Intelligent Decision Technologies 16 (2022) 205–215
DOI 10.3233/IDT-210058
IOS Press

Attention-based bidirectional LSTM with embedding technique for classification of COVID-19 articles

Rakesh Dutta[a] and Mukta Majumder[b,*]
[a]*Department of Computer Science and Application, Hijli College, Kharagpur, India*
[b]*Department of Computer Science and Application, University of North Bengal, Siliguri, India*

Abstract. The epidemic of COVID-19 has thrown the planet into an awfully tricky situation putting a terrifying end to thousands of lives; the global health infrastructure continues to be in significant danger. Several machine learning techniques and pre-defined models have been demonstrated to accomplish the classification of COVID-19 articles. These delineate strategies to extract information from structured and unstructured data sources which form the article repository for physicians and researchers. Expanding the knowledge of diagnosis and treatment of COVID-19 virus is the key benefit of these researches. A multi-label Deep Learning classification model has been proposed here on the LitCovid dataset which is a collection of research articles on coronavirus. Relevant prior articles are explored to select appropriate network parameters that could promote the achievement of a stable Artificial Neural Network mechanism for COVID-19 virus-related challenges. We have noticed that the proposed classification model achieves accuracy and micro-F1 score of 75.95% and 85.2, respectively. The experimental result also indicates that the propound technique outperforms the surviving methods like BioBERT and Longformer.

Keywords: Bidirectional LSTM, attention mechanism, word embedding, text classification, COVID-19

1. Introduction

The novel coronavirus, named SARS COV-2, was first reported in early December 2019. This is an epidemic of respiratory illness called COVID-19. It is a complex illness and can emerge in several forms and severity levels risking organ failure and death [1–4]. As the pandemic progresses, cases rise, and patients experience acute repository problems, causing a large number of casualties; hence there are several reasons to be concerned about this viral outbreak [5]. These lead to an extreme urge indeed to find solutions to these COVID-19 related problems. Apart from these, the challenges remain with the humongous amount of data that researchers and medical practitioners have to deal with in fighting this pandemic. Finding a solution to the problem and guiding the research in the appropriate direction, one of the most important assignments is the automation in COVID-19 document (article) classification to speed up the research effort and search process. We have used the LitCovid dataset compiled by the National Institutes of Health (NIH) to explore how the document classification models can be useful to the scenario.

This dataset is updated on a daily basis, and new research papers are assigned and manually categorized into eight divisions: -General, Transmission Dynamics (Transmission), Treatment, Case Report, Epidemic Forecasting (Forecasting), Prevention, Mechanism, and Diagnosis. We have used this LitCovid dataset for multilevel document classification. Though this dataset consists of various biomedical specialties, it differs from general biomedical topics such as hallmark of cancer [6], chemical exposure methods [7], and diagnosis codes [8]. The collective emphasis of the LitCovid

*Corresponding author: Mukta Majumder, Department of Computer Science and Application, University of North Bengal, Siliguri, India E-mail: mukta_jgec_it_4@yahoo.co.in.

ISSN 1872-4981/$35.00

dataset upon this COVID-19 pandemic situation differentiates it from open-domain data and academic research papers classification, such as IMDB or aRxiv Academic Paper Dataset (AAPD) [9]. The LitCovid dataset is a series of 8,000 research articles (as of 10/06/2020) on the novel coronavirus imposes further challenges as these research articles do not have share topics.

Classifying a set of documents into an already defined class is a crucial task in Natural Language Processing (NLP), having application in several areas like recommender system [10], spam filtering [11], etc. Some of the well-known methods of text classification are rule-based methods such as decision tree classifier [12], statistical methods like the Bayesian classifier [13], and methods based on neural networks [14]. It is difficult to find a universal approach that will consistently work for all classes of text classification, topic classification, question classification, and sentiment analysis, etc. Most of the existing researches emphasized on the type of phrases or sentences for the classification and clustering task [15]. These methods did not consider the relationship between words for solving text classification problems. But, the classification of text should be based on all the contexts. Traditional approaches for text classification have a certain weakness. Deep learning technology [16] has been able to achieve notable results in many fields, such as speech recognition [17], and text classification [18] in recent times. Research findings on deep learning approaches are one of the two forms in text classification: (1) Neural network models with word feature vector [19] and (2) Classification using each other and to the whole learned word vectors [20]. Deep learning models are of two types: convolutional neural networks (CNNs) [21] and recurrent neural networks (RNNs) [22]. CNNs can only focus on the local responses from the spatial data but not on the succeeding correlation. On the contrary, RNNs allow sequential modelling but cannot perform a parallel way to extract the features. Text classification is a sequential modelling task. The sequential RNN models are mostly used in text classification. But traditional RNNs are even exploding with greater data strings against their gradient of vanishing state. A kind of RNN architecture, long short-term memory (LSTM) has a hidden memory unit and explains vanishing gradient and gradient explosion problems [23]. LSTM also plays a pivotal role in NLP. Bi-directional long short-term memory (BiLSTM) is a further development of LSTM, which can access preceding and succeeding contexts. At the same time, LSTM has access to only the historical context. For this reason, BiLSTM can work out the sequential problem better than LSTM. A number of achievements were made to text classifications by applying LSTM and BiLSTM [24–27]. A high-dimensional vector as input to LSTM causes a rise in the network parameter. Embedding operation extracts the required features and can reduce the dimension of the vectors. BiLSTM cannot focus on crucial information; for this, the attention mechanism can be used to highlight such info. The combination of these two can improve upon each other and enhance the classification accuracy. This article proposed a deep learning architecture for text classification, which involves BiLSTM and an attention mechanism (AM).

