ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal
Regular Issue, Vol. 11 N. 3 (2022), 309-329
eISSN: 2255-2863
DOI: https://doi.org/10.14201/adcaij.27902

An Optimized Deep ConvNet Sentiment Classification Model with Word Embedding and BiLSTM Technique

Roop Ranjan[a] and AK Daniel[b]

[a, b] Madan Mohan Malaviya University of Technology, Gorakhpur, Uttar Pradesh, India, 273001

✉ roop.ranjan@gmail.com, danielak@rediffmail.com

KEYWORDS

sentiment classification; deep learning; long short term memory networks; emotion analysis; FastText

ABSTRACT

Sentiment Classification is a key area of research in natural language processing and it is frequently applied across a range of industries. The goal of Sentiment Analysis is to figure out if a product or service received a negative or positive response. Sentiment Analysis is widely utilized in several commercial fields to enhance the quality of services (QoS) for goods or services by gaining a better knowledge of consumer feedback. Deep learning provides cutting-edge achievements in a variety of complex fields. The goal of the study is to propose an improved approach to evaluating and categorising sentiments into different groups. This study proposes a novel hybridised model that combines the benefits of deep learning technologies Dual LSTM (Long Short-Term Memory) and CNN (Convolution Neural Network) with the word embedding technique. In addition, attention-based BiLSTM is used in a multi-convolutional approach. Standard measures were used to verify the validity of the proposed model's performance. The results show that the proposed model has an enhanced accuracy of 97.01%, which is significantly better than existing models.

1. Introduction

Sentiment Analysis is a set of linguistic operations which apply to digitized texts, such as publications and comments from social networks, as well as press articles, as part of the autonomous processing of natural language. Its goal is to determine the sentiment represented in a text and predict its polarity (positive or negative) toward a specific subject. Sentiment Analysis is a technique for the automatic extraction of sentiment data from unstructured texts. Several fields, including machine learning, natural language processing (NLP), and data mining, use Sentiment Analysis (Liu et al., 2012). Researchers are becoming increasingly interested in the different methods of analysing this massive amount of data for extraction of the views stated on a specific topic. Sentiment Analysis is used by corporations to learn how their customers feel about their products and services. People may gain feedback on products, services, events, and other topics thanks to the widespread use of social media platforms (Pang et al., 2008). People can express their opinions and share their reviews on social media such as Twitter, Instagram, blogs, and news sites. The capability to notice and respond to negative customer feedback is critical to a company's success. (Kim et al., 2017).

Statistical machine learning approaches are great for simple Sentiment Analysis, but they are not suitable for more complex text classification problems. (Neethu et al., 2013).

Word embedding is a technique for representing written words as a set of numeric values or vectors. It generates similar vector representations for words with similar meanings. The distributed representation-based word embedding systems Word2Vec (Lebret et al., 2013) and Glove (Pennington et al., 2014) are now the most widely utilised. Deep learning approaches are currently preferred over machine learning methods such as Support Vector Machines (SVM), Naive Bayes, Decision Trees, and Random Forests, which are often used for sentiment categorization, according to a recent study (Rehman et al., 2019).

RNN is a deep learning technique that is commonly used in natural language processing (NLP) for predicting the upcoming word on the basis of the previous words in the given sentence. Convolutional Neural Networks (CNN) are a type of artificial neural network that can accurately recognise information in a variety of situations (Simonyam et al., 2014). Due to the cost-effectiveness of large manufacturing of efficient Graphics Processing Unit cards and the large size of datasets, deep learning has emerged as a superior method to machine learning in sentiment classification applications (Chollet et al., 2017). Compatibility between text representation techniques and their algorithms is critical in many text-classification tasks. Therefore, it becomes feasible for providing high-performance classification with the assistance of a suitable representation of texts and a classification model.

With the advancements in cognitive computing and artificial intelligence, deep learning techniques have emerged as an increasingly popular method for Sentiment Analysis in several fields (Chen et al., 2018; Hwang et al., 2017). As a result, a growing number of researchers use neural networks to automatically build feature representations from text sequences. Currently, the prominent deep learning algorithms for sentiment categorization include RNN, CNN, and LSTM. Huang et al., 2015, proposed combining bidirectional LSTM (BiLSTM) and CRF models to get superior outcomes by using both backward input and forward input characteristics. Ma and Hovy, 2016, proposed a combined approach using CNNs and LSTM-CRF to identify entities.

Each deep learning approach has a distinctive property that is actively used to achieve a particular objective or assess datasets. The text representation technique, which translates texts into numeric form, has both benefits and disadvantages. To resolve the difficulties and obtain higher accuracy, a combined approach is critical. It would require ideal text depiction and an appropriate machine learning algorithm for emotion analysis. In the numeric format representation of user sentiments, any text representation approach (e.g., word embedding, character embedding) is not fully complete. To address these issues and to evaluate a dataset of tweets from various Indian airline services, a new hybrid model is proposed using deep learning techniques that employ multiple representations of data.

To achieve the objective, the main contribution of this research is to use the advantages of different deep learning techniques i.e., CNN and DualLSTM followed by another layer of CNN. A CNN layer is used for feature extraction, which is then passed to a bidirectional LSTM to obtain long-duration dependencies. Following these steps, the second layer of CNN is used to begin learning the retrieved properties from scratch. This process simplifies the learning of the features, and the machine gains a better knowledge of the classes. Word embedding was used in our research to represent words. Furthermore, word vectorization is combined with the orientation score characteristics as the sentiment characteristics set. The embedding parameters were merged and put into a CNN layer and a DualLSTM network layer. To test the validity and effectiveness, the model is implemented on five different sets on the basis of the variation in the length of the reviewed text. The proposed model has significantly improved results in comparison to previous models.

The major contribution of the research are as follows:

1. Word embedding techniques Word2Vec, Glove, and FasText have been utilised to render tweets as numerical vectors. The models are pre-trained word vectors which can capture word semantics and are trained on a huge collection of words. The research that has been carried out has focused on several models of vectors of words to test the performance of the model.
2. By combining the framework of CNN with DualLSTM, a multi-stage ConvNet (CNN) model-based technique for text sentiment classification is presented. The ConvNet model obtains localized features from word embedding, the Bi-LSTM model captures long-distance dependencies, and the outcome of the classification are the classified features.
3. The efficiency of the proposed model is compared with several deep learning methods and the outcome proves the effectiveness of the model.

The rest of the research paper is structured as follows: Section 2 provides an overview of state-of-the-art research, particularly relating to the application of Sentiment Analysis in relevant fields. The techniques that have been involved in this proposal are discussed in Section 3. Section 4 provides detailed information on the dataset and classifier models that were utilized in the experiments. The experimental findings are outlined in Section 5, which is followed by a discussion in Section 6. In the last section, the article is concluded and future lines of research are described.

