electronics

MDPI

Article

Chinese Multicategory Sentiment of E-Commerce Analysis Based on Deep Learning

Hongchan Li, Jianwen Wang, Yantong Lu, Haodong Zhu * and Jiming Ma

School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China; 332107040614@email.zzuli.edu.cn (J.W.)
* Correspondence: 2011016@zzuli.edu.cn

Abstract: With the continuous rise of information technology and social networks, and the explosive growth of network text information, text sentiment analysis technology now plays a vital role in public opinion monitoring and product development analysis on networks. Text data are high-dimensional and complex, and traditional binary classification can only classify sentiment from positive or negative aspects. This does not fully cover the various emotions of users, and, therefore, natural language semantic sentiment analysis has limitations. To solve this deficiency, we propose a new model for analyzing text sentiment that combines deep learning and the bidirectional encoder representation from transformers (BERT) model. We first use an advanced BERT language model to convert the input text into dynamic word vectors; then, we adopt a convolutional neural network (CNN) to obtain the relatively significant partial emotional characteristics of the text. After extraction, we use the bidirectional recurrent neural network (BiGRU) to bidirectionally capture the contextual feature message of the text. Finally, with the MultiHeadAttention mechanism we obtain correlations among the data in different information spaces from different subspaces so that the key information related to emotion in the text can be selectively extracted. The final emotional feature representation obtained is classified using Softmax. Compared with other similar existing methods, our model in this research paper showed a good effect in comparative experiments on an e-commerce text dataset, and the accuracy and F1-score of the classification were significantly improved.

Keywords: BERT; deep learning; dynamic word vector; BiGRU; sentiment analysis; MultiHeadAttention mechanism

Citation: Li, H.; Wang, J.; Lu, Y.; Zhu, H.; Ma, J. Chinese Multicategory Sentiment of E-Commerce Analysis Based on Deep Learning. *Electronics* **2023**, *12*, 4259. https://doi.org/10.3390/electronics12204259

Academic Editor: Byung-Gyu Kim

Received: 4 September 2023
Revised: 24 September 2023
Accepted: 7 October 2023
Published: 15 October 2023

1. Introduction

As the Internet continues to evolve, netizens can express their emotions and attitudes towards various things on a variety of public platforms, and as a large amount of information exists on the Internet, research on sentiment analysis using networks' texts has attracted many people's attention. Sentiment analysis on social platforms can actively monitor the development of trends in public opinion on the Internet, which is helpful for government and social functional departments to grasp such trends in public opinion. The sentiment analysis of e-commerce platforms [1] is helpful for enterprises to understand product conditions and consumer needs in a timely manner and is conducive to enterprise development and economic recovery; this will have profound significance for enterprises and the national economy.

Currently, sentiment analysis is a research hotspot [2] that involves obtaining people's attitudes from natural language fields, such as text and images. Text is the most concise and direct way to convey emotions, and we express emotions and thoughts mainly through words and language; therefore, the most direct and meaningful way to understand netizens' attitudes towards things is from the emotions expressed in text. The sentiment mining and analysis of text is a significant branch task of natural language analysis technology [3]. The task of e-commerce text emotions analysis is the analysis of text with emotional

subjectivity [4], extraction of entities from valuable comments in the text, processing and summarization of them, and making reasonable inferences about the emotions expressed by reviewers. According to the granularity of the text processing, the task level of the emotions analysis is roughly classified into chapter fields, attribute fields, and sentence fields. The research in this paper aimed to conduct a sentiment analysis of these three categories of e-commerce texts.

The development of the sentiment analysis of text can be categorized into sentiment lexicon-based approaches, machine learning [5], and deep learning approaches [6]. Sentiment dictionaries are not flexible enough to adapt to new words in this new era. Machine learning methods require labor costs and cannot extract text features accurately from massive amounts of information. Our model uses a combined model based on deep learning, which has the advantages of high classification accuracy, strong applicability, and scalability. Text data are high-dimensional and complex, and traditional binary classification can only classify sentiment from positive or negative aspects. This does not fully cover the various emotions of users, and thus, natural language semantic sentiment analysis has limitations. Our model performs ternary classification on the emotion classification, which fills the gap left by traditional binary classification, which is not comprehensive enough in terms of emotion categories. The novelty of our model is reflected in the following aspects: First, as the traditional word vector model can only obtain static word representations, we used the current advanced BERT word vector pretraining model to generate dynamic word vectors. This dynamic word vector embedding is more accurate and facilitates downstream tasks. Secondly, we used BiGRU to extract richer contextual information representing emotional features to obtain a more comprehensive and deep text feature representation. The MultiHeadAttention mechanism can obtain attention information from multiple levels and obtain correlations among the data in different information spaces from different subspaces, which is an improvement in the attention mechanism. This research paper presents a neoteric model that combines BERT's dynamic representation of word vectors with other deep learning models. Compared with existing methods, our method introduced in this research shows good performance, and the accuracy and F1-score of this classification were significantly improved.

2. Related Work

This section reviews the progress on the research and development of emotions analysis. The technologies and advancements in commonly employed text sentiment classification primarily manifest in three aspects: sentiment lexicon-based approaches, approaches utilizing conventional machine learning algorithms, and approaches utilizing deep learning.

2.1. Research Using Sentiment Lexicon-Based Approaches

The text classification approach to the emotional lexicon is based on the constructed emotional dictionary. It employs an emotional lexicon to analyze the emotional value of words within the text, calculates the similarity between each word and the terms in the emotional lexicon, and then assigns weights to the calculated outcomes. Finally, the overall sentiment orientation of the content is determined through the result of the weighted calculation. Darwish et al. [7] presented a mixed sentiment classification approach that combines dictionary technology with fuzzy classification. To realize the establishment of a multilingual emotional dictionary, Ahmed [8] presented a weakly supervised network model that combines manual and automatic classification simultaneously. The sentiment label of the words in the different fields varied according to the extracted vocabulary, in one case, film and television reviews, as Tong [9] established a special emotional dictionary suitable for this field. Emotional dictionaries are not very flexible, and many new words will appear in this era of the rapid development of information technology, so emotional dictionaries face challenges in keeping up with the evolving times and are gradually being replaced.

2.2. Research Using Machine Learning Based Approaches

The traditional machine learning based approach differs from an emotional dictionary. It trains the model based on certain data and learns language knowledge from the marked data, including support vector machines, decision trees, and maximum entropy. By leveraging statistical algorithms, it extracts features from numerous marked and unmarked corpora to generate the ultimate sentiment analysis outcome. In a text sentiment classification study, Li et al. [10] applied the tag maximum entropy model to classify sentiment on datasets, such as from Weibo and Twitter, with an accuracy rate of 86.06%. To achieve the division of emotional polarity in the evaluation of various electronic product classifications, Kamal et al. [11] used the fusion of machine learning methods and rules to analyze features. Ruz et al. [12] used a Bayesian classifier to perform sentiment classification on a Spanish dataset and then training examples; compared to alternative approaches, the Bayesian network classifier performed better at the task of predicting emotional polarity. Based on a Twitter dataset, Hasan et al. [13] used a hybrid method combining naive Bayesian and support vector machines to design a sentiment classifier and compared sentiment analysis techniques in political analysis. Baid et al. [14] used methods such as maximum entropy and multinomial naive Bayes to test their sentiment classification effects on a movie review dataset, among which multinomial naive Bayes performed best, achieving 88.5%. Ahmad et al. [15] designed an optimized sentiment analysis framework (OSAF) using SVM technology; the OSAF framework uses cross-validation and SVM grid search methods to conduct experiments on the precision and recall with the three datasets. To detect strongly associated negative tweets, Birjali et al. [16] proposed an analysis algorithm based on the WordNet Linguistics English Dictionary training set, which uses semantic sentiment analysis and machine learning algorithms to extract negative ideas from Twitter data; this study showed good precision and accuracy in sentiment analysis for detecting negative thoughts. Mathapati et al. [17] designed an adaptive classifier of sentiment that selects reliable tweets from a specific domain and preserves feature weights at each iteration while also updating domain-adaptive words.

