuji Institute of Engineering and Technology, Davangere-577004, India, Email: patilns_12@rediffmail.com
Professor, Bapuji Institute of Engineering and Technology, Davangere-577004, India, Email: poornimateju@gmail.com
Assistant Professor, Bapuji Institute of Engineering and Technology, Davangere-577004, India, Email: preethib027@gmail.com
Professor, Bapuji Institute of Engineering and Technology, Davangere-577004, India, Email: vinuprasad.hp@gmail.com
Assistant Professor, Bapuji Institute of Engineering and Technology, Davangere-577004, India, Email: punithshetty8@gmail.com

ABSTRACT

Opinion classification deals with the area of human behavior analysis and attempts to develop an automated system to determine the viewpoint of people towards a variety of units such as events, topics, products, services, organizations, individuals, and issues. Opinion analysis from the natural text can be regarded as a text and sequence classification problem which poses high feature space due to the involvement of dynamic information that needs to be addressed precisely. This paper introduces effective modeling of human opinion analysis from the social media data subjected to complex and dynamic content. Firstly, a customized preprocessing operation based on natural language processing(NLP) mechanisms as an effective data treatment process towards building quality-aware input data. On the other hand, a suitable deep learning technique, namely Bi-LSTM, is implemented for the opinion classification, followed by a data modeling process where truncating and padding is performed manually to achieve better data generalization in the training phase. The design and development of the model are carried on the MatLab tool. The performance analysis has shown that the proposed system offers a significant advantage in terms of classification accuracy along with less training time due to a reduction in the feature space by the data treatment operation.

Keywords: Bi-LSTM and NLP

1. INTRODUCTION

Opinion analysis is used for many purposes, such as determining the mood of social media users about a topic, their views on social events, and market price, and products [1], [2]. On the other hand, Twitter is widely preferred as a data source in opinion and sentiment analysis studies because it is a popular social network and convenient for collecting data in different languages and content [3]. However, considering opinion analysis as a text and sequence classification problem. However, the social media data is composed of short texts, dynamic representation, due to which the text corpus becomes sparse and semi-unstructured [4]. This poses a significant problem in terms of system response time and classification performance, especially on large text corpus [5]. For this reason, various preprocessing, text representation, and data modeling techniques are used to address the issues associated with classification performance arising from sparse data quality and high feature space. Text representation can be realized with meaningful information extracted from text content in traditional methods such as Bag of Words (BOW) Skip-gram and N-grams [6-8]. In the BoW model, attributes are words extracted from text content, and the order of these is not much important. The n-gram model, which can be applied at the word and character level, is generally more successful at the character level and is robust against situations such as spelling mistakes and the use of abbreviations because it is language-independent. On the other hand, the skip gram is an unsupervised mechnaims that determines the most relevant words for a given text. The structural and statistical properties of structured or semi-structured texts are also used in the text representation. In addition to these traditional methods, various methods based on graphs, linear algebra, and LDA (latent Dirichlet allocation) [9-10] are used to address text classification problems. In addition, there are studies where the very popular word-to-vectors method is also used in text representation in lower space [11]. In the word-to-vector method, an n-dimensional vector of each word seen in the document is obtained, and these vectors are clustered [12]. Documents are represented with the number of members (words) in each set of the respective document, and thus texts are represented in lower dimensions. With the advancement in machine learning and deep learning techniques, the existing literature presents various models for mining opinions from the text and, more specifically, from social media data. Among which recurrent neural network (RNN) and long-short-term-memory (LSTM) are widely adopted in the context opinion classification from the rich natural language [13-14]. The RNN and LSTM are the most suitable model for the sequence classification problems. The problem of opinion mining can also be regarded as a sequence classification task as the text sentences consist of multiple chunks of a word in a sequence to represent meaningful information. The work carried out by Pergola et al. [15] suggested a deep learning model for the topic-oriented attention system towards sentiment analysis. The outcome shows adoption of the RNN offers better higher accuracy in the prediction phase. The work of Ma et al. [16] implemented a variant of RNN, namely LSTM, with a layered attention mechanism for opinion mining. The study exhibited that their model outperforms the other existing models for aspect-based opinion analysis. The authors in the study of Xu et al. [17] designed an advanced word representation technique based on the weighted word-vectors and implemented a Bi-LSTM model, with a feedforward neural network to classify the sentiment from the comment data.

The work done by Alattar and Shaalan [18] presented a Filtered-LDA model to reveal sentiment variations in the Twitter dataset. The model adopts various hyperparameters to obtain reasons that cause sentiment variations. Fu et al. [19] have suggested an enhanced model that uses an LSTM model combination of the sentiment and word embedding to better represent the words followed by an attention vector. Jiang et al. [20] proposed a bag-of-words text representation method based on sentiment topic words composed of the deep neural network, sentiment topic words, and context information and performed well in Sentiment Analysis. The work of Pham et al. [21] suggested a combined approach of multiple Convolutional Neural Network (CNN), emphasizing word embeddings using different NLP mechanisms such as Word2Vec, GloVe, and the one-hot encoding. Han et al. [22] developed an advanced learning model based on joint operation CNN and LSTM for text representation. The work of Majumder et al. [23] exhibited the correlation between sarcasm recognition and sentiment analysis. The authors have introduced a multitasking classification model that improves both sarcasm and sentiment analysis tasks. The study of Rezaeinia et al. [24] presented an improved word embeddings technique based on Part-of-Speech tagging and sentiment lexicons. This method provides a better form of performance regarding sentiment analysis.

