IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 February 2024, accepted 16 March 2024, date of publication 25 March 2024, date of current version 1 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3381515

RESEARCH ARTICLE

Text Sentiment Analysis of Douban Film Short Comments Based on BERT-CNN-BiLSTM-Att Model

AIXIANG HE[1,2] AND MIDETH ABISADO[1]

[1]College of Computing and Information Technologies, National University, Manila 1008, Philippines
[2]College of Big Data and Artificial Intelligence, Anhui Xinhua University, Hefei 230088, China

Corresponding author: Aixiang He (heaixiang_1010@126.com)

This work was supported in part by Colleges and Universities in Anhui Province Natural Science Foundation Project under Grant 2022ah051870; in part by Anhui Provincial Higher Department of Education Quality Engineering Project under Grant 2022jcjs055, Grant 2020zdxsjg155, Grant 2017ghjc229, Grant 2016mooc197, and Grant 2017jyxm0524; and in part by the Computer Science and Technology Professional Support Discipline Project of Anhui Xinhua University under Grant fcxk202102.

ABSTRACT To solve the problems of polysemy and feature extraction in the text sentiment analysis process, a BERT-CNN-BiLSTM-Att hybrid model has been proposed for text sentiment analysis. The BERT pre-training model was established to break up the text input into words and obtain a dynamic word vector that was then input into the CNN and the BiLSTM models respectively. Later, the local features of the word vector, extracted using CNN, and the global features, extracted using BiLSTM, were fused, and the key information of the Douban movie review dataset was highlighted using the attention mechanism to realize sentiment categorization of the dataset. The results of comparison between the constructed model and Word2Vec-BiLSTM, Word2Vec-CNN, Word2Vec-CNN-BiLSTM-Att, BERT, BERT-CNN and BERT-BiLSTM models show that the model that runs against the test dataset has an increased accuracy by 4.63%,4.37%,3.64%,2.63%,2.56% and 5.54% respectively. The experimental findings reveal that BERT-CNN-BiLSTM-Att's sentiment analysis method is more accurate in performing sentiment classification.

INDEX TERMS BERT-CNN-BiLSTM-Att, sentiment analysis, hybrid model, film short text reviews comments.

I. INTRODUCTION

In the era of Web2.0, a myriad of concise texts encapsulating semantic nuances and subjective sentiments from users has proliferated on the Internet. This voluminous data constitutes a comprehensive portrayal and vital manifestation of users' consciousness and viewpoints, significantly influencing netizens' opinions, attitudes, judgments, and decision-making. More specifically, when selecting a movie on a digital platform, netizens frequently turn to the opinions shared by fellow moviegoers as their primary point of reference before making a purchasing decision [1]. Short texts exist in various forms such as text messages, product reviews, movie reviews, video captions, and subtitles, which can be found on Weibo, Twitter, Douban, Facebook and other social media platforms. Thus, how to utilize the computer technology precisely and efficiently and analyze sentimental information from voluminous short texts automatically is very important to product analysis, topic monitoring, public opinion surveillance, user modeling, opinion analysis, etc.

Sentiment analysis, often called as opinion mining, is a technique to analyze people's perspectives, attitudes, and other subjective feelings about certain items and their associated qualities. It relates to the judgment of emotional tendencies in writings that contain subjective messages [2]. Depending on the number of emotional categories involved, the emotion tendency analysis tasks are categorized into two groups (positive, negative), three groups (positive, negative,

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna D'Ulizia.

neutral) and multiple groups (happy, excited, sad, angry, etc.) [3]. Analysis of textual emotion tendency mainly contains: text representation and feature extraction, model training and analysis of the results [4]. Because of the characteristics of short text such as randomness, high ambiguity, and brevity, the problems of insufficient density, excessive noise, and a lack of independent context arise from the process of representing and extracting features from the text. As a result, correct features cannot be extracted, and text representation with richer contextual semantics becomes impossible [5]. Traditional methods completely separate text representation and feature extraction from model training, and their work focuses mainly on the former [6]. Traditionally, text representation and feature extraction are completed in a heuristic way, which is typically to manually construct emotion dictionaries (including Chinese emotion dictionaries [7] and English emotion dictionaries [8]) or feature rules (including syntactic features [9], emotion dictionary features [10], TF-IDF (Term Frequence-Inverse Document frequency) features [11], etc.).

This traditional method, which first uses heuristics to obtain text features and then combines with machine learning classifiers for classification, highly relies on the construction of heuristics and expert knowledge, and requires manual intervention, which greatly reduces work efficiency and limits the use of big data [12].

When tackling natural language processing tasks using deep learning, it's necessary to organize, digitize, and represent the text as word vectors for easy computer processing. Because of its high dimensionality, sparsity, and restricted ability to express features, the early word vector created based on the bag-of-word model is not suitable for feature extraction. Proposal of the word embedding-based distributed word representation approach makes it possible to analyze the short text-based emotion trend using deep learning.

The proposed word embedding-based approach converts short texts into low-dimensional real vectors after studying a big corpus. Word vectors are then input into a deep neural network to automatically extract contextual cues. The resulting codes are used to analyze the emotion trend. As the word embedding approach has been continually being improved, pre-trained language models have been proposed to determine word similarity.Word2Vec [13] was proposed in 2013, GloVe [14] in 2014, OpenAI GPT [15] in 2016, ELMo [16] and BERT [17]in 2018. In 2019, Transformer XL [18] and XLNet [19] that relied on Transformers [20] architecture were invented. Pre-training models that have been widely applied to emotional tendency analysis of short texts include Word2Vec, GloVe and BERT.

