Research Article

Performance Analysis of Hybrid Deep Learning Models with Attention Mechanism Positioning and Focal Loss for Text Classification

Sunil Kumar Prabhakar ¹, Harikumar Rajaguru,² and Dong-Ok Won ³

¹Department of Artificial Intelligence, Korea University, Anam-dong, Seongbuk-gu, Seoul 02841, Republic of Korea

²Department of ECE, Bannari Amman Institute of Technology, Sathyamangalam 638401, India

³Department of Artificial Intelligence Convergence, Hallym University, Chuncheon 24252, Gangwon, Republic of Korea

Correspondence should be addressed to Dong-Ok Won; dongok.won@hallym.ac.kr

Received 1 August 2021; Accepted 5 October 2021; Published 26 October 2021

Academic Editor: Sikandar Ali

Copyright © 2021 Sunil Kumar Prabhakar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Over the past few decades, text classification problems have been widely utilized in many real time applications. Leveraging the text classification methods by means of developing new applications in the field of text mining and Natural Language Processing (NLP) is very important. In order to accurately classify tasks in many applications, a deeper insight into deep learning methods is required as there is an exponential growth in the number of complex documents. The success of any deep learning algorithm depends on its capacity to understand the nonlinear relationships of the complex models within data. Thus, a huge challenge for researchers lies in the development of suitable techniques, architectures, and models for text classification. In this paper, hybrid deep learning models, with an emphasis on positioning of attention mechanism analysis, are considered and analyzed well for text classification. The first hybrid model proposed is called convolutional Bidirectional Long Short-Term Memory (Bi-LSTM) with attention mechanism and output (CBAO) model, and the second hybrid model is called convolutional attention mechanism with Bi-LSTM and output (CABO) model. In the first hybrid model, the attention mechanism is placed after the Bi-LSTM, and then the output Softmax layer is constructed. In the second hybrid model, the attention mechanism is placed after convolutional layer and followed by Bi-LSTM and the output Softmax layer. The proposed hybrid models are tested on three datasets, and the results show that when the proposed CBAO model is implemented for IMDB dataset, a high classification accuracy of 92.72% is obtained and when the proposed CABO model is implemented on the same dataset, a high classification accuracy of 90.51% is obtained.

1. Introduction

A representation topic in NLP analysis is text classification. For managing tremendous amount of text documents in the fields of web mining, information retrieval, and NLP, text classification plays a vital role [1]. The knowledge gained from text expression is employed in an efficient manner; thus, the assignment of one or more predefined topics to a natural text document is done easily in text classification. Versatile machine learning techniques have been utilized for the purpose of text classification such as Bayesian techniques [2], support vector machine [3], K-nearest neighbor (KNN) [4], neural networks [5], and hidden Markov model (HMM)

[6]. In recent years, a dramatic improvement is shown by the deep learning techniques for text classification purposes. A good performance is achieved by Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) when implemented for medical image recognition, face recognition, voice recognition, etc. [7]. In order to synthesize images and voice successfully, generative models like Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN) are utilized successfully and implemented in the entertainment industry [8]. Similarly, deep learning has been successfully implemented in NLP fields too. For analyzing and representing the human languages in an automatic manner, NLP seems to be of great utility. CNNs

and RNNs have caught huge attention in implementing NLP tasks such as text classification, sentiment analysis, and text summarization [9]. In recent years, CNN has attracted huge attention from the researchers as it shows a good performance in text classification. When CNNs are exploited for text classification, the embedding of text in multidimensional vector space is the most important task. In text classification, word embedding is constantly implemented, where word2vec dependent on skip-gram model is the most commonly used technique [10]. An improved RNN architecture is LSTM, where a gating mechanism which consists of an input gate, forget gate, and output gate is utilized [11]. The determination of data in the previous state whether to retain it or to forget it is explained by these gates. The problem of long-term data conservation along with the vanishing gradient issue is solved by the gating mechanism efficiently. The super ability of LSTM to extract text information in a versatile manner plays a significant part in text classification. In recent years, the utility of LSTM has been explored to a great extent, and researchers are always modifying or revamping LSTMs so that their accuracy can be further improved.

Some of the recent works involving deep learning for efficient text classification are stated as follows. The simplest deep learning models utilized for text representation are feedforward neural networks [12]. Deep Average Network (DAN) is an example of this model, and due to its simplicity, it outperforms other models easily [13]. Text is viewed as a sequence of words in RNN based models. Under the category of RNN based models, a tree-LSTM model was developed by Tai et al. [14], and an extended chain structured LSTM was developed by Zhu et al. [15]. To capture the local structure of a word sequence, RNNs are quite versatile, but when remembering long-range dependencies, they face a lot of difficulties. The patterns across the space are recognized by the CNNs when dealing with text classification. A simple CNN based model for text classification was proposed by Kim [16], character level CNN was proposed by Zhang et al. [17], and text encoding by CNN was proposed by Prusa and Khoshgoftaar [18]. In order to overcome the problem of pooling caused by CNN, capsule networks are utilized for text classification. The representation of sentences as a vector is adapted by capsules in capsule neural networks [19]. Other works involving capsule neural networks for classification are discussed in [20–22]. The models with attention mechanism for text classification were again proposed by many researchers. A hierarchical attention network for text classification was proposed by Yang et al. [23]. The hierarchical attention model was extended to cross lingual sentiment classification by Zhou et al. [24]. A directional self-attention network for RNN/CNN was presented by Shen et al. [25], and a LSTM model with inner attention was presented by Liu et al. [26]. To extract the sentence embeddings in a predictable manner, self-attention was used by Lin et al. [27]. A neural semantic encoder (NSE) which is nothing but a memory-augmented neural network was proposed by Munkhdalai and Yu [28]. To process the input questions and to generate their respective answers, a dynamic memory network was proposed by Kumar et al. [29].