2. Related Work

Sentiment Analysis has received increasing attention from researchers as a result of the growth of NLP, and various implementations of word-level embedding have been carried out. State-of-the-art research has discovered that hybrid algorithms produced improved classification results. Jiang *et al.*, 2016, created a text representation strategy using bag-of-words based on sentiment topical terms, which included a deep neural network, context information, and sentiment topic terms that performed significantly better in Sentiment Analysis. Whereas Rezaeinia et al., 2019, presented an enhanced word embeddings technique based on POS tagging methodology and lexicons of sentiment that outperformed pre-trained embeddings of words utilized in Sentiment Analysis. To improve the performance of sentiment classification, Liu et al., 2016, presented a model in which machine learning was coupled with deep learning. The efficiency of the presented method was shown using datasets with Turkish and Chinese languages in their research. Pham et al., 2018, presented an approach that combined several CNNs and was centered on embeddings of words from Glove, Word2Vec, and it performed well in aspect-level sentiment classification tasks. Han et al., 2019, developed a hybrid neural network approach for document representation that included user and product information and used CNNs and LSTM.

An innovative deep learning methodology with hybrid CNNs and Bi-LSTM features was demonstrated in Wint et al., 2018, that coupled the power of CNNs with Bi-LSTM. The authors achieved unique vectors of features provided as input to the LSTM layer using both separate pre-trained vectors of words. The BERT model for representing text, proposed by Devlin et al., 2019, may better depict the modifying associations in texts and performed well in Sentiment Analysis tests. By combining a neural topical approach into representations of word-level semantics, Liu et al., 2019 proposed latent topic information of the given text, as well as a novel topic-based attention method for texts to look at the syntactic of words using the topics given for word association. BiLSTMs were used in another research Guo et al., 2018, to extract dependency knowledge from vocabulary and the position of the sentence. In the proposed hybrid strategy, BiLSTM and CNN were used to produce n-gram features from text categorization by applying multiple CNNs to given LSTM outputs.

In another approach, Zhou and Long, 2018, performed the classification of texts from reviews of Chinese products using the Bi-LSTM approach, with the help of CNN for extracting features using word embedding. The combination of CNN and BiLSTM for classification in the experiments led to superior classification results in comparison to using CNN and BiLSTM separately. In another research (Sun et al., 2014), a multilayer CNN with LSTM was utilized for analyzing the sentiments of users on a given dataset with the help of the application of social media from Tibetan. The characteristics were retrieved using the assistance of a three-layered CNN. The collected features were sent to a two-layer LSTM network as input. This hybrid model, based on deep learning, outperformed CNN and LSTM according to the results. A hybrid Bi-LSTM model based on attention-mechanism is proposed in another research (Zheng et al., 2019). The proposed model efficiently integrated BiLSTM and CNN for the classification of text using Word2Vec with an attention mechanism.

In addition, Fan et al., 2017, utilized a single model of CNN and a dual-stacked LSTM to analyse Indian tweets on a sequential basis. The characteristics produced from the CNN layer are fed into the LSTM network in their hybrid experiments. Like earlier hybrid research, the researchers utilized CNN to extract characteristics. Zhou et al., 2016, proposed a dual language representational learning model using attention, in which the scattered semantics of several text documents in both target and source languages were learned. Documents are modelled with LSTM networks. They also proposed a hierarchy-based attention methodology for multilingual LSTM networks, which outperformed the benchmark data set.

Furthermore, the Convolutional Neural Network (Xu et al., 2019) method might efficiently extract local information. Also, the work has been done utilizing CNN to handle the issue of orientation analysis. AF-LSTM was proposed by Tay et al., 2018. To represent context and aspect terms at the word level, Atrial Fibrillation-LSTM (AF-LSTM) employs circular correlation and convolution. Through the information fusion procedure, the targeted knowledge is then integrated into the representation of sentences. Yu et al., 2019, utilized a multi-way Gated Recurrent Unit (GRU) combined with an attention-based mechanism to classify brief text in e-commerce reviews, with promising results. These approaches demonstrate that when deep learning is coupled with an attention mechanism short text Sentiment Analysis can improve.

In another research, Huang et al., 2018, presented an AOA model i.e., Attention-over-Attention (AOA) model. With the help of two fine-grained attention mechanisms, the AOA model extracts text-level association between aspects and contexts that allow the sentence representation to focus automatically on the elements that belong to sentences that are significantly more important for the aspect terms expressions. The TDLSTM model is proposed by Tang et al., 2016. In the proposed model the sentence was split into two sections with aspect. The hidden state of both sections was modeled using dual long short-term networks (LSTM). After that, the two portions were merged to create a specified target representation that was then put into the activation softmax algorithm for emotion categorization. Baziotis et al., 2017, utilized LSTM coupled with an attention-based model to assign weights to decisive words via the attention-based mechanism, which improved the effect of keywords of emotion sentences and obtained significantly better results in the categorization of emotions in Twitter reviews.

Although several researchers have proposed deep neural network architectures for sentiment classification problems based on the CNN or RNN, few researchers have comprehensively examined the performance of several classification models based on deep learning methods. Seo et al., 2020, proposed an analytical comparison of several sentiment classifications based on deep learning model architectures to extract useful significance for the development of the sentiment classification approach. For the identical model structure, input of word-level produced better results for classification than the input based on character-level. In another study, Hu et al., 2018, showed that models built using deep learning outperformed standard techniques, such as algorithms based on dictionary methods, the Naive Bayes, or SVM. They did not, however, disclose quantifiable performance metrics such as F-Measures or level of accuracy. Dzikienė et al., 2019, presented a performance comparison of conventional machine learning methodologies Naïve-Bayes Multinomial, Support Vector Machine and deep learning (LSTM and CNN) approaches on the Lithuanian internet comments dataset. Features based on morphological, character information, conventional machine learning approaches were applied. The deep learning approaches were applied on both Word2Vec and FastText embeddings. In their experiments, LSTM outperformed SVM and Naïve Byes multinomial methods. Yin et al., 2017, compared the results of LSTM, CNN, and GRU sentiment classification. However, the utility of their experimental outcomes was of less use since they could not focus on enough structure of differences in the model, and their conclusive result was based on the experimental results of only one dataset.

RNN was utilized by Socher et al., 2013, to tackle text categorization difficulties. A model called Sentiment Treebank was introduced that surpassed all earlier techniques in terms of various criteria when the training was performed on the fresh tree-bank, which could represent the impacts of negativity accurately. Yang et al., 2016, established a hierarchy-based text categorization strategy based on the attention-based mechanism that effectively captured the text's main sentiment information. Huang et al., 2015, suggested combining bidirectional LSTM (BiLSTM) and CRF models to produce superior outcomes by using both backward and forward input characteristics. Sentic-LSTM was developed by author to explicitly incorporate explicit and implicit information, and an extended version of Sentic-LSTM was presented to deal with a combined work involving aspect detection of target-dependent and aspect-based classification of polarity (Ma et al., 2018). The authors presented a refinement method of word vector that improved all word vectors. By improving pre-trained vectors of the word and using intensity ratings of sentiments provided by sentimental lexicons, the proposed model performed better in Sentiment Analysis (Gu et al., 2018). By training a huge corpus of text, Peters et al., 2018, proposed a textual representation strategy using a deep learning model, the researchers built a text depiction framework in the English language that incorporated grammar and sentiment elements.