1.1 Problem Description

Based on the literature review, it has been analyzed that various research works have been done in the context of opinion classification. Existing works on data treatment have more focused on simple preprocessing operations such as tokenization, removal of punctuation, and stop words, which do not provide effective data modeling and are not enough to deal with the high feature space complexity in the training phase of the learning model. Also, an effective data treatment process is one of the primary steps that contribute to higher accuracy in the classification process. Most of the existing works did not emphasize much on effective data modeling prior to training rather; they just did the normal preprocessing operation. The current research work effectively handles the significant problems associated with data quality and classification accuracy. The significant contribution of the proposed work is highlighted as follows:

- Exploratory analysis is carried out to understand the characteristics of data and the need for preprocessing operation
- Suitable data treatment operation is carried out by performing some customized cleaning and data filtering operation based on the requirement of preprocessing
- Data modeling and preparation are done by performing manual data truncation and padding process to maintain the uniform length of the text sentences belonging to the training dataset.
- Implementation of Bi-LSTM learning model followed by suitable training parameters

1.2 Proposed Solution

The proposed study aims to present an enhancement in opinion classification from the natural language data (text) using customized preprocessing operation and a suitable deep learning approach. In order to perform opinion analysis, the study considers text data generated on the social media platform, which consists of dynamic information (text, symbol, number, punctuation, etc.). However, the accuracy of any text data-based classification system highly depends on the quality of the data. Suppose the dataset is ambiguous and consists of complex and dynamic information. In that case, the machine learning or deep learning model may deliver misleading, biased outcomes and seriously harm decision-making processes. In this regard, the proposed study presents customized data preprocessing operation in order to provide a suitable treatment and cleaning process to the text dataset captured from social media. To date, various deep learning algorithms or models are present with their own advantages and limitations. However, selecting a suitable deep learning model becomes another significant problem in the context of human behavior (opinion) analysis. Therefore, the current study considers opinion classification as a sequence classification problem since the current study deals with text sentences and attempts to implement a class of recurrent neural networks.

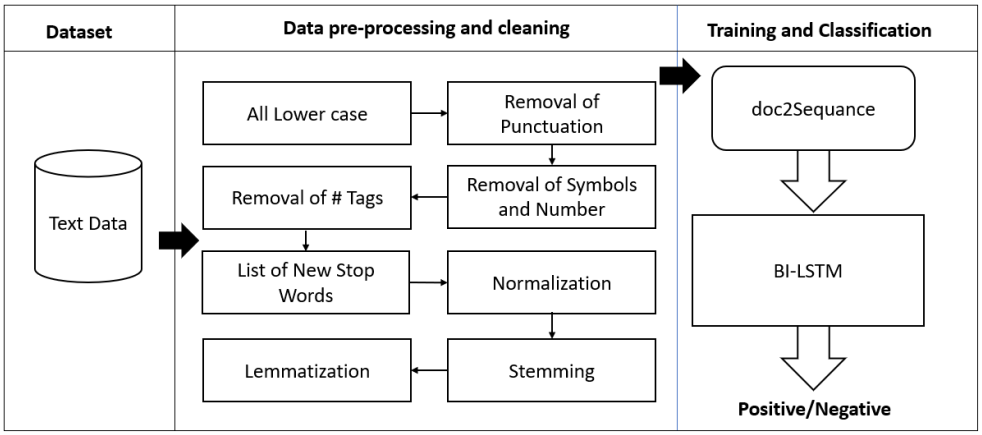

Fig 1 schematic architecture of the proposed system

The schematic architecture of the proposed system is shown in figure 1. The design and development of the proposed model are carried out so that it could better and more accurately seek to identify people's viewpoints towards entities such as events, topics, products, services, organizations, individuals, and issues.

The remaining part of this paper is described as follows: Section 2: discusses a proposed methodology and implementation procedure adopted in system design and development. This section highlights the dataset, its cleaning, treatment, and suitable modeling and development of the deep learning model for opinion classification. Section 3: presents the outcome and performance assessment of the proposed system, and finally, section 4 concludes the entire work discussed in this paper.

2. PROPOSED METHOD

This section presents an exploratory analysis of the dataset and strategy adopted in preprocessing the text data. Also, an implementation procedure is discussed for the deep learning-based opinion mining process followed by the word embedding process and model training.

2.1 Data Visualization and Analysis

The proposed study makes use of a social media dataset for opinion mining. Table 1 highlights the few samples of data in order to understand the characteristics and complexity of the dataset considered in the current work from the natural language processing viewpoint.

Table 1 Visualization of Text Data Samples

SI. No	**Text**	**Label**
1	friday hung out with kelsie and we went and saw The Da Vinci Code SUCKED!!!!!	0
2	Harry Potter is AWESOME I don't care if anyone says differently!..	1
3	<---Sad level is 3. I was writing a massive blog tweet on Myspace and my comp shut down. Now it's all lost *lays in fetal position*	0
4	BoRinG): whats wrong with him?? Please tell me..... :-/	0
5	I want to be here because I love Harry Potter, and I really want a place where people take it serious, but it is still so much fun.	1

Specifically, the Twitter dataset is considered, which is downloaded from the Kaggle. Twitter data are the unique form of text data that reflects information, opinion, or attitude that the users share publicly. The rationale behind considering tweets for the analysis is that the tweet texts are associated with dynamic representation and are semi-unstructured because they contain different forms of text representation, as shown in table 1. The user shares their thoughts in their own way in native format, especially with the different styles, containing the numbers, digits, punctuation, subjective context where some texts are small, and some texts are capital in between the sentences. The dataset considered in this study is labeled where each text belongs to two different opinion contexts, i.e., 0 and 1, where zero means negative opinion and 1 is subjected to the positive opinion. In order to make the dataset more friendly, the labels are updated from numerical to categorical labels as shown in table 2.

