



Article

Chinese Short-Text Sentiment Prediction: A Study of Progressive Prediction Techniques and Attentional Fine-Tuning

Jinlong Wang , Dong Cui and Qiang Zhang *

School of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China

* Correspondence: zhangqiang@hebeu.edu.cn

Abstract: With sentiment prediction technology, businesses can quickly look at user reviews to find ways to improve their products and services. We present the BertBilstm Multiple Emotion Judgment (BBMEJ) model for small-sample emotion prediction tasks to solve the difficulties of short emotion identification datasets and the high dataset annotation costs encountered by small businesses. The BBMEJ model is suitable for many datasets. When an insufficient quantity of relevant datasets prevents the model from achieving the desired training results, the prediction accuracy of the model can be enhanced by fine-tuning it with additional datasets prior to training. Due to the number of parameters in the Bert model, fine-tuning requires a lot of data, which drives up the cost of fine-tuning. We present the Bert Tail Attention Fine-Tuning (BTAFT) method to make fine-tuning work better. Our experimental findings demonstrate that the BTAFT fine-tuning approach performs better in terms of the prediction effect than fine-tuning all parameters. Our model obtains a small sample prediction accuracy of 0.636, which is better than the ideal baseline of 0.064. The Macro-F1 (F1) evaluation metrics significantly exceed other models.

Keywords: emotion recognition; small-sample recognition; deep learning; interpretability; bert fine-tuning



Citation: Wang, J.; Cui, D.; Zhang, Q. Chinese Short-Text Sentiment Prediction: A Study of Progressive Prediction Techniques and Attentional Fine-Tuning. *Future Internet* **2023**, *15*, 158. <https://doi.org/10.3390/fi15050158>

Academic Editor: Luis Javier Garcia Villalba

Received: 7 March 2023

Revised: 19 April 2023

Accepted: 21 April 2023

Published: 23 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the growth of the Internet and the popularity of smartphones, an increasing number of individuals prefer to share their ideas on social networking sites [1]. A great deal of textual material is kept on the Internet, for instance, in online shopping platforms, such as Taobao and Jingdong, where people can express comments and opinions. The vast volume of social text information available on the Internet has a high commercial value. Mining this information can be a useful way for businesses to understand public attitudes toward their products, for celebrities to understand the emotions generated by their fans in response to their participation in public events, and for public institutions to understand Internet-based public opinion [2].

Sina Weibo, which means “micro blog” in Chinese, is a Chinese social networking platform similar to Twitter. Weibo are similar to Twitter tweets in that users can express their opinions by posting them. Super topics are a tag system for Weibo content. By adding a relevant super talk tag to a user’s Weibo, the Weibo can be seen by more users who care about the topic; these tags also make it easier for users to quickly filter relevant content. The information in Weibo is also an important source of data for the social and natural sciences. Sina Weibo, a popular online social media platform in China, has become popular among Chinese netizens, with hundreds of millions of users using the Sina Weibo platform to keep up with the COVID-19 epidemic during the outbreak [3].

Users expect that complaints about products will be responded to in a timely manner. However, companies and users have a one-to-many relationship. Companies often receive huge amounts of information and are not able to see complaints from users in a timely

manner. With automated sentiment analysis, it is possible to quickly look for negative comments in the messages.

For companies, getting feedback from users can be an effective way to improve products and understand whether the design and service of the product is satisfying. For example, for a new product launch, social information about the product is collected on the Internet and, through sentiment analysis, labels such as surprised, angry, and fear are classified. Users are mostly satisfied with texts labelled 'like'; those labelled 'surprised' highlight the features of the product; those labelled 'angry' and 'fear' often express the product's shortcomings. With this information, product sales strategies can be adjusted to provide ideas for the design of the next generation of products and to improve user satisfaction, thereby increasing the sales rate of the product.

There are many small Internet businesses in China, which operate local businesses such as dating, brokerage, and renting [4]. These small local companies have few users, more local visibility than national visibility, and user communication intermingled with local languages as their major features. It is crucial for these businesses to remain up to date on consumer requirements and feedback. Although the field of natural language processing has accomplished a great deal with regard to text sentiment, these tiny enterprises do not have a large amount of data, and the human categorization of datasets is a substantial initial investment. If existing large datasets are employed, they run into the issue of local dialect features and the many types of sentiment analysis required by businesses. Predicting emotion based on a small sample size can assist small firms in resolving associated issues.

Early machine learning techniques for sentiment analysis include support vector machines (SVM), naive Bayesian classifiers (NBC) [5], and classification and regression trees (CRT) [6]. Although certain results have been achieved, the machine learning techniques have obvious flaws: manually extracted features are insufficient and cannot further explicate the emotion of the paragraph, and more complex tasks require more complex function derivation, thereby limiting the system's capacity to handle complex tasks. In the early years, a lot of effort was made to seek answers to these limitations; then, neural networks were developed, and increasing attention was focused on deep learning.

Deep-learning-based sentiment analysis models avoid the need for the manual extraction of paragraph features and do not require more complex function derivations. However, deep learning models require a large amount of data for training, and the annotation of these training data needs to be carried out manually. This problem is present not only in the field of sentiment analysis, but also in other research that uses deep learning techniques. The advent of transformer networks and the advancement of transfer learning have opened greater possibilities for natural language processing. With the training of a large number of unlabeled datasets, models have a more comprehensive semantic understanding and can be fine-tuned for specific tasks to improve the accuracy of feature extraction. Network models based on Bert are starting to be employed [7].

For sentiment prediction, increased sentiment granularity can more correctly capture public impressions, enhance the credibility of social survey data, and simplify decision-making. Nevertheless, with more emotional content, trying to strike a balance between sentiment granularity and prediction accuracy is another huge obstacle.

The field of Chinese small-sample sentiment analysis is plagued by insufficient model training and the insufficient interpretability of deep learning models. We propose a progressive prediction approach to address these issues.

This method does not directly predict sentiment classification; rather, it first predicts whether the sentence contains sentiment, then predicts again with the predicted information and features to detect whether the statement is positive, negative, or neutral, and finally combines the predicted results with the output of BiLSTM-MultiHeadAttention and feeds them into the neural network to obtain the final multi-classification results.

We have made the following contributions:

- To improve the network's interpretability, a progressive prediction structure is proposed;
- An idea is provided for multi-category sentiment analysis tasks;

- A solution is offered to the problem of inadequate datasets for sentiment analysis jobs;
- For the Bert model's fine-tuning, a novel approach is proposed that produces good results with a significantly smaller dataset.

2. Related Work

Textual information on users' attitudes toward particular events, posted on the Internet, is of tremendous social importance; hence, it is essential to investigate the mood expressed in language. In this section, we provide a quick overview of the approaches employed thus far in the field of text sentiment analysis, introducing the most recent research results from classical to deep learning techniques.

2.1. Traditional Emotion Classification Methods

Early approaches to sentiment classification include sentiment-lexicon-based classification methods and machine-learning-based classification methods. Sentiment-dictionary-based classification methods calculate the sentiment score of a text by matching the text content to a sentiment dictionary using specific rules; the text is usually compiled by manual filtering. Dmitry Kan developed rules based on the grammatical features of the Russian language and classified the emotions of the Russian language through an emotion dictionary [8]. Liang Xu et al. [9] used this method to classify sentiment in Chinese. With the expansion of the Internet and the quickening of information iteration, the meanings of words change; however, frequent changes in dictionary classification and the generalization of new words require a significant amount of manpower. Despite the fact that heuristics were later proposed to construct sentiment dictionaries, it is difficult to adapt the heuristics to other sentiments because they are only applicable to a limited number of a priori-selected dimensions [10]. Sentiment classification issues can be solved with machine learning. These techniques include, but are not limited to, SVM, NBC, CRT, Random Forest (RF), and K-nearest Neighbor (KNN) [5,11,12]. Random forests and support vector machines are frequently referenced in relevant publications, and research demonstrates that random forests lead in computational performance, with SVMs performing even better [13]. Although machine learning algorithms may model jobs from the perspective of several features, the extraction of these features must be performed manually, and the models are difficult to deploy on a broad scale [14].