Deep learning model components that have been widely applied to emotional tendency analysis of short texts include: long short-term memory network (LSTM), convolutional neural network (CNN), memory network (MN), capsule network (CapsNets), graph convolutional neural network (GCN) and attention mechanism [21]. Long short-term memory networks are often used to capture the global semantics of short texts, while convolutional neural networks (CNNs) extract local features from short texts by performing convolution and pooling operations. The memory networks can better solve the long-term memory problem by adding additional storage units. The capsule networks learn the semantic relationship between words through dynamic routing, reducing information loss.

The feature information and structure information in short text can be fused by using graph convolutional neural network (GCN). At last, the combined attention (Att) mechanism is utilized to continue to learn the semantics between the texts and to further capture some minor emotional semantic information. Therefore, a BERT-CNN-BiLSTM-Att-based sentiment analysis model is proposed in this study, which is then trained against Douban film reviews. Through comparison and analysis of the experimental results, the BERT-CNN-BiLSTM-Att mixed model is proven to yield texts with less semantic loss and be more accurate than the BERT-CNN-BiLSTM method.

II. SENTIMENT ANALYSIS MODEL CONSTRUCTION

A. MODEL DESIGN

Convolutional neural networks (CNN) and recurrent neural networks (RNN) are two deep learning frameworks that are popular in the field of natural language processing (NLP) [22]. In a text sentiment analysis challenge, the input is a one-dimensional linear sequence of unknown length. It is critical to evaluate not only the local structure and properties of the words in the input text, but also their positional relationships and temporal order. The CNN network, which mines the local structure between words using convolutions and other processes, is well-suited to capturing local information of the text. In CNN, signal propagation is limited to the top layer, and the independent processing of samples makes it challenging to capture spatial and temporal information. RNN networks with a linear sequence structure gather inputs from front to rear. The outputs of a neuron can operate on itself in a direct manner in the following cycle, achieving the goal of time series modeling. In most RNN networks, the inputs determine the network extension. The longer the length of the inputs to the network, the deeper the RNN network becomes. In some complex cases, the RNN network may be too deep because of the expanded network, and as the distance between the contexts increases, become more dependent in the long run. This makes it unable to learn long-distance connection information, resulting in the phenomena of ''gradient disappearance''. To address the aforementioned issue, a short- and long-term memory unit of LSTM (a variation of RNN network) is proposed to complete the memory process in time through the switch of the unit door and avoid ''gradient disappearance'' [23]. LSTM networks outperform typical RNN networks in terms of capturing longer distance dependencies. However, both RNN and LSTM networks can only forecast the output of the next moment using the preceding moment's time series information. In some circumstances, the output of the present time may be tied to both the previous and future states. To collect context information in sentiment analysis

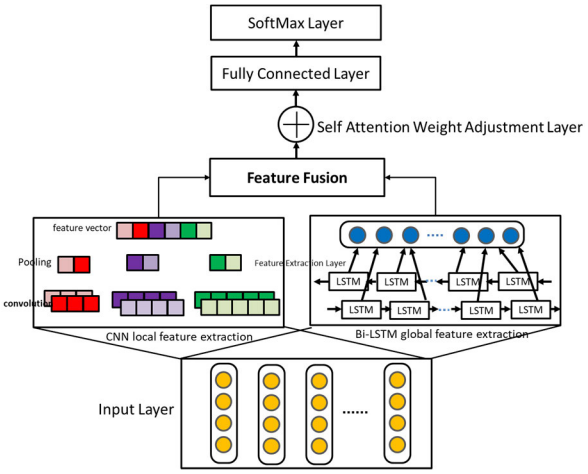

FIGURE 1. BERT-CNN-BiLSTM-Att model.

tasks, Bi-LSTM is typically used in conjunction with forward LSTM and backward LSTM. Furthermore, in emotion analysis applications, the dynamic word vector derived from the BERT model can be used to tackle the polysemy problem [24]. Therefore, based on the above single deep learning network, this study integrates the advantages of CNN network [25] in extracting local text features, of Bi-LSTM network in extracting global text features, and of Self-Attention mechanism in capturing various aspects of text structure, and of the BERT network's dynamic word vector to propose a novel emotion analysis model of BERT-BiLSTM-CNN-Att hybrid neural network coupled with multi-feature fusion of self-attention mechanism [26].

B. OVERALL STRUCTURE OF THE MODEL

For the structure of the BERT-CNN-BiLSTM-Att model, refer to FIG 1. The model comprises five layers: the input, feature extraction, feature fusion, self-attention weight adjustment, fully connected, and output layers.

(1) Input layer: mainly responsible for the vectorized representation of text. In this study, the BERT model is selected to transform the word vector, thus avoiding the ambiguity caused by word segmentation and establishing the dynamic word vector; the formatted expression is an input into the feature extraction layer.

(2) Feature extraction layer: This layer takes charge of extracting features from the text. On the basis of the distinctive characteristics of CNN and BiLSTM models, one component extract local text features via convolutional operations, while another component utilizes BiLSTM to capture global text features based on the contextual information.

(3) Feature fusion layer: In this layer, feature vectors established via local CNN and global BiLSTM feature extraction are combined.

(4) Self-attention weight adjustment layer: The attention weight is computed using the self-attention mechanism to finalize representation of text features.

(5) Fully connected output layer: The final features of the text, computed by the Attention mechanism, are fed into

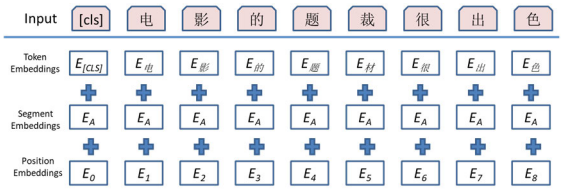

FIGURE 2. Input of BERT model.

the fully connected layer. To reduce overfitting problems, the Dropout mechanism is incorporated. In order to get the results of the emotion categorization, the text will finally run through the SoftMax layer.

The working mechanism of each layer of the BERT-CNN-BiLSTM-Att model and how each layer works together will be further described in detail below.