This research aims to provide an automated way to segregate articles into medical document repositories and research article repositories. This research also aims to a multi-class classification system based on word associative features. Till now, a relatively small amount of research works have done on multi-label classification, especially using the associative classification feature. The research also compares various classification approaches based on word frequency, N-gram features, and syntactic-semantic features.

The rest of the paper is organized as follows: Section 2 represents the literature review. Section 3 describes the preliminaries of the long short-term memory model. Section 4 illustrates the proposed attention mechanism based BiLSTM with word embedding. Section 5 describes the experimental set up. The result and discussion are presented in Section 6. And the conclusion is drawn in Section 7.

2. Literature review

LSTM is explicitly used to handle sequential data in deep learning. In recent times, the range of implementations for LSTM has grown exponentially. Many applications were solved by LSTM and its growing versions to produce desired outcomes. A significant area of study is the integration of LSTM and other architectures of neural networks.

The succession of LSTM with the attention layer can achieve more reliable results. The attention mechanism is beneficial in the classification problem, sentiment analysis, question answering, etc., mainly for sequential context. Lezoray and Cardot suggested a neural network architecture to classify data spread among a high number of groups [28]. A hierarchical attention-based network was proposed by Yang et al. for six large-scale

classification problems [29]. The model consisted of two components, first to reflect the hierarchical composition of texts. Second, the hierarchical model used two attention layers for document representation, one for word level and the other for sentence level. Li et al. revealed a novel model applying a word-level attention mechanism for statement representations and sentence-level attention mechanisms for more significant context modelling [30]. Paulus et al. presented a novel intra-attention layer in the neural network model that serves across the input and continuously produced output individually [31]. It was a unique training approach that merged reinforcement learning (RL) and supervised learning. A distinct neural network system called FusionNet was proposed by Huang et al. that extended attention mechanisms to focus on three aspects [32]. First, it implemented a forward hidden attention layer to extract the context's history using word embedding, and then it proposed an enhanced attention method that more usefully employed the "history of word" idea. Next, it suggested a multi-level attention layer to achieve the answer in one text (such as a question). The Bi-Directional Attention Flow (BIDAF) mechanism was introduced by Seo et al. [33]. It was a multi-stage hierarchical method that used a bi-directional attention layer to get the knowledge from context representation. Daniluk et al. suggested a language model with an attention mechanism that distributed representation for word, differentiable weight for memory, and encoded the next-word relationship [34]. A shallow architecture based neural network for natural language prediction was suggested in the literature [35]. It obtained excellent results in some parameters on the Stanford Natural Language Inference (SNLI) dataset.

Luo recommended a deep learning model based on LSTM and word embedding for clinical text classification [36]. Hu et al. introduced an LSTM model based on the context keywords, which was fine-tuned on vocabulary words. Their model obtained better accuracy than the baseline LSTM and other machine learning semantic models [37]. Huang et al. determined a part-of-speech (POS) tagging-based LSTM network to enhance the sentence representation [38]. An enhanced version of LSTM called RvNNs to represent the context's compositional semantics was found in the literature [39]. The model used sentence correlation and semantic composition to increase classification accuracy. Tang et al. proposed a model based on the neural network to learn the document representation feature vector in a bottom-up fashion [40]. The model first learned sentence representation with the convolutional neural network. Afterward, the semantics of sentences and their relations were adaptively encoded in document representation with a gated recurrent neural network.

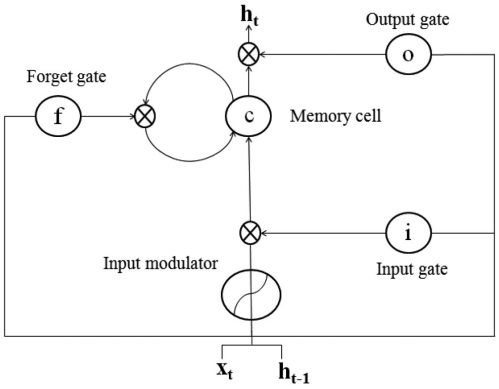


Fig. 1. Illustrate the four units of LSTM.

3. Long short-term memory

A recurrent feed-forward neural network that has a hidden state is called RNNs. The hiding state is triggered at any period of time by the preceding states. RNNs can manage variable-vector-length sequences and dynamically sculpt the contextual information. Long short-term memory is an artificial recurrent neural network (RNN), can solve the issue of the vanishing gradient by eliminating the self-connected hidden layers from memory cells. The storage block uses specially designed memory cells to store information; discovering and leveraging long-range meaning is more accessible.

The memory unit helps the network to know when and how to acquire new knowledge and overlook previous information. LSTM units are composed of four factors as shown in Fig. 1; input gate (i), forget gate (f), output gate (o), and cell activation vector (c) composed of partly lost past c_{t-1} memory and modulated current ($\tilde{ct}$) memory. t specifies the time of *t-th* moment.

