

Research Article

GloVe-CNN-BiLSTM Model for Sentiment Analysis on Text Reviews

Li Xiaoyan ^{1,2}, Rodolfo C. Raga,³ and Shi Xuemei¹

¹College of Computer, Huainan Normal University, Huainan 232000, China

²College of Computing and Information Technologies, National University, Manila, Philippines

³Computer Studies and Engineering Department, Jose Rizal University, Mandaluyong City, Philippines

Correspondence should be addressed to Li Xiaoyan; lxy0722_321@163.com

Received 25 May 2022; Revised 10 August 2022; Accepted 8 October 2022; Published 22 October 2022

Academic Editor: Rosalba Liguori

Copyright © 2022 Li Xiaoyan 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.

Nowadays, social media networks generate a tremendous amount of social information from their users. To understand people's views and sentimental tendencies on a commodity or an event timely, it is necessary to conduct text sentiment analysis on the views expressed by users. For the microblog comment data, it is always mixed with long and short texts, which is relatively complex. Especially for long text data, it contains a lot of content, and the correlation between words is more complex than that in short text. To study the sentiment classification of these mixed texts composed of long-text and short-text, this research proposes an optimized GloVe-CNN-BiLSTM-based sentiment analysis model. In this model, GloVe is used to vectorize words, and CNN is given to represent part space character. BiLSTM is used to build temporal relationship. Twitter's comment data on COVID-19 is used as an experimental dataset. The results of the experiments suggest that this method can effectually identify the sentimental tendency of users' online comments, and the accuracy of sentiment classification on complete-text, long-text, and short-text can achieve to 0.9565, 0.9509, and 0.9560, respectively, which is obviously higher than other deep learning models. At the same time, experiments show that this method has good field expansion.

1. Introduction

Sentimental analysis (SA) [1], sometimes named opinion mining or sentiment artificial intelligence, is a significant task in natural language processing. Text sentiment analysis is the process of analyzing, processing, induction, and reasoning the subjective text with sentiment color. It categorized the text as positive, negative, neutral, or conflicting sentimental polarity. With the rapid development of the Internet, people increasingly like to express their opinions on an object or event on Internet social platforms (Twitter, Facebook, microblog, etc.), such as goods purchased in online stores, newly released movies, epidemic development, and even some hot events. People's comments on these often contain sentiments and sentimental tendencies. If we can collect these comments and analyze the users' sentimental tendencies in these comments, we can understand the users' word-of-mouth and provide strong support for subsequent decision-making. Government agencies can also collect data

on social platforms, analyze people's views on some policies and public hot events, and take timely countermeasures to maintain social stability. This article will focus on sentiment analysis of COVID-19 reviews on Twitter which have been wreaking havoc all over the world these years.

The traditional algorithms of sentiment analysis are categorized into several categories: approaches based on machine learning (such as support vector machine, naive Bayes, maximum entropy, and k -nearest neighbor algorithm), dictionary-based methods, and hybrid methods [2]. Kang et al. [3] proposed an improved naive Bayes classifier, when the accuracy of the two classes was expressed as the average value, and it addresses the problem of reducing the average accuracy. Chen and Tseng [4] used two multiclass methods based on SVM: one-versus-all SVM and single-machine multiclass SVM to classify comments; this method can classify comments accurately with high quality. He and Zhou [5] proposed a strategy to realize the sentiment classification task by using feature-level supervision rather than

instance-level supervision, and the priori information extracted from the existing emotion dictionary is combined with the sentiment classifier model learning to obtain the initial classifier. This method outperforms existing weakly supervised sentiment classification algorithms and can be used to classify text with relevant prior knowledge. Polignano et al. [6] studied the issue of analyzing social media traces to determine an individual's propensity for empathy. Utilizing information from social media, they employ linear regression algorithms to forecast the user's level of empathy. The findings demonstrate a significant relationship between empathy and personality attributes. In the field of sentiment analysis, the effect of the above text sentiment classification model is not satisfactory. It is difficult to adapt to the rich language environment. It is also hard to be applied on a large scale and heavily depends on the feature selection strategy and the optimization of model parameters. In computer vision and natural language processing, as an extended field of machine learning, deep learning has been extensively used.

The basic structure of deep learning is actually deep neural network, deep learning model transforms the raw data into even higher hierarchical abstract representations through some simpler nonlinear models, and it can learn complex function features through combined multilayer transformation. Therefore, as comparison to the conventional machine learning model, deep learning can be better applied to sentiment classification tasks and raise the property of the model. The existing deep neural network methods mainly include CNN (conventional neural network), RNN, and LSTM (long short-term memory). CNN can efficiently capture the local sentiment information of the text when used in the field of sentiment analysis, but it is hard to consider the long-distance dependence of the text and the order of the text. As a temporal recursive neural network, LSTM integrates the text through sequence, which apply to conduct and forecast milestone with comparatively long interval and delay in time series. Considering that LSTM can only use the text's forward information, but not the text's backward information, a bidirectional long short-term memory network is proposed to incorporate text context information into the model at the same time, improving the model's prediction performance. However, because of the large input dimension, using the BiLSTM model directly may result in considerable computing overhead. In this paper, CNN and BiLSTM are combined. CNN is used to create a pool layer, which is further transmitted to LSTM along the pipeline. It can lower the dimension of the original data's word vector matrix and then integrate the BiLSTM model for sentiment analysis. Thereby, the model's operational efficiency and forecast accuracy can be improved much more. Experiments with the CNN-BiLSTM model suggested in this research will be conducted on COVID-19 online review on Twitter. By extracting important words from twitter and embedding words, some words are difficult to extract their essence, such as satire and irony. Here, we can further divide words into multiple regions and use convolution layers to extract further features to solve them. In this paper, the GloVe model is used for word embedding.

TABLE 1: Related work.

| References | Model |
|------------|----------------------|
| Ref [8] | CNN |
| Ref [9] | CNN+RNN+GloVe |
| Ref [10] | SDCNN |
| Ref [11] | LSTM+GloVe |
| Ref [12] | BIGRU+CNN |
| Ref [13] | 2D CNN-LSTM |
| Ref [14] | Two-channel CNN-LSTM |
| Ref [15] | MLP+CNN+LSTM |
| Ref [16] | BiLSTM+CNN |

The remainder of this article is organized as follows: Section 2 proposes a short literature review of neural models for sentiment analysis and text classification. Section 3 presents the related methods and describes the proposed model in detail. Experimental results are presented in Section 4. Finally, Section 5 summarizes this paper and puts forward some directions for future research.