1) INPUT LAYER

The input layer is of great significance to generating the vectorized representation of text. The data of the collected original corpus will be preprocessed before completion of this representation. To address the challenge from word polysemy, the BERT model utilizes the Embedding layer to extract the word sequence and convert it into word vectors.

A groundbreaking innovation of the proposed BERT model is to employ the Transformer Decoder (containing mask multiple attention) as an extractor and adopt relevant mask training decoder methods. Despite lacking text generation capabilities due to the dual encoding approach, BERT utilizes the complete contextual information for each word to the maximum extent during the encoding of the input text. BERT exhibits a greater ability to extract semantic information than a unidirectional encoder.

The input of the BERT model is a word vector that is composed of an initial word vector, a segment vector, and a position vector. The word vector typically contains randomly initialized values that are learned during model training to obtain the overall semantic messages of the text. The position vector accounts for the variation in semantic information conveyed by words appearing at different positions within the text, efficiently resolving the problem of the attention mechanism's disregard for word order. The BERT model assigns distinct vectors to words in different positions, as shown in FIGURE 2.

In BERT, the input text is marked with the [cls] symbol at the beginning of the sentence. BERT divides English words that are processed into more granular semantic units. For example, the word ''watch'' is divided into ''watch'' and ''watching''. For a text in Chinese to be processed, however, words are directly interpreted as the fundamental units of the text, with no distinction made between tenses. The output of the BERT model varies from different downstream text analysis tasks. The output of the [cls] symbol itself does not carry explicit semantic information, but can effectively capture the overall emotional information that a sentence contains. The corresponding output vector of [cls] is commonly used for text classification tasks.

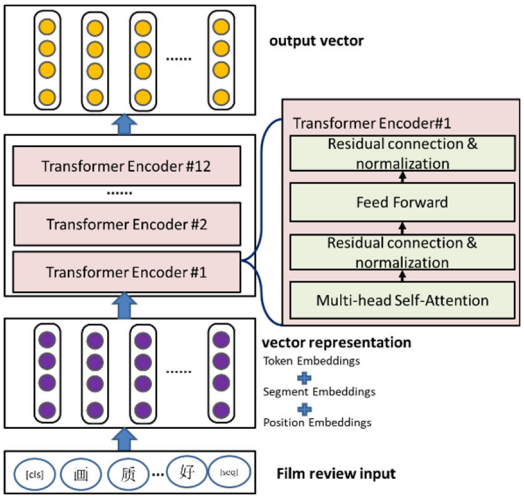

FIGURE 3. BERT model structure.

At the pre-training phase, the BERT model needs to perform two tasks: the masking language model and the subsequent sentence prediction. In the masking language model task, BERT, which works as similarly as completing a ''cloze'' test, predicts the missing words according to the context after randomly masking 15% of the words in each sentence. The 15% masking rate is not negligible, as it ensures that the model doesn't overly rely on specific words for downstream tasks. Masked words are substituted by ''[mask]'' 80% of the time and a random word 10% of the time, and keep unchanged 10% of the time. In the subsequent sentence prediction task, BERT checks whether the second sentence comes up behind the first to determine the sequential orders of pairs of interconnected sentences. These pairs consist of 50% of the sentences that were located originally in the subsequent place and 50% randomly selected from the corpus. Introduction of the ''subsequent sentence prediction'' task aims to align with natural language processing tasks in NLP that involve understanding the relationship between two sentences. By performing these two tasks, the pre-trained BERT model can output a word vector that can accurately and comprehensively capture the semantic message of the entire text.

As depicted in **FIGURE** 3, the BERT model follows a series of steps. First, the results from Token embedding, segment embedding, and position embedding are integrated. Then, the text is sequentially passed through 12 Transformer encoders as the input vector. In each encoder, there are 12 self-attention mechanisms comprising a multi-head self-attention module that produces and maps enhanced semantic vectors for each word in the text to varying semantic spaces. Later, these enhanced semantic vectors are processed by linear transformation, layer normalization, and random dropout before entering the fully connected layer. The ReLU function serves as the activation function. Following this, another round of linear transformation, layer normalization, and random dropout is performed. The output is an improved

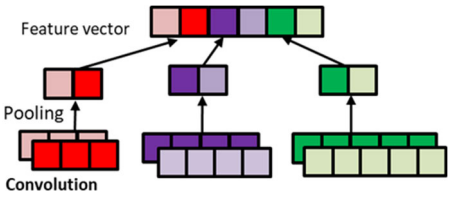

FIGURE 4. CNN-based local feature extraction.

semantic vector, of which the length is the same as that of the original word vector. This vector serves as the input for the subsequent Encoder and will be processed by the remaining 11 Transformer encoders.

The symbol W_i is used here to represent words in the text. Then, the BERT model transforms the word W_i into a dynamic word vector that are identifiable to the computer. Let S_j be a sentence composed of n words, k the dimension of the word vector, and $V(W_i)$ the K-dimensional word vector for the word W_i. The, sentence vector representation of the sentence S_j is shown in formula (1).

$$S_j = \{V(W_n)\} \in R_{n*k} \tag{1}$$

2) FEATURE EXTRACTION LAYER

According to the different characteristics of CNN network and RNN network, the feature extraction layer is composed of: CNN-based local feature extraction and Bi-LSTM -based global feature extraction. The specific structure is as follows:

(1) CNN-based local feature extraction

The Convolutional Neural Network (CNN) was initially developed as a type of feedforward neural network primarily designed for computer vision tasks. However, as research progressed, researchers began exploring its application in the field of Natural Language Processing (NLP) by leveraging convolutional windows and operations to extract local features from text, yielding promising results. One of the key characteristics of a CNN is its ability to share weights and establish local connections. Additionally, it allows for the utilization of different window sizes for multi-layer convolutional operations and downsampling. A CNN is usually made up of an input layer, convolutional layer, pooling layer, fully connected layer, and output layer. In this study, the CNN takes a sentence vector matrix S_j obtained by concatenating the input layers. Convolution and pooling operations are performed in CNN to represent the local features of the text. The specific structure is displayed in **FIGURE** 4.