Graph neural network models for text classification were also utilized by Peng et al.; they convert the text to graph-of-words format [30]. A graph convolutional network (GCN) and its respective variants were utilized in [31]. A graph CNN which explored the document-word relations and the cooccurrence of words was built in [32]. For the text matching applications, Siamese Neural Networks (SNNs) were used widely [33]. Transformers were also utilized for text classification as they allow for more parallelization than other deep learning models, so that very huge models can be trained quite efficiently [34].

Hybrid models were also developed; they combine the usage of LSTM and CNN, so that the local and global features of the words and sentences can be captured well [35]. To combine the unique merits of LSTM, CRF, and CNN and to address the individual weaknesses, a Bi-LSTM-CRF-CNN hybrid model was also proposed in [36]. A convolutional LSTM (C-LSTM) was proposed by Zhou et al., where a sequence of higher-level phrases is extracted by CNN and then passed on a LSTM network so that the sentence representation is obtained [37]. A CNN-RNN model was proposed by Tang et al., where a CNN is utilized to learn the representation of sentences and a gated RNN is used to learn a document representation [38]. For document modeling, a hierarchical model called dependency sensitive D-SCNN was proposed by Zhang et al. [39]. A hybrid deep learning architecture called HDLTex which utilizes Multi-layer Perceptron (MLP), RNN, and CNN was utilized by Kowsari et al. so that the count hierarchy can be well understood for easy text classification [40]. Based on word2vec's Continuous Bag of Words (CBOW) design, a Bi-LSTM neural network architecture was designed by Melamud et al. where, for the variable length sentence contexts, the general context embedding function was learned very efficiently [41]. A famous text classification model dependent on word2vec and LSTM was proposed by Xiao et al. so that the text can be classified efficiently in the security field, where, in order to overcome the high dimensionality, a pretrained word2vec model was used [42]. A hybrid model utilizing CNN and LSTM with the effective use of normalization techniques, dropout techniques, and rectified linear units was proposed by Rehman et al., and the results show a high accuracy and precision rate [43]. A hybrid approach was proposed by She and Zhang where the local features were extracted with the help of CNN and the weakness of the LSTM was tackled well by this model [44]. An efficient classification model termed CNN-COIF-LSTM was proposed in [45], where the experiments were conducted with different variants. A local CNN-LSTM model was proposed by Wang et al. where the regional information in sentences is considered along with the long-distance dependencies between sentences [46]. By combining different word embedding with different learning approaches like LSTM, Bi-LSTM, CNN, and GRU, a normal hybrid model was proposed by Salur and Aydin [47]. A self-interaction technique with attention perspective intermingled with label embedding was utilized by Dong et al. for efficient text classification, where attention schemes have been given good importance [48].

Boosting of the model is done by attention mechanism so that the learnable parameters are reduced, and the accuracy is improved further. The total quality of the input features shrinks as the model utilizes CNN to extract the features from various positions in a sentence. The contextual information is extracted from the features of LSTM. Over the inputs, the bias alignment is utilized by the attention mechanism, and therefore the weights to the input components are assigned, which are highly correlated with the classification. Thus, during the training phase, the number of parameters to be learned is still mitigated. For the variable length sequences, the weight distribution is also enhanced by the attention mechanism. To make sure that the model has a good accuracy rate, the weights that were pretrained utilizing word2vec are utilized. Later, the performance metrics are analyzed thoroughly.

In this work, two hybrid models with respect to the positioning of the attention mechanism is utilized and developed for efficient text classification. The organization of the work is as follows: Section 2 discusses the implementation of the CBAO model, and Section 3 discusses the implementation of the CABO model, followed by the results and discussion in Section 4 and conclusion in Section 5.

2. Deep Learning Framework

Model-CBAO Model

2.1. Word2vec Process. The natural language is transformed into distributed vector representations by the popular sequence embedding technique called word2vec. In a multi-dimensional space, the capturing of contextual word-to-word relationships is done, and therefore it is usually utilized as a significant step for prediction in information and semantic retrieval assignments. There are two definite components in the word2vec process, namely, CBOW and skip-gram. When the context words are given, the target word is inferred by the CBOW component. When an input word is specified, the context words are inferred by the skip-gram component.

2.2. CNN Modeling. In the text classification tasks, 2D CNNs have been recently used in a similar way to image classification tasks and have performed better than the sequenced based techniques. A convolution layer and subsampling layer are utilized to generate a group of convolutions and pooling; thus, the feature map is constructed by the CNN. A 1D cross-correlation operation is utilized by the convolution layer in 1D CNN by means of involving a sliding convolution window in our work. Once the input text progresses from left to right, the sliding convolution window has variable size kernels. A 1D global maximum pooling layer is present in the max-overtime pooling layer which utilizes it efficiently; thus, the number of features required to encode the text is reduced. A 1D CNN structure which is utilized for text classification is presented in Figure 1.

2.3. Bi-LSTM Process. The Bi-LSTM neural network comprises LSTM units that operate in both directions so the past

FIGURE 1: A 1D-CNN structure for text classification.

and future context information can be incorporated very easily. Without the need to retain any duplicate context information, the long-term dependencies can be well read by Bi-LSTM. For sequential modeling issues, this Bi-LSTM has excellent demonstrations, and therefore it can be applied to text classification very efficiently. In order to apprehend the possessions in two contexts, the propagation of the two parallel layers of the Bi-LSTM network is done in two directions with forward and reverse passes. Figure 2 illustrates the sequential feature capturing process using Bi-LSTM architecture.

