IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 5 December 2023, accepted 17 December 2023, date of publication 5 January 2024, date of current version 20 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3350190

RESEARCH ARTICLE

A Parallel Two-Channel Emotion Classification Method for Chinese Text

LIU NA, TAO CAO, SHUCHEN BAI, AND DANQING LI
School of Information Science and Engineering, Dalian Polytechnic University, Dalian 116034, China

Corresponding author: Liu Na (liuna@dlpu.edu.cn)

This work was supported in part by the Liaoning Provincial Department of Education General Project under Grant JYTMS20230427.

ABSTRACT The complex structure of Chinese, including the dual-granularity feature of words and phrases, poses unique challenges for sentiment analysis, which is significantly different from the alphabetic word formation mechanism of English text. To capture Chinese semantics more accurately, an advanced parallel channel sentiment classification strategy is designed. In this study, the advanced pre-training model ERNIE with Word2vec is first adopted to enrich the embedding of Chinese text at word granularity. Following that, fine-grained features at the word level are extracted by a multi-window convolutional strategy, and the word vector sequences are deeply learnt using BiGRU network to ensure the comprehensive capture of contextual information in both directions. To further optimise the model, an Attention mechanism is introduced to ensure effective delivery of information and improve computational efficiency. After this series of innovative designs, the microblog comment dataset is used as an experimental case, and the hyperparameters are adjusted to determine the optimal parameters. Six comparison models are selected to verify the effectiveness of MsCBA on three datasets. The classification accuracies of the proposed model on the three datasets are 93.64%, 90.00% and 92.61%, respectively, which are better than the comparison models. This study provides an efficient and innovative approach for Chinese sentiment analysis, which sheds new light on the field of NLP.

INDEX TERMS Attention model, Chinese text classification, deep neural networks, dual-channel model, Ernie.

I. INTRODUCTION

With the continuous development of social networking media, social media platforms such as Weibo, WeChat, and Twitter have greatly influenced people's lifestyles [1], and more and more users do not only browse content, but also like to express their opinions on social media. The huge amount of data generated has attracted the attention of many researchers both at home and abroad, and in the face of such a large amount of information rich in emotional content, there is an urgent need for an analysis method, which leads to an important branch of natural language processing - text sentiment analysis [2]. Research on sentiment analysis can be traced back to the beginning of the 21st century, and it is now being extended to application areas, and has become a hot spot in academia for NLP [3], machine learning [4], [5], data analysis, and other fields. With the continuous progress of sentiment analysis research in recent years, accompanied by some new problems, the content posted on social media is gradually overloaded with medium-length texts to short texts, and the text content is characterized by short and flexible language, complex emotional expressions, many colloquial words on the Internet, and fast update, etc., which bring about problems for traditional sentiment analysis techniques (sentiment assessment using sentiment lexicons [6], sentiment evaluation utilizing conventional machine learning methods [7]) have brought a new challenge. On the other hand, deep learning methods represented by artificial neural networks have been gradually applied to the task of sentiment analysis with greater success. Currently, text sentiment analysis is widely used in e-commerce, opinion analysis, stock prediction, recommendation systems, etc. [8],

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

[9]. For communities and businesses, this in-depth sentiment analysis can more accurately capture and understand the feedback, emotions and needs of Chinese-speaking users. In the business world, this accuracy can help brands adjust their marketing strategies in a more targeted manner and improve the effectiveness of their adverts and products. On digital platforms and social networks, precise sentiment assessment can refine content suggestions and boost user interaction and satisfaction. And in the research field, the method provides scholars with a more efficient and comprehensive tool to delve into the complexity of Chinese sentiment. In conclusion, Chinese text sentiment analysis has its unique value and benefits in a variety of fields, including business, social and academic research. Therefore, sentiment analysis of Chinese texts in social media has important research significance and application value.

In Chinese text sentiment analysis tasks, the mostly used approach is to select word granularity units as text features for feature representation. Two training models (CBOW and Skip-gram) by Word2Vec are used to train low-dimensional Word Embedding [10], [11]. And most short text classification models using convolutional neural networks mostly use word embedding as input when extracting features and use convolutional neural networks with a single convolutional kernel to extract local features of text [12], without fully considering the effect of random initialization of network parameters, which leads to poor classification of the models. Considering that individual words also contain semantic information, this paper investigates more deeply the relationship between sentiment expression and the fusion of character and word features, and trains character/word vectors separately for the sentiment analysis task. Experiments show that the word fusion-based model can substantially increase the effectiveness of the Chinese sentiment analysis task compared with the traditional fusion models based on character vectors or word vectors.

II. RELATED WORK

Over recent times, as deep learning has evolved, significant advancements have been witnessed in areas like computer vision and speech processing. Therefore, more and more researchers have started to extend the application of deep learning networks to NLP field. Currently, deep learning-based sentiment analysis models [13] can be broadly categorized into standalone models and composite models [14], [15].

During the initial phases of sentiment analysis research, standalone models like Convolutional Neural Network (CNN) [16], Long Short-Term Memory Network (LSTM) [17], Bi-Directional Long Short-Term Memory Network (BiLSTM) [18], and Gated Recurrent Unit (GRU) were predominantly employed. These models are now frequently considered as benchmark models [19]. However, standalone models sentiment analysis models do not fully extract and exploit emotional features during feature extraction, and for this reason, a wide range of scholars are increasingly using sentiment analysis methods based on mixed models combined with attention mechanisms [20] to complement the shortcomings of single models. Li et al. [21] proposed a combined CNN and BiLSTM model, the using CNN model to extract local feature vectors of text, and the BiLSTM to extract contextually relevant global feature vectors, and finally fusing the feature vectors extracted by the two models, which fully complement each other and effectively avoid the problem of gradient disappearance or gradient dispersion of traditional recurrent neural networks. Han et al. [22] incorporated the Attention mechanism into their model, by merging both local and global textual semantics captured by CNN and BiLSTM. Building on deep learning and multi-channel foundations, their approach excelled in assessment metrics, advancing the field of text sentiment analysis. Cheng et al. [23] proposed to use a parallel model (MC-AttCNN-AttBiGRU) based on the combination of CNN and BiGRU based on the Attention mechanism. The model can focus on the key words in the text for sentiment polarity discrimination using the Attention mechanism and increases the ability to extract features for sentence content by using a CNN model for local feature extraction and a BiGRU model for contextual semantic extraction.

In terms of character and word granularity representation, the current research work fully demonstrates that fused character and word multiple granularities have great advantages over those based on character granularity or word granularity alone. Yin et al. [24] proposed a character and word fusion sentiment recognition method, each obtaining feature representations from word granularity aspect and word granularity aspect, obtaining deep hidden features of the microblog text by LSTM or BiLSTM, and finally fusing the features to obtain character and word fusion features for sentiment analysis, which is significantly better than the methods using character or word features alone in terms of performance, but does not fully take into account the correlation between features such as lexical and syntactic features and word and phrase features is not fully considered. Li et al. [25] utilized existing linguistic insights and sentiment resources to create distinct feature channels. This ensured that the model comprehensively grasped the emotional content of sentences prior to leveraging BiLSTM to extract valuable sentiment resource details. However, the model's performance in recognizing sentiment polarity in Chinese texts is limited, especially with cross-domain words, sentiment-laden words, and emerging internet terms. Liu et al. [26] proposed a parallel channel model that can compute short textual word and character vectors simultaneously and extract the features through TextCNN and DPCNN networks respectively, and then the extracted word feature vectors and phrase feature vectors are operated by fusion to complete the text classification task through the fully connected layer and Softmax layer. The model algorithm outperforms the current mainstream comparison models in all evaluation metrics. Zhang et al. [27] proposed a lexical feature-assisted discrimination of word meanings and improved text representation to address the