Looking at the studies described in these papers, it is evident that a variety of methods have been implemented for the classification of sentiments using deep learning techniques. The approaches and research described above were primarily employed for extracting semantic information for features from the sentence dimension while ignoring the information-based features of the dimension of the word vector. In this research, two Convolutional layers (ConvNet) are used with an attention-based BiLSTM layer to extract the semantic information of the local characteristics of the word vector in the word insertion dimension. This research also uses max pooling to achieve significantly comprehensive local feature information. Section 3 discusses the proposed model in detail.

3. Proposed Method

The core idea behind the proposed mode is to use two distinct deep neural networks, namely ConvNet and DualLSTM. Then a hybrid ConvNet + DualLSTM + ConvNet model is used to detect users' view orientation towards the services represented in Comments in tweets as shown in Table 5. To reduce unnecessary information in sentences, it is required to create a base sentiment dictionary with negative, positive, and neutral texts that contain only relevant sentiment words that are included manually, as well as a set of rules based on a grammatical sentence that includes associated words and degree adverbs. Among Word2Vec, Keras and FastText; FastText embedding has shown the most optimized performance. Therefore FastText is chosen as the word embeddings method in the proposed research. Further, the hybrid model is proposed because ConvNet extracts local features from comments, DualLSTM captures contextual information from both directions as well as long-range dependencies, and the hybrid model combines the benefits of both complimentary ConvNet and DualLSTM architectures.

In this section, a detailed explanation of the proposed model is discussed. The proposed model consists of the following layers:

3.1 Pre-Processing Layer
3.2 Word Embedding Layer
3.3 ConvNet Layer(Multi)
3.4 Pooling Layer
3.5 DualLSTM Layer
3.6 Attention and Dense Layer
3.7 Output Layer

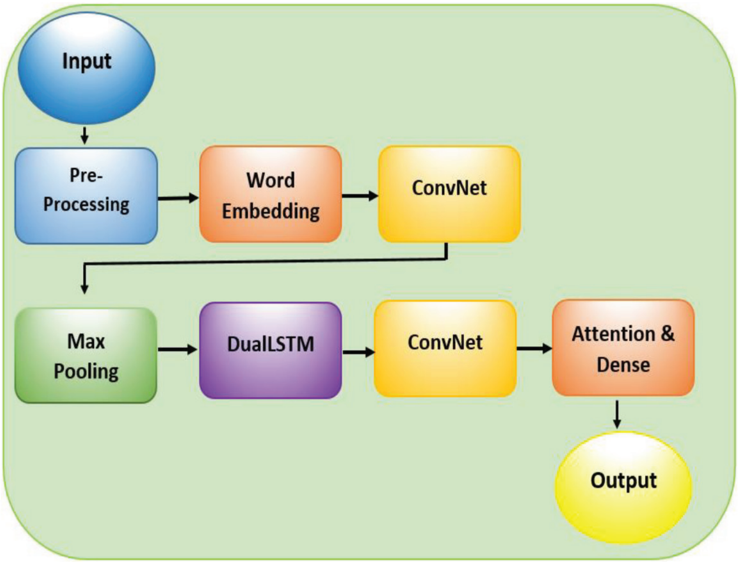


Figure 1. Proposed Model of Sentiment Classification

3.1. Pre-Processing Layer

Raw information usually contains words or symbols that computers cannot comprehend, therefore it is required to clean the data and restructure it in an understandable format, and therefore data cleaning is the important stage in NLP. The pre-processing of input texts in Figure 1 was implemented on the dataset to obtain excellent classification performance in categorizing text review data. Before the word-based representation phase (Word Embedding), these preprocessing steps are used to reduce unwanted contents and convert the dataset into a usable form. Initially, all texts in the dataset are converted to lower cases. Further, irregular spacing between words is trimmed to a single space. Punctuations, numerals, and unformatted characters in tweets were eliminated. Grammatical Error Correction (GEC) by NLP-Progress is used to correct all grammatical mistakes like spellings, grammar, punctuations.

3.2. Word Embedding Layer

One of the most important stages in the text classification process is the accurate representation of texts. In this study Word2Vec, FastText Embedding and Glove Embedding are used.

The steps for converting input text to Word Vector Representation are as follows:

- The method takes as input a text labelled as C with m tokens (words) with each token transferred to its corresponding word vector representation *s*. The text is reorganized in a word embedding concatenated sequence.
- $C = [s_1; s_2, s_3;::: :s_m]$, where s_i is the embedding of the word for the i^{th} word, which is projected to a given vector $s_i \in S^d$.
- A sentence $\in S^{dxm}$) matrix is created for each text input, where *d* indicates the embedding dimension and has been defined as the length of the sentence.
- The sentence's matrix is now transferred to the ConvNet layers, where it will be processed further.

3.3. ConvNet Layer

As illustrated in Figure 2, the architecture of a Convent comprises an input layer, an output layer, and five different hidden layers. The input layer accepts a textual message that has been padded to a predetermined length of words, followed by a word embedding layer. The attention layer follows the word embedding layer, to extract high-level feature vectors. The attention layer is a sub-unit made up of context vectors that line up the source input with the goal output. Figure 6a shows an illustration of the attention mechanism in the upper right corner. The SpatialDropout1D (for dropout) layer uses feature vectors as inputs derived from the attention layer. On top of the dropout layer, a ConvNet layer with convolution filters and a ReLU activation function is applied. The details of ConvNet model are shown in Figure 2.

- The ConvNet Module uses a convolutional based operation «*», between the matrix of text $C \in S^{dxm}$
- A filtering matrix $P \in S^{mxk}$, which provides an output as matrix *OP*, is termed as a features map.
- The features map is learned as per the following equation:

$$O_{ab} = (C * P) = k(D \circ t_{a:a+n-1,b+d-1} + y) \tag{1}$$

Where y represents the biasing vector, D represents the weighted matrix, and k is the convolutional operation's nonlinearity activation function. A nonlinearity function Rectified Linear Unit (ReLU) is applied here for speeding up the process of training and validation and provides better results.