2.4. Attention Mechanism. There are two important issues in the RNN based seq2seq models. There are chances for information loss along with vanishing gradient problem, as the compression of all the information in one fixed-size vector cannot be done. When the length of the word increases, the accuracy deteriorates to a great extent. Therefore, to enhance the prediction accuracy, attention mechanism has been proposed recently. When the input sequences become too low, sometimes the accuracy can deteriorate. In every time step, the prediction of the output word is done by the decoder, and the referencing of the full input sentences from the encoder is done once again. All the input sentences need not be referenced in an equal aspect. More attention is given to the input word that is highly related to the word which has to be predicted. To each input, the initial weights are assigned by the attention mechanism. During the training phase, the update of these weights is done based on the correlation between the input and ultimate prediction.

2.5. CBAO Model. Considerable research attention has been devoted to LSTMs, and many studies based on them are quite popular. To manage the drawbacks of the vanishing gradients, many gating schemes utilized by LSTMs are engaged to aid them in tracking long-term dependencies when long sequences are taken into input. However, to extract both the local content information and the context information of future tokens, the existing LSTMs fail to a large extent. The various relationships between specific parts of a document cannot be recognized properly, and therefore the accuracy of LSTM is further damaged. To handle these drawbacks efficiently, a 1D CNN is proposed here. The distinct merits of LSTM and CNN are leveraged well in this study, and therefore a hybrid model based on it is proposed

FIGURE 2: Sequential feature capturing using Bi-LSTM.

where, for text feature extraction, a CNN is utilized and, for sentiment classification, the Bi-LSTM component with an attention mechanism is utilized. Between the adjacent words, the complex association is captured so that the classification accuracy is increased, thereby achieving the objective of this study. The proposed CBAO model architecture is indicated in Figure 3.

2.5.1. Sequence Embedding Layer. The preprocessing of the text was done before the data was fed to the model. Preprocessing encompasses tasks like eliminating redundant, meaningless, and duplicate words along with the conversion of other forms of words to approximations. A distinct and quite meaningful sequence of words is provided by the preprocessed dataset, where a unique identification is provided to each word. The distributed specification of every preprocessed input token is learned thoroughly by the embedding layer. The latent alliances between the words which are more probable to emerge in similar conditions are reflected by the token representation. Vectors pretrained by word2vec's skip-gram model were utilized.

2.5.2. 1D-CNN Model. Extraction of features from the input text is the main role of the convolution process. From the original text, the low-level semantic features are extracted by the convolutional layers, and the number of dimensions is mitigated so that the text is processed as sequential data. Several 1D convolutional kernels are utilized in this study so that the convolution is performed over the input vectors. By

means of concatenating the embedding vectors of the component words, a vector of sequential text is obtained and defined as

$$Y_{1:T} = [y_1, y_2, y_3, \dots, y_T], \quad (1)$$

where the number of tokens in the text is indicated by T . Convolution kernels with different sizes are used for $Y_{1:T}$ so that the fundamental unigram, bigram, and trigram features using a 1D CNN are captured. When a window of " s " words that extends from $t: t+s$ is considered as an input during the t^{th} convolution, the features are generated for that particular window by the convolution process and are represented as follows:

$$h_{s,t} = \tanh(W_s y_{t:t+s-1} + b_s), \quad (2)$$

where the embedding vectors of the words in the window are represented by $y_{t:t+s-1}$, the learnable weight matrix is represented as W_s , and the bias is represented as b_s . In the different regions of the text, each filter is implemented, and therefore the feature map of the filter with a suitable convolution size s is represented as

$$h_s = [h_{s1}, h_{s2}, h_{s3}, h_{s4}, \dots, y_{T-s+1}]. \quad (3)$$

Between several adjacent words, the hidden correlations are well captured, and this is one of the primary advantages of utilizing a convolution kernel with varied widths. By utilizing max-outline pooling, the number of trainable parameters is reduced during the process of feature learning and serves as a salient characteristic of employing a CNN for textual feature extraction. Many convolution channels act upon the input, and every unique channel comprises values in various time steps. The largest value of all time steps is achieved from the output of each convolution channel during max-overtime pooling. In the feature maps, the max-overtime pooling is implemented with a convolution size " s " for each convolution kernel and is represented as

$$p_s = \text{Max}^t(h_{s1}, h_{s2}, h_{s3}, h_{s4}, \dots, y_{T-s+1}). \quad (4)$$

The concatenation of p_s for every filter size $s = 1, 2, 3$ is done to obtain the final feature map by extracting the unigram, bigram, and trigram hidden features and is represented as

$$h_s = [p_1, p_2, p_3]. \quad (5)$$

The total number of dimensions in the input features is mitigated and passed to the classifier or prediction model by utilizing the CNN over LSTM.

2.5.3. Bi-LSTM Attention Layer. In this model, the primary classification constituent was constructed on an attention based Bi-LSTM. The correlation is not the same between each word and the final classification even though the input features are shrunk by the CNN which is meant for the prediction purposes. The advantages of CNN and Bi-LSTM are leveraged in this study. In order to encode the long-distance word dependencies in an effective manner, Bi-LSTM is used. The features generated and created by the

FIGURE 3: The proposed CBAO model architecture.