① Convolution layer

Convolution layer extracts local features mainly through convolutional operations. Generally speaking, a CNN often with several convolutional layers can guarantee comprehensive extraction of text features through multi-layer convolutions. Each convolution layer can set filters of different sizes (convolution kernel) for different feature mappings of the same text in a bid to learn more feature information. When $r \times k$ size filters are adopted to extract the local features of the sentence S_j, the results obtained after convolutions can be

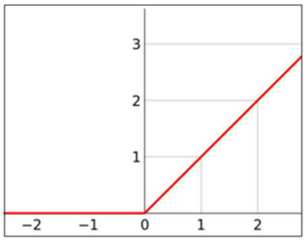

FIGURE 5. ReLU function image.

expressed as shown in formula (2).

$$C_i = f(A \cdot V(w_{i:i+r-1}) + b) \tag{2}$$

Here, A represents the filter parameter, indicating the convolutional operation; $V(w_{i:i+r-1})$ denotes the vector of r lines from i to $i+r-1$, b the bias quantity, C_i the local eigenvector obtained from i to $i+r-1$ through the convolutional operation, and $f(\cdot)$ the activation function. When the text features are collected by the CNN network, the width k of the filter is consistent with the size of the word vector.

In this study, STEP length is set to 1, and three convolution kerns of different sizes (3,4,5) are used to extract local features of sentences 3-gram, 4-gram and 5-gram respectively. To accelerate training convergence, the ReLU function serves as the activation function in this study, with its mathematical expression shown in formula (3).

$$f(x) = \max(0, x) \tag{3}$$

Activation functions are introduced in neural network models to capture the nonlinear relationships within the network. Common activation functions include Sigmoid, tanh, and ReLU functions, among others. The ReLU function is essentially a piecewise linear function, of which the graph is depicted in **FIGURE** 5. Through unilateral inhibition, the ReLU function allows the neurons inside the neural network to be sparsely activated. Specifically, all negative values and zeros are mapped to 0, while positive values remain unchanged. When deep classification models are trained, not all features are relevant to the target. Therefore, the model sparsely activated by the ReLU function can better identify relevant features and fit the training data more effectively. Another advantage of the ReLU function is that it does not exhibit saturation, and positive values are preserved without a need for performing complex exponential operations. Consequently, the ReLU function is free from the problem of gradient vanishing. In general, the ReLU function overmatches the Sigmoid and tanh functions in terms of the convergence speed. In this study, the ReLU function is chosen as the activation function.

After the filter performs convolutional operations on the sentence S_j from top to bottom in accordance with $STEP = 1$, the local eigenvector set C_{set} of the sentence S_j is finally obtained, which is expressed as formula (4).

$$C_{set} = \{C_1, C_2, \cdots, C_{i+r-1}\} \tag{4}$$

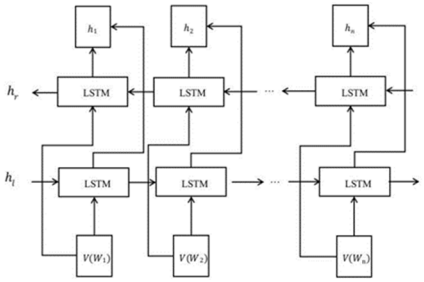

FIGURE 6. BiLSTM network model structure diagram.

② Pooling layer

The pooling layer is commonly known as the downsampling layer. Its primary purpose is to downsample the extracted features and preserve the essential information using specific methods. This downsampling process serves multiple purposes, including reducing the number of parameters, achieving feature dimension reduction, preventing overfitting, and enhancing the model's fault tolerance. Currently, the maximum pooling and average pooling are the two most popular pooling techniques. While the maximum pooling selects the maximum value from each region, the average pooling is used to figure up the average value within each pooling region. These pooling methods help capture the most salient features from the extracted feature maps while discarding less significant details.

By incorporating pooling layers into the model, the overall complexity of the network is reduced, and the model becomes more robust and efficient in handling variations and noise in the input data. In this study, the MaxPooling method is chosen to replace C_i with the maximum eigenvalue in the local eigenvector obtained by the convolutional operation, which can be expressed as formula (5).

$$d_i = \text{Max}(C_i) \tag{5}$$

Finally, the final feature representation result d_{set} of the CNN-based module for local feature extraction is obtained by fusing all the features obtained through convolution of all filters after pooling operation.

(2)Bi-LSTM global feature extraction

The Long Term Dependency and "Gradient Disappearance" issues with the typical RNN recurrent neural network are resolved by the LSTM, a unique type of recurrent neural network, which incorporates input, forgetting, output, and memory gates. In essence, BiLSTM is a two-layer LSTM structure. The structure of LSTM is displayed in **FIGURE** 6.

As can be seen from **FIGURE** 6, BiLSTM is made up of a layer of forward LSTM network structure and a layer of reverse LSTM structure. In **FIGURE** 6, $V(W_1), V(W_2), \cdots, V(W_n)$ is the input, h_l_l represents the forward LSTM, h_r represents the reverse LSTM, $h_1, h_2, \cdots, h_n$ is the output. If the output vector of BiLSTM hidden state at time t is h_t, the computational formula of h_t is shown as follows.

$$h_t = h_{lt} \oplus h_{rt} \tag{6}$$

where h_{lt} is the forward LSTM hidden layer output vector at time t, h_{rt} the output vector of the reverse LSTM hidden layer at time t, and the symbol $\oplus$ denotes vector concatenation.

