

Research Article

Sentiment Analysis Method of Network Text Based on Improved AT-BiGRU Model

Xinxin Lu¹ and Hong Zhang²

¹School of Computer Science and Technology, Zhoukou Normal University, Zhoukou 466001, Henan, China

²School of Network Engineering, Zhoukou Normal University, Zhoukou 466001, Henan, China

Correspondence should be addressed to Xinxin Lu; luxinxin@zknu.edu.cn

Received 4 December 2020; Revised 13 January 2021; Accepted 7 April 2021; Published 24 May 2021

Academic Editor: Wenzheng Bao

Copyright © 2021 Xinxin Lu and Hong Zhang. 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.

In order to solve the problems existing in the current method of emotional analysis of network text, such as long training time, complex calculation, and large space cost, this paper proposes an Internet text sentiment analysis method based on the improved AT-BiGRU model. Firstly, the textblob package is imported to correct spelling errors before text preprocessing. Secondly, pad_sequences are used to fill in the input layer with a fixed length, the two-way gated recurrent network is used to extract information, and the attention mechanism is used to highlight the key information of the word vector. Finally, the GNU memory unit is transformed, and an improved BiGRU that can adapt to the recursive network structure is constructed. The proposed model is experimentally demonstrated on the SemEval-2014 Task 4 and SemEval-2017 Task 4 datasets. Experimental results show that the proposed model can effectively avoid the text sentiment analysis bias caused by spelling errors and prove the effectiveness of the improved AT-BiGRU model in terms of accuracy, loss rate, and iteration time.

1. Introduction

With the continuous development and progress of information technology, the Internet has entered a period of rapid development. People have gradually completed the transition from information acquirers to manufacturers, and they are more and more inclined to publish online shopping, news media, reading, and watching movies on online platforms. From other perspectives, these sentimental evaluations seem insignificant, but they actually contain rich emotional information. Quickly and accurately analyzing and extracting expressions of emotional tendencies from massive review data has very important reference value and research value for government public opinion monitoring, corporate market research, and personal consumption choices. Sentiment analysis [1–4] mainly refers to the use of natural language processing and computer linguistics techniques to identify and extract subjective information in the original material and to

find out the bipolar views and attitudes of opinion speakers on certain topics.

The most important thing in sentiment analysis research is sentiment classification technology. Emotion classification is mainly to identify and classify the emotions expressed in the text, such as positive, negative, and so on, to obtain potential information. At present, the most researched emotion classification technologies are mainly divided into the following three types. The first method is based on the emotion dictionary [5]. The method based on sentiment dictionary is mainly to segment documents, find words of different parts of speech, and calculate their corresponding scores. This method relies too much on the emotional dictionary and has serious domain characteristics, and the effect is not ideal [6]. The second method is based on manual feature extraction. The method based on manual feature extraction is a traditional machine learning method, which requires a large amount of prelabeled data. Then, machine learning algorithms such as support vector machine [7],

Naive Bayes, and conditional random fields are used for emotion classification. Among them, the conditional random field is a discriminative probability learning model, which has good effects in sequence labeling, named entity recognition, and Chinese word segmentation. The third method is based on deep learning [8–11]. This type of deep neural network-based model has achieved better results than traditional classifiers in the field of sentiment analysis.

However, most of the current text sentiment analysis models do not take into account the problem of spelling errors, and different text comment data may appear different in length, which is difficult to transform into a unified word vector. At the same time, mainstream text sentiment analysis models take a long time to train and cannot fully extract text features. In response to the above problems, a network text sentiment analysis method based on the improved AT-BiGRU model is proposed. Its innovation points are as follows:

- (i) For spelling errors that are not considered by traditional sentiment analysis models, the `textblob` package is imported to correct before text preprocessing, which improves text sentiment analysis errors caused by spelling errors.
- (ii) We fill the input layer with `pad_sequences` to a fixed length, use a two-way gated recurrent network to extract information, and use the attention mechanism to highlight the key information of the word vector. The problems of complex calculation and high space cost of most text sentiment analysis models are improved.
- (iii) The GNU memory unit is transformed, and an improved BiGRU that can adapt to the recursive network structure is constructed, which enhances the effectiveness of the model in terms of accuracy, loss rate, and iteration time.

1.1. Related Works. The amount of text data has experienced a development process from a small amount to a large amount. Correspondingly, when people recognize the commercial and social value of text data, text sentiment classification is a major task in the field of natural language processing (NLP), and its research methods are also undergoing an evolutionary process. Dictionary-based and machine learning-based methods belong to traditional text sentiment classification methods. With a small amount of data, using traditional sentiment classification methods can achieve good classification results. The method based on deep learning is an algorithm model that has been applied in the field of NLP after the deep learning model has achieved good results.

1.2. Method Based on Sentiment Dictionary. Building an emotional dictionary based on emotional knowledge and using it as a tool is the traditional method of judging the emotional polarity of subjective texts. Most of the emotional dictionaries are constructed manually, and the basic

principle is to summarize and organize widely used emotional words based on experience. When the text is input, it matches the content of the dictionary, looking for the emotional words in the text that overlap with the emotional dictionary and then judging the emotional polarity of the text.

The emotion dictionary can be traced back to 1998. Whissell asked 148 subjects to use 5 additional words to describe terms such as mathematics, physics, television, newspaper, biology, and technology. Then, it matches with sentiment words widely used in sentiment dictionary. However, no matter how the emotional dictionary is expanded and perfected, there is the boundary of a dictionary. It cannot cover all emotional expressions and the new words that appear with the development of the times cannot be included in time, which makes the accuracy of text emotional judgments low.

1.3. Method Based on Machine Learning. Learning is a kind of continuous intelligent behavior that human beings have. At present, computers have also initially possessed this ability, namely, machine learning. The core of machine learning is learning, and how to make machines learn like humans is the focus of research in the field of machine learning. The principle of sentiment analysis of text based on machine learning is that after manually extracting text features, the computer processes the text according to a specific algorithm and then outputs sentiment classification. Compared with the method that completely relies on artificially constructing the emotional dictionary, machine learning has obvious advantages. On the one hand, it can effectively relieve the burden of labor and reduce irrational judgments. On the other hand, it can build a huge database and update the lexicon in time according to the development of the times. However, traditional machine learning methods require manual screening of emotional characteristics, which is a huge workload.

