future internet

Article

Microblog Sentiment Analysis Based on Dynamic Character-Level and Word-Level Features and Multi-Head Self-Attention Pooling

Shangyi Yan, Jingya Wang * and Zhiqiang Song

College of Information and Cyber Security, People's Public Security University of China, Beijing 100038, China; 201621440006@stu.ppsuc.edu.cn (S.Y.); songtosong0914@gmail.com (Z.S.)
* Correspondence: wangjingya@ppsuc.edu.cn

Abstract: To address the shortcomings of existing deep learning models and the characteristics of microblog speech, we propose the DCCMM model to improve the effectiveness of microblog sentiment analysis. The model employs WOBERT Plus and ALBERT to dynamically encode character-level text and word-level text, respectively. Then, a convolution operation is used to extract local key features, while cross-channel feature fusion and multi-head self-attention pooling operations are used to extract global semantic information and filter out key data, before using the multi-granularity feature interaction fusion operation to effectively fuse character-level and word-level semantic information. Finally, the Softmax function is used to output the results. On the weibo_senti_100k dataset, the accuracy and F1 values of the DCCMM model improve by 0.84% and 1.01%, respectively, compared to the best-performing comparison model. On the SMP2020-EWECT dataset, the accuracy and F1 values of the DCCMM model improve by 1.22% and 1.80%, respectively, compared with the experimental results of the best-performing comparison model. The results showed that DCCMM outperforms existing advanced sentiment analysis models.

Keywords: dynamic character and word encoding; multi-head self-attention pooling; multi-granularity feature interactive fusion; microblog sentiment analysis

check for updates

Citation: 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. https://doi.org/10.3390/fi14080234

Academic Editor: Michael Sheng

Received: 25 June 2022
Accepted: 23 July 2022
Published: 29 July 2022

1. Introduction

In recent years, Internet social media platforms have emerged rapidly, and because of their advantages of rapid dissemination, wide audience, and ease of use, more and more Internet users are choosing to use Internet social platforms to express their opinions and exchange information, resulting in a huge amount of text data with emotional overtones. In particular, Microblog, as the world's largest Chinese social platform, has become an important way for the public to express their emotions, opinions, and attitudes [1]. According to Microblog's Q4 2021 earnings report, Microblog reached 573 million monthly active users by the end of Q4 2021, up 10% year-on-year, and 249 million daily active users, up 11% year-on-year. Due to the popularity of Microblog and the low threshold for posting information, the discourse on this social media platform provides the most up-to-date and extensive information. Therefore, sentiment analysis based on microblogs has always been a hot topic for research. By analyzing the sentiment of microblogs and mining valuable information, we can obtain information about people's preferences for specific products, and their concerns and emotional changes regarding government policies or social phenomena. This is important for enterprise product marketing, online opinion monitoring and analysis, national and social security, etc. [2].

However, there are significant differences between microblog sentiment texts and ordinary texts: microblog sentiment texts have a large number of "different meanings for the same word"; they are more colloquial and contain many Internet buzzwords with a higher degree of irregularity; and they are shorter and contain limited information, which requires a higher level of ability in semantic understanding within the model. All these present considerable challenges in microblog sentiment analysis [3].

Lexicon and traditional machine learning are the main approaches used in early microblog sentiment analysis studies. Although lexicon and rule-based methods are simple, easy to implement, and perform rapid classification, their accuracy depends on the quality of the sentiment lexicon and the rationality of the manually designed rules. It requires extensive manual intervention; thus, the actual classification effect is not ideal [4]. Sentiment analysis methods based on traditional machine learning need to extract feature information from the text and then use classification algorithms to classify the sentiment polarity of the text, generally using three classification algorithms: Naive Bayes (NB), Support Vector Machine (SVM), and Maximum Entropy (ME) [5]. Although the effect of these methods is better than when using lexicon-based sentiment analysis methods, their generalization ability is poor and their accuracy decreases rapidly with increasing data volume and the emergence of various new language scenarios.

In recent years, the application of deep learning models to solve sentiment analysis problems has gradually become the norm with the rapid development of deep learning [6]. By training deep learning networks and automatically adjusting their weighting parameters, large-scale data can be processed efficiently with superior accuracy and time and effort savings. The task of text sentiment analysis can be efficiently accomplished using deep learning networks. The advantages of using deep learning models for the sentiment analysis of microblogs are obvious when compared with earlier sentiment analysis methods. However, due to the characteristics of microblog texts, which require models with a high level of semantic understanding, there are still some shortcomings in existing models with respect to accomplishing the task of sentiment analysis in microblogs.

First, the accuracy of the text coding of most existing models needs to be improved. Most of the commonly used text representation methods focus only on character-level or word-level features. In recent years, models that combine character-level and word-level text features have appeared one after another; for example, Jin et al. [7] used CNN (Convolutional Neural Network) for feature extraction of word-level text after completing word text embedding, a Bi-GRU (Bidirectional Gated Recurrent Unit) combined with the attention mechanism for feature extraction of word-level text, and finally fused the two to enhance the effect. Zhang [8] et al. adopted dynamic encoding for character-level text and static encoding for word-level text, using a dual-channel network to complete a named entity recognition task, etc. The above studies prove that joint character- and word-level text features can indeed improve model performance, but the dynamic encoding is not used for all features when they are encoded. Although the existing deep learning models have become more comprehensive in obtaining text information, the accuracy of text coding still needs to be improved.

Second, the existing models use a combination of CNN and RNN (Recurrent Neural Network) to make full use of the local and global semantic information in the text, which is time-consuming. CNN has excellent local learning ability but cannot obtain global semantic information. RNN can capture the global information of the input sequence, but local key feature acquisition ability is poor. To solve this problem, Hu et al. [9] used multi-channel modeling to combine CNN with Bi-LSTM to improve the effect of text classification in medical records. Yan [10] et al. fused CNN with BiGRU as a way to achieve sentiment analysis of text. Although the above models can obtain both the local and global semantic information in a text, they have a high time cost due to the complex structure of RNN models.

Third, existing models have typically adopted pooling strategies to reduce the parameters and speed up model convergence, but the ordinary pooling approach has the drawback of losing key text information. Max-pooling [11] and average-pooling [12] are the two most common pooling approaches in the existing deep learning models. Max-pooling filters the salient features in the text and discards other non-significant features, while average-pooling averages the text features. However, neither the salient nor average features are the most key features of the text. Using the above pooling approaches will result in the loss of some important information, and are not able to achieve dynamic optimization of features.

Fourth, existing character–word association feature models mostly use direct splicing or simple summation to integrate character-level and word-level information, making it difficult to achieve effective utilization and fusion of character–word features. Tong et al. [13] and Chen et al. [14] used simple splicing to fuse the word information of text. The word-level information and character-level information contained in a text are complementary to each other, and the importance of both varies in different contexts, so existing methods of combining character-level and word-level information are not appropriate.

To improve the effectiveness of microblog sentiment analysis, we propose a microblog sentiment analysis model (DCCMM) based on Dynamic Character and Word Encoding, CNN, Cross-Channel Feature fusion, Multi-Head Self-Attention Pooling, and Multi-Granularity Feature Interaction Fusion.

The innovations of this paper are as follows.

1. To cope with the frequent occurrence of multiple meanings and the limited information contained in microblog text, and to solve the problems of the inaccuracy and incompleteness of the information presented by existing text coding methods, we propose a dual-stream dynamic coding method for characters and words, using WOBERT Plus and ALBERT to achieve word-level dynamic embedding and character-level dynamic embedding of text, respectively, so that the model is able to obtain more comprehensive and accurate semantic information.
2. To cope with the irregularity of microblog text and its high requirement for model semantic understanding, and to solve the problem of the high time cost required for combining CNN and RNN to extract local and global semantic information, we use multi-scale convolutional kernels to extract local key features with different levels of granularity and propose the combination of cross-channel feature fusion and multi-head self-attention to extract global semantic information. The multi-scale convolutional kernel can reduce the adverse effects caused by the high degree of microblog irregularity, and the multi-head self-attention mechanism can realize parallel operations and reduce the training time of the model. The above algorithm extracts both the local key features and global semantic information in the text, which improves the semantic understanding of the model and reduces the time required for model convergence.
3. To cope with the limited information contained in the microblog text and its high requirements with respect to model semantic understanding, and to solve the problem of the loss of key features in the pooling layer employed by most existing models, we propose a new pooling method: multi-head self-attention pooling, so that the model can adaptively extract key features that affect the results of microblog sentiment analysis, instead of the salient or mean features. The model fully considers the contribution of each feature to the final result, in order to improve the effectiveness of the sentiment analysis of the model.
4. To cope with the short length and limited content of microblog data, and to solve the problem whereby existing models cannot effectively utilize and fuse textual information at the character and word levels, we propose the multi-granularity feature interaction fusion mechanism to enhance the interactivity between character-level and word-level information, thus achieving complementary character-level and word-level information, strengthening the proportion of important information, and filtering out noise, resulting in the output of higher quality joint character-level and word-level feature information. This can improve the model's perception of sentiment polarity, helping the model better cope with the limited information contained in microblog texts and improving the effectiveness of microblog sentiment analysis.