LSTM calculates the deeply hidden unit h_t, given input x_t as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i). \tag{1}$$

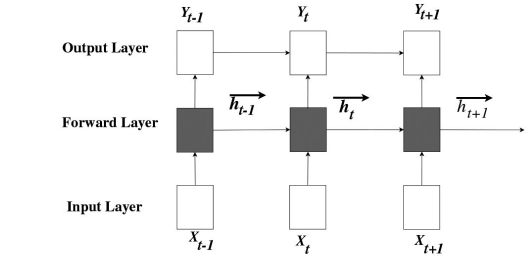
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f). \tag{2}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o). \tag{3}$$

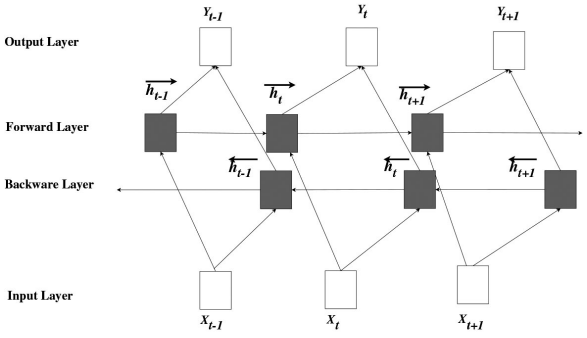
$$\widetilde{c_t} = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c). \tag{4}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \widetilde{c_t}. \tag{5}$$

$$h_t = o_t \otimes \tanh(c_t). \tag{6}$$



(a) Illustration of a LSTM model.



(b) Illustration of a BiLSTM model.

Fig. 2. Illustration of LSTM and BiLSTM model.

Where i_t, f_t, o_t and c_t represent the value of i, f, o and c at the moment t, respectively. W signifies the self-updating weights of the hidden layer and b represents the vector of a bias. $\sigma(.)$ represents the sigmoid function and $\tanh(.)$ signifies hyperbolic tangent function. The output for all gates and hidden states are between the range [0–1]. The operator $\otimes$ signifies multiplication with each element in the state.

Graphical representation of a regular LSTM system can be found in Fig. 2a. Only the historical meaning can be exploited by a regular LSTM system. However, the incomplete understanding of the contextual meaning occurs for the lack of knowledge in further context. The integration of a forward hidden layer and a backward hidden layer of BiLSTM are shown in Fig. 2b. BiLSTM understands the contextual meaning of both the preceding and succeeding direction. This system has developed using backpropagation [41].

4. Attention mechanism based BiLSTM with word embedding feature vector

The proposed technique introduces a unique architecture by adding BiLSTM with word embedding and attention layers. The suggested architecture is termed as attention-based BiLSTM (A-BiLSTM). The embedding layer in A-BiLSTM extracts semantic features for sentences from the corpus. And then, we integrate a forward hidden layer and a backward hidden layer to access all the previous and successive context information.

The attention mechanism (AM) for the single word representation evokes more interest in the terms that contribute to the meaning of the context and can help to clarify the semantic of sentences. In A-BiLSTM, two attention layers process the previous and successive contextual characteristics, respectively. These characteristics are concatenated together in AM and served into the classifier softmax. The A-BiLSTM model architecture is shown in Fig. 3.

4.1. Word embedding

One-Hot Encoding method represents the context word in the vector space, though this method suffers from two downsides; the order of the word is not correctly maintained and the dimension of the vector is too high. One-to-one word embedding is more reliable and efficient compared to One-Hot Encoding. Consider N is the number of total words present in the text, the vector representation of m-th word in the text is wr_m, where m belongs to [1–N]. The embedding vector of each word in the BiLSTM is W_e. Equation (7) articulates the embedding vector representation of each word X_m.

$$X_m = W_e wr_m. \tag{7}$$

The word embedding is used in many different applications in Natural Language Processing (NLP). The proposed method uses the word2vec embedding framework recommended by Mikolov et al. [42]. Word embedding is a technique for generating word vectors using the skip-gram and the continuous bag-of-words (CBOW) model. These models optimize the word vector space while learning.

4.2. BiLSTM with attention mechanism

Text classification, intuitively, is the retrieval of sequential knowledge. The feature sequence obtains from the embedding layer contains sequential information. For sequential modelling, BiLSTM is versatile and can further derive descriptive information from the functional sequences provided by the embedding layer. The purpose of BiLSTM is to construct a text-level vector representation of the terms (words). Different words have different forms of meaning, assigning different weights to the terms to maintain the context's sequence