Table 2 Visualization of updated Label

SI. No	Existing Lable	Updated Lable
1	0	Negative
2	1	Positive
3	0	Negative
4	0	Negative
5	1	Positive

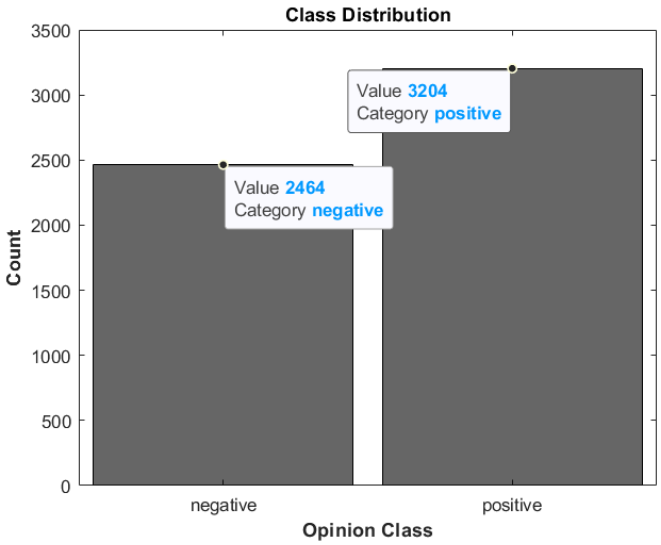

Fig 2 Distribution of the Opinion class

The distribution of the opinion class is shown in figure 2, where 2464 texts are subjected to the negative opinion class, and 3204 texts are subjected to the positive opinion class. Based on the analysis, it is identified that the dataset is a little imbalanced with a difference of 740, which may lead the learning model to be biased towards positive opinions in the classification process. In order to deal with this imbalance factor, the proposed study focuses more on the treatment of the unstructured and rich natural language (text). Therefore, the proposed study does not perform any sampling or scaling (up and downscaling) operation over the text data. Instead, it executes a preprocessing operation from the viewpoint of feature engineering, which will provide a precise representation of the input text data and enables better generalization in the training phase of the learning model.

In the current analysis, approximately 5000 texts were considered, split into training, validation, and testing set. Initially, the dataset is split into an 80:20 ratio, where 80% is considered for the training set, and the remaining 20% of the dataset is considered for the testing set. Again, the training set is split into an 80:20 ratio where 80% is kept for model training, and 20% is considered for the validation. Figure 3 demonstrates the strategy adopted in the dataset partitioned to prevent the learning model from overfitting and accurately validate the trained model's effectiveness.

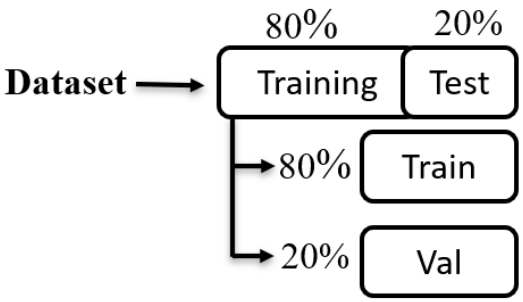

Fig.3 partition of the dataset

2.2 Preprocessing

The discussion in the previous section was all about data visualization and preliminary analysis, which shows that the text dataset consists of dynamic contents, semi-structured sentences, punctuation, numbers, short-words/text, and stop words. Therefore, an effective preprocessing operation is required to provide accurate treatment and cleaning operation to correct the text data in a suitable form. In this regard, the preprocessing operation is executed based on the NLP mechanism, and some customized data modeling is carried out to achieve better precision in the data treatment process. The entire data modeling and treatment process is executed out in the following manner discussed as follows:

***i) Removal of Unconnected Term*:** In this step of execution, the algorithm considers the elimination of punctuations, URLs, tickers, hashtags (#), number, digits, special characters, extra-wide spaces, and removal of short sentences whose length is limited to only two words. Also, all the text phrases are transformed to lower case, and emoticons are changed to the relevant words.

ii) Tokenization: Further tokenization of the text data is carried, which is an important step of the NLP. In this process, the text sentences are split into multiple and smaller elements called tokens. These tokens are strings with known meanings that help in understanding the context.

iii) Modelling of Stopwords*:* In this process, the proposed study performs modeling of stop words that could carry important meaning. However, the stopwords are those with less entropy value that does not describe the contextual meaning; therefore, they can be discarded from the text sentences. But in the proposed preprocessing operation, a customized operation is carried out, where a list of the stop words are edited to choose the stop words (such as are, aren't, aren't, did, didn't, and many more) that could carry important meaning for the opinion mining or classification.

iv) Stemming/Lemmatization: Stemming and lemmatization are important steps of text normalization after executing the above data treatment procedures. The stemming operation over text provides a stem representation of the text. Similarly, the lemmatization is also performed to get a base form of the text according to the POS and grammar protocol. The computing procedure for Twitter dataset preprocessing is discussed in the Pseudo constructs as follows:

Algorithm:1 Data Preprocessing

Input: Text dataset (T)

Output: Preprocessed Texts (PT)

Start:
Init DF

1. Load: DF → $f1$(filename)
2. DF. Lable → $f2$(Label rename: {′0′, ′1′}{′negative′, ′positive′})
3. Split DF: [Training, Test] ← $f3$(DF,0.2)
4. Split Training: [Train, Val] ← $f3$(Training,0.2)
5. **def function**: dataclean(text)
6. Text_lower = re. findall(′DF. text′, ′(. |||[a − z]|||[A − Z]))) do
7. lower(DF. text)
8. Text= $f4$(′DF. text′, @\w+′,′ #′, RT[\s] +,′ https? :\\S+′)
9. Tok= $f5$(DF. Text)
10. new_ stopwords = stopwords
11. Init N // list of not stop words
12. new_ stopwords(ismember (new_ stopwords, N)) = [ ];
13. Tok= $f6$(DF. Text, new_ stopwords)
14. DF. Text= $f7$(Tok, Stem)
15. DF. Text= $f7$(Tok, Lem)
16. return = PT
17. **For each** text **from** Train **do**
18. DF_p = dataclean(train. text)