1.4. Deep Learning-Based Methods. With the rise and application of deep learning, many researchers have begun to use deep learning to solve emotion classification problems. Among them, multilayer perceptron (MLP) [12], CNN [13], RNN [14], attention mechanism [15–17], and other neural network structures are widely used in text sentiment analysis tasks and can get a better semantic representation of sentences.

In reference [18], CNN was first proposed to solve the problem of NLP part of speech tagging. Yin et al. [19] proposed to apply CNN to sentiment analysis tasks and achieved good results. With the widespread application of CNN in sentiment classification, its shortcomings are becoming more and more obvious. CNN can only mine the local information of the text and lacks the capture effect of long-distance dependence. The recurrent neural network (RNN) makes up for this deficiency [19]. Compared with CNN, RNN has a memory function, can capture dynamic information in serialized data, and has achieved good results in sentiment classification tasks. Tang et al. [20] modeled the text at the text level and proposed a

hierarchical RNN model. Although RNN is suitable for context processing, when dealing with long-distance dependence problems, gradient explosions will occur. In response to this problem, Hochreiter and Schmidhuber [21] proposed an LSTM model to optimize the internal structure of the RNN. Zhu et al. [22] used LSTM to model text, divide it into word sequences, and then perform sentiment classification. Traditional LSTM can only effectively use the above information, ignoring the downward information, which affects the accuracy of sentiment classification to a certain extent. Xu et al. [23] proposed an LSTM model with a caching mechanism to capture long-term emotional information, which has been widely used in existing sentiment analysis tasks. But for the emotion of a specific target, its local features are reflected in different places in the sentence, and LSTM cannot capture the weight difference of sequence features. In order to overcome this problem, the attention mechanism widely used in the field of machine translation is introduced into sentiment analysis tasks [24].

Tian et al. [25] used two-way gated recurrent unit (GRU) combined with attention mechanism to achieve better results in short text emotions. It proves that the attention mechanism can correctly pay attention to the relevant parts of the text through weight calculation, and it also improves the interpretability of the model. Delvin et al. [26] proposed a pretraining language model BERT based on the deep two-way transformer. The internal structure of the model is completely based on self-attention. It is proved that the introduction of attention mechanism can more easily capture the long-distance interdependent features in sentences. Attention can also directly connect any two words in a sentence, and the distance between long-distance dependent features is greatly shortened, which is conducive to effective use of these features. In addition, the attention mechanism also directly helps increase the parallelism of computation. Wang et al. [27] combined LSTM and attention mechanism to propose an ATAE aspect sentiment analysis model. The model integrates the information encoded in the input and adds an attention mechanism after the hidden state of LSTM output. Cheng et al. [28] proposed the HEAT model, which further uses the hierarchical attention mechanism to capture aspect information to complete the sentiment analysis of specific aspects of the sentence, thereby improving the accuracy of aspect-level sentiment analysis. Xue and Li [29] proposed a gated convolutional neural network GCAE, which adds aspect information in the convolution process to classify emotions in different aspects. Jiang et al. [30] proposed a fine-grained LSTM-CNN attention classification model, but LCA differs from this model in the following points. Firstly, the two-way LSTM was selected to more fully mine the context information. Secondly, the CNN layer in LCA performs an attention operation before pooling, which can effectively retain the local information lost by the pooling operation. Finally, LCA can be used not only for aspect sentiment analysis but also for general text classification tasks. Therefore, the ability of attention to fuse the remote dependencies of LSTM and the local features of CNN can be better used to effectively improve the accuracy of emotion classification.

2. Proposed Improved AT-BiGRU Model

The improved AT-BiGRU overall model takes into account the spelling errors of the network evaluation text. And through the combination of the improved BiGRU model and the attention mechanism, the model training time is shortened. It solves the problems of complex calculation and high space cost of most text sentiment analysis models. The model is mainly divided into three parts: text vectorization input layer, hidden layer, and output layer. Among them, the hidden layer consists of four layers: BiGRU layer, attention layer, dropout layer, and dense layer.

The word vector obtained by text preprocessing passes through the input layer and enters the BiGRU layer of the neural network to extract features. The word vector obtained by text preprocessing is extracted by input layer and neural network BiGRU layer, and then the key information of the word vector is highlighted by attention mechanism, through dropout layer to prevent over fitting, then through full connection layer, and finally into softmax layer for text emotion classification.

Compared with BiLSTM, AT-BiGRU model is faster in training, easier to capture the relationship between text contexts, and requires fewer training parameters; the attention mechanism can assign different weight information to different word vectors, highlighting the importance of words; therefore, the combination of the two can not only increase the training speed but also capture the key emotional information of the text, improve the accuracy of text classification, and more easily obtain the essential characteristics of the text.

The improved AT-BiGRU model is shown in Figure 1.

2.1. Text Preprocessing. The text form of online reviews is relatively free and has extremely unstructured characteristics. It is not possible to directly use the computer to classify the emotions of the web comment text. At this time, it is necessary to transform the text information into corresponding real number vectors for analysis and processing. This paper uses GloVe to realize the mapping of discrete text to real number space. This method is based on the statistical information of global vocabulary co-occurrence to learn word vectors, thus combining the statistical information with the local context window method.

In order to save more co-occurrence information between text vocabulary, the GloVe model constructs an approximate matrix of vocabulary co-occurrence matrix. The calculation formula is shown in formulas (1)–(3).

$$X_i = \sum_{k=1}^V X_{i,k}, \quad (1)$$

$$P_{ik} = \frac{X_{i,k}}{X_i}, \quad (2)$$

$$R_{i,j,k} = \frac{P_{i,k}}{P_{j,k}}, \quad (3)$$

where X_i represents the sum of words appearing in a row of matrix word i ; V represents the total number of words in the

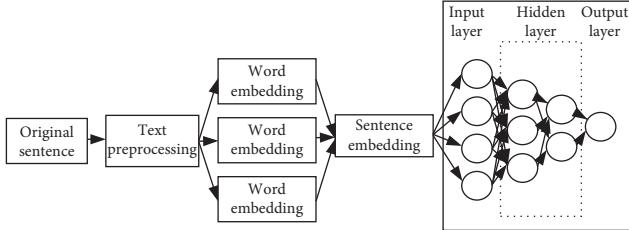


FIGURE 1: Improved AT-BiGRU model diagram.

dictionary; $X_{i,j}$ represents the number of times the word j and the word i appear together in the fixed window in the training corpus; P_{ik} represents the probability that the word k appears in the fixed window in the word i ; and $R_{i,j,k}$ represents the relationship between the three words i , j , and k . If the value of $R_{i,j,k}$ is very large, it means that the words i and k are related, but the words J and K are not; if the value of $R_{i,j,k}$ is small, it means that the words j and k are related, but the words T and K are not. If the value of $R_{i,j,k}$ approaches 1, it means that the words j and k are related, the words i and k are related or the words j and k are not related, and the words i and k are not related. Compared with the original probability P_{ik} , $R_{i,j,k}$ can better distinguish the relationship between words.