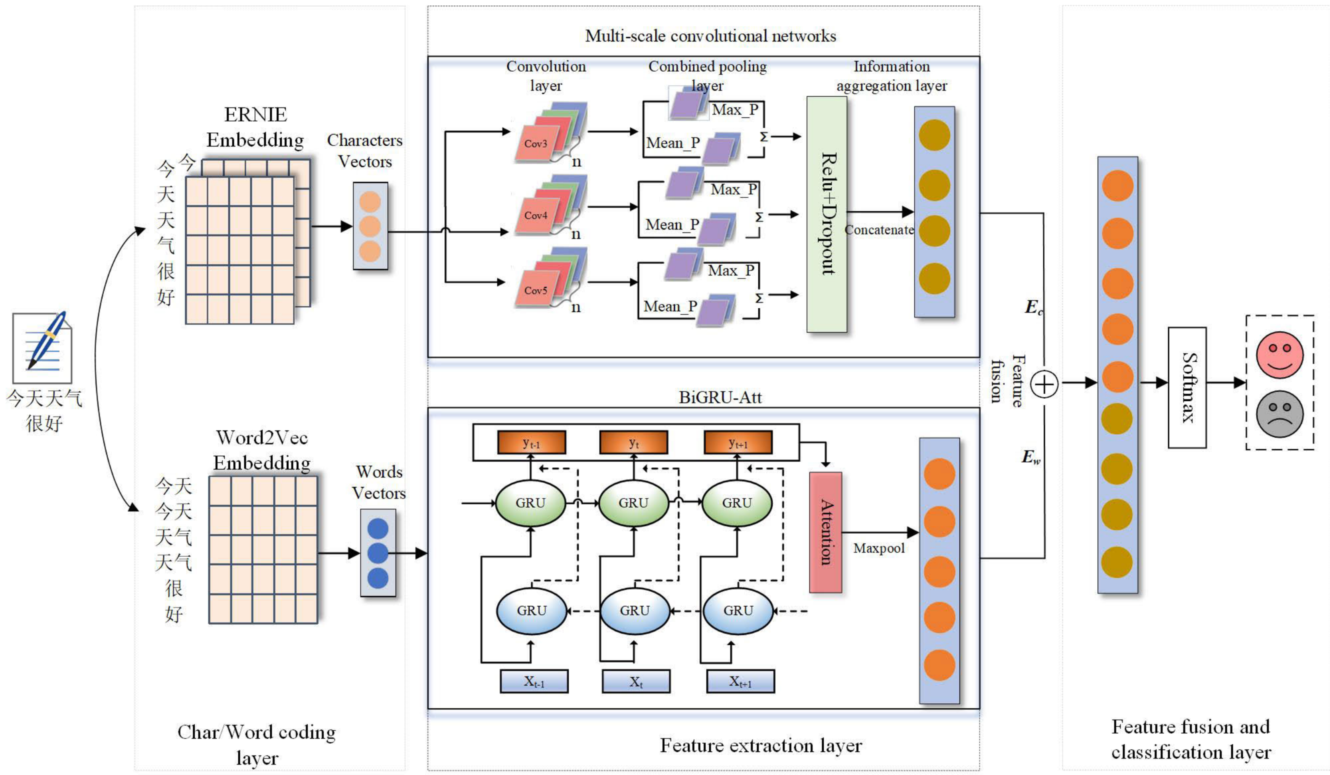


FIGURE 1. **Overall structure of the model.**

situation that semantic information of different contextual lexical items is not well distinguished, using the unique advantages of CNN and BiGRU to learn separately and construct a feature complementary fusion model CNN-BiGRU as a deeper feature extraction to obtain a deep combined semantics of text for increasing the recognition accuracy. However, the word vector generation approach is relatively homogeneous and the advantages of word-level granularity are not fully considered. Vyas et al. [28] An automated framework has been developed that employs the LSTM neural network as the preferred model. This system integrates dictionary-driven sentiment analysis with supervised machine learning methods to glean and classify feelings from Twitter. Amid the COVID-19 crisis, this setup proved invaluable in deciphering public emotions.

Although previous works have achieved substantial research results, most of these methods focus on CNN and BilSTM approaches, which have certain limitations and lead to difficulties in improving the accuracy rate. To address the above problems, this paper proposes a parallel dual-channel sentiment analysis model, which consists of two main components: a multi-convolutional kernel CNN module and a BiGRU-Attention module. The primary contributions of this study include:

1) Enriching text representation at the word granularity level using pre-trained ERNIE and Word2Vec models
2) On the one hand, the character granularity vectors are fed into the multi-window convolution to capture more fine-grained word-level feature information.
3) On the other hand, by substituting the traditional BiLSTM network with the BiGRU system, word vectors are produced by replicating each word based on its character count. This approach, when fed into the BiGRU system, effectively reduces the model's parameter size and optimally harnesses data from both forward and backward directions. The incorporation of the Attention mechanism further addresses information overload, enhancing both precision and efficiency.
4) Splicing two-channel high-level feature vectors for classification, and completing the classification task through the Softmax layer.
5) Model evaluation was performed on the Weibo_Senti dataset. The experiments proved that the model in this paper has better results than the single word vector model or the single phrase vector model in three evaluation indexes: precision rate, recall rate, F1 score, and accuracy value.

III. MODEL INTRODUCTION

The overall structure of the model proposed in this paper is shown in Figure 1, which consists of the following four parts.

1) Encoding layer: In this paper, two mapping functions, ERNIE Embedding and Word2Vec Embedding, are constructed for character-to-word vector and word-to-word vector mapping, respectively, in the encoding

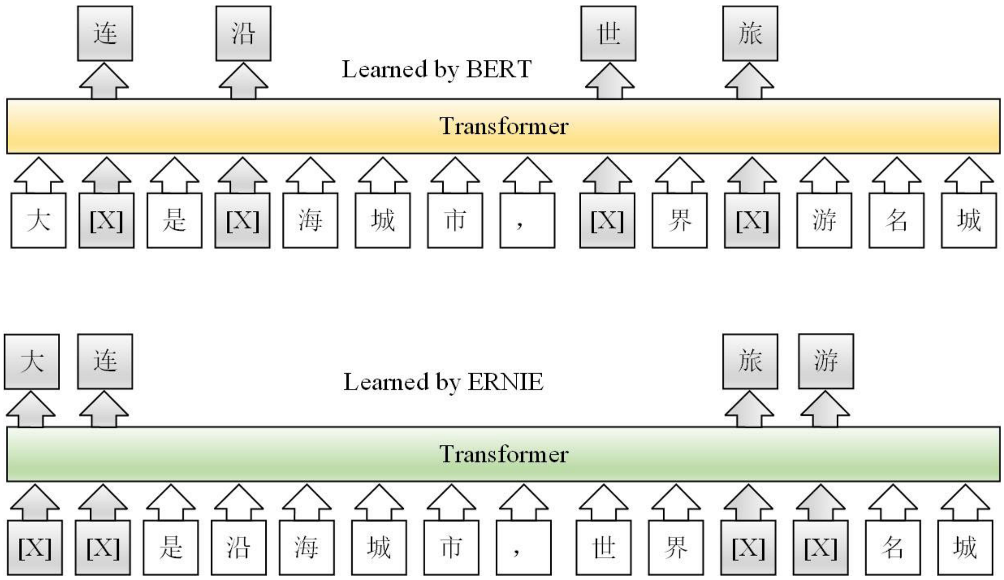


FIGURE 2. **Comparison of mask policies between BERT and ERNIE.**

process, and the mapped character vectors and word vectors are used as the inputs of the multi-window CNN and BiGRU models.

2) Multi-scale convolutional layer: Three one-dimensional convolutional kernel sets with sizes of 3, 4, and 5are used to jointly scan the word vectors and capture the multi-level N-Gram information, and the pooling layer adopts a combination of maximum pooling and average pooling, after which the multi-level feature information is stitched together as the character vectors of word granularity.
3) BiGRU-Attention layer: The architecture primarily integrates the attention mechanism with the BiGRU neural network. Word vectors representing the text are fed into the model via the input layer. Concurrently, BiGRU extracts features as the attention layer emphasizes semantic details, resulting in a comprehensive hidden layer representation of the text. The full-text implicit representation is obtained to enhance the robustness of word-level information, and the feature vectors are connected by maximum pooling and average pooling as word-level granularity.
4) Feature fusion and classification layer: fusing character-granularity feature vectors with word-granularity feature vectors, use a linear layer to learn the fusion weights of the two vectors, and finally input the fused vectors into Softmax for classification.

A. CODING LAYER

The encoding layer consists of 2 main parts, which are character-level vector representation based on BRNIE and word-level vector representation based on Word2Vec model.

1) CHARACTER LEVEL VECTOR REPRESENTATION

Knowledge enhanced Semantic representation model is a Chinese oriented pre-training model, similar to BERT model in general, and also based on Transformer encoder. However, the main difference between BERT and ERNIE is that the mask only covers a single character in BERT. If the relationship between words in phrases or entities is not fully considered, syntactic and semantic information cannot be well expressed. ERNIE added entity information such as syntax and grammar to the training task to enhance the language expression ability of the model. ERNIE has stronger versatility and expansibility, and the corpus of ERNIE training introduces multiple knowledge. Therefore, ERNIE is more suitable for sentiment analysis of Chinese than BERT. a comparison of different masking strategies of BERT and ERNIE is shown in Figure 2.

The ERNIE model uses a multi-headed attention mechanism for character vector computation, and several self-attention mechanisms are used in the process to obtain rich semantic information, and then the results of each calculation are stitched together to obtain the final result. The computational equation of attention Self-Attention is as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where, Q, K and V represent the three matrix representations of a char vector, and d_k represents the dimension of the vector. In this paper, when ERNIE was used for char vector training, the model parameters were set as 768 hidden layer size, 12 hidden layers, 12 heads of multiple attention mechanisms, and 18000 vocab number. The char vector obtained by training can be expressed as:

$$C = [c_1, c_2, \cdots, c_n]^T \tag{2}$$

2) WORD LEVEL VECTOR REPRESENTATION

The Word2Vec model for word representation includes two primary architectures: CBOW and Skip-gram. In this study, we employ the CBOW method for training word vectors. Before training, we utilize the Jieba tool for segmenting words. Figure3 illustrates this structure.

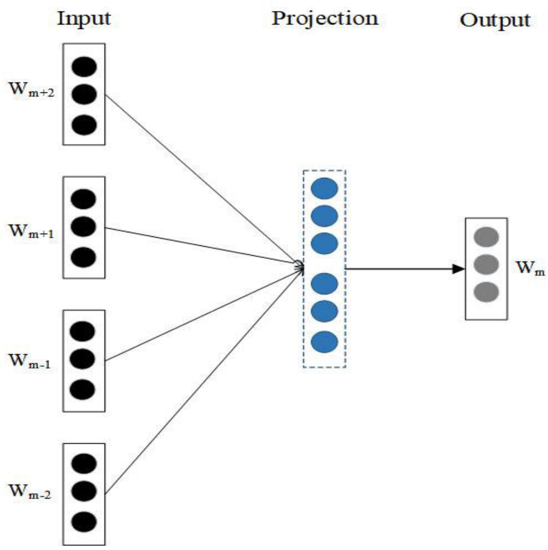


FIGURE 3. CBOW model.