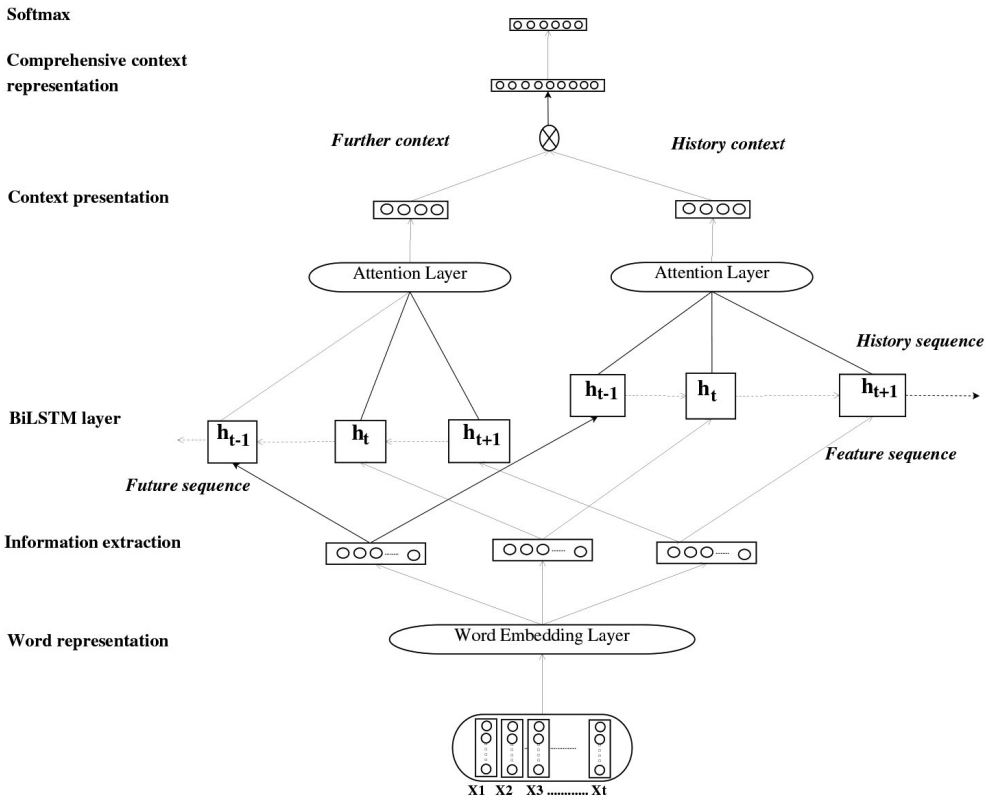


Fig. 3. The architecture of the A-BiLSTM.

information and the entire text's sentiment. The method of emphasis is to assign various weights to the terms to increase the comprehension of the whole text under consideration. Hence, BiLSTM and the attention mechanism will certainly increase the efficiency of classification. BiLSTM gets word annotations by summarizing information of both forward and backward directions, and thus annotations integrate contextual information. BiLSTM comprises forward LSTM (symbolized as $\overrightarrow{LSTM}$) that reads the feature sequences from Lc_1 to Lc_{100} and backward LSTM (symbolized as $\overleftarrow{LSTM}$) which reads from Lc_{100} to Lc_1. Formally, the BiLSTM outputs are detailed as follows:

$$\overrightarrow{h_f} = \overrightarrow{LSTM}(Lc_n), n\varepsilon[1, 100]. \tag{8}$$

$$\overleftarrow{h_b} = \overleftarrow{LSTM}(Lc_n), n\varepsilon[100, 1]. \tag{9}$$

An annotation is achieved by the forward hidden unit ($\overrightarrow{h_f}$) and the backward hidden unit ($\overleftarrow{h_b}$) for a given sequence of features Lc_n. These units describe the contextual information of the entire text and execute the word embedding. In order, to minimize the influence of non-keywords, the attention mechanism concentrates on the features of the keywords and it is considered as a fully-connected layer and a softmax classifier. The working principle of the attention mechanism in A-BiLSTM is detailed below.

The term annotation $\overrightarrow{h_f}$ is used to find $\overrightarrow{u_f}$ by single-layer perceptron as an intermediate hidden representation of $\overrightarrow{h_f}$. $\overrightarrow{u_f}$ is calculated as follows:

$$\overrightarrow{u_f} = \tanh(w\overrightarrow{h_f} + b). \tag{10}$$

Where weight and bias in the neuron are expressed as w and b, $\tanh(.)$ is a function of a hyperbolic tangent. To calculate the importance of each term, the model uses $\overrightarrow{u_f}$ and a word or term level context vector $\overrightarrow{v_f}$. And then it uses the softmax function to get the normalized weight $\overrightarrow{a_f}$ of any term. It is formulated as follows:

$$\overrightarrow{a_f} = \frac{\exp(\overrightarrow{u_f} \times \overrightarrow{v_f})}{\sum_{i=1}^{N}(\exp(\overrightarrow{u_f} \times \overrightarrow{v_f}))}. \tag{11}$$

Where N is the quantity of terms in the text and $\exp(.)$ is the method of exponential. The word-level contextual vector $\overrightarrow{v_f}$ can be interpreted as a high-level expression of the descriptors over the words and is initialized at random and studied together during the process of training.

Subsequently, a weighted quantity of term annotations $\overrightarrow{a_f}$ is used to represent forward context F_c. The F_c is a component of the attention layer outcome and expressed as follows:

$$F_c = \sum(\overrightarrow{a_f} \times \overrightarrow{h_f}). \tag{12}$$

And $\overleftarrow{a_b}$ is used to measure the hidden backward unit $\overleftarrow{h_b}$. Similarly, the contextual meaning of the hidden backward representation H_c is the part of the attention layer outcome and expressed as:

$$H_c = \sum(\overleftarrow{a_b} \times \overleftarrow{h_b}). \tag{13}$$

By integrating the forward contextual meaning F_c and the backward contextual meaning H_c, A-BiLSTM acquires annotations for a given sequence of features Lc_n.

Finally, $S = [F_c, H_c]$ is obtained as a detailed contextual representation. The complete representations of the context meaning are considered to be the text classification characteristics. The dropout layer and the softmax classification layer are used in A-BiLSTM to generate the probability distribution to accomplish classification. The objective of the dropout layer is to avoid overfitting. In order, to test the classification efficiency of the models, cross-entropy (widely used loss function) is currently used. It is more beneficial than the mean square error method. Adam optimizer is selected in our approach to minimize the loss function of the network. Adam optimizer has been demonstrated as an efficient and effective back propagation algorithm that fine-tunes the model parameters [43]. In the stochastic gradient descent method, the cross-entropy as the loss function will reduce the probability of a gradient disappearance. The loss function can be designated as follows in Eq. (14).

$$L_{total} = -\frac{1}{num}\sum_{Sp} [y \ln o + (1-y)\ln(1-o)]. \tag{14}$$

Where *num* is the number of data points for training, Sp defines the sample of training, y is the sample label, and o is the A-BiLSTM outcomes.