3.4. Pooling Layer

The pooling layer receives the output of the convolutional layer. The convolutional layer minimizes the content representation even further by selecting the maximum value obtained from a pool of values and eliminating the irrelevant data. The procedure of pooling is represented as follows:

$$V_{a,b} = \max\left(O_{a+b-1,b+d-1}\right) \tag{2}$$

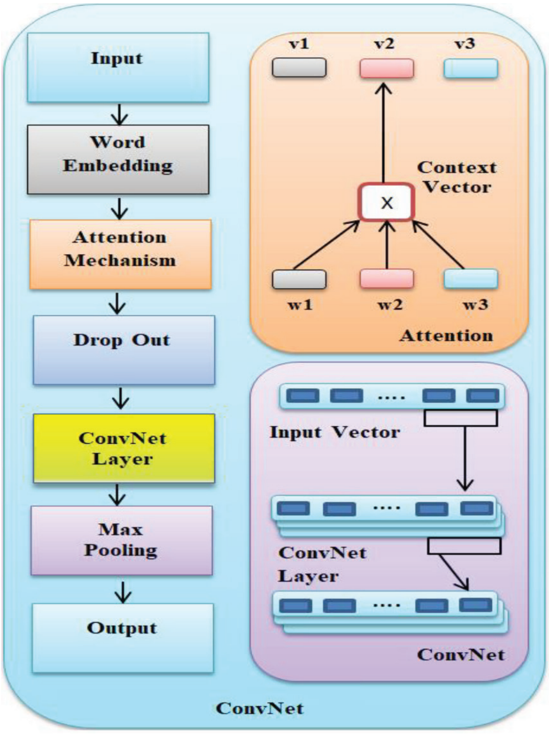


Figure 2. Architecture of ConvNet

3.5. DualLSTM Layer

To make precise predictions, the model must understand the long-duration dependence on text data. Since the convolutional layer lacks this capacity, DualLSTM has been used to incorporate this component into the proposed model. The model learns from the data in both directions i.e., left-to-right and right-to-left using DualLSTM. As a result, the DualLSTM layer enhances classification accuracy. In the Bidirectional LSTM, there are two autonomous LSTMs i.e., ahead LSTM and backward LSTM.

The hidden state «h» is computed by forwarding LSTM using the hidden previous state «h_{t-1}» and the input vector «z_t» whereas the backward LSTM uses the hidden future state «h_{t+1}» and the vector input «z_t» to compute the hidden state «g».

Finally, using the following Equation 3, both direction vectors(backward and ahead) have been merged as the last state(hidden) in the DualLSTM to create a series of output for vectors(hidden) $H= [h_1; h_2; h_3;......h_n]$.

$$\overleftrightarrow{h} = \overrightarrow{h} \oplus \overleftarrow{h} \tag{3}$$

The following equations are used to implement the DualLSTM cell.

Forward LSTM:

$$R = \begin{bmatrix} h_{t-1} \\ z_t \end{bmatrix}$$

$$\sigma f p_t = (W_{fp} \cdot R + y_{fp}) \tag{4}$$

$$\sigma i p_t = (W_{ip} \cdot R + y_{ip}) \tag{5}$$

$$\sigma o p_t = (W_{op} \cdot R + y_{op}) \tag{6}$$

$$c_{pt} = m \odot c_{pt\text{-}1} + ip_t \odot \tanh\,(W_{cp} \cdot R + y_{cp}) \tag{7}$$

$$h_t = op_t \odot \lambda(c_{pt}) \tag{8}$$

Backward LSTM:

$$X = \begin{bmatrix} h_{t+1} \\ z_t \end{bmatrix}$$

$$\sigma f p_t = (W_{fp} \cdot R + y_{fp}) \tag{9}$$

$$\sigma i p_t = (W_{ip} \cdot R + y_{ip}) \tag{10}$$

$$\sigma o p_t = (W_{op} \cdot R + y_{op}) \tag{11}$$

$$cp_t = k_{pt} \odot c_{pt\text{-}1} + ip_t \odot \tanh\,(W_{cp} \cdot R + y_{cp}) \tag{12}$$

$$h_t = op_t \odot (cp_t) \tag{13}$$

Where weight matrices are represented by W_{fp}, W_{ip}, W_{op}, and y_{fp}, y_{ip}, y_{op} as associated biases, which are the input gate parameters, forget gate parameters, and output gate parameters respectively.

sigmoid function for activation is represented by σ.
symbol-wise multiplication is denoted by $\odot$.
g_t is representing the vector in hidden state and the input vector is denoted by z_t.
λ is a tangent function.
The current state is denoted by cp_t, the previous state is shown by cp_{t-1} and future state is represented by cp_{t+1}

3.6. Attention & Dense Layer

There are certain words in a statement that are irrelevant for polarity detection but on the other hand, some words are decisive. The attention-based mechanism is used to draw attention to informative content. Therefore, this layer was created to automatically extract the significant terms.

Eq. 14 is used to calculate the word significance vector e_t. It uses the whole DualLSTM hidden states *h,* as input to the attention layer. *W* stands for weight, *y* for bias, and *tanh* for activation function.

$$e_t = \tanh\,(W_h h_t + y_n) \tag{14}$$

Finally, to create the output of the attention mechanism, Eq.15 is used to calculate a weighted summation.

$$cp_t = \sum_{t=1}^{m} n_t h_t \tag{15}$$

The attention layer output $cp = [cp_1, cp_2, cp_3, \ldots\ldots\; cp_m]$ is provided as input for the following layer.

Further conversion of the matrix of context retrieved from the preceding layer to a vector context that provides the input for the classification layer's final stage is performed. The following Eq. 16 is used to execute the flatten layer operation.

$$f = [cp_1 * cp_2 * cp_3 \ldots\ldots * cp_T] \tag{16}$$

3.7. Output Layer

The proposed approach for determining the class of sentiments in terms of negativity, positivity, or neutrality has reached its conclusion. The output of the flatten layer is provided to a softmax activation function, that calculates the likelihood of the sentiment classification. The final output is computed as:

$$O_j = \sum_{i=1}^{m} w_i * k_i + y \tag{17}$$

4. Data Acquisition and Experimental Setup

4.1. Data Acquisition

The proposed model has been tested on a dataset acquired from user tweets on Twitter regarding Indian Airlines between June 1st and August 31st, 2021. The dataset contains 24,235 tweets. Domestic flights began after the second wave of Covid-19, hence the data was obtained during that time period to test the response of the travellers to their services. Therefore, this research provides an insight into the service quality of Indian Airlines after Covid-19. A Rest API-based tool named Tweepy has been used. Three sentiment classes have been represented i.e., positive, negative, and neutral. The dataset has been divided into two distinct subsets i.e., Training Set and Validation Set. 75% of the tweets (18176 tweets) are grouped as the Training Set and 25% (6059) of the tweets are grouped into the Validation Set.

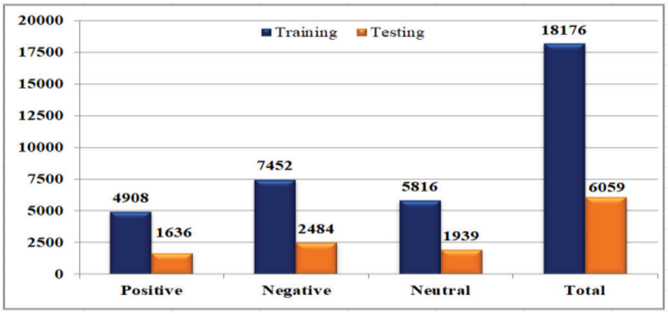


Figure 3. Classified number of Tweets in the dataset

Figure 3 shows the graphical representation of review classification in different categories

Table 1. Categorization Details of the Tweet Dataset

Dataset Size	Positive	Negative	Neutral
24235	6544	9936	7755

Table 2. Categorization of the Training Set

Dataset Size	Positive	Negative	Neutral
18176	4908	7452	5816

Table 3. Categorization of the Validation Set

Dataset Size	Positive	Negative	Neutral
6059	1636	2484	1939

The total number of tweets in the dataset, tweets in the training set, and tweets in the validation set are provided in Tables 1, Table 2, and Table 3, respectively.