By using Bi-LSTM to extract the global feature from sentence S_j and fusion, the final feature of the Bi-LSTM global feature extraction module is expressed as $\mathrm{BL}_{\mathrm{set}}$.

3) FEATURE FUSION LAYER

The feature fusion layer mainly fuses the local feature vectors d_{set} and $\mathrm{BL}_{\mathrm{set}}$ obtained by the feature extraction layer composed of CNN and Bi-LSTM modules. Finally, the global feature vector F that contains both the local features of the text is obtained. F is represented as shown below.

$$F = d_{\mathrm{set}} \oplus \mathrm{BL}_{\mathrm{set}} \tag{7}$$

where, the symbol $\oplus$ represents vector concatenation.

4) SELF ATTENTION MECHANISM WEIGHT ADJUSTMENT LAYER

Inspired by biological observation behavior, the attention mechanism is a mechanism which aligns internal experiences with external sensations to enhance the precision of focusing on specific areas. In the context of machine learning, the attention mechanism is employed to efficiently extract important features from sparse data. As a result, it has been widely applied to natural language processing tasks, particularly machine translation tasks. The self-attention mechanism, as an improved attention mechanism, is less dependent on external information and can effectively capture internal correlations among data or features. Consequently, it has gained popularity in various text classification tasks. In tasks such as emotion analysis, each word in the text contributes differently, with emotion-related words playing more critical role than ordinary words. By leveraging the self-attention mechanism, we can uncover internal relationships and similarities between words or phrases in the text, thereby facilitating the extraction of more representative text features. In this study, the self-attention mechanism is utilized for training and updating model parameters. This mechanism enables the model to focalize important elements within the text and enhance the ability to capture relevant information effectively. The Attention calculation formula is shown in equation (8).

$$\mathrm{Attention} = \mathrm{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \tag{8}$$

In self-attention, Q=K=V corresponds to the n-dimensional vector output by the feature fusion layer. In the above formula, $\sqrt{d_k}$ is the regularization factor that can prevent Softmax values from being either 0 or 1 due to excessive dot product values of $Q \cdot K^T$, when the results are normalized using the Softmax function.

5) FUll CONNECTION LAYER AND OUTPUT LAYER

The self-attention mechanism weight adjustment layer provides valuable features that are given into the fully connected layer of the model, which is eventually meant to resolve the classification issue. This allows the Softmax classifier to produce the final classification results. A Dropout mechanism, which discards some trained parameters during model training, is placed between the fully connected layer and the Softmax layer to prevent from overfitting.

TABLE 1. Part of movie review data.

Comment text content	Label
不懂太专业的，但一个故事简单纯粹就足够吸引我了!	1
特意注册豆瓣的号来给五星，哈哈……	1
只有面子，没有里子的所谓大片!	0
不知道为什么被发现吹的那么厉害真心觉得不怎么样。	0

6) BERT-CNN-BiLSTM-Att MODEL

To put it briefly, the BERT-CNN-BiLSTM-Att model extracts the dynamic word vector from the input text by segmenting it with the pre-training BERT model. It then feeds the vector into the CNN and BiLSTM models, respectively, extracting the local and global features for feature fusion. Finally, it combines with the attention mechanism to highlight the important information and realize sentiment classification of the Douban movie review dataset.

III. EXPERIMENTAL DESIGN AND RESULT ANALYSIS

A. EXPERIMENTAL DATA ACQUISITION AND PRE-PROCESSING

In this experiment, the Scrapy crawler is used to extract 28 movie review datasets from the Top250 Douban website for emotion categorization, and the data is stored as csv files. Data preparation is critical to deep learning model training, as it improves accuracy and reduces execution time. The data is pre-processed in the following four ways: 1) Determine the positive and negative sentiment of sentences. 2) Remove sentences with comment characters more than 30. 3) Remove comments that are all in English, all in characters, and all in emoticons. 4) Remove uncommon characters. After random extraction of the pre-processed data, the dataset of this experiment consists of 400,000 movie reviews. The labeled data sets are then split into training set, validation set, and test set in an 8:1 to 1:1 proportion, and each review is categorized as "positive" or "negative." Positive ratings are given a value of one, while negative reviews are given a value of zero. Some of the statistics are presented in **Table** 1. Finally, test sets are utilized to ensure that the model functions properly. This study employs various key toolkits that are designed for counting evaluation metrics and creating the workflow in a bid to evaluate and analyze the effectiveness of the proposed model.

B. EXPERIMENTAL PARAMETER SETTING

To carry out the experiment, a number of mathematical and statistical tools are used. After the mathematical model is established, Python3.8 programming language is used to code the model. To build and train the network model to operate

TABLE 2. Main parameter settings of the model.

Parameter Name	Parameter Value
Word vector dimension	768
Sliding window size	3,4,5
Activation function	ReLU/tanh
Pooling method	Max-Pooling
Dropout	0.5
Epoch	100
max_length	30
output_dim	2
Learning rate	0.0001
Loss function	Cross entropy
Optimizer	Adam
Hidden layer size	512
Batch-size	1024
head_count	8
head_dim	256
num_layer	8

and process the corpus, the nn module of Torch, especially its automatic derivation mechanism, is used in this study to build the neural network. The self-defined network can be constructed by rewriting the __init__ function and the forward function without a need for backpropagation. Torch's optim module provides several optimization algorithms for training models. With the data reader created by Torch Dataset and DataLoader, the data can be read and scrambled in batches automatically. To train the model, we import bert models, tokenizer encoders, configuration files, and optimizers from the transformers' library. Use pandas to read csv files for building data readers. Use the matplotlib library to complete the drawing. Regarding the hyperparameters, set the learning rate as 1e-4, and the number of training rounds as 1,000; utilize the Adam optimizer for optimization, and adopt the batch_norm approach for regularization.

The settings of experimental parameters include the parameters of BERT model, and the experimental parameters of CNN and BiLSTM-Att model. The model parameters are shown in **Table** 2.