2.2. Deep Learning Methods

Researchers are devoting an increasing amount of time and energy to the study and development of deep learning and sentiment categorization models that are powered by deep learning. Convolutional neural networks (CNN) play a significant role in computer vision, and the You Only Look Once (YOLO) [15] network model has various applications in target identification. Natural language processing has also benefited from its use. The attention-based convolutional multi-label sentiment classification neural network uses the convolutional neural network to multi-label sentiment classification, and the model separates the attitudes carried by the phrases by replicating the human comprehension process [16]. The convolutional kernel is able to extract the entire utterance's characteristics more fully, and the addition of an attention mechanism further enhances the text sentiment analysis capability. A CNN-based model combines machine learning and convolutional neural networks to detect negative statements in text [17]. This method is comparable to the random forest classifier, and it is concluded that convolutional neural networks are more innovative and promising as classifiers. However, this method can only be applied to a single dimension, namely aggressiveness, and it is difficult to scale up for multicategorical sentiment analysis tasks [18]. This technique makes use of a "deep" network; however, the structure of the network is preset according to the amount of the text that is to be processed and is analogous to more conventional forms of mechanical learning.

Liao et al. [19] used a convolutional neural network for sentiment analysis and designed a simple convolutional neural network model for benchmarking. This network has

only one convolutional layer; the text features are extracted block by block through the pooling layer into the fully-connected layer; the dropout operation is added to prevent the occurrence of overfitting; and the final output is binary classification data.

Word2Vec is a neural network model proposed by Tomas Mikolov et al. [20] for training distributed word-embedding representations. This model captures the semantic similarity between words, and this similarity can be expressed in the word vector space. The word vectors are trained by context, and the context is predicted based on the words. When dealing with Chinese natural language tasks, Chinese words are composed of Chinese characters, and there is no space between words for segmentation due to the peculiarity of the Chinese language, which relies on readers' experience and context to recognize words.

Xu et al. [21] proposed using the word vector model for learning microblogging text features and proposed the CNN_Text_Word2vec model in combination with CNN networks. The advantages and disadvantages of the two different coding models, word granularity and letter granularity, were investigated for Chinese text sentiment analysis. The results show that Word2vec has a significant advantage over word-granularity-trained vectors in Chinese sentiment classification tasks.

CNN networks, however, are poor at learning the sequential dimension, which eliminates the lexical sequential features in sentences and reduces recognition accuracy [14]. Recurrent neural networks (RNN) can recognize contextual information on sequences and preserve the sequential features of words; however, due to the gradient disappearance problem, the capture of long-term memory by RNN does not achieve the desired results [22]. Thus, based on RNN, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) were proposed to make up for the shortages of RNN and solve the problem of gradient disappearance by selective forgetting. In the realm of emotion separation, LSTM has produced promising results [23]. Bidirectional LSTM (BiLSTM) and Bidirectional GRU (BiGRU) are, therefore, proposed to acquire contextual information in both directions. However, the network complexity for lengthy texts is quite significant [24].

The work of Felbo et al. [25] implements a system model through a two-layer LSTM network and an attention mechanism. This work pre-trains the models with millions of texts with emojis in social media so that they learn the features of the emotional content of the text and then fine-tune the phase where the features learned in the last LSTM layer may be too complex for the migration learning task. Therefore, making them directly accessible to the previous layer is more beneficial for migration. Ankush Chatterjee et al. [26] send the input user discourse to two LSTM layers using two different word embedding matrices, one for the semantic embedding and the other for sentiment embedding. The two obtained features are combined and passed into a fully-connected neural network with hidden layers for prediction. This method preserves both the semantic and sentiment information of the sentences and improves the prediction accuracy. However, because the authors do not provide a pre-training method, this model requires more labeled data for training, and the LSTM cannot encode back-to-front information; therefore, the method is not effective for classification at a finer granularity. The BiLSTM structure is more comprehensive for extracting information before and after the utterance, so some studies have used BiLSTM instead of LSTM and have made good progress [27].

2.3. Bert-Based Feature Extraction

The proposal of transfer learning has led to the beginning of pre-training models in the field of sentiment analysis and to attempts to improve the feature extraction ability of the models by pre-training them with unsupervised learning and training for task-specific accuracy using supervised fine-tuning.

Huang et al. [28] use pre-trained Bert for sentiment recognition in text conversations. Their method successfully captures the contextual sentiment information of the sentences and collects sentiment tags with specific sentiment-related tags such as #sad, #angry, #fear for model fine-tuning via a Twitter stream API in the fine-tuning phase. Puneet Kumar et al. [29] proposed a novel two-channel system for multiclass text sentiment

recognition using the Bert model to extract text features from input sentences in the form of embedding vectors, which are fed into the BiLSTM-CNN network and the CNN-BiLSTM network. The outputs of both channels appear as embedding vectors, which are concatenated and fed into the sentiment classification module. This resolves the problem that the initial convolution layer of the CNN-LSTM channel loses some order and sequence-related information of the text. The subsequent BiLSTM layer of the LSTM-CNN channel cannot fully utilize the text features in the input.

In summary, deep learning has made great progress in sentiment recognition tasks in recent years and, in the existing research, deep learning is better than machine learning for sentiment recognition tasks. For small-sample training sets, the same method of pre-training combined with task-specific fine-tuning often achieves good results. For example, a feature-extraction method based on Bert, combined with downstream tasks for emotion recognition, where Bert was trained with a large number of unlabeled datasets, performed well, and for subsequent tasks, BiLSTM replaced LSTM to obtain more complete semantic information. The accuracy was improved to some extent by using dual-channel technology, incorporating the advantages of BiLSTM and CNN. However, for the field of Chinese multiclassification sentiment analysis, the training of small samples for specific events still needs to be studied.

3. Proposed Method

This section describes the content of our task in detail, the relevant data and the processing details of cleaning the data, and our model architecture and related parameters.

3.1. Task Content

Our task was to analyze small-sample multicategory sentiment, train a model using different sizes of small-sample datasets, and compare the effectiveness of sentiment prediction compared to other models with small-sample datasets. Instead of making a direct judgment on the sentiment represented by the text for the input text, our model devises a multi-stage prediction method. The first stage predicts whether the input content carries sentiment and produces two possible results: present and absent. The second stage uses the obtained results as a reference and combines these with the extracted semantic features to predict whether the sentiment is positive, negative, or neutral. The third stage combines the results of the first two times to predict multicategory sentiment. To improve the accuracy of the model for small-sample predictions, the Bert model was fine-tuned using other datasets (labeled dichotomous dataset and labeled metaclassifier dataset) through the first and second tasks.

3.2. Composition and Structure of The Model

When carrying out Natural Language Processing (NLP) tasks, Word2Vec model and Glove model are often used to transform text into vectors, but these word-embedding methods struggle to represent the different meanings of words in different contexts. Using Bert to extract the emotional features embedded in social media solves this problem well, overcoming the problem of contextual influence on word meaning as well as allowing the fine-tuned word vectors to change according to the relevant downstream tasks to obtain more accurate and comprehensive semantic features.

3.2.1. Bert

We used a Bert module containing 12 multi-headed attentions with 768 hidden cells as the Bert layer in Figure 1. For the input, sentence $S = w_1, w_2, w_3, \dots, w_n$ was entered into this module by adding the classification tokens [CLS: classification] and [SEP: separator].

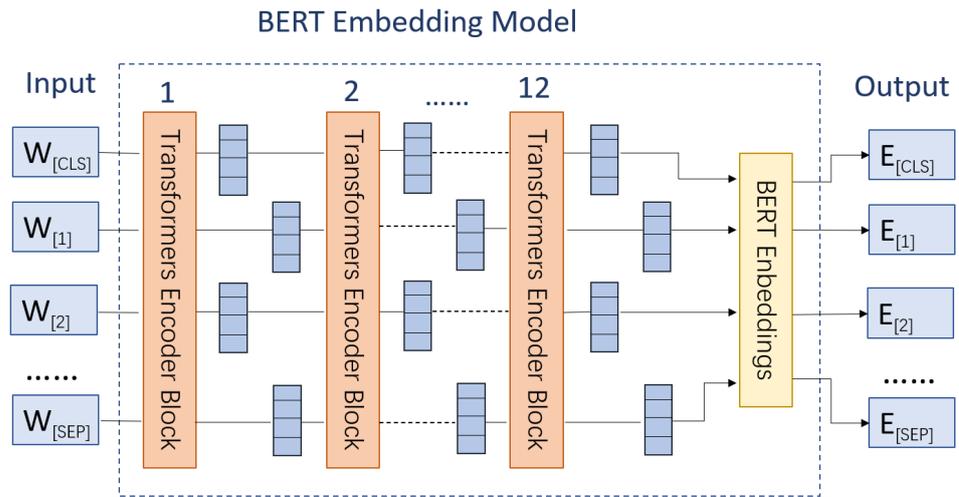


Figure 1. Bert embedding model [29].