The overall learning technique (A-BiLSTM) is outlined as Procedure 1, where $\bigoplus$ denotes the concatenation between the further context and the historical context.

Procedure 1: A-BiLSTM
Input:
D: a collection of LitCovid (COVID-19) dataset, $D = D_t \cup D_e$, where D_t is the training dataset and D_e is the testing dataset.
C: a collection of labelled data classes $C = \{C_1, C_2, C_3, \ldots, C_6\}$
Output:
L: a collection of possible class names assign to D_e.

Step1: Embedding layer is used to create feature vectors of D_t with Eq. (7);
Step2: Apply BiLSTM to access current and past contextual features $\overrightarrow{h_f}$ and $\overleftarrow{h_b}$ from the feature vectors, using Eqs (8) and (9);
Step3: Two attention layers are applied to gain further and historical context representation F_c and H_c from the forward and backward contextual features, using Eqs (12) and (13);
Step4: Integrate the further and historical context representations to achieve detailed context representations ($S = [F_c, H_c]$).
Step5: The comprehensive context representation is fed into the softmax classifier to get the corresponding class names (C).
Step6: The loss function is used to fine-tune the model parameters, using Eq. (14).
Step7: Repeat step 2–6 to learn or train the proposed model using D_t.
Step8: Testing is performed on D_e to predict the possible classes ($L_{c_{temp}}$).

The key achievements and uniqueness of A-BiLSTM are as follows:

1. The distributional word embedding layer is used for reducing the dimensional space. It separates semantic low-level features from the raw text.
2. BiLSTM derives contextual information from the semantic characteristics at the low-level. It can access both the preceding and succeeding contextual information directly from the text.
3. In BiLSTM, the forward hidden layer and backward hidden layer use their respective attention mechanism. The layers of the attention mechanism in A-BiLSTM allow interpretation of text semantics vectors in more detail.

5. Experiments

Experiments are conducted to determine the efficiency of the proposed text classification approach on the COVID-19 dataset (LitCovid).

5.1. LitCovid dataset

The LitCovid dataset[1] is a list of recently released PubMed papers that are specifically linked to the novel Coronavirus. It includes upwards of 14,000 research papers, and about 2,000 new articles are published per week [44]. This dataset is a rich resource to keep researchers up to date with the latest crisis of COVID-19. We have taken 8,002 articles from the originally 14,000 plus papers for our classification task.

In the LitCovid dataset eight topics (classes) are identified, these are Prevention, Treatment, Diagnosis, Mechanism, Case Report, Transmission, Forecasting, and General. In this pandemic situation, it is essential to highlight and classify the most crucial classes for taking the necessary precautionary measures and knowledge transfer. Hence, we have given importance to the Covid related classes and excluded General and Forecasting for our consideration. We split the LitCovid dataset into, training and testing with the ratio 8:2 and 80789 tokens are considered as our vocabulary size $|V|$.

5.2. Argument settings

Our study provides micro-F1, as μF1 score to obtain the average efficiency of the multi-label document classification model. μF1 score helps to compare the performance of the proposed model with the other classification models prominently. The input sequence X_m is used to define *m-th* word embedding. It is the distributed representation of words in an input sentence during A-BiLSTM training [45]. The scale of the word embedding is 100. The BiLSTM memory dimension is set to 128. For the training datasets, the batch size is set as 50. The withdrawal rate for the dropout layer is 0.5. The back propagation algorithm and Adam's stochastic optimization process are used to train the network over time. After completion of each training iteration, the network is tested on validation data.

5.3. Feature vector

The most crucial aspect of document or article classification is to transform the text into a vector containing the word frequencies, grammatical feature, syntactic feature, n-gram feature, and semantically associative classification features. These feature vectors can be produced in many ways [46]. To determine which features provide the best accuracy for our classification problem, we have analyzed multiple techniques for the same as discussed below.

Bag of words: The Bag of Words model is applied to represent the text by transforming it into a bag of words (BOW), which keeps counting the total occurrences of each word in a vocabulary. It is a way of extracting features from the text. The vocabulary includes words, word sequences (token n-grams), or letter sequences (character n-grams) [47]. In this feature, each vocabulary word is represented with a numerical value.

Term Frequency-Inverse Document Frequency: (TF-IDF): TF-IDF is still the most basic model to represent the documents. The tf_{ij} parameter is defined as the number of times the term i appears in document j. The greater value of tf_{ij} means the term i is more significant. The df_i parameter is the number of the document in which the term i occurs. The higher value of i occurs more frequently. If term i can be recognized as important for document j, it should have a large TF (tf_{ij}) and a small IDF (df_i). Here, TF-IDF is defined by Eq. (15).

$$\textit{TF-IDF}_{ij} = tf_{ij} \times \left(\frac{N}{df_i + 1} \right). \tag{15}$$

N-grams: An extension of the bag of words is the N-grams. An N-gram is a contiguous sequence of n-terms in a corpus.

Word2vec: Word2Vec is a shallow neural network model of two layers used to learn term associations from a large context. It takes a large corpus as input and creates a set of vectors. The term vectors are arranged such a way that they share similar definitions or semantically equivalent in the corpus and close to each other in the vector space. Word2Vec is a predictive model for learning word embeddings from raw text.