The rest of this article is organized as follows. Section 2 introduces the related key technologies. Section 3 gives an overview of our proposed method and describes the key content of our method. The experimental results and discussion are presented in Section 4. Finally, Section 5 concludes this article.

2. Related Key Technologies

2.1. Text Feature Representation Techniques

The Word2vec word embedding language model can be used to transform the text after word separation into static word vectors, and the distance between vectors represents the semantic relatedness between words. A major drawback of this model, however, is that it cannot take into account the influence of context on word vectors [15]. For this reason, dynamic word vectors were created. Among them, the most representative is the BERT [16] pre-training language model, proposed by combining the advantages of ElMo [17] and GPT [18], which realizes the encoding of text with dynamic vectors, and the generated vectors can be adjusted in time according to different contexts. It can effectively deal with the problem of "polysemy". However, the model can only dynamically encode word-level text, and the time and resources consumed for training are significant. In response to the above drawbacks of the BERT model, Lan et al. [19] proposed the ALBERT model, which applies embedded layer factorization and cross-layer parameter sharing to significantly reduce the number of parameters and replaces the NSP task in the BERT model with an SOP task to improve the model's understanding of the information in the text. To solve the problem of unregistered words, Liu et al. [20] proposed the RoBERTa model by encoding bytes. Cui et al. [21] proposed the BERT-wwm model by replacing the original masking only of sub-words with a full-word mask. Joshi et al. [22] designed and implemented a sub-word-level pre-training method and proposed the SpanBERT model by randomly adding masks to neighboring sub-words instead of masking individual words.

However, the above RoBERTa, BERT-wwm, and SpanBERT models were all models that were designed and improved based on the characteristics of English words. In 2020, Jianlin Su et al. [23] proposed the word-based Chinese pre-trained language model WOBERT; using Chinese words as the processing unit can reduce the length of the model sequence and increase the processing speed of the text model. In addition, Chinese words are rich in semantic information and low in uncertainty compared to characters. The complexity of the model was reduced, while not reducing its performance. In March 2021, based on the WOBERT model, the word list was reconstructed and the training method was improved, resulting in the proposed WOBERT Plus model, which further improved the coding accuracy of Chinese words [24]. WoBERT Plus modifies the previous word splitting operation when processing input text in Chinese word units. It adds a "pre-tokenizer" operation to avoid forcing Chinese characters to be separated by spaces in the previous splitting operation. Therefore, the WoBERT Plus model can experiment with dynamic encoding of Chinese word-level text. Therefore, facing the task of sentiment analysis of Chinese microblog speech, we use ALBERT to complete the character embedding of the text and WOBERT Plus to complete the word embedding of the text, and a dual-stream information dynamic coding method is adopted to extract comprehensive and accurate information.

2.2. Attention Mechanism

Attention mechanisms are widely used in various fields of artificial intelligence, and their principle is to improve a model's ability to capture important features by adjusting the weight parameters so that the model focuses on key features and automatically ignores noise present in the information [25]. Whether in speech recognition, sentiment analysis, image processing, or another task, attention mechanism has now become one of the most noteworthy core techniques of deep learning. Sangeetha et al. [26] used a multi-head attention layer to further adjust the encoding information of the text after the embedding layer, and then used LSTM to learn global semantic information from the text, improving the performance of the model in a student emotion recognition task. India et al. [27] proposed a new deep learning model by combining the multi-head self-attention mechanism with CNN to improve the effect of speech recognition. Fang et al. [28] fused the word embedding matrix with the word vector matrix, processed by the multi-head self-attention mechanism, followed by feature extraction using CNN to improve the recognition of fake news.

In recent years, graph convolutional networks have gradually been applied in the field of natural language processing with good results. YAO et al. [29] used graph convolutional networks for text classification and demonstrated that the model was still robust in the face of small amounts of data. Chi et al. [30] used graph convolutional networks to augment paraphrase generation.

However, Jiang et al. [31] found that graph convolutional networks could be said, to some extent, to be a special form of self-attention mechanism by means of reasoning and demonstration, and conducted comparative experiments on text classification tasks under the same conditions. The performance of the model using the self-attention mechanism was significantly better than that of the comparative model using graph convolutional networks, which proves that the performance of the self-attention mechanism is better than that of graph convolutional networks in the field of text classification. The self-attention mechanism adjusts the weight parameters by paying attention to the relationship between characters or words at different positions in the text sequence, to obtain an interactive representation of the sequence, as well as its global semantic information. Multi-head self-attention employs multiple self-attention modules in parallel, integrating information from different levels, resulting in more comprehensive and diverse data features being obtained. The microblog sentiment analysis problem is essentially a text classification problem, so we decided to improve the multi-head self-attention mechanism to complete the microblog sentiment analysis task.

3. DCCMM Microblog Sentiment Analysis Model

As shown in Figure 1, the DCCMM model consists of the dual-stream dynamic coding layer, the convolution layer, the cross-channel feature fusion layer, the multi-head self-attention pooling layer, the multi-granularity feature interaction fusion layer, and the Softmax classification and output layer.

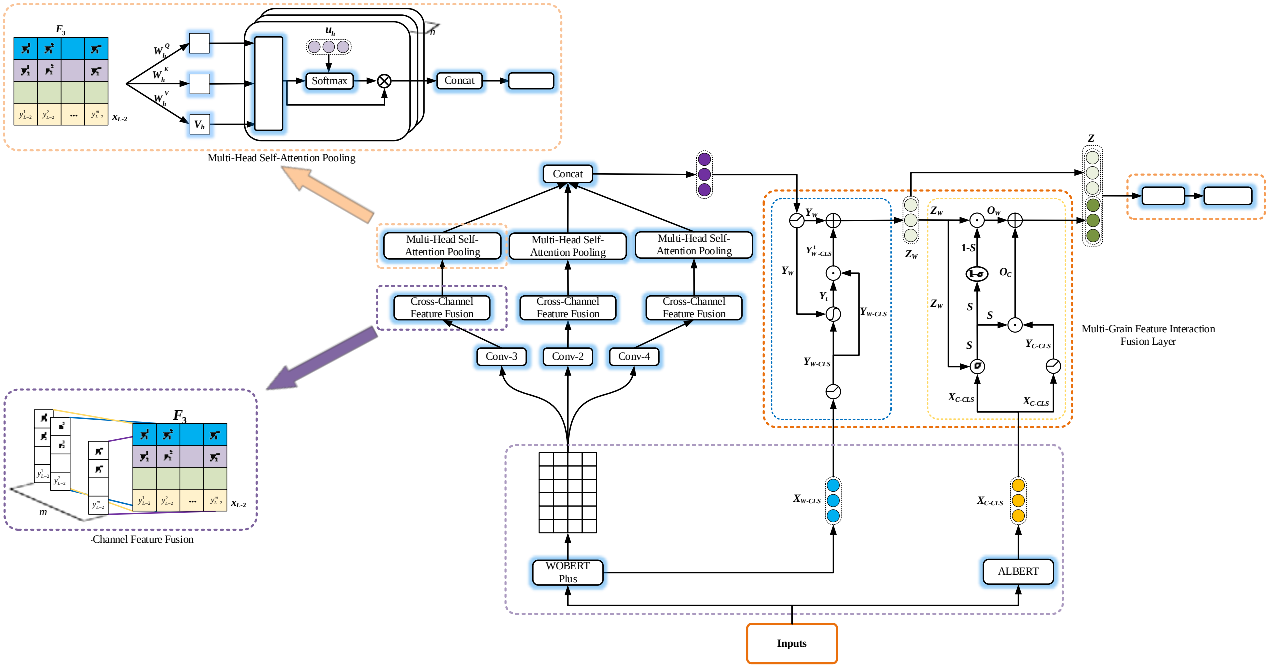

Figure 1. DCCMM model diagram.