C. EXPERIMENTAL EVALUATION INDEX

In this study, the model is tested on test data sets. The experimental effect is evaluated using three indicators: Accuracy (A), Recall (R) and F1 (F1-score). Accuracy means the percentage of the reviews accurately classified in the review text. Recall rate means the percentage of all real reviews that are correctly classified in the sample. The geometric mean divided by the arithmetic mean yields the value of F1. The formula for calculating the evaluation indicators is presented in formula (9) to (11).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{F1} = \frac{2 \times P \times R}{P + R} \tag{11}$$

TP and TN denote the number of positive and negative categories correctly predicted, respectively. FP and FN the number of positive and negative categories that are incorrectly predicted, respectively. In this experiment, the positive category is treated as the positive sample, and the negative category is treated as the negative sample to assess the precision rate of the model in emotion categorization.

TABLE 3. Experimental results.

Model Type	ACCURACY/%	Recall/%	F1-score/%
Word2Vec-BiLSTM	80.3	76.05	78.12
Word2Vec-CNN	80.56	77.85	78.89
Word2Vec-CNN-BiLSTM-Att	81.29	78.01	79.62
BERT	82.3	80.5	81.4
BERT-CNN	82.37	83.1	82.74
BERT-BiLSTM	79.39	78.15	79.17
BERT-CNN-BiLSTM-Att	84.93	86.55	85.73

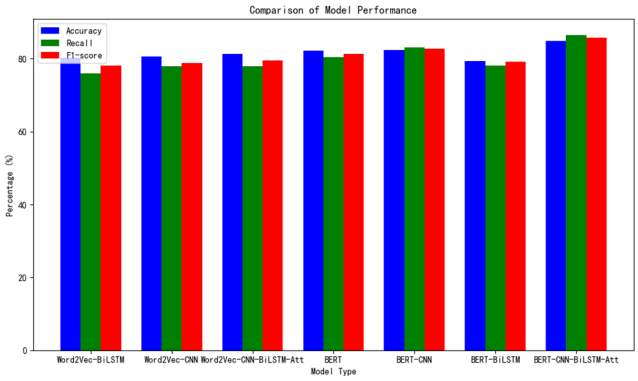

FIGURE 7. Comparison of experimental results.

D. EXPERIMENTAL RESULTS AND ANALYSIS

In this study, two separate pre-training models, namely Word2vec and BERT, are utilized to transform the word vector, and the two pre-training models are trained on the same data set, and the results of comparison demonstrate BERT's powerful capability of text representation. The models under consideration include Word2Vec-BiLSTM, Word2Vec-CNN, Word2Vec-CNN-BiLSTM-Att, BERT, BERT-CNN, BERT-BiLSTM, BERT-CNN-BiLSTM-Att, and so on. The impact of various word vectors and models on emotion categorization accuracy is determined through a comparison of emotion categorization accuracies to validate whether the proposed BERT-CNN-BiLSTM-Att model is more accurate in terms of emotion categorization than the traditional deep learning models.

Movie review data set consists of 80% training set, 10% validation set and 10% test set. In this study, the proposed model is trained using the training set before being assessed using the test set. BERT selects features based on word context and updates the term vector dynamically as context changes. The comprehensive index demonstrates that the BERT pre-training model is more advantageous to the proposed model in terms of extracting text information. As a result, a BERT model with high text representation capability is chosen for this model to realize text feature representation. The results could be found in **Table** 3.

TABLE 4. Experimental results on jingdong.

Model Type	ACCURACY/%	Recall/%	F1-score/%
Word2Vec-BiLSTM	88.5	86.1	87.2
Word2Vec-CNN	90.7	89.5	90.0
Word2Vec-CNN-BiLSTM-Att	91.2	92.0	91.6
BERT	91.0	90.5	90.7
BERT-CNN	92.5	91.3	91.9
BERT-BiLSTM	91.7	94.5	93.1
BERT-CNN-BiLSTM-Att	95.7	93.5	94.6

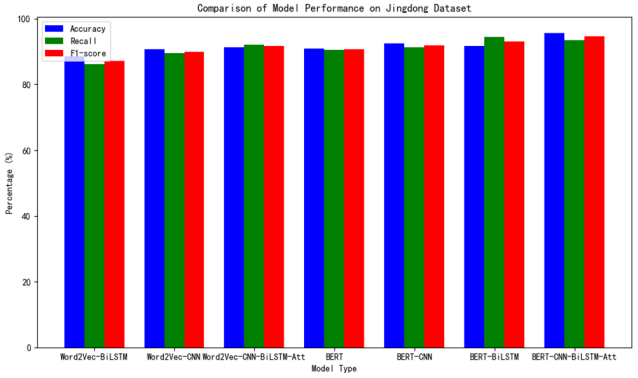

FIGURE 8. Comparison of experimental results on Jingdong.

When it comes to verifying the accuracy of the model constructed in this study, the proposed method is found to yield achieved remarkable results. To represent the word vector, this study selects a BERT model that can use context information of words to extract features and dynamically adjust word vectors according to different context information. When processing time series, conventional LSTM tends to ignore the previous context information. BiLSTM can process text from both directions. However, not all words contribute equally to text analysis, and the attention mechanism can highlight important words in the text. The model constructed in this study is compared with Word2Vec-BiLSTM,Word2Vec-CNN,Word2Vec-CNN-BiLSTM-Att, BERT,BERT-CNN, BERT-BiLSTM, BERT-CNN-BiLSTM-Att. The comparison results are displayed in **FIGURE** 7.