2. Related Work

With the proposal of the concept of deep learning, the research of sentiment analysis has ushered in a new stage of development. Polignano and Basile [7] demonstrated HAnSEL with a system built on a group of classifiers, including the support vector machine algorithm, random forests, and a multilayer perceptron deep neural network. The authors formalized communications as a concatenation of word2vec phrase vectors and a TF-IDF bag of words. Rani and Kumar [8] used CNN to study sentiment analysis in different languages. Variable numbers of convolution layers are used in the experiments, along with different numbers and sizes of filters. Abid et al. [9] created a hybrid architecture that used RNN first to capture long-term dependencies with CNN utilizing a global average pooling layer, and GloVe was obtained by unsupervised learning in light of sizable Twitter corpora. Fan et al. [10] proposed a SDCNN model which was built based on convolutional neural network with sparse dropout, when compared to CNN, and SDCNN enhances the model's classification performance much more. For sentiment identification in Twitter, Chatterjee et al. [11] suggested SS-BED, a multichannel LSTM model; in this model, GloVe is employed in parallel as pretrained word embeddings, and three LSTM modules are used to address long text dependencies. Alotaibi et al. [12] introduced the multichannel deep learning framework, which combines the bidirectional gated recurrent unit (BiGRU), transformer block, and convolutional neural network (CNN) to classify Twitter comments into two categories: both aggressive and passive aggressive. Zhao et al. [13] put forward a 2D CNN-LSTM network to recognize emotion, it comprises of four local feature learning blocks (LFLBs) and one LSTM layer, and the experimental results reveal that the constructed network performed well in speech sentiment recognition task. Li et al. [14] proposed a two-channel CNN-LSTM family

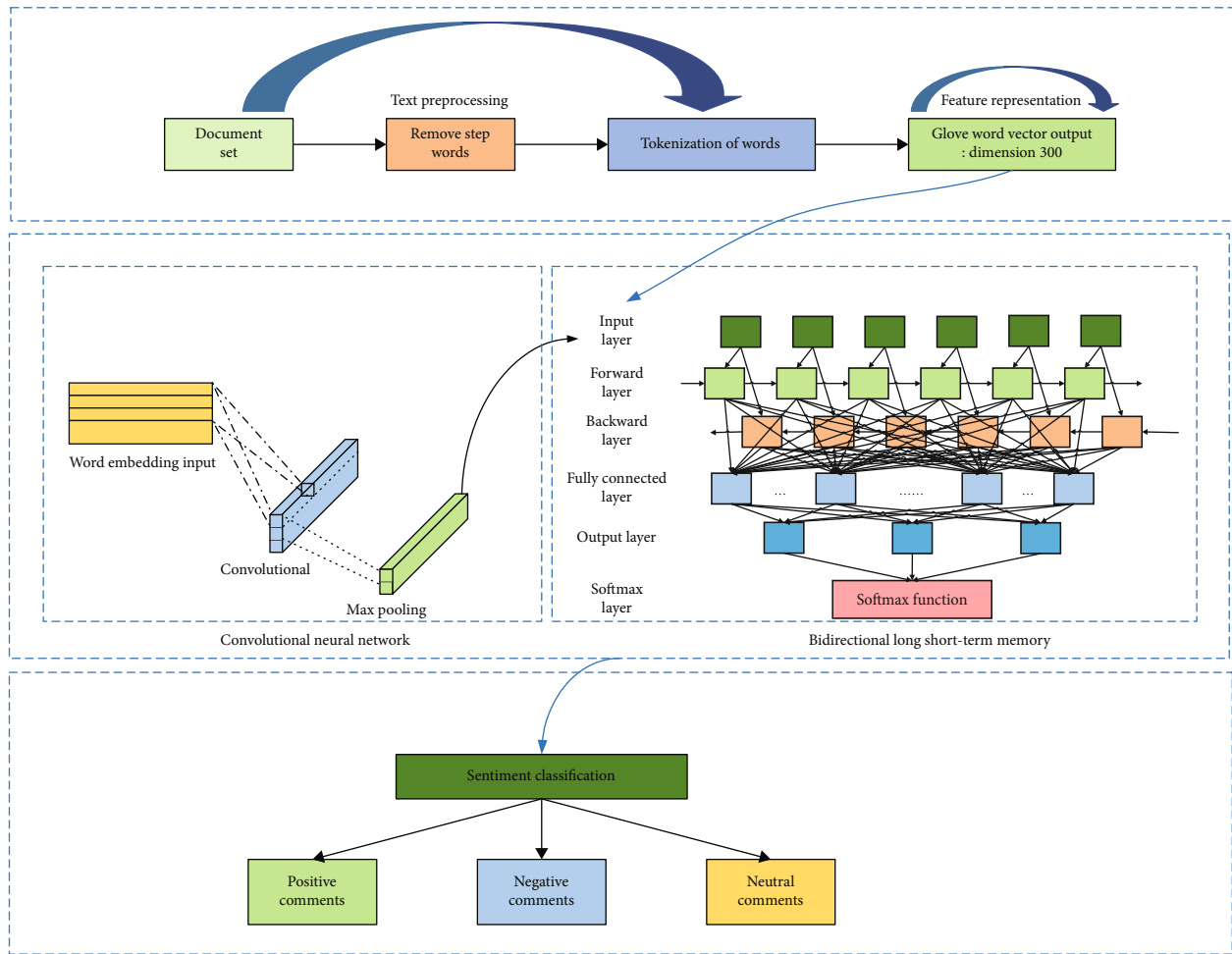


FIGURE 1: Proposed model.

model for dictionary integration and experiments on some challenging dataset, such as Stanford Sentiment Treebank; the results reveal that the proposed strategy outperforms several standard approaches. Munandar et al. [15] employed a hybrid neural network architecture to classify sentiment in multidomain short messages, MLPs (multilayer perceptron), CNNs, and LSTMs are used to build the architecture; the results of the experiments suggest that the proposed model may effectively address classification challenges in natural language processing. Polignano et al. [16] propose a sentiment classification model based on BiLSTM and CNN deep neural network. The model is mediated by a certain degree of self-attention. The authors used three word embedding methods to experiment on three datasets, and the experiment results show that the FastText vector space allows obtaining the best results for the identification of the emotion. Comparative table about above related works is shown in Table 1.

To classify text accurately, appropriate text feature representation is particularly important. Text feature representation can compress the dimension of text word vector space on the premise of correctly identifying the feature words of text content and distinguish different types of text through

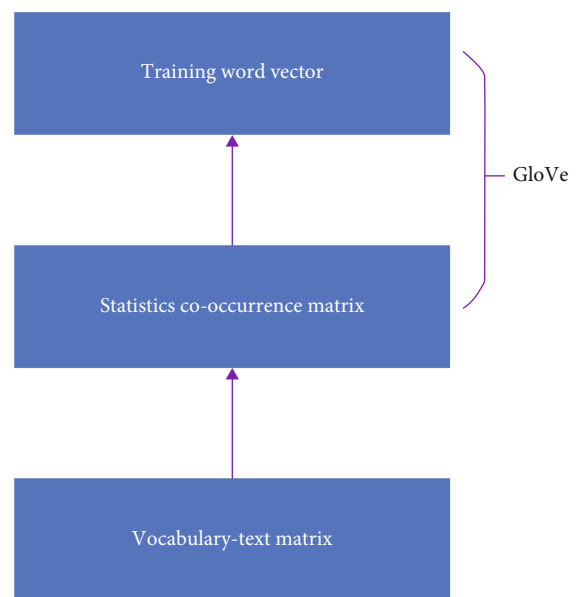


FIGURE 2: GloVe model.