2.3. Research Using Deep Learning Based Approaches

Research using deep learning based approaches is currently the most popular method in the emotions analysis realm. Deep learning has high classification accuracy, strong applicability, and scalability, and it enjoys broad utilization in the domains of picture processing, speech processing, and sentiment analysis. For example, in order to make up for the shortcomings of RNN, with its risks of gradient explosion and gradient disappearance, Hochreiter [18] proposed a gate recurrent neural network, LSTM. Kim [19] was the first to use a CNN for text tasks, using convolution and pooling operations to extract the representative features in the text and then classifying the sentiment outcomes of the input. Jay et al. [20] used a hybrid deep learning model convolutional neural network (CNN) and gated recurrent unit (GRU) to conduct sentiment analysis on e-learning platform text data, which performed well in the classification. Tan et al. [21] proposed to use the LSTM-CNN model to extract text semantic information when dealing with answer selection tasks. Kumar et al. [22] proposed an interactive gated convolutional network (IGCN) that can learn the target and the corresponding context through a bidirectional gating mechanism. GRU [23] is an alternative implementation of the LSTM network; it keeps the structure simple while maintaining the LSTM's effect. Basiri et al. [24] used independent bidirectional GRU and LSTM layers to obtain contextual information and employed an attention-based network on this output layer. Madasu et al. [25] designed a sequential convolutional attention recurrent network (SCARN) model through experiments on sentiment analysis problems, it was found that the performance of the SCARN model was superior to other circular convolutional architectures. Gopalakrishnan et al. [26] proposed six LSTM models to conduct a sentiment analysis of an online Twitter debate dataset, and the model proves that the parameter settings and model layer settings will cause the experimental results to be different.

3. Related Theories

In this part, we present the basic modules within our method, namely, BERT, Multi-HeadAttention Mechanism, CNN, and BiGRU.

3.1. BERT

BERT is a neural-network-based bidirectional transformer encoder representation model for natural language preprocessing [27], and at its core is the technology of dynamically generated word vectors. Before the BERT model was introduced, Bengio [28] presented the NNLM method, which converts text into mathematical relationships and learns the text's characteristics. The Google team released Word2vec, which uses the idea of the co-occurrence of words. Its internal expansion is based on skip-gram and CBOW [29]. However, the words represented by Word2vec do not contain context. Neither Word2vec nor the improved GloVe has solved the problem of the word vectors obtained after training still being static. The traditional model can only represent each word in a fixed form. BERT consists of a bidirectional transformer encoder, which has strong feature extraction capabilities and can dynamically represent word vectors according to contextual words. Natural language has complex semantic features. In varying contexts, the identical word can convey distinct ideas. The traditional model generation of word vectors will reduce the accuracy of the downstream task results. Therefore, using BERT can dynamically generate text word vectors, making this word embedding more accurate, which is more comprehensive to the adhibition of downstream tasks.

A BERT model is obtained by spending a lot of training time and training costs on a large-scale corpus. A trained BERT model is suitable for model network parameters of general natural language processing tasks, and it finds extensive application in the downstream works of emotions analysis. Google has opened up the source code of the model, and when using it, we are solely required to adjust it to the pretrained model. It can learn word representation and dynamically generate word vectors based on the contextual information. In contrast to the static Word2vec method, the pretraining effectiveness of BERT can solve the problem of polysemous word representation, and it also has a stronger context fusion ability. The BERT construction is demonstrated in Figure 1 below. This article is a sentence-level sentiment analysis, which adds a [CLS] flag vector and a sentence-end [SEP] flag vector to each sentence; the [CLS] symbol is in front of words, and the symbol output outcome is used as the sentiment classification result of the input sentence; [SEP] is used as a separator between two sentences, so that using symbols different from words to output as sentiment classification results does not carry obvious semantic information, which can represent text information more objectively and output more effectively.

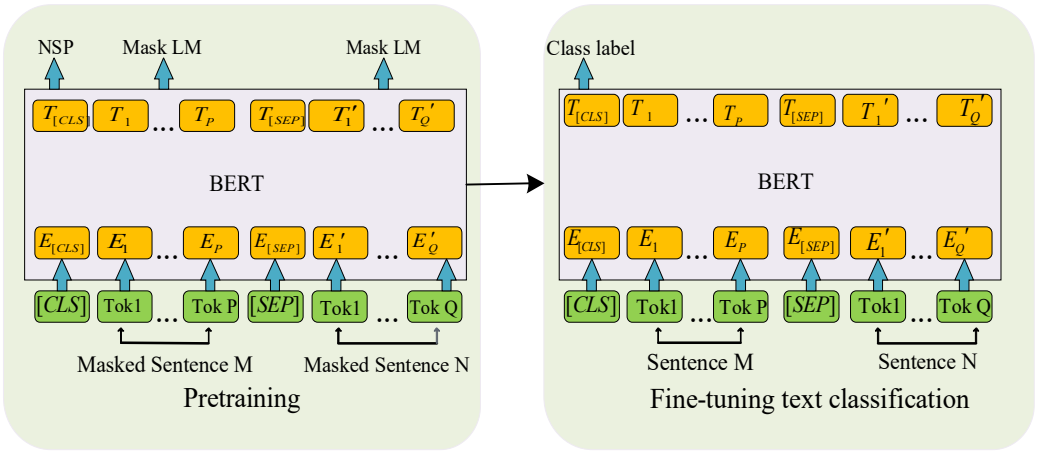

Figure 1. BERT model structure.

3.2. The MultiHeadAttention-Based Approach

The attention-based idea is largely inspired by human perception. When humans observe things with their eyes, they always focus on the important parts. Human beings have a focusing mechanism of attention in their thinking system. Researchers have designed an attention mechanism to simulate this mechanism so that machines can learn to perceive the important parts of data without paying too much attention to other irrelevant information. Bahdanau et al. presented an attention-based approach to natural language assignment [30], and researchers also combined the attention-based approach with basic methods, such as the CNN [31], LSTM [32], and RNN [33].

The Google team first proposed the MultiHeadAttention-based approach [34]. The MultiHeadAttention-based idea is an improvement on the attention approach. The previous attention approach only focused on a specific feature from one side, while the MultiHeadAttention-based method simulates human thinking from multiple aspects. It obtains correlation data in different information spaces from different subspaces by querying key–value pairs, maps the input to each information subspace to capture the correlations among data, and applies a series of mathematical conversions on the information matrix to obtain the attentional expression of the input content and more comprehensive emotional information. The MultiHeadAttention-based method can obtain attention messages from multiple levels, which has significant advantages in comparison to the conventional attention method. Therefore, this article used the MultiHeadAttention idea to distinctively pay close attention to the key information of the text, and its structure is shown in Figure 2 below.

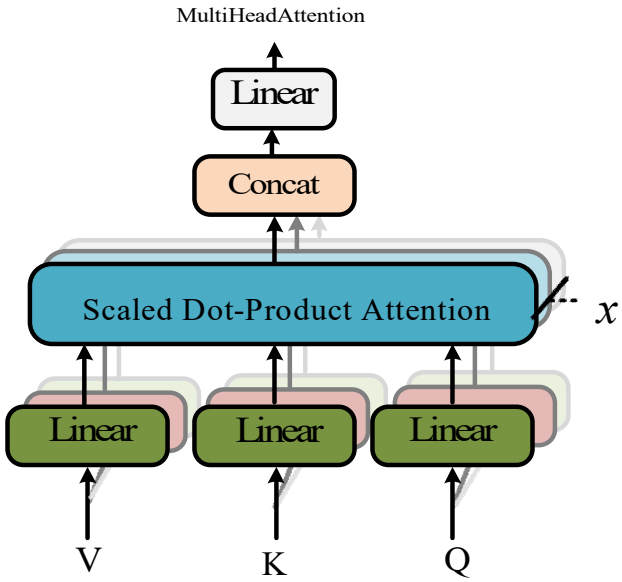

Figure 2. The MultiHeadAttention model's construction.

Suppose the input text $G = \left[g_1, g_2, \cdots, g_N\right]$, where g_N is the Nh word of the sentence G, and the word corresponds to a vector in D-dimensional space in which $G \in R^{N \times D}$. The matrix G is linearly transformed and cut to obtain a query matrix, a key matrix, and a value matrix, which are recorded as $Q \in R^{N \times D}$, $K \in R^{N \times D}$, and $V \in R^{N \times D}$, and they are mapped to different subspaces. The formula is as follows:

$$[Q_1, Q_2, \cdots, Q_x] = \left[QU^{Q_1}, QU^{Q_2}, \cdots, QU^{Q_x}\right] \quad (1)$$

$$[K_1, K_2, \cdots, K_x] = \left[KU^{K_1}, KU^{K_2}, \cdots, KU^{K_x}\right] \quad (2)$$

$$[V_1, V_2, \cdots, V_x] = \left[VU^{V_1}, VU^{V_2}, \cdots, VU^{V_x}\right] \tag{3}$$

In the above formula, U^{Q_i}, U^{K_i}, and U^{V_i} are transformation matrices, Q_i is the query matrix of the subspace, K_i is the key matrix of the subspace, and V_i is the value matrix of the subspace. Then, the attention value for each mapped subspace is computed, and the formula is as follows:

$$attention_i = softmax\left(\frac{Q_i, {K_i}^T}{\sqrt{d}}\right)V_i \tag{4}$$

In the above formula, $attention_i$ is the attention value of the subspace, and $\sqrt{d}$ represents the change of the matrix to a normal distribution. Then, the attention value of each mapped subspace is spliced. After the splicing operation is completed, the next linear transformation process is executed. Next, the input G and attention size of the entire input sentence are residually connected. Finally, the output matrix M is obtained.