3.2.2. BiLSTM

One problem with CNN is that it loses the text order and position information, which makes the processing of long texts difficult. LSTM eliminates this problem. The fundamental unit of LSTM is composed of an input gate, a forget gate, an output gate, and a memory cell. The input gate controls which information is input, the forgetting gate controls which information is forgotten in the neuron, and the output gate controls which information is output. See the formulas below:

$$input_i = \sigma(W_{input} \cdot [T_i, L_{i-1}, C_{i-1}] + b_{input}) \tag{1}$$

$$forget_i = \sigma(W_{forget} \cdot [T_i, L_{i-1}, C_{i-1}] + b_{forget}) \tag{2}$$

$$output_i = \sigma(W_{output} \cdot [T_i, L_{i-1}, C_{i-1}] + b_{output}) \tag{3}$$

where $input_i$, $forget_i$, and $output_i$ indicate the states of the input gate, the forget gate, and the output gate, respectively; σ represents the sigmoid activation function; W represents the weight matrix, and b represents the matrix bias term. T represents the input vector at time i , T_{i-1} is the previous moment's hidden state of the LSTM cell, and C_{i-1} is the previous moment's storage information. The storage details are displayed below.

$$L_i = output_i \cdot \tanh(C_i) \tag{4}$$

Although LSTM solves the problem of processing long text, it cannot extract the contextual information in the text; therefore, BiLSTM is introduced. Its structure is shown in Figure 2. The BiLSTM structure is a forward LSTM and a backward LSTM. After the input data go through the forward BiLSTM and the backward BiLSTM, the output data are put through the CAT operation to form the final output.

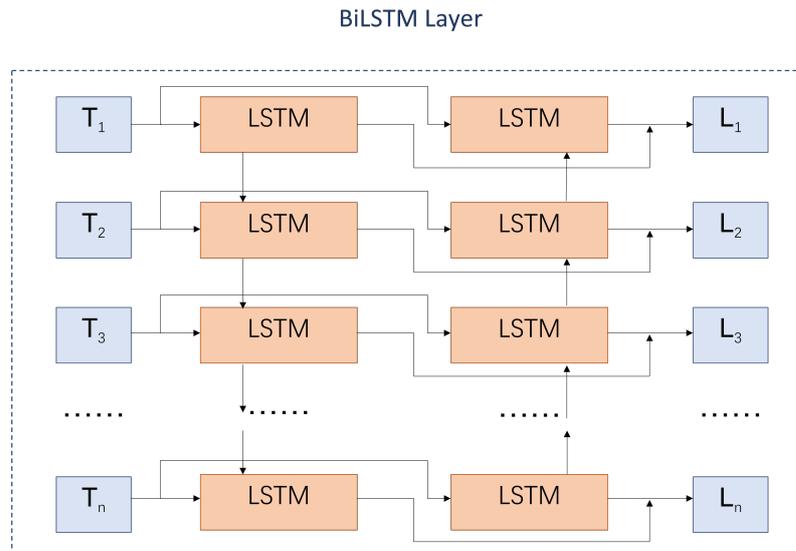


Figure 2. BiLSTM layer [30].

3.2.3. Multi-Headed Self-Attention

Attentional mechanisms originate from the study of human vision, and when processing complex information, humans choose to focus on only a portion of the information. When the model processes a word in one position in a sentence, self-attention allows for it to look at the words in other positions in the sentence to see if it can find some clues that will help it to encode the sentence. The self-attentive mechanism has corresponding encoders for the input information x : query, key, and value. The formulas are as follows:

$$Q = W^q x \tag{5}$$

$$K = W^k x \tag{6}$$

$$V = W^v x. \tag{7}$$

The formula for calculating the self-attention is as follows:

$$Attention(Q, K, V) = V \cdot \text{softmax}\left(\frac{K^T Q}{\sqrt{d}}\right); d \text{ is the dimension of } K \text{ or } Q. \tag{8}$$

We used a multi-headed attention mechanism based on self-attention, which allows for the better extraction of multiple semantics. The structure is shown in Figure 3. The calculation equations are as follows.

$$Q^i = W_i^q x + b_i^q \tag{9}$$

$$K^i = W_i^k x + b_i^k \tag{10}$$

$$V^i = W_i^v x + b_i^v \tag{11}$$

$$head_i = Attention(Q^i, K^i, V^i) \tag{12}$$

$$MultiHeadAttention(Q, K, V) = Concat(head_1, head_2, \dots, head_n)W^o + b^o \tag{13}$$

The output of BiLSTM is mapped onto n attention heads, the self-attention head of each attention head is calculated separately, and all the attention heads are stitched together to form the final attention output [31].

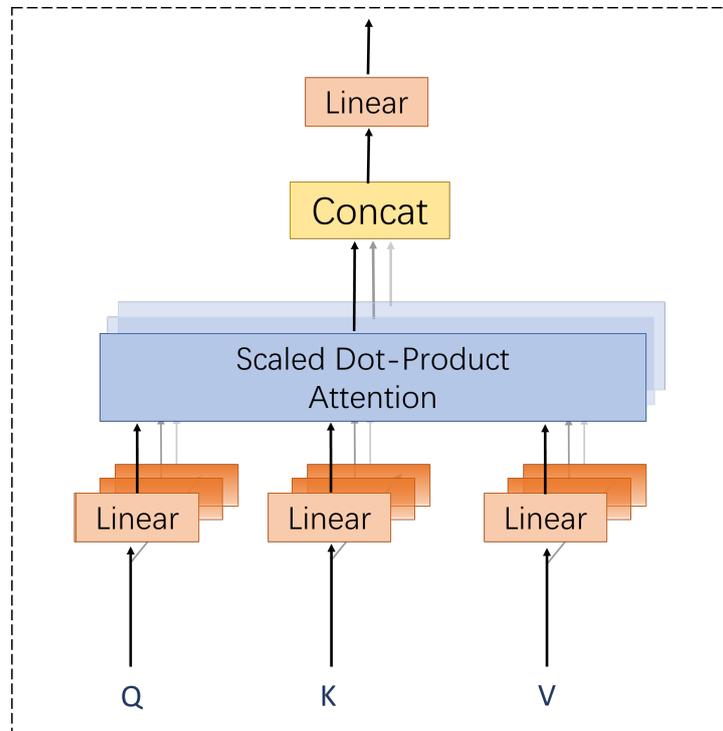


Figure 3. Multi-head attention [31].

3.2.4. Model Structure

The semantic information of the text content is extracted by the Bert layer, and the output of [cls] is taken for sentiment presence detection and positive and negative sentiment detection. A BiLSTM layer is connected after the Bert module to extract the temporal features of the sentences. After that, the extracted features are added to the attention module, and the attention and temporal features are summed to obtain the feature output layer. Batch normalization is performed on the feature output layer, and the final sentiment features are later input into the fully-connected network.

As shown in Figure 4, after the above steps, two outputs are obtained for prediction—one is the output of Bert’s [cls] character, and the other is the output of the sentiment feature with attention—and we use multi-stage prediction for sentiment prediction, after which these two features are input into the following three stages.