4.2. Experimental Setup

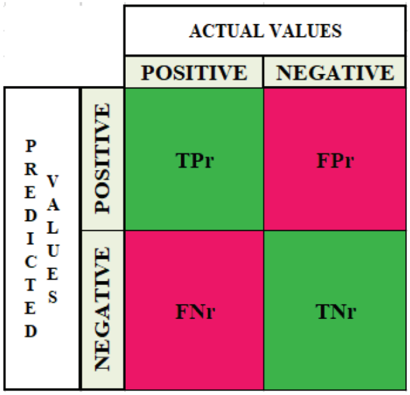
Experiments have been conducted utilising Google's services. Google offers a cloud-based file storage service called Google-Drive, which we used to store our data. The Google Colab system was used for this research, which is a free cloud-based tool provided by Google for Machine Learning developers that is built on Jupyter notebook for undertaking machine learning research using Python. Experiments with the proposed model have been carried out using the Keras API with Tenserflow as a backend.

4.3. Parameter Settings

Hyper-parameter optimization must be used to get high model performance. Underfitting and overfitting are also avoided using hyper-parameter settings. To improve accuracy, the randomised search strategy was applied. The hyper parameters were identified as Size of Kernel to be 5, Embedding Dimensions were 300(Keras), Size of Filter was128. The activation function used here was ReLu. Batch size for each epoch was 128 with a total number of 50 epochs. Learning rate was set at 0.01 with Adam Optimizer used for optimization.

4.4. Performance Metrics

The performance of the proposed system was assessed using the standard evaluation matrix illustrated in Figure 4.



		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TPr	FPr
	NEGATIVE	FNr	TNr

Figure 4. Standard Evaluation Parameters

The standard validation parameters are described below:

- True Negative (TN) - These are accurately forecasted negative outcomes, demonstrating that the value of actual class is zero and the outcome of the anticipated class is zero i.e. correct prediction of negative classes.
- True Positive (TP) - TP are observed positives that are accurately predicted and indicate that the outcome of the actual class and the outcomes of the expected class are positive i.e. correct prediction of positive classes.

False negative and false positive occur if actual class is different from the anticipated class.

- False Positive (FP) – Positive observations of the predicted class and negative observations of the actual class, indicating inaccurate prediction of positive classes.
- False Negative (FN) - When the actual class is positive while the anticipated class is negative i.e. errorneous prediction of negative classes.

Using these standard parameters the following rules were implemented for the evaluation of the effectiveness of the proposed hybrid model:

Precision (P_r) = The proportion of correctly anticipated positive outcomes to total projected positive outcomes is referred to as precision.

$$P_r = \frac{TP_r}{TP_r + FP_r} \tag{18}$$

Recall (R_r) = the proportion of correctly predicted positive outcomes in the positive class to the total number of observations.

$$R_r = \frac{TP_r}{TP_r + FN_r} \tag{19}$$

F-Measure (F_r) = F-Measure is defined as the average of Recall and Precision. Both false negatives and false positives are taken into account when calculating the final score.

$$F_r = \frac{2*(P_r * R_r)}{P_r + R_r} \tag{20}$$

Accuracy (A_r) = The most essential performance metric is accuracy, which is simply the fraction of predicted observations that match all observed.

$$A_r = \frac{TN_r + TP_r}{TN_r + TP_r + FN_r + FP_r} \tag{21}$$

5. Experimental Results and Analysis

This section provides detailed information on the results and a comparative study of performance.

5.1. Comparison of Word Embeddings

The proposed model has been used for the classification of sentiments. The overall performance of the word embedding is assessed using a weighted average of P_r, R_r, and F_r.

The experiments have been performed to evaluate the overall efficiency of classification on word2vec, Glove and the FastText embedding methods on the used dataset. The performance of efficiency of different word embeddings is represented in Table 4. Glove embedding was observed to be less efficient and lower accuracy as compared to the other two methods. The FastText method of embedding attained optimized performance.

Table 4. Performance of Word Embeddings

Word Embedding	**Weighted Average(%)**			**Acc(%)** ***(Ar)***
	Pr	***Rr***	***Fr***	
Word2Vec	9236	92.48	92.42	92.83
Glove	9332	94.15	93.73	94.22
FastText	95.92	95.56	95.74	96.32

Compared with Word2Vec, Glove embedding has shown improved efficiency by 0.96%, 1.67%, 1.31% and 2.49% on P_r, R_r, $F_r$and A_r Furthermore, the efficiency of Word2Vec is comparable to hat of FastText Embedding and the improvement has been observed to be 2.56%, 3.08%, 3.32%, 3.49% in terms of precision, recall, F-measure, and accuracy parameters respectively. When comparing Keras and FastText, it has been noticed that the FastText method has performed better than Word2Vec Embedding on the given dataset in the text, if used as mathematical symbols.

5.2. Performance Comparison with Deep Learning Methods

CNN vs Proposed Model: In this experiment, the classification of emotion exhibited in textual airlines reviews using the proposed model is compared to that of a single layer CNN model (Figure 5). In comparison to the suggested Attention-based proposed model, a single layered CNN model produced unsatisfactory outcomes (*Pr*: 86.35%, *Rr*: 86.98, and *Fr*: 86.66% for positive classification, *Pr*: 86.01%, *Rr*: 86.88 and *Fr*: 86.44% for neutral classification, *Pr*: 85.32%, *Rr*: 84.33 and *Fr*: 84.82% for negative classification and *Ar*: 86.35% accuracy). The reason for CNN's poor performance is that it is unable to retain the text's sequencing order required for the text categorization problem for keeping record of the details of ordering to provide improved classification results (*Online*).

Table 5. Performance Comparison of Deep Learning Methods

Models with FastText	**Class 1**			**Class 2**			**Class 3**			**Acc%** ***(Ar)***
	Pr	***Rr***	***Fr***	***Pr***	***Rr***	***Fr***	***Pr***	***Rr***	***Fr***	
CNN	86.35	86.98	86.66	86.01	86.88	86.44	84.32	84.22	84.27	86.35
BiLSTM	88.15	88.98	88.56	90.11	89.32	89.71	85.32	84.33	84.82	89.32
CNN-BiLSTM	89.21	88.32	88.76	83.35	83.98	83.66	87.65	87.33	87.49	90.35

Table 5. Performance Comparison of Deep Learning Methods ((continued))

Models with FastText	Class 1			Class 2			Class 3			Acc% (Ar)
	Pr	***Rr***	***Fr***	***Pr***	***Rr***	***Fr***	***Pr***	***Rr***	***Fr***	
BiLSTM-Attention	87.65	86.32	86.98	93.26	93.14	93.20	90.25	91.25	90.75	92.63
Proposed Model	94.65	94.12	94.38	96.98	96.25	96.61	95.12	95.56	95.34	97.01