The GloVe model constructs a function $F(w_i, w_j, w'_k)$ to maximize the ratio close to P_{ik}/P_{jk} as the convergence target of the model, so that the word vector contains the information contained in the co-occurrence matrix, where $w, w' \in R^d$ are the corresponding word vectors. In the face of noise data and other words that cause an unbalanced co-occurrence relationship between words and some unreasonable co-occurrence relationships, words will be given very small weights, which is not conducive to model learning parameters. Therefore, a weight equation $f(X_{i,j})$ is introduced when constructing the loss function, and the constructed loss function is shown in the following equation:

$$J = \sum_{i,j=1}^V f(X_{i,j}) \left((w_i^T w_j' + b_i + b_j') - \log(X_{ij}) \right)^2, \quad (4)$$

where $X_{i,j}$ represents the number of times the words w_i and w_j appear together in the window; w_i^T represents the transposition of the word vector in the context of the word i when the word w_i is used as the context; w_j' is the word vector of j when w_j is the center word of context; b_i and b_j' denote bias; V is the total number of words in the dictionary.

3. Algorithm Design

3.1. Improved BiGRU. In order to adapt the GRU to the recursive network structure, the ordinary GNU memory unit needs to be modified so that it can accept the input of two child nodes. In the following, this improved dual-input GRU is referred to as BiGRU for short. Assuming there is a BiGRU node unit j , the output of node j is h_j , and the outputs of the left and right child nodes of node j are h_j^L and h_j^R , respectively. The calculation method of h_j is shown in the following formula:

$$h_j = z_j (\omega^L h_j^L + \omega^R h_j^R) + (1 - z_j) h_j, \quad (5)$$

where ω^L, ω^R are the weights corresponding to the left and right child nodes of the GRU unit, and $\omega^L + \omega^R = 1$.

The update function of BiGRU node j is z_j . The main function of the update function is to control whether the BiGRU is updated, which is similar to the function of a control gate, so it can also be called an update gate. The update gate will determine how to update the cell state according to the input vector x_j and the child node output h_j^L, h_j^R . The specific calculation method of the update gate z_j of the BiGRU is shown in the following formula:

$$z_j = \sigma(W_z x_j + U_z (\omega^L h_j^L + \omega^R h_j^R)). \quad (6)$$

The calculation method of the candidate output of BiGRU node j as \tilde{h}_j is shown in formula (7), where f represents the sigmoid function, \odot represents the dot multiplication operation, W and U are parameter matrices, and $x_j \in R^d$ is the d -dimensional vector representation of the word.

$$\tilde{h}_j = f(W_h x_j + U_h (r_j \odot (\omega^L h_j^L + \omega^R h_j^R))). \quad (7)$$

The reset gate r_j of the BiGRU mainly controls whether to reset the memory unit. When the value of the reset function is close to 0, the GRU can effectively ignore the historical information, which can effectively prevent long-term dependence. The specific calculation method of the reset function is shown in the following formula:

$$r_j = \sigma(W_r x_j + U_r (\omega^L h_j^L + \omega^R h_j^R)). \quad (8)$$

The function σ is a nonlinear function, and the tanh function is usually used. For the BiGRU, $h_j, h_j^L, r_j, z_j \in R^d$, and d are the dimensions of the input word vector, and emotion information prediction can be achieved through softmax.

3.2. Attention Mechanism. The essential idea of the attention mechanism is shown in Figure 2: source can be assumed to be composed of a series of $\langle \text{Key}, \text{Value} \rangle$ data pairs. Key-value queries have three basic elements: Query, Key, and Value. The calculation process of the attention value can be summarized as follows. Firstly, obtain the weight coefficient of each Key's corresponding Value by calculating the correlation between each Query and each Key, and then perform a weighted summation of the weight and the corresponding key value. Therefore, the essential idea of the attention mechanism can be described as a mapping from a query to a series of key-value pairs, which is expressed as follows:

$$\text{Attention}(\text{Query}, \text{Source}) = \sum_{i=1}^{L_x} \text{Similarity}(\text{Query}, \text{Key}_i) * \text{Value}_i, \quad (9)$$

where $L_x = \|\text{Source}\|$ represents the length of the data source.

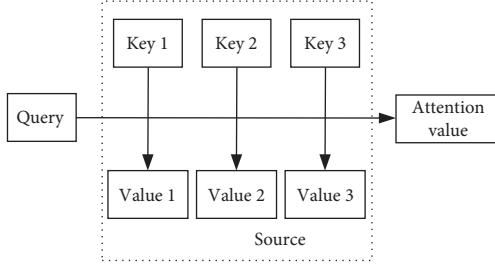


FIGURE 2: The essence of attention mechanism.

The specific calculation process of the attention mechanism can be abstracted into the three stages as shown in Figure 3.

Among them, K (Key) represents keywords, Q (Query) represents query, F represents function, V (Value) represents weight value, Sim represents similarity, a represents weight coefficient, and A (Attention Value) represents attention value.

In the first stage, the weight coefficient of each Key corresponding to Value is obtained by calculating the correlation between each Query and each Key.

In the second stage, a similar softmax function is introduced to normalize the weights, which can highlight the weights of important elements. a_i is the weight coefficient corresponding to Value, and the specific calculation is shown in the following formula:

$$a_i = \text{Soft max}(\text{Sim}_i) = \frac{e^{\text{Sim}_i}}{\sum_{j=1}^{L_x} e^{\text{Sim}_j}}. \quad (10)$$

In the third stage, the weight and the corresponding key value are weighted and summed to get the final attention value.