(1) Dual-Stream Dynamic Coding Layer: We propose a dual-stream dynamic encoding method for characters and words to deal with the frequent occurrence of polysemy and the limited information contained in microblog remarks, and to solve the problems of

inaccurate text encoding and incomplete information. WOBERT Plus and ALBERT are used to implement word-level dynamic embedding and character-level dynamic embedding of text, respectively, so that the model can obtain more comprehensive and accurate semantic information. This encoding method is more suitable for sentiment analysis of microblog speeches than the previous methods of static encoding or the dynamic encoding of only a single text feature.

(2) Convolution Layer: We use multi-scale convolution kernels to perform convolution operations on adjacent text features to deal with the high degree of irregularity in microblog data. Multi-scale convolution kernels can extract local key information with different granularities and improve the model's ability to perceive emotional polarity.

(3) Cross-Channel Feature Fusion Layer: We propose a cross-channel feature fusion mechanism to ensure the integrity of the textual information extracted by the convolutional layer, while enhancing the expressiveness of the local features of microblog speeches, so that the subsequent multi-head self-attention pooling layer learns more accurate global semantic information.

(4) Multi-Head Self-Attention Pooling Layer: We propose a new pooling method, multi-head self-attentive pooling, to cope with the limited information contained in microblog text and its requirement for a high level of semantic understanding in the model, and to solve the problem of the loss of key features inherent to most existing model pooling layers. The previous operations of the multi-head self-attention pooling layer in our model only obtain the local information of the text, and do not extract global dependency information. In addition, both average-pooling and max-pooling suffer from the loss of key information, which is not conducive to the acquisition of sentiment information from the microblog text. Our proposed multi-head self-attention pooling mechanism achieves feature optimization and parameter reduction while extracting the global features of the text, and can adaptively extract key features that affect the classification results of the model, fully considering the contribution of each feature to the final results.

(5) Multi-Granularity Feature Interaction Fusion Layer: We propose a multi-granularity feature interaction fusion mechanism to cope with the short length and limited content of microblog data, and to solve the problem whereby it is difficult for existing models to effectively utilize and fuse character and word information in the text. In this model, the information granularity of the extracted features varies among channels, and it is not desirable to use simple summation or vector splicing. The multi-granularity feature interaction fusion mechanism can enhance the interactivity between character and word information, realize the complementarity between character-level and word-level information, and strengthen the proportion of important information, while filtering out noisy information, thus resulting in the output of higher quality joint character–word feature information, in order to improve the model's ability to perceive emotional polarity and better cope with the limited amount of information characteristic of microblog text.

(6) Softmax Classification and Output Layer: The joint feature vector outputs from the multi-granularity feature interaction fusion layer are linearly varied and fed into the Softmax function to derive the results of the sentiment analysis.

3.1. Dual-Stream Dynamic Coding Layer

Chinese characters and words both contain important semantic information, and there are differences in the information they contain. Therefore, it is necessary to extract both character-level and word-level information when performing microblog sentiment analysis, to achieve complementarity between them and improve model performance. The character-based pre-trained language model does not rely on word separation tools, thus avoiding the problem of inaccurate word boundary delineation, does not include semantic noise caused by improper word separation, and does not need to consider the impact of unregistered words on downstream tasks; the word-based model can make the processing sequence shorter and speed up the model's operation. In addition, the semantic uncertainty of word meaning is lower than that of character meaning, and its semantic information is richer, which helps improve the accuracy of the text classification task. Therefore, we combine the advantages of character and word pre-training models, and propose a dual-stream dynamic encoding mechanism that uses WOBERT Plus and ALBERT to complete text dynamic word-level encoding and character-level encoding, respectively. Both the WOBERT Plus and ALBERT models need to add a CLS vector when encoding text. CLS does not represent the information of a single character or word, but contains the semantics of the entire sentence.

(1) WOBERT Plus word-level dynamic encoding

The WOBERT Plus model is used to train word vectors on the input text, and its output contains two parts: the CLS vector and the word vector matrix.

Here is the word vector matrix $\boldsymbol{H} \in \mathbb{R}^{L\times d}$, $\boldsymbol{H} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_L]^T$, $\boldsymbol{H}_{i:j} = [\boldsymbol{w}_i, \boldsymbol{w}_{i+1}, \ldots, \boldsymbol{w}_j]^T$ $(j \geq i)$, where L is the number of words in the input text, d is the word vector dimension ($d = 768$), and each row of the word vector matrix represents a vector of individual word maps.

(2) ALBERT character-level dynamic encoding

The ALBERT model is used to train character vectors on the input text, and the CLS vectors imply the semantics of the whole sentence. To reduce the model parameters and complexity, only the encoded CLS vectors are output.

The CLS vector $\boldsymbol{X}_{\boldsymbol{C-CLS}} \in \mathbb{R}^d$, d is the word vector dimension.

3.2. Convolution Layer

To capture word-level data information at different granularities, the DCCMM model uses three different sizes of convolutional kernels (2, 3, and 4) to sense the text vector matrix $\boldsymbol{H_W}$, so that the model can obtain more comprehensive information. The number of convolutional kernels of each size is 256.

$$y_i^m = f(\boldsymbol{W}_k^m \cdot \boldsymbol{H}_{i:i+k-1} + b_i^m) \tag{1}$$

$$\boldsymbol{Y}_k^m = (y_1^m, y_2^m, y_3^m, \ldots, y_{n-k+1}^m)^T \tag{2}$$

where m is the convolution kernel number, i is the sliding window number, k represents the convolution size, b_i^m represents the bias term, $\boldsymbol{W}_k^m$ represents the weight matrix corresponding to the convolution kernel, f is the activation function, y_i^m represents the result obtained after the convolution operation, and $\boldsymbol{Y}_k^m$ is the feature column vector obtained by stitching the results of the convolution operation.

3.3. Cross-Channel Feature Fusion Layer

The information fusion operation is performed on the high-level features located in multiple channels after the convolution operation, and then the high-level semantic feature matrix $\boldsymbol{F}_k$ is obtained.

$$\boldsymbol{F}_k = Concat(\boldsymbol{Y}_k^1, \boldsymbol{Y}_k^2, \boldsymbol{Y}_k^3, \ldots \boldsymbol{Y}_k^m) \tag{3}$$

where *Concat* represents the vector stitching operation, k represents the number of the matrix, and m is the convolution kernel number.

Taking the semantic feature matrix obtained by a convolution kernel with a size of 3 as an example, the cross-channel feature fusion layer is further explained, as shown in the model diagram in Figure 1:

$$\boldsymbol{F}_3^T = [\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \ldots, \boldsymbol{x}_l^T, \ldots \boldsymbol{x}_{L_1-2}^T] \tag{4}$$

$$\begin{aligned} &\boldsymbol{x}_1 = (y_1^1, y_1^2, \ldots, y_1^m) \\ &\boldsymbol{x}_l = (y_l^1, y_l^2, \ldots, y_l^m) \\ &\ldots \\ &\boldsymbol{x}_{L_1-2} = (y_{L-2}^1, y_{L-2}^2, \ldots, y_{L-2}^m) \end{aligned} \tag{5}$$

According to the convolution formula, feature values $y_l^1 \sim y_l^m$ are obtained by convolving different convolution kernels of the same size (in this case, the convolution kernel size is 3) with the local text word vectors. Since the convolutional kernel weight matrices are different from one another, the semantic feature values $y_l^1 \sim y_l^m$ represent high-level features of the same region of text captured from different perspectives, i.e., the textual information contained in the same three words. Therefore, the row vectors of matrix $\boldsymbol{F}_3$ cover the high-level information of the same region of text extracted from different angles by convolutional operations. The same is true for the remaining semantic feature matrices $\boldsymbol{F}_2$ and $\boldsymbol{F}_4$, except that they have different degrees of information granularity. The cross-channel feature fusion mechanism can differentiate the original word vector matrix into three high-level semantic information matrices with different granularities of semantic information, and the three matrices cover the local key information extracted from different angles with different granularities, enhancing the local text feature representation, and thus enabling the multi-head attention pooling layer to learn more word-level text information and ensure the extraction of more accurate global dependent information.

3.4. Multi-Head Self-Attention Pooling Layer

Convolution and cross-channel feature fusion operations only mine and integrate local information and do not employ contextual semantic information. Global semantic information is essential in the face of microblogging speech, which requires a high level of semantic understanding in the model. In addition, the semantic information matrix contains some redundant information and noisy data, so it is necessary to optimize the features, filter out the key features, and reduce the number of parameters.