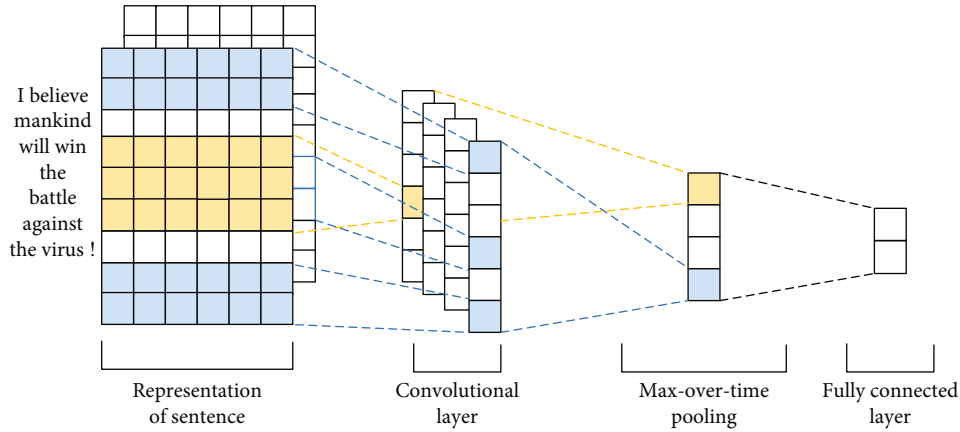


FIGURE 3: CNN sentiment feature extraction model.

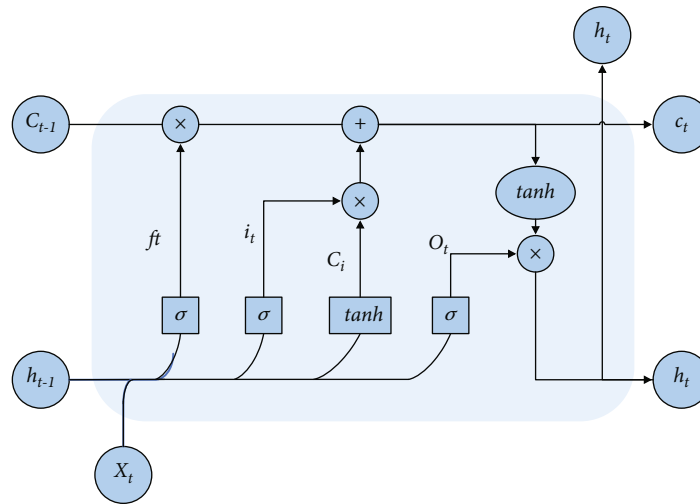


FIGURE 4: LSTM.

feature items. In practice, word vectorization is often used for semantic feature representation. At present, there are two types of widely used: one is based on global matrix decomposition, such as LSA [17], and the other is the local context window, such as the skip gram used by word2vec. Among them, the main advantage of LSA is to use statistical information for semantic analysis, but its effect on lexical analogy is poor. Although word2vec has good lexical analogy performance, it is limited by the characteristics of local windows and is hard to use the global lexical cooccurrence statistics effectively. GloVe combines the advantages of the above two, combines the global statistical information with the local context window, and has a better effect of word vectorization. Yanan and Dagang [18] compared GloVe with the word2vec word vector for text feature extraction and then used SVM for text classification. Through experiments, it is proved that GloVe has better effect in text classification. In addition, the Bert [19] model released by Google in 2018 has achieved the best results in 11 classic NLP tasks and has become a highly sought-after word vector model. Of course, there are still some outstanding problems in the practical

application of Bert. These problems need to be discussed by researchers through further experiments.

Inspired by the above methods of feature extraction using word vectorization and constructing classification model using neural network, this paper proposes a text sentiment analysis model combining GloVe with CNN-BiLSTM. In the feature representation stage, the GloVe word vector can lower the dimensionality of text properties, and then the sentiment classification model is constructed by combining the CNN-BiLSTM model to realize the purpose of making full use of text context to construct classification model. Experiments show that this method can obtain better classification effect.

3. Methodology

Taking online comments of COVID-19 as an example, this paper constructs a sentiment classification model based on GloVe-CNN-BiLSTM. First, we should do text processing, such as removing stop words, lemmatization, and tokenization of words. Word vectorization is carried

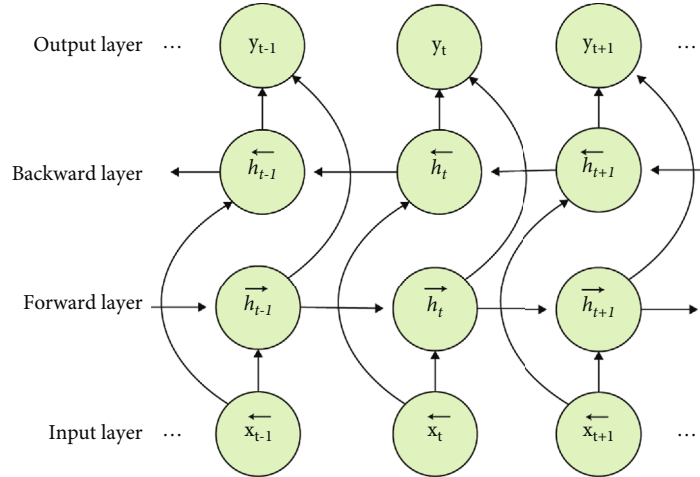


FIGURE 5: BiLSTM.

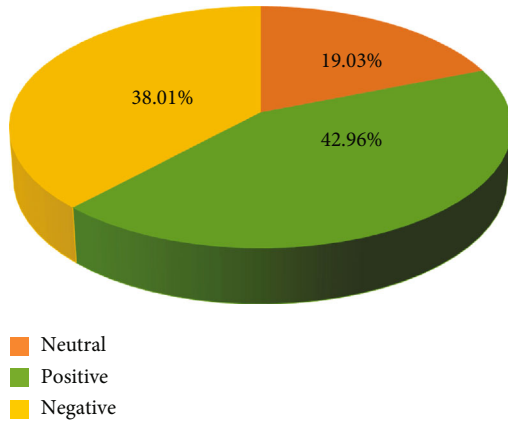


FIGURE 6: Data distribution.

out through GloVe, which contains as much text semantic and grammatical information as possible, while reducing the dimension of vector space. Then, the neural network model CNN-BiLSTM is constructed for training. It can not only take advantage of CNN to extract local features but also take advantage of BiLSTM to consider the global features of text sequence. Figure 1 shows the architecture of the model, and it is mainly separated into two sections: text representation and CNN-BiLSTM-based sentiment classification model building.

3.1. GloVe Model. The GloVe model is an effective method to make use of global corpus statistics and optimize the learning model based on context window. Its main goal is to vectorize words and output word vectors through input corpus. The implementation method is as follows: first, construct a word cooccurrence matrix based on the entire corpus; next, the learning word vector is processed due to the cooccurrence matrix and GloVe model. The GloVe model is shown in Figure 2.