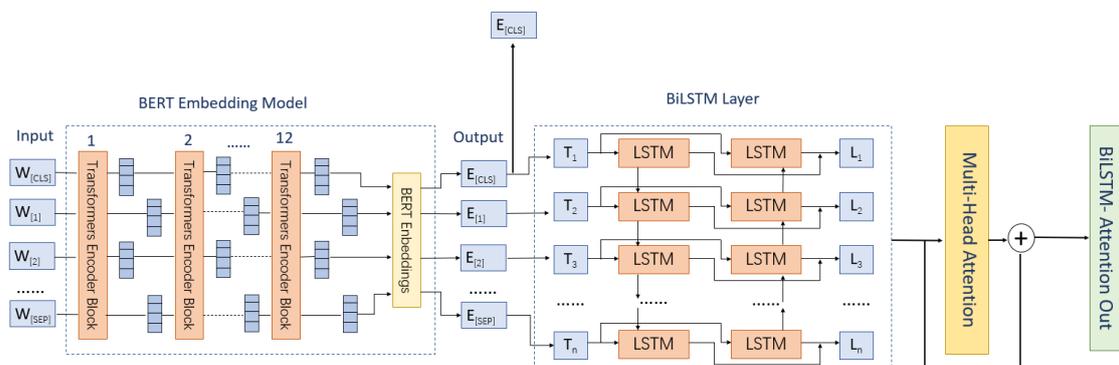


Figure 4. Feature extraction stage structure.

The first stage is the judgment of presence of emotion (JPOE) (Figure 5); the features of [cls] characters will be used as input. This is a linear layer with 768 input layers and 2 hidden layers, and the predicted results are output as a 2-dimensional tensor.

The second stage is the positive and negative prediction of emotion (JPNE) (Figure 5); the prediction result of the first stage and the features of the [cls] character will be used as input. This is a linear layer consisting of two layers: the first one is input as 768, the hidden layer as 50, and the second layer is input as 52 and output as 3. In the second layer, the output of the first fully-connected layer and the output of the previous stage are stitched together as input to obtain a prediction result for a three-dimensional vector.

The third stage is the multi-category sentiment prediction stage (MEJ) (Figure 5); the prediction results of the input JPOE, JPNE, and the sentiment features will be given attention. The first layer is the input layer of 500×768 , with a hidden layer of 50, which will focus on the emotional features' reshaping after the input, while the second layer of the input is 55 with an output of 6. The second layer will contain the output of the first layer before the link as well as the first two stages of the output of the stitching to obtain the final emotional prediction results.

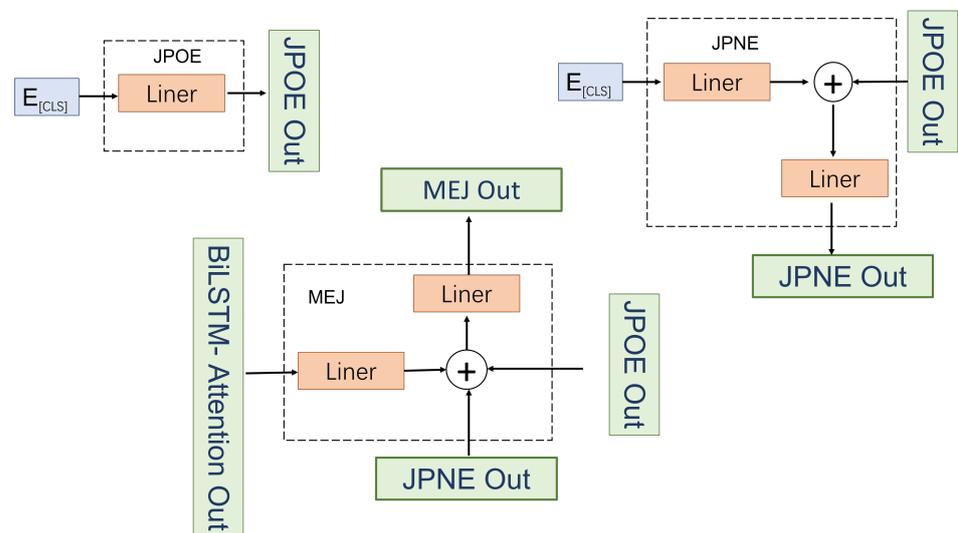


Figure 5. JPOE, JPNE, and MEJ structure.

3.3. Training Method

The training method is shown in Figure 6. We divided the training into three moments: T_1 , T_2 , and T_3 . The JPOE module takes the output of the [cls] character of Bert as input in the above three moments. The JPNE module takes the output of the [cls] character and the manually annotated JPOE module output label as input in moment T_1 and the output of the [cls] character and the JPOE module output in moment T_1 as input in moment T_2 . The JPOE module output in moment T_3 takes the output of the [cls] character and the JPOE module output in moment T_2 as input. The output of the [cls] character and the output of the JPOE module at time T_2 are used as inputs at time T_3 . In the third stage, T_1 , the moment for the JPOE module and the JPNE module result, is replaced with its label, and the correct value is passed on. In T_2 and T_3 , the moments for the JPOE module and JPNE module result are used as the result of the previous moment.

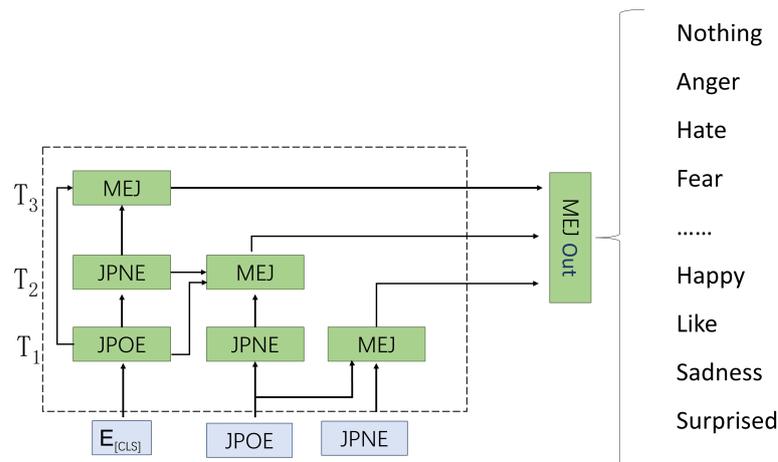


Figure 6. Training programs.

In the output at time T_1 , the inputs of the MEJ and JPNE modules contain the correct results of the previous item marked by humans, so they are not misled by the incorrect results obtained from the previous item. The T_3 moment eliminates human intervention. In the calculation of the loss function, the outputs of the different phases of T_1 , T_2 , and T_3 are considered.

The loss function is calculated as follows.

$$loss = loss_{JPOE}^{T_1} + loss_{JPNE}^{T_1} + loss_{MEJ}^{T_1} + loss_{MEJ}^{T_2} + loss_{MEJ}^{T_3} \quad (14)$$

3.4. Fine-Tuning Method

The Bert model is pre-trained on a large and annotated corpus with a masked language model (MLM) target and then fine-tuned on task-specific supervised data [32]. The fine-tuning of migration learning based on the Bert model is achieved by adding a specific classification layer (Bert+task-specific model) behind the Bert network, outputting the input from Bert as contextual features, passing the contextual features into the subsequent task, and performing end-to-end training. To fine-tune Bert on specific supervised data, a larger dataset is required for the problem of oversized models, resulting in high training costs [33]. In order to achieve better results when fine-tuning Bert using smaller datasets, we tried to reduce the number of parameters in the fine-tuning phase of the model by freezing most of the parameters of Bert in the fine-tuning phase and training only the attention parameters of the last encoder layer to achieve the fine-tuning effect.

We fine-tuned the model using Chinese sentiment classification datasets, which are compatible with all sentiment classification datasets due to the design of our subsequent task structure. JPOE and JPNE predict the polarity of the sentiment. For different datasets with different sentiment labels, previous sentiment classification models can only use a single dataset. Due to our use of stage prediction, all multi-categorization datasets can be planned for positive sentiment and negative sentiment, so JPOE and JPNE are compatible with other datasets. MEJ can be decided based on whether the pre-training dataset is compatible with the task we want to perform.

4. Experiments

4.1. Fine-Tuning the Dataset

The data used for our task came from the following datasets:

- ccf_tcci2018: This dataset includes more than 40,000 sentences, characterized as null, like, sad, disgust, anger, and happiness, and labeled 0 to 5 in order. The data were obtained from the NLPCC Emotion Classification Challenge (17,113 items in training data; 2242 items in test data) and manually labelled after filtering the Weibo data (23,000 items in training data; 2500 items in test data).