As sown in **FIGURE** 7, the BERT-CNN-BiLSTM-Att model has a 2.63% higher accuracy than the BERT model, because the BERT-CNN-BiLSTM-Att model integrates the advantages of BERT, CNN, BiLSTM and attention mechanism. This proves that a multi-model feature enhancement approach works better than a single-model one. The BERT model has the shortcomings of overfitting and poor generalization ability because of many parameters involved in the pre-training process, and less change of internal parameters in subsequent natural language processing tasks. In addition, **FIGURE** 7 also reveals a higher experimental accuracy of BERT as the pre-training language model, compared to Word2vec. This means that BERT's pre-training model is powerful. It has been demonstrated that both the self-attention mechanism and the multihead self-attention mechanism are capable of efficiently integrating the pertinent context-specific information, or even the entire phrase, into the current word vector. The precision rate of BERT-CNN-BiLSTM-Att model goes up by 5.58% compared to BERT-BiLSTM model. The model employs BiLSTM to extract global features concurrently with local features extracted by CNN. To make emotion categorization more accurate, an attention mechanism that gives varying weights to different areas of the text is added.

To validate the BERT-CNN-BiLSTM-Att model's generalization capacity, 4000 Jingdong reviews are crawled, and then divided in a proportion of 8:1:1 for training, validation, and testing. The comparison results of the various models are displayed in **Table** IV and **FIGURE** 8.

In summary, the BERT-CNN-BiLSTM-Att model can more accurately classify emotional reviews than the previous six models to some extent, indicating that the BERT pre-training model is more conducive to the model's extraction of text information. Therefore, this study combines BERT, CNN, BiLSTM and attention mechanisms to form a combined model. The experimental findings show that the BERT-CNN-BiLSTM-Att model is superior to other models in emotion categorization.

IV. CONCLUSION

In this study, a BERT-CNN-BiLSTM-Att sentiment analysis model is proposed to analyze short text movie reviews. It is trained on Douban movie review dataset to verify its effectiveness. This study assesses the effectiveness of the proposed model with three indicators: precision rate, recall rate, and F1 value. To summarize, BERT-CNN-BiLSTM-Att has the highest emotion categorization accuracy among the comparison models. Performance of these models can be improved by using together with other machine learning techniques. Through this study, we can expect to extend the same techniques to a wide range of deep learning applications and investigate how different model-combined approaches work on small amounts of data when combined with appropriate deep learning strategies to achieve more satisfactory results. As this study only considers sentiment analysis of binary classified texts rather than multiclassified texts, a further work on sentiment analysis of multiclassified texts will be performed to get richer semantic text information. And we can expand the scope of the related data to cover colorful characters, emoticons, secluded characters, etc. To summarize, the model can be further explored through a variety of experiments to obtain emotion categorization results in a more accurate and faster manner. Thus, merchants can get real customer feedback through these reviews, finding out the product's problems, and formulating reasonable sales strategies to produce greater economic results.

REFERENCES

[1] P. Racherla and W. Friske, "Perceived 'usefulness' of online consumer reviews: An exploratory investigation across three services categories," *Electron. Commerce Res. Appl.*, vol. 11, no. 6, pp. 548–559, Nov. 2012, doi: 10.1016/j.elerap.2012.06.003.

[2] C. Yang, X. Wang, M. Li, and J. Li, "Research on fusion model of BERT and CNN-BiLSTM for short text classification," in *Proc. 4th Int. Conf. Comput. Eng. Appl. (ICCEA)*, Hangzhou, China, Apr. 2023, pp. 525–529, doi: 10.1109/ICCEA58433.2023.10135222.

[3] Z. Quan, T. Sun, M. Su, and J. Wei, "Multimodal sentiment analysis based on cross-modal attention and gated cyclic hierarchical fusion networks," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–12, Aug. 2022, doi: 10.1155/2022/4767437.

[4] Y. Zheng, Y. Long, and H. Fan, "Identifying labor market competitors with machine learning based on maimai platform," *Appl. Artif. Intell.*, vol. 36, no. 1, Dec. 2022, Art. no. 2064047, doi: 10.1080/08839514.2022.2064047.

[5] A. Hassan and A. Mahmood, "Deep learning approach for sentiment analysis of short texts," in *Proc. 3rd Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2017, pp. 705–710, doi: 10.1109/ICCAR.2017.7942788.

[6] A. S. Raamkumar and Y. Yang, "Empathetic conversational systems: A review of current advances, gaps, and opportunities," *IEEE Trans. Affect. Comput.*, vol. 14, no. 4, pp. 1–20, Dec. 2022, doi: 10.1109/TAFFC.2022.3226693.

[7] S. Niklander and G. Niklander, "Combining sentimental and content analysis for recognizing and interpreting human affects," in *Communications in Computer and Information Science*. Cham, Switzerland: Springer, 2017, pp. 465–468.

[8] C. Baziotis, N. Pelekis, and C. Doulkeridis, "DataStories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis," in *Proc. 11th Int. Workshop Semantic Eval.*, 2017, pp. 747–754, doi: 10.18653/v1/s17-2126.

[9] N. Jia and C. Zheng, "Multimodal emotion recognition by integrating audio, text and expression," *Chin. J. Appl. Sci.*, vol. 41, no. 1, pp. 55–70, 2023, doi: 10.3969/j.issn.0255-8297.2023.01.005.

[10] V. Rizzello, B. Böck, M. Joham, and W. Utschick, "Reverse ordering techniques for attention-based channel prediction," *IEEE Open J. Signal Process.*, vol. 5, no. 4, pp. 248–256, Jul. 2024, doi: 10.1109/OJSP.2023.3344024.

[11] M. Fahmi, F. Yudanto, N. Nazhifah, Y. Sari, and Afiahayati, "Deep learning approach for aspect-based sentiment analysis on Indonesian hospitals reviews," in *Proc. 8th Int. Conf. Informat. Comput. (ICIC)*, Dec. 2023, pp. 1–6, doi: 10.1109/icic60109.2023.10381908.