The GloVe model can be described by the following formula:

$$J = \sum_{i,j}^N f(X_{ij}) (V_i^T V_j + b_i + b_j - \ln(X_{ij}))^2, \quad (1)$$

where X is the cooccurrence matrix and the number of times the words i and j that appear together in one window is represented by the element X_{ij} . The window size is generally 5~10, and V_i and V_j represent the word vector of word i and word j . b_i and b_j are the deviation term, N is the dimension of cooccurrence matrix $N \times N$, and f is the weight function, where f must have the following characteristics:

When the cooccurrence number of words is 0, the weight is also 0, that is, $f(0) = 0$.

- (1) When the cooccurrence number of words is greater, its weight will not decline, that is, $f(x)$ satisfies the continuity and non decrement
- (2) When words appear too frequently, there will be no over weighting, that is, $f(x)$ can be assigned a relatively small value. To sum up, the weight function $f(x)$ has the following formula:

$$f(x) = \begin{cases} (x/x_{\max})^\alpha, & x < x_{\max} \\ 1, & x \geq x_{\max} \end{cases} \quad (2)$$

The effect was better through the experiment when $x_{\max} = 100$, $\alpha = 0.75$, by formula (1), GloVe can directly use the corpus word vector of the document by itself for calculation, and it has strong maneuverability and high flexibility.

3.2. Convolutional Neural Network. The convolutional neural network model which is a feedforward neural network is used to extract topic salient features from text context features. CNN's structure is divided into three sections: The

| | User name | Screen name | Location | TweetAt | OriginalTweet | Sentiment | sentiment | length |
|-------|-----------|-------------|------------------------------|------------|---|--------------------|--------------------|--------|
| 0 | 3799 | 48751 | London | 16-03-2020 | @MeNyrbie @Phil_Gahan @Christiv https://t.co/i... | Neutral | Neutral | 111 |
| 1 | 3800 | 48752 | UK | 16-03-2020 | advice Talk to your neighbours family to excha... | Positive | Positive | 237 |
| 2 | 3801 | 48753 | Vagabonds | 16-03-2020 | Coronavirus Australia: Woolworths to give elde... | Positive | Positive | 131 |
| 3 | 3804 | 48756 | ÚT: 36.319708, -82.363649 | 16-03-2020 | As news of the regions first confirmed COVID... | Positive | Positive | 250 |
| 4 | 3805 | 48757 | 35.926541, -78.753267 | 16-03-2020 | Cashier at grocery store was sharing his insig... | Positive | Positive | 180 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 81691 | 44951 | 89903 | Wellington city, New zealand | 14-04-2020 | Airline pilots offering to stock supermarket s... | Neutral | Neutral | 102 |
| 81692 | 44952 | 89904 | NaN | 14-04-2020 | Response to complaint not provided citing COVI... | Extremely negative | Extremely negative | 138 |
| 81693 | 44953 | 89905 | NaN | 14-04-2020 | You know its getting tough when @KameronWild... | Positive | Positive | 136 |
| 81694 | 44954 | 89906 | NaN | 14-04-2020 | Is it wrong that the smell of hand sanitizer i... | Neutral | Neutral | 111 |
| 81695 | 44955 | 89907 | i love you so much he/him | 14-04-2020 | @TartiiCat Well new/used Rift S are going for ... | Negative | Negative | 255 |

81696 rows × 8 columns

FIGURE 7: Instance of dataset.

TABLE 2: Parameters' setting of GloVe-CNN-BiLSTM.

| Parameters | Value |
|--|--------------------------|
| Word vector dimension | 300 |
| Convolution layer filters | 50 |
| Number of convolution kernels | 256 |
| Convolution layer kernel size | 2 |
| Convolution layer padding | Valid |
| Convolution layer activation function | ReLU |
| Pooling layer pool size | 1 |
| Pooling layer activation function | ReLU |
| Number of hidden units in BiLSTM layer | 64 |
| Dropout rate | 0.7, 0.8 |
| Loss | categorical_crossentropy |
| Optimizer | Adam |
| Learning rate | 0.0001 |

TABLE 3: Instance of short-text from dataset.

| |
|---|
| Coronavirus fun fact: if you cough at the grocery store, you get the whole aisle to yourself pretty quickly. |
| Went to the supermarket yesterday and the toilet paper was gone. Has this anything to do with the Corona virus? |
| This is the line outside @target in as customers wait for the store to open this morning |
| My thoughts on impacts of coronavirus on food markets |
| My work is capitalizing on the demand for packaged food and making us stay open as opposed to closing for all our health and safety |
| There is in the country; the more empty shelves people see, the more buying ensues the more food is out of stock |

S_{ij} by means of step size 1, the set of local feature vectors C is finally obtained:

$$C = \{c_1, c_2, \dots, c_{r-h+1}\}. \quad (4)$$

The maximum pooling approach is used to extract the feature with the highest value to replace the entire local feature acquired by the convolution operation, and the size of the feature vector can be considerably decreased by pooling operation:

$$d_i = \max C. \quad (5)$$

Finally, at the full connection layer, all pooled features are integrated, yielding the following output vector D :

$$D = \{d_1, d_2, \dots, d_n\}. \quad (6)$$

The CNN sentiment feature extraction model is shown in Figure 3.

3.3. BiLSTM. The LSTM model which is shown in Figure 4 is a time cyclic neural network that was created to address the general RNN's long-term reliance issue. Compared with ordinary cyclic neural network, LSTM adds gate cells to RNN, which can be divided into input gates according to functions output gate and forge gate that are collectively referred to as long-term and short-term memory units. LSTM unit can remember the value in any time interval, and three gating units control the information flow in and

input layer is the first portion, the convolution layer and pool layer are the second part, the fully linked multilayer perceptual classifier is the third part, and the core of CNN is the second part. The CNN model is shown in Figure 3. Suppose a comment text $W = \{w(1), w(2), \dots, w(n)\}$, in the remark text W , the word $w(i)$ is translated into the matching word vector $V(w(i))$ by GloVe, and a sentence matrix S_{ij} is created from the sentence made by word $w(i)$ here.

$S_{ij} = \{V(w(1)), V(w(2)), \dots, V(w(i))\}, 1 \leq i \leq n$. In the CNN model, S_{ij} is the input of the convolution layer, and the convolution layer uses a filter of size $r \times k$ to convolute the sentence matrix S_{ij} and extract the local semantic properties of S_{ij} . The calculating formula is as follows:

$$c_{ij} = f(F \times V(w(i : i + r - 1)) + b), \quad (3)$$

where F is the filter of $r \times k$, f is the ReLU nonlinear conversion, $V(w(i : i + r - 1))$ is the r -line word vector from i to $i + r - 1$ in S_{ij} , b is the offset, and c_{ij} is the local semantic characteristics of the j -th sentence made of i words extracted via CNN. As the filter gliders through the whole

TABLE 4: Instance of long-text from dataset.