End

A rich natural language (text) requires an effective data treatment operation to perform precise classification or analysis of the opinion. The above-mentioned algorithmic steps exhibit a vital operation of correcting the raw text data for further analysis. The algorithm takes an input of raw text data (T), and after undergoing several stages of preprocessing operation, it provides precise and cleaned data (PT). Initially, an empty vector is initialized as DF (data frame) that stores text data in a structured format using function $f1$(x) (line-1). Further, in the next step, the label field of the dataset gets updated in a more friendly manner using function $f2$(x), where numerical labels are replaced with the categorial name such that 0 is replaced with negative label and 1 is replaced with a positive label (line2). Afterward, a traintest split function $f3$(x) is used to perform splitting of the dataset to extract the training set and testing set in the ratio of 80% and 20%, respectively (line3). The training set is further split into two other sub-set of training and validation in the ratio 80/20 using the same function $f3$(x). The study further builds a function, namely 'clean data,' to carry out a preprocessing operation over the raw input text data. The first operation executed in this function is lowering text data to the all-lowercase (line6-7). Further, elimination of unrelated data such as #tags, tickers, short sentences, URLs, emojis, punctuation, digits, and missing text is carried out using function $f4$(x) to make input text data free from any form of ambiguity and impreciseness. In the next step, tokenization of the input text is carried out using NLP function $f5$(x), which provides a set of chunks of the input data. Further, the stopwords are removed to make data free from the words that do not have significant meaning in the phrase.

However, in this process, the proposed study performs a custom operation to filter out some important words from the list of stopwords. Initially, a vector, namely new_stopwords, are initialized that holds a list of all stopword from the NLP library, and another empty vector N gets initialized that holds a list of not stop word (line10-11). Afterward, a condition gets initialized that checks the existence of not stop word in the list of stop words and stores it in a new variable new stop word (line12). In the next step, a function *f*6(x) is used, which basically takes an input argument of text and a vector of new stop words to remove stop words from the input text (line13). In the next step, the algorithm executes the text data normalization process using function *f*7(x) that applied the stemming and lemmatization method to make words or text in its root format. After executing all the above-mentioned operations, the algorithm finally returns a preprocessed text (PT). In order to preprocess the training set, the algorithm calls the user-defined function 'data clean, which provides a cleaned text to be further analysed for opinion classification using a deep learning model. Figure 4 demonstrates word cloud visualization for the preprocessed training dataset.

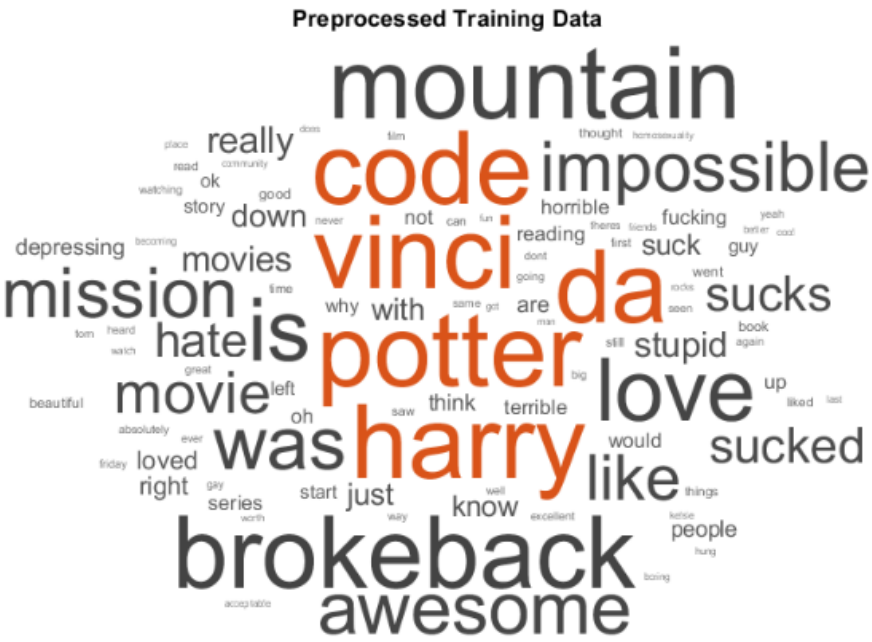

Fig.4 wordcloud visualization of the preprocessd training set

Figure 4 shows the frequency distribution of the words using word cloud that exhibits the text which is more in text data. It can be analyzed that peoples share their opinion regarding a movie. The bold highlighted text is more frequent in the text that reveals the words are mostly related to personal thought like the movie was awesome, harry potter, horrible, etc. Table 3 Summarizes the data treatment operations carried out by the preprocessing algorithm.

Table 3 Highlights of preprocessing algorithm

Data Arguments	**Treatment Feat**
Punctuations !,?,.,”	Eliminated
ULR’s	Changed to ′URL′
#Tags, tickers, special character	Eliminated
Emoticons	Transformed to relevant phrase
Upper case [**A** − **Z**]	Transformed to lower case [a − z]
Stop word	Customized and removed from the input text data
Tokenization	Input data into set of chunks
Normalization	Stemming and Lemmatization

2.3 Opinion classification using deep learning model

The classification of a public opinion involves several procedures viz. i) word encoding, ii) data padding and truncating, iii) constructing and training Bi-LSTM, and iv) validation of trained model via introducing new text data from the testing set. This section discusses the modeling of the learning model to perform opinion classification followed by an algorithmic approach.