However, the commonly used pooling approach suffers from information loss, which is not conducive to the operation of a model for sentiment analysis of microblog speech. To address these problems, we propose a multi-head self-attention pooling mechanism based on the multi-head self-attention mechanism. The multi-head self-attention pooling mechanism can establish connections between the isolated rows of the matrix and pass information among them to obtain the long-range dependencies of text, enabling the model to extract key text features from multiple subspaces separately by adjusting the weight parameters.

The multi-head self-attention pooling mechanism can be divided into two steps: multi-head self-attention operations and multi-head pooling operations. An example of the multi-head self-attention pooling mechanism is illustrated by the operation performed in the h-th subspace.

First, the multi-head self-attention algorithm is executed. Initialize the parameter matrices $\boldsymbol{W}_h^Q$, $\boldsymbol{W}_h^K$, $\boldsymbol{W}_h^V$, and perform the self-attention operation on the semantic information matrix $\boldsymbol{F}$, so that each of the row vectors of the matrix $\boldsymbol{F}$ passes information to one another and establishes the connection, and thus the global semantic information is included in each row vector, and the matrix $\boldsymbol{head}_h$ is obtained. Then, the multi-head pooling algorithm we proposed on top of the multi-head self-attention mechanism is executed. Initialize the parameter vector $\boldsymbol{u}_h$, perform matrix multiplication operation between $\boldsymbol{head}_h$ and $\boldsymbol{u}_h$ to obtain the joint vector $\boldsymbol{O}_h$ of the two, calculate the probability distribution vector $\boldsymbol{\alpha}_h$ between each row vector using the Softmax function, and finally multiply $\boldsymbol{\alpha}_h^T$ with the matrix $\boldsymbol{head}_h$ to increase the weight of key information by adjusting the weights to achieve feature optimization and parameter reduction in order to obtain the vector $\boldsymbol{head}'_h$. For the model to learn information from different representation subspaces, different information is obtained from each subspace by repeating the operations several times in parallel to extract richer data features. Finally, the information captured in each subspace is stitched together to obtain the result of multi-head self-attention pooling $\boldsymbol{M}$.

$$\boldsymbol{Q_h} = \boldsymbol{F}\boldsymbol{W}_h^{\boldsymbol{Q}} \tag{6}$$

$$\boldsymbol{K_h} = \boldsymbol{F}\boldsymbol{W}_h^{\boldsymbol{K}} \tag{7}$$

$$\boldsymbol{V_h} = \boldsymbol{F}\boldsymbol{W}_h^{\boldsymbol{V}} \tag{8}$$

$$\boldsymbol{head_h} = Attention(\boldsymbol{Q_h}, \boldsymbol{K_h}, \boldsymbol{V_h}) = softmax\left(\frac{\boldsymbol{Q_h}\boldsymbol{K}_h^{\boldsymbol{T}}}{\sqrt{D}}\right)\boldsymbol{V_h} \tag{9}$$

$$\boldsymbol{O_h} = \boldsymbol{head_h} \times \boldsymbol{u_h} \tag{10}$$

$$\boldsymbol{\alpha_h} = softmax(\boldsymbol{O_h}) \tag{11}$$

$$\boldsymbol{head}'_{\boldsymbol{h}} = \boldsymbol{\alpha}_h^{\boldsymbol{T}} \times \boldsymbol{head_h} \tag{12}$$

$$\boldsymbol{M} = Concat(\boldsymbol{head}'_{\boldsymbol{1}}, \boldsymbol{head}'_{\boldsymbol{2}}, \ldots, \boldsymbol{head}'_{\boldsymbol{n}}) \tag{13}$$

where $\boldsymbol{W}_h^{\boldsymbol{Q}}$, $\boldsymbol{W}_h^{\boldsymbol{K}}$, and $\boldsymbol{W}_h^{\boldsymbol{V}}$ are the linear mapping weight matrices of the h-th subspace of matrix $\boldsymbol{F}$, $\boldsymbol{head_h}$ is the feature matrix generated by the h-th subspace through the self-attentive mechanism, $\boldsymbol{u_h}$ is the dynamically updated parameter vector within the subspace, $\boldsymbol{O_h}$ is the joint vector generated by the joint action of $\boldsymbol{head_h}$ and $\boldsymbol{u_h}$, $\boldsymbol{\alpha_h}$ is the weight vector, and $\boldsymbol{head}'_{\boldsymbol{h}}$ is the vector obtained from the feature matrix $\boldsymbol{head_h}$ after the condensation of key features. n represents the number of subspaces in the multi-head self-attention pooling mechanism. D denotes the dimensionality of each high-level semantic vector, and $\boldsymbol{M}$ is the feature resulting from the information collocation of n subspaces.

From the model diagram in Figure 1, it can be seen that three multi-head self-attention layer channels are operating in parallel. The vectors generated by the three channels are named $\boldsymbol{M_2}$, $\boldsymbol{M_3}$, and $\boldsymbol{M_4}$ according to the different sizes of the convolutional kernels, and the vectors generated by the three channels are stitched together to obtain the word-level semantic information $\boldsymbol{X_W}$.

$$\boldsymbol{X_W} = Concat(\boldsymbol{M_2}, \boldsymbol{M_3}, \boldsymbol{M_4}) \tag{14}$$

3.5. Multi-Granularity Feature Interaction Fusion Layer

The information granularity of the CLS tag vector $\boldsymbol{X_{W-CLS}}$ output by WOBERT Plus, the CLS tag vector $\boldsymbol{X_{C-CLS}}$ output by ALBERT, and the multi-head self-attention pooling layer output vector $\boldsymbol{X_W}$ are not the same; $\boldsymbol{X_{W-CLS}}$ and $\boldsymbol{X_W}$ belong to the word-level semantic information and $\boldsymbol{X_{C-CLS}}$ belongs to the character-level semantic information. Different information granularities have different levels of importance in different contexts and produce different effects in the classification results. In addition, factors such as improper word separation may introduce noise into the word-level text information, and the character-level semantic information may present the problem of containing insufficient semantic information, necessitating further processing of the information. Therefore, it is not reasonable to process character-level and word-level semantic information by means of direct splicing or simple summation. To enhance the interactivity between character- and word-level information, realize the complementarity between character-level and word-level information, and strengthen the weight of important information while filtering out noise, thus resulting in the output of more effective joint character and word feature information, we propose a multi-granularity feature interaction fusion mechanism in this paper. The multi-granularity feature interaction fusion mechanism is shown in Figure 2. The multi-granularity feature interaction fusion mechanism mainly consists of two parts: word-level semantic refinement and word-level and character-level semantic fusion.

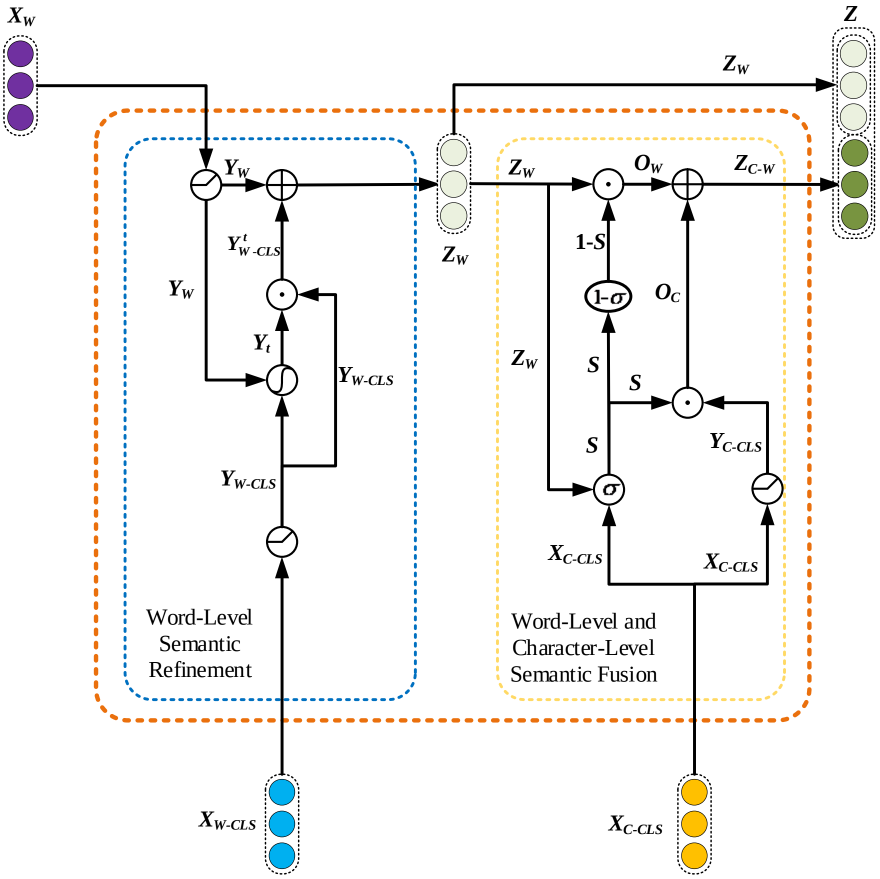

Figure 2. Multi-granularity feature interaction fusion mechanism diagram.