My wife work's retail& a customer came in yesterday, coughing everywhere, saying they have COVID-19. They requested a deep clean of the store—her company objected to due to cost, recommending the team spray disinfectant& clean themselves. We are going to die/get sick due to capitalism

For corona prevention, we should stop to buy things with the cash and should use online payment methods because corona can spread through the notes. Also, we should prefer online shopping from our home. It is time to fight against COVID-19? #govindia #IndiaFightsCorona

Curious, do we think retail shoppers will do a lot of online shopping because they are home and unable to go out or do we think everyone is too spooked to get that extra pair of shoes? #economy #onlineshopping #coronavirus #covid19 #stayhome

Advice talk to your neighbours family to exchange phone numbers create contact list with phone numbers of neighbours schools employer chemist GP set up online shopping accounts if possible adequate supplies of regular meds but not over order

With 100 nations infected with COVID-19, the world must not play fair with China. 100 governments must demand China to adopt new guidelines on food safety; the Chinese government is guilty of being irresponsible with life on a global scale to closing for all our health and safety

@10DowningStreet @grantshapps what is being done to ensure food and other essential products are being re-stocked at supermarkets and panic buying actively discouraged? It cannot be left to checkout staff to police the actions of the selfish and profiteer

TABLE 5: Comparison results of three models.

| Model | Accuracy | F1-score | Loss function |
|------------------|----------|----------|---------------|
| GloVe-CNN-BiLSTM | 0.9565 | 0.9553 | 0.2588 |
| CNN-BiLSTM | 0.9528 | 0.9514 | 0.2862 |
| TextCNN | 0.9429 | 0.9416 | 0.2598 |

out of the unit. It is this advantage of selective reading and writing information that greatly makes up for the defects of gradient explosion and gradient disappearance.

In the long short-term memory neural network part, for the input gate i , output gate o , and forget gate f at time t , there are the following operating formulas, respectively:

$$\begin{aligned}
 i_t &= \sigma(W^i x_t + U^i h_{t-1}), \\
 f_t &= \sigma(W^f x_t + U^f h_{t-1}), \\
 o_t &= \sigma(W^o x_t + U^o h_{t-1}), \\
 c'_t &= \tan h(W^c x_t + U^c h_{t-1}), \\
 c_t &= i_t \times c'_t + f_t \times c'_{t-1}, \\
 h_t &= o_t \times \tan h(c_t).
 \end{aligned} \tag{7}$$

Among them, W^i , W^f , W^o , W^c , U^i , U^f , U^o , and U^c are all weight matrices, and the LSTM structure diagram is as follows:

Although LSTM solves the long-term dependency problem, it is hard to utilize the contextual information of the text. The model design concept of BiLSTM is to make the feature data obtained at time t have information between the past and the future at the same time. Experiments have shown that this neural network structure model has better text feature extraction efficiency and performance than a single LSTM structure model. In text sentiment classification, BiLSTM also considers the context of the text, and uses

the output of the CNN pooling layer as the input of two LSTM networks with opposite time series. The forward LSTM can obtain the above information of the input sequence, and the backward LSTM can obtain the above information of the input sequence. The context information of the input sequence is then calculated by vector splicing to obtain the final hidden layer representation. It is worth mentioning that the LSTM neural network parameters in BiLSTM are independent of each other, and they only share the word-embedding word vector list. The BiLSTM model is shown by Figure 5.

4. Experimental Study

4.1. Experimental Dataset. The experimental data in this paper is the COVID-19 comment dataset, and we use Python to grab COVID-19 comment data from Twitter through the method of web crawler. The dataset has a total of 81696 rows and 8 columns of comments, including 35093 positive comments, 31060 negative comments, and 15543 neutral comments. The dataset is divided into the training set and test set, and the ratio is 8:2. The data distribution is shown by Figure 6, and the instance of dataset is shown by Figure 7.

4.2. Experimental Parameter Setting. To improve the performance of the sentiment analysis model of the comment content, this paper adjusts the super parameters of the constructed GloVe-CNN-BiLSTM model, among which the parameters with great influence mainly include the window size of CNN filter, the dimension of GloVe word vector, the number of filters in convolution layer, and the output dimension of BiLSTM. The GloVe-CNN-BiLSTM neural network model is constructed by the Python language and Tensorflow2 deep learning framework. The operating system of our experiments is Windows 10, the processor is Intel(R) Core(TM) i9-10900K CPU @3.70 GHZ 3.70 GHZ, and GPU is GeForce RTX 3080. The optimal parameter settings are shown in Table 2.

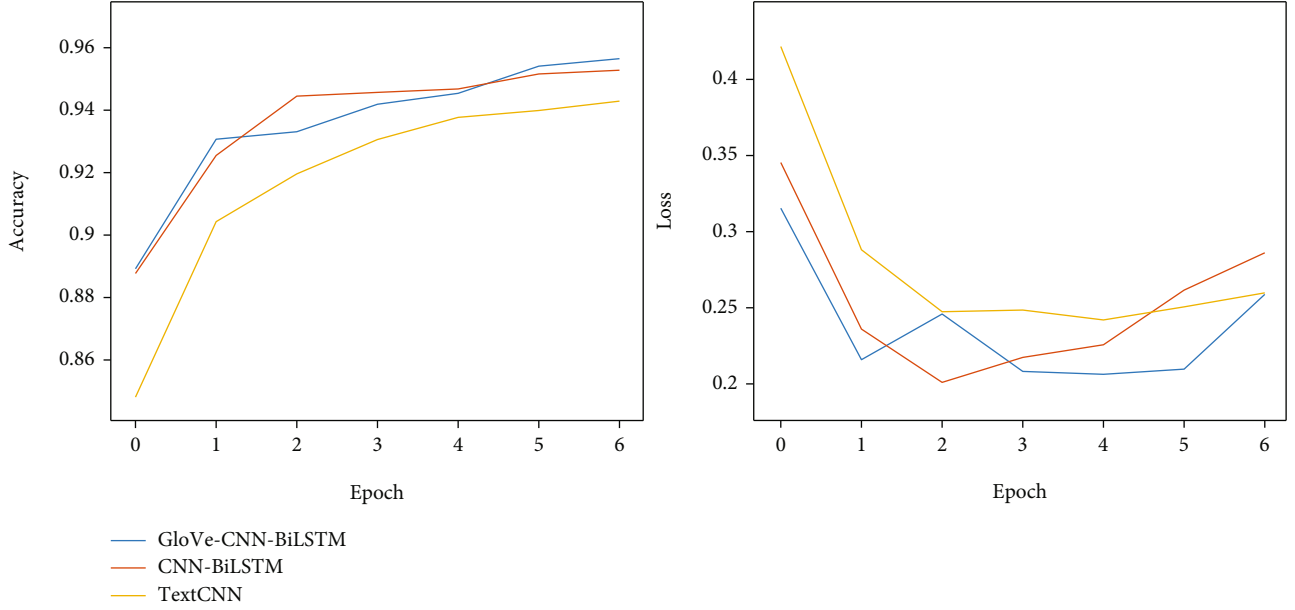


FIGURE 8: Accuracy-loss of the GloVe-CNN-BiLSTM model, CNN-BiLSTM model, and TextCNN model.