- ccf_tcci2013: The data were obtained from Sina Weibo, with a total of 40,000 Weibo containing 8 types of emotions: anger, disgust, fear, happiness, like, sadness, and surprise. One Weibo may contain several different individual emotions, and the emotion classification is based on the theme classification.

4.2. Experimental Dataset

The SMP2020-EWECT dataset was provided by the Social Computing and Information Retrieval Research Center of Harbin Institute of Technology, originating from Sina Weibo and provided by the Micro Hotspot Big Data Research Institute. The dataset is divided into two parts. The first part is the general microblogging dataset. This dataset was obtained randomly from the microblogging content, without identifying specific topics, covering a wide range of subjects. The second part is the COVID-19 Weibo dataset, which contains Weibo content obtained during the pandemic using relevant keyword filtering related to the COVID-19 pandemic. Therefore, the training set of this evaluation contains the above two types of data: general microblog training data and COVID-19 Weibo training data. The test set is also divided into the general microblog test set and the COVID-19 Weibo test set. Each Weibo was labelled as one of the following six categories: neutral, happy, angry, sad, fear, or surprise. A total of 27,768 Weibo were included in the training dataset, 2000 Weibo in the validation set, and 5000 Weibo in the test dataset. The COVID-19 Weibo training dataset consists of 8606 Weibo, while the validation set contains 2000 Weibo, and the test dataset contains 3000 Weibo.

4.3. Data Normalization

This operation replaces email addresses, url addresses, phone numbers, Weibo super talk logos, usernames, numbers, etc., in the data. This content should be meaningless, and even special Twitter names and email addresses can interfere with the prediction results. See Table 1 for specific replacement options.

Table 1. Data cleaning example.

Filtering Items	Example	After Data Cleaning
Email	举报邮箱125****dd@qq.com Report email 125 **** dd@qq.com	举报邮箱<Email> Report email <Email>
Phone	24小时热线电话133****1845 24-h hotline 133****1845	24小时热线电话<Phone> 24-h hotline <Phone>
Weibo name	//@央视新闻: 筑梦航天 //@CCTV News: Building Dreams of Space	筑梦航天 Building Dreams of Space
Weibo super talk	#如何美白#亲测维生素c的效果不错。 #How to whiten# Pro-vitamin C works well.	亲测维生素c的效果不错。 Pro-tested vitamin C works well.
URL	欢迎访问: https://www.baidu.com Welcome to: https://www.baidu.com	欢迎访问: <URL> Welcome to: <URL>

4.4. Scoring Function

Accuracy and Macro_F values are used for evaluation.

$$P_e = \frac{TP_e}{TP_e + FP_e} \tag{15}$$

$$R_e = \frac{TP_e}{TP_e + FN_e} \tag{16}$$

$$F_e = \frac{2 \times P_e \times R_e}{P_e + R_e} \tag{17}$$

$$Macro_F = \frac{1}{n} \sum_{e \in Emotions} F_e \tag{18}$$

$$Accuracy = P_e \tag{19}$$

4.5. Experiment 1: Determine Small Datasets and Validate the Performance of Our Model on Large Datasets

This experiment addresses the question of where to draw the line between small and large datasets, while validating the performance of our model when trained with large datasets. The experiments use Accuracy and Macro_F values as evaluation metrics, using mainstream deep learning models as comparisons. The SMP2020-EWECT dataset was used for the experiments, and the number of each sentiment classification included in the training set is shown in Figure 7. In the original SMP2020-EWECT training set, the number of each sentiment classification was different. The larger subset of data created from the original data in this case will contain far more classifications with larger numbers than smaller numbers, while the amount of data used for the different classifications is balanced when creating a subset of data with smaller numbers. We expanded the smaller number of categories by adding them repeatedly, so that the number of categories was balanced. Considering the difference in the number of different kinds of emotions in the dataset, the most emotions in the SMP2020-EWECT general dataset contained more than 8000 pieces of data, and the majority of emotions contained more than 4000 pieces of data; therefore, 8, 80, 800, and 8000 data points were intercepted for each category to build a new dataset. Most emotions in the SMP2020-EWECT COVID-19 Weibo dataset contained fewer than 2000 data points, so 2, 20, 200, and 2000 pieces of data were intercepted for each category to build a new dataset.

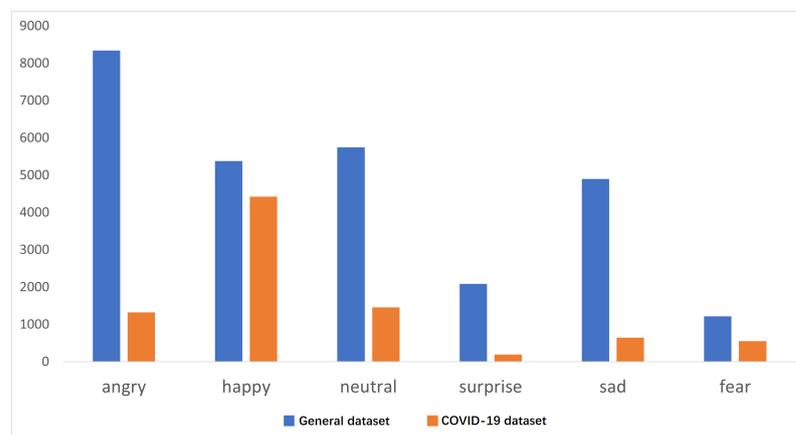


Figure 7. Volume of data for different moods.

The relevant parameters of the model are listed in Table 2, and Table 3 lists the configurations of the experimental environment.

Table 2. Experimental parameters of our model.

Parameter Name	Parameter Value
Attention-head	8
Batch_size	32
Learning Rate	1×10^{-5}
Dropout	0.1

Table 3. Experimental environment.

Experimental Environment	Experimental Setting
Operating System	Ubuntu 9.4
Programming Language	Python 3.8.10
Deep Learning Framework	Torch 1.13.1 cu117
Graphics Card	RTX 3090

4.6. Baseline

1. TextCNN [34] extracts the local key features using convolutional operations and filters the salient features using the maximum pool. Then, this information is used for classification;
2. Bi-LSTM [35] is used to extract contextual semantic and sequential information from text;
3. Bi-LSTM-Attention [36] adds an attention mechanism to the Bi-LSTM model, which enables the model to adjust the weight parameters according to different levels of data importance and increase the weight of key information;
4. RCNN [37] first extracts global semantic information using Bi-LSTM and then extracts salient features using max-pooling. Then, this information is used for classification;
5. BERT-Bi-LSTM [38] is a classification model that uses Bert to generate dynamic feature vectors and then uses the Bi-LSTM network for feature extraction;
6. BERT-Bi-LSTM-atten [39] uses Bert to generate dynamic feature vectors, then uses the Bi-LSTM network for feature extraction, and finally uses the attention mechanism for feature optimization.

The results of the general dataset experiment are shown in Table 4. The experimental data of the COVID-19 Weibo dataset are shown in Table 5. The experiments show that the SMP2020-EWECT general dataset contains 80 pieces of data per classification, and the SMP2020-EWECT COVID-19 Weibo dataset contains 20 classifications per classification. None of the model's Acc and F1 values reached 0.6 when the amount of data per classification was less than 10^2 , defined as a small dataset; if the data per classification was less than 10, this was defined as an ultra-small dataset. The performance of our model on large datasets is comparable to that of the current mainstream sentiment analysis models, with both the general dataset and the COVID-19 Weibo dataset outperforming the other models.

Table 4. Experimental results of the SMP2020-EWECT general dataset.

Model	8		80		800		8000	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
RCNN	0.312	0.248	0.489	0.449	0.656	0.640	0.702	0.673
TextCNN	0.321	0.279	0.593	0.532	0.721	0.695	0.750	0.722
BiLSTM	0.287	0.112	0.334	0.315	0.604	0.588	0.715	0.664
BiLSTM-Attention	0.273	0.153	0.373	0.345	0.628	0.590	0.727	0.694
Bert-BiLSTM	0.467	0.426	0.586	0.534	0.690	0.654	0.738	0.703
Bert-BiLSTM-Attention	0.425	0.389	0.544	0.508	0.701	0.677	0.734	0.703
BBMEJ	0.466	0.035	0.599	0.484	0.715	0.666	0.746	0.708

Table 5. Experimental results of the SMP2020-EWECT COVID-19 Weibo dataset.