3.3. CNN

CNN is an advanced deep learning model; it has a good feature extraction capacity and classification capacity. Its design is inspired by the processing of visual signals by the visual cortex in neuroscience. Compared with traditional machine learning algorithms, convolutional neural networks do not need to manually extract features, and they can automatically learn features and process high-dimensional data and time–space sequence data, such as images and natural languages. Convolutional neural networks were first widely used in the domain of machine images and were later used in the domain of text manipulation. At present, with the wide application and optimization of CNN models, many variants have emerged, such as the ResNet model, DenseNet model, and Inception network model. To enhance the capturing of the local relevance of the context, Kim [19] used a CNN to solve a text classification problem through sentence modeling and used multiple convolution kernels to extract the key information in the sentence by convolving the sentence matrix. After this, many researchers have also applied convolutional neural networks to text classification tasks, as they have the ability to lower the dimensionality of a text's content and improve the robustness [35]. Their structure is shown in Figure 3 below.

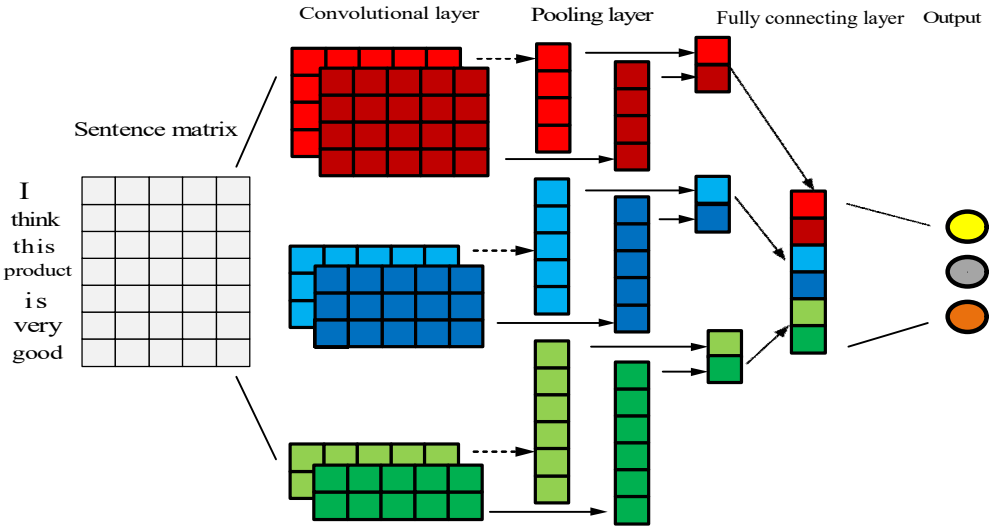

Figure 3. Convolutional neural network structure diagram.

A CNN includes two parts: convolution and pooling. The convolutional operation serves as the paramount component of a CNN. The convolutional operation uses a learnable kernel of convolution to slide the text context data to obtain the characteristics map that can capture the local and global characteristics of the data, and this effectively complete the

data representation. The pooling layer performs sampling on the feature layer, which can reduce the amount of memory and number of calculations required and can also enhance the application ability of the model and the invariance of the features. Convolution and pooling are used to extract local contextual information.

3.3.1. Generating the Feature Matrix of the Text

The text input W contains n words, and $W = [W_1, W_2, \cdots, W_n]$. When a CNN extracts the local features of a text, each word of context is first converted into a word embedding with dimensionality u. Next, the feature matrix corresponding to the text data W is obtained. The obtained word vector corresponding to each word W_i is expressed as follows:

$$W_i = (w_{i1}, w_{i2}, \cdots, w_{iu}), i \in [1, n] \tag{5}$$

The word embedding of all of the context in the text data W is sorted in sentence order to obtain the matrix representation; then, the text W is converted into a feature space $R^{n \times u}$ matrix; finally, the context words matrices are generated, which is fed into the convolution operation.

3.3.2. Convolution and Pooling

The feature W obtained above is fed into the convolution operation, the convolutional neural network performs repeated filtering operations on the corresponding positions on the text feature matrix W. The s filters perform an h-gram convolution in the generated text matrix according to the sliding window of the designated stride length; then, the filter operates on distinct submatrices produced by the context feature matrices when sliding, and the convolution kernel produces (n − h + 1) features. After the convolution operation, the feature map is generated so that the feature vector of the text can be obtained; when the filter operates on the words contained within each text content, the features of the text can be obtained.

The pooling layer follows the convolutional process, and the resulting features of the convolutional process are pooled in the following pooling process, which can reduce and purify the expression features obtained above. The pooling process has two pooling methods: one takes the maximum value, and the other takes the average value. The maximum value method has the capability to preserve the most crucial segment of the characteristic, and the average value method maintains the mean segment of the characteristic. The formulas corresponding to the maximum value method and the average pooling method are (6) and (7), respectively:

$$\hat{Q} = max\{C_i\} \tag{6}$$

$$a = \frac{\sum C_i}{s - h} \tag{7}$$

Finally, after the pooling layer, the fully connected layer connects the outputs of the previous two layers to identify and classify the input data.

3.4. BiGRU

An RNN is susceptible to two scenarios when processing sequence information, grads dispersion and grads blast, and an LSTM can avoid these problems. The GRU is obtained by simplifying the LSTM model, which is founded upon the LSTM. GRU better solves the long-term memory problem in RNN and the gradient problem in backpropagation, and it lets the method's construction remain simple while preserving the LSTM's role. GRU only has the reset gate r_t and the update gate z_t; r_t is used to indicate the severity of the information being ignored, and z_t is used to indicate the state of the information of a unit. GRU calculates the output of the current unit, which depends on the present feature import and the past unit's result, and then uses it as the import of subsequent units.

At a certain time t, the GRU unit performs continuous update processing. During the unit update process, z_t is employed to command the retention and forgetting of condition messages. The greater its value represents a higher number of messages received by the cell unit during the past time step. The equation for z_t is given by the following expression:

$$z_t = \sigma\left(\beta^{(z)} x_t + \gamma^{(z)} h_{t-1} + b^{(z)}\right) \tag{8}$$

The r_t is utilized to command the extent to which the condition information from the past time step is disregarded. If this value is very small, a larger amount of information from the past time step is disregarded. The calculation method of r_t is as follows:

$$r_t = \sigma\left(\beta^{(r)} x_t + \gamma^{(r)} h_{t-1} + b^{(r)}\right) \tag{9}$$

At time t, the computation of the candidate hidden state $\widetilde{h_t}$ and the hidden state h_t is determined by the following equations:

$$\widetilde{h_t} = \tanh\left(\beta^{(h)} x_t + \gamma^{(h)} r_t h_{t-1} + b^{(h)}\right) \tag{10}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t} \tag{11}$$

In the above formula, β and γ represent the weight distribution, σ represents the sigmoid function, x_t represents the input sequence information, $*$ represents multiplication, and b represents the bias term.

The one-way GRU only encodes the information from one direction and only considers the influence of the previous words on the latter. The context of the text will affect the words in the text, so we cannot ignore the influence of the following words on the previous words. Therefore, we added a reverse GRU and proposed a two-way GRU (BiGRU). In this network, two-way word dependencies are considered such that there are two transmission paths of information. Compared with BiLSTM [36], the BiGRU model reduces the complexity and number of model parameters and reduces the cost. It can manage features from both orientations. Compared with GRU, BiGRU not only has a stronger learning ability but can also extract richer contextual information, make full use of contextual information to enhance the effect of text semantic mining, and more effectively capture the relational features of the context such that the accuracy of the final output results is significantly improved. At time t, we assume that the forward status is $\overrightarrow{h_t}$ and the backward status is $\overleftarrow{h_t}$, and the value of a hidden status is weighted by $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$. A diagram of the model structure of the feature extraction using BiGRU is shown in Figure 4 below.