In this paper, Word2Vec is used to train the word vector, the vector dimension is 300, and the trained word vector can be expressed as:

$$W = [w_1, w_2, \cdots, w_n]^T \tag{3}$$

B. FEATURE EXTRACTION LAYER

1) MULTI-SCALE CONVOLUTIONAL NETWORKS

The multi-scale convolutional network consists of a combination of multi-window convolution and maximum pooling and average pooling. The structure of the multi-window convolutional network is shown in Figure 4.

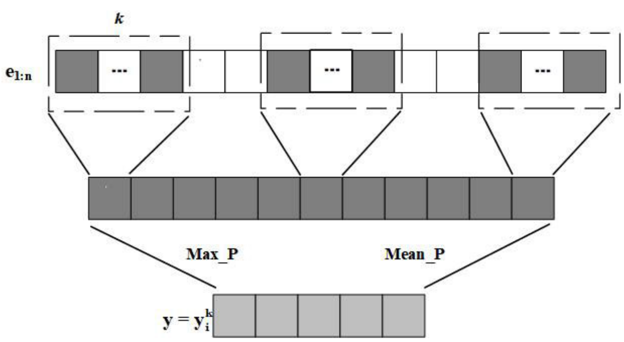


FIGURE 4. Structure of multi-scale convolutional neural network.

The main steps in the figure above are the convolution layer and the pooling layer. First, use the window size $k_i(i = 1, 2, 3, \ldots, n)$. The convolution kernel captures text local information.

$$y_i^k = e_{i:i+k-1} \cdot W + b \tag{4}$$

Then, the combination of maximum pooling and average pooling is used to reduce dimensions, capture the most significant features to a greater extent, and enhance the learning of semantic features around the upper layer to obtain feature vectors. The calculation process is shown in equations (5)∼(7).

$$y_{max}^k = max\_P(\{y_1^k, y_2^k, \ldots.., y_{n-k+1}^k\}) \tag{5}$$

$$y_{mean}^k = mean\_P(\{y_1^k, y_2^k, \ldots.., y_{n-k+1}^k\}) \tag{6}$$

$$y^k = y_{max}^k + y_{mean}^k \tag{7}$$

Different sizes of convolution kernel Windows will lead to different local information information, aiming to achieve multi-level semantic vectors. Therefore, in this paper, three sets of convolution kernel joint filtering (wherein $k = [k_1 = 3, k_2 = 4, k_3 = 5]$) is used to learn different N-Gram information. Finally, the features with different N-Gram information are aggregated to obtain multi-level character level latent semantic vector E_c.

$$E_c = [y^{k_1}, y^{k_2}, y^{k_3}] \tag{8}$$

where, W is the weight matrix, b is the bias vector, feature vector y^k, y is the multi-level character level latent semantic vector, P_{max} represents the result of maximum pooling, P_{mean} is the result of average pooling, and y^k is the final pooling result after the fusion of the two parts.

2) BIGRU-ATTENTION LAYER

The BiGRU network consists of an input layer, bidirectional hidden layers (both forward and backward), and an output layer. Data is fed into the input layer and at each time step t, it's simultaneously directed to both the forward and backward hidden layers. This means that the input data, x is processed by two GRU networks moving in opposite directions concurrently. The sequence y, produced by the output layer, is influenced by both GRUs. Figure 5 illustrates the GRU's architectural layout.

Given x_t as the input at time t, the GRU network's computation can be described as:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{9}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{10}$$

$$\widetilde{h_t} = tanh(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{11}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \widetilde{h_t} \tag{12}$$

where, z_t and r_t represent the update door and the reset door; $\widetilde{h_t}$ represents the state of candidate hidden layer state; h_{t-1} and h_t represent the hidden layer state at time t-1 and t respectively. W and U are weights; b is offset; σ stands for Sigmoid function.

The state of GRU is one-way transmission from front to back, thus ignoring the role of the following words on the full text; Compared with GRU, the output of BiGRU depends on the common influence of the forward and backward states, which solves the problems existing in GRU and improves the

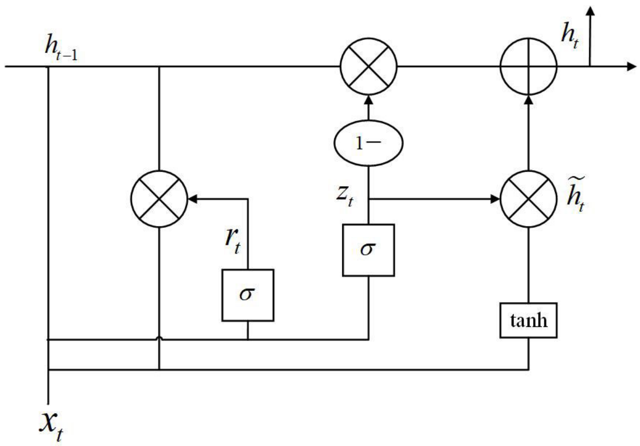


FIGURE 5. GRU network structure.

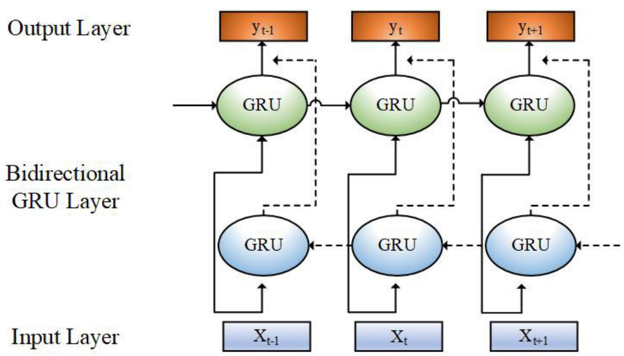


FIGURE 6. Model structure of BiGRU.

output results. The model structure of BiGRU is shown in the figure 6.

The mathematical expression of BiGRU network structure is as follows:

$$\overrightarrow{h_t} = GRU(x_t, \overrightarrow{h}_{t-1}) \tag{13}$$

$$\overleftarrow{h_t} = GRU(x_t, \overleftarrow{h}_{t-1}) \tag{14}$$

$$h_t = f(W_{\overrightarrow{h}_t} \overrightarrow{h_t} + W_{\overleftarrow{h_t}} \overleftarrow{h_t} + b_t) \tag{15}$$

where: At time t, $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ correspond to the states of the forward and backward hidden layers. The weights of these states are denoted by $W_{\overrightarrow{h}_t}$ and $W_{\overleftarrow{h_t}}$ respectively, while b_t signifies the bias for the hidden layer state at that specific time.

The attention mechanism assigns weight parameters according to the importance of information. For sentences, there are differences in importance between words. The purpose of introducing attention mechanism in this paper is to extract semantic information of important words in sentences. The internal structure of attention mechanism is mainly constructed by Encoder module and Decoder module. Encoder is an encoder, which transforms input information to obtain semantic vector. Decoder is a decoder, which also gets output information after transformation. The calculation process of attention value is as follows: Firstly, the attention distribution of input information is calculated to obtain the attention score function. Then, the attention score function is numerically converted through Softmax, and the weighted summation is performed according to the weight coefficient. The structure of the attention mechanism model is shown in Figure 7.

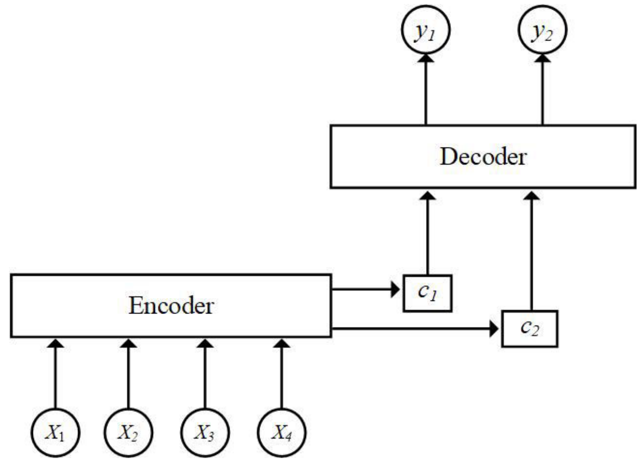


FIGURE 7. Structure of attention mechanism model.

Let the input of the neural network be $H=[h_1, h_2, \ldots, h_n]$, where H is the number of input information. h_m represents the input information vector and h_t represents the query vector used to search for key information. Normalization is performed by Softmax function to convert the original calculated score into a probability distribution with the sum of ownership weights equal to 1, highlighting the weight of important elements. The formula is as follows:

$$\alpha_m = \frac{exp(s(h_m, h_t))}{\sum_{j=1}^{N} exp(s(h_m, h_t))} \tag{16}$$

where, α_m represents the probability of obtaining the m-th input information, and the probability vector composed of all α_m is the attention distribution.

The function of attention function $s(h_m, h_t)$ is to calculate the similarity between two vectors. In this paper, additive model is used. Finally, the probability of all input information is summarized and summed in a weighted average way to obtain the value of attention V. The formula is as follows:

$$s(h_m, h_t) = v^T tanh(Wh_m + Uh_t) \tag{17}$$

$$V = \sum_{m=1}^{N} a_m h_m \tag{18}$$

where, W, U and V are trainable parameters within the neural network, while d denotes the dimensionality of the input data.