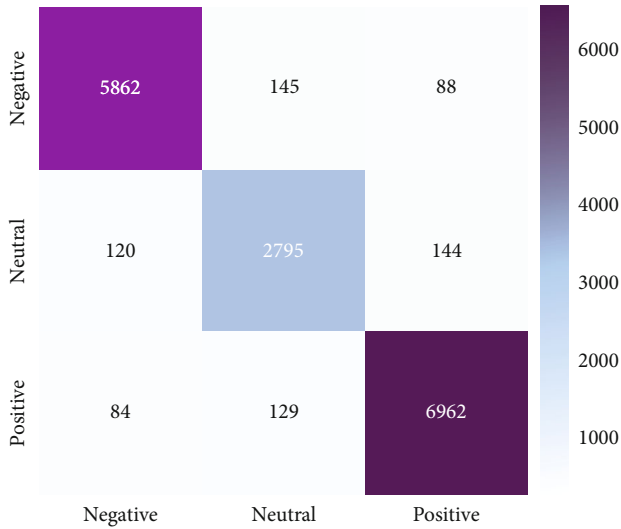


FIGURE 9: Confusion matrix of the GloVe-CNN-BiLSTM model.

TABLE 6: Comparison results of three models.

| Model | Accuracy | F1-score | Loss function |
|------------------|----------|----------|---------------|
| GloVe-CNN-BiLSTM | 0.9509 | 0.9586 | 0.2553 |
| CNN-BiLSTM | 0.9464 | 0.9572 | 0.3270 |
| TextCNN | 0.9226 | 0.9315 | 0.3264 |

4.3. Experimental Evaluation Index. In this paper, accuracy, F1-score and loss function are used to be evaluation indexes of experiments. For a given test dataset, accuracy refers to the ratio of the number of samples correctly classified by the classifier to the total number of samples, that is, the accuracy of the test dataset when the loss function is 0-1. The loss

function is used to measure the prediction of the model, and the lower the loss function, the better the model. Generally, the concerned class is regarded as the positive class, and other classes are regarded as the negative class. The prediction of the classifier on the test dataset is either correct or incorrect. The total number of the four cases is recorded as follows:

TP: predict the positive class as the number of positive classes

FN: predict the number of positive classes to the number of negative classes

FP: predict negative classes as positive classes

TN: predict the number of negative classes as the number of negative classes

$$\text{Precision} : p = \frac{TP}{TP + FP}, \quad (8)$$

$$\text{Recall} : R = \frac{TP}{TP + FN}.$$

F1-score is the harmonic average of precision and recall:

$$F1 - \text{score} = \frac{2PR}{P + R} = \frac{2TP}{2TP + FP + FN}. \quad (9)$$

4.4. Experimental Results and Analysis. In the experimental part of this paper, three experiments are conducted, namely, complete-text sentiment analysis, long-text sentiment analysis, and short-text sentiment analysis. First, text should be split by length, short text is defined with a length of less than 170, and long text is defined with a length of 170 to 300. After splitting, there are 54167 long-text data and 27529 short-text data. Complete-text refers to the original text without splitting. Then evaluation indexes of the proposed

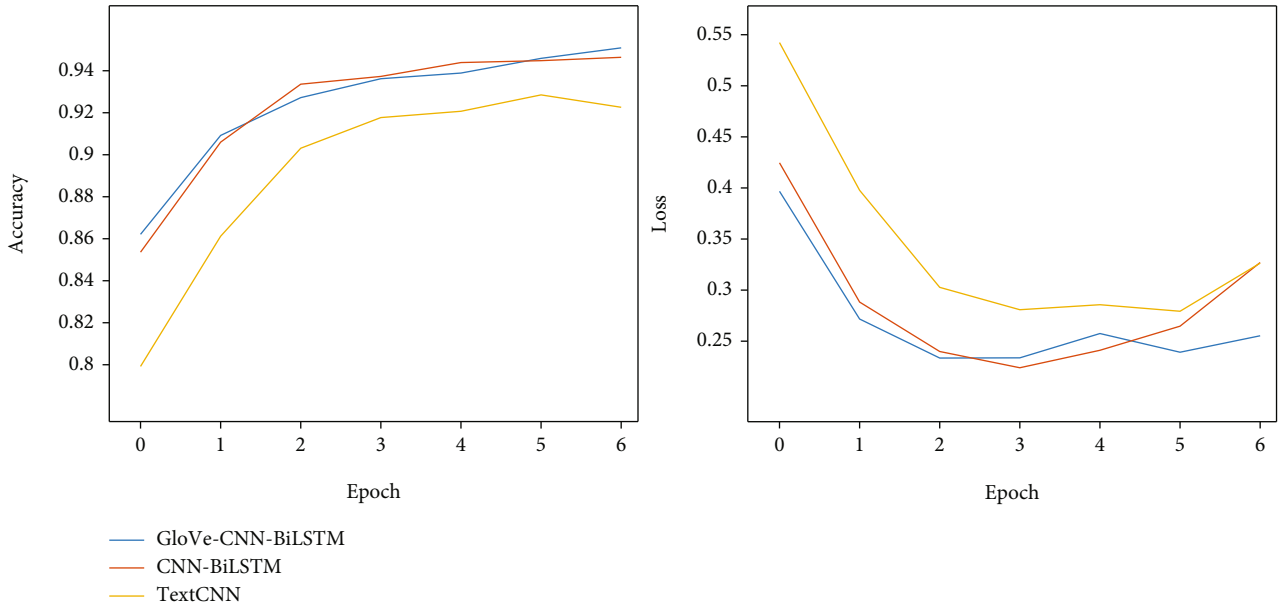


FIGURE 10: Accuracy-loss of the GloVe-CNN-BiLSTM model, CNN-BiLSTM model, and TextCNN model.

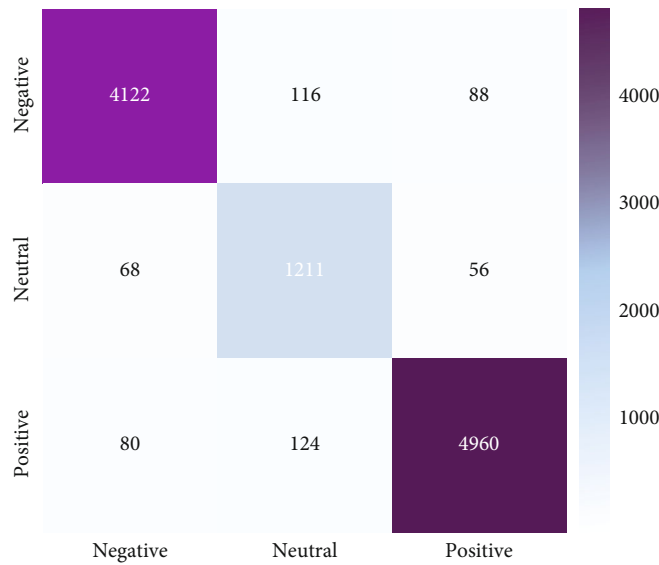


FIGURE 11: Confusion matrix of the GloVe-CNN-BiLSTM model.