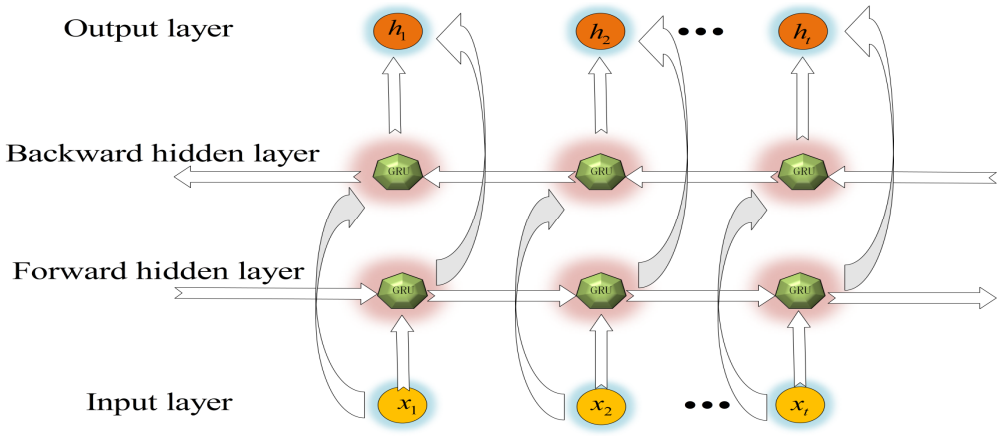

Figure 4. BiGRU model structure diagram.

4. Proposed Model

The structure presented in our paper is visually depicted in Figure 5 below. The structure includes an input layer, BERT word embedding layer, convolution layer, BiGRU layer, MultiHeadAttention layer, fully connected layer, and output layer.

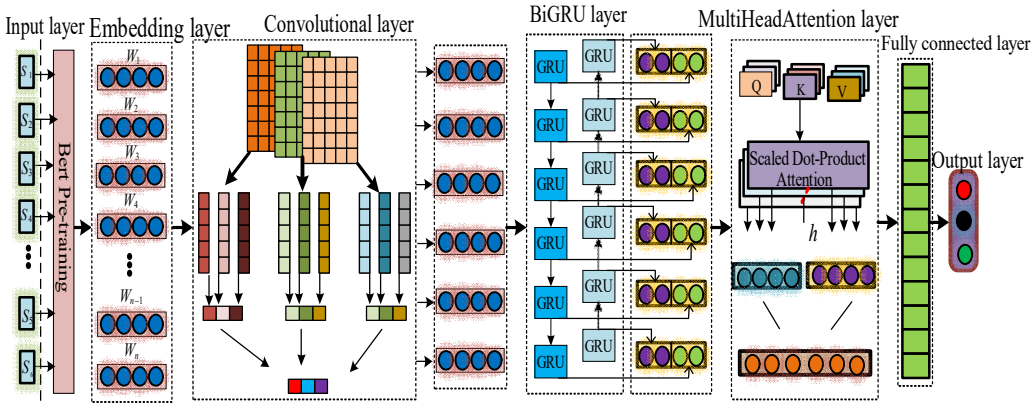

Figure 5. Proposed model structure.

4.1. Bert Word Vector Embedding Layer

This input sentence text is S, $S = [S_1, S_2, \cdots, S_i, \cdots, S_n]$, and S_n is the nth word of the text S. In the BERT layer, after passing the BERT model, the word S_i is transformed into W_i. The text is denoted as $W = [W_1, W_2, \cdots, W_i, \cdots, W_n]$, the words are converted to word vectors W_i of dimension u, $W_i = (w_{i1}, w_{i2}, \cdots, w_{iu}), i \in [1, n]$. All word vectors in the text data W were sorted according to the sentence order, and the matrix representation of W was obtained as shown in the following formula:

$$W_{1:n} = W_1 \oplus W_2 \oplus \cdots \oplus W_n \tag{12}$$

where W_n is the word vector representation of the nth word in the text content, n represents the length of the text, the word vector dimension is u, and $\oplus$ represents the connection operator.

Then, the text W is converted to a characteristic $R^{n \times u}$ matrix, and the generated text matrix is as follows:

$$W = \begin{pmatrix} w_{11} & \cdots & w_{1u} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nu} \end{pmatrix} \tag{13}$$

4.2. CNN Layers

The text feature matrix W serves as the input for the CNN layer, and s filters are used to perform h-gram convolution operations in the generated text matrix based on the designated stride length. This filter performs repeated filtering operations on the corresponding positions on the text feature matrix W and operates on various submatrices produced by text matrices when sliding, and the convolution kernel generates $n - h + 1$ features. The convolution operation can obtain the feature vector of the sentence, and the equation for generating the feature map is given by the following expression:

$$q_i = f(U \times W_{i:i+h-1} + d) \tag{14}$$

In the above formula, f is a nonlinear activation function, d is a bias item, and U is a weight. The filter can obtain the following text features by operating on the words in each sentence:

$$\mathrm{Q} = [q_1, q_2, \cdots, q_{n-h+1}], Q \in R^{n-h+1} \tag{15}$$

Then, the pooling layer performs a pooling operation after the convolutional operation. The pooling operation can reduce the dimensions and purify the text features, reducing

the complexity of the calculations. The maximum pooling method can extract the significant feature of the text, so we chose the maximum pooling method. The maximum pooling method formula and the final output of the CNN layer are shown in the following Equations (16) and (17):

$$\hat{Q} = max\{q_i\} \tag{16}$$

$$\mathrm{C} = [Q_1, Q_2, \cdots, Q_{n-h+1}] \tag{17}$$

In addition, we added a dropout layer to the CNN to prevent overfitting and improve the model's performance.

4.3. BiGRU Layers

The word vector C with text emotional features outputted by the CNN layer serves as the input for BiGRU. BiGRU is an improvement to the GRU. It consists of two relatively independent GRU with opposite directions; therefore, through two channels, the previous state and the future state jointly determine the current output. It obtains the previous information by considering the context information at the same time, and it can also obtain the impact of the feature at the next moment based on the current information, which can extract the text feature more effectively. It not only has a stronger learning ability but can also extract richer contextual information, making full use of contextual information to enhance the effect of semantic text mining. The data input by the CNN layer are passed to the forward and backward hidden layers at the same time; after the forward GRU and the reverse GRU, the input word vectors are processed in both directions. The data flow into two forward and reverse GRU networks at the same time, and word vectors are extracted to provide richer information representing emotional features; this has the capability to leverage text information to its fullest extent and enhance the mining effect of text semantics. The outcome of the output layer is jointly determined by two GRU in opposite directions, which significantly enhances the accuracy of the ultimate outcome.

During the training process, BiGRU uses two *GRU* with opposite front and rear directions to encode information bidirectionally.

The computation formula for the forward GRU is given by the following expression:

$$\overrightarrow{h_t} = GRU\left(C, \overrightarrow{h_{t-1}}\right), t \in [1, n-h+1] \tag{18}$$

The computation formula for the reverse GRU is given by the following expression:

$$\overleftarrow{h_t} = GRU\left(C, \overleftarrow{h_{t+1}}\right), t \in [n-h+1, 1] \tag{19}$$

Then, the value of a hidden state is weighted by $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$. The output expression combined with contextual emotion is as follows:

$$H_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{20}$$

The BiGRU network extracts the contextual semantic information in the text, and the final output is as follows:

$$H = [H_1, H_2, \cdots, H_{n-h+1}] \tag{21}$$

4.4. MultiHeadAttention Layer

The information extracted at the BiGRU layer contains rich context features. H goes into the attention layer, and the MultiHeadAttention mechanism can selectively extract the words related to emotion in the text. The traditional attention mechanism is limited to acquiring attention information solely from a single level. In our research, MultiHea-

dAttention is used to obtain correlations among data in different information spaces from different subspaces.

The word vector semantic information H, extracted by the BiGRU network, is first linearly transformed and cut into three matrices, Q, K, and V, in the same dimension, and the attention value of each mapped subspace is calculated according to the above Formula (4), and then the attention value of each subspace is spliced. After the splicing operation is completed, the next linear transformation operation is performed. The calculation formula is as follows:

$$Multi\_attention = concat(attention_1, attention_2, \cdots, attention_x)U^O \tag{22}$$

where *Multi_attention* represents the attention value of the entire input sentence, *concat* represents splicing the attention value, and U^O is the transformation matrix. After, the attention value *Multi_attention* and H of the entire sentence are residually spliced. The formula is as follows:

$$M = residual\_connect(H, Multi\_attention) \tag{23}$$

In the above formula, for the sentence matrix $M \in R^{N \times D}$, M is the outcome of the MultiHeadAttention, *residual_connect* represents the residual connection, and finally, the sentence matrix is M, which is obtained after the residual connection is the final result of the MultiHeadAttention layer.