C. FEATURE FUSION AND CLASSIFICATION LAYER

1) FEATURE FUSION

After feature mapping and multi-window convolution network at input layer and feature extraction and operation at BiGRU layer, two different feature vectors, character-level feature vector E_c and word-level feature vector E_w, are

Algorithm 1 Emotion Classification of Chinese Microblog Text Based on BiGRU-Attention Module

Input: Chinese review text data set
Output: Predicted result polarity
1. Preprocess the original statistical data to obtain a new input sequence x_t.
2. Initialize network weight $W_z, r, h, U_z, r, h,$bias $b_z, r, h.$
3. Input the vector x_t into BiGRU and calculate the status value of the current hidden layer of one-way GRU according to equations (9)- (12).
4. Combine the forward and reverse stacked one-way GRUs networks to obtain the output
5. Calculate the weight αm corresponding to each output vector according to equations (16) and (17)
6. The input sequence information is summarized in a weighted average way, and the attention value V is calculated according to Equation (18).
7. Input the obtained word-level feature vector E to the output layer after maximum pooling.

Algorithm 2 Microblog Text Classification Based on MsCBA Model

Input: Chinese review text data set
Output: Predicted result polarity
1: Begin
2: **for** $i = 0$ **to** x-1
3: C = ERNIE.Embedding.from_pretrained(i)
4: W=Word2Vec.Embedding.from_pretrained(i)
5: E_{ci}=Multi-window CNN(C)
6: E_{wi}=BiGRU(W)-Attention
7: E=$ReLU$(torch.cat(E_{ci},E_{wi}) $* w + b$)
8: P=$Softmax$(E)
9: End

obtained through a linear layer to learn the weight coefficient of feature fusion, as shown in Equation (19).

$$E = ReLU((E_c \oplus E_w) * W + b) \tag{19}$$

where, W stands for the weight coefficient matrix, while b signifies the bias vector. The function $ReLU(\cdot)$ is utilized as the activation function.

2) CLASSIFICATION

Finally, the feature vector E is normalized by the layer and the input of Softmax classifier is directly classified.

$$P = Softmax(E) \tag{20}$$

where, E is the text classification feature representation of the final fusion char and word feature vector, and P is the probability corresponding to the prediction category.

D. THE TRAINING METHOD OF THE MODEL IN THIS PAPER

The model training in this paper takes microblog data as input; Firstly, char/word vector coding was carried out on the partitioned data set, and then feature vector extraction was carried out using two models. Finally, the integrated char and word feature training model is used to complete the emotion analysis task of microblog text.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. DATASETS

Three datasets are used in this experiment, namely, DATA1,DATA2 and DATA3. Comments from DATA1's microblogs are the primary focus of our experiment. To assess the universal applicability of the MsCBA model we've introduced, we've incorporated two testing datasets, DATA2 and DATA3, both aimed at categorizing public emotions across various sectors. The DATA 1 dataset (https://github.com/WAng91An/sentiment_classification_pytorch/tree/master/data) uses the weibo_senti microblogging assessment task dataset, totalling 119,988 entries. The data in the dataset are divided into two categories: positive and negative. The positive sentiment data (59,993) is labelled as 1, and the negative sentiment data (59,995) is labelled as 0.DATA2(https://raw.githubuse rcontent.com/SophonPlus/ChineseNlpCorpus/master/datasets/waimai_10k/waimai_10k.csv) encompasses feedback from takeout consumers, consisting of 4,000 commendations and 7,987 criticisms.DATA3(https://github. com/lizeyuking/textSentimentanalysis/blob/main/online_shopping_10_cats.csv) encompasses feedback from an e-commerce site, tallying 31,728 positive remarks and 31,045 critiques. In summary, these three datasets cover three different domains, namely social media, takeaway platforms and e-commerce platforms, which provide a wide range of data sources for this experiment to test and validate the performance and generalisation ability of the MsCBA model.

The three datasets were divided into training and test sets in the experimental phase with a ratio of 8:2. Table 1 shows some examples of positive and negative samples.

B. MODEL PARAMETER SETTING AND EVALUATION INDEX

The operating system of this experimental model is Windows 10, CPU is Intel (R)Core(TM) i5-7300HQCPU @2.5GHz, GPU is NVIDIA GeForceGTX 1650Ti 8G, programming environment is Pycharm. The programming language is python3.7. The model is mainly divided into two parts: CNN and BiGRU. A large number of super parameters are set and modified in the experimental design, which are adjusted according to the accuracy and loss rate after each iteration. After several iterations, the setting of the main experimental superparameters is shown in Table 2.

The impact of various hyperparameters on MsCBA's efficacy was assessed using the Weibo_Senti classification task. We focused on Dropout rates, Epochs and convolution kernel.

TABLE 1. Partial positive and negative samples.

Data sets	Comment data	Label	Categorisation
DATA 1	新年第一天家里就有喜事！/On the First Day of the New Year, there's a happy event in the family.	1	POS
	白折腾一圈图什么啊？/What's the point of tossing around for nothing?	0	NEG
DATA 2	送餐师傅很好，味道好极啦/The chef was very good and it tasted great!	1	POS
	不吃辣，都给的辣/We don't eat spicy food, we're given spicy food.	0	NEG
DATA 3	洗发水还可以，比在实体店里便宜/The shampoo is ok, cheaper than in a physical store	1	POS
	还说可以去头皮屑，都没有效果。/It also said it would remove dandruff, neither of which worked.	0	NEG

TABLE 2. Setting of superparameters of this experiment.

parameter	value	parameter	value
Epochs	5	optimizer	Adam
Dropout	0.2	Learning rate	5e-5
Batchsize	128	Loss function	Cross entropy
convolution kernel size	3,4,5	Number of convolutional output channels	256
Word2Vec word vector dimension	300	ERNIE char vector dimension	768

Starting with Dropout, this mechanism aims to counteract model overfitting. Upon analysis of its effects (as illustrated in Table 3 and Figure 8), a consistent performance improvement was observed when the Dropout rate was set below 0.2 on the Weibo_Senti dataset. However, setting the Dropout rate between 0.2 and 0.4 led to a decline in evaluation metrics, implying reduced model performance. Beyond this range, performance started to improve again, peaking at a Dropout of 0.8. Despite some higher evaluation metric values at elevated Dropout levels, it's vital to note that higher Dropout rates can mean losing critical information, affecting model precision. Therefore, a Dropout of 0.2 was deemed optimal.

Next, we turned our attention to the number of Epochs, which denotes the cycles of feedback in deep learning. As the neural network undergoes iterative training, the ideal number of Epochs becomes crucial. The relationship between Epochs and model performance is highlighted in Table 4 and Figure 9. Here, the model showcased its best performance at Epochs=5 for the Weibo_Senti dataset. As the Epoch count exceeded 2, the model's efficacy first dropped and then started to climb. However, during certain iterations, specifically when Epochs were set at 8, 9, or 10, the training halted prematurely. This suggests that increasing Epochs might transition the model from underfitting to overfitting. After careful observation, Epochs=5 emerged as the most fitting choice for our proposed model.

TABLE 3. Select dropout.

Dropout	Precision	Recall	F1 score
0.1	0.91301	0.90105	0.90699
0.2	**0.91372**	0.90754	0.91062
0.3	0.90293	0.90105	0.90199
0.4	0.89498	0.90220	0.89858
0.5	0.89892	0.90332	0.90112
0.6	0.90529	0.90218	0.90373
0.7	0.91037	0.90308	0.90671
0.8	0.91204	**0.90989**	**0.91096**
0.9	0.91011	0.90054	0.90530

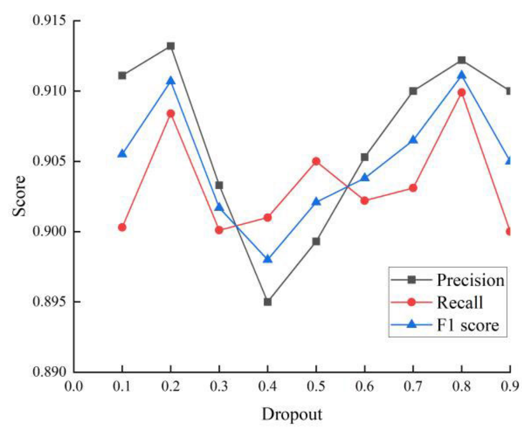


FIGURE 8. Comparison of Dropout parameter Settings.

Finally, when constructing a multi-scale convolutional neural network (MsCNN) for microblog sentiment analysis, the window size setting of the convolutional kernel is a key parameter, which directly affects the model's feature extraction capability and the final prediction performance. In order to explore the effects of different convolutional kernel window sizes on the model prediction results, we first fixed the number of convolutional kernels and tested the effects of

TABLE 4. Selection of epochs.

