IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 19, 2020, accepted April 14, 2020, date of publication April 17, 2020, date of current version May 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988550

Sentiment Classification Using a Single-Layered BiLSTM Model

ZABIT HAMEED, (Member, IEEE), AND BEGONYA GARCIA-ZAPIRAIN, (Member, IEEE)
eVida Research Group, University of Deusto, 48007 Bilbao, Spain

Corresponding author: Zabit Hameed (zabithameed@deusto.es)

This work was supported by the eVida research group, University of Deusto, Bilbao, Spain, under Grant IT 905-16.

ABSTRACT This study presents a computationally efficient deep learning model for binary sentiment classification, which aims to decide the sentiment polarity of people's opinions, attitudes, and emotions expressed in written text. To achieve this, we exploited three widely practiced datasets based on public opinions about movies. We utilized merely one bidirectional long short-term memory (BiLSTM) layer along with a global pooling mechanism and achieved an accuracy of 80.500%, 85.780%, and 90.585% on MR, SST2 and IMDb datasets, respectively. We concluded that the performance metrics of our proposed approach are competitive with the recently published models, having comparatively complex architectures. Also, it is inferred that the proposed single-layered BiLSTM based architecture is computationally efficient and can be recommended for real-time applications in the field of sentiment analysis.

INDEX TERMS Bidirectional long short-term memory, deep learning, long-term dependencies, natural language processing, sentiment analysis.

I. INTRODUCTION

Natural Language Processing (NLP) is an illustrious field of computer science associated with the interaction between human and machine languages [1]. In NLP, language modeling is one of the crucial tasks aiming at the assignment of probability to a sequence of words and is used in various domains, including text categorization [2]. An automatic text classification plays a vital role in numerous applications like email spam detection [3] and sentiment classification [4]. To achieve this, an efficient representation of a document is a key step in order to retrieve the associated sentiment. A conventional and significant approach used for the depiction of a text corpus is called bag-of-words (BoW). However, it ignores the order and semantic of words while characterizing a text [5], [6]. On the other hand, n-gram models, an extension of BoW, are considered prominent for statistical language modeling but suffer from data sparsity [5], [6]. To this direction, word embedding [7], [8], a representation of word as a low-dimensional vector, offered significant results and gained tremendous success in text classification, compared to the aforementioned techniques.

The associate editor coordinating the review of this manuscript and approving it for publication was Alicia Fornés.

In recent years, neural networks [9] gained popularity due to their ability to learn features automatically and to train complex models on giant datasets. Especially, deep learning models manifested various state-of-the-art performances in language translation [10] and sentiment classification [11]. Currently, convolutional neural network (CNN) [12] and recurrent neural network (RNN) [13], based on word vectors [7], [8], are widely utilized; where the former helps in the extraction of local features from text data and the latter deals with long-term dependencies among words in the text corpus. Also, long short-term memory (LSTM) [14], a variant of RNN, outperforms the traditional RNN due to its ability to memorize the sentiments of words in long texts. To this end, various studies [35]–[41] investigated the individual and combined effects of CNN and LSTM along with the lexicon and attention mechanisms for the sentiment classification. Although these state-of-the-art studies outperformed several existing models, their existing underlying structures are relatively more complex.

The present study introduces a comparatively simple and robust approach for binary sentiment classification. The main contributions of this paper are: 1) We employed a standard and highly practiced unsupervised embedding method for weight initialization, called global vector (GloVe) [8]. 2) Then, we utilized merely one bidirectional

LSTM (BiLSTM) layer [22], [26] together with the ensemble of one global maximum layer and one global average pooling layer. 3) Of note, we solely focused on the long-term dependencies among words in this paper. 4) Our contribution focused on the development of a framework that gives a computationally efficient and robust performance with few parameters. 5) The results demonstrate that our framework offers competitive results compared to the recently proposed studies with complicated architectures [35]–[41].

The remaining sections of this paper are organized as follows. Section II presents the related research work. Section III describes the methods along with our proposed model. Section IV depicts the experimental setup and section V shows the performance metrics. Section VI discusses the results and finally, section VII outlines the conclusion and future prospects.

II. RELATED WORK

In this section, we present the literature of sentiment analysis by highlighting the traditional machine learning and deep learning approaches.

A. MACHINE LEARNING APPROACH

Sentiment analysis is a vital task in NLP and extensive research has been conducted in this area during the last few decades [11]. In NLP systems, words are scrutinized as discrete entities where a model exploits minimal information regarding the possible interaction among them. The traditional approach leverages BoW model for mapping a text corpus into a feature vector, followed by a machine learning classifier. Various studies utilized conventional machine learning practices for sentiment classification tasks. For instance, Moraes *et al.* [15] employed BoW approach for document representation and made an empirical comparison between the classification performance of support vector machine (SVM), naive bayes (NB), and artificial neural network (ANN). Similarly, Paltoglou and Thelwall [16] utilized BoW features together with the emotionally-annotated sentences for the representation of documents. Recently, Mozetič *et al.* [17] used BoW model along with SVM and NB classifiers for Twitter sentiment analysis of thirteen different languages. These methodologies mainly rely upon the BoW representation which may omit information regarding the semantics and ordering of words. Furthermore, numerous authors leveraged the n-gram model by incorporating the word order when mapping a document into a feature vector. For example, Pang *et al.* [18] used the n-gram approach for the depiction of movie reviews and employed NB, SVM, and maximum entropy (MaxEnt) for the binary sentiment classification. Likewise, Wang and Manning [19] utilized n-grams features in conjunction with SVM and NB classifiers for text classification. Recently, Tripathy *et al.* [20] utilized different n-gram models for portraying a text corpus of movie reviews and then analyzed the performances of four different machine learning classifiers, namely, SVM, NB, MaxEnt, and stochastic gradient descent (SGD). These approaches took into account the ordering of words yet suffered from the issue of data sparsity.