3.3. Model Structure. The specific structure of the improved AT-BiGRU model is shown in Figure 4.

3.3.1. Input Layer. The datasets of this article are SemEval-2014 Task 4 and SemEval-2017 Task 4. The input layer is mainly to preprocess the comment data. Before the formal text processing operation, considering the possible spelling errors in the comments, import the textblob package to correct the possible spelling errors. m words compose the text a of l sentences, which is $a = \{s_1, s_2, \dots, s_l\}$, and the j th sentence in the sample is denoted as $s_j = \{w_{j1}, w_{j2}, \dots, w_{jm}\}$. Perform text vector operations to make $w \in w^a$. The specific steps of text vectorization are as follows:

- (1) Read data and perform data cleaning.
- (2) Aiming at the phenomenon that the length of the word vector is different, the data are vectorized into the form of a specified length of 400 (if the sentence length is less than the specified value, special symbols are automatically filled in the back by default; if the sentence length is greater than the specified value,

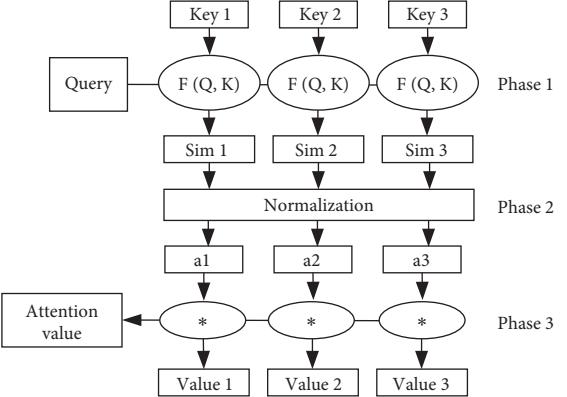


FIGURE 3: The three-stage calculation process of the attention mechanism.

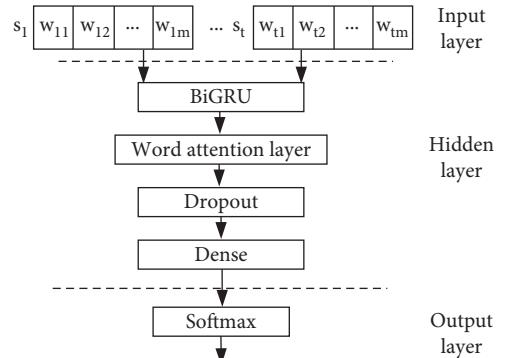


FIGURE 4: Improved AT-BiGRU model diagram.

the first 400 words will be retained by default, and the extra part will be truncated).

- (3) Initialize the data randomly and divide the training set and the test set according to 8:2.
- (4) After vectorizing the data, each comment becomes an index vector of uniform length, and each index corresponds to a word vector.

After the above four steps, the input data become the word matrix formed by the index corresponding word vector, that is, the uniform length of the word vector after processing is set to 400. Using the form of 100-dimensional vector of glove.6B.100d, word vectors that cannot be found in glove.6B.100d are initialized randomly. Let c_{ji} be the i -th word vector of the j -th sentence; a piece of comment data with a length of 400 is represented as

$$c_{j1:j1000} = c_{j1} \oplus c_{j2} \oplus \dots \oplus c_{j1000}. \quad (11)$$

Among them, \oplus represents the connection operator between the word vector and the word vector, and $c_{j1:j1000}$ represents the word vector matrix of the j th sentence. According to the index, each word in each comment corresponds to the word vector in glove.6B.100d to generate a word vector matrix.

3.3.2. Hidden Layer. The calculation of the hidden layer is mainly divided into two steps:

Step 1. Calculate the word vector output by the BiGRU layer. The text word vector is the input vector of the BiGRU layer, and the purpose of the BiGRU layer is mainly to extract the deep features of the text from the input text vector. The word vector of the t -th word of the j -th sentence input at time i is c_{ijt} . After feature extraction from the BiGRU layer, the relationship between contexts can be learned more fully and semantic coding can be performed. Specific calculation formula:

$$h_{ijt} = \text{BiGRU}(c_{ijt}). \quad (12)$$

Step 2. Calculate the probability weight that each word vector should be assigned. In order to highlight the importance of different words to the sentiment classification of the entire text, the attention layer is introduced. The input of the attention layer is the output vector h_{ijt} of the previous layer that has been activated by the BiGRU neural network layer. The weight coefficients are specifically calculated by the following formulas:

$$\begin{aligned} u_{ijt} &= \tanh(w_w h_{ijt} + b_w), \\ \alpha_{ijt} &= \frac{\exp(u_{ijt}^T u_w)}{\sum_{nt} \exp(u_{ijt}^T u_w)}, \\ s_{ijt} &= \sum_{i=1}^n \alpha_{ijt} h_{ijt}. \end{aligned} \quad (13)$$

Among them, h_{ijt} is the output vector of the previous BiGRU neural network layer, w_w is the weight coefficient, b_w is the bias coefficient, and u_w is the randomly initialized attention matrix. The attention mechanism matrix is the cumulative sum of the product of the different probability weights assigned by the attention mechanism and the state of each hidden layer and is obtained by using the softmax function for normalization.

3.3.3. Dropout Layer. In order to avoid the occurrence of overfitting, a dropout layer is added between the attention layer and the fully connected layer. In the neural network, some nodes are randomly ignored, and nodes are randomly selected each time, which can effectively prevent the learned model from performing well on the training data and poor performance on the test data. The following describes the main workflow of the dropout layer and how it works in a specific neural network.

(1) Workflow

(1) Forward propagation: input the output result of the attention layer to the dropout layer. The dropout layer randomly ignores the preset nodes in the internal hidden layer (the ignored nodes are backed up and saved), and then the remaining nodes are propagated forward to obtain the predicted label.

(2) Backpropagation: compare the original label with the value obtained from the predicted label and adjust the parameters through backpropagation.

(3) Update the parameters: the adjusted value is only updated on the nodes that are not ignored, and the other ignored nodes are not updated. Repeat the corresponding operation on the training dataset in the next iteration process until the number of iterations is reached and the model is trained.

(2) *How the Dropout Layer Works* The calculation method of the $(l+1)$ layer of ordinary neurons:

$$\begin{aligned} z_i^{l+1} &= w_i^{(l+1)} y^l + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}). \end{aligned} \quad (14)$$

After joining the dropout layer, a probability distribution is required for the y value through Bernoulli(p). The probability distribution is $(0, 1)$, and the calculation formula is as follows:

$$\begin{aligned} r_j^{(l)} &\sim \text{Bernoulli}(p), \\ \tilde{y}^{(l)} &= r^{(l)} * y^{(l)}, \\ z_i^{(l+1)} &= w_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}). \end{aligned} \quad (15)$$

Among them, $(l+1)$ represents the neuron of the $(l+1)$ th layer, $r_j^{(l)} \sim \text{Bernoulli}(p)$ represents the probability distribution of the j th neuron of the l th layer subject to the Bernoulli function, Bernoulli(p) satisfies $(0, 1)$, $w_i^{(l+1)}$ and $b_i^{(l+1)}$, respectively, represent the weight matrix and the corresponding displacement matrix of the $(l+1)$ -th layer, $f(z_i^{(l+1)})$ represents the computer result z of the i -th hidden layer of the $(l+1)$ -th neural network, and the nonlinear function is represented by the f function.