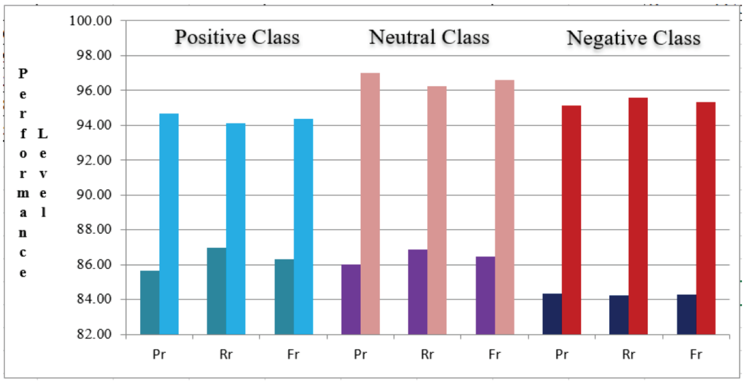


Figure 5. Performance Comparison of CNN and proposed model

BiLSTM vs. Proposed Model: The performance of the Bi-LSTM model was analyzed and comparison was performed with the proposed model in the next experiment (Figure 6). The Bi-LSTM delivers lower outcomes (*Pr*: 88.15%, *Rr*: 88.98%, Fr: 87.95% for positive classification, *Pr*: 85.32%, *Rr*: 84.33, *Fr*: 84.82% for neutral classification, *Pr*: 90.11%, *Rr*: 89.32, *Fr*: 89.71% for negative classification, *Ar*:89.32% accuracy) than the suggested model. The reason for degradation in the performance of the Bi-LSTM model is that the process is costly in terms of memory utilization due to double LSTM cell requirements (Zhu et al., 2018).

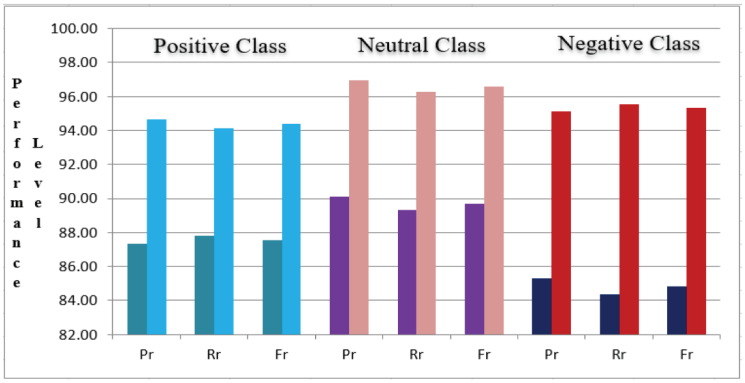


Figure 6. Performance Comparison of BiLSTM and Proposed Model

CNN-BiLSTM vs. Proposed Model: Furthermore, the research was performed to compare the performance results obtained from the proposed model to the CNN-BiLSTM model (Figure 7). In comparison to the suggested technique, the CNN-BiLSTM delivers poor outcomes (Pr: 89.21%, Rr: 88.32%, Fr: 88.76% for positive classification, *Pr*: 83.35%, *Rr*: 83.98, *Fr*: 83.66% neutral classification, *Pr*: 87.65%, *Rr*: 87.33, *Fr*: 87.49% for negative classification and *Ar*: 90.35% accuracy). When compared to the suggested model, BiLSTM performs poorly because it lacks the attention mechanism (Yao et al., 2017).

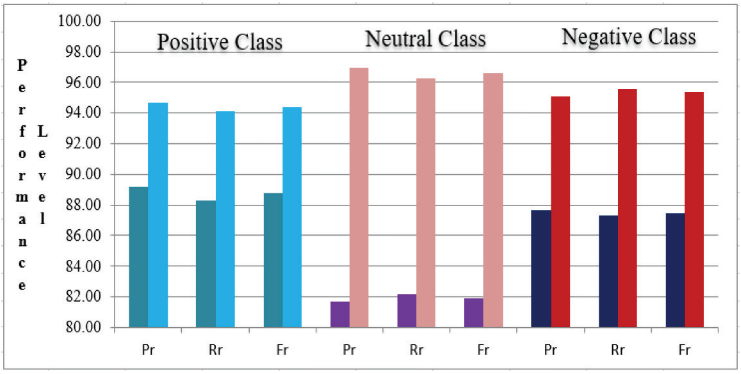


Figure 7. Performance Comparison of CNN-BiLSTM and Proposed Model

BiLSTM-Attention vs. Proposed Model: In the final phase of comparison the experiments were performed for comparison of results of BiLSTM-Attention model with proposed model (Figure 8). Comparing to the suggested technique, the CNN-BiLSTM delivers poor outcomes (*Pr*: 87.65%, *Rr*: 86.32%, *Fr*: 86.98% for positive classification, *Pr*: 93.26%, *Rr*: 93.14, *Fr*: 93.20% neutral classification, *Pr*: 90.25%, *Rr*: 91.25, *Fr*: 90.75% for negative classification and *Ar*: 92.63% accuracy). It was observed that the Attention mechanism has the disadvantage of having to pay attention to all words on the side of source for each target text, which is costly and makes translating longer sequences unfeasible (Zhang et al., 2018).

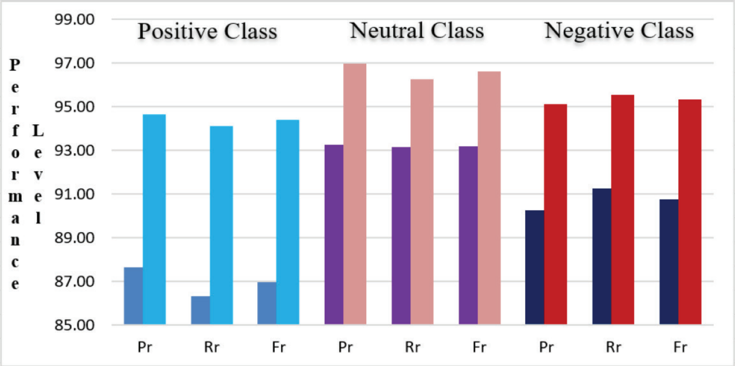


Figure 8. Performance Comparison of BiLSTM-Attention and Proposed Model

Except for the proposed model, none of the models perform consistently across all classes, as shown in Table 4. The experimental outcomes show that the BiLSTM model exhibits much lower performance for negative class classification whereas CNN-BiLSTM shows degraded performance for neutral class classification. In terms of positive class categorization, the BiLSTM-Attention model performs poorly. Finally, the consistent and highest level of performance of the proposed model was with *Pr*: 94.65%, *Rr*: 94.12%, *Fr*: 94.38% for positive classification, *Pr*: 96.98%, *Rr*: 96.25, *Fr*: 96.61% neutral classification, *Pr*: 95.12%, *Rr*: 95.56, *Fr*: 95.34% for negative classification and *Ar*: 97.01% accuracy).

Figure 9 shows the overall categorization accuracy of several deep learning approaches. The CNN approach has an accuracy of 86.35 percent, BiLSTM has an accuracy of 89.32 percent, and a CNN-BiLSTM and BiLSTM-Attention approach have an accuracy of 90.35 percent and 92.63 percent, respectively. With a 97.01 percent accuracy rating, the suggested model exceeds all previous models.