To sum up, these classical machine learning strategies are still a choice of interest in sentiment analysis. However, their limitations include the lack of word order and data sparsity. Moreover, these methods are mainly dedicated to the manual extraction of features in order to train a classifier such as SVM. However, the selection of effective features requires domain expert knowledge, which is labor-intensive and more complicated.

B. DEEP LEARNING APPROACH

Deep learning techniques play a crucial role in NLP where most of the tasks are associated with the methods relying on distributed word representation models, such as Word2Vec [7] and GloVe [8]. These word embedding methods portray words into meaningful dense vectors and are excessively used in sentiment analysis. This approach of word mapping is preferred over the traditional BoW model due to its ability to encounter the semantic and syntactic characteristics of words within a document. Also, word embedding offers dimensionality reduction and thus helps to overcome the issue of data sparsity in BoW approach. In deep learning scenario, words are first depicted as dense vectors, and then a classifier based on neural network is used for text categorization. Especially, RNN is capable of extracting appropriate features from data, and could be a prominent choice to capture the semantics of long texts. However, an RNN is a biased model since it gives high priority to recently occurred words in a sequence, which might reduce its efficiency when capturing the semantic of an entire document [21]. As a result, LSTM [14] was introduced in order to overcome the shortcomings of long-term dependencies in the RNN models. An extension of LSTM is called BiLSTM which is comprised of two LSTM cells; forward LSTM and backward LSTM [22]. The difference between both is the fact that LSTM considers all the previous words in order to extract a sentiment of the text, whereas BiLSTM scrutinizes past as well as future words for the same task [22].

Recently, numerous studies leveraged pre-trained word embedding algorithms as an input to the deep learning models for sentiment classification tasks. For instance, Zhang and Wallace [23] used pre-trained Word2Vec [7] and GloVe [8] embedding vectors to train CNN models and improved the sentiment classification of various datasets. Wang *et al.* [24] utilized pre-trained GloVe vectors as inputs to the attention-based LSTM network and achieved state-of-the-art results for aspect-level sentiment categorization. Also, Fu *et al.* [25] employed GloVe vectors and proposed a lexicon-enhanced LSTM model along with attention mechanism. It is manifested that the proposed architecture incorporates sentiment information of words and thus provides high performance in sentiment analysis tasks. Similarly, Xu *et al.* [22] initialized the input feature vectors with sentiment-oriented Word2Vec and found that BiLSTM outperforms CNN and LSTM in extracting the sentiments of Chinese comment texts.

These novel architectures usually followed either series or parallel combinations of deep learning models along with lexicons or attention mechanisms. However, these approaches are less straight-forward and resource-intensive at the same time. In summary, these novel frameworks provided promising results in terms of sentiment classification but may not be suitable for real-time applications because of their underlying complexity.

To this end, we focused on the simplicity and stability of the architecture, by taking into account only the long-term dependency in a text. We concluded that using a pre-trained embedding with a single-layer BiLSTM along with global pooling concept could produce competitive results, compared to recently presented approaches [36]–[41]. In the following subsections, we introduced the LSTM network along with its associated mathematical details. Then, we presented the detailed framework of our proposed model.

C. LSTM NETWORK

In 1997, Hochreiter and Schmidhuber proposed an LSTM network [14] in order to overcome the shortcomings of vanishing and exploding gradient in the RNN model. The key concept behind an LSTM model is to regulate the cell states by using three gates, namely, input, forget and output gates, as depicted in Figure 1. The forget gate (f_t) determines whether to forget or keep the information of previous state (c_{t-1}) by looking at the values of input (x_t) and hidden state (h_{t-1}) and its output value maybe a 0 or 1. Similarly, the input gate (i_t) decides how much information of the input text (x_t) and h_{t-1} should pass in order to update the cell state, and its output maybe a 0 or 1. The value of c_t represents the generated cell state as a result of mathematical operations on c_{t-1}, f_t and i_t. The output gate (o_t) controls the flow of information from the current cell state to the hidden state, and its value maybe a 0 or 1. The mathematical details for these gates are given in equations (1) – (5). Where $x_t \in R^n$ is the input vector, $W \in R^{v*n}$, $b \in R^v$ and the superscripts n and v depict the dimension of the input vector and the number of words in the dataset or vocabulary, respectively. At any time, t, the inputs to LSTM are input vector x_t, previously hidden state h_{t-1}, and previous cell state c_{t-1}, whereas the outputs are current hidden state h_t and current cell state c_t. The symbol $\odot$ represents element-wise vectors multiplication.

$$f_t = sigmoid(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \quad (1)$$
$$i_t = sigmoid(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \quad (2)$$
$$c_t = c_{t-1} \odot f_t + i_t \odot tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \quad (3)$$
$$o_t = sigmoid(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \quad (4)$$
$$h_t = o_t \odot tanh(c_t) \quad (5)$$

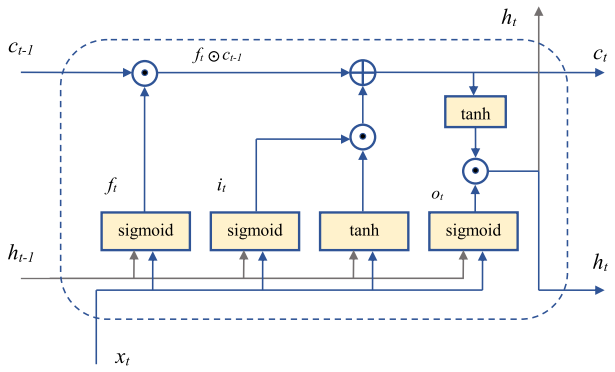

FIGURE 1. An illustrative block diagram of LSTM network [14].

III. PROPOSED MODEL ARCHITECTURE

Sentiment analysis or opinion mining can be categorized into binary classification or multi-class classification. In binary classification, each document d_i is classified as a label C, where d_i belongs to ($D = d_1, d_2, d_3, \ldots, d_m$) and C depicts any of two predefined classes ($C = 2$). On the other hand, in multi-class classification, each document d_i is categorized as a label C in the predefined multi classes ($C > 2$). In this study, we utilized a binary approach for the prediction of a positive or negative sentiment of a document. Thus in our case, C is 2 and m is the number of documents (movie reviews) in a given dataset.