3.3.4. Output Layer. The input of the output layer is the output of the dense layer. The softmax function is used to calculate the input of the output layer to classify the text. The specific formula is as follows:

$$y_j = \text{soft max}(w_1 s_{ijt} + b_1), \quad (16)$$

where w_1 represents the weight coefficient matrix to be trained from the attention mechanism layer to the output layer, b_1 represents the bias to be trained, and y_j is the output prediction label.

3.4. Model Training. Take SemEval-2014 Task 4 and SemEval-2017 Task 4 datasets, preset parameters, and number of iterations N as input. The text vectorized input layer processes the dataset in the form of word vectors and classifies the dataset with the AT-BiGRU model.

Let the word vector corresponding to a word in the text be x_t in the form of the word vector corresponding to the dataset. Processing of each comment in the dataset:

For hop = 1 to h,

$$\begin{aligned}
\overrightarrow{h}_t &= \text{GRU}\left(x_t, \overrightarrow{h}_{t-1}\right), \\
\overleftarrow{h}_t &= \text{GRU}\left(x_t, \overleftarrow{h}_{t-1}\right), \\
h_t &= w_t \overrightarrow{h}_t + v_t \overleftarrow{h}_t + b_t, \\
u_{ijt} &= \tanh(w_w h_{ijt} + b_w),
\end{aligned} \tag{17}$$

$$\alpha_{ijt} = \frac{\exp(u_{ijt}^T u_w)}{\sum_t \exp(u_{ijt}^T u_w)},$$

$$s_{ijt} = \sum_{i=1}^n \alpha_{ijt} h_{ijt}.$$

End for.

$y_j = \text{soft max}(w_1 s_{ijt} + b_1)$ calculated by softmax is compared with the original label, and the objective function is

$$\text{loss} = - \sum_j \hat{y}^{(j)} \log y^{(j)}. \tag{18}$$

Through the above training steps, using formula (18), feature extraction is performed on the words from 1 to h , and the corresponding weights are assigned to the cumulative sum. The dense layer further extracts features and finally performs classification in the softmax output layer. Then, the results of the multiplication of each comment tag value and $\log y^{(j)}$ are accumulated. The sum of the accumulated values is negative, and the opposite is taken to minimize the loss and reduce the calculation error. Adam is used as the training device to make the model training and convergence faster. In the process of backward error propagation along time, the weights and offsets are adjusted and updated according to the errors until the iterations are reached or a fixed precision is reached.

4. Experimental Results and Analysis

4.1. Experimental Environment and Model Parameters. The proposed model is experimentally demonstrated on the SemEval-2014 Task 4 and SemEval-2017 Task 4 datasets. The experimental environment is shown in Table 1.

The parameter settings of the improved AT-BiGRU model are shown in Table 2.

4.2. Evaluation Index. When evaluating the performance indicators of word vectors, the confusion matrix method is used. TP means that the positive class is predicted as a positive class number, TN predicts a negative class as a negative class number, FP predicts a negative class as a positive class number, and FN means that a positive class is predicted as a negative class number. It can be represented by the confusion matrix in Table 3.

TABLE 1: Experimental environment.

| Experimental environment | Configuration |
|---------------------------|--------------------------------------|
| Operating system | Windows 10 (64 bit) |
| CPU | Intel (R) Core (TM) i7-4790, 3.6 GHz |
| RAM | 8G |
| Hard disk | 1T |
| Programming language | Python 3.5 |
| Deep learning framework | Keras 2.0 |
| Word vector training tool | Word2vec |

TABLE 2: Model parameters.

| Parameters | Value |
|--|---------------------|
| Embedding number of hidden layer nodes | 64 |
| BIGRU number of hidden layer nodes | 64 |
| Dense_1 | 100 |
| Dense_2 | 1 |
| Loss function | binary_crossentropy |
| Batch_size | 50 |
| Word vector dimension | 100 |

TABLE 3: Representation of confusion matrix method.

| | Positive | Negative |
|-------|----------|----------|
| True | TP | TN |
| False | FP | FN |

When dealing with sentiment analysis tasks, there are usually four evaluation indicators: precision, accuracy, recall, and $F1$ value.

Precision can be defined as

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{19}$$

The accuracy rate is expressed as the ratio of the number of correctly classified samples on the test dataset to the total number of samples. The formula is expressed as

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{20}$$

The recall rate can be expressed as

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{21}$$

$F1$ value is the harmonic mean value of precision rate and recall rate, expressed as

$$F1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{22}$$

It can be seen from equation (22) that the value of $F1$ will increase with the increase in accuracy and accuracy. Generally speaking, the accuracy rate is for the prediction result, which means the correct number of samples whose

prediction is positive. The recall rate is for the training set, which represents the number of positive examples predicted to be correct in the sample, including the positive class prediction in the sample as the positive class (TP) and the positive class prediction in the sample as the negative class (FN).

4.3. Experimental Datasets. The experiment uses the SemEval-2014 Task 4 and SemEval-2017 Task 4 datasets. For the single-topic text sentiment classification task, the Twitter text sentiment classification task of SemEval-2017 Task 4 is adopted. The sentiment polarity classification is mainly carried out on 12284 Twitter text data, which are divided into three categories: negative, neutral, and positive. For example, the emotional polarity expressed in the text “I really want to try the Mannequin Challenge” is positive. The experimental data required for the single-topic sentiment classification task are shown in Table 4.

For emotional tasks with different topics in the text, the restaurant dataset in SemEval-2014 Task 4 is used, which contains customer review text. There are three kinds of emotions: positive, neutral, and negative, and five themes {food, price, service, ambience, anecdotes/misellaneous}. SemEval-2017 Task 4 contains Twitter comment data. The topic of the comment is extracted from the text. The emotional polarity of the topic includes positive and negative. Table 5 shows the statistics of the data required for experiments on sentiment classification tasks of different topics.

4.4. Activation Function Comparison Experiment. In the neural network, in order to avoid pure linear combination, the activation function introduces a nonlinear factor to the neuron to improve the expressive ability of the model. The activation functions commonly used in traditional neural networks include sigmoid, tanh, ReLU, etc., and the function images are shown in Figure 5.

The above three activation functions are selected for experiments, and the experimental results are shown in Table 6.

It can be seen from Table 6 that the AT-BiGRU model using the sigmoid activation function has the best accuracy and loss rate in sentiment classification tasks. Compared with the model using tanh activation function and ReLU activation function, the accuracy rate is increased by 22.65% and 11.41%, and the loss rate is decreased by 0.108 and 0.189, respectively.

4.5. Dropout Selection Experiment. The results of the dropout selection experiment are shown in Table 7.

It can be seen from Table 7 that when dropout is set to 0.2, the model has the highest accuracy and the shortest time consumption. When the value of dropout is 0.1, the model achieves the smallest loss rate. When the value of dropout is 0.2, the difference between the loss rate and the lowest value is only 0.001. Therefore, when the value of dropout is 0.2, the overall performance of the model is optimal.

TABLE 4: Experimental data of single-subject sentiment classification task.

| | Positive | Neutral | Negative |
|-------|----------|---------|----------|
| Train | 1900 | 4750 | 3178 |
| Test | 475 | 1187 | 794 |

TABLE 5: Experimental data statistics of sentiment classification tasks for different topics.

| Dataset | | Positive | Neutral | Negative |
|------------|-------|----------|---------|----------|
| Restaurant | Train | 2179 | 500 | 839 |
| | Test | 657 | 94 | 222 |
| Twitter | Train | 14897 | — | 3997 |
| | Test | 2463 | — | 3722 |

4.6. Iterative Experiment. The proposed method and the methods in [20, 22, 26] are trained on the training set, and comparative experiments are performed on the test set. The relationship between the accuracy on the test set and the number of iterations is shown in Figure 6.

It can be seen from Figure 6 that the overall accuracy of each model is continuously improved from bottom to top, and the accuracy of the proposed method is always higher than the other three comparison methods. Compared with the maximum pooling, the attention layer highlights the important information more quickly, the deep-level features of the extracted text, and the features that quickly converge and quickly improve the accuracy rate. The accuracy rate in the initial training is higher than that of the other three, and the training effect is better. At the same time, the overall accuracy of the proposed method changes steadily, but the accuracy may be lower in some iterations. But the accuracy rate is always higher than that of the other three comparison methods. On the whole, the accuracy curves of the proposed method and methods in [20, 22] are relatively close, while the accuracy curve of reference [26] has relatively large fluctuations. It can be seen that the proposed method performs better and is more stable in extracting deep features of text. In terms of the number of iterations, it is not that the more the iterations, the higher the accuracy. Each method has its optimal number of iterations to achieve the highest accuracy. For example, the proposed method has the highest accuracy rate in the fourth iteration, while reference [22] has the highest accuracy rate in the sixth iteration. Based on the above analysis, the proposed method can effectively improve the accuracy of training data with the least number of iterations.

In addition, the change trend curve of the time required for four different methods to complete an iteration under the same experimental conditions is shown in Figure 7.

It can be seen from Figure 7 that the iteration time of each method generally does not fluctuate much, and the overall time tends to be stable. Generally, after the minimum iteration time has passed, when training again, the training time will no longer fluctuate greatly. The method in [20]

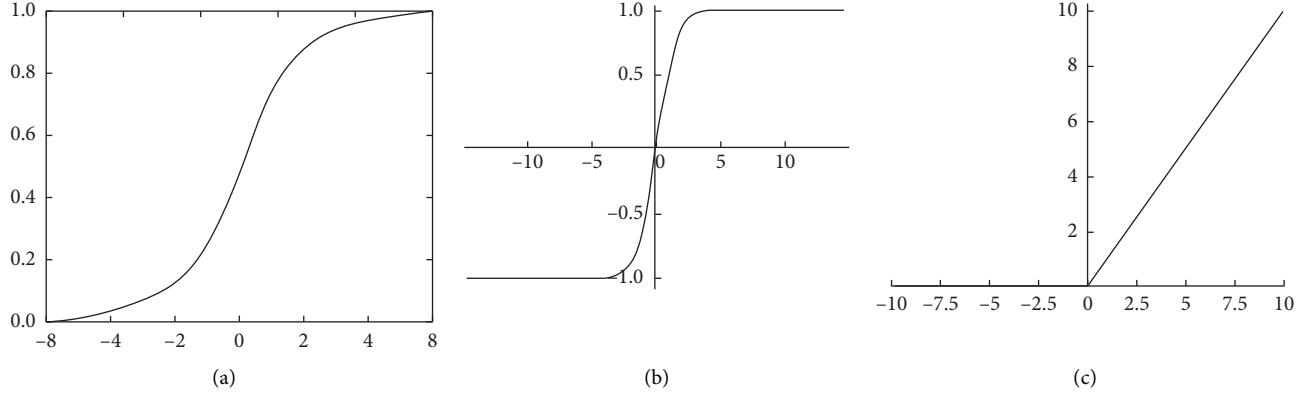


FIGURE 5: The activation function. (a) Sigmoid activation function. (b) Tanh activation function. (c) ReLU activation function.

TABLE 6: Comparison of experimental results of different activation functions.

| Activation function | Sigmoid | Tanh | ReLU |
|---------------------|---------|-------|-------|
| Accuracy (%) | 89.23 | 72.75 | 80.09 |
| Loss rate | 0.283 | 0.391 | 0.472 |

TABLE 7: Dropout selection experiment results.

| Dropout | Accuracy (%) | Loss rate | Time (s) |
|---------|--------------|-----------|----------|
| 0.1 | 89.08 | 0.275 | 1138 |
| 0.2 | 89.21 | 0.276 | 1129 |
| 0.3 | 89.16 | 0.278 | 1224 |
| 0.4 | 89.03 | 0.279 | 1265 |
| 0.5 | 88.97 | 0.281 | 1241 |

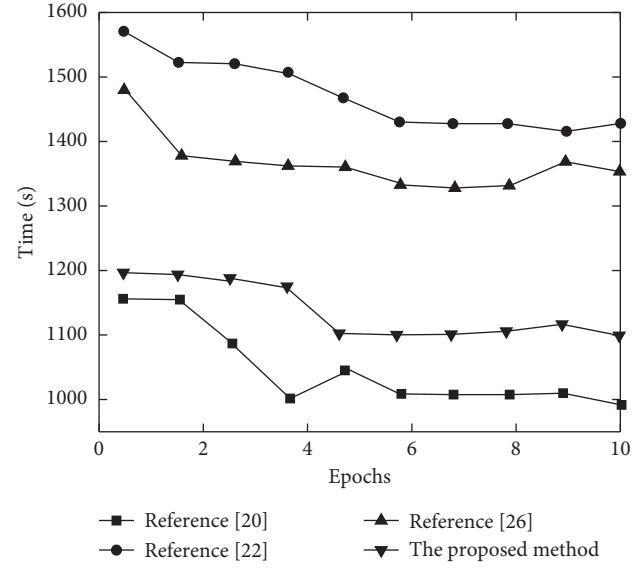


FIGURE 7: The iteration time of four methods.

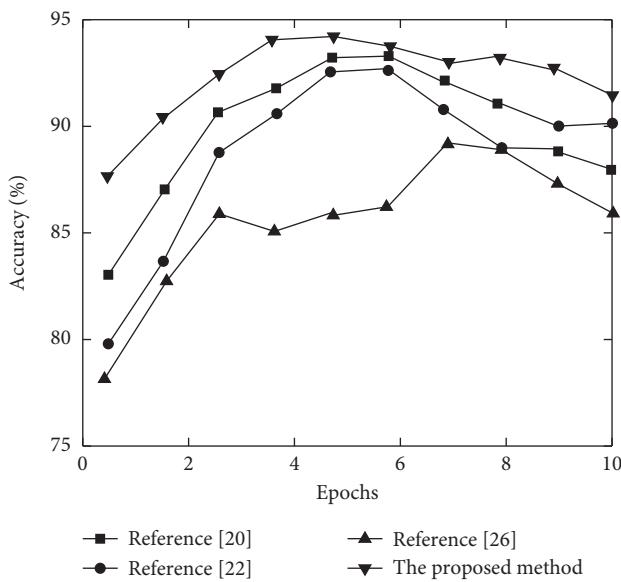


FIGURE 6: Changes in accuracy of different methods.

takes the shortest time to complete one iteration training, which is inseparable from the convergence speed of the training iteration of the maximum pooling layer of the

convolution in the RNN model. The iteration time of the proposed method is higher than that of reference [20] but lower than the LSTM model of reference [22] and the BERT model of reference [26] because the improved AR-BiGRU model in the proposed method has the characteristics of faster calculation and fewer parameters than the LSTM model. Reference [22] takes the longest time because the LSTM neural network is relatively complicated to calculate, increasing the calculation time; while highlighting the key information, it also increases the weighted calculation time.

4.7. Model Loss Rate Comparison. Similarly, the loss rate changes of the proposed method and the methods of 20, 22, 26 in 10 iterations are shown in Figure 8.

It can be seen from Figure 8 that the loss rate of the four methods on the training set shows an overall downward trend as the number of iterations increases and eventually stabilizes. But on the test set, only the overall trend of the loss rate of the proposed method decreases as the number of iterations increases. The loss rate of the proposed method is

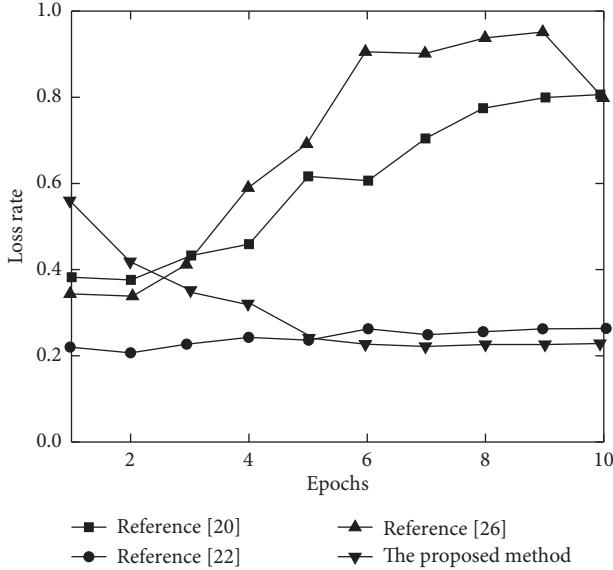


FIGURE 8: Model loss rate change graph.

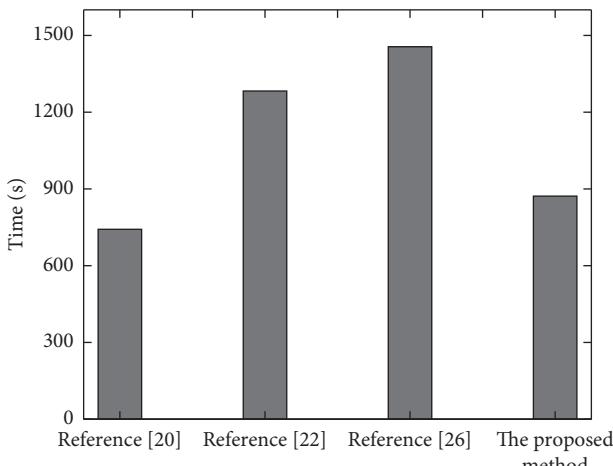


FIGURE 9: The comparison of time performance results.

the lowest. Because it adopts the BiGRU model combined with the attention mechanism, it can solve the problem of long range and global dependence of text in sentiment classification tasks, so the model has better generalization ability. Reference [26] adopts the BERT model based on self-attention mechanism, which makes it easier to capture long-distance interdependent features in sentences. However, due to the greater dependence on sentence length, the loss rate is higher. The RNN model is used in reference [20], and the loss rate is low, about 0.24, which is about 0.02 more than the proposed method.

4.8. Time Performance Comparison. In addition, the time consumption of different methods in the network text classification task is shown in Figure 9.

It can be seen from Figure 9 that the proposed method takes a relatively short time in completing the network text classification, which is about 980 s. Reference [20] uses a

single RNN with simple structure and easy training, so it takes the shortest time, about 750 s. However, the classification accuracy of this method is not high, so the overall performance is poor. The LSTM model is used in reference [22], and the BERT model is used in reference [26]. Both models are more complicated because the time to complete the text classification is longer, both exceeding 1200 s. The proposed method adopts an improved AT-BiGRU model, in which the attention mechanism can be well paralleled, which solves the problem that the cyclic neural network cannot be parallelized. Therefore, the model training efficiency is improved to a certain extent.

5. Conclusion

The mobile client provides convenience for users to express their opinions. Now most online platforms provide browsing and comment-related functions. Therefore, user comment data are tens of thousands every day and show an exponential growth model. Analysis of these data can generate huge commercial and social value.

Aiming at the problems of current mainstream sentiment analysis models, a new neural network model based on improved AT-BiGRU is proposed. Before the formal text preprocessing, the textblob package is imported to correct some possible spelling errors. We use pad_sequences technology to fill the word vector with uniform length and the embedding layer to unify the word vector into a fixed embedded layer word matrix form. The BiGRU neural network is used to fully extract the text context information, and the attention model is used to highlight the key information of the text. The proposed model is experimentally demonstrated on the SemEval-2014 Task 4 and SemEval-2017 Task 4 datasets. The experimental results show that the proposed model effectively avoids the bias of text sentiment analysis caused by spelling errors. The effectiveness of the improved AT-BiGRU model in terms of accuracy, loss rate, and iteration time is verified. In the next study, we will consider incorporating topic word information into word vectors, so as to better perform text sentiment analysis tasks for multiple topics in the text.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the 2021 Key Development Plan Project of the Science and Technology Department of Henan Province (no. 212102210400).

References

- [1] L. Wang, J. Niu, and S. Yu, "SentiDiff: combining textual information and sentiment diffusion patterns for twitter sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 10, pp. 2026–2039, 2020.
- [2] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813–830, 2016.
- [3] D. Deng, L. Jing, J. Yu, and S. Sun, "Sparse self-attention LSTM for sentiment lexicon construction," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1777–1790, 2019.
- [4] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings using intensity scores for sentiment analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 3, pp. 671–681, 2018.
- [5] B. Zhang, D. Xu, H. Zhang, and M. Li, "STCS lexicon: spectral-clustering-based topic-specific Chinese sentiment lexicon construction for social networks," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1180–1189, 2019.
- [6] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.
- [7] S. Ding, J. Wu, and H. Li, "Chinese micro-blogging opinion recognition based on SVM model," *Journal of the China Society for Scientific and Technical Information*, vol. 35, no. 12, pp. 1235–1243, 2016.
- [8] M. U. Salur and I. Aydin, "A novel hybrid deep learning model for sentiment classification," *IEEE Access*, vol. 8, pp. 58080–58093, 2020.
- [9] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, ACL, Doha, Qatar, October 2014.
- [10] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 655–665, ACL, Maryland, MA, USA, June 2014.
- [11] X. Wang, Y. Liu, C. Sun et al., "Predicting polarities of tweets by composing word embeddings with long short-term memory," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 1343–1353, Beijing, China, July 2015.
- [12] A. S. Góes and R. Oliveira, "A process for human resource performance evaluation using computational intelligence: an approach using a combination of rule-based classifiers and supervised learning algorithms," *IEEE Access*, vol. 8, pp. 39403–39419, 2020.
- [13] M. Ling, Q. Chen, Q. Sun, and Y. Jia, "Hybrid neural network for sina weibo sentiment analysis," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 983–990, 2020.
- [14] R. L. Rosa, G. M. Schwartz, W. V. Ruggiero, and D. Z. Rodriguez, "A knowledge-based recommendation system that includes sentiment analysis and deep learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2124–2135, 2019.
- [15] G. Liu, X. Huang, X. Liu et al., "A novel aspect-based sentiment analysis network model based on multilingual hierarchy in online social network," *The Computer Journal*, vol. 63, no. 1, pp. 410–424, 2020.
- [16] Y. Cheng, L. Yao, G. Xiang, G. Zhang, T. Tang, and L. Zhong, "Text sentiment orientation analysis based on multi-channel CNN and bidirectional GRU with attention mechanism," *IEEE Access*, vol. 8, pp. 134964–134975, 2020.
- [17] C. Chen, Z. Wang, and W. Li, "Tracking dynamics of opinion behaviors with a content-based sequential opinion influence model," *IEEE Transactions on Affective Computing*, vol. 11, no. 4, pp. 627–639, 2020.
- [18] R. Collobert, J. Weston, L. Bottou et al., "Natural language processing (almost) from scratch[J]," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.
- [19] W. Yin, K. Kann, M. Yu et al., "Comparative study of CNN and RNN for natural language processing," 2017, <https://arxiv.org/abs/1702.01923>.
- [20] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1422–1432, Lisbon, Portugal, September 2015.
- [21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [22] X. Zhu, P. Sobhani, and H. Guo, "Long short-term memory over recursive structures," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, pp. 1604–1612, Lille, France, July 2015.
- [23] J. Xu, D. Chen, X. Qiu et al., "Cached long short-term memory neural networks for document-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1660–1669, Austin, TX, USA, November 2016.
- [24] N. Jiang, F. Tian, J. Li, X. Yuan, and J. Zheng, "MAN: mutual attention neural networks model for aspect-level sentiment classification in SIoT," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2901–2913, 2020.
- [25] Z. Tian, W. Rong, L. Shi, J. Liu, and Z. Xiong, "Attention aware bidirectional gated recurrent unit based framework for sentiment analysis," *Knowledge Science, Engineering and Management*, Springer, in *Proceedings of the 2018 Conference of the International Conference on Knowledge Science, Engineering and Management*, pp. 67–78, August 2018.
- [26] J. Delvin, M. Chang, K. Lee et al., "BERT: PreTraining of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, Minneapolis, MN, USA, June 2019.
- [27] Y. Wang, M. Huang, L. Zhao et al., "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 606–615, Austin, TX, USA, November 2016.
- [28] J. Cheng, S. Zhao, J. Zhang et al., "Aspect-level sentiment classification with HEAT (Hierarchical Attention) network," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 97–106, Singapore, November 2017.
- [29] W. Xue and T. Li, "Aspect based sentiment analysis with gated convolutional networks," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 2514–2523, Melbourne, Australia, July 2018.
- [30] M. Jiang, W. Zhang, M. Zhang et al., "An LSTM-CNN attention approach for aspect-level sentiment classification," *Journal of Computational Methods in Sciences and Engineering*, vol. 5, no. 19, pp. 859–868, 2019.