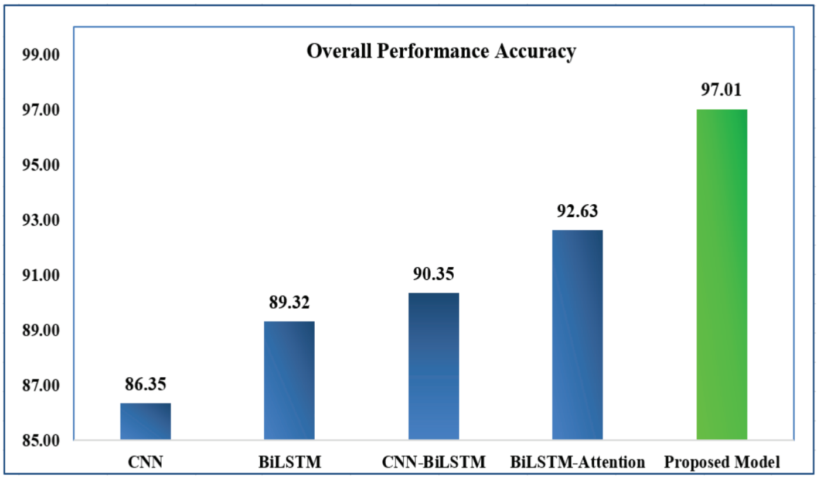


Figure 9. Performance Comparison of Overall Accuracy

5.3. Validation of Classification using ROC Curve

To verify the correctness for classification of the proposed model, a comparison of Training Accuracy and Training Loss has been performed among the deep learning model. A receiver operating characteristic (ROC) curve graph (online) has been used to illustrate the evaluation results. In Figure 10, of ROC Curve graph X-axis denotes FPR (False Positive Rate) and Y-axis denotes TPR (True Positive Rate) of the classification. The area under the curve is a number that goes from 0 to 1, the AUC (Area under the ROC Curve) under ROC values near 1 imply that the model is performing well. As observed in the figure, the performance of the proposed model is better than other models. Figure 10 shows that the proposed model shows promising results in terms of classification.

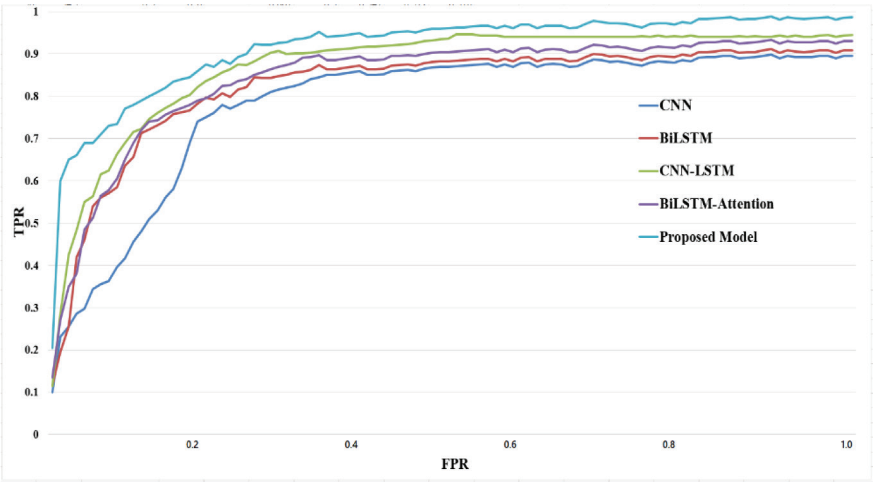


Figure 10. Comparison of the proposed model for classification using ROC Curves

In Figure 10 it is noticeable that the proposed model exhibited better performance than the other deep learning models. The observations obtained from ROC curves demonstrate that the proposed model achieves higher accuracy.

6. Conclusion

This study produced a sentiment analyzer that could be used to extract people's opinions regarding Indian Airline services as expressed on social media. The three pre-trained word embedding algorithms that were trained and assessed were Word2Vec, Glove, and FastText, with FastText outperforming the others in terms of word vectorization accuracy. Further experiments on deep learning models were implemented using FastText embeddings as it has shown the highest accuracy. The research findings revealed that the proposed model worked admirably on the obtained dataset, even beating the baseline classifier. The proposed model achieved an F-measure of 94.38% for the classification of positive reviews, 96.61% for neutral, and 95.32% for negative reviews classification. The overall accuracy achieved from the proposed model is 97.01%, which significantly outperforms other deep learning models. The results show that other models did not perform consistently for positive, negative, and neutral classes and the proposed model proved to be consistent. The proposed model achieves the highest accuracy of classification over other models. The outcomes confirmed the utility of the proposed model as a viable option for addressing the users' emotions regarding the services on social media.

References

Baziotis, C., Pelekis, N., and Doulkeridis C., 2017. Data Stories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based Sentiment Analysis. *In Proc. 11th Int. Workshop Semantic Eval* (pp. 747-754).

Chen, M., Herrera, F., and Hwang, K., 2018. Cognitive computing: Architecture, technologies and intelligent applications. *IEEE Access*, vol. 6 (pp. 19774-19783)

Chollet, F., 2017. *Deep Learning With Python*. Shelter, Island: Manning.

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol* (pp. 4171-4186).

Dzikienė, K., Damaševičius, J., and Woniak, R. M., 2019 Sentiment Analysis of Lithuanian Texts Using Traditional and Deep Learning Approaches. *Computers*, *8*, 4. [Online]. Available: https://doi.org/10.3390/computers8010004

Fan, Y.X., Guo, J.F., and Lan, Y.Y., 2017. Context-based deep semantic sentence retrieval model. Chin. J. Inform. Sci. 31(5), 161-167

Gu, S., Zhang, L., Hou, Y., and Song, Y., 2018. A position-aware bidirectional attention network for aspect-level Sentiment Analysis. In *Proceedings of International Conference on Computing Linguistics* (pp. 774-784).

Guo, Y., Li, W., Jin, C., Duan, Y., and Wu, S., 2018. An integrated neural model for sentence classification. In *Proc. Chin. Control Decis. Conf. (CCDC)* (pp. 6268-6273).

Han, H., Bai, X., and Li, P., 2019. Augmented sentiment representation by learning context information. *Neural Comput. Appl.*, vol. 31, n. 12 (pp. 8475-8482).

Hu, R., Rui, L., Zeng, P., Chen, L., and Fan, X., 2018. Text Sentiment Analysis: A review. *In Proc. IEEE 4th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2018 (pp. 2283-2288).

Huang, B., Qu, Y., and Carley, K., 2018. Aspect level sentiment classification with attention-over-attention neural networks. In *Proc. Conf. Social Comput* (pp. 197-206).