The text data is inherently a combination of sequences of words, which have dependency means they are interconnected to each other via directed links that reveal the association owned by the connected words. Therefore, the proposed study implements a specific class of deep learning techniques: Long-short-term memory (LSTM), which is most efficient for learning long-term dependencies between sequences of text sentences. Also, the classification of opinion from the sequential data corpus (text dataset) can be treated as a sequence classification problem. The LSTM suits better than the other deep learning model and shallow machine learning techniques. LSTM is an improved class of the recurrent neural networks (RNN), designed to deal with sequential data by distributing their weights across the sequence. LSTM addresses the problem of gradient vanishing by employing its gates mechanisms and captures long-term relationships. The LSTM can be numerically defined as follows:

$$\mathrm{h_t} = \mathrm{f}(\mathrm{W_h} \cdot \mathrm{x_t} + \mathrm{U_t} \cdot \mathrm{h_{t-1}} + \mathrm{b_h}) \ldots (1)$$

Where $\mathrm{x_t}$ denotes current input sequence data, $\mathrm{h_t}$ denotes a hidden state of the neural network, $\mathrm{W_h}$ and $\mathrm{U_t}$ refers to the weights, and $\mathrm{b_h}$ denotes the bias. The f(x) is a non-linear function (i.e., tangent function) to learn and classify operations. Though LSTM has many advantages, it is also subjected to a limitation that it suffers in considering post word information as the sequences of the text data is read in a single feedforward direction. Therefore, the proposed study implements a Bi-LSTM learning model whose outputs are stacked together, one for forward and one for backward. The hidden states of the feedforward LSTM unit (h_t^F) and hidden states of backward LSTM unit (h_t^B) are concatenated to form a single hidden layer of Bi-LSTM (h_t^{Bi}) numerically expressed as follows:

$$h_t^F \oplus h_t^B \rightarrow h_t^{Bi} \ldots (2)$$

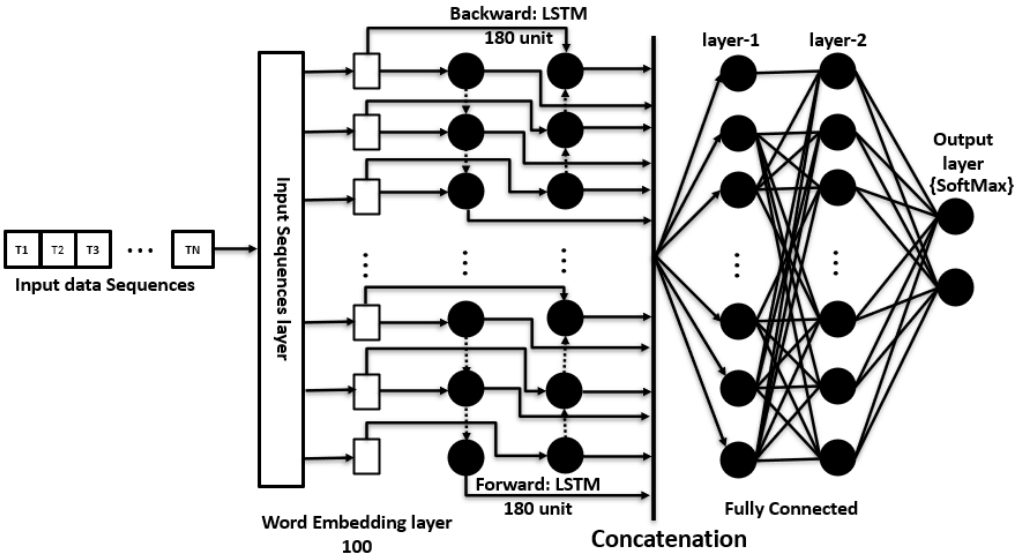

Fig.4 proposed deep learning model Bi- LSTM for opinion classification

The above figure 4 shows the architecture of the proposed deep learning Bi-LSTM model for the opinion classification from the natural language (text). In order to train the learning model, the sequence of input preprocessed text data needs to be converted into numeric sequences. In this regard, the study uses a word encoding mechanism that maps the training dataset into integer sequences. The encoding technique adopted in the proposed study is one-hot encoding vectorization. In addition, padding and truncation are further carried out to make text data of the same length. However, there is an option in the training process to pad and shorten input sequences automatically. But this option does not much effectively applicable for word vectors sequences. Therefore, the study performs this process manually by determining the length of text sentences, and then the text sequences that are longer than identified target value are truncated, and the sequences shorter than the target value are left-padded. A histogram plot in figure 5 shows the length of the text sentences that belongs to the training dataset.