The architecture of our proposed model is shown in Figure 2. It is comprised of an input layer, an embedding layer, a BiLSTM layer followed by the ensemble of global average and global maximum pooling layers, and one sigmoid layer at the output. Further details of every layer are provided in the following subsections.

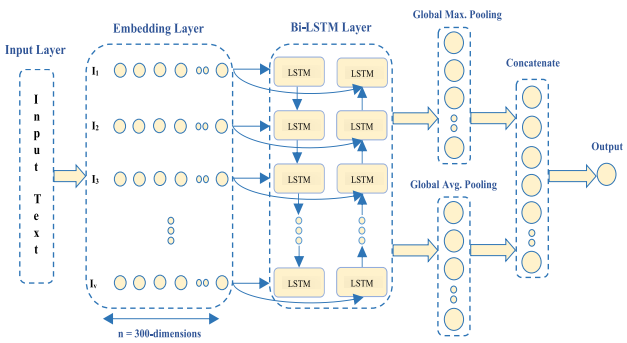

FIGURE 2. The proposed computationally efficient architecture along with layers from input to output of the model.

A. INPUT LAYER

The input layer is considered as a starting point of the network. Let $w_1, w_2, w_3, \ldots, w_v$ be the total number of unique words in the dictionary $D = d_1, d_2, d_3, \ldots d_m$, then $i_1, i_2, i_3, \ldots, i_v$ correspond to the total unique indices. Indices represent natural numbers where the subscripts 1 and V depict the first and last index in the vocabulary, respectively. The input layer carries data samples as a sequence of unique indices of same length.

B. EMBEDDING LAYER

The second layer of our proposed architecture is called embedding layer where every index, corresponding to a

unique word in the data set, is transformed into a real-valued feature vector. These real-valued vectors are stacked together to form a matrix, called an embedding matrix, as shown in the equation (6). The intuition behind the embedding matrix is that every row depicts a unique index which in turn corresponds to a unique word in the vocabulary. The embedding matrix has a dimension of $v * d$, where v depicts the size of dataset vocabulary and d portrays the dimension of dense vector. In our paper, we used a 300-dimensional Glove vector [8] as a pre-trained word embedding vector.

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ r_{3,1} & r_{3,2} & \cdots & r_{3,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{v-1,1} & r_{v-1,2} & \cdots & r_{v-1,n} \\ r_{v,1} & r_{v,2} & \cdots & r_{v,n} \end{bmatrix} \quad (6)$$

C. BiLSTM LAYER

In the LSTM network [14], information propagate entirely in a forward direction which indicates that the state of time t only depends on the information before t. However, to characterize the whole semantic of an input review, subsequent information are equally effective as the previous ones. Thus, for a better representation of contextual information, Bidirectional LSTM (BiLSTM) model [22], [26] was employed. The BiLSTM model is composed of two LSTM networks and is capable of reading input reviews in both directions, forward and backward. The forward LSTM processes information from left to right and its hidden state can be shown as $\overrightarrow{h}_t = LSTM(x_t, \overrightarrow{h}_{t-1})$ whereas the backward LSTM processes information from right to left and its hidden state can be expressed as $\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t+1})$. Finally, the output of BiLSTM can be summarized by concatenating the forward and backward states as $h_t = [\overrightarrow{h}_t, \overleftarrow{h}_t]$.

D. GLOBAL POOLING LAYER

The outputs of BiLSTM layer are simultaneously passed to the global maximum and global average pooling layers. The former and latter layer retrieves a maximum and an average value of each feature in the BiLSTM layer, respectively. We exploited merely one global pooling layer (each) as an alternative for the dense layer(s).

E. CONCATENATE LAYER

The concatenate layer takes the global maximum and the global average layers and merges them into a single layer before passing it to the final layer.

F. OUTPUT LAYER

At the output, we utilized binary cross-entropy as a loss function for binary sentiment classification and its mathematical details are provided in equation (7).

Binary cross entropy

$$= -\frac{1}{m}\sum_{i}^{m}(y_i * log\,(p(y_i)) + (1 - y_i) * log(1 - p(y_i))) \quad (7)$$

where m depicts the total number of review samples, y_i shows true labels and $p(y_i)$ represents the probability of true labels.

IV. EXPERIMENTAL SETUP

All the experiments of this study were implemented on corsair one core-i7 computer having 16GB RAM, and Nvidia GeForce GTX 1080 Ti graphic card with 11GB memory. We employed Python 3.7 as the programming language with Tensorflow 1.14 and Keras 2.3 installed on Windows 10. In the following subsections, we provide an overview of the datasets, followed by the preprocessing of data, and eventually, we describe the hyperparameter settings in order to optimize our proposed model.

TABLE 1. Characteristics of the datasets after preprocessing.

Dataset	No. of Classes	Average length	Maximum length	Dataset size	Test set size
MR	2	18.2	50	10662	CV
IMDB	2	221.8	2370	50000	CV
SST2	2	17.4	50	9613	1821

A. ANALYSIS OF DATASETS

We considered three freely available and highly practiced datasets comprised of public opinions about movies, namely, Movie Review (MR) dataset [27], ACL Internet Movie Database (IMDB) [28], and Stanford Sentiment Treebank (SST2) [29]. The main task is to determine whether a movie belongs to a positive or negative class, and the characteristics of all the datasets are shown in Table 1. Further details of these datasets are as follows:

- The MR dataset was introduced by Pang and Lee [27] in 2005 for the sentiment classification task. It is a balanced dataset containing 10,662 short reviews, among which 5,331 carry positive sentiments whereas other 5,331 portray negative sentiments. The average length of overall opinions is noted as 18.2 tokens whereas the maximum review recorded as 50 tokens.
- The IMDB dataset was proposed by Maas *et al.* [28] in 2011 as a benchmark for sentiment analysis. It is also a balanced dataset containing 50,000 reviews, among which 25,000 are positive-labeled and 25,000 are negative-labeled opinions. The key aspect of this dataset is that the majority of reviews comprised of several sentences. The average length is observed as 221.88 words, with the longest review of 2,370 tokens.
- The SST2 dataset was presented by Socher *et al.* in 2013 [29] and is comprised of 9,613 review samples. It is a variant of the MR dataset and is not a stable dataset. In this case, the average length is 17.4 words with a maximum review of 50 tokens.