Model	2		20		200		2000	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
RCNN	0.483	0.136	0.432	0.193	0.626	0.474	0.738	0.582
TextCNN	0.165	0.106	0.508	0.126	0.657	0.525	0.702	0.556
BiLSTM	0.134	0.110	0.440	0.222	0.620	0.453	0.714	0.544
BiLSTM-Attention	0.427	0.159	0.444	0.255	0.625	0.469	0.721	0.543
Bert-BiLSTM	0.387	0.189	0.525	0.316	0.619	0.447	0.747	0.591
Bert-BiLSTM-Attention	0.469	0.165	0.555	0.364	0.645	0.508	0.747	0.596
BBMEJ	0.423	0.171	0.544	0.420	0.626	0.482	0.751	0.616

4.7. Experiment 2: Training the Model Using a Small Amount of Data from SMP2020-EWECT

In order to more fully investigate the effect of a small amount of data on the model accuracy and F1 values, we took out 1, 2, 4, 8, 16, and 32 pieces of data for each of the 6 sentiments, forming 6 small-sample data sets. The prediction accuracy and Macro-F1 were obtained by training the 6 small-sample datasets and obtaining predictions on the test set.

To improve the accuracy of our model, we used a fine-tuning scheme to fine-tune our model Bert encoder and some parameters using the *ccf_tcci_2013* and *ccf_tcci2018* datasets.

The results of the general dataset experiment are shown in Table 6. The experimental data of the COVID-19 Weibo dataset are shown in Table 7. The experimental results show that our model achieves the highest accuracy and Macro-F1 values on every small dataset, outperforming the mainstream deep learning models and the latest sentiment analysis models. This shows that our model is effective. Our model achieves very good results over both the general dataset and the COVID-19 Weibo dataset, with the highest correctness and Macro-F1 values. In the COVID-19 Weibo dataset, our model also maintains the highest correct rate, and the Macro-F1 value is 0.019 higher than that of our model for a single sentiment with 32 data. However, the correct rate is 0.064 lower than ours, and the change curve of our model’s correct rate and Macro-F1 value indicate that our model shows less fluctuation compared to the other models.

Table 6. Training results for small samples of the general dataset.

Model	1		2		4		8		16		32	
	Acc	F1										
RCNN	0.294	0.083	0.379	0.194	0.285	0.242	0.379	0.366	0.377	0.370	0.498	0.439
TextCNN	0.160	0.108	0.430	0.316	0.259	0.125	0.320	0.278	0.394	0.291	0.406	0.409
Bi-LSTM	0.218	0.084	0.187	0.076	0.223	0.127	0.324	0.267	0.240	0.176	0.331	0.313
Bi-LSTM-Attention	0.210	0.058	0.310	0.187	0.238	0.148	0.259	0.192	0.219	0.216	0.362	0.270
BERT-Bi-LSTM	0.293	0.076	0.277	0.185	0.333	0.319	0.451	0.421	0.433	0.394	0.520	0.462
BERT-Bi-LSTM-atten	0.315	0.209	0.359	0.236	0.480	0.358	0.438	0.393	0.488	0.463	0.555	0.423
BBMEJ-BTAFT	0.339	0.270	0.448	0.282	0.586	0.439	0.593	0.480	0.621	0.488	0.629	0.521

Table 7. Training results for small samples of the COVID-19 Weibo dataset.

Model	1		2		4		8		16		32	
	Acc	F1										
RCNN	0.513	0.113	0.511	0.129	0.463	0.219	0.513	0.113	0.430	0.314	0.539	0.310
TextCNN	0.466	0.147	0.265	0.212	0.165	0.097	0.473	0.245	0.116	0.121	0.345	0.175
Bi-LSTM	0.074	0.048	0.281	0.155	0.411	0.125	0.424	0.258	0.425	0.228	0.431	0.264
Bi-LSTM-Attention	0.504	0.117	0.507	0.125	0.510	0.125	0.367	0.183	0.452	0.250	0.378	0.283
BERT-Bi-LSTM	0.513	0.113	0.231	0.176	0.348	0.242	0.508	0.345	0.537	0.338	0.572	0.424
BERT-Bi-LSTM-atten	0.513	0.113	0.253	0.102	0.384	0.207	0.469	0.362	0.472	0.338	0.513	0.387
BBMEJ-BTAFT	0.594	0.295	0.652	0.368	0.672	0.354	0.665	0.387	0.654	0.436	0.636	0.405

5. Ablation Experiments

To verify whether the fine-tuning of Bert's last layer of attention had a positive impact on the results of the experiment, we designed the following ablation experiment.

Ablation experiment 1: Validate the model's fine-tuning results with a small dataset after fine-tuning all Bert parameters.

Ablation experiment 2: Test the results achieved in a small dataset using a model without a small amount of parameter fine-tuning.

Ablation experiment 3: No fine-tuning of the Bert model, but fine-tuning of the JPOE and JPNE models to test the results achieved in a small dataset.

Experiment 1: After fine-tuning all Bert parameters and tested with a small dataset, the results achieved by our model were not good; the Macro-F1 was bad and did not exceed the results before fine-tuning. Our fine-tuned model Macro-F1 performed well, with fewer fluctuations, and the correctness rate remained above a high level.

Experiment 2: We tested the results in a small dataset using a model without a small amount of parameter fine-tuning. The model without a small amount of fine-tuning also outperformed the other models in terms of results, but the accuracy of the fine-tuned model was always higher than that of the less fine-tuned model. The Macro-F1 value exceeded that of the fine-tuned model by 0.034 at a single classification of 32, which did not open a large gap and gave the previously fine-tuned model a clear advantage.

Experiment 3: There was no fine-tuning of the Bert model, but fine-tuning of JPOE and JPNE lead to a higher accuracy than that of the model without fine-tuning; however, it was second to the effect of using the BTAFT fine-tuning scheme, which is equal in terms of recall.

The results of the experiments are shown in Figures 8 and 9. The ablation experiments demonstrate that our model is structurally effective and improves the original model for Chinese sentiment classification of small datasets. The ablation experiments demonstrate that a small amount of fine-tuning is better than fine-tuning all model parameter fine-tuning, and the performance of the model is even better after a small amount of parameter fine-tuning.

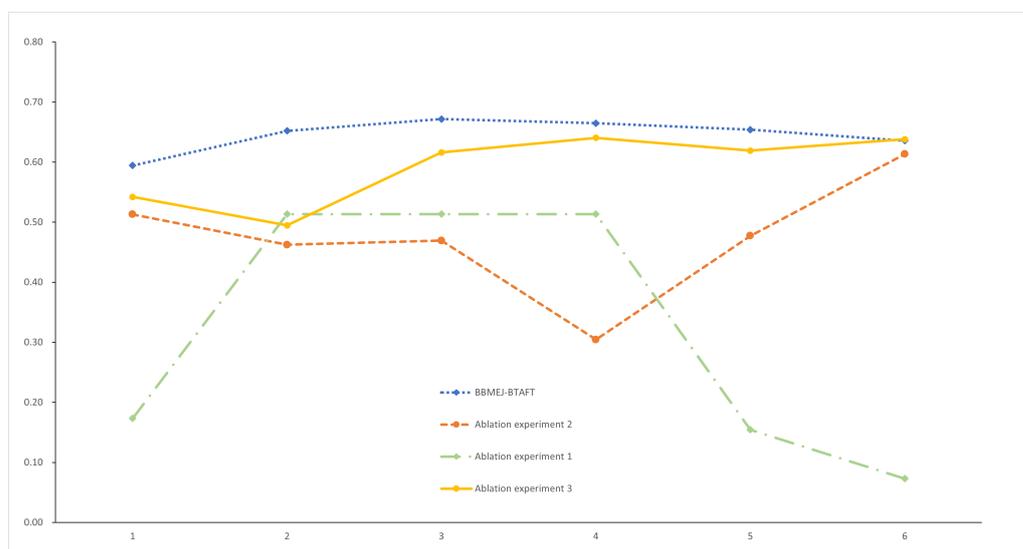


Figure 8. Ablation experiment accuracy.