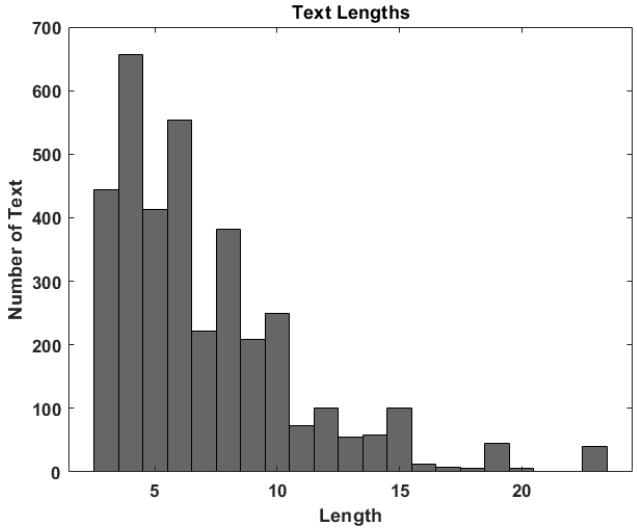

Fig.5 visualization of the text length

Based on the analysis from figure 5, it is analyzed that most of the text sentences have fewer than 23 chunks. Therefore, this length will be considered as a target length to truncate and pad the training dataset. The study also considers embedding layers in the modeling of the Bi-LSTM learning model. Including a word embedding layer will offer a mapping of a word in the lexicon to numerical vectors instead of a scalar. Also, it obtains semantic information of text phrase, which eventually maps the text phrase or word with similar meanings have similar vectors. After adding the word embedding layer, a Bi-LSTM layer is then considered and added with 180 hidden units. Finally, two fully connected layers and one output layer with softmax function are added for a sequence-to-label classification. The configuration details of the model development and training options are highlighted in Tables 4 and 5, respectively.

Algorithm:2 Opinion Classification

Input: Preprocessed Texts (PT)
Output: Opinion: Positive(P) or Negative (N)
Start:

1. Load: DF → $f1$(filename)
2. Apply Algorithm 1: Pre-processing training, and validation dataset
3. train ← dataclean(train.text)
4. val_x ← dataclean(train.text)
5. Prepare data for model training
6. Execute word encoding
7. enc← $f7$(train_x)
8. do truncating and padding
9. Compute: target_length ← f_{max} (length(train_x)
10. train_x ← $f8(enc$, train_x, target_length)
11. val_x ← $f8(enc$, val_x, target_length)
12. Execute **Model Development**
13. Init, I (input layer size), D ( ),O ( ),N ( ), C( )
14. Config layers
15. Inputlayer (I)
16. Embedding layer (D,N)
17. Bi-LSTM (O, outputmode, ‘last’)
18. Fully connected (C)
19. Outputlayer (activation function, softmax)

End

Table 4 configuration details of the model development

6 x 1 Layers of the proposed learning models		
1	Sequence Input	Sequence input with 1 dimension
2	Word Embedding Layer	Word embedding layer with 100 dimensions
3	Bi-LSTM	180 hidden cells
4	Fully Connected	2
5	Classification Output	Softmax
6	Loss function	crossentropyex

Table 5 details of the model training option

Training details		
1	Optimizer	Adam
2	MiniBatchSize	32
3	InitialLearnRate	0.01
4	GradientThreshold	1
5	MaxEpochs	100
6	L2Regularization	0.0006

3. RESULTS AND DISCUSSION

This section discusses the outcome obtained and performance analysis of the proposed learning model for opinion classification. The analysis of proposed system performance is evaluated concerning the accuracy, precision, recall rate, and F1-score briefly described as follows:

3.1 Performance Indicator

$\mathbf{Accuracy} = \frac{\mathrm{TP+TN}}{\mathrm{TP+FP+FN+TN}}$, $\mathbf{recision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$, $\mathbf{Recall} = \frac{TP}{TP+FN}$ and $\mathbf{F1\_Score} = 2 \times \left(\frac{\mathrm{Precision \times Recall}}{\mathrm{Precision+Recall}}\right)$

TP → True Positive (correctly classified positive value, i.e., actual and classified class = yes)
TN → True Negative (correctly classified negative value i.e., actual and classified class = no)
FP → False Positive (Actual class = no and classified class = yes)
FN → False Negative (Actual class = yes and classified class = no)

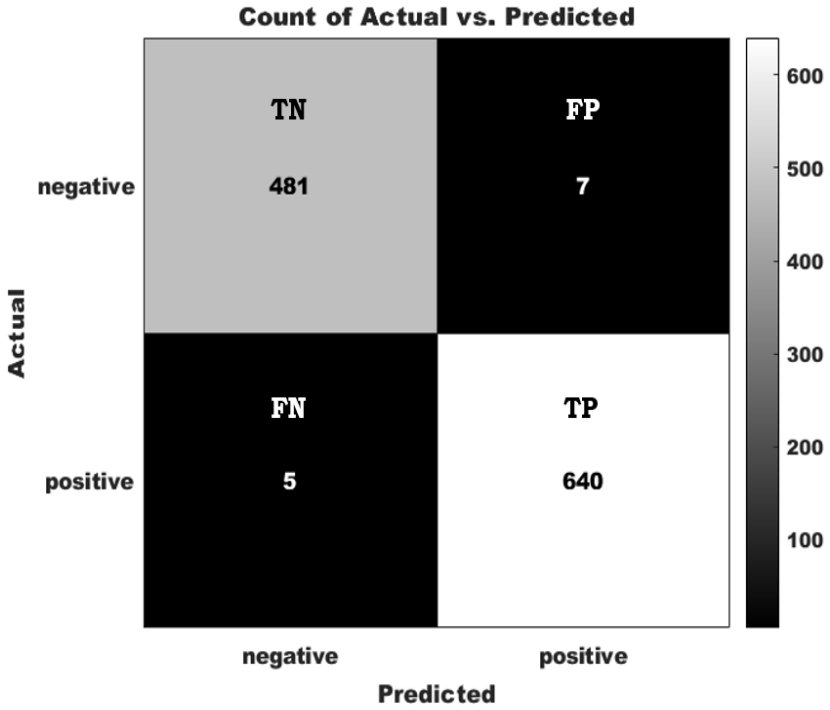

Fig.5 heatmap of the confusion matrix

The above figure 5 shows a heatmap of the confusion matrix, which provides an assessment of the output of the trained Bi-LSTM model. A closer analysis of the heatmap shows that out of 1133 text data, 640 text data belong to true positive, which means that 640 were classified as positive opinion where there is actual total 645 text that reflects positive opinion. Also, 481 texts were classified as true negative, where there is actual total 489 text that reflects negative opinion. In order to better understand the performance of the proposed system, the study performs comparative analysis with other machine learning models concerning multiple performance parameters such as accuracy, precision, recall rate, and F1_score. Table 6 highlights the quantified outcome of the proposed Bi-LSTM based model and other machine learning models.