Huang, Z., Xu, W., and Yu, K., 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv: 1508.01991*. [Online]. Available: https://doi.org/10.48550/arXiv.1508.01991

Hwang, K., and Chen, M., 2017. Big Data Analytics for Cloud. *IoT and Cognitive Computing*. Hoboken, NJ, USA: Wiley,

Jiang, Z., Gao, S., and Chen, L., 2019. Study on text representation method based on deep learning and topic information. *Computing*, vol. 102, n. 3 (pp. 623-642).

Kim, K., Chung, B. S., Choi, Y., Lee, S., Jung, J.-Y., and Park, J., 2014. Language independent semantic kernels for short-text classification, *Expert Syst. Appl.*, vol. 41, n. 2, pp. 735_743, Feb. 2014.

Lebret, R., and Collobert, R., 2013. Word emdeddings through Hellinger PCA., arXiv:1312.5542. [Online]. *Available:* https://doi.org/10.48550/arXiv.1312.5542

Liu, B., 2012 Sentiment Analysis and opinion mining. *Synth. Lect. Hum. Lang. Technol*., 5, 1-167.

Liu, G., Xu, X., Deng, B., Chen, S., and Li, L., 2016. A hybrid method for bilingual text sentiment classification based on deep learning. In *Proc. 17th IEEE/ACIS Int. Conf. Softw. Eng*., *Artif. Intell*., *Netw. Parallel/Distrib.Comput. (SNPD)* (pp. 93-98).

Liu, W., Cao, G., and Yin, J., 2019. Bi-level attention model for Sentiment Analysis of short texts. *IEEE Access*, vol. 7, (pp. 119813-119822).

Ma, Y., Peng, H., Khan, T., Cambria, E., and Hussain, A., 2018. Sentic LSTM: A hybrid network for targeted aspect-based Sentiment Analysis. *Cognitive Computing*, vol. 10, n. 4.

Ma, X., and Hovy, E., 2016. End-to-end sequence labeling via bi-directional LSTM-CNNS-CRF. *arXiv:1603.01354*. [Online]. Available: https://arxiv.org/abs/1603.01354

Neethu, M. S., and Rajasree, R., 2013. Sentiment Analysis in Twitter using machine learning techniques. 2013, *In Proc. 4th Int. Conf. Comput*., *Commun.Netw. Technol. (ICCCNT)*, pp. 1-5.

Pang, B., and Lee, L., 2008. Opinion mining and Sentiment Analysis. *Found. Trends Inf. Retr*., 2, 1-135.

Pennington, J., Socher, R., and Manning, C., 2014. Glove: Global vectors forword representation. In *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)* (pp. 1532-1543).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer L., 2018. Deep contextualized word representations. *Journal of Associative Computing Linguistics*, vol. 1 (pp. 2227-2237). [Online]. Available: https://www.quora.com/What-are-some-of-thelimitations-or-drawbacks-of-Convolutional-Neural-Networks

Pham, D.H., and Le, A.C., 2018. Exploiting multiple word embeddings and one-hot character vectors for aspect-based Sentiment Analysis. *Int. J. Approx. Reasoning*, vol. 103 (pp. 1-10).

Rehman, A. U., Malik, A. K., Raza, B., and Ali, W., 2019. A hybrid CNN-LSTM model for improving accuracy of movie reviews Sentiment Analysis, *Multimedia Tools Appl.*, vol. 78, n. 18 (pp. 26597-26613).

Rezaeinia, S. M., Rahmani, R., Ghodsi, A., and Veisi H., 2019. Sentiment Analysis based on improved pre-trained word embeddings. *Expert Syst. Appl.*, vol. 117 (pp. 139-147)

Seo, S., Kim, C., Kim, H., Mo, K., and Kang, P., 2020. Comparative Study of Deep Learning-Based Sentiment Classification. In *IEEE Access*, vol. 8 (pp. 6861-6875).

Simonyan, K., and Zisserman, A., 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. *London, U.K.: Univ. of Oxford*

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C., 2013. Recursive deep models for semantic compositionality over a sentiment Treebank. *In Proc. Conf. Empirical Methods Natural Lang. Process* (pp. 1631-1642).

Sun, B., Tian, F., and Liang, L., 2018. Tibetan micro-blog Sentiment Analysis based on mixed deep learning. In *Proc. Int. Conf. Audio, Lang. Image Process. (ICALIP)* (pp. 109-112).

Tang, D., Qin, B., Feng, X., and Liu, T., 2016.Effective LSTMs for target- dependent sentiment classification. In *Proc. COLING 26th Int. Conf. Comput. Linguistics* (pp. 3298-3307).

Tay, Y., Tuan, L., and Hui, S., 2018. Learning to attend via word-aspect associative fusion for aspect-based Sentiment Analysis. In *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)* (pp. 5956-5963).

Wint, Z. Z., Manabe, Y., and Aritsugi, M., 2018. Deep learning based sentiment classification in social network services datasets. In *Proc. IEEE Int. Conf. Big Data, Cloud Comput*., *Data Sci. Eng. (BCD)* (pp. 91-96).

Xu, G., Meng, Y., Qiu, X., Yu, Z., and Wu, X., 2019. Sentiment Analysis of comment texts based on BiLSTM,. IEEE Access, vol. 7, pp. 51522-51532.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E., 2016. Hierarchical attention networks for document classification. *In Proc Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol* (pp. 1480-1489).

Yao, X. L., 2017. Attention-based BiLSTM neural networks for sentiment classification of short texts. In *Proceedings of International Conference Information Science Cloud Computing* (pp. 110-117).

Yin, W., Kann, K., Yu, M., and Schütze, H., 2017. Comparative study of CNN and RNN for natural language processing, 2017, arXiv: 1702.01923. [Online]. Available: https://doi.org/10.48550/arXiv.1702.01923

Yu, Q., Zhao, H., and Wang Z., 2019. Attention-based bidirectional gated recurrent unit neural networks for Sentiment Analysis. In *Proc. 2nd Int. Conf. Artif. Intell. Pattern Recognit.* Cham, Switzerland (pp. 67-78). Springer.

Zhang, Y., Wang, J., and Zhang, X., 2018. YNU-HPCC at SemEval-2018 task 1: BiLSTM with attention based Sentiment Analysis for affect in tweets. In *Proceedings of 12th Int. Workshop Semantic Eval.* (pp. 273-278). [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html

Zheng, J., and Zheng, L., 2019. A hybrid bidirectional recurrent convolutional neural network attention-based model for text classification, *IEEE Access*, vol. 7 (pp. 106673-106685)

Zhou, K., and Long, F., 2018. Sentiment Analysis of text based on CNN and bi-directional LSTM model. In *Proc. 24th Int. Conf. Autom. Comput. (ICAC)*, (pp. 1-5).

Zhou, X., Wan, X., and Xiao, J., 2016. Attention-based LSTM network for cross-lingual sentiment classification. In *Proc. Conf. Empirical Methods Natural Lang. Processing* (pp. 247-256).

Zhu, Y., Gao, X., Zhang, W., Liu, S., and Zhang, Y., 2018. A bi-directional LSTM CNN model with attention for aspect-level text classification. *Future Internet*, vol. 10, n. 12,116.