(1) Word-Level Semantic Refinement

Word-level semantic feature refinement is used to correct and remove noise from the information to achieve an effective fusion of word-level granularity information $\boldsymbol{X_{W-CLS}}$ and $\boldsymbol{X_W}$.

$$\boldsymbol{Y_W} = \mathrm{ReLU}(\boldsymbol{W_1 X_W} + \boldsymbol{b_1}) \tag{15}$$

$$\boldsymbol{Y_{W-CLS}} = \mathrm{ReLU}(\boldsymbol{W_2 X_{W-CLS}} + \boldsymbol{b_2}) \tag{16}$$

$$\boldsymbol{Y_t} = \tanh(\boldsymbol{Y_W} + \boldsymbol{Y_{W-CLS}} + \boldsymbol{b_3}) \tag{17}$$

$$\boldsymbol{Y^t_{W-CLS}} = \boldsymbol{Y_{W-CLS}} \odot \boldsymbol{Y_t} \tag{18}$$

$$\boldsymbol{Z_W} = \boldsymbol{Y^t_{W-CLS}} + \boldsymbol{Y_W} \tag{19}$$

where $\boldsymbol{W_1}$, $\boldsymbol{W_2}$ are weight matrices, $\boldsymbol{b_1}$, $\boldsymbol{b_2}$, $\boldsymbol{b_3}$ are bias terms, and $\boldsymbol{Y_{W-CLS}}$ and $\boldsymbol{Y_W}$ are word-level semantic vectors obtained after feature condensation and noise filtering for $\boldsymbol{X_{W-CLS}}$ and $\boldsymbol{X_W}$, respectively. $\boldsymbol{Y_t}$ is the weight vector generated by combining $\boldsymbol{Y_{W-CLS}}$ and $\boldsymbol{Y_W}$. $\boldsymbol{Y^t_{W-CLS}}$ is the vector of moderation factors generated using $\boldsymbol{Y_{W-CLS}}$ and $\boldsymbol{Y_t}$. Sign $\odot$ denotes Hadamard product. $\boldsymbol{Z_W}$ is the vector after implementing key feature correction for $\boldsymbol{Y_W}$, which effectively incorporates word-level semantic information.

Since the ReLU function can map negative values to zero, Equation (15) uses the ReLU function to generate the vector $\boldsymbol{Y_W}$ by initially filtering the noisy regions in the word-level vector $\boldsymbol{X_W}$.

Similarly, Equation (16) uses the ReLU function to filter the noisy region in $\boldsymbol{X_{W-CLS}}$ to generate $\boldsymbol{Y_{W-CLS}}$ so that the modulation factor vector $\boldsymbol{Y^t_{W-CLS}}$ does not contain invalid and disruptive information that could lead to noisy information in the vector $\boldsymbol{Z_W}$.

Since the tanh function maps the eigenvalues to the interval (−1,1), the range it can regulate is broader and more flexible compared to that available to the Sigmoid function. Equation (17) generates the weight vector $\boldsymbol{Y_t}$ using the tanh function for $\boldsymbol{Y_{W-CLS}}$ and $\boldsymbol{Y_W}$.

Equation (18) performs the Hadamard product operation on $\boldsymbol{Y_t}$ and the filtered feature vector $\boldsymbol{Y_{W-CLS}}$. The features contained in $\boldsymbol{Y_{W-CLS}}$ are assigned positive and negative weights according to their roles in generating the modulation factor vector $\boldsymbol{Y^t_{W-CLS}}$. If the element value in $\boldsymbol{Y^t_{W-CLS}}$ is 0, this means that the feature values in $\boldsymbol{Y_W}$ do not need to be corrected. If it is positive, it plays an enhancing and complementary role to the feature values in $\boldsymbol{Y_W}$, and if it is negative, it plays an attenuating role to the feature values in $\boldsymbol{Y_W}$.

Equation (19) adds the moderation factor vector $\boldsymbol{Y^t_{W-CLS}}$ to the feature values contained in the word-level semantic information vector $\boldsymbol{Y_W}$ bit by bit, eliminates the noisy features in $\boldsymbol{Y_W}$, and corrects the feature values in the key regions. Using this approach causes $\boldsymbol{Z_W}$ effectively to fuse the word-level information contained in $\boldsymbol{X_{W-CLS}}$ and $\boldsymbol{X_W}$ in order to achieve word-level semantic feature refinement.

(2) Word-Level and Character-Level Semantic Fusion

Although character-level semantic information does not involve noise caused by improper word separation or poor word list quality, its semantic information is not as rich as word-level information and may present the problem of there being insufficient semantic information; therefore, it is necessary to further process its information. In addition, when performing multi-granularity information fusion, the information with different granularities will have different importance for the sentiment analysis results. Therefore, when using multi-granularity information for sentiment analysis, it is necessary to distinguish the importance of word-level information from that of character-level information for the sentiment analysis of the target sequence and enhance the weight of important information to obtain more effective joint sentiment features across granularities.

$\boldsymbol{Z_W}$ is the word-level granularity information after refinement, and $\boldsymbol{X_{C-CLS}}$ is the character-level granularity information; the two have different information granularity, so the character-level and word-level semantic fusion operation should be adopted differently from the word-level semantic feature refinement to obtain the interaction vector containing both character-level and word-level information. Character-level and word-level semantic fusion is shown in Equations (20)–(24).

$$\boldsymbol{S} = \mathrm{Sigmoid}(\boldsymbol{W_3 X_{C-CLS}} + \boldsymbol{W_4 Z_W} + \boldsymbol{b_4}) \tag{20}$$

$$\boldsymbol{Y_{C-CLS}} = \mathrm{ReLU}(\boldsymbol{W_5 X_{C-CLS}} + \boldsymbol{b_5}) \tag{21}$$

$$\boldsymbol{O_W} = \boldsymbol{Z_W} \odot (1 - \boldsymbol{S}) \tag{22}$$

$$\boldsymbol{O_C} = \boldsymbol{Y_{C-CLS}} \odot \boldsymbol{S} \tag{23}$$

$$\boldsymbol{Z_{C-W}} = \boldsymbol{O_W} + \boldsymbol{O_C} \tag{24}$$

where $\boldsymbol{W_3}$, $\boldsymbol{W_4}$, and $\boldsymbol{W_5}$ are weight matrices, $\boldsymbol{b_4}$ and $\boldsymbol{b_5}$ are bias terms, $\boldsymbol{S}$ is the weight vector generated jointly by character-level semantic information $\boldsymbol{X_{C-CLS}}$ and word-level semantic information $\boldsymbol{Z_W}$, $\boldsymbol{Y_{C-CLS}}$ is the character-level semantic vector generated by $\boldsymbol{X_{C-CLS}}$ after adaptive fine-tuning and feature condensation, $\odot$ is the Hadamard product, $\boldsymbol{O_W}$ is the word-level significant feature, $\boldsymbol{O_C}$ is the character-level significant feature, and $\boldsymbol{Z_{C-W}}$ is the interaction vector output by the dynamic fusion mechanism of character-level and word-level semantics.

Equation (20) is the weight vector $\boldsymbol{S}$ generated by $\boldsymbol{Y_{C-CLS}}$ and $\boldsymbol{Z_W}$ under the action of the Sigmoid function. $\boldsymbol{S}$ can adaptively adjust the output ratio of the two according to the different importance of character-level and word-level information, and conditionally calculate the interaction vector between different granularity information.

Equation (21) is the character-level semantic information vector $\boldsymbol{X_{C-CLS}}$ subjected to feature condensation and adaptive fine-tuning to obtain the character-level semantic vector $\boldsymbol{Y_{C-CLS}}$.

Equation (22) is the Hadamard product operation of $\boldsymbol{Z_W}$ and the weight vector $\boldsymbol{S}$ to obtain the word-level conditional vector $\boldsymbol{O_W}$.