Table 6 quantified outcome for comparative analysis

Model	Accuracy	Precision	Recall	F1_Score
SVM	85.018%	81.815%	78.961%	83.931%
Naïve Bayes	72.235%	79.81%	77.481%	80.063%
LSTM	94.975%	90.11%	94.36%	91.321%
Bi-LSTM	98.940%	98.918%	99.224%	99.071%

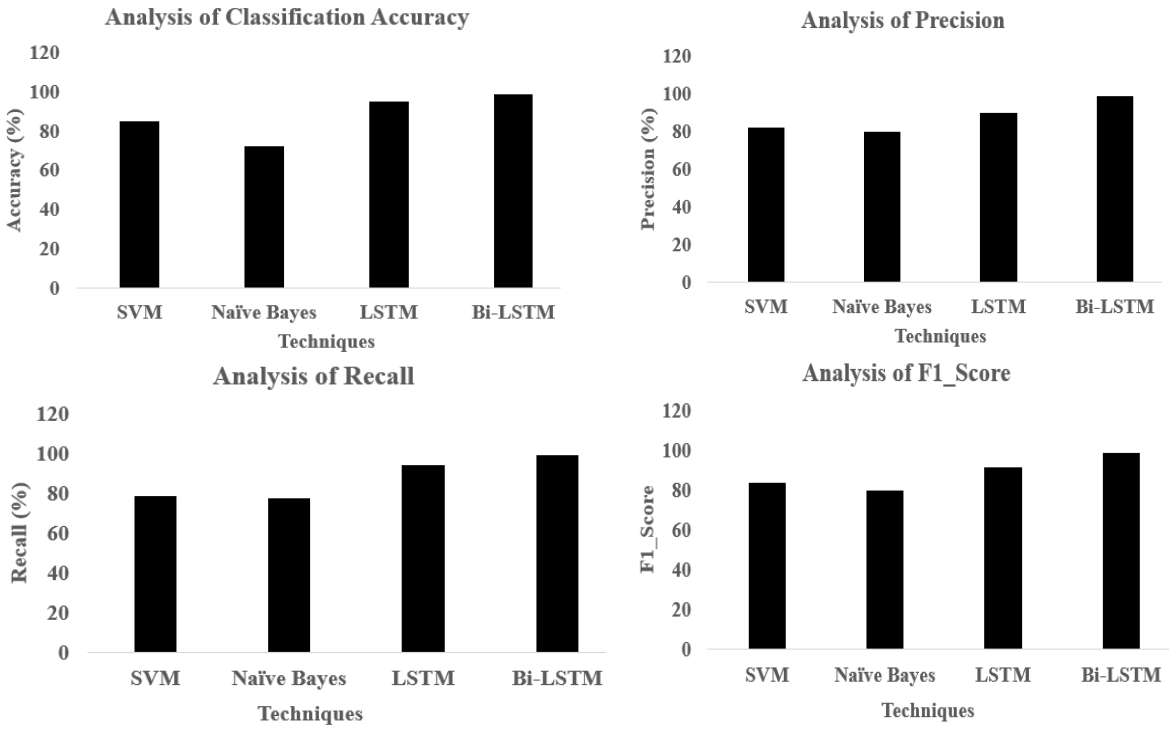

Fig. 6 Comparative analysis

The above figure shows a comparative analysis in order to validate the effectiveness and scope of the proposed work. The comparative analysis is carried with respect to the proposed Bi-LSTM, LSTM, support vector machine (SVM), and probabilistic model, i.e., Naïve Bayes. It can be seen that the proposed Bi-LSTM model outperforms LSTM and shallow machine learning models. The SVM requires ample training time that makes it inefficient and computationally expensive. Also, the SVM is not much scalable to have high feature space in the training process. For the naïve Bayes, it depends on an often-faulty consideration of equally important and independent features, leading to biases in the prediction.

In addition, the proposed system based on Bi-LSTM is enriched with proposed preprocessing operations that overcome the lexical sparsity and ambiguity issue in the training dataset and enable the better generalization of data in the training phase. The significance of the proposed work is the effective data treatment which improves the quality of data and adequate modeling of deep learning technique with a selection of suitable training parameters for the precise classification, which considers the contextual information by dealing with both forward and backward dependencies.

4. CONCLUSION

The proposed work has addressed the opinion classification problem for rich natural language. The proposed study exploits the advantage of utilizing a deep learning technique and effective data treatment on the performance of the opinion analysis from the rich and dynamic text data. A Bi-LSTM learning model is used that has the ability to capture contextual information to generalize textual features in a better way.

The feature space complexity is reduced significantly prior to the training process, which is carried out in the preprocessing and after preprocessing, where the padding and the truncating process is performed manually. The results show the scope and effectiveness of Bi-LSTM in the context of sequence classification problems. The proposed system achieved good performance in terms of accuracy, precision, recall rate, and F1-measure results over the existing learning models. In future work, the proposed study may extend towards processing different languages emphasizing different lexical approaches.