4.5. Softmax Output Layer

The sentence is represented by $BERT$ dynamically generated word vectors, and the local information of the text is derived via the utilization of convolution and pooling operations, and then $BiGRU$ is used to extract richer contextual information representing emotional features to obtain a more comprehensive and deep text feature representation, and next the $MultiHeadAttention$ layer selectively extracts the emotion-related features in the context features. Finally, to forecast the sentimental classes of the sentence, the result M is obtained using the $MultiHeadAttention$ layer, which is converted into the final emotion representation X with the fully connected layer, and the $softmax$ formula is utilized for carrying out three classification tasks, finally obtaining the input emotion representation:

$$Y = softmax(\psi_c X + d_c) \tag{24}$$

In the above formula, ψ_c represents the weight matrix, while d_c corresponds to the bias term.

5. Experiments and Analysis

In this section, we analyze the experimental results and model evaluation of the emotion research task, including the selection and division of the datasets, the selection of the hyperparameters, and the selection of the evaluation indicators, and judge the performance of the method based on the relevant parameters. We compare the model used in our research with other deep learning models and set up an ablation experiment according to the method used in our research for a performance comparison.

5.1. Dataset

The dataset used is a comment dataset for an e-commerce platform that has been screened. The emotions expressed in this selected dataset are partitioned into three emotional categories: attitudes that express positive emotions, attitudes that express negative emotions, and attitudes that express neutral emotions. Among them, the results of the three emotional marks in this paper are P (indicating text with a positive sentiment), N (indicating text with a negative sentiment), and Q (representing neutral text). This paper selected 9120 comment data that express positive emotions, 9660 pieces of comment

data showing negative emotions, and 4960 pieces of comment data showing neutral emotions. The collective situation of the emotional polarity labels in the dataset is presented in Table 1 below.

Table 1. Examples of polarity labels in the dataset.

Emotional Category	List of Dataset Contents	Label
Positive	"It feels very good and is very convenient to use."	P
Negative	"Really regret buying it, not good at all."	N
Neutral	"Overall, there are good points, but there are also bad points."	Q

5.2. Evaluation Indicators

This article selected four evaluation criteria that are extensively utilized in the field of emotions analysis: accuracy, precision, recall, and F1–score. This article conducted a three-category sentiment analysis on the text, and the training dataset and the test dataset were partitioned at a 4:1 ratio. For the selected evaluation index m = 3, and the evaluation index measurement method adopted the macro method and then evaluated the four evaluation standards in this way.

5.2.1. Accuracy Rate

The accuracy rate demonstrates a model's capability to evaluate the entire dataset, indicating the ability of the model to correctly judge the results as positive, neutral, and negative. It denotes the portion of accurately classified samples in the overall sample, and the equation is given by the following formula:

$$Accuracy = \frac{\sum_{j=1}^{m} TP_j}{\sum_{j=1}^{m} TP_j + FP_j} \tag{25}$$

5.2.2. Precision Rate

The precision rate is the proportion of the count of samples precisely forecasted as class j to the entire count of samples forecasted as class j, and the equation is given by the following formula:

$$Precision = \frac{1}{m}\sum_{i=j}^{m} \frac{TP_j}{TP_j + FP_j} \tag{26}$$

5.2.3. Recall Rate

The recall rate is the correct prediction of the count of samples of class j compared to the actual number of samples of category j, and the formula is as follows:

$$Recall = \frac{1}{m}\sum_{j=1}^{m} \frac{TP_j}{TP_j + FN_j} \tag{27}$$

5.2.4. F1-Score Rate

The F1–score considers considers both factors precision and recall, and reflects the classification capacity more comprehensively. The better the performance of the classifier, the closer the F1 score is to 1, and it is based on the harmonic average of the precision and recall. The formula is as follows:

$$F1\text{–}Score = \frac{2Precision \times Recall}{Precision + Recall} \tag{28}$$

5.3. Experimental Data Processing and Its Progress

To enhance the model's performance, this research employed jieba word segmentation for the text segmentation and removed stop words and illegal characters after the word segmentation. During the data reading process, the dataset's files were read and stored in

three arrays, the labels were merged into one label array, and the text was tokenized using Tokenizer; that is, the text was converted into a sequence of numbers. The experimental operation's process was completed on a Windows 10 operating system, using CPU for the training, the programming language used was Python, the development tool used was PyCharm, the deep learning framework used was TensorFlow, and the processor was an Intel(R) Core (TM) i5-5200U CPU RAM = 12.0 GB.

5.4. Determination of Experimental Parameters and Experimental Results

In this section, we select the hyperparameters, perform the individual training and separate tuning of the hyperparameters, select a combination of hyperparameters related to the model, and conduct comparative training with other hyperparameter combinations to determine the optimal hyperparameters of the model in this paper's combination of parameters.

The model's manifestation can be significantly influenced by the number of iterations. When the number of iterations is too limited, the ability to learn features is insufficient, and the effect is not good. Excessive iterations can lead to the occurrence of overfitting, which will not make the model perform better. The outcomes of the tests are displayed in Table 2 and Figure 6. Once the counts of iterations surpassed 20, the model's results started to deteriorate, which can be attributed to the occurrence of overfitting. When the number of iterations fell below 20, the model had an insufficient ability to learn features, resulting in the effect being nonoptimal. From the analysis, 20 was chosen as the optimal iterative parameter for the model in this paper.

Table 2. Iterative experiment results.

Epochs	Accuracy	Precision	Recall	F1-Score
8	86.0%	87.2%	88.1%	87.6%
11	86.5%	87.9%	88.8%	88.3%
14	87.7%	88.6%	89.0%	88.7%
17	88.6%	89.2%	90.1%	89.6%
20	89.4%	90.3%	91.0%	90.6%
23	88.9%	89.5%	90.4%	89.9%
26	87.7%	88.2%	88.9%	88.5%

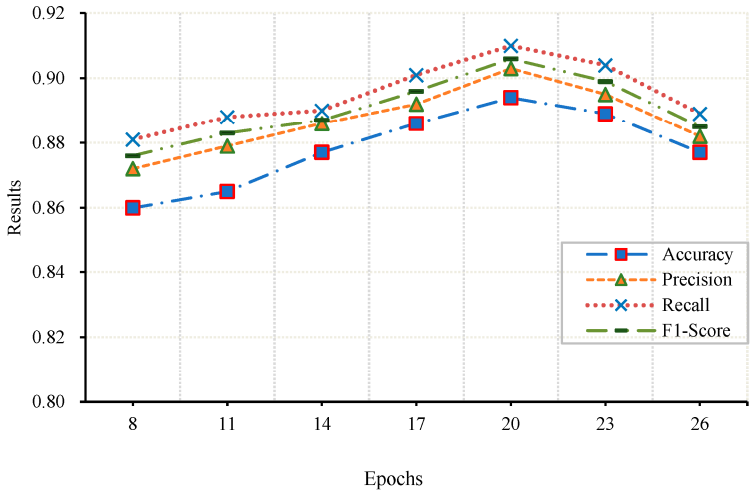

Figure 6. Experimental results diagram of the iterative experiment.

The performance of the model was affected by the text input's length. A longer text length requires filling in many zero values, which will reduce the performance and accuracy of the representation. If the text input is too short, the review length will be greater and needs to be intercepted, thereby affecting the emotional characteristics of the captured

text. Through statistical calculations, the length of most sentences in our dataset was concentrated at about 180 words, so we selected comparison parameters in an appropriate value range for the experiments. The outcomes of these experiments are displayed in Table 3 and Figure 7. When the text input length was 210, the experiment achieved the best results, which were roughly consistent with the previous statistics. Therefore, this paper chose 210 as the text input length, which is close to the sentence length of most input data.

Table 3. Text length experiment results.