Epochs	Precision	Recall	F1 score
1	0.80598	0.79827	0.80211
2	0.89521	0.88321	0.88917
3	0.87545	0.86437	0.86988
4	0.85439	0.86542	0.85987
5	0.91311	0.90871	0.91091
6	0.89288	0.90099	0.89692
7	0.91079	0.89594	0.90330
8	0.90041	0.88945	0.89490
9	**0.91340**	0.91060	**0.91200**
10	0.91244	**0.91127**	0.91186

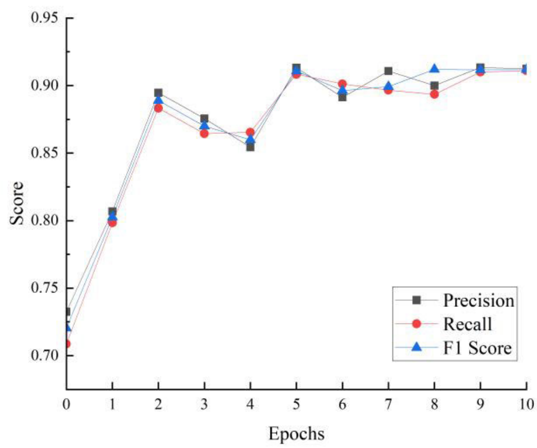


FIGURE 9. Comparison of Epochs parameter Settings.

window sizes from 1 to 10, respectively. In addition, we also evaluate the impact of combining convolutional kernels with different window sizes on the prediction results in search of higher prediction accuracy.

While it is a common practice in natural language processing tasks to use different sized convolutional kernels to capture multi-scale features of text, it is equally important to explore the single effect of the same convolutional kernel window size. Such an experimental setup can help researchers understand how the window size itself affects the model's learning ability and ultimately its prediction performance without introducing scale variations. This is especially crucial in our study, as the limited length and complexity of microblog texts may not require overly complex multi-scale feature extraction mechanisms.

By experimenting with Figure 10(a), an attempt was made to identify the optimal window size for a single scale to determine which window size would be most effective in capturing key features of sentiment expression in the absence of multi-scale complexity. This experiment provides an important baseline, showing that even at a single scale, window size adjustments can significantly alter model performance. When the window size was set to 5, the model reached its peak prediction accuracy, a finding that suggests that appropriate context width remains critical even in the absence of multi-scale features. In addition, this experiment provides a comparative context for subsequent experiments with more complex convolutional kernel combinations. Understanding how a single window size affects performance can help better explain the effects of different combinations of convolutional kernel window sizes observed in Figure 10(b). Without this comparison, it may not be possible to accurately judge the actual benefits of multi-scale feature extraction. Thus, even though the single window size experiment appears on the surface to be less straightforward than the multiscale feature extraction strategy, it still provides valuable insights that enable a more comprehensive understanding of the impact of convolutional kernel scale on sentiment classification performance.

As shown in Figure 10(b) when comparing the effects of different combinations of convolutional kernel window sizes, it is found that the combination (3,4,5) provides the best prediction accuracy. This optimal combination is effective because of its excellent performance in multi-scale feature extraction, which captures the short and concise sentiment expressions in the microblog text and fuses them into meaningful sentiment predictions. This combination avoids over-integration or loss of information and ensures that the model captures key local patterns while retaining sensitivity to global semantics. This is particularly important when dealing with microblogging datasets in non-standard and colloquial languages, which contain a large number of informal exchanges that are typically shorter and emotionally charged. Therefore, the combination of (3,4,5) demonstrates the effectiveness of multi-window sizes in capturing features, but also demonstrates that key sentiment signals in microblog texts can be effectively extracted and integrated through a carefully selected combination of window sizes, thus improving the accuracy of sentiment classification.

For binary sentiment analysis tasks, various evaluation measures are employed to gauge the efficacy of different classifiers. Commonly adopted metrics in this domain encompass accuracy, precision, recall, and F1 scores [29]. Further, both macro and weighted averages for precision, recall, and F1 have been considered [30], [31], [32]. Sentiment can be bifurcated into positive and negative. A sentiment that arises from an uptick in positivity or a downturn in negativity is classified as positive. Conversely, a sentiment stemming from a decline in positive factors or an uptick in negative ones is deemed negative. As depicted in Table 5, the confusion matrix serves as a fundamental method to assess accuracy.

In this matrix:T_P represents instances where the model correctly predicts a genuinely positive sample.F_N signifies instances where the model inaccurately labels a truly positive sample as negative.F_P denotes cases where the model mistakenly labels an actual negative sample as positive.T_N stands for instances where the model correctly predicts a genuinely negative sample.

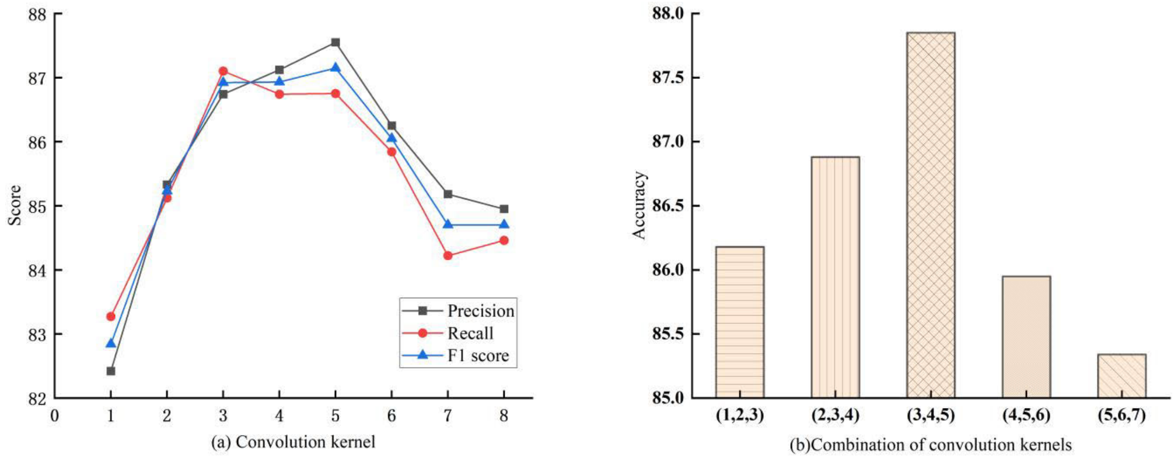


FIGURE 10. Influence of convolutional check network.

TABLE 5. Confusion matrix.

Actual classification	Prediction classification	
	Positive sentiment	Negative sentiment
Positive sentiment	T_P	F_N
Negative sentiment	F_P	T_N

The accuracy Acc, precision P_{pos},recall R_{pos} and F1 $F1_{pos}$ can be obtained by Eqs.(21)~(24).

$$P_{Pos} = \frac{T_p}{T_P + F_P} \tag{21}$$

$$R_{Pos} = \frac{T_P}{T_P + F_N} \tag{22}$$

$$F1_{Pos} = \frac{2 \times P_{Pos} \times R_{Pos}}{P_{Pos} + R_{Pos}} \tag{23}$$

$$Acc = \frac{T_p + T_N}{T_P + F_P + T_N + F_N} \tag{24}$$

The formulas for calculating the macro-average values of precision, recall, and F1 are provided in Equations (25) ~(27)

$$P_{Macro} = \frac{1}{2}(P_{pos} + P_{neg}) \tag{25}$$

$$F1_{Macro} = \frac{1}{2}(F1_{pos} + F1_{neg}) \tag{26}$$

$$R_{Macro} = \frac{1}{2}(R_{pos} + R_{neg}) \tag{27}$$

Given the imbalanced nature of most experimental datasets, we've opted for the weighted averages of precision, recall, and F1 as our evaluation metrics. These are computed as illustrated in Equations (28)~ (30).

$$P_{Weighted} = \omega_{pos}P_{pos} + (1 - \omega_{pos})P_{neg} \tag{28}$$

$$R_{Weighted} = \omega_{pos}R_{pos} + (1 - \omega_{pos})R_{neg} \tag{29}$$

$$F1_{Weighted} = \omega_{pos}F1_{pos} + (1 - \omega_{pos})F1_{neg} \tag{30}$$

where, the weights ω_{pos} and ω_{neg} are the proportions of positive comment samples and negative comment samples in the sample population,respectively.

C. COMPARISON EXPERIMENTAL MODELS

To ascertain the efficacy of the model introduced in this study, we've chosen a few models for juxtaposition, encompassing:

Word2Vec+MsCNN: Using the Word2Vec word vector as the embedding vector, the multi-scale CNN network extracts local feature information through convolution and pooling

Word2Vec+BiGRU: Using the Word2Vec word vector as the embedding vector, the bidirectional BiGRU network is used to obtain information on word meanings from both the preceding and following context.

BERT+Softmax: The first token position (CLS position) of the output of the last layer of Bert is treated as the representation of the sentence, which is then classified by the full connection layer.

ERNIE+Softmax: The first digit [CLS] after ERNIE coding is used as the classification feature and is directly connected to Softmax for classification.

ERNIE_MsCNN+Word2Vec_BiGRU(MsCB): using ERNIE as the word vector, the multi-scale CNN network extracts the local feature information through convolution and pooling, and the Word2Vec word vector is used as the embedding vector, and the bi-directional BiGRU network is used to obtain the upper and lower lexical information, which enters into the output layer in parallel, and then the feature vectors of the two channels are spliced together as the final text classification features.

(ERNIE⊕Word2Vec)-CNN-BiAGRU: ERNIE is used as the word vector, Word2vec word vector is used as the embedding vector, the word vectors are fused into the input, local feature information is extracted by convolution and pooling through multi-scale CNN network, then a bidirectional BiGRU network is used to obtain the upper and lower lexical information, and combined with the focusing feature of the attention mechanism of the attention mechanism, and then it enters into the output layer in parallel. MsCBA.