[12] X. Zhang, S. Huang, J. Zhao, X. Du, and F. He, "Exploring deep recurrent convolution neural networks for subjectivity classification," *IEEE Access*, vol. 7, pp. 347–357, 2019, doi: 10.1109/ACCESS.2018.2885362.

[13] R. Man and K. Lin, "Sentiment analysis algorithm based on BERT and convolutional neural network," in *Proc. IEEE Asia–Pacific Conf. Image Process., Electron. Comput. (IPEC)*, Apr. 2021, pp. 769–772, doi: 10.1109/IPEC51340.2021.9421110.

[14] A. N. Azhar and M. L. Khodra, "Fine-tuning pretrained multilingual BERT model for Indonesian aspect-based sentiment analysis," in *Proc. 7th Int. Conf. Advance Inform. Concepts, Theory Appl. (ICAICTA)*, Sep. 2020, pp. 1–6, doi: 10.1109/ICAICTA49861.2020.9428882.

[15] L. Kryeziu and V. Shehu, "Pre-training MLM using BERT for the albanian language," *SEEU Rev.*, vol. 18, no. 1, pp. 52–62, Jun. 2023, doi: 10.2478/seeur-2023-0035.

[16] G. S. Chauhan, R. Nahta, Y. K. Meena, and D. Gopalani, "Aspect based sentiment analysis using deep learning approaches: A survey," *Comput. Sci. Rev.*, vol. 49, Aug. 2023, Art. no. 100576, doi: 10.1016/j.cosrev.2023.100576.

[17] S. Tabinda Kokab, S. Asghar, and S. Naz, "Transformer-based deep learning models for the sentiment analysis of social media data," *Array*, vol. 14, Jul. 2022, Art. no. 100157, doi: 10.1016/j.array.2022.100157.

[18] H. Chen and Y. Chen, "Research on microblog sentiment analysis based on multimodal and multiscale feature fusion," in *Proc. 4th Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE)*, May 2022, pp. 605–608, doi: 10.1109/CISCE55963.2022.9851054.

[19] S. Imron, E. I. Setiawan, J. Santoso, and M. Hery Purnomo, "Aspect based sentiment analysis marketplace product reviews using BERT, LSTM, and CNN," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 3, pp. 586–591, Jun. 2023, doi: 10.29207/resti.v7i3.4751.

[20] Y. Chen and W. Lin, "Sentiment analysis of Chinese micro-blog by combining BERT-BiGRU and multi-scale CNN," *J. Chin. Acad. Electron. Sci.*, vol. 18, no. 1, pp. 939–945, 2023.

[21] H. Deng, D. Ergu, F. Liu, Y. Cai, and B. Ma, "Text sentiment analysis of fusion model based on attention mechanism," *Proc.Comput. Sci.*, vol. 199, pp. 741–748, Jan. 2022, doi: 10.1016/j.procs.2022.01.092.

[22] A. Sen, G. Rajakumaran, M. Mahdal, S. Usharani, V. Rajasekharan, R. Vincent, and K. Sugavanan, "Live event detection for people's safety using NLP and deep learning," *IEEE Access*, vol. 12, pp. 6455–6472, 2024, doi: 10.1109/ACCESS.2023.3349097.

[23] Ł. Maziarka, D. Majchrowski, T. Danel, P. Gaiński, J. Tabor, I. Podolak, P. Morkisz, and S. Jastrzębski, "Relative molecule self-attention transformer," *J. Cheminformatics*, vol. 16, no. 1, pp. 1–14, Jan. 2024, doi: 10.1186/s13321-023-00789-7.

[24] S. Mo, H. Wang, B. Li, Z. Xue, S. Fan, and X. Liu, "Powerformer: A temporal-based transformer model for wind power forecasting," *Energy Rep.*, vol. 11, pp. 736–744, Jun. 2024, doi: 10.1016/j.egyr.2023.12.030.

[25] V. Mashurov, V. Chopuryan, V. Porvatov, A. Ivanov, and N. Semenova, "Gct-TTE: Graph convolutional transformer for travel time estimation," *J. Big Data*, vol. 11, no. 1, p. 15, Jan. 2024, doi: 10.1186/s40537-023-00841-1.

[26] A. U. Rahman, Y. Alsenani, A. Zafar, K. Ullah, K. Rabie, and T. Shongwe, "Enhancing heart disease prediction using a self-attention-based transformer model," *Sci. Rep.*, vol. 14, no. 1, p. 514, Jan. 2024, doi: 10.1038/s41598-024-51184-7.

AIXIANG HE was born in Susong, Anqing, Anhui, China, in 1978. She received the degree in electronic technology from Hefei Union University, China, in 2001, the bachelor's degree in e-commerce from Beijing Normal University, China, in 2008, and the master's degree in computer application from Anhui University, China, in 2011. She is currently pursuing the Ph.D. degree with the School of Computer and Information Technology, National University, Manila, Philippines.

Since 2001, she has been an Associate Professor with Anhui Xinhua University. She presided more than three quality engineering projects of Anhui Provincial Department of Education and one major online teaching reform research project of Anhui Province. She has published more than ten research articles in domestic and foreign journals, including three by EI and one by CSCD. Her research interests include machine learning and natural language processing.

MIDETH ABISADO received the master's degree in information technology from the Technological University of the Philippines in 2004, the Master of Science degree in computer science from Mapua University in 2016, and the Doctorate degree in information technology from the Technological Institute of the Philippines in 2019. She is a Professor with the Graduate School Department, College of Computing and Information Technologies, National University, Manila. She has worked in artificial intelligence research, with a particular interest in affective computing, natural language processing, and image processing. She has contributed to over 50 research papers in various respected journals and conferences. She is also the Principal Investigator of a major research project on Philippine AI-Powered Disease Surveillance using Social Media Analytics, funded by the Philippine Department of Science and Technology.