CNN are received and extracted by the final hidden layer. The easy and full-fledged access to the contextual information, for both preceding and subsequent information, is provided by the Bi-LSTM. The information procured by Bi-LSTM can be considered as two various textual renditions. To the Bi-LSTM model, the features reaped from the CNN are fed so that a representation of the sequence is produced. The attention layer gets this final feature representation and thus the features that are highly correlated are selected for final classification. The prediction accuracy is considerably increased by the attention mechanism, and the number of learned weights required for the prediction is reduced. The attention mechanism utilized in this proposed model is Bahdanau attention with its respective attention scores.

3. Deep Learning Framework Model-CABO Model

3.1. CABO Model. The proposed CABO model comprises 4 layers: word embedding layer, CNN layer with attention mechanism, Bi-LSTM layer, and output Softmax layer. Initially, when a text is given as input, this text is transferred into a discourse vector by the word embedding layer, and this is done utilizing a dictionary index. The extraction of the features from the text is done with the help of CNN layers with attention mechanism. The features are learned deeply by means of using Bi-LSTM layer. Finally, with the help of Softmax output layer, the classification result is given as the output by the model with a good loss function. The detailed explanation is given in the following sub-sections. Figure 4 shows the architecture of the proposed CABO model.

3.1.1. Word Embedding. Mapping the words to real vector is the primary idea of word embedding. In order to gain a good structural expression of the data, the preprocessing of the text should be done before word embedding is done. To remove the low frequency words, the preprocessing of the words is done. In the sequence $\{w_1, w_2, \dots, w_l\}$, the length of the input text is projected as l . In order to get the word vector matrix $V = \{v_1, v_2, \dots, v_l\}$, word2vec was used, where $V \in \mathbb{R}^{l \times s^w}$, where s^w indicates the word vector size. For every distance, the word vector has to be fine-tuned very carefully during the training phase, so that the feature extraction

performance is improved. This is done in order to avoid the confusion as words can have various meanings in specific contexts.

3.1.2. CNN with Attention Mechanism. For the purpose of feature extraction, the most widely used connection-based model is CNN. The connection-based multiple filters which have similar window size are present in the convolution layer, and they progress towards the output of the final layer. In order to learn the local features of the word vectors, 2 ResNet blocks are utilized, and they are shown in Figure 5. In one block, 3 convolutional layers are present, followed by batch normalization and ReLU activation. The building block is defined as follows:

$$z = F(y, \{w_i\}) + y, \quad (6)$$

where, for one ResNet block, the input and output vectors are represented by y and z , respectively. After the activation of ReLU, the final output of the ResNet layer is expressed as $\{h_1, h_2, \dots, h_l\}$. In order to explore the important components of the high level semantic, the attention mechanism is added to the ResNet layer. For all the states $\{h_1, h_2, \dots, h_l\}$,

$$\begin{aligned} u_i^t &= s_t W h_i, \\ a_i^t &= \text{soft max}(u_i^t), \\ e^t &= \sum_i^t q_i^t h_i, \end{aligned} \quad (7)$$

where e^t indicates the encoded state denoted by the weighted sum of $\{h_1, h_2, \dots, h_l\}$ at time t . The weight of h_i is expressed as q_i^t . $W \in \mathbb{R}^{s \times s}$ and $s_t \in \mathbb{R}^s$ are utilized for transformation of h_i into a scalar. The outputs of the attention mechanism and ResNet are multiplied by the model, and it sends the output to the next layer.

3.2. Bi-LSTM Module. In order to overcome the gradient vanishing issue of the RNN, LSTM has been proposed. The duplication of the first recurrent layer in the network is done by Bi-LSTM so that the two layers can exist side by side. By utilizing the sample sentence, the demonstration of the bi-directional network is done. "I am very tired and therefore I want to AAA for the whole day." From the understanding of the word "tired," the prediction of the AAA may be "sleep," "eat,"

FIGURE 4: The proposed CABO model architecture.

“dance,” “sing,” “play,” etc. Based on the following words “the whole day,” the irrelevant options that are unsuitable in this context can be identified and removed. A Bi-LSTM learns the information from the extracted context while the Uni-LSTM learns the information from only one direction. The framework of Bi-LSTM is shown in Figure 6. The input is expressed as matrix Y , and the output is expressed as matrix O . The output of the forward layer memory cell is expressed by the sequence $\{h_1, h_2, \dots, h_l\}$, and the output of the backward layer memory cell is expressed as $\{h'_1, h'_2, \dots, h'_l\}$. The full operation of Bi-LSTM is expressed as follows:

$$\begin{aligned} h_t &= f(U_t y_t + W_{t-1} h_{t-1}), \\ h'_t &= f(U'_t y_t + W'_{t+1} h'_{t+1}), \\ o_t &= g(V_t h_t + V'_t h'_t). \end{aligned} \quad (8)$$

The weight matrices of the network are expressed by U_i , V_i , and W_i , respectively. The nonlinear activation functions are expressed as $f(\cdot)$ and $g(\cdot)$.

Based on the forward layer state h_t and the backward layer state h'_t , the computation of the output o_t is done at each time step t .

3.3. Output Layer with Good Focal Loss for Both the CBAO and CABO Models. Softmax classification is used in the output layer for both models. The number of text classes is nothing but the output size, and (9) is used to express the conditional probability value of each type, where, to achieve the probability normalization, Softmax nonlinear activation is utilized as

$$p_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}, \quad (9)$$

where the probability of reflecting the features of the class i $\{z_1, z_2, \dots, z_N\}$ is denoted by p_i , and it indicates the output. i and $j \in \{1, 2, \dots, N\}$, where the number of classes is expressed by N .