Text Length	Accuracy	Precision	Recall	F1-Score
170	88.3%	89.4%	90.2%	89.7%
190	88.7%	90.1%	90.3%	90.1%
210	89.8%	90.5%	91.4%	90.9%
240	89.4%	89.9%	90.3%	90.0%

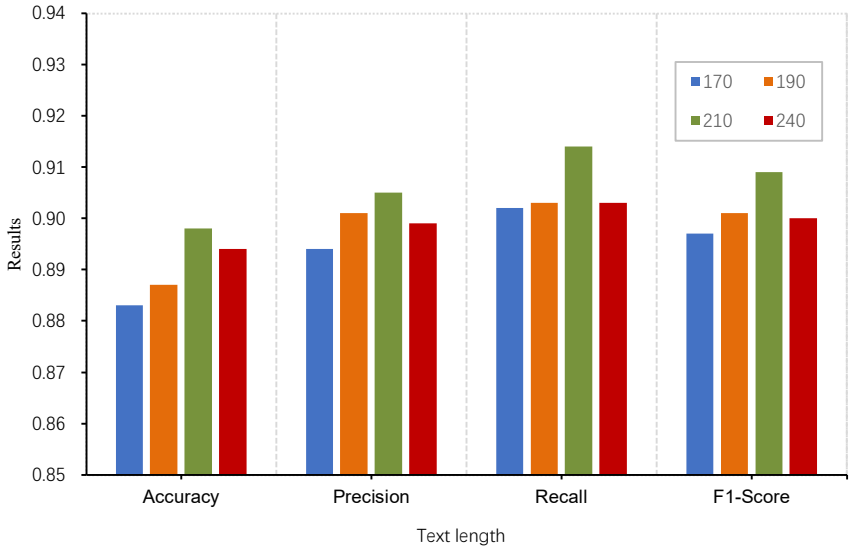

Figure 7. Text length experiment results.

The model will have an overfitting phenomenon during the training process, and in order to avoid overfitting, we introduced a dropout value in the research. The phenomenon of overfitting usually shows that the loss function of the training data will be relatively small, and the accuracy rate will be relatively high; however, the prediction accuracy rate on the test data will be low, and the loss function will be relatively large. The introduction of the dropout value will reduce the complex co-adaptive relationship among neurons so that it will not rely too much on some local features, which can enhance the model's capacity for generalization and enhance the performance of the neural network. As shown in Table 4 and Figure 8 below, overfitting can be effectively avoided when dropout = 0.5 and the model's performance is the best at this time.

Table 4. Dropout value experimental results.

Dropout	Accuracy	Precision	Recall	F1-Score
0.2	87.1%	87.4%	88.4%	87.8%
0.3	87.8%	88.3%	88.9%	88.5%
0.4	88.3%	88.7%	89.5%	89.0%
0.5	88.7%	89.4%	90.2%	89.7%
0.6	88.1%	88.6%	89.2%	88.8%
0.7	87.5%	87.8%	88.8%	88.2%

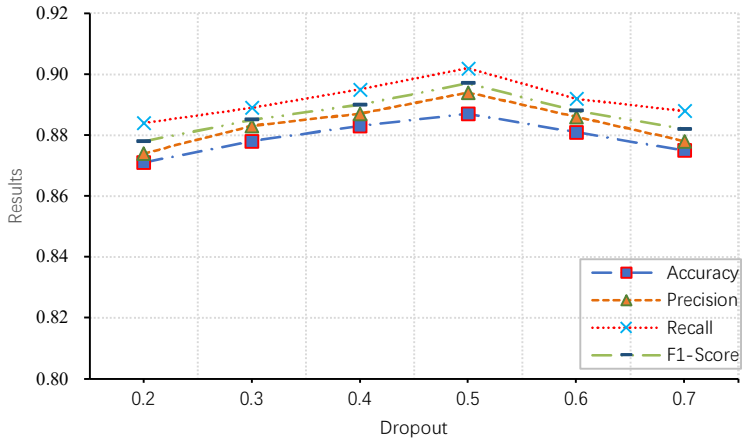

Figure 8. Dropout value experimental results.

The learning rate is an important hyperparameter in deep learning. It is used to control the parameter update range of the model during the training and decide the speed and step size of renewing the weight parameters in iterations. Our goal is to optimize the weight parameters by minimizing the loss function. The learning rate controls the magnitude during the parameter update. Increasing the learning rate can speed up the convergence process, and it may also result in oscillating around the best solution or not converging to the optimal solution. In the event that the learning rate is too diminutive, the loss will exhibit no changes over an extended duration and the optimization process will be very slow, or even impossible to achieve optimal solution.

Using an appropriate learning rate is a key step during the training. We needed to dynamically adjust the learning rate to balance the convergence speed and model performance during the training, such as through learning rate scheduling or using adaptive learning rate algorithms. This paper dynamically adjusted the learning rate and combined this result under various learning rates and the change trend of the loss function. Through experiments and data analysis, the model in this paper works best when the learning rate is 0.0001. The experiment is shown in Table 5 and Figure 9.

Table 5. Experimental results of the learning rate.

Learning Rate	Accuracy	Precision	Recall	F1-Score
0.01	86.9%	87.5%	88.0%	87.7%
0.001	87.9%	88.1%	89.3%	88.6%
0.0001	88.6%	89.3%	90.1%	89.6%
0.00001	88.1%	88.4%	89.1%	88.7%
0.000001	86.8%	87.0%	87.4%	87.1%

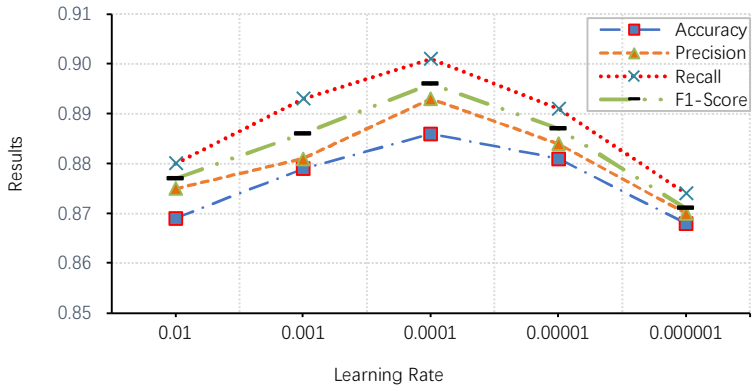

Figure 9. Experimental results of the learning rate.

The batch size refers to the quantity of text samples that are provided as input to the model in one iteration. If the value is too large, the model may fall into a poor local optimum and cannot achieve a better performance. If the batch size is too large, multiple data can be processed at the same time to learn more detailed features, but this may cause the model to overfit the training data. In addition, a larger batch size requires more memory to store input data and gradient information; if the memory is insufficient, training may fail. On the contrary, too small of a batch size will increase the training time, which may lead to instability in the training process and may also lead to underfitting of the model, making the model's generalization capacity worse, because it can only see a small amount of data, making it difficult to learn global features. Therefore, selecting the appropriate batch size is crucial to attain the desired research outcome. As shown in Table 6 and Figure 10, the experiment shows that when the batch size is 22, the model achieves the best effect.

Table 6. Batch size experimental results.

Batch Size	Accuracy	Precision	Recall	F1-Score
22	89.9%	90.6%	91.2%	90.8%
44	89.3%	90.1%	90.8%	90.4%
88	88.7%	89.0%	90.5%	89.7%
176	88.1%	88.6%	90.1%	89.3%
352	87.8%	88.2%	89.3%	88.7%

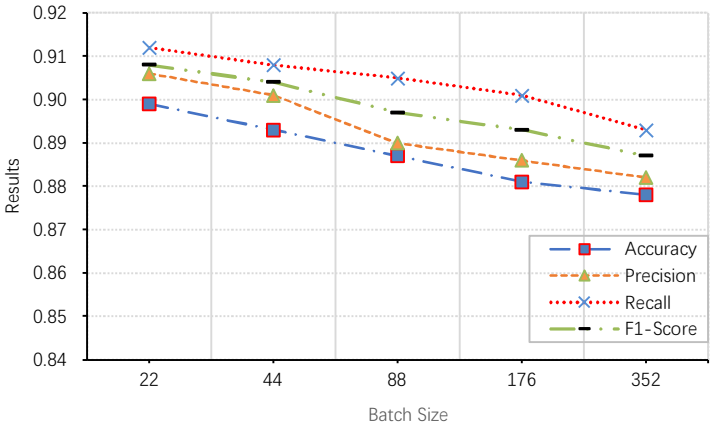

Figure 10. Batch size experiment results.

Through the above experiments, the optimal hyperparameter combination of the model in this paper is summarized. The hyperparameter combination is presented in the following Table 7.

Table 7. Model parameter settings.