MsCBA: ERNIE is used as word vector, multi-scale CNN network extracts local feature information through convolution and pooling, Word2vec word vector is used as embedding

TABLE 6. Values of *P, R,F1* and *Acc* of each model.

Encoding mode	Model	Index	Precision	Recall	F1 Score	Accuracy
Word2Vec	Word2vec-MsCNN	Pos	0.90281	0.91282	0.90779	90.12%
		Neg	0.91325	0.90171	0.90744	
		Macro avg	0.90803	0.90727	0.90762	
		Weighted avg	0.90803	0.90727	0.90762	
	Word2vec-BiGRU	Pos	0.89256	0.88124	0.88686	85.44%
		Neg	0.82993	0.81340	0.82158	
		Macro avg	0.86125	0.84732	0.85422	
		Weighted avg	0.86125	0.84732	0.85422	
ERNIE/BERT	BERT-Softmax	Pos	0.92331	0.92528	0.92429	91.50%
		Neg	0.90659	0.90435	0.90547	
		Macro avg	0.91495	0.91482	0.91488	
		Weighted avg	0.91495	0.91482	0.91488	
	ERNIE-Softmax	Pos	0.93228	0.92769	0.92998	92.05%
		Neg	0.91172	0.91228	0.91200	
		Macro avg	0.92200	0.91999	0.92099	
		Weighted avg	0.92200	0.91999	0.92099	
Word2Vec+ERNIE	ERNIE_MsCNN+Word2vec_BiGRU	Pos	0.94503	**0.94214**	0.94358	93.26%
		Neg	0.92221	0.92102	0.92162	
		Macro avg	0.93362	0.93158	0.93260	
		Weighted avg	0.93362	0.93158	0.93260	
	(ERNIE ⊕ Word2vec)-CNN-BiAGRU	Pos	0.92318	0.93224	0.92769	92.71%
		Neg	0.92371	**0.93101**	0.92735	
		Macro avg	0.92345	0.93163	0.92752	
		Weighted avg	0.92345	0.93163	0.92752	
	MsCBA	Pos	**0.94732**	0.94211	**0.94471**	**93.64%**
		Neg	**0.93221**	0.92337	**0.92777**	
		Macro avg	**0.93977**	**0.93274**	**0.93624**	
		Weighted avg	**0.93977**	**0.93274**	**0.93624**	

vector, bi-directional BiGRU network is used to obtain upper and lower lexical meaning information and combined with attention mechanism to focus, and then it enters into the output layer serially, and then splices the feature vectors of the two channels to be used as the final text classification features.

D. ANALYSIS OF EXPERIMENTAL RESULTS

In the experiment, Tensorflow2.2 deep learning framework was used to build the neural network model. According to the relevant provisions and evaluation indicators of Weibo_Senti evaluation task, Weibo_Senti data set is used to evaluate the effects of the seven models proposed in this paper (+ represents the parallel channel, - represents the serial channel, and ⊕ represents the char and word vector fusion input). All models were tested three times, and the average value was taken as the final result. The values of *P*, *R*, *F1* and *Acc* of each model are shown in Table 6.

First, we explore two Word2Vec-based coding models for the task of sentiment analysis: the Word2Vec-MsCNN vs. the Word2Vec-BiGRU. We analyse the performance of these two models on positive (Pos) and negative (Neg) sentiment categories, with a special focus on precision, recall, F1 scores, and overall accuracy. Word2Vec-MsCNN achieves 90.12% in terms of overall accuracy. This excellent performance reveals its efficiency in extracting and processing sentiment information. Meanwhile, both the macro-averaged and weighted-averaged performance metrics are about 0.908, further proving its robustness and wide applicability. Meanwhile, Word2Vec-BiGRU exhibits an overall accuracy of 85.44%, which is also an impressive figure. Especially considering that BiGRU's structure is able to capture long-range dependencies in text, it provides a powerful tool for handling complex sentiment expressions. Its macro-averaged and weighted average performance index is around 0.854, highlighting its balance and consistency across different sentiment categories. Overall, both models demonstrate their unique strengths. Word2Vec-MsCNN marks its strong performance with its high accuracy, while Word2Vec-BiGRU ensures in-depth interpretation of sentiment information through its structural features. These findings provide valuable insights into the field of sentiment analysis and demonstrate the value of each of the two approaches.

Second, in the latest field of natural language processing, ERNIE and BERT have attracted a great deal of research interest as cutting-edge pre-trained models. In this study, we compare the performance of Softmax classifiers based on these two models in sentiment analysis tasks. The core objective of this work is to explore in detail how these two models perform in terms of precision, recall, F1 scores, and overall accuracy. There are some differences between the performance of BERT-Softmax and ERNIE-Softmax. For positive sentiment (Pos), BERT-Softmax has a precision of 0.92331, a recall of 0.92528, as well as an F1 score of 0.92429 for an overall accuracy of 91.50%. And for negative sentiment

TABLE 7. Takeaway reviews individual P,R,F1 and ACC values.

Encoding mode	Model	Index	Precision	Recall	F1 Score	Accuracy
Word2Vec	Word2vec-MsCNN	Pos	0.82776	0.78734	0.80704	83.58%
		Neg	0.84624	0.85432	0.85026	
		Macro avg	0.83700	0.82083	0.82865	
		Weighted avg	0.84007	0.83197	0.83584	
	Word2vec-BiGRU	Pos	0.80175	0.82079	0.81116	80.53%
		Neg	0.80334	0.80127	0.80230	
		Macro avg	0.80255	0.81103	0.80673	
		Weighted avg	0.80281	0.80778	0.80526	
ERNIE/BERT	BERT-Softmax	Pos	0.83122	0.85276	0.84185	84.15%
		Neg	0.84256	0.84435	0.84345	
		Macro avg	0.83689	0.84856	0.84265	
		Weighted avg	0.83878	0.84716	0.84292	
	ERNIE-Softmax	Pos	0.85741	0.82780	0.84235	86.73%
		Neg	0.87338	0.88641	0.87985	
		Macro avg	0.86540	0.85711	0.86110	
		Weighted avg	0.86805	0.86685	0.86734	
Word2Vec+ERNIE	ERNIE_MsCNN+Word2vec_BiGRU	Pos	0.88121	**0.89174**	0.88644	89.77%
		Neg	0.90116	**0.90561**	**0.90338**	
		Macro avg	0.89119	**0.89868**	**0.89491**	
		Weighted avg	0.89450	**0.90098**	**0.89773**	
	(ERNIE ⊕ Word2vec)-CNN-BiAGRU	Pos	0.89149	0.89003	**0.89076**	88.52%
		Neg	0.89251	0.87223	0.88225	
		Macro avg	0.89200	0.88128	0.88651	
		Weighted avg	0.89217	0.87817	0.88509	
	MsCBA	Pos	**0.90229**	0.84182	0.87101	**90.00%**
		Neg	**0.90324**	0.89871	0.90097	
		Macro avg	**0.90277**	0.87027	0.88599	
		Weighted avg	**0.90292**	0.87973	0.89097	

(Neg), it exhibits a precision of 0.90659, a recall of 0.90435, and an F1 score of 0.90547. Taken together, its macro and weighted average precision, recall, and F1 score are about 0.9149. In comparison, ERNIE-Softmax has a precision of 0.93228, a recall of 0.92769, and an F1 score of 0.92998 for positive sentiment classification, with an overall accuracy of 92.05%. For negative sentiment, the precision is 0.91172, the recall is 0.91228, as well as the F1 score is 0.91200. Overall, the macro-averaged and weighted mean of this model is about 0.92 in terms of precision, recall, and F1 score. Intuitively, compared to BERT-Softmax, ERNIE-Softmax has about 0.55% improvement in overall accuracy, and about 0.57% improvement in F1 scores for positive sentiment and 0.65% improvement in F1 scores for negative sentiment. These data reflect the slight advantage of ERNIE-Softmax over BERT-Softmax in sentiment analysis. These comparative results provide valuable insights for deeper exploration of the variability and optimisation directions of pre-trained models in real-world applications.

Finally we explored three models fusing ERNIE and Word2Vec techniques, the ERNIE_MsCNN+Word2Vec_BiGRU (or MsCB for short) model utilising ERNIE's word vectors combined with a multi-scale CNN network, and Word2vec's word vectors processed in parallel with a bi-directional BiGRU, to achieve a 93.26% accuracy. For positive evaluation, the F1 score is 0.94358, while for negative evaluation it is 0.92162.

The (ERNIE⊕Word2Vec)-MsCNN-BiAGRU model, on the other hand, uses a fronting strategy to fuse word vectors with word vectors and incorporates an attention mechanism. The overall accuracy of this model is 92.71%, which is slightly lower than the MsCB model, specifically by 0.55 percentage points. However, it performed well in terms of F1 scores for both positive and negative ratings, which were 0.92769 and 0.92735, respectively.

The MsCBA model, although similar in strategy to the MsCB model, is characterised by direct feature splicing after parallel processing. It achieves the highest accuracy of 93.64% on the test data, which is an improvement of 0.38 percentage points relative to the MsCB model and 0.93 percentage points relative to the (ERNIE⊕Word2vec)-CNN-BiAGRU model. The F1 score for positive evaluation is 0.94471, which is an improvement of 0.00113 over the MsCB model, while the F1 score for negative evaluation is 0.92777, which is still an excellent performance.

In summary, all three models show their unique advantages and efficient performance. However, in our tests, the MsCBA parallel model is particularly outstanding, improving both in overall accuracy and F1 score, highlighting its potential and value in text sentiment classification tasks.