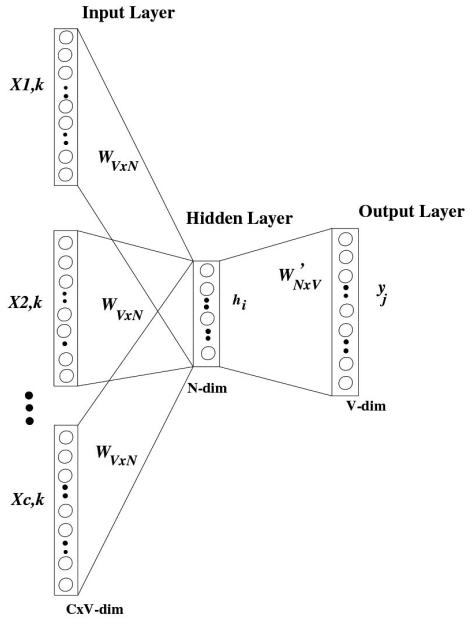
It can be implemented as, Continuous Bag-of-Words (CBOW) architecture, shown in Fig. 4a, and Skip-Gram architecture, shown in Fig. 4b.

6. Results and discussions

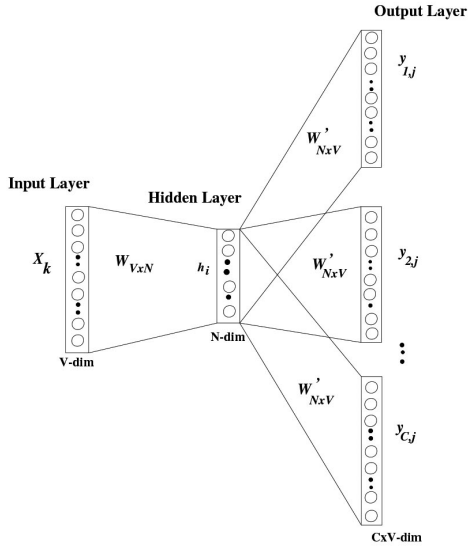
In this section, we have evaluated the efficiency of our proposed method and compare it with the existing systems on the same dataset.

During a severe viral outbreak such as this pandemic, the classification models must deliver accurate results for attaining necessary precautionary measures. Since biomedical article tagging is a very time-consuming process, so, automatically tagging and achieving maxi-

[1] https://github.com/dki-lab/covid19-classification.



(a) Continuous Bag-of-Words (CBOW) model.



(b) Skip-Gram model.

Fig. 4. Architecture of Word2vec model.

mum accurate output is essential in this situation. We have designed a scheme based on BiLSTM and attention layers with different features like n-grams, Bag-of-word, tf-idf, and word2vec. Subsequently, we have found the accuracy and μF1 score for each classification scheme. The assessed result for each model with different feature dimensions is depicted in Table 1.

In Table 1, we have depicted that the BiLSTM includes various features namely: 2-gram, 3-gram, Bag-of-word, tf-idf, and word2vec, and obtains the μF1 scores 73.1, 73.9, 72.2, 73.1, and 80.2, respectively. Furthermore, we have observed a remarkable μF1 score when the attention layers are included in BiLSTM (namely A-BiLSTM) with word2vec. The μF1 scores for A-BiLSTM with the features tf-idf, and word2vec are 74.51, and 85.25, respectively.

Table 1
Experimental results on LitCovid dataset

Model name	Feature name	Accuracy (%)	μF1 score
Bi-LSTM	W2V(100)	69.25	80.2
Bi-LSTM + attention	W2V(100)	75.95	85.2
Bi-LSTM	TF-IDF (5000)	51.76	73.1
Bi-LSTM + attention	TF-IDF (5000)	56.34	74.5
Bi-LSTM	BOW (5000)	51.44	72.2
Bi-LSTM	2-gram	56.86	73.9
Bi-LSTM	3-gram	53.66	73.1

We have shown a comparative study of our proposed model A-BiLSTM with the models (BioBERT and Longformer) presented by Gutierrez et al. in Table 2 [44].

BiLSTM can view both the previous and subsequent qualitative information in contrast to LSTM. Therefore it can recognize the meaning of each word in the text more efficiently. The emphasized method is specifically used to describe the effect of every word of the sentences.

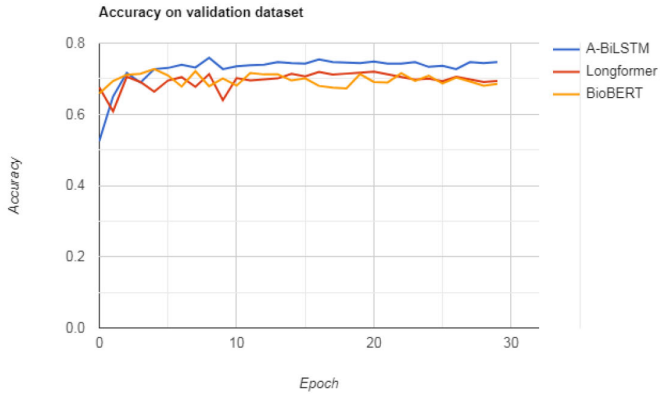
In our classification model, the input sequence X_m is used to define *m-th* word embedding. It is the distributed representation of words in an input sentence during A-BiLSTM training. The scale of the word embedding is 100 whereas the other two models Longformer and BioBERT used embedding with size 512. In our training the batch size is set as 50, on the other hand, Longformer and BioBERT used batch size 6 and 5 respectively. In our case, the withdrawal rate for the dropout layer is 0.5 whereas in the other two models the withdrawal rates were 0.2 and 0.1 respectively.