In order to solve the class imbalance problem, focal loss function is utilized instead of the cross entropy. By adding a Gaussian weight, the focal loss is improved so that the small weights can be enhanced. The Gaussian weight is greater if there are a smaller number of samples in a particular class, and therefore the attention paid by the model is more. The specification of the improved loss function [49] is expressed as

$$\begin{aligned} \text{Loss}(p_i) &= -\alpha_i (1 - p_i)^\gamma \log(p_i), \\ \alpha_i &= \alpha + \beta \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{C_i^2}{2\sigma^2}\right), \end{aligned} \quad (10)$$

where the weighting factor is expressed by α_i , and the count of every class is expressed by C_i . To the cross-entropy loss, a modulating factor $(1 - p_i)^\gamma$ is added by the focal loss with parameter $\gamma \geq 0$. The focal loss is equivalent to cross entropy when $\gamma = 0$. α is decreased when γ increases. The $\gamma = 2.5$ and $\alpha = 0.5$; $\alpha_i \in [0, 1]$ is managed in this work after several trial-and-error experiments. To control the weight of loss of every class, the parameters β and σ are utilized effectively.

4. Results and Discussion

In this section, the datasets analyzed along with the parameter settings of the proposed deep learning techniques, performance evaluation metrics, and comparison with previous works are discussed in detail.

FIGURE 5: ResNet block architecture.

FIGURE 6: Bi-LSTM framework.

4.1. Datasets Analyzed. To evaluate the performance of the proposed model, the experiments are conducted on three large-scale public datasets as follows.

4.1.1. Yelp 2016. This dataset is obtained from the Yelp Dataset Challenge (<https://www.yelp.com/dataset/challenge/>). A five-level rating is present in this dataset which ranges from 1 to 5, and therefore the documents can be classified into five classes.

4.1.2. Amazon Review. It is obtained from Amazon products data (<http://jmcauley.ucsd.edu/data/amazon/>). From May 1996 to July 2014, the product reviews and the metadata from Amazon are obtained in this dataset. To the product reviews, a five-level rating is provided ranging from 1 to 5. All the other necessary and in-depth details of the dataset can be obtained from the mentioned web link.

4.1.3. IMDB. IMDB dataset has been developed (<https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>) for the purpose of binary sentiment

classification of the movie reviews. In this dataset, the numbers of positive and negative reviews are equally divided such as 25,000 reviews for training set and 25,000 reviews for test set.

4.2. Parameter Settings of the Proposed Deep Learning Techniques. As far as the IMDB dataset is concerned, the proposed two models have the following specifications when performing the experiment. The embedding of the word vector utilized in this investigation was done using skip-gram technique in word2vec. The embedding size was set to 500. The dropout rate was set as 0.4, and the batch size was set to 128. The Adam optimizer was utilized along with L2 regularizer. As far as the Yelp 2016 and Amazon review datasets are concerned, the proposed two models have the following specifications. Once the embedding of the word vector using skip-gram technique in word2vec is done, the update of the word embedding matrix is randomly generated based on the stochastic gradient descent process. The embedding dimension is set as 500, and in all the layers, the scale of the gradients is almost similar. The loss function utilized is focal loss function,

and the batch size is set to 50. When the norm exceeds a specific threshold, the scaling gradient is utilized to adopt the gradient clipping. The models are trained well with a learning rate of 0.001 by the stochastic gradient approach. The number of batches is set accordingly so that overfitting does not occur. The kernel size of the CNN is set as 5, the dropout rate is set as 0.5, and the number of filters is considered as 512 in our experiment. The activation function is ReLU, and the weights are updated using Adam optimizer. For different parameters, an adaptive and independent learning rate was designed using Adam by means of computing the first order moment and second order moment of the gradient. To protect against overfitting, random selection of the data is done by the dropout layer, thereby making the model more effective and robust. A good and reasonable dropout rate along with input dimensionality is necessary for modeling as these factors can affect the time efficiency to a large extent.

4.3. Performance Evaluation Metrics. The following performance metric are used for evaluation in our work.

The classifier performance is computed by accuracy and is expressed as

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (11)$$

The sensitivity is expressed as

$$\text{sensitivity} = \frac{TP}{TP + FN}. \quad (12)$$

The specificity is expressed as

$$\text{specificity} = \frac{TN}{FP + TN}. \quad (13)$$

The precision is expressed as

$$\text{precision} = \frac{TP}{TP + FP}. \quad (14)$$

The geometric mean (g -mean) is indicated as follows:

$$g - \text{mean} = \sqrt{\frac{TP \times TN}{(TP + FN) \times (TN + FP)}}. \quad (15)$$

The Matthews Correlation Coefficient (MCC) which is nothing but a Chi-square statistic is expressed as

$$\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FN) \times (TP + FP) \times (TN + FP) \times (TN + FN)}} \quad (16)$$

From Table 1, it is observed that CBAO model attained a high accuracy of 92.72% in IMDB dataset. The model reached an accuracy of 63.78% for Yelp 2016 dataset and 66.02% for Amazon review dataset. IMDB dataset exhibits a low error rate of 7.28% and a high error rate of 36.22% and 33.98% in Yelp 2016 dataset and Amazon review dataset, respectively. MCC is a balanced parameter which implies better narrative in the classification of any model. The CBAO model with IMBD dataset exhibits a high MCC value of

0.8544, and the other two datasets ebb at 0.273 and 0.3165 MCC values, respectively.

From Table 2, it is observed that the CABO model reached a high accuracy of 90.51% in IMDB dataset. The model showed an accuracy of 68.15% for Yelp 2016 dataset and 60.75% for Amazon review dataset. IMDB dataset indicates a low error rate of 9.49% and a high error rate of 31.85% and 39.25% in Yelp 2016 dataset and Amazon review dataset, respectively. The CABO model with IMBD dataset exhibits a high MCC value of 0.809, and the other two datasets display low MCC values of 0.367 and 0.218, respectively.

