Journal of Network Intelligence ©2024 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **9**, Number **3**, August **2024**

A Weibo Sentiment Analysis Model Based on Attention Mechanisms and Deep Neural Networks

Chao Deng*

School of Water Conservancy Engineering
Chongqing Water Resources and Electric Engineering College, Chongqing 402160, P. R. China
dengchao@cqsdzy.com

Yu Chen

School of Management
St. Louis University, Baguio 400000, Philippines
2244477@slu.edu.ph

*Corresponding author: Chao Deng

Received November 15, 2023, revised February 17, 2024, accepted May 5, 2024.

Abstract. *Sentiment Analysis has become a central research focus in the field of natural language processing in recent years, drawing significant attention from scholars. Weibo sentiment analysis can analyze the emotions expressed by users on Weibo through machine learning algorithms. This is helpful to understand the emotional state of college students and help ideological and political education. Traditional models for sentiment analysis face several issues, including neglecting edge information in texts, disruption of text sequence features by pooling layers, and shortcomings in feature extraction and key information identification. In order to further enhance the effectiveness of sentiment analysis, this paper proposes a Text Sentiment Analysis Model (AM_DNN) for Sina Weibo using a dynamic deep neural network based on attention mechanisms and bidirectional gated recurrent unit (BiGRU) technology. To begin with, this paper employs wide convolutional kernels to extract edge features from the text and utilizes dynamic k-max pooling to preserve the sequential features of the text in relation to its position. Subsequently, a parallel hybrid structure of Deep Convolutional Neural Network (DCNN) and Bidirectional Gated Recurrent Unit (BiGRU) is established to mitigate the issue of partial feature loss, concurrently retaining local features and global context information, thereby enhancing the model's feature extraction capabilities. Following the fusion of features, an attention mechanism is introduced in this paper to globalize its impact, thereby strengthening the model's ability to identify key information. To validate the effectiveness of the proposed algorithm, the SMP2020-EWECT Weibo dataset is used for testing, encompassing six emotion categories such as joy, anger, and sadness. The AM_DNN model proposed in this paper is compared with CNN, BiLSTM, Transformer, BERT models, BERT_CNN, TextGCN, and BERT_GCN. The results indicate that, considering the comprehensive evaluation metrics of accuracy, precision, recall, and F1 score, the improved AM_DNN model outperforms other models, demonstrating superior classification performance in Weibo sentiment analysis.*

Keywords: sentiment analysis; weibo; deep neural networks; attention mechanism; gated recurrent unit

---

1. **Introduction.** Sentiment analysis has emerged as a core research central in the field of natural language processing in recent years, garnering significant attention from scholars due to its widespread application across various domains. Weibo, as a real-time social

platform for sharing brief messages, has been widely utilized for its convenience and other advantages. Analyzing sentiment in Weibo texts holds the potential for in-depth exploration of information, leading to substantial commercial and societal value. Weibo's emotional analysis has certain value in ideological education. Ideological education aims at guiding individuals to form correct values and emotional attitudes, prompting them to actively face life, care about society and shape a healthy psychological state.