Equation (23) is the Hadamard product operation of $\boldsymbol{Y_{C-CLS}}$ and the weight vector $1-\boldsymbol{S}$ to obtain the character-level conditional vector $\boldsymbol{O_C}$.

Equation (24) adds the word-level conditional vector $\boldsymbol{O_W}$ with the character-level conditional vector $\boldsymbol{O_C}$ to achieve a fusion between different granularity features, and generates an interaction vector $\boldsymbol{Z_{C-W}}$ that contains both character- and word-level semantic information.

Finally, the multi-grain feature interaction fusion layer stitches the word-level semantic information $\boldsymbol{Z_W}$ together with the character-level and word-level semantic feature interaction information $\boldsymbol{Z_{C-W}}$.

$$\boldsymbol{Z} = Concat(\boldsymbol{Z_W}, \boldsymbol{Z_{C-W}}) \tag{25}$$

The final classification results are obtained by feeding vector $\boldsymbol{Z}$, which combines character-level and word-level information, into the Softmax classifier.

4. Experiment and Result Analysis

4.1. Dataset

We selected two publicly available datasets for the experiments.

Dataset 1 is the microblog comment corpus weibo_senti_100k (https://github.com/SophonPlus/ChineseNlpCorpus/blob/master/datasets/weibo_senti_100k/intro.ipynb accessed on 24 June 2022) from Sina Weibo, which contains sentiment annotation data. All the data performed sentiment annotation. The dataset contains approximately 120,000 microblog texts, including approximately 60,000 positive and 60,000 negative sentiment texts. We divided the dataset into a training set, a validation set, and a test set according to the ratio of 8:1:1. The information of the dataset is shown in Table 1.

Table 1. Dataset information.

Data Item	Weibo_Senti_100k	SMP2020-EWECT
Number of emotion categories	2	6
Total amount of data	119,988	48,374
Training set	95,990	38,699
Validation set	11,999	-
Test set	11,999	9675

Dataset 2 is the SMP2020-EWECT dataset (https://github.com/BrownSweater/BERT_SMP2020-EWECT/tree/main/data/raw accessed on 24 June 2022), which contains more than 48,000 microblog sentiment data on two topics: “Usual” and “Virus”. The microblog content for the “Usual “theme is not topic-specific and covers a wide range of topics. The “Virus” theme is related to COVID-19, the microblog data, which were obtained by filtering using relevant keywords during the epidemic period. The data of each topic contained six categories of emotions: happy, angry, sad, fear, surprise, and neutral. We merged and organized the original dataset and divided it into a training set and a test set according to the ratio of 4:1. The information on the dataset is shown in Table 1.

4.2. Experimental Parameter and Environment Settings

The convolution kernel sizes of the convolution layer are 2, 3, and 4, respectively. The number of convolution kernels of each size is 256. The number of heads in the multi-head attention mechanism is 8. The batch size is 16. The epoch is 10. The learning rate is 1×10^{-5}, and dropout is 0.3. The experimental parameter setting is shown in Table 2. The experimental environment is shown in Table 3.

Table 2. Experimental parameter setting.

Parameter Name	Parameter Value
Convolution kernel size	2, 3, 4
Number of convolution kernel	256
Number of head	8
Epoch	10
Batch size	16
Dropout	0.3
Learning rate	1×10^{-5}

Table 3. Experimental environment.

Experimental Environment	Experimental Setting
Operating System	Windows 10
Programming language	Python 3.7.6
Deep learning framework	Pytorch 1.11.0
Graphics card	NVIDIA Tesla V100

4.3. Comparison of Experimental Results and Analysis

To evaluate the comprehensive performance of the models, we use the accuracy and F1 score as the evaluation metrics of the experiments. Mainstream deep learning models and the latest sentiment analysis models corresponding to each dataset were set as the comparison models for comparison with the DCCMM model. A comparison of the experimental results is shown in Tables 4 and 5.

Table 4. Performance comparison of different models on the weibo_senti_100k dataset.

Model	ACC (%)	F1 (%)
TextCNN	95.12	95.11
Bi-LSTM	95.38	95.39
RCNN	95.73	95.75
Bi-LSTM-Attention	95.90	95.91
ALBERT	97.26	97.25
WOBERT Plus	97.12	97.08
MDMLSM	93.19	—
ACL-RoBERTa-CNN	96.82	95.26
CTBERT	96.78	97.44
CBMA	97.65	97.51
DCCMM	**98.49**	**98.52**

Table 5. Performance comparison of different models on the SMP2020-EWECT dataset.

Model	ACC (%)	F1 (%)
TextCNN	75.80	69.87
Bi-LSTM	75.65	69.09
BiLSTM-Attention	76.03	70.36
RCNN	74.59	68.14
ALBERT	75.68	70.08
WOBERT Plus	78.06	73.61
BERT- Bi-LSTM-Attention	77.36	71.29
BERT-HAN	78.77	72.63
BERT + CFEBLS	78.85	73.89
DCCMM	**80.07**	**75.49**

(1) TextCNN [32]: Local key features are extracted by convolutional operation, and obvious features are filtered by max-pooling. Then, this information is used for classification.
(2) Bi-LSTM [33]: A bidirectional LSTM is used to extract contextual semantic and sequential information from the text.
(3) Bi-LSTM-Attention [34]: Based on the above Bi-LSTM model, the attention mechanism is added to enable the model to adjust the weight parameters according to the different levels of data importance and increase the weight of key information.
(4) RCNN [35]: Global semantic information is first extracted using Bi-LSTM, and obvious features are extracted using max-pooling. Then, this information is used for classification.
(5) ALBERT [19]: The feature vector corresponding to the position of the ALBERT model (CLS) is used for sentence representation.
(6) WOBERT Plus [24]: The feature vector corresponding to the position of the WOBERT Plus model [CLS] is used for sentence representation.
(7) BERT-Bi-LSTM-Attention [36]: Dynamic character-level vectors are generated using BERT, followed by the Bi-LSTM network to extract features, and then feature optimization is performed using the attention mechanism.
(8) BERT-HAN [37]: The model proposed in [37].
(9) BERT + CFEBLS [38]: The model proposed in [38].
(10) MDMLSM [39]: The model proposed in [39].
(11) ACL-RoBERTa-CNN [40]: The model proposed in [40].
(12) CBMA [41]: The model proposed in [41].
(13) CTBERT [42]: The model proposed in [42].
(14) DCCMM: The model proposed in this paper.

On the basis of a comparison of the experimental results, it can be seen that the DCCMM model achieves the highest accuracy and F1 values on each dataset, outperforming the mainstream deep learning models and the latest sentiment analysis models. On the weibo_senti_100k dataset, the accuracy and F1 values of the DCCMM model improve by 0.84% and 1.01%, respectively, compared to the best-performing comparison model. On the SMP2020-EWECT dataset, the accuracy and F1 values of the DCCMM model improve by 1.22% and 1.80%, respectively, compared with the experimental results of the best-performing comparison model. This fully verifies the advanced and effective performance of the DCCMM model for the microblog sentiment analysis task.

The DCCMM model achieves dynamic coding at both the character level and the word level in the coding layer, integrating character-level and word-level information, and extracting more comprehensive information, making it better able to cope with the frequent occurrence of multiple meanings of words in microblog speech. Using the multi-scale convolutional kernel to extract local features in the text, DCCMM weakens the negative influence of irregularities in the language of microblog speech on the results of sentiment analysis. Cross-channel feature fusion and multi-head self-attention pooling are used to extract global semantic information and achieve key information filtering. Finally, the character-level and word-level semantic information is effectively fused and condensed through a multi-granularity interactive fusion mechanism. Therefore, DCCMM has a strong sentiment analysis capability and has an excellent ability to cope with the task of microblog remarks sentiment analysis.

4.4. Ablation Experiment Results and Analysis

4.4.1. Ablation Experiment of Multi-Granularity Feature Interaction Fusion Mechanism

To verify the effect of the multi-granularity feature interaction fusion mechanism on the model performance in DCCMM, four sets of ablation experiments were conducted in this paper. Except for the change in the multi-granularity interaction fusion module, the other model parameters remained unchanged. The results are shown in Table 6. The specific settings of each experiment are as follows.

Table 6. Ablation experimental model performance comparison.