Different dataset characteristics and their challenges need to be explicitly considered when exploring the broad adaptability of advanced text categorisation models such as MsCBA. In this study, two unique datasets, DATA2 and

TABLE 8. E-commerce reviews individual P,R,F1 and ACC values.

Encoding mode	Model	Index	Precision	Recall	F1 Score	Accuracy
Word2Vec	Word2vec-MsCNN	Pos	0.90221	0.89034	0.89624	89.37%
		Neg	0.88324	0.89908	0.89109	
		Macro avg	0.89273	0.89471	0.89367	
		Weighted avg	0.89283	0.89466	0.89369	
	Word2vec-BiGRU	Pos	0.88622	0.87114	0.87862	87.48%
		Neg	0.86832	0.87357	0.87094	
		Macro avg	0.87727	0.87236	0.87478	
		Weighted avg	0.87737	0.87234	0.87482	
ERNIE/BERT	BERT-Softmax	Pos	0.91883	0.90141	0.91004	90.97%
		Neg	0.90204	0.91692	0.90942	
		Macro avg	0.91044	0.90917	0.90973	
		Weighted avg	0.91053	0.90908	0.90973	
	ERNIE-Softmax	Pos	**0.93140**	0.91830	0.92480	92.09%
		Neg	0.91125	0.92280	0.91699	
		Macro avg	0.92133	0.92055	0.92090	
		Weighted avg	0.92143	0.92053	0.92094	
Word2Vec+ERNIE	ERNIE_MsCNN+Word2vec_BiGRU	Pos	0.92824	0.91447	0.92130	91.95%
		Neg	0.91012	0.92532	0.91766	
		Macro avg	0.91918	0.91990	0.91948	
		Weighted avg	0.91928	0.91984	0.91950	
	(ERNIE ⊕ Word2vec)-CNN-BiAGRU	Pos	0.92102	**0.92611**	0.92356	92.40%
		Neg	**0.92625**	0.92286	**0.92455**	
		Macro avg	0.92364	0.92449	0.92406	
		Weighted avg	0.92362	0.92450	0.92401	
	MsCBA	Pos	0.93024	0.92572	**0.92797**	**92.61%**
		Neg	0.92131	**0.92726**	0.92428	
		Macro avg	**0.92578**	**0.92649**	**0.92613**	
		Weighted avg	**0.92582**	**0.92648**	**0.92615**	

DATA3, are used to evaluate the generalisation performance of MsCBA models. The experimental results are shown in Tables 7 and 8.

DATA2, a takeaway review dataset, presents a data imbalance problem similar to a real-world scenario. There are more negative reviews than positive reviews, which may make the classification task more difficult as most machine learning models will tend to predict categories with more occurrences. However, MsCBA shows a distinct advantage in this scenario, with Precision, Recall, and F1 Score of 0.90229, 0.84182, and 0.87101, respectively, on the Positive sample (Pos), while these values are 0.90324, 0.89871, and 0.90097 on the Negative sample (Neg). particularly. striking is the overall accuracy of 90.00%. surpassing other coding patterns and models.

As for DATA3, an e-commerce review dataset, its main challenge is the long length of the text. Longer texts usually contain more information, but they can also introduce noise and redundancy, which complicates sentiment analysis. Despite the fact that long text tends to increase the complexity of parsing, MsCBA still achieves excellent performance on this dataset. On the positive samples, Precision, Recall and F1 Score were 0.93024, 0.92572 and 0.92797, respectively. for the negative samples, these values were 0.92131, 0.92726 and 0.92428. impressively, the overall accuracy reached 92.61% ahead of other methods.

In summary, the MsCBA model not only performs well on standard microblog comment datasets (e.g., DATA1), but also demonstrates its strong generalisation and adaptability in the face of unbalanced sample distributions (e.g., DATA2) and long texts (e.g., DATA3). This finding has far-reaching implications for the field of natural language processing, especially considering that real-world data are often messy, uneven and diverse. Therefore, the MsCBA model not only provides an effective tool for sentiment analysis in microblogging, takeout and e-commerce platforms, but also MsCBA lays a solid foundation for potential applications in a variety of real-world application scenarios, providing new directions and insights for future research and applications.

E. ABLATION EXPERIMENT AND ANALYSIS

ERNIE_BiGRU+Word2Vec_BiGRU: Input vectors take char granularity information and word granularity information, and then use BiGRU to extract features.

ERNIE_MsCNN+Word2Vec_MsCNN: Input vectors take char granularity information and word granularity information, and then use multi-scale convolutional networks to extract features.

MsCB: Based on the MsCBA model proposed in this paper, the attention mechanism in parallel channels is removed.

In order to see the performance differences between models more clearly, ablation experiments were conducted on different sentiment analysis models, and the results are shown in Table 9.

Firstly, the ERNIE_BiGRU+Word2Vec_BiGRU model takes word granularity and word granularity information

TABLE 9. Values of P, R and F1 in ablation experiments of each model.

Model	Precision(%)	Recall(%)	F1 Score(%)
ERNIE_BiGRU+Word2vec_BiGRU	89.85	90.14	90.00
ERNIE_MsCNN+Word2vec_MsCNN	90.66	90.48	90.57
MsCB	92.11	**92.43**	92.27
MsCBA	**92.24**	92.37	**92.31**

TABLE 10. Comparison results of downsampling experiments.

Data set	Model	Precision(%)	Recall(%)	F1 Score(%)
$\text{Weibo\_Senti}_{80\%}$	MsCB	90.45	90.64	90.55
	MsCBA	**91.33**	**91.15**	**91.24**
$\text{Weibo\_Senti}_{50\%}$	MsCB	88.35	89.47	88.91
	MsCBA	**89.48**	**90.39**	**89.93**
$\text{Weibo\_Senti}_{20\%}$	MsCB	84.54	85.35	84.94
	MsCBA	**86.75**	**87.44**	**87.09**

as input vectors and uses BiGRU for feature extraction. It achieves 89.85%, 90.14% and 90.00% in precision, recall and F1 score, respectively.

Secondly, the ERNIE_MsCNN+Word2Vec_MsCNN model is also based on word-granularity and word-granularity information and extracts features by multi-scale convolutional network. Compared with the previous model, the precision increased by 0.81 percentage points, the recall increased by 0.34 percentage points, and the F1 score increased by 0.57 percentage points to 90.57%. The reason for this is that in feature extraction, the CNN model pays more attention to the extraction of local phrase features, and cannot fully utilise the feature information at the same level, and lacks the feature representation of the sentence system, thus using only a single CNN will drastically reduce the feature learning ability of the model. In contrast, BiGRU network is able to learn serialised information, make full use of the features of the sentence system, and increase the connection with the CNN pooling layer. So using different feature extraction models in the feature extraction phase tends to give better performance.

The MsCB model is obtained by removing the attention mechanism in the parallel channel based on the MsCBA model proposed in this paper. The model achieves 92.11%,

92.43% and 92.27% results in precision, recall and F1 score, respectively, which are 1.45 percentage points, 0.95 percentage points and 1.70 percentage points better than the ERNIE_MsCNN+Word2Vec_MsCNN models, respectively.

Finally, the MsCBA model, which is the focus of our study, shows excellent performance in overall, achieving 92.24% precision, 92.37% recall and 92.31% F1 score. All these metrics are improved relative to the MsCB model.

In summary, it can be seen that both multi-scale convolutional networks and attention mechanisms have their unique value and impact in text classification tasks. And the MsCBA model, as a strategy that incorporates multiple techniques, demonstrates its strong potential in complex text classification scenarios.

F. COMPARATIVE EXPERIMENTAL ANALYSIS OF DOWNSAMPLING

In order to test the performance of the MsCBA model when the data set is small, the data randomly selected from the Weibo_Senti dataset are divided into 80%, 50% and 20% of the original dataset respectively as the new data sets Weibo_Senti80%, Weibo_Senti50% and Weibo_Senti20%. The total number of Chinese text data in the original Weibo_Senti dataset is 120,000, and the total number of Chinese text data in Weibo_Senti80% is 96,000, the total number of data in Weibo_Senti50% is 60,000, and the total number of data in Weibo_Senti20% is 24,000. The positive to negative ratio of the three data sets was 1:1. Since MsCBA and MsCB have better experimental results during the test, these two models are selected and downsampling is adopted for testing and comparison. The experimental results are shown in Table 10.

In order to explore the robustness and performance of the model on different sized datasets, we conducted a downsampling experiment on the Weibo_Senti dataset. In this experiment, by random sampling, we obtained three sub-datasets: $\text{Weibo\_Senti}_{80\%}$, $\text{Weibo\_Senti}_{50\%}$ and $\text{Weibo\_Senti}_{20\%}$. Considering that both MsCBA and MsCB models have shown good performance in previous experiments, we chose these two models for this downsampling comparison experiment.

From the results in Table 7, it can be observed that the performance of both models decreases as the size of the dataset decreases, which is in line with expectations since less data tends to imply less information, which may lead to overfitting or failure of the model to learn adequately. Nonetheless, the MsCBA model outperformed on all doensampled datasets.

Notably, on the Weibo_Senti$_{80\%}$ dataset, MsCBA enhances accuracy by 0.88%, recall by 0.51%, and F1 score by 0.69% when compared to MsCB.