Currently, Weibo text sentiment classification techniques primarily fall into two categories: those based on sentiment dictionaries and those based on machine learning [1]. The former relies on the construction of sentiment dictionaries. However, building a comprehensive sentiment dictionary that can encompass continually emerging network neologisms and other out-of-vocabulary words currently faces significant challenges. Moreover, this construction process is time-consuming and labor-intensive and continues to increase in complexity. The latter overcomes this limitation, alleviating the heavy burden of manually constructing dictionaries [2,3,4]. Nevertheless, the effectiveness of sentiment analysis is constrained by factors such as text feature selection, text vectorization representation, and classifier construction. To enhance sentiment classification effectiveness, many researchers have delved into Weibo text sentiment classification based on the aforementioned factors. Jia and Peng [5] conducted in-depth research on Weibo text sentiment classification by selecting feature words from sentiment dictionaries, combining word2vec to obtain text feature vectors. Jiang et al. [6] categorized words into six classes, constructed a six-dimensional vector representing the emotional vector of words, and combined word vectors to develop a novel sentiment analysis method, further improving classification effectiveness. Basiri et al. [7] building on word2vec training to generate word vectors, combined emoticons to extract emotional features embedded in the text, resulting in a well-performing final textual vectorized representation. Antonakaki et al. [8] designed a sentiment analysis method that integrates multiple features, capturing text characteristics through the fusion of vector features, dictionary features, and emoticon features to enhance text vectorization representation and improve sentiment analysis effectiveness. Chiny et al. [9] proposed a text vector representation method based on multiple features, integrating part-of-speech features and position feature vectors on the foundation of word vectors, although their consideration of part-of-speech features was not comprehensive enough. However, these methods insufficiently consider the impact of semantic features, part-of-speech features, sentiment features, emoticon features, and propagation features on sentiment analysis in Weibo text vectorization representation. Consequently, the capabilities of text vectorization representation are lacking.

In response to the aforementioned challenges, this paper integrates deep neural networks and attention mechanisms. Building upon the generation of word vectors, we comprehensively consider the semantic, part-of-speech, emotional, and propagation features of Weibo texts. We propose a Weibo text sentiment analysis model based on deep neural networks and attention mechanisms, aiming to improve sentiment classification effectiveness.

2. **Weibo Sentiment Analysis Model Based on Deep Neural Networks and Attention Mechanism.**

2.1. **Word Embedding Layer.** The Weibo sentiment analysis model proposed in this paper is composed of four components: Word Embedding Layer, Feature Extraction Layer, Attention Layer, and Output Layer, as shownin Figure 1.

In the Weibo sentiment analysis model proposed in this paper, it is necessary to transform raw textual data into vector representations for use as input to the model. Drawing inspiration from the work of Basiri et al. [10], which proposes a word vector model pre-trained on a large-scale corpus, we adopt this model for text vectorization representation. During the training process, we perform fine-tuning by incorporating the dataset. This model, when processing input sequences, considers all positions and assigns distinct attention weights to each position, enabling the model to more accurately capture the relationships between words, sentences, and contexts.

For instance, in a text containing n vocabulary words, we represent the i-th word, denoted as $w_i \in V^d$, as a d-dimensional word vector obtained through GloVe word embedding. Subsequently, these word vectors are assembled into a text matrix T, represented as $T = [w_1, w_2, \ldots, w_n]$. This text matrix T serves as the input representation of the textual vector for the model.

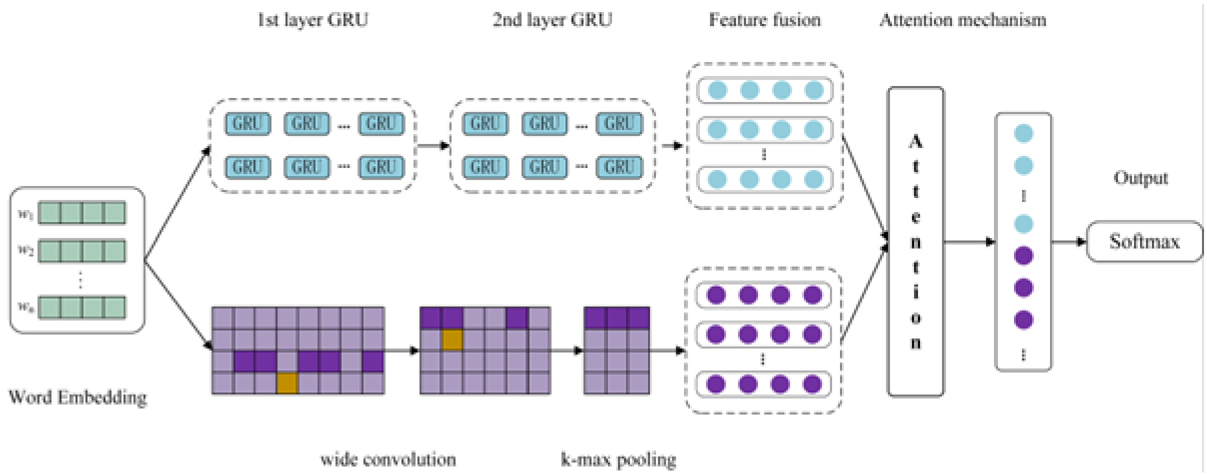

Figure 1. Model architecture

2.2. **Improving Bi-GRU Layer for Global Feature Extraction.** Traditional unidirectional recurrent neural networks face a significant bottleneck in effectively capturing contextual relationships within text, leading to suboptimal performance in complex sequence prediction and classification tasks [11,12]. The emergence of bidirectional recurrent neural networks addresses this limitation by integrating contextual information in both directions, meeting the pressing demands for handling more intricate sequence processing. In this study, we introduce the Gated Recurrent Unit (GRU), an extended type of recurrent neural network, and successfully construct a bidirectional GRU model by combining it with a reverse GRU, providing a more powerful solution for sequence tasks.

Considering the diverse meanings that Chinese vocabulary may exhibit in different contexts, accurate comprehension of their context necessitates the utilization of contextual information. Additionally, in Chinese text, the preceding and subsequent sentences may convey similar facts but exhibit vastly different sentiments [13,14]. To delve deeper into contextual information in news text, this study employs the bidirectional Gated Recurrent Unit (Bi-GRU) network. The Bi-GRU neural network demonstrates outstanding performance in handling sentiment information in Chinese text. Its uniqueness lies in its ability to flexibly process both forward and backward sequence data, effectively leveraging past and future information [15,16,17]. By organically integrating this information, Bi-GRU comprehensively and deeply understands the context of the text. The output of the Bi-GRU neural network is represented as shown in Equation (1).

$$H_t = \left[\overrightarrow{h_t} \oplus \overleftarrow{h_t}\right] \tag{1}$$

This article employs a strategy of element-wise addition to fuse forward and backward outputs to obtain the final output. Due to varying data lengths, it is necessary to standardize the text length for training the model. However, when the text length is insufficient, padding with zero vectors at the end may have a negative impact on the initial semantics of the model, thus diminishing classification performance [18,19]. To address this issue, this paper introduces a mask matrix to enhance the GRU neural network, enabling it to ignore the influence of zero vectors on text semantics during the training process. The specific operation is illustrated in Figure 2.

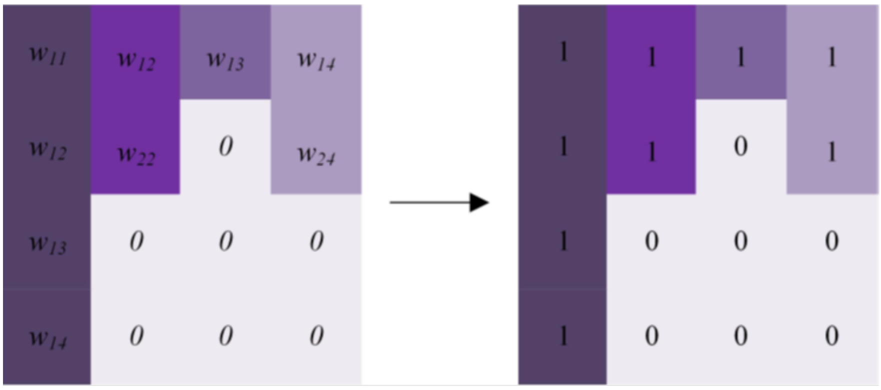

Figure 2. Mask matrix

In Figure 2, the left side represents the matrix of vectors after supplementing with zero vectors, while the right side depicts the matrix of vectors after the masking operation. If the data at a matrix position is from the original data, it is marked as 1 in the mask matrix; if the data at a matrix position is a zero vector, it is marked as 0 in the mask matrix. With the introduction of the mask matrix, we redefine the output equation of the GRU neural network, as shown in Equation (2).

$$h_t = (1 - z_t) * \left[K \otimes h_{t-1} + (1 - K) \otimes \tilde{h}_{t-1}\right] + z_t * \tilde{h}_t \tag{2}$$

where K represents the masking state, denoted by 0 or 1. And ht-1 represents the output from the previous time step, and zt represents the output of the update gate. When the masking state is 0, the corresponding operation tensors are reduced to mitigate the impact of zero vectors on text semantics, thereby enhancing the model's accuracy and efficiency.

By bidirectionally modifying the GRU neural network, a Bidirectional GRU neural network is constructed. Additionally, leveraging RoBERTa as an efficient neural network model, this paper introduces it as an input to the enhanced Bi-GRU, extracting text features from a higher dimension. This enables more rapid and accurate deep learning features by exploiting contextual relationships, which is expected to perform well in sentiment analysis tasks on microblog data in this study.

2.3. **Local Feature Extraction.**

2.3.1. *Convolutional Layer.* In traditional convolutional operations, there is a challenge in extracting n-gram features, namely, the length of the extracted feature information is shortened compared to the original information [20,21]. In contrast, the design of wide convolution does not necessitate complete coverage of the entire text matrix. Instead, it achieves the extraction of longer feature information than the original data by employing zero-padding in uncovered regions.

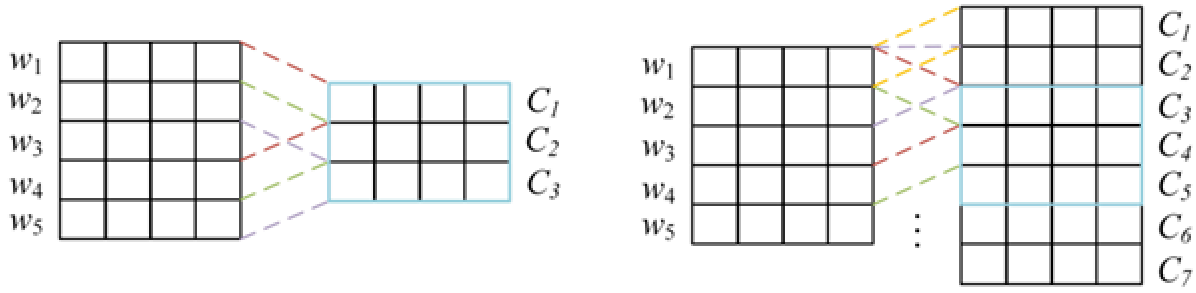

Figure 3. Standard Convolution and Wide Convolution

As illustrated in Figure 3, for convolutional kernels of the same size, the features extracted by conventional convolutional operations represent only a subset of the features extracted by wide convolution. The additional features extracted by wide convolution are predominantly concentrated at both ends of the text, effectively utilizing edge information and resulting in more comprehensive feature extraction. This design allows the model to better capture long-distance dependency relationships in the text, contributing to an improved expressive capacity of the features.

Wide convolution is applied to the text matrix $T = [w_1, w_2, \ldots, w_n]$ obtained from the embedding layer, where a convolutional operation with a kernel size of m extracts features from the text matrix in the j-th convolution operation, as shown in Equation (3).

$$C_j = \left(w^\top * S_{j-m+1:j} + b\right) \tag{3}$$

where $w^\top$ represents the convolutional kernel parameters optimized during the training process, initialized using the MSRA initialization strategy. $T_{j-m+1:j}$ denotes the text information that needs to be extracted by the convolutional kernel in the j-th convolution operation, covering rows $j-m+1$ to j in the text matrix. The feature matrix C extracted from the word embedding matrix T through wide convolution is represented by Equation (4), with dimensions of $d \times (s + m - 1)$.

$$C = [C_1, C_2, \cdots, C_{s+m-1}] \tag{4}$$

2.3.2. *Pooling Layer.* Following the convolutional operation, dynamic k-max pooling is employed to further refine the features initially extracted by the convolutional layer.

In comparison to max pooling, k-max pooling extracts the top k significant feature values, enabling the retention of more critical feature information. Simultaneously, it preserves the relative positional information of these features in the original text, effectively maintaining the sequential characteristics of the text information and preventing information loss [21, 22]. The dynamic adjustment of the k value is determined jointly by the input text length and the current network layer. In shallower layers of the network, a larger k value is chosen to extract more feature values for subsequent steps. Conversely, in deeper layers of the network, the k value is gradually reduced to retain important features while reducing the number of network parameters. This dynamic variation enhances the flexibility of the network structure. The selection method for the k value is expressed as Equation (5).

$$k_l = \max\left(k_{\text{top}}, \left|\frac{L-l}{L} n\right.\right) \tag{5}$$

In the equation, L represents the total number of convolutional layers in the network; l denotes the current layer where the pooling operation takes place; k_{top} signifies the fixed pooling value at the top layer; and n represents the length of the input sentence. Since the k value is dynamically selected based on both the number of network layers and the input text length, the entire convolutional network undergoes dynamic variations in response to the choice of k value.

2.3.3. *Local Feature Output.* The convolutional layer and pooling layer collaboratively form a feature extraction layer. To deeply extract local features from the text, such a feature extraction layer can be repetitively constructed, creating a deep, dynamically characterized convolutional neural network [23, 24]. Multiple convolutional kernels can be simultaneously employed for feature extraction at each layer. Let D_j^i denote the feature matrix extracted by the j-th convolutional kernel in the i-th layer. This can be represented as follows.

$$D_j^i = \sum_{k=1}^{n} m_{j,k}^i * D_k^{i-1} \tag{6}$$

where $m_{j,k}^i$ represents the parameter matrix, D_k^{i-1} denotes the feature matrix obtained by the k-th convolutional kernel in the previous layer. Following the aforementioned computational procedures, the final output is expressed as shown in Equation (7) and Equation (8).

$$D^{\text{top}} = \left[D_1^{\text{top}}, D_2^{\text{top}}, \cdots, D_n^{\text{top}}\right] \tag{7}$$

$$D_j^{\text{top}} = \sum_{k=1}^{n} m_{j,k}^{\text{top}} * D_k^{\text{top}-1} \tag{8}$$

2.3.4. *Attention Layer.* In microblog sentiment analysis, to better capture the dependency relationships between words and enhance attention to important vocabulary, it is crucial to consider the varying importance of different words in the text, along with their respective feature weights [26,27,28]. Therefore, introducing an attention mechanism is a beneficial approach. The attention mechanism assists in precisely learning the weight of each word, facilitating a better understanding of the relationships between words for a more accurate representation of the text. The structure of the attention mechanism is illustrated in Figure 4.

Within the attention mechanism, the output of each Bi-GRU layer is multiplied by the matrix W_g, yielding the sentence vector u_t. Subsequently, a linear transformation of u_t is performed through the tanh activation function. Weight calculation is then executed to obtain the weight value t. Ultimately, the output data from each time step H_t is multiplied by the corresponding weight t, and the weighted outputs from all time steps are summed to form the sentiment feature vector R. The specific computational equations are presented as follow:

$$u_t = \tanh(W_g H_t + b_g) \tag{9}$$

$$\alpha_t = \frac{\exp(u_t)}{\sum_t \exp(u_t)} \tag{10}$$

$$R = \sum_t \alpha_t H_t \tag{11}$$

where H_t represents the output of the Bi-GRU layer at time t, and W_g and b_g denote the parameter vector and bias vector, respectively.