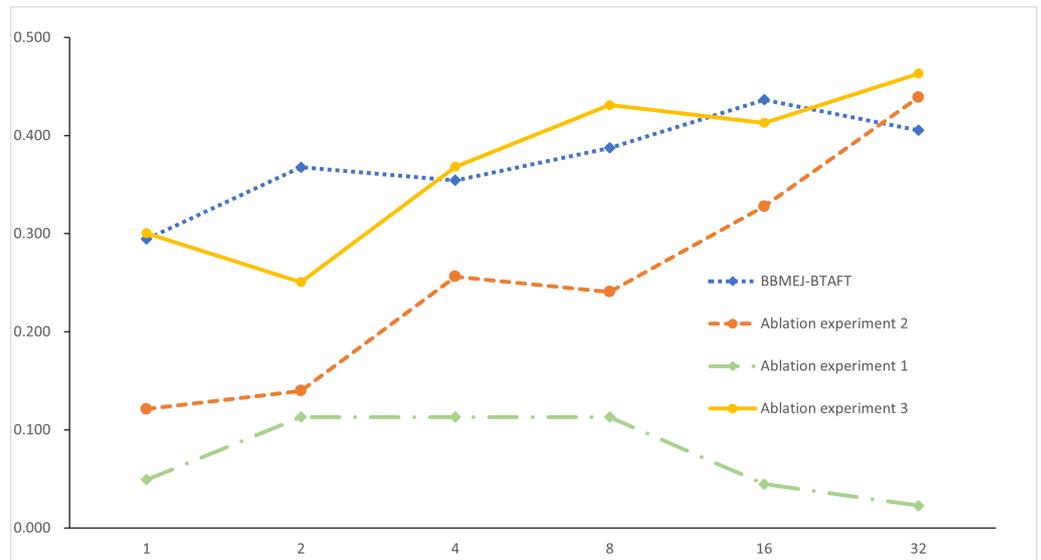


Figure 9. Ablation experiment Macro-F1.

6. Interpretability Verification

To more intuitively verify the interpretability of the model, we projected the output of JPNE in 3D space (Figures 10 and 11); the x, y, and z axes represent the probabilities of negative sentiment, positive sentiment, and neutral sentiment, respectively. It can be seen that the points predicting the same emotion are much tighter. Anger, fear, and sadness belong to the same negative emotion, and they have a smaller cluster spacing.

We can verify that the prediction of JPNE has an effect on the final decision by feeding MEJ the wrong values of JPNE and JPOE in the prediction phase. Then, we can determine if this affects the prediction results made by MEJ.

Figure 12 shows the confusion matrix of the prediction results for the normal case, and Figure 13 shows the matrix perturbed by the wrong JPNE and JPOE values, showing that the accuracy of the MEJ prediction decreases and the prediction of neutral sentiment increases after corrupting the prediction results of JPOE and JPNE. The accuracy of the other sentiments decreases because JPNE and JPOE do not provide more accurate results. This suggests that JPOE and JPNE have an effect on the prediction of MEJ.

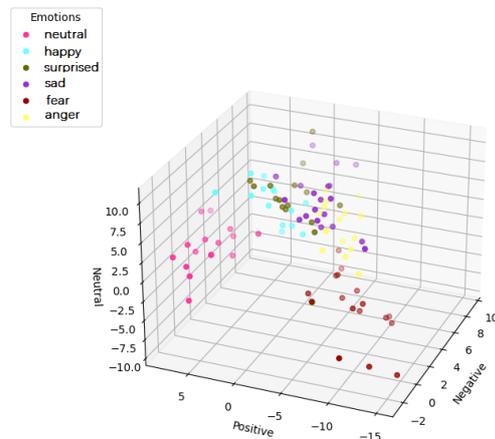


Figure 10. Scatterplot of positive emotion distribution.

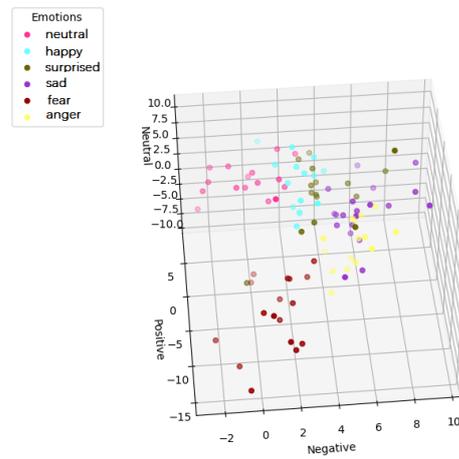


Figure 11. Scatterplot of negative emotion distribution.

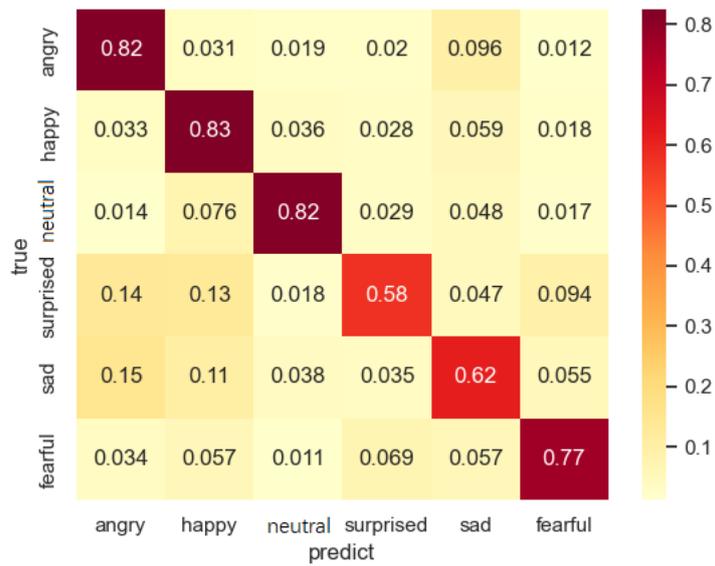


Figure 12. Before interfering with JPOE and JPNE.

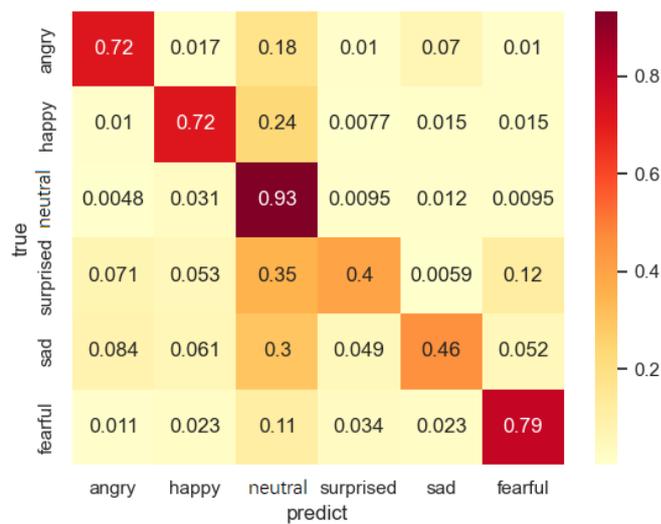


Figure 13. After interfering with JPOE and JPNE.

7. Conclusions

Compared with existing methods, our approach has the following advantages:

- Transfer learning can be performed for different sentiment analysis datasets, which improves the processing ability of small datasets, so the model does not need a large amount of data to achieve satisfactory results.
- The problem of insufficient datasets for sentiment recognition is solved.
- The interpretability of deep learning is enhanced, so that deep learning, as a black box model, has higher credibility. A reasonable interpretation of the model's decisions can be achieved through attention and a JPOE–JPNE structure.
- The improvements to the Bert fine-tuning approach make the Bert's encoding more task-adapted and reduce the input of the training dataset. Therefore, the results on small datasets are significantly improved by transfer learning with the Bert fine-tuning approach.

Our model has plenty of room for improvement, and we will continue to enhance it by adding convolutional networks and feature fusion to increase the accuracy. The calculation parameters of the loss function of the model also need to be adjusted to achieve the best state.

Author Contributions: Conceptualization, J.W. and Q.Z.; writing—original draft, J.W. and Q.Z.; methodology, J.W. and D.C.; software, J.W. and D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository.