B. PREPROCESSING OF TEXT DATA

Preprocessing aims at removing absurd information from raw text data, and is considered as a primary and vital task in

sentiment analysis. Sun *et al.* [30] precisely reviewed the importance of various preprocessing approaches in sentiments prediction. However, to keep the dataset in the ideal format, we followed the approach of Tripathy *et al.* [20] during the cleaning process. To this end, first, we removed the punctuation from training and testing data. Then, we eliminated all the English stopwords and lowercased all the text reviews. We omitted the lemmatization as it did not improve the classification performance of our model [20]. We utilized Natural Language Toolkit (NLTK) [31] during the overall preprocessing operations. Further, we counted the number of unique words in the whole dataset and created a dictionary. Eventually, we assigned an index, representing a natural number, to every unique word in the dataset. In this way, we represented every review text as a combination of unique indices. In order to feed our model, these indices were further converted into low dimensional dense vectors, which has been discussed in section III.

C. HYPERPARAMETERS SETTING

Neural networks are capable of learning complicated relationships among their inputs and outputs [32]. However, many of these connections might be the result of sampling noise, so they will be present during the training process but actually do not exist in real test data. This issue may lead to overfitting and thus reduces the predictive capability of the model [32]. We employed the following methods [22], [32] in order to reduce the overfitting in our proposed system. The optimal values of hyperparameters are provided in Table 2 and are briefly discussed in the succeeding subsections.

TABLE 2. Optimal hyperparameters of the proposed model.

Hyperparameter	MR dataset	IMDb dataset	SST2 dataset
Train approach	CV	CV	Train/val/test
Optimizer	RMSProp	RMSProp	RMSProp
Loss function	Cross-entropy	Cross-entropy	Cross-entropy
Learning rate	0.0001	0.0001	0.0001
Batch size	64	64	64
BiLSTM nodes	16	16	16
Max. length	40	400	45
Epochs	70	45	70
Drop out	0.3	0.3	0.3
Regularizer	L2	L2	L2

1) TRAINING APPROACH

Cross-validation is a significant approach for the evaluation of a model's performance towards the unseen data. Thus, for MR and IMDB datasets, we split the training data into 10 folds by adopting cross-validation approach for the model's optimization [35]–[40]. These subsets are balanced and mutually exclusive, among which nine are used for training whereas one is used for validation. On the other hand, we exploited the train/valid/test (6920/872/1821) approach for SST2 dataset [29]. During both of the aforementioned approaches, we checked the predictive performance of our model on different hyperparameters which are explained in the successive subsections.

2) OPTIMIZER

Neural networks learn the underlying and sophisticated patterns from data by using a stochastic gradient descent (SGD) optimization algorithm [32]. In this paper, we used a variant of SGD optimizer, called root mean squared propagation (RMSprop), proposed by T. Tieleman and G. Hinston [33]. RMSprop is adaptive in the learning process and has the capability to work with mini-batches. It leverages the exponentially weighted averages of the gradients to update its parameters and is normally considered a prominent choice for RNN-based models [33].

3) LOSS FUNCTION

Optimization is a process of searching for parameters that can maximize or minimize a specific function, called an objective function [32]. When minimizing its value, it is also called a loss function, cost function, or error function. An important aspect of deep learning is to choose a suitable loss function and then reduce its value close to zero. In our paper, we used binary cross-entropy as a loss function which is an ideal choice for the binary classification, as explained in section III.

4) LEARNING RATE

In neural networks, an algorithm uses backpropagation approach to update the model weights by the amount called step size or learning rate [32]. It ranges from zero to one and its proper value is crucial for the optimization of weights and offsets of a given model [32]. Its low value may lead to tedious calculations and longer training duration whereas its high value may produce instability in the system [32]. After different setups, we finally selected a learning value of 0.0001.

5) BATCH SIZE

Batch size demonstrates the number of samples a neural network can process before updating its internal parameters [32]. The smaller value of batch size may slow down the training process while its high value requires high memory. Also, a larger batch size may degrade the generalization ability of the deep learning model. Thus, following Fu *et al.* [25], we used the batch size of 64 in order to achieve a low-computational and robust system.

6) NETWORK NODES

A network node is a computational unit that is comprised of weighted input connection(s), a transfer function, and an output connection [32]. Specifically, if the number of nodes within a BiLSTM layer is too low, the learning ability of the network will be limited and may result in underfitting. Alternatively, if the number of nodes is too high, the complexity increases and the model may overfit [22]. As mentioned earlier, our primary goal is to get a computational-friendly system. To this end, we selected one BiLSTM layer with

merely 16 nodes and found that the model performed well without any underfitting or overfitting.

7) MAXIMUM LENGTH

In order to train the model, we need to keep all the input samples of equal length, called maximum length. Thus, if the length of the input data is less than the maximum length, zeros will be padded, otherwise, input samples will be truncated. Ideally, it can be imagined that increasing the length of input data could improve the model's predictive ability, but this might not happen because of data sparsity [22]. Alternatively, if the maximum length is too low, we discard too much useful information [22]. The optimal maximum lengths for MR, IMDB and SST2 datasets are 40, 400, and 45 tokens, respectively. It can be noticed from Figure 3, that increasing the input review length does not improve the model's performance after maximum length.