Experiment Number	**Weibo_Senti_100k**		**SMP2020-EWECT**	
	ACC (%)	**F1 (%)**	**ACC (%)**	**F1 (%)**
Ablation experiment 1	97.26	97.25	75.68	70.08
Ablation experiment 2	97.12	97.08	78.06	73.61
Ablation experiment 3	97.33	97.32	78.63	74.21
Ablation experiment 4	97.71	97.70	79.02	74.45
DCCMM	**98.49**	**98.52**	**80.07**	**75.49**

Ablation experiment 1: Only the CLS vector in the ALBERT model is used to complete the microblog sentiment analysis task; i.e., the X_{C-CLS} vector is dimensionally transformed and fed directly into the Softmax classifier to obtain the experimental results.
Ablation experiment 2: Only the CLS vector in the WOBERT Plus model is used to complete the microblog sentiment analysis task, i.e., the X_{W-CLS} vector is dimensionally transformed and fed directly into the Softmax classifier to obtain the experimental results.
Ablation experiment 3: Only the word-level semantic information vector X_W is used to complete the microblog sentiment analysis task, i.e., the vector X_W is dimensionally transformed and fed directly into the Softmax classifier to obtain the experimental results.
Ablation experiment 4: The fusion of X_{C-CLS}, X_{W-CLS} and X_W is performed using direct splicing, without going through the multi-granularity feature interaction fusion mechanism, which is fed into the Softmax classifier to derive the experimental result.

The DCCMM model outperforms the other models in ablation experiments 1–3 on both datasets. On the dataset weibo_senti_100k, the accuracy and F1 score of the DCCMM model are improved by 1.23% and 1.27%, respectively, compared to the model in ablation experiment 1, by 1.37% and 1.44%, respectively, compared to the model in ablation experiment 2, and by 1.16% and 1.20%, respectively, compared to the model in ablation experiment 3.

On the dataset SMP2020-EWECT, the accuracy and F1 score of the DCCMM model are improved by 4.39% and 5.41%, respectively, compared to the model in ablation experiment 1, by 2.01% and 1.88%, respectively, compared to the model in ablation experiment 2, and by 1.44% and 1.28%, respectively, compared to the model in ablation experiment 3. The result shows that only using character-level or word-level semantic information to complete the sentiment analysis task is ineffective, and using both character-level and word-level semantic information in combination to complement each other can improve model performance.

The DCCMM model outperforms the model in ablation experiment 4 on both datasets. On the weibo_senti_100k dataset, the accuracy and F1 score of the DCCMM model are improved by 0.78% and 0.82%, respectively, compared to the model in ablation experiment 4. On the SMP2020-EWECT dataset, the accuracy and F1 score of the DCCMM model are improved by 1.05% and 1.04%, respectively, compared to the model in ablation experiment 4. This verifies the effectiveness of the proposed multi-granularity interactive fusion mechanism. Direct splicing of the character-level and word-level semantic information cannot realize the fusion between them effectively. The multi-granularity interactive fusion mechanism can enhance the interaction between character-level and word-level information, realize the complementarity between character-level and word-level information, strengthen the weight of important information, and can filter out noise at the same time, resulting in the output of more effective joint character–word feature information and improving model performance.

4.4.2. Multi-Head Self-Attention Pooling Ablation Experiment

To verify the effect of multi-head self-attention pooling on the model's performance, three sets of ablation experiments were conducted in this paper. Except for the change in the multi-head self-attention pooling layer, all model parameters remained unchanged. The results of the multi-head self-attention pooling ablation experiment are shown in Table 7.

Table 7. Ablation experimental model performance comparison.

Experiment Number	**Weibo_Senti_100k**		**SMP2020-EWECT**	
	ACC (%)	**F1 (%)**	**ACC (%)**	**F1 (%)**
Ablation experiment 1	97.41	97.31	78.88	74.12
Ablation experiment 2	97.48	97.48	78.71	73.77
Ablation experiment 3	97.56	97.56	79.03	74.53
DCCMM	**98.49**	**98.52**	**80.07**	**75.49**

Ablation experiment 1: The multi-head self-attention pooling layer is replaced with the max-pooling layer in the model.
Ablation experiment 2: The multi-head self-attention pooling layer is replaced with the average-pooling layer in the model.
Ablation experiment 3: The multi-head self-attention pooling layer is replaced with the self-attentive pooling layer in the model, i.e., all pooling operations are performed in only one subspace instead of in multiple subspaces.

The accuracy and F1 score of the DCCMM model are improved by 1.08% and 1.21%, respectively, compared to the model in ablation experiment 1 on the weibo_senti_100k dataset. The accuracy and F1 score are improved by 1.19% and 1.37% compared with the model in ablation experiment 1 on the SMP2020-EWECT dataset. This shows that max-pooling only filters the obvious features and loses some key information, verifying the effectiveness of the multi-head self-attention pooling mechanism proposed in this paper.

The accuracy and F1 values of the DCCMM model are improved by 1.01% and 1.04%, respectively, compared to the model in ablation experiment 2 on the weibo_senti_100k dataset. The accuracy and F1 values are improved by 1.36% and 1.72%, respectively, compared to the model in ablation experiment 2 on the SMP2020-EWECT dataset. Mean-pooling extracts the mean features, but the mean features are not necessarily important. It still cannot perform the targeted extraction of key region features, while multi-head self-attention pooling can enhance the weight of important features and achieve adaptive extraction, which is beneficial to the model's grasp of the sentiment tendency of microblog speech.

The accuracy and F1 score of the DCCMM model are improved by 0.93% and 0.96% compared with the model in ablation experiment 3 on the weibo_senti_100k dataset. The accuracy and F1 score on the SMP2020-EWECT dataset are improved by 1.04% and 0.96% compared with the model in ablation experiment 3. This verifies that, compared to single subspace self-attention pooling, multi-head self-attention pooling can extract important features in the text from different perspectives, thus improving the sentiment analysis capability of the model.

4.4.3. Multi-Head Self-Attention Ablation Experiment

Existing models use a combination of CNN and RNN to make full use of the local and global semantic information of text, but RNN has a high time cost due to the problem of its model structure. We use a combination of CNN and the multi-head self-attention mechanism to extract local and global semantic information of text, shortening the training time of the model. To verify the effect of the multi-head attention mechanism and RNN on the time cost of the model, we conducted four sets of ablation experiments using four common RNN models, GRU, LSTM, Bi-GRU, and Bi-LSTM. All model parameters were kept constant except for the change to the multi-head self-attention part of the models. Accuracy, F1 value, time used to train a single epoch of the model, and the number of epochs used for model convergence were used as evaluation metrics.

As shown in Table 8, the time used to train a single epoch and the number of epochs used for model convergence for the models in ablation experiments 1–4 are higher than those of the DCCMM model in this paper, but their model performance is slightly lower than that of the DCCMM model. It can be concluded that the extraction of global semantic information using the multi-head self-attention mechanism takes less time compared to the RNN model, without degradation in the model performance.

Table 8. Ablation Experimental Model Performance Comparison.

Experiment Number	Weibo_Senti_100k				SMP2020-EWECT			
	ACC (%)	F1 (%)	Time (min)	Epochs	ACC (%)	F1 (%)	TIME (min)	Epochs
Ablation experiment 1	98.32	98.32	66.32	4	79.41	74.45	26.57	5
Ablation experiment 2	98.25	98.26	67.53	4	79.52	74.77	27.38	6
Ablation experiment 3	98.36	98.34	66.92	4	79.56	74.83	26.48	6
Ablation experiment 4	98.31	98.34	67.35	5	79.68	74.98	27.20	7
DCCMM	**98.49**	**98.52**	**56.62**	**3**	**80.07**	**75.49**	**22.27**	**5**

5. Conclusions

Compared with existing models, the advantages of DCCMM model are as follows. The DCCMM model achieves dynamic coding at both the character level and the word level in the coding layer, integrating character-level and word-level information, and extracting more comprehensive information, making it better able to cope with the frequent occurrence of multiple meanings of words in microblog speech. Using the multi-scale convolutional kernel to extract local features in the text, DCCMM weakens the negative influence of irregularities in the language of microblog speech on the results of sentiment analysis. Cross-channel feature fusion and multi-head self-attention pooling are used to extract global semantic information and achieve key information filtering. Finally, the character-level and word-level semantic information is effectively fused and condensed through a multi-granularity interactive fusion mechanism. Therefore, DCCMM has a strong sentiment analysis capability and has an excellent ability to cope with the task of microblog remarks sentiment analysis. The DCCMM model achieves the best results in both the comparison and ablation experiments, indicating that the DCCMM can efficiently accomplish the task of microblog sentiment analysis.