TABLE 7: Comparison results of three models.

| Model | Accuracy | F1-score | Loss function |
|------------------|----------|----------|---------------|
| GloVe-CNN-BiLSTM | 0.9560 | 0.9489 | 0.2640 |
| CNN-BiLSTM | 0.9499 | 0.9476 | 0.2712 |
| TextCNN | 0.9480 | 0.9468 | 0.3039 |

model are calculated on complete-text dataset, long-text dataset, and short-text dataset, respectively. Other deep learning models (CNN-BiLSTM, TextCNN) are used to do comparative experiments. The instance of short-text is shown in Table 3.

The instance of long-text is shown in Table 4.

4.4.1. *Experiment on Complete-Text Dataset.* The experimental results on complete-text dataset are shown in Table 5 and Figures 8 and 9. Table 5 shows comparison of the three models. Figure 8 shows the accuracy and loss of three models on the test sets. From Table 5 and Figure 8, it is observed that the accuracy in GloVe-CNN-BiLSTM model is higher than CNN-BiLSTM model and TextCNN model, and the loss function in GloVe-CNN-BiLSTM is lower than CNN-BiLSTM model and TextCNN model. It shows that the robust performance of GloVe-CNN-BiLSTM model is better than the CNN-BiLSTM model and TextCNN model. The confusion matrix of GloVe-CNN-BiLSTM model is shown in Figure 9; as a visualization tool, confusion matrix can be used to evaluate the classification accuracy. The test

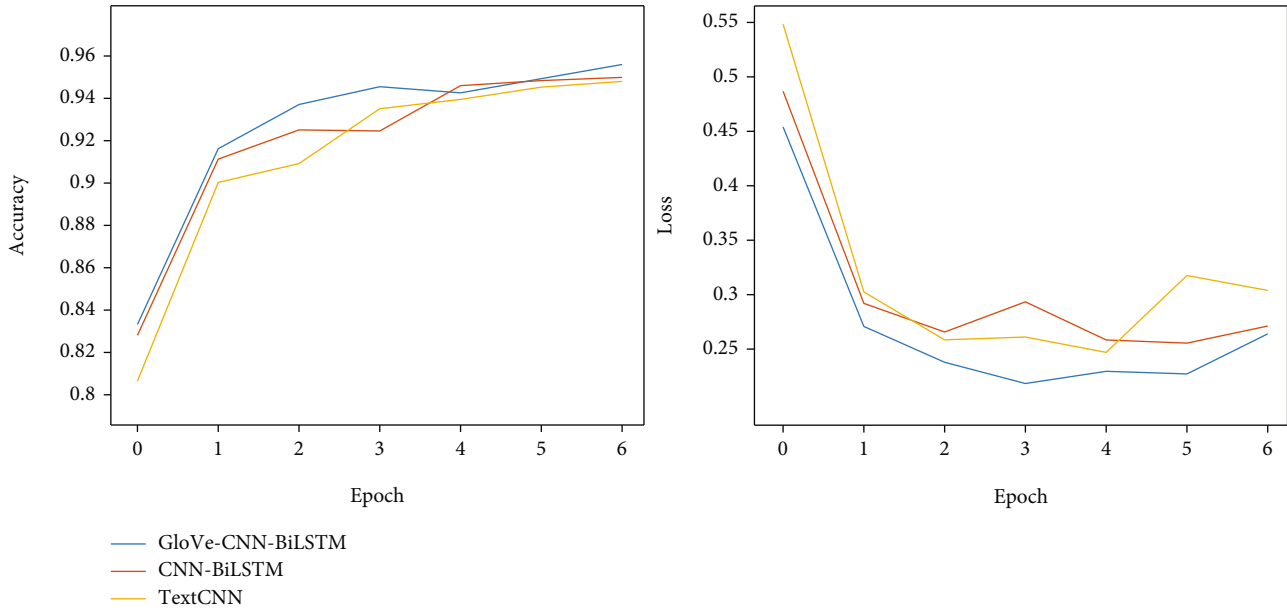


FIGURE 12: Accuracy-loss of the GloVe-CNN-BiLSTM model, CNN-BiLSTM model, and TextCNN model.

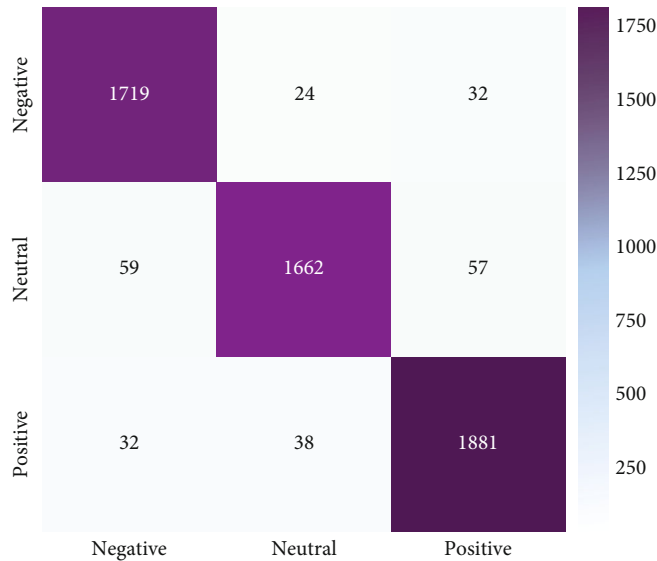


FIGURE 13: Confusion matrix of the GloVe-CNN-BiLSTM model.

dataset is 16329, the positive comments are 7175, the neutral comments are 3059, and the negative comments are 6095. From the confusion matrix, it can be seen that 6962 comments are predicted as positive comments, 2795 comments are predicted as neutral comments, and 5862 comments are predicted as negative comments. Combining Table 5 and Figures 8 and 9, it shows that a better performance is acquired in the GloVe-CNN-BiLSTM model.

4.4.2. *Experiment on Long-Text Dataset.* The experimental results on long-text dataset are shown in Table 6 and Figures 10 and 11. Table 6 shows the comparison of the three models. Figure 10 shows the accuracy and loss of three models on the test sets. From Table 6 and Figure 10, it is

observed that the accuracy in GloVe-CNN-BiLSTM model is higher than the CNN-BiLSTM model and TextCNN model, and the loss function in GloVe-CNN-BiLSTM is lower than the CNN-BiLSTM model and TextCNN model. It shows that the robust performance of GloVe-CNN-BiLSTM model is better than the CNN-BiLSTM model and TextCNN model. The confusion matrix of GloVe-CNN-BiLSTM model is shown in Figure 11; as a visualization tool, confusion matrix can be used to evaluate the classification accuracy. The test dataset is 10825, the positive comments are 5164, the neutral comments are 1335, and the negative comments are 4326. From the confusion matrix, it can be seen that 4122 comments are predicted as positive comments, 1211 comments are predicted as neutral comments,

and 4960 comments are predicted as negative comments. Combining Table 6 and Figures 10 and 11, it shows that a better performance is acquired in the GloVe-CNN-BiLSTM model.