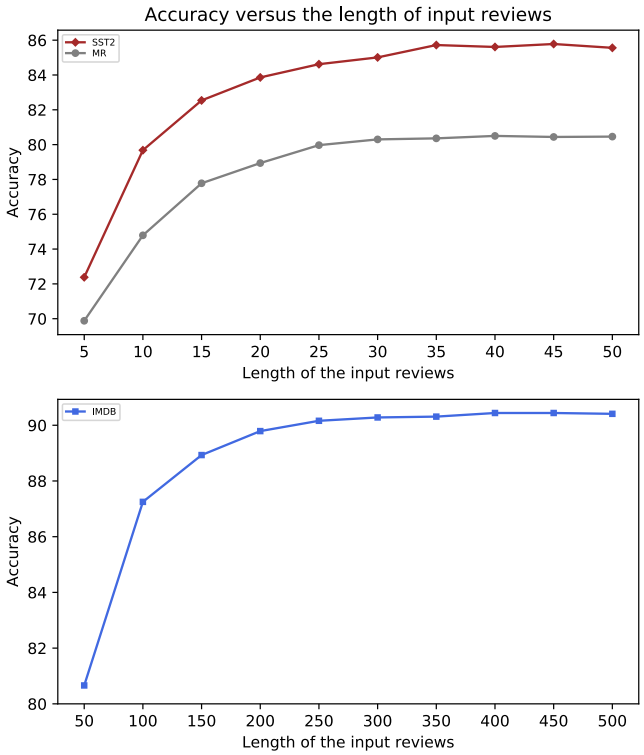

FIGURE 3. The accuracy of proposed model against the selected length of the input reviews.

8) EPOCHS

Epochs represent the number of times a learning algorithm works through the entire training dataset [32]. The generalization ability of a model increases with the number of epochs, however, overfitting may be generated with too many epochs. Thus, an appropriate figure of epochs is required while taking into account the generalized behavior and overfitting of the model. From Figure 4, it can be noticed that the accuracy of our model is not improving after a specific point in every dataset. Specifically, the optimal values of epochs for MR,

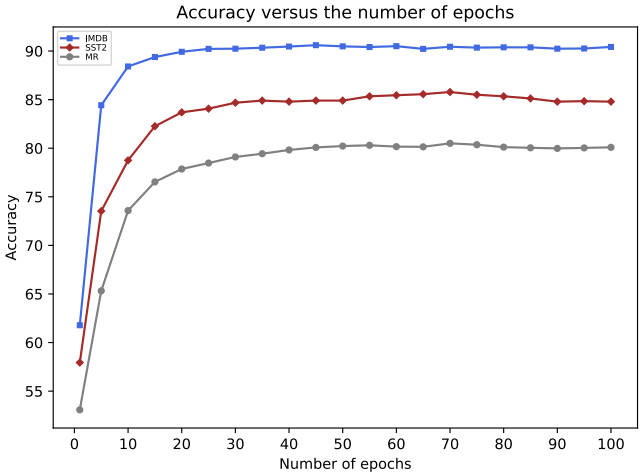

FIGURE 4. The accuracy of proposed model against the selected number of epochs.

IMDB, and SST2 datasets are found to be 70, 45, and 70, respectively.

9) DROPOUT

The dropout, which was developed by Srivastava *et al.* [34], is the simplest approach to reduce overfitting in neural networks. It randomly removes units along with their connections within a network in order to make it thinned. In the embedding layer, we used a dropout value of 0.3, and in BiLSTM layer, we employed dropout and recurrent dropout of the same value.

10) REGULARIZER

Regularization of parameters is an effective approach to prevent deep learning models from overfitting [32]. In fact, it reduces only the generalization error of a model rather than the training error. We can regularize a model by adding a penalty, also called regularizer, to the loss function [32]. Its proper usage makes the model more resistive to overfitting. We applied L2 regularization (also called weight decay) of value 0.001 in the embedding layer as well as in the BiLSTM layer.

V. PERFORMANCE EVALUATION

The overall performance of our proposed model relies on the elements of the confusion matrix, also called error matrix or contingency table. This evaluation matrix contains four terms, namely, True Positive (T_P), False Positive (F_P), True Negative (T_N), and False Negative (F_N) as shown in the Table 2. For a given class c in our problem, which could be either positive or negative, $\bar{c}$ represents the corresponding opposite class. The T_P represents the number of reviews of class c which were correctly classified as class c. Whereas the F_P indicates the number of reviews of class $\bar{c}$ which were incorrectly classified as class c. Also, T_N depicts the number of reviews of class $\bar{c}$ which were correctly classified as $\bar{c}$. Whereas F_N represents the number of reviews of class c that

TABLE 3. Performance of the proposed model for all datasets.

Dataset	Normalized Confusion Matrices			Performance Evaluation Parameters				Average values	
	Predict class → Actual class ↓	0	1	Precision (%)	Recall (%)	F1 score (%)	Test size	Accuracy (%)	F1 score (%)
MR	0	0.8139	0.1861	79.97	81.39	80.67	CV	**80.500**	**80.495**
	1	0.2039	0.7961	81.05	79.61	80.32	CV		
IMDb	0	0.9055	0.0945	90.61	90.55	90.58	CV	**90.585**	**90.580**
	1	0.0938	0.9062	90.55	90.62	90.58	CV		
SST2	0	0.8355	0.1645	87.45	83.55	85.46	912	**85.780**	**85.775**
	1	0.1199	0.8801	84.25	88.01	86.09	909		

were erroneously classified as $\bar{c}$. The effectiveness of our model is evaluated by extracting four parameters from the confusion matrix, namely, precision, recall, accuracy and $F1$ score. These four metrics are discussed briefly below:

- Precision: It quantifies the exactness of a model, and is defined as the ratio of correctly predicted reviews (T_P) to the total number of predicted reviews ($T_P + F_P$) in any class c, where c may be positive or negative class.

$$Precision = T_P/(T_P + F_P) \tag{8}$$

- Recall: It computes the completeness of a model, and is defined as the ratio of correctly predicted reviews (T_P) to the total number of actual reviews ($T_P + F_N$) in any class c, where c may be positive or negative.

$$Recall = T_P/(T_P + F_N) \tag{9}$$

- Accuracy: It evaluates the correctness of a model and is calculated as the ratio of correctly predicted to the total number of reviews.

$$Accuracy = (T_P + T_N)/(T_P + T_N + F_P + F_N) \tag{10}$$

- F_1 score : It represents the harmonic average of precision and recall, and is usually used for the optimization of a model towards either precision or recall.

$$F_1 score = 2*(Precision*Recall)/(Precision+Recall) \tag{11}$$

All the performance parameters of our proposed approach are provided in Table 3.