The back propagation algorithm and Adam's stochastic optimizer are used to train the network over time. The Longformer and BioBERT used Adam optimizer with a linear warmup (1000 steps) and linear decay (*AdamW*). The learning rate of our proposed model is 0.001 while, in the other two models, it was 2×10^{-5}. The number of trainable parameters of our proposed model (*A-BiLSTM*), Longformer, and BioBERT are 19M, 148M, and 108M respectively.

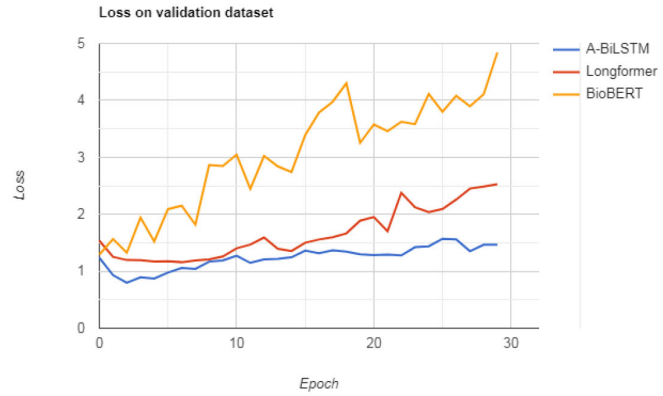
From Table 2, we have observed that our proposed model A-BiLSTM achieved higher accuracy and micro-F1 score than the surviving models.

Table 2
Comparative study of the proposed model with existing models on LitCovid dataset

Model name	Parameter selection	Learning rate	Number of parameters	Accuracy (%)	μF1 score
Bi-LSTM + attention (proposed model)	Embedding: 100 Batch size: 50 Dropout: 0.5 Optimizer: Adam stochastic Activation function: "softmax"	0.001	$\sim$ 19M	75.95	85.2
Longformer	Embedding: 512 Batch size: 6 Dropout: 0.2 Optimizer: AdamW Activation function: "GeLU"	2×10^{-5}	$\sim$ 148M	69.20	80.7
BioBERT	Embedding: 512 Batch size: 3 Dropout: 0.1 Optimizer: AdamW Activation function: "GeLU"	2×10^{-5}	$\sim$ 108M	68.50	81.2



(a) Comparative study of A-BiLSTM, Longformer, and BioBERT in terms of accuracy on validation dataset.



(b) Comparison of A-BiLSTM, Longformer, and BioBERT in terms of loss on validation dataset.

Fig. 5. Comparison of proposed technique with existing techniques.

A comparative study of A-BiLSTM, Longformer, and BioBERT in terms of accuracy and loss function on the validation dataset are shown in Fig. 5a and 5b respectively.

7. Conclusion

This paper introduces a deep learning based multi-label classification system, called A-BiLSTM, for COVID-19 article classification. The embedding layer, BiLSTM, and attention mechanism with different identified candidate features are used to enhance the performance of the classifier. By efficiently classifying and categorizing the relevant documents (research articles) into appropriate classes the proposed system can help to deal with the crisis of novel Coronavirus and direct the research into a proper direction so that treatment and diagnosis of COVID-19 virus can be speed up and necessary precautionary measure can be taken. To test the efficiency of our proposed method, experiments have been carried out on the LitCovid dataset which is a collection of research articles related to the novel Coronavirus, where articles are labelled with different classes. The comparative study reveals that the proposed technique outperforms the state-of-the-art surviving approaches.

References

[1] Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, et al. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. The Lancet. 2020; 395(10223): 497–506.

[2] Chen N, Zhou M, Dong X, Qu J, Gong F, Han Y, et al. Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: A descriptive study. The Lancet. 2020; 395(10223): 507–513.

[3] Wang D, Hu B, Hu C, Zhu F, Liu X, Zhang J, et al. Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus-infected pneumonia in Wuhan, China. Jama. 2020; 323(11): 1061–1069.

[4] Liu K, Fang YY, Deng Y, Liu W, Wang MF, Ma JP, et al. Clinical characteristics of novel coronavirus cases in tertiary hospitals in Hubei Province. Chinese medical journal. 2020.

[5] Guo T, Fan Y, Chen M, Wu X, Zhang L, He T, et al. Cardiovascular implications of fatal outcomes of patients with coronavirus disease 2019 (COVID-19). JAMA Cardiology. 2020; 5(7): 811–818.

[6] Baker S, Silins I, Guo Y, Ali I, Högberg J, Stenius U, et al. Automatic semantic classification of scientific literature according to the hallmarks of cancer. Bioinformatics. 2016; 32(3): 432–440.

[7] Larsson K, Baker S, Silins I, Guo Y, Stenius U, Korhonen A, et al. Text mining for improved exposure assessment. PloS One. 2017; 12(3): e0173132.

[8] Du J, Chen Q, Peng Y, Xiang Y, Tao C, Lu Z. ML-Net: Multi-label classification of biomedical texts with deep neural networks. Journal of the American Medical Informatics Association. 2019; 26(11): 1279–1285.

[9] Yang P, Sun X, Li W, Ma S, Wu W, Wang H. SGM: sequence generation model for multi-label classification. arXiv preprint arXiv:180604822. 2018.