Figure 7 demonstrates the performance measures of major six parameters for the CBAO model. It is found from Figure 7 that CBAO model attained a higher accuracy, 92.72%, in IMDB dataset than in the other two datasets. The model reached an accuracy of 63.78% for Yelp 2016 dataset and 66.02% for Amazon review dataset. For Yelp 2016 dataset, a low value of F -score is due to lesser specificity of 57.14%, and again a low value of F -score is achieved in Amazon review dataset due to the same reason. IMDB dataset achieved a high g -mean value of 92.68% when compared to the g -mean values of 63.43% and 65.80% in the Yelp 2016 and Amazon review datasets, respectively.

Figure 8 indicates the performance measures of five minor parameters, namely, error rate, FPR, false positive rate (FPR), false negative rate (balanced coefficient rate (BCR) and J-coefficient, for the CBAO model. It is found from Figure 8 that IMDB dataset demonstrates a low error rate of 7.28% and a high error rate of 36.22% and 33.98% in Yelp 2016 dataset and Amazon review dataset, respectively. In the case of J-coefficient parameter, the CBAO model is well distinguished by a higher value of 85.71% in the IMDB dataset. For the Yelp 2016 dataset and Amazon review dataset, the J-coefficient maintained the values of 48.27% and 46.15% accordingly. This low value of J-coefficient is due to the presence of high error rate in the classification of the CBAO model for Yelp 2016 dataset and Amazon review dataset.

Figure 9 exhibits the performance measures of major six parameters for the CABO model. It is found from Figure 9 that CABO model attained a high accuracy of 90.51% in IMDB dataset compared to the other two datasets. The model reached an accuracy of 68.15% for Yelp 2016 dataset and 60.75% for Amazon review dataset. In the case of Yelp 2016 dataset, a low value of F -score is due to lesser sensitivity of 60.12%, and a low value of F -score is achieved in Amazon review dataset for the same reason. IMDB dataset achieved a high g -mean value of 90.38% when compared to the g -mean values of 67.67% and 60.45% in the Yelp 2016 dataset and Amazon review dataset, respectively.

Figure 10 demonstrates the performance measures of five minor parameters, namely, error rate, FPR, FNR, BCR, and J-coefficient, for the CABO model. It is found from Figure 10 that IMDB dataset demonstrates a low error rate of 9.49% and a high error rate of 31.85% and 39.25% in Yelp 2016 dataset and Amazon review dataset, respectively. In the case of J-coefficient parameter, the CABO model is distinguished by a higher value of 82.6% in IMDB dataset. For the Yelp 2016 and Amazon review datasets, the J-coefficient maintained the values of 48% and 40.74% accordingly. This

TABLE 1: Results of the proposed CBAO model.

Performance metrics on the datasets (%)	Yelp 2016 dataset	Amazon review dataset	IMDB dataset
Accuracy	63.78	66.02	92.72
Sensitivity	70.43	60.63	90.21
Specificity	57.14	71.42	95.23
Precision	60.86	66.66	94.73
g -mean	63.43	65.80	92.68
MCC	0.273	0.3165	0.8544

TABLE 2: Results of the proposed CABO model.

Performance metrics on the datasets (%)	Yelp 2016 dataset	Amazon review dataset	IMDB dataset
Accuracy	68.15	60.75	90.51
Sensitivity	60.12	54.81	95.31
Specificity	76.19	66.69	85.71
Precision	70.58	61.11	86.36
g -mean	67.67	60.45	90.38
MCC	0.367	0.218	0.809

FIGURE 7: Performance measures analysis for the proposed CBAO model.

low value of J-coefficient is due to the prevalence of high error rate in the classification of the CBAO model for Yelp 2016 dataset and Amazon review dataset.

4.4. Comparison with Other Works. Selected important works published in high quality peer-reviewed journals are taken for comparison in this work. As far as IMDB dataset is concerned, non-deep learning techniques and deep learning techniques were utilized for classification. The most famous non-deep learning techniques produced a classification accuracy of 67.40% using linear discriminant analysis (LDA) [50] and 87.80% using support vector machine (SVM) [51]. A classification accuracy of 83.2% was reported for multitask deep neural networks (DNN) [52], 94.99% was reported for block sparse LSTM [53], 94.10% was reported for virtual adversarial

FIGURE 8: Error rate, FPR, FNR, BCR, and J-coefficient performance measures analysis for the proposed CBAO model.

FIGURE 9: Performance measures analysis for the proposed CABO model.

FIGURE 10: Error rate, FPR, FNR, BCR, and J-coefficient performance measures analysis for the proposed CABO model.

training [54], and a large BERT model [55] produced an accuracy of 95.79%. As far as Amazon datasets are concerned, a character level CNN [17] produced 59.46%, deep pyramid CNN [56] produced 65.82%, CCapsNet [57] produced 60.95%, large BERT model [55] produced 62.20%, and base BERT model [55] produced 61.60%. As far as Yelp datasets are concerned, the character level CNN [17] produced 62.05%, deep pyramid CNN [56] produced 69.40%, CCapsNet [57] produced 65.85%, BERT fine-tuned model [58] produced 62.92%, and base BERT model [55] produced 70.58%. However, when our results with the proposed two models are considered and analyzed, the results obtained are pretty good as a high classification accuracy of 92.72% was obtained with the proposed CABO deep learning model for IMDB dataset, a good classification accuracy of 66.02% was obtained with the proposed CABO deep learning model for Amazon review model, and again a good classification accuracy of 68.15% was obtained with the proposed CABO model for Yelp 2016 dataset. Consequently, when comparing our work with the previously obtained results, the results seem to be promising and effective though many of the previously published results have surpassed our results by obtaining slightly more classification accuracy. In the field of machine learning and pattern recognition, it is not always important to obtain a better classification accuracy, but the methodology implemented along with the interpretation of the results is quite important, which has been very well focused on in our work. However, this research mainly focused on the analysis of attention mechanism positioning methodology and, from the results obtained, the method seems to be pretty fine for carrying out future analysis with a similar kind of strategy not only for text classification but for other domains too.