VI. RESULTS AND DISCUSSION

In this section, we analyzed the competitiveness and effectiveness of our results with the recently published studies by taking into consideration the complexity of the models. First, we mentioned the performance metrics of each dataset. Then, we compared and discussed the results of every dataset with state-of-the-art frameworks in terms of accuracy.

A. MR RESULTS

For MR dataset, the performance metrics of our proposed model are depicted in Table 3. The precision scores for negative and positive reviews are noted as 79.97% and 81.05%, respectively. Also, the recall values for negative and positive sentiments are recorded as 81.39% and 79.61%, respectively. Finally, our proposed model achieved an accuracy of 80.50% along with an F1 score of 80.495%. The competitiveness of our results (accuracy) can be compared with recently proposed studies based on complex architectures, as shown in Table 4. For instance, Socher *et al.* [35] suggested matrix-vector recursive neural network (MV-RNN) architecture for sentiment prediction of movie reviews and obtained an accuracy of 79.00% on MR dataset. The MV-RNN framework relies on the construction of parse trees where complexity of the model increases with longer reviews. In contrast, our model works irrespectively of the input length and offered 1.5 percentage points better accuracy than the MV-RNN approach. Similarly, Fu *et al.* [25] proposed a lexicon-enhanced LSTM model with attention mechanism (ALE-LSTM) and reached an accuracy of 80% on MR dataset. Furthermore, the authors exploited word embedding based ALE-LSTM (WALE-LSTM) approach and got an accuracy of 79.9% on the MR dataset. In both approaches, authors merged various concepts, for example, lexicon and attention mechanisms, but could not achieve remarkable result. Conversely, our approach used a single-layered

TABLE 4. Result comparisons of MR dataset with other methods.

Method	Model complexity	Accuracy (%)
MV-RNN [35]	Combination of parse trees and RNN model	79.00
WALE-LSTM [25]	Fusion of lexicon and attention mechanisms along with LSTM	79.90
BiGRU+CNN [36]	Serial fusion of BiGRU and CNN methodologies	78.30
CNN-GRU-multilevel and multitype fusion [37]	Combination of multilevel and multi-type features based on CNN and GRU networks	80.20
Proposed	**Single-layered BiLSTM**	**80.50**

BiLSTM network and achieved 0.5 and 0.6 percentage points greater accuracy than ALE-LSTM and WALE-LSTM, respectively. Also, Zhang *et al.* [36] suggested a novel architecture by leveraging the series combination of bidirectional gated recurrent unit (BiGRU) and CNN (BiGRU+CNN) and achieved an accuracy of 78.30% on MR dataset. Although this study combined BiGRU and CNN, it could not offer promising results. Contrary to BiGRU+CNN, our model carries solely a BiLSTM network and provided 1.2 percentage points higher accuracy than [36]. Finally, Usama *et al.* [37] presented various models by encountering multilevel and multitype fusion of features along with the combination of LSTM, GRU, and CNN models. Amongst all, the CNN-GRU-multilevel & multitype fusion and CNN-GRU-multilevel & multitype fusion models showed the best results with the accuracies of 79.80% and 80.20%, respectively. On the opposite side, we did not follow the features manipulation and amalgamation of methodologies but still, our model outperformed their best model by 0.3 percentage point.

In summary, our single-layered BiLSTM based model offered significant results on MR dataset, and are competitive with recently published studies [25], [35]–[37], as depicted in Table 4.

B. IMDb RESULTS

For IMDb dataset, the performance metrics of our proposed methodology are also presented in Table 3. The precision measurements for negative and positive reviews are 90.61% and 90.55%, respectively. Similarly, the recall outcomes for negative and positive reviews are 90.55% and 90.62%, respectively. In this case, our proposed framework provided an accuracy of 90.585% as well as an F1 score of 90.580%. Similar to MR results, we compared the competitiveness of IMDb results (accuracy) with recently proposed studies having complex architectures. For instance, Long *et al.* [38] integrated the effects of cognition based attention (CBA) and local text context-based attention (LA) models, followed by the application of the LSTM clasifier and achieved the highest accuracy of 90.10% on IMDb dataset. Although this study exploited two different attention mechanisms in conjunction with the LSTM model, it could not attain satisfying results. Conversely, our approach did not require such concepts and still offered 0.48 percentage point better accuracy than [38]. Similarly, J. Camacho-Collados and M. T. Pilehvar [39] analyzed the combined effects of CNN and LSTM models and got a maximum accuracy of 88.9% on IMDb dataset. Even though this study used different cleaning processes along with the fusion of CNN and LSTM networks, it could not get promising results. On the other side, we employed only BiLSTM as a core layer and achieved comparably 1.68 percentage points better accuracy. Also, Fu *et al.* [25] utilized ALE-LSTM and WALE-LSTM architectures on the IMDb dataset and reported an accuracy of 89.30% and 89.50%, respectively. Again, this paper exploited distinct attention phenomena but only acquired reasonable results. Specifically, our proposed study surpassed ALE-LSTM and WALE-LSTM models by 1.28 and 1.08 percentage points in accuracy, respectively. Finally, Ma *et al.* [40] suggested a feature-based fusion adversarial RNN with attention mechanism (FARNN-Att) and reported an accuracy of 89.22% on the IMDb dataset. Here, the authors used BiLSTM layer with attention phenomenon together with the adversarial training as a regularization technique but could not gain an encouraging results. However, we simply trained a one-layered BiLSTM network with a global pooling layer approach and got 1.36 percentage points higher accuracy than [40].

In summary, our single-layered BiLSTM based approach provided notable results on the IMDb dataset in contrast to the aforementioned studies [25], [38]–[40] with relatively complex structures, as portrayed in Table 5.