On the Weibo_Senti$_{50\%}$ dataset, the MsCBA model improves precision, recall, and F1 score by 1.13 percentage points, 0.92 percentage points, and 1.02 percentage points, respectively, compared to the MsCB model.

On the Weibo_Senti$_{20\%}$ dataset, the most notable variation is observed. Here, MsCBA boosts precision by 2.21%, recall by 2.09%, and F1 score by 2.15% compared to MsCB, even when using just 20% of the original data.

In summary, despite the impact of downsampling on model performance, MsCBA demonstrates consistent strengths across all subsets. This strongly suggests its robustness and efficiency in the face of scenarios with sparse or insufficient data.

V. CONCLUSION AND FUTURE WORK

This study proposes a revolutionary parallel dual-channel sentiment analysis framework by thoroughly exploring the cutting-edge techniques of NLP sentiment analysis, especially when comparing the effectiveness of single models with hybrid models. Based on this, the pre-trained ERNIE and Word2Vec are further combined to incorporate fine-grained representations of words and phrases into the model to ensure richer and deeper semantic capture. Through the combination of multi-window convolution and BiGRU, not only local word-level features are successfully extracted, but also richer contextual information can be captured. By introducing the Attention mechanism, the model efficiently and accurately handles large-scale data, resulting in excellent performance in the evaluation of all three major datasets.

This work not only looks at the research on sentiment analysis from a new perspective, but also demonstrates the performance enhancement that comes from the convergence of multiple techniques. This not only sets a new benchmark for academia, but also provides practical and efficient solutions for industry. The emphasis on, and fusion of, word granularity further highlights the research value and uniqueness of Chinese NLP. All in all, this paper provides insightful methods and tools for researchers and practitioners of sentiment analysis.

However, given the growth of multimodal information (e.g., text, audio, and images), future work will consider integrating these data sources to capture sentiment more comprehensively. At the same time, enhancing the interpretability of models to help researchers and practitioners better understand the modelling decision-making process will be one direction. Extending the model to accommodate multilingual scenarios is also planned, given the nuances of emotion brought about by cross-cultural and linguistic differences, and understanding how to accurately capture sentiments with irony. In response to the problem of category imbalance in sentiment datasets, more attention will also be paid to optimising the model's performance on such data in subsequent research.

REFERENCES

[1] H. Chen, S. Jin, W. Lin, Z. Zhu, and Y. Liu, "Quantitative analysis on the communication of COVID-19 related social media rumors," *J. Comput. Res. Dev.*, vol. 58, no. 7, pp. 1366–1384, 2021.

[2] J. Liu and F. Gu, "Unbalanced text sentiment analysis of network public opinion based on BERT and BiLSTM hybrid method," *J. Intell.*, vol. 41, no. 4, pp. 104–110, 2022.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[4] J. Zhong, W. Liu, S. Wang, and H. Yang, "Review of methods and applications of text sentiment analysis," *Data Anal. Knowl. Discov.*, vol. 5, no. 6, pp. 1–13, 2021.

[5] T. Wang and W. Yang, "Review of text sentiment analysis methods," *Comput. Eng. Appl.*, vol. 57, pp. 11–24, 2021.

[6] J. Sun and Q. Lv, "ON sentiment analysis of Chinese microblogging based on lexicon and machine learning," *Comput. Appl. Softw.*, vol. 31, pp. 177–181, 2014.

[7] L. Wang, X. Mu, and H. Liu, "Using SVM method optimized by improved particle swarm optimization to analyze emotion of Chinese text," *Comput. Sci.*, vol. 47, no. 1, pp. 231–236, 2020.

[8] C. Deng, H. Yu, and G. Fan, "Integrating dynamic collaborative filtering and deep learning for recommendation," *Comput. Sci.*, vol. 46, pp. 28–34, 2019.

[9] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowl.-Based Syst.*, vol. 226, Aug. 2021, Art. no. 107134.

[10] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi, "Sentiment analysis based on improved pre-trained word embeddings," *Expert Syst. Appl.*, vol. 117, pp. 139–147, Mar. 2019.

[11] A. Khatua, A. Khatua, and E. Cambria, "A tale of two epidemics: Contextual Word2Vec for classifying Twitter streams during outbreaks," *Inf. Process. Manage.*, vol. 56, no. 1, pp. 247–257, Jan. 2019.

[12] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.

[13] A. Mabrouk, R. P. D. Redondo, and M. Kayed, "Deep learning-based sentiment classification: A comparative survey," *IEEE Access*, vol. 8, pp. 85616–85638, 2020.

[14] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.

[15] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 604–624, Feb. 2021.

[16] S. L. Ramaswamy and J. Chinnappan, "RecogNet-LSTM+CNN: A hybrid network with attention mechanism for aspect categorization and sentiment classification," *J. Intell. Inf. Syst.*, vol. 58, no. 2, pp. 379–404, Apr. 2022.

[17] H. Jelodar, Y. Wang, R. Orji, and S. Huang, "Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 10, pp. 2733–2742, Oct. 2020.

[18] W. Li, F. Qi, M. Tang, and Z. Yu, "Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification," *Neurocomputing*, vol. 387, pp. 63–77, Apr. 2020.

[19] L.-X. Luo, "Network text sentiment analysis method combining LDA text representation and GRU-CNN," *Pers. Ubiquitous Comput.*, vol. 23, nos. 3–4, pp. 405–412, Jul. 2019.

[20] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Gener. Comput. Syst.*, vol. 115, pp. 279–294, Feb. 2021.

[21] Y. Li and H. Dong, "Text sentiment analysis based on feature fusion of convolution neural network and bidirectional long short-term memory network," *J. Comput. Appl.*, vol. 38, no. 11, pp. 3075–3080, 2018.

[22] P. Han, W. Zhang, and Z. Zhang, "Sentiment analysis of Weibo posts on public health emergency with feature fusion and multi-channel," *Data Anal. Knowl. Discov.*, vol. 5, no. 11, pp. 68–79, 2021.

[23] Y. Cheng, L. Yao, G. Zhang, and T. Tang, "Text sentiment orientation analysis of multi-channels CNN and BiGRU based on attention mechanism," *J. Comput. Res. Dev.*, vol. 57, pp. 2583–2595, 2020.

[24] H. Yin, J. Xu, S. Li, and G. Zhou, "Emotion recognitionon microblog based on character and word feature," *Comput. Sci.*, vol. 45, pp. 105–109, 2018.

[25] W. Li and F. Qi, ''Sentiment analysis based on multi-channel bidirectional long short term memory network,'' *J. Chin. Inf. Process.*, vol. 33, pp. 119–128, 2019.

[26] S. Liu, G. Wang, J. Peng, and K. Li, ''Chinese short text classification algorithm based on hybrid features of characters and words,'' *Comput. Sci.*, vol. 49, pp. 282–287, 2022.

[27] X. Zhang, L. Liu, X. Dai, and L. Liu, ''CNN_BIGRU text classification model based on part of speech features,'' *Comput. Appl. Softw.*, vol. 38, no. 11, pp. 155–161, 2021.

[28] P. Vyas, M. Reisslein, B. P. Rimal, G. Vyas, G. P. Basyal, and P. Muzumdar, ''Automated classification of societal sentiments on Twitter with machine learning,'' *IEEE Trans. Technol. Soc.*, vol. 3, no. 2, pp. 100–110, Jun. 2022, doi: 10.1109/TTS.2021.3108963.

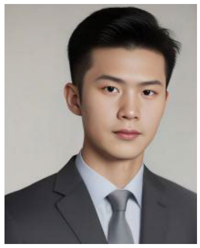
[29] D. M. Powers, ''Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation,'' *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[30] L. Liu, H. Shao, Z. Chen, X. Jiang, and M. Qian, ''RoBERTa-WWM-ext: A robustly optimized BERT pretraining approach,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3762–3772, 2020.

[31] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, ''BioBERT: A pre-trained biomedical language representation model for biomedical text mining,'' *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, Feb. 2020.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ''BERT: Pre-training of deep bidirectional transformers for language understanding,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186.

LIU NA was born in December 1978. She received the Ph.D. degree in engineering from Dalian Maritime University, in July 2012, majoring in computer application technology. Currently, she is the Director of the Teaching and Research Office of Network Engineering and the Secretary of the Computer Party Branch of the School of Information. She is also an Associate Professor and a Master Tutor. Her research interests include data mining and machine learning. She has published more than 20 academic articles in important academic journals at home and abroad, among which more than ten articles have been included by SCI and EI, and won one award of Dalian Natural Science Outstanding Academic Paper.



TAO CAO was born in Yongcheng, Henan. He is currently pursuing the master's degree in Dalian. His research interests include heavily in the realm of NLP, especially sentiment analysis. Through his research, he has understand that NLP isn't just a collection of techniques and algorithms; it's a powerful tool that unveils the intricacies of human psychology and emotions. He eagerly looks forward to engaging with peers from the broader academic and industrial communities, exploring the future trajectories of artificial intelligence and NLP. His research interests include wider audience and strive to contribute meaningfully to the advancements in this field.

SHUCHEN BAI was born in Zhumadian, Henan, China, in March 2000. She received the bachelor's degree in the Internet of Things Engineering from the Henan College of Engineering. She is currently pursuing the master's degree with the Dalian University of Technology, specializing in computer technology. Her research interests include machine learning and natural language processing.

DANQING LI was born in Bozhou, Anhui. She received the bachelor's and master's degrees from the Dalian Polytechnic University, focusing on natural language processing and recommender systems.