5. Conclusion and Future Work

The technique of categorizing text into a group of words is called text classification. The analysis of text can be done automatically by text classification using NLP by means of assigning a set of predefined categories depending on its context. Thus, NLP which is classified into rule-based systems,

machine-based systems, and hybrid systems is utilized for various applications such as topic detection, sentiment analysis, and natural language inference purposes. In this work, a fair survey of text classification models is initially introduced, and then two kinds of hybrid models are designed by giving special emphasis to the positioning of the attention mechanism. The results show that when the proposed CABO deep learning model is tested on IMDB dataset, Amazon review dataset, and Yelp 2016 dataset, a high classification accuracy of 92.72%, 66.02%, and 63.78% is obtained. For the proposed CABO deep learning model, an error rate of 7.28%, 33.98%, and 36.22% is obtained for IMDB dataset, Amazon review dataset, and Yelp 2016 dataset, respectively. Similarly, when the proposed CABO deep learning model is tested on IMDB dataset, Amazon review dataset, and Yelp 2016 dataset, a high classification accuracy of 90.51%, 60.75%, and 68.15% is obtained. For the proposed CABO deep learning model, an error rate of 9.49%, 39.25%, and 31.85% is obtained for IMDB dataset, Amazon review dataset, and Yelp 2016 dataset, respectively. In the future, we aim to work with a plethora of other deep learning models for efficient text classification.

Data Availability

The data will be made available to researchers upon request to the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported and funded by the Korean National Police Agency (Project Name: AI-Based Crime Investigation Support System/Project Number: PR10-02-000-21) and partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1F1A1061814).

References

- [1] B. Krawczyk, B. T. McInnes, and A. Cano, "Sentiment classification from multi-class imbalanced twitter data using binarization," in *Proceedings of the International Conference on Hybrid Artificial Intelligence Systems*, pp. 26–37, La Rioja, Spain, June 2017.
- [2] Z. Wei and F. Gao, "An improvement to naive bayes for text classification," *Procedia Engineering*, vol. 15, pp. 2160–2164, 2011.
- [3] E. Leopold and J. Kindermann, "Text categorization with support vector machines. How to represent texts in input space?" *Machine Learning*, vol. 46, no. 1–3, pp. 423–444, 2002.
- [4] E. H. Han, G. Karypis, and V. Kumar, "Text categorization using weight adjusted K-nearest neighbor classification," *Advances in Knowledge Discovery and Data Mining*, Springer, vol. 2035, pp. 53–65, Berlin, Heidelberg, 2001.
- [5] B. Yu, Z.-B. Xu, and C.-H. Li, "Latent semantic analysis for text categorization using neural network," *Knowledge-Based Systems*, vol. 21, no. 8, pp. 900–904, 2008.