TABLE 5. Result comparisons of IMDB dataset with other methods.

Method	Model complexity	Accuracy (%)
LSTM+CBA+LA [38]	Fusion of two different attention mechanisms with LSTM	90.10
CNN+LSTM [39]	Combination of CNN and LSTM with different cleaning processes	88.90
WALE-LSTM [25]	Fusion of lexicon and attention mechanisms along with LSTM	89.50
FARNN-Att [40]	Attention mechanism and adversarial training with BiLSTM	89.22
Proposed	**Single-layered BiLSTM**	**90.585**

C. SST2 RESULTS

For SST2 dataset, the performance measurements of our proposed methodology are ultimately portrayed in Table 3. In this setting, the precision values for negative and positive reviews are 87.45% and 84.25%, respectively. Also, the recall scores for negative and positive reviews are 83.55% and 88.01%, respectively. Eventually, the proposed framework gave an accuracy of 85.780% together with an F1 score of 85.775%. Similar to MR and IMDb results, we compared the robustness of SST2 results (accuracy) with currently published studies with composite architectures. For instance, Socher *et al.* [41] suggested Recursive Neural Tensor Networks (RNTN) which offered an accuracy of 85.40% on the SST2 dataset. Although the RNTN model achieved better performance, its performance mainly depends on the construction of parse trees [35] where the model complexity increases with longer reviews. Also, these annotated sparse trees are not necessarily available for every dataset. Conversely, our proposed network works irrespectively of the input length and also surpassed the RTRN model by 0.38 percentage point in accuracy. Similarly, Zhang *et al.* [36] recommended a system based on the pipelining of BiGRU and CNN (BiGRU+CNN) and got an accuracy of 85.40% on SST2 dataset. Opposite to BiGRU+CNN, our proposed structure depends on BiLSTM only and still provided 1.2 percentage points higher accuracy than [36]. Recently, Usama *et al.* [37] investigated the parallel fusion of CNN and LSTM models along with the multitype selection of features and obtained an accuracy of 85.70% on SST2 dataset. In contrast, we employed a single BiLSTM

layer with few nodes and achieved the same accuracy. Finally, Yang *et al.* [42] demonstrated the combined effect of capsule networks in conjunction with LSTM (Capsule-LSTM) and acknowledged an accuracy of 86.40% on SST2 dataset. It can be noticed that Capule-LSTM model outperforms our proposed architecture by an accuracy of 0.62 percentage point on SST2 dataset. It is important to mention that although [42] slightly outperformed our model but its underlying architecture is relatively more complex compared to our proposed system.

TABLE 6. Result comparisons of SST2 dataset with other methods.

Method	Model complexity	Accuracy (%)
RNTN [41]	Fusion of parse trees and RNTN model	85.40
BiGRU+CNN [36]	Pipelining of BiGRU and CNN networks	85.40
CNN-LSTM-multitype fusion [37]	Parallel fusion of CNN and LSTM with multitype features selection	85.70
Capsule-LSTM [42]	Merging of capsule networks with LSTM model	**86.40**
Proposed	**Single-layered BiLSTM**	85.78

In summary, it is inferred that our single-layered BiLSTM based system achieved satisfactory results on SST2 dataset but could not outperform a few recently proposed models with complex architectures [37], [42], as given in Table 6.

VII. CONCLUSION

In this work, we presented a deep learning model that reduces the computational cost and runtime for the same task previously implemented with comparatively complex architectures. Specifically, we proposed a single-layered BiLSTM model with a lower number of parameters. We conclude, that when dealing with long-term dependencies, as in the three datasets about movie reviews we have been working with, we can effectively predict the sentiments by using a simple low-cost computational method. Our approach achieved competitive results and outperformed several novel methods in terms of accuracy, especially on MR and IMDB datasets. Thus, the results mentioned in this study demonstrated that it can be possible to use much simpler architecture to achieve the same level of classification performance. However, our model could not outperform a few novel frameworks on SST2, which might be the result of a random selection of test samples in the SST2 dataset. In general, we recommend our system for balanced datasets.

The future prospect of this study includes multi-class sentiment classification along with the multilingual approach. Also, the ensemble of BiLSTM and Bidirectional Gated Recurrent Unit (BiGRU) can be applied for a better classification performance. Another important direction is the automatic cleaning and classification of text in real-time applications of sentiment analysis. Lastly, it would be interesting to apply similar simpler approaches in other NLP applications such as speech recognition and machine translation.