REFERENCES

[1] Khan M., Malviya A., Yadav S.K. (2021) Big Data and Social Media Analytics- A Challenging Approach in Processing of Big Data. In: Kumar A., Mozar S. (eds) ICCCE 2020. Lecture Notes in Electrical Engineering, vol 698. Springer, Singapore. https://doi.org/10.1007/978-981-15-7961-5_59

[2] Cheok A.D., Edwards B., Muniru I. (2017) Human Behavior and Social Networks. In: Alhajj R., Rokne J. (eds) Encyclopedia of Social Network Analysis and Mining. Springer, New York, NY. https://doi.org/10.1007/978-1-4614-7163-9_235-1

[3] Calabrese, Barbara & Cannataro, Mario & Ielpo, Nicola. (2015). Using Social Networks Data for Behavior and Sentiment Analysis. 285-293. 10.1007/978-3-319-23237-9_25.

[4] Le B., Nguyen H. (2015) Twitter Sentiment Analysis Using Machine Learning Techniques. In: Le Thi H., Nguyen N., Do T. (eds) Advanced Computational Methods for Knowledge Engineering. Advances in Intelligent Systems and Computing, vol 358. Springer, Cham. https://doi.org/10.1007/978-3-319-17996-4_25

[5] Sharma A, Ghose U. Sentimental analysis of twitter data with respect to general elections in india. Procedia Computer Science. 2020 Jan 1;173:325-34.

[6] M. Kanakaraj and R. M. R. Guddeti, "NLP based sentiment analysis on Twitter data using ensemble classifiers," 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), 2015, pp. 1-5, doi: 10.1109/ICSCN.2015.7219856.

[7] Mutinda, James & Mwangi, Waweru & Okeyo, George. (2021). Lexicon-pointed hybrid N-gram Features Extraction Model (LeNFEM) for sentence level sentiment analysis. Engineering Reports. 3. 10.1002/eng2.12374.

[8] Shobana J, Murali M. Improving feature engineering by fine tuning the parameters of Skip gram model. Materials Today: Proceedings. 2021 Feb 27.

[9] Lovera FA, Cardinale YC, Homsi MN. Sentiment Analysis in Twitter Based on Knowledge Graph and Deep Learning Classification. Electronics. 2021 Jan;10(22):2739.

[10] M. F. A. Bashri and R. Kusumaningrum, "Sentiment analysis using Latent Dirichlet Allocation and topic polarity wordcloud visualization," 2017 5th International Conference on Information and Communication Technology (ICoIC7), 2017, pp. 1-5, doi: 10.1109/ICoICT.2017.8074651.

[11] Chen, Q., Sokolova, M. Specialists, Scientists, and Sentiments: Word2Vec and Doc2Vec in Analysis of Scientific and Medical Texts. SN COMPUT. SCI. 2, 414 (2021). https://doi.org/10.1007/s42979-021-00807-1

[12] Parikh Y, Palusa A, Kasthuri S, Mehta R, Rana D. Efficient word2vec vectors for sentiment analysis to improve commercial movie success. InAdvanced Computational and Communication Paradigms 2018 (pp. 269-279). Springer, Singapore.
[13] Kaur, H., Ahsaan, S.U., Alankar, B. et al. A Proposed Sentiment Analysis Deep Learning Algorithm for Analyzing COVID-19 Tweets. Inf Syst Front (2021). https://doi.org/10.1007/s10796-021-10135-7
[14] Londhe A., Rao P.V.R.D.P. (2021) Aspect Based Sentiment Analysis – An Incremental Model Learning Approach Using LSTM-RNN. In: Singh M., Tyagi V., Gupta P.K., Flusser J., Ören T., Sonawane V.R. (eds) Advances in Computing and Data Sciences. ICACDS 2021. Communications in Computer and Information Science, vol 1440. Springer, Cham. https://doi.org/10.1007/978-3-030-81462-5_59
[15]Pergola G, Gui L, He Y (2019) TDAM: a topic-dependent attention model for sentiment analysis. Inf Process Manag 56(6):102084
[16]Ma Y, Peng H, Cambria E (2018) Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In; Proceedings of the AAAI conference on artificial ıntelligence vol 32. pp 5876–5883
[17]Xu G, Meng Y, Qiu X, Yu Z, Wu X (2019) Sentiment analysis of comment texts based on BiLSTM. IEEE Access 7:51522–51532
[18] F. Alattar and K. Shaalan, "Using Artificial Intelligence to Understand What Causes Sentiment Changes on Social Media," in IEEE Access, vol. 9, pp. 61756-61767, 2021, doi: 10.1109/ACCESS.2021.3073657.
[19] X. Fu, J. Yang, J. Li, M. Fang and H. Wang, "Lexicon-Enhanced LSTM With Attention for General Sentiment Analysis," in IEEE Access, vol. 6, pp. 71884-71891, 2018, doi: 10.1109/ACCESS.2018.2878425.
[20] Z. Jiang, S. Gao, and L. Chen, ‘‘Study on text representation method based on deep learning and topic information,’’ Computing, vol. 102, no. 3, pp. 623–642, Sep. 2019.
[21] D.-H. Pham and A.-C. Le, ‘‘Exploiting multiple word embeddings and onehot character vectors for aspect-based sentiment analysis,’’ Int. J. Approx. Reasoning, vol. 103, pp. 1–10, Dec. 2018.
[22] H. Han, X. Bai, and P. Li, ‘‘Augmented sentiment representation by learning context information,’’ Neural Comput. Appl., vol. 31, no. 12, pp. 8475–8482, Dec. 2019.
[23] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, A. Gelbukh, and E. Cambria, ‘‘Sentiment and sarcasm classification with multitask learning,’’ IEEE Intell. Syst., vol. 34, no. 3, pp. 38–43, May 2019
[24] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi, ‘‘Sentiment analysis based on improved pre-trained word embeddings,’’ Expert Syst. Appl., vol. 117, pp. 139–147, Mar. 2019.