- [6] P. Frasconi, G. Soda, and A. Vullo, "Hidden Markov models for text categorization in multi-page documents," *Journal of Intelligent Information Systems*, vol. 18, no. 2-3, pp. 195–217, 2002.
- [7] J. Hoffman, O. Navarro, F. Kastner, B. Janssen, and M. Hubner, "A Survey on CNN and RNN implementations," in *Proceedings of the Seventh International Conference on Performance, Safety and Robustness in Complex Systems and Applications*, pp. 33–39, Venice, Italy, April 2017.
- [8] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, IEEE, Venice, Italy, October 2017.
- [9] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," 2017, <https://arxiv.org/abs/1702.01923>.
- [10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, <https://arxiv.org/abs/1301.3781>.
- [11] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Computer Science*, vol. 7, 2016.
- [12] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext. zip: compressing text classification models," 2016, <https://arxiv.org/abs/1612.03651>.
- [13] M. Iyyer, V. Manjunatha, J. B. Gruber, and H. Daumé, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 1681–1691, Beijing, China, July 2015.
- [14] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, vol. 1, Beijing, China, July 2015.
- [15] X. Zhu, P. Sobihani, and H. Guo, "Long short-term memory over recursive structures," in *Proceedings of the International Conference on Machine Learning*, vol. 37, pp. 1604–1612, Lille France, July 2015.
- [16] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, October 2014.
- [17] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, pp. 649–657, 2015.
- [18] J. D. Prusa and T. M. Khoshgoftaar, "Designing a better data representation for deep neural networks and text classification," in *Proceedings of the 2016 IEEE 17th International Conference on Information Reuse and Integration*, Pittsburgh, PA, USA, July 2016.
- [19] J. Kim, S. Jang, E. Park, and S. Choi, "Text classification using capsules," *Neurocomputing*, vol. 376, pp. 214–221, 2020.
- [20] R. Aly, S. Remus, and C. Biemann, "Hierarchical multi-label classification of text with capsule networks," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 323–330, Hamburg, Germany, January 2019.
- [21] M. Yang, W. Zhao, L. Chen, Q. Qu, Z. Zhao, and Y. Shen, "Investigating the transferring capability of capsule networks for text classification," *Neural Networks*, vol. 118, pp. 247–261, 2019.
- [22] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, and Z. Zhao, "Investigating capsule networks with dynamic routing for text classification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3110–3119, Brussels, Belgium, November 2018.
- [23] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "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*, pp. 1480–1489, CA, USA, June 2016.
- [24] X. Zhou, X. Wan, and J. Xiao, "Attention-based lstm network for cross-lingual sentiment classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 247–256, Austin, TX, USA, November 2016.
- [25] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: directional self-attention network for rnn/cnn-free language understanding," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, LA, USA, February 2018.
- [26] Y. Liu, C. Sun, L. Lin, and X. Wang, "Learning natural language inference using bidirectional lstm model and inner-attention," 2016, <https://arxiv.org/abs/1605.09090>.
- [27] Z. Lin, M. Feng, C. N. d. Santos et al., "A structured self-attentive sentence embedding," 2017, <https://arxiv.org/abs/1703.03130>.
- [28] T. Munkhdalai and H. Yu, "Neural semantic encoders," in *Proceedings of the conference Association for Computational Linguistics. Meeting*, vol. 1, Vancouver, Canada, July 2017.
- [29] A. Kumar, O. Irsoy, P. Ondruska et al., "Ask me anything: dynamic memory networks for natural language processing," in *Proceedings of the 33rd International Conference on Machine Learning*, NY, USA, June 2016.
- [30] H. Peng, J. Li, Y. He et al., "Large-scale hierarchical text classification with recursively regularized deep graph-cnn," in *Proceedings of the 2018 World Wide Web Conference*, pp. 1063–1072, Cancun, Mexico, April 2018.
- [31] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7370–7377, 2019.
- [32] F. Wu, T. Zhang, A. H. D. Souza, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, vol. 97, pp. 6861–6871, CA, USA, June 2019.
- [33] P. Neculoiu, M. Versteegh, and M. Rotaru, "Learning text similarity with siamese recurrent networks," in *Proceedings of the Repl4NLP workshop at ACL2016*, Berlin, Germany, August 2016.
- [34] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.
- [35] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-Aware neural language models," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ USA, February 2016.
- [36] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using bilstm-crf and cnn," *Expert Systems with Applications*, vol. 72, pp. 221–230, 2017.
- [37] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," 2015, <https://arxiv.org/abs/1511.08630>.
- [38] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1422–1432, Lisbon, Portugal, September 2015.

- [39] R. Zhang, H. Lee, and D. R. Radev, "Dependency sensitive convolutional neural networks for modeling sentences and documents," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, CA, USA, June 2016.
- [40] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "Hdltext: Hierarchical deep learning for text classification," in *Proceedings of the 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 364–371, Cancun, Mexico, December 2017.
- [41] O. Melamud, J. Goldberger, I. Dagan, S. Riezler, and Y. Goldberg, "Context2vec: learning generic context embedding with bidirectional LSTM," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 51–61, Berlin, Germany, August 2016.
- [42] L. Xiao, G. Wang, and Y. Zuo, "Research on patent text classification based on Word2Vec and LSTM," in *Proceedings of the 2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 1, pp. 71–74, Hangzhou, China, December 2018.
- [43] A. U. Rehman, A. K. Malik, B. Raza, and W. Ali, "A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis," *Multimedia Tools and Applications*, vol. 78, no. 18, Article ID 26597, 2019.
- [44] X. She and D. Zhang, "Text classification based on hybrid CNN-LSTM hybrid model," in *Proceedings of the 2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2, pp. 185–189, Hangzhou, China, December 2018.
- [45] Y. Luan and S. Lin, "Research on text classification based on CNN and LSTM," in *Proceedings of the 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 352–355, Dalian, China, March 2019.
- [46] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Tree-structured regional CNN-LSTM model for dimensional sentiment analysis," *IEEE/ACM trans. Audio speech lang. Processes*, vol. 28, pp. 581–591, 2019.
- [47] M. U. Salur and I. Aydin, "A novel hybrid deep learning model for sentiment classification," *IEEE Access*, vol. 8, Article ID 58080, 2020.
- [48] Y. Dong, P. Liu, Z. Zhu, Q. Wang, and Q. Zhang, "A fusion model-based label embedding and self-interaction attention for text classification," *IEEE Access*, vol. 8, Article ID 30548, 2020.
- [49] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the 2017 IEEE International Conference on Computer Vision*, pp. 2380–7504, Venice, Italy, October 2017.
- [50] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 142–150, Portland, Oregon, June 2011.
- [51] S. Wang and C. D. Manning, "Baselines and bigrams: simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers*, vol. 2, pp. 90–94, Jeju Island, Korea, July 2012.
- [52] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4487–4496, Florence, Italy, July 2019.
- [53] S. Gray, A. Radford, and D. P. Kingma, "GPU kernels for block-sparse weights," *Block-Sparse GPU Kernels*, vol. 3, 2017.
- [54] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," 2016, <https://arxiv.org/abs/1605.07725>.
- [55] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?" in *Proceedings of the China National Conference on Chinese Computational Linguistics*, vol. 11856, pp. 194–206, Springer, Kunming, China, October 2019.
- [56] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 562–570, Vancouver, Canada, July 2017.
- [57] H. Ren and H. Lu, "Compositional coding capsule network with k-means routing for text classification," 2018, <https://arxiv.org/abs/1810.09177>.
- [58] Q. Xie, Z. Dai, E. Hovy, M. T. Luong, and Q. V. Le, "Un-supervised data augmentation," 2019, <https://arxiv.org/abs/1904.12848>.