4.4.3. Experiment on Short-Text Dataset. The experimental results on short-text dataset are shown in Table 7 and Figures 12 and 13. Table 7 shows comparison of the three models. Figure 12 shows the accuracy and loss of three models on the test sets. From Table 7 and Figure 12, it is observed that the accuracy in the GloVe-CNN-BiLSTM model is higher than the CNN-BiLSTM model and TextCNN model, and the loss function in GloVe-CNN-BiLSTM is lower than CNN-BiLSTM model and TextCNN model. It shows that the robust performance of the GloVe-CNN-BiLSTM model is better than the CNN-BiLSTM model and TextCNN model. The confusion matrix of GloVe-CNN-BiLSTM model is shown in Figure 13; as a visualization tool, confusion matrix can be used to evaluate the classification accuracy. The test dataset is 5504, the positive comments are 1951, the neutral comments are 1778, and the negative comments are 1775. From the confusion matrix, it can be seen that 1719 comments are predicted as positive comments, 1662 comments are predicted as neutral comments, and 1881 comments are predicted as negative comments. Combining Table 7 and Figures 12 and 13, it shows that a better performance is acquired in the GloVe-CNN-BiLSTM model.

From the results of the upper experiments, it can be demonstrated that the model proposed in this paper has obvious advantages; this is due to the strong feature extraction ability and nonlinear fitting ability of deep learning, which greatly improves the prediction performance of deep learning model. In addition, compared with other deep learning models, the model proposed in this paper also has significant superiority, which is due to the GloVe model and the timing characteristics of the BiLSTM model. BiLSTM is the combination of forward LSTM and backward LSTM, which can contact the relationship of context, so it can have better prediction performance. On the other hand, the proposed model uses CNN to reduce the dimension of features, so it can extract features more effectively and use them for sentiment analysis, which further improves the prediction accuracy and operation efficiency of the BiLSTM model.

5. Conclusions

Nowadays, the research on sentiment classification of online comments has always been one of the important tasks of NLP. For sentiment classification, feature extraction and classifier design are particularly important. To solve the sentiment classification of Twitter online comments text which is mixed with long and short text, this paper proposed the optimized GloVe-CNN-BiLSTM model. We use pretrained GloVe word-embedding vectors as the initial weights of the embedding layer, then CNN-BiLSTM is used to construct the sentiment analysis model of online comments. This paper uses Twitter's COVID-19 comment dataset to verify

the experimental results. We conduct experiments on complete-text dataset, long-text dataset, and short-text dataset, respectively. The experimental results show that the accuracy of the GloVe-CNN-BiLSTM model can achieve 0.9565 on complete-text dataset, 0.9509 on long-text dataset, and 0.9560 on short-text dataset, which is much higher than the CNN-BiLSTM model and TextCNN model. The sentiment analysis of online comments helps government departments to timely grasp the public's views on some political events and public opinion guidance and formulate correct relevant policies. Therefore, the model proposed in this paper has important practical significance. Also, the model proposed in this paper has good domain expansibility. In the field of marketing, companies use it to develop strategies to understand how customers feel about products or brands, how people react to their campaigns or product launches, and why consumers do not buy certain products. It is helpful for enterprises to improve their product sales. In the future work, we will apply our new method to the sentiment analysis of the Chinese online comments text which is also mixed with long-text and short-text.

Data Availability

The data used to support the results of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declared that there is no conflict of interest in publishing this paper.

Acknowledgments

This research was supported by the Special Research Project of Humanities and Social Sciences of the Ministry of Education (Grant No. 18JDSZ3039).

References

- [1] B. Liu, "Sentiment analysis and opinion mining!," *Synthesis Lectures on Humzan Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.
- [2] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: a survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.
- [3] H. Kang, S. J. Yoo, and D. Han, "Senti-lexicon and improved naive Bayes algorithms for sentiment analysis of restaurant reviews," *Expert Systems with Applications*, vol. 39, no. 5, pp. 6000–6010, 2012.
- [4] C. C. Chen and Y. D. Tseng, "Quality evaluation of product reviews using an information quality framework," *Decision Support Systems*, vol. 50, no. 4, pp. 755–768, 2011.
- [5] Y. He and D. Zhou, "Self-training from labeled features for sentiment analysis," *Information Processing Management*, vol. 47, no. 4, pp. 606–616, 2011.
- [6] M. Polignano, P. Basile, G. Rossiello, M. de Gemmis, and G. Semeraro, "Learning inclination to empathy from social media footprints," in *Proceedings of the 25th Conference on*

- User Modeling, Adaptation and Personalization*, pp. 383-384, Bratislava, Slovakia, 2017, July.
- [7] M. Polignano and P. Basile, "Hansel: Italian hate speech detection through ensemble learning and deep neural networks," *EVALITA Evaluation of NLP and Speech Tools for Italian*, vol. 12, p. 224, 2018.
- [8] S. Rani and P. Kumar, "Deep learning based sentiment analysis using convolution neural network," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3305–3314, 2019.
- [9] F. Abid, M. Alam, M. Yasir, and C. Li, "Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter," *Future Generation Computer Systems*, vol. 95, pp. 292–308, 2019.
- [10] X. Fan, A. Runa, Z. Pei, and M. Jiang, "An improved convolutional neural network for text classification," *Journal of Physics: Conference Series*, vol. 2066, no. 1, article 012091, 2021.
- [11] A. Chatterjee, U. Gupta, M. K. Chinnakotla, R. Srikanth, M. Galley, and P. Agrawal, "Understanding emotions in text using deep learning and big data," *Computers in Human Behavior*, vol. 93, pp. 309–317, 2019.
- [12] M. Alotaibi, B. Alotaibi, and A. Razaque, "A multichannel deep learning framework for cyberbullying detection on social media," *Electronics*, vol. 10, no. 21, p. 2664, 2021.
- [13] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," *Biomedical Signal Processing and Control*, vol. 47, pp. 312–323, 2019.
- [14] W. Li, L. Zhu, Y. Shi, K. Guo, and E. Cambria, "User reviews: sentiment analysis using lexicon integrated two-channel CNN–LSTM family models," *Applied Soft Computing*, vol. 94, article 106435, 2020.
- [15] D. Munandar, A. F. Rozie, and A. Arisal, "A multi domains short message sentiment classification using hybrid neural network architecture," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 2181–2191, 2021.
- [16] M. Polignano, P. Basile, M. de Gemmis, and G. Semeraro, "A comparison of word-embeddings in emotion detection from text using BiLSTM, CNN and self-attention," in *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, pp. 63–68, Larnaca, Cyprus, 2019, June.
- [17] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.
- [18] Z. Yanan and T. Dagang, "Research on text classification based on GloVe and SVM," *Software Guide*, vol. 17, no. 6, pp. 45–48, 2018.
- [19] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, arxiv: 1810.04805.