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

References

1. Li, Z.; Zhou, L.; Yang, X.; Jia, H.; Li, W.; Zhang, J. User Sentiment Analysis of COVID-19 via Adversarial Training Based on the BERT-FGM-BiGRU Model. *Systems* **2023**, *11*, 129. [CrossRef]
2. Yan, S.; Wang, J.; Song, Z. Microblog Sentiment Analysis Based on Dynamic Character-Level and Word-Level Features and Multi-Head Self-Attention Pooling. *Future Internet* **2022**, *14*, 234. [CrossRef]
3. Zeng, L.; Li, R.Y.M.; Zeng, H. Weibo users and Academia's foci on tourism safety: Implications from institutional differences and digital divide. *Heliyon* **2023**, *9*, e12306. [CrossRef]
4. Xie, K.; Mao, Z.E.; Wu, J. Learning from peers: the effect of sales history disclosure on peer-to-peer short-term rental purchases. *Int. J. Hosp. Manag.* **2019**, *76*, 173–183. [CrossRef]
5. Jamal, N.; Xianqiao, C.; Aldabbas, H. Deep Learning-Based Sentimental Analysis for Large-Scale Imbalanced Twitter Data. *Future Internet* **2019**, *11*, 190. [CrossRef]
6. Bibi, R.; Qamar, U.; Ansar, M.; Shaheen, A. Sentiment Analysis for Urdu News Tweets Using Decision Tree. In Proceedings of the 2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA), Honolulu, HI, USA, 29–31 May 2019; pp. 66–70.
7. Petrolini, M.; Cagnoni, S.; Mordonini, M. Automatic Detection of Sensitive Data Using Transformer- Based Classifiers. *Future Internet* **2022**, *14*, 228. [CrossRef]
8. Kan, D. Rule-based approach to sentiment analysis at ROMIP 2011. In *Komp'iuternaia Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoi Konferentsii Dialog*. 2012; Volume 24. Available online: <https://www.scimagojr.com/journalsearch.php?q=21100325444&tip=sid&clean=0> (accessed on 1 April 2023).
9. Xu, L.; Li, L.; Jiang, Z.; Sun, Z.; Wen, X.; Shi, J.; Sun, R.; Qian, X. A Novel Emotion Lexicon for Chinese Emotional Expression Analysis on Weibo: Using Grounded Theory and Semi-Automatic Methods. *IEEE Access* **2021**, *9*, 92757–92768. [CrossRef]
10. Madani, Y.; Erritali, M.; Bengourram, J.; Sailhan, F. A hybrid multilingual fuzzy-based approach to the sentiment analysis problem using SentiWordNet. *Int. J. Uncertain. Fuzziness-Knowl.-Based Syst.* **2020**, *28*, 361–390. [CrossRef]
11. Perikos, I.; Hatzilygeroudis, I. Recognizing emotions in text using ensemble of classifiers. *Eng. Appl. Artif. Intell.* **2016**, *51*, 191–201. [CrossRef]

12. Naz, S.; Sharan, A.; Malik, N. Sentiment classification on twitter data using support vector machine. In Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Santiago, Chile, 3–6 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 676–679.
13. Chatzakou, D.; Vakali, A.; Kafetsios, K. Detecting Variation of Emotions in Online Activities. *Expert Syst. Appl.* **2017**, *89*, 318–332. [[CrossRef](#)]
14. Gan, C.; Feng, Q.; Zhang, Z. Scalable multi-channel dilated CNN-BiLSTM model with attention mechanism for Chinese textual sentiment analysis. *Future Gener. Comput. Syst.* **2021**, *118*, 297–309. [[CrossRef](#)]
15. Yan, W.; Wang, X.; Tan, S. YOLO-DFAN: Effective High-Altitude Safety Belt Detection Network. *Future Internet* **2022**, *14*, 349. [[CrossRef](#)]
16. Shin, J.; Kim, Y.; Yoon, S.; Jung, K. Contextual-CNN: A Novel Architecture Capturing Unified Meaning for Sentence Classification. In Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, China, 15–17 January 2018; pp. 491–494. [[CrossRef](#)]
17. Gordeev, D. Detecting state of aggression in sentences using CNN. In Proceedings of the Speech and Computer: 18th International Conference, SPECOM 2016, Budapest, Hungary, 23–27 August 2016; Proceedings 18; Springer: Berlin/Heidelberg, Germany, 2016; pp. 240–245.
18. Kratzwald, B.; Ilić, S.; Kraus, M.; Feuerriegel, S.; Prendinger, H. Deep learning for affective computing: Text-based emotion recognition in decision support. *Decis. Support Syst.* **2018**, *115*, 24–35. [[CrossRef](#)]
19. Liao, S.; Wang, J.; Yu, R.; Sato, K.; Cheng, Z. CNN for situations understanding based on sentiment analysis of twitter data. *Procedia Comput. Sci.* **2017**, *111*, 376–381. [[CrossRef](#)]
20. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
21. Xu, D.; Tian, Z.; Lai, R.; Kong, X.; Tan, Z.; Shi, W. Deep learning based emotion analysis of microblog texts. *Inf. Fusion* **2020**, *64*, 1–11. [[CrossRef](#)]
22. Zhang, Z.; Wang, L.; Zou, Y.; Gan, C. The optimally designed dynamic memory networks for targeted sentiment classification. *Neurocomputing* **2018**, *390*, 36–45. [[CrossRef](#)]
23. Gao, M.; Xiao, Q.; Wu, S.; Deng, K. An Improved Method for Named Entity Recognition and Its Application to CEMR. *Future Internet* **2019**, *11*, 185. [[CrossRef](#)]
24. Xu, G.; Meng, Y.; Qiu, X.; Yu, Z.; Wu, X. Sentiment Analysis of Comment Texts Based on BiLSTM. *IEEE Access* **2019**, *7*, 51522–51532. [[CrossRef](#)]
25. Felbo, B.; Mislove, A.; Søgaard, A.; Rahwan, I.; Lehmann, S. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv* **2017**, arXiv:1708.00524.
26. Chatterjee, A.; Gupta, U.; Chinnakotla, M.K.; Srikanth, R.; Galley, M.; Agrawal, P. Understanding emotions in text using deep learning and big data. *Comput. Hum. Behav.* **2019**, *93*, 309–317. [[CrossRef](#)]
27. Chen, T.; Xu, R.; He, Y.; Wang, X. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Syst. Appl.* **2017**, *72*, 221–230. [[CrossRef](#)]
28. Huang, Y.H.; Lee, S.R.; Ma, M.Y.; Chen, Y.H.; Yu, Y.W.; Chen, Y.S. EmotionX-IDEA: Emotion BERT—an Affectional Model for Conversation. *arXiv* **2019**, arXiv:1908.06264.
29. Kumar, P.; Raman, B. A BERT based dual-channel explainable text emotion recognition system. *Neural Netw.* **2022**, *150*, 392–407. [[CrossRef](#)]
30. Yang, C.; Sheng, L.; Wei, Z.; Wang, W. Chinese Named Entity Recognition of Epidemiological Investigation of Information on COVID-19 Based on BERT. *IEEE Access* **2022**, *10*, 104156–104168. [[CrossRef](#)]
31. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
32. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
33. Zaken, E.B.; Ravfogel, S.; Goldberg, Y. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv* **2021**, arXiv:2106.10199.
34. Kim, Y. *Convolutional Neural Networks for Sentence Classification Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Emnlp 2014, Doha, Qatar, 25–29 October 2014, a Meeting of Sigdat, a Special Interest Group of the Acl*; Association for Computational Linguistics: Doha, Qatar, 2014.
35. Cho, K.; Van Merriënboer, B.; 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.
36. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers), Berlin, Germany, 7–12 August 2016; pp. 207–212.
37. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.

38. Peng, S.; Zeng, R.; Liu, H.; Chen, G.; Wu, R.; Yang, A.; Yu, S. Emotion classification of text based on BERT and broad learning system. In Proceedings of the Web and Big Data: 5th International Joint Conference, APWeb-WAIM 2021, Guangzhou, China, 23–25 August 2021; Proceedings, Part I 5; Springer: Berlin/Heidelberg, Germany, 2021; pp. 382–396.
39. Song, G.; Huang, D. A sentiment-aware contextual model for real-time disaster prediction using Twitter data. *Future Internet* **2021**, *13*, 163. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.