However, the sentiment analysis method proposed in this paper is only effective for textual information, and cannot address multimodal information such as pictures, videos, or voice. The information it can extract is still not sufficiently comprehensive.

Therefore, in future work, our next plan is to modify the model structure by introducing operations such as convolution with strides and capsule modules to enable the model to handle multimodal information while reducing the parameters of the model, and to use the multimodal information for experiments to adjust the model structure and parameters in order to improve the usefulness of the model.

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

Funding: This study is supported by the Major Projects of National Social Science Fund of China under Grant 20AZD114, "Kunpeng" Research Fund Project of CCF-Green Alliance Technology under Grant CCF-NSFOCUS 2020011, and Public Safety Behavioral Sciences Laboratory Open Subject Fund Program of PPSUC under Grant 2020SYS08.

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. Gu, M.; Guo, H.; Zhuang, J.; Du, Y.; Qian, L. Social Media User Behavior and Emotions during Crisis Events. *Int. J. Environ. Res. Public Health* **2022**, *19*, 5197. [CrossRef] [PubMed]
2. Alsini, A.; Huynh, D.Q.; Datta, A. Hashtag Recommendation Methods for Twitter and Sina Weibo: A Review. *Future Internet* **2021**, *13*, 129. [CrossRef]
3. Li, H.; Ma, Y.; Ma, Z.; Zhu, H. Weibo Text Sentiment Analysis Based on BERT and Deep Learning. *Appl. Sci.* **2021**, *11*, 10774. [CrossRef]
4. Alharbi, N.M.; Alghamdi, N.S.; Alkhammash, E.H.; Al Amri, J.F. Evaluation of sentiment analysis via word embedding and RNN variants for Amazon online reviews. *Math. Probl. Eng.* **2021**, *2021*, 5536560. [CrossRef]
5. Lee, G.T.; Kim, C.O.; Song, M. Semisupervised sentiment analysis method for online text reviews. *J. Inf. Sci.* **2021**, *47*, 387–403. [CrossRef]
6. Jamal, N.; Xianqiao, C.; Aldabbas, H. Deep Learning-Based Sentimental Analysis for Large-Scale Imbalanced Twitter Data. *Future Internet* **2019**, *11*, 190. [CrossRef]
7. Wenzhen, J.; Hong, Z.; Guocai, Y. An efficient character-level and word-level feature fusion method for Chinese text classification. *J. Phys. Conf. Ser.* **2019**, *1229*, 012057. [CrossRef]
8. Zhang, Q.; Wu, M.; Lv, P.; Zhang, M.; Yang, H. Research on named entity recognition of chinese electronic medical records based on multi-head attention mechanism and character-word information fusion. *J. Intell. Fuzzy Syst.* **2022**, *42*, 4105–4116. [CrossRef]
9. Hu, C.; Zhang, S.; Gu, T.; Yan, Z.; Jiang, J. Multi-Task Joint Learning Model for Chinese Word Segmentation and Syndrome Differentiation in Traditional Chinese Medicine. *Int. J. Environ. Res. Public Health* **2022**, *19*, 5601. [CrossRef]
10. Yan, W.; Zhou, L.; Qian, Z.; Xiao, L.; Zhu, H. Sentiment Analysis of Student Texts Using the CNN-BiGRU-AT Model. *Sci. Program.* **2021**, *2021*, 8405623. [CrossRef]
11. You, H.; Yu, L.; Tian, S.; Ma, X.; Xing, Y.; Xin, N.; Cai, W. MC-Net: Multiple max-pooling integration module and cross multi-scale deconvolution network. *Knowl.-Based Syst.* **2021**, *231*, 107456. [CrossRef]
12. Yang, P.; Zhou, H.; Zhu, Y.; Liu, L.; Zhang, L. Malware Classification Based on Shallow Neural Network. *Future Internet* **2020**, *12*, 219. [CrossRef]
13. Tong, X.; Wang, J.; Jiao, K.; Wang, R.; Pan, X. Robustness Detection Method of Chinese Spam Based on the Features of Joint Characters-Words. In Proceedings of the International Conference on Computer Engineering and Networks, Singapore, 6 October 2020; pp. 845–851.
14. Chen, W.; Fan, C.; Wu, Y.; Lou, Z. A Chinese Character-Level and Word-Level Complementary Text Classification Method. In Proceedings of the 2020 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taipei, Taiwan, 3–5 December 2020; pp. 187–192.
15. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; p. 26.
16. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-Training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
17. Matthew, E.P.; Mark, N.; Mohit, I.; Matt, G.; Christopher, C.; Kenton, L.; Luke, Z. Deep Contextualized Word Representations. *arXiv* **2018**, arXiv:1802.05365.
18. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative, Pre-Training. 2018. Available online: https://www.cs.ubc.ca/~{}amuham01/LING530/papers/radford2018improving.pdf (accessed on 18 June 2022).
19. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
20. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
21. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z. Pre-Training with whole word masking for chinese bert. *IEEE/ACM Trans. Audio Speech Lang. Processing* **2021**, *29*, 3504–3514. [CrossRef]
22. Joshi, M.; Chen, D.; Liu, Y.; Weld, D.S.; Zettlemoyer, L.; Levy, O. Spanbert: Improving pre-Training by representing and predicting spans. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 64–77. [CrossRef]
23. Su, J. Speed Up without Losing Points: Chinese WoBERT Based on Word Granularity. 2020. Available online: https://kexue.fm/archives/7758 (accessed on 18 June 2022).
24. ZhuiyiTechnolog: Chinese BERT with Word as Basic Unit. 2021. Available online: https://github.com/ZhuiyiTechnology/WoBERT (accessed on 18 June 2022).
25. Chen, S.; Zhang, H.; Lei, Z. Person Re-Identification Based on Attention Mechanism and Context Information Fusion. *Future Internet* **2021**, *13*, 72. [CrossRef]
26. Sangeetha, K.; Prabha, D. Sentiment analysis of student feedback using multi-Head attention fusion model of word and context embedding for LSTM. *J. Ambient. Intell. Humaniz. Computing* **2021**, *12*, 4117–4126. [CrossRef]
27. India, M.; Safari, P.; Hernando, J. Self multi-Head attention for speaker recognition. *arXiv* **2019**, arXiv:1906.09890.
28. Fang, Y.; Gao, J.; Huang, C.; Peng, H.; Wu, R. Self multi-Head attention-based convolutional neural networks for fake news detection. *PLoS ONE* **2019**, *14*, e0222713. [CrossRef]
29. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7370–7377.

30. Chi, X.; Xiang, Y. Augmenting paraphrase generation with syntax information using graph convolutional networks. *Entropy* **2021**, *23*, 566. [CrossRef] [PubMed]
31. Jiang, H.; Zhang, R.; Guo, J. A Comparative Study of Graph Concolutional Networks and Self-Attention Mechanism on Text Classification. *J. Chin. Inf. Processing* **2021**, *35*, 84–93.
32. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
33. 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.
34. 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, Berlin, Germany, 7–12 August 2016; Volume 2, pp. 207–212.
35. 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 January 2015; pp. 2267–2273.
36. Song, G.; Huang, D. A sentiment-Aware contextual model for real-time disaster prediction using Twitter data. *Future Internet* **2021**, *13*, 163. [CrossRef]
37. Zhao, H.; Fu, Z.; Zhao, F. Microblog Sentiment Analysis Based on BERT and Hierarchical Attention. *Comput. Eng. Appl.* **2022**, *58*, 156–162.
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 Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, Guangzhou, China, 23–25 August 2021; pp. 382–396.
39. Wang, B.; Shan, D.; Fan, A.; Liu, L.; Gao, J. A Sentiment Classification Method of Web Social Media Based on Multidimensional and Multilevel Modeling. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1240–1249. [CrossRef]
40. Mu, Z.; Zheng, S.; Wang, Q. ACL-RoBERTa-CNN Text Classification Model Combined with Contrastive Learning. In Proceedings of the 2021 International Conference on Big Data Engineering and Education (BDEE), Guiyang, China, 12–14 August 2021; pp. 193–197.
41. Qiu, H.; Fan, C.; Yao, J.; Ye, X. Chinese Microblog Sentiment Detection Based on CNN-BiGRU and Multihead Attention Mechanism. *Sci. Program.* **2020**, *2020*, 8865983. [CrossRef]
42. Tang, F.; Nongpong, K. Chinese sentiment analysis based on lightweight character-level bert. In Proceedings of the 2021 13th International Conference on Knowledge and Smart Technology (KST), Bangsaen, Thailand, 21–24 January 2021; pp. 27–32.