Parameter	Parameter Value
Number of attention heads	8
Learning rate	0.0001
Word vector dimension for attention	128
Epoch value	20
Number of BiGRU hidden layers	64
Entered text length	210
Convolution kernel size	3
Dropout value	0.5
Batch size value	22
Number of convolution kernels	250

5.5. Model Comparison

To validate the efficacy of our model at text sentiment analysis, the following comparative experiments were designed. We selected seven representative deep learning models to conduct comparative experiments on the datasets used in this paper [15,34–39]. The comparison model is segmented into three components: the initial part is a BERT-based model, the second component is a conventional deep learning model method, and the third component is a combination model related to the model in this paper.

By analyzing Table 8, it is evident that the effect of the traditional deep learning model is not satisfactory. The model presented in our research has significant advantages in *A*ccuracy, *Precision*, *Recall*, and *F*1–*score*. This is because this article introduces the BERT language model to convert text into dynamic word vectors and then combines MultiHeadAttention, CNN, and BiGRU . Among these, BiGRU has lower model complexity and simple parameters, which reduces the cost. It extracts richer contextual information from two directions and makes full use of contextual information to enhance the text's semantic mining effect. Therefore, the model that combines BiGRU and CNN performs better than the CNN model alone, with a 1.5% increase in *A*ccuracy and a 1.9% increase in the *F*1–*Score*. In addition, incorporating the attention idea into the traditional deep learning model can further improve the performance of the model, as it can capture the text content related to emotional color. Our research used the MultiHeadAttention method, which was obtained after optimization and improvement of the attention mechanism. It can obtain attention information at multiple levels and obtain the correlation among data in different information spaces from different subspaces. By subjecting the text to multiple linear transformations, it can acquire a comprehensive understanding of emotional information through learning the attention representation. After adding the BERT pretrained method, the emotion prediction performance experienced a substantial improvement; this is attributed to the fact that the traditional language models Word2vec and GloVe have not solved the problem of word vectors obtained after training still being static. BERT is an advanced pretrained language model that can dynamically generate word vectors according to the context, which can make word representation more accurate and have stronger context fusion capabilities, and it is more conducive to the development of downstream tasks.

Table 8. Experimental results of the comparative models.

Models	Accuracy	Precision	Recall	F1-Score
CNN [19]	83.4%	85.2%	86.3%	85.7%
BiGRU [37]	84.7%	85.8%	86.7%	86.2%
CNN + Att [38]	85.2%	87.3%	88.4%	87.8%
BERT [39]	81.8%	82.6%	84.9%	83.7%
BERT + CNN [40]	87.4%	88.7%	89.0%	88.8%
BiGRU + Att [41]	87.3%	89.2%	89.6%	89.3%
BiGRU + CNN [42]	86.2%	87.0%.	88.3%	87.6%
Ours	89.4%	90.3%	91.1%	90.6%

The experimental outcome indicates that the classification accuracy of our proposed method was better than that of all of the compared models, which fully reflects the advantage of our method over the other models. The following aspects serve as the primary manifestations of this: We first used the advanced BERT method to convert the input word into a dynamic word vector, which can obtain a more comprehensive and accurate word vector embedding. Afterwards, a CNN was employed for the extraction of crucial features of the text through convolution and pooling operations, and then we used BiGRU to extract richer contextual information representing emotional features to obtain a more comprehensive and deep text feature representation. Finally, we used the MultiHeadAttention mechanism to obtain correlations among the data in different information spaces from different subspaces to selectively extract the features related to emotion in the context features and obtain more comprehensive emotional information.

5.6. Ablation Experiment

With the aim of evaluating the effect of employing the MultiHeadAttention mechanism and BiGRU in our research, we set up comparative experiments and conducted ablation experiments on the basis of our proposed model.

BERT + CNN: BiGRU and MultiHeadAttention were removed from our structure, while BERT and CNN were retained. The BERT model transformed the input text into dynamic word vectors, and then it used the CNN model to perform convolution and pooling operations. Finally, the outcome entered the output layer.

BERT + CNN + BiGRU: MultiHeadAttention was removed from our model, and BiGRU was retained. The remaining model was BERT + CNN + BiGRU. BERT transformed the input text into dynamic word vectors, and then the CNN performed convolution and pooling operations; afterwards, BiGRU extracted richer contextual information representing emotional features. Finally, the outcome entered the output layer. This group of experiments studied the impact of the introduction of BiGRU on the model's performance.

BERT + CNN + M_att: BiGRU was removed from the model, and MultiHeadAttention was retained. The remaining model was BERT + CNN + MultiHeadAttention. The BERT model converted the input text into dynamic word vectors, and then it used the CNN model to perform convolution and pooling operations; afterwards, it used the MultiHeadAttention to extract the features related to emotion in the context features. Finally, the outcome entered the output layer. This group of experiments studied the impact of introducing MultiHeadAttention on the model's performance.

Our model: BERT + CNN + BiGRU + MultiHeadAttention.

Table 9 and Figure 11 display the test outcomes of the ablation experiments. It is better to retain all methods of our model than removing BiGRU and MultiHeadAttention at the same time. The *A*ccuracy rate exhibited a 2% increase. After adding BiGRU, the model's accuracy increased by 1.3%. This is because BiGRU can more effectively capture the distant dependency relationship in the context, thereby extracting richer contextual information in the text representing emotional features and can obtain a more comprehensive and deep text feature representation, thereby improving the model's effect. Furthermore, it is evident that adding the MultiHeadAttention mechanism can improve the model's manifestation, and the model achieved a 1.1% increase in accuracy. The sentiment word representing the sentiment color had a good effect on predicting the text's polarity. We introduced the MultiHeadAttention mechanism from different subspaces to obtain correlations among the data in different information spaces to selectively extract features related to emotions in the context features and obtain more comprehensive and important emotional information, thereby enhancing the model's performance.

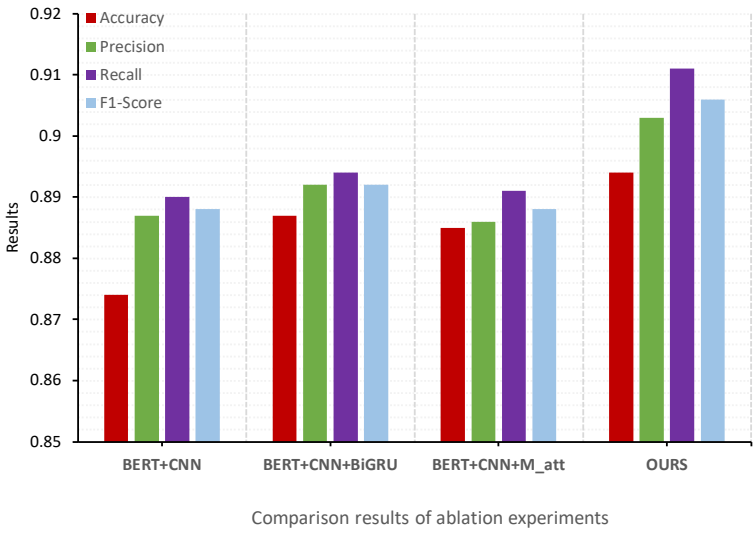

Figure 11. Comparison of the model outcomes from the ablation experiments.

Table 9. Comparison of the model outcomes from the ablation experiments.

Model	Accuracy	Precision	Recall	F1-Score
BERT + CNN	87.4%	88.7%	89.0%	88.8%
BERT + CNN + BiGRU	88.7%	89.2%	89.4%	89.2%
BERT + CNN + Mu_att	88.5%	88.6%	89.1%	88.7%
Ours	89.4%	90.3%	91.1%	90.6%

6. Conclusions

Deeping learning is currently the most popular method in the sentiment analysis domain. The performance of traditional deep learning models, in many aspects, can be further improved. We propose a new model that combines BERT and other deep learning model architectures. This model first employs BERT to convert the input text into dynamic word vectors and complete the word vector embedding. Furthermore, it uses a CNN model to perform the convolution and pooling operations to extract important features of the text, which can further decrease the dimensions of the feature vectors and enhance the model's robustness. Then, it uses BiGRU to extract richer context information from the text to obtain a comprehensive text feature representation. Finally, the MultiHeadAttention mechanism is used to obtain the features of words representing strong emotions from the entire text, further enhancing the degree of emotion. By comparing with other models listed in the literature, our research method acquired the optimal experimental results, and the accuracy and F1-score of the classification significantly improved.