[10] Watanabe A, Sasano R, Takamura H, Okumura M. Generating personalized snippets for web page recommender systems. In: 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). IEEE; Vol. 2, 2014. pp. 218–225.

[11] Almeida TA, Silva TP, Santos I, Hidalgo JMG. Text normalization and semantic indexing to enhance instant messaging and SMS spam filtering. Knowledge-Based Systems. 2016; 108: 25–32.

[12] Phu VN, Tran VTN, Chau VTN, Dat ND, Duy KLD. A decision tree using ID3 algorithm for English semantic analysis. International Journal of Speech Technology. 2017; 20(3): 593–613.

[13] Lee LH, Isa D, Choo WO, Chue WY. High Relevance Keyword Extraction facility for Bayesian text classification on different domains of varying characteristic. Expert Systems with Applications. 2012; 39(1): 1147–1155.

[14] Jingsheng L, Ting J. Hierarchical text classification based on bp neural network. Journal of Computational Information Systems. 2009; 5: 581–590.

[15] Anoop V, Asharaf S. Conceptualized phrase clustering with distributed k-means. Intelligent Decision Technologies. 2019; 13(2): 153–160.

[16] Schmidhuber J. Deep learning in neural networks: An overview. Neural Networks. 2015; 61: 85–117.

[17] Marasek K, et al. Deep belief neural networks and bidirectional long-short term memory hybrid for speech recognition. Archives of Acoustics. 2015; 40(2): 191–195.

[18] Tai KS, Socher R, Manning CD. Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:150300075. 2015.

[19] Pang B, Lee L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. arXiv preprint cs/0506075. 2005.

[20] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. Journal of Machine Learning Research. 2011; 12(ARTICLE): 2493–2537.

[21] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM. 2017; 60(6): 84–90.

[22] Funahashi Ki, Nakamura Y. Approximation of dynamical systems by continuous time recurrent neural networks. Neural Networks. 1993; 6(6): 801–806.

[23] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation. 1997; 9(8): 1735–1780.

[24] Liu W, Liu P, Yang Y, Gao Y, Yi J. An Attention-Based Syntax-Tree and Tree-LSTM Model for Sentence Summarization. International Journal of Performability Engineering. 2017; 13(5).

[25] Nowak J, Taspinar A, Scherer R. LSTM recurrent neural networks for short text and sentiment classification. In: International Conference on Artificial Intelligence and Soft Computing. Springer. 2017. pp. 553–562.

[26] Chen T, Xu R, He Y, Wang X. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. Expert Systems with Applications. 2017; 72: 221–230.

[27] Niu X, Hou Y, Wang P. Bi-directional LSTM with quantum

attention mechanism for sentence modeling. In: International Conference on Neural Information Processing. Springer. 2017. pp. 178–188.

[28] Lezoray O, Cardot H. A neural network architecture for data classification. International Journal of Neural Systems. 2001; 11(01): 33–42.

[29] Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016. pp. 1480–1489.

[30] Li H, Min MR, Ge Y, Kadav A. A context-aware attention network for interactive question answering. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017. pp. 927–935.

[31] Paulus R, Xiong C, Socher R. A deep reinforced model for abstractive summarization. arXiv preprint arXiv:170504304. 2017.

[32] Huang HY, Zhu C, Shen Y, Chen W. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. arXiv preprint arXiv:171107341. 2017.

[33] Seo M, Kembhavi A, Farhadi A, Hajishirzi H. Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:161101603. 2016.

[34] Daniluk M, Rocktäschel T, Welbl J, Riedel S. Frustratingly short attention spans in neural language modeling. arXiv preprint arXiv:170204521. 2017.

[35] Parikh AP, Täckström O, Das D, Uszkoreit J. A decomposable attention model for natural language inference. arXiv preprint arXiv:160601933. 2016.

[36] Luo Y. Recurrent neural networks for classifying relations in clinical notes. Journal of Biomedical Informatics. 2017; 72: 85–95.

[37] Hu F, Li L, Zhang ZL, Wang JY, Xu XF. Emphasizing essential words for sentiment classification based on recurrent neural networks. Journal of Computer Science and Technology. 2017; 32(4): 785–795.

[38] Huang M, Qian Q, Zhu X. Encoding syntactic knowledge in neural networks for sentiment classification. ACM Transactions on Information Systems (TOIS). 2017; 35(3): 1–27.

[39] Wu D, Chi M. Long short-term memory with quadratic connections in recursive neural networks for representing compositional semantics. IEEE Access. 2017; 5: 16077–16083.

[40] Tang D, Qin B, Liu T. Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015. pp. 1422–1432.

[41] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks. 2005; 18(5–6): 602–610.

[42] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781. 2013.

[43] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014.

[44] Gutierrez BJ, Zeng J, Zhang D, Zhang P, Su Y. Document classification for covid-19 literature. arXiv preprint arXiv: 200613816. 2020.

[45] Bengio Y, Ducharme R, Vincent P, Janvin C. A neural probabilistic language model. The Journal of Machine Learning Research. 2003; 3: 1137–1155.

[46] Kulkarni A, Shivananda A. Converting text to features. In: Natural Language Processing Recipes. Springer. 2019. pp. 67–96.

[47] Wieting J, Bansal M, Gimpel K, Livescu K. Charagram: Embedding words and sentences via character n-grams. arXiv preprint arXiv:160702789. 2016.