REFERENCES

[1] G. Yoav, *Neural Network Methods for Natural Language Processing* (Synthesis Lectures on Human Language Technologies), vol. 10. San Rafael, CA, USA: Morgan & Claypool, 2017.
[2] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surveys*, vol. 34, no. 1, pp. 1–47, Mar. 2002.
[3] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proc. Int. Conf. Web Search Web Data Mining (WSDM)*, CA, USA, 2008, pp. 219–230.
[4] Z. Xiao, X. Li, L. Wang, Q. Yang, J. Du, and A. K. Sangaiah, "Using convolution control block for Chinese sentiment analysis," *J. Parallel Distrib. Comput.*, vol. 116, pp. 18–26, Jun. 2018.
[5] E. Boiy and M.-F. Moens, "A machine learning approach to sentiment analysis in multilingual Web texts," *Inf. Retr.*, vol. 12, no. 5, pp. 526–558, Oct. 2009.
[6] A. Abdi, S. M. Shamsuddin, S. Hasan, and J. Piran, "Deep learning-based sentiment classification of evaluative text based on multi-feature fusion," *Inf. Process. Manage.*, vol. 56, no. 4, pp. 1245–1259, Jul. 2019.
[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Representations (ICLR)*, AZ, USA, 2013.
[8] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543.
[9] Y. Goldberg, "A primer on neural network models for natural language processing," *J. Artif. Intell. Res.*, vol. 57, pp. 345–420, Nov. 2016.
[10] P.-Y. Huang, F. Liu, S.-R. Shiang, J. Oh, and C. Dyer, "Attention-based multimodal neural machine translation," in *Proc. 1st Conf. Mach. Transl., Shared Task Papers*, Berlin, Germany, vol. 2, 2016, pp. 639–645.
[11] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 8, no. 4, p. e1253, Jul. 2018.
[12] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha Qatar, 2014, pp. 1–6.
[13] H. Zhang, L. Xiao, Y. Wang, and Y. Jin, "A generalized recurrent neural architecture for text classification with multi-task learning," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, New York, NY, USA, Aug. 2017, pp. 1–7.
[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[15] R. Moraes, J. F. Valiati, and W. P. G. Neto, "Document-level sentiment classification: An empirical comparison between SVM and ANN," *Expert Syst. Appl.*, vol. 40, no. 2, pp. 621–633, Feb. 2013.
[16] G. Paltoglou and M. Thelwall, "More than bag-of-words: Sentence-based document representation for sentiment analysis," in *Proc. Recent Adv. Natural Lang. Process. (RANLP)*, Hissar, Bulgaria, 2013, pp. 546–552.
[17] I. Mozetič, M. Grčar, and J. Smailović, "Multilingual Twitter sentiment classification: The role of human annotators," *PLoS ONE*, vol. 11, no. 5, 2016, Art. no. e0155036.
[18] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. Empirical Methods Natural Lang. Process. (EMNLP)*, Philadelphia, PA, USA, 2002, pp. 79–86.
[19] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proc. Assoc. Comput. Linguistics (ACL)*, Jeju, South Korea, 2012, pp. 90–94.
[20] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Syst. Appl.*, vol. 57, pp. 117–126, Sep. 2016.
[21] G. Rao, W. Huang, Z. Feng, and Q. Cong, "LSTM with sentence representations for document-level sentiment classification," *Neurocomputing*, vol. 308, pp. 49–57, Sep. 2018.
[22] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522–51532, 2019.
[23] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," in *Proc. Int. Joint Conf. Natural Lang. Process.*, Taipei, Taiwan, 2017, pp. 1–18.
[24] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, TX, USA, 2016, pp. 606–615.

[25] X. Fu, J. Yang, J. Li, M. Fang, and H. Wang, "Lexicon-enhanced LSTM with attention for general sentiment analysis," *IEEE Access*, vol. 6, pp. 71884–71891, 2018.
[26] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[27] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, MI, USA, 2005, pp. 115–124.
[28] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Ng, and C. Potts, "Learning word vectors for sentiment analysis," *Assoc. Comput. Linguistics (ACL), Hum. Lang. Technol.*, OR, USA, 2011, pp. 142–150.
[29] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Empirical Methods Natural Lang. Process. (EMNLP)*, Washington, DC, USA, 2013, pp. 1631–1642.
[30] S. Sun, C. Luo, and J. Chen, "A review of natural language processing techniques for opinion mining systems," *Inf. Fusion*, vol. 36, pp. 10–25, Jul. 2017.
[31] E. Loper and S. Bird, "NLTK: The Natural Language Toolkit," in *Proc. ACL Workshop Effective Tools Methodol. Teaching Natural Lang. Process. Comput. Linguistics*, Philadelphia, PA, USA, 2002, pp. 1–8.
[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[33] T. Tieleman and G. Hinston, "Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude," *Coursera, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.
[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1958–7929, 2014.
[35] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proc. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn. (EMNLP-CoNLL)*, Jeju Island, South Korea, 2012, pp. 1201–1211.
[36] D. Zhang, L. Tian, M. Hong, F. Han, Y. Ren, and Y. Chen, "Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification," *IEEE Access*, vol. 6, pp. 73750–73759, 2018.
[37] M. Usama, W. Xiao, B. Ahmad, J. Wan, M. M. Hassan, and A. Alelaiwi, "Deep learning based weighted feature fusion approach for sentiment analysis," *IEEE Access*, vol. 7, pp. 140252–140260, 2019.
[38] Y. Long, L. Qin, R. Xiang, M. Li, and C.-R. Huang, "A cognition based attention model for sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Copenhagen, Denmark, 2017, pp. 462–471.
[39] J. Camacho-Collados and M. T. Pilehvar, "On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis," in *Proc. Empirical Methods Natural Lan. Process. (EMNLP)*, Brussels, Belgium, 2018, pp. 1–7.
[40] Y. Ma, H. Fan, and C. Zhao, "Feature-based fusion adversarial recurrent neural networks for text sentiment classification," *IEEE Access*, vol. 7, pp. 132542–132551, 2019.
[41] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Empirical Methods Natural Lang. Process. (EMNLP)*, Washington, DC, USA, 2013, pp. 1631–1642.
[42] M. Yang, W. Zhao, L. Chen, Q. Qu, Z. Zhao, and Y. Shen, "Investigating the transferring capability of capsule networks for text classification," *Neural Netw.*, vol. 118, pp. 247–261, Oct. 2019.

ZABIT HAMEED (Member, IEEE) was born in Khyber Pakhtunkhwa, Pakistan, in 1988. He received the B.Sc. degree in electrical engineering from the CECOS University of Information Technology and Emerging Sciences, Peshawar, Pakistan, in 2011, and the M.Sc. degree in electrical engineering from the Institute of Space Technology, Islamabad, Pakistan, in 2016. He is currently pursuing the Ph.D. degree in engineering with the University of Deusto, Bilbao, Spain. He is a registered Electrical Engineer with the Pakistan Engineering Council. He is also working with the eVida research group. His research interest includes machine and deep learning for time-series data and images.

BEGONYA GARCIA-ZAPIRAIN (Member, IEEE) was born in San Sebastián, Spain, in 1970. She graduated in telecommunication engineering from the University of Basque Country, Spain, in 1994. She received the Ph.D. degree in computer science and artificial intelligence from the University of Deusto, Spain, in 2004.

From 2002 to 2008, she served as the Director of the Telecommunication Department, University of Deusto, Spain, where she is working as a Full Professor, since 2011. In 2001, she started eVida research group, which is recognized by the Government of the Basque Country, Spain, and the European Network of Living Labs (ENoLL).