The current research work indicates a direction for future research. Firstly, emotional words representing emotional color have a great effect on predicting emotional tendency; in the future, we can introduce advanced emotional lexicons and further enhance the emotional strength of word vectors, thereby further improving the accuracy of the experiment. Secondly, the combination model in this paper is a serial structure, and the applicability may not be as extensive as a parallel mechanism, which inspires us to use a parallel mechanism for the model as a next step.

Author Contributions: Conceptualization, H.L. and J.W.; Methodology, J.W. and H.L.; Software, J.W. and Y.L.; Validation, J.W.; Resources, H.L. and H.Z.; Writing—original draft preparation, J.W.; Writing—review and editing, J.W. and H.L.; Formal analysis, J.W. and J.M.; Investigation, J.W.; Data Curation, J.W.; Visualization, J.W.; Supervision, J.W. and H.L.; Project Administration, H.L.; Funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Project of Science and Technology Research in Henan Province of China (No. 232102210035).

Data Availability Statement: The data presented in this study can be provided upon request.

Acknowledgments: The authors would like to thank the editors and the anonymous reviewers for their helpful comments and suggestions, which have improved the presentation.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cao, J. Big Commerce Data Knowledge Representation. In *E-Commerce Big Data Mining and Analytics. Advanced Studies in E-Commerce*; Qin, Z., Shuai, Q.H., Zhang, R.G., Ye, Q.W., Xiong, L., Cao, J., Eds.; Springer: Singapore, 2023; pp. 99–111.
2. Raghunathan, N.; Kandasamy, S. Challenges and Issues in Sentiment Analysis: A Comprehensive Survey. *IEEE Access* **2023**, *11*, 69626–69642. [CrossRef]
3. Hussain, A.; Cambria, E. Semi-supervised learning for big social data analysis. *Neurocomputing* **2018**, *275*, 1662–1673. [CrossRef]
4. Huang, H.; Adeleh, A.Z.; Mumtaz, B.M. Sentiment Analysis in E-Commerce Platforms: A Review of Current Techniques and Future Directions. *IEEE Access* **2023**, *11*, 90367–90382. [CrossRef]
5. Wawre, S.V.; Deshmukh, S.N. Sentiment classification using machine learning techniques. *Int. J. Sci. Res.* **2016**, *5*, 819–821.
6. Asudani, D.S.; Nagwani, N.K.; Singh, P. Impact of word embedding models on text analytics in deep learning environment: A review. *Artif. Intell. Rev.* **2023**, *56*, 10345–10425. [CrossRef] [PubMed]

7. Madbouly, M.M.; Darwish, S.M.; Essameldin, R. Modifed fuzzy sentiment analysis approach based on user ranking suitable for online social networks. *IET Softw.* **2020**, *14*, 300–307. [CrossRef]
8. Ahmed, M.; Chen, Q.; Li, Z.H. Constructing Domain-Dependent Sentiment Dictionary for Sentiment Analysis. *Neural Comput. Appl.* **2020**, *32*, 14719–14732. [CrossRef]
9. Tong, R.M. An operational system for detecting and tracking opinions in on-line discussions. In Proceedings of the Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification, New York, NY, USA, 1–3 June 2001; pp. 1–6.
10. Li, J.; Rao, Y.; Jin, F.; Chen, H.; Xiang, X. Multi-label maximum entropy model for social emotion classification over short text. *Neurocomputing* **2016**, *210*, 247–256. [CrossRef]
11. Kamal, A.; Abulaish, M. Statistical features identification for sentiment analysis using machine learning techniques. In Proceedings of the 2013 International Symposium on Computational and Business Intelligence, New Delhi, India, 24–26 August 2013.
12. Ruz, G.A.; Henriquez, P.A.; Mascareno, A. Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers. *Future Gener. Comput. Syst.* **2020**, *106*, 92–104. [CrossRef]
13. Hasan, A.; Moin, S.; Karim, A.; Shamshirband, S. Machine learning-based sentiment analysis for twitter accounts. *Math. Comput. Appl.* **2018**, *23*, 11. [CrossRef]
14. Baid, P.; Gupta, A.; Chaplot, N. Sentiment analysis of movie reviews using machine learning techniques. *Int. J. Comput. Appl.* **2017**, *179*, 45–49. [CrossRef]
15. Ahmad, M.; Aftab, S.; Bashir, M.S.; Hameed, N.; Ali, I.; Nawaz, Z. SVM optimization for sentiment analysis. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 393–398. [CrossRef]
16. Birjali, M.; Beni-Hssane, A.; Erritali, M. Machine learning and semantic sentiment analysis based algorithms for suicide sentiment prediction in social networks. *Procedia Comput. Sci.* **2017**, *113*, 65–72. [CrossRef]
17. Mathapati, S.; Nafeesa, A.; Manjula, S.H.; Venugopal, K.R. OTAWE-Optimized topic-adaptive word expansion for cross domain sentiment classification on tweets. *Adv. Mach. Learn. Data Sci.* **2018**, *705*, 213–224.
18. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
19. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arXiv* **2014**, arXiv:1408.5882.
20. Jay, K.D.; Anupam, D.; Joann, R. A Hybrid Deep Learning Technique for Sentiment Analysis in E-Learning Platform with Natural Language Processing. In Proceedings of the 2022 International Conference on Software, Telecommunications and Computer Networks, Split, Croatia, 22–24 September 2022.
21. Tan, M.; Santos, C.D.; Xiang, B.; Zhou, B. LSTM-based Deep Learning Models for non-factoid answer selection. *arXiv* **2016**, arXiv:1511.04108.
22. Kumar, A.; Narapareddy, V.T.; Srikanth, V.A.; Neti, L.B.M.; Malapati, A. Aspect-based sentiment classification using interactive gated convolutional network. *IEEE Access* **2020**, *8*, 22445–22453. [CrossRef]
23. Cho, K.; Merrienboer, B.V.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
24. Basiri, M.E.; Nemati, S.; Abdar, M. ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Gener. Comput. Syst.* **2021**, *115*, 279–294. [CrossRef]
25. Madasu, A.; Rao, V.A. Sequential learning of convolutional features for effective text classification. *arXiv* **2019**, arXiv:1909.00080.
26. Gopalakrishnan, K.; Salem, F.M. Sentiment analysis using simplified long short-term memory recurrent neural networks. *arXiv* **2020**, arXiv:2005.03993.
27. Devlin, J.; Chang, M.W.; Lee, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
28. Bengio, Y.; Ducharme, R.; Vincent, P. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
29. Mikolov, T.; Sutskever, I.; Chen, K. Distributed representations of words and phrases and their compositionality. *arXiv* **2013**, arXiv:1310.4546.
30. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
31. Yin, W.; Schütze, H.; Xiang, B.; Zhou, B. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 259–272. [CrossRef]
32. Yang, M.; Tu, W.; Wang, J.; Xu, F.; Chen, X. Attention based LSTM for target dependent sentiment classification. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 5013–5014.
33. Liu, B.; Lane, L. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv* **2016**, arXiv:1609.01454.
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, L. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
35. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [CrossRef]
36. Huang, P.; Zheng, L.; Wang, Y.; Zhu, H.J. Sentiment Analysis of Chinese Text Based on CNN-BiLSTM Serial Hybrid Model. In Proceedings of the 2021 10th International Conference on Computing and Pattern Recognition, Shanghai, China, 15–17 October 2021; pp. 309–313.
37. Yin, X.; Liu, C.; Fang, X. Sentiment analysis based on BiGRU information enhancement. *J. Phys. Conf. Ser.* **2021**, *1748*, 032054. [CrossRef]

38. Su, Y.J.; Chen, C.H.; Chen, T.Y.; Cheng, C.C. Chinese microblog sentiment analysis by adding emoticons to attention-based CNN. *J. Internet. Technol.* **2020**, *21*, 821–829.
39. Alaparthi, S.; Mishra, M. BERT: A sentiment analysis odyssey. *J. Mark. Anal.* **2021**, *9*, 118–126. [CrossRef]
40. Liu, N.; Zhao, J.H. A BERT-Based Aspect-Level Sentiment Analysis Algorithm for Cross-Domain Text. *Comput. Intell. Neurosci.* **2022**, *2022*, 8726621. [CrossRef] [PubMed]
41. Lu, X.X.; Zhang, H. Sentiment Analysis Method of Network Text Based on Improved AT-BiGRU Model. *Sci. Program.* **2021**, *2021*, 6669664. [CrossRef]
42. Miao, Y.; Ji, Y.; Peng, E. Application of CNN-BiGRU Model in Chinese short text sentiment analysis. In Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, Sanya, China, 20–22 December 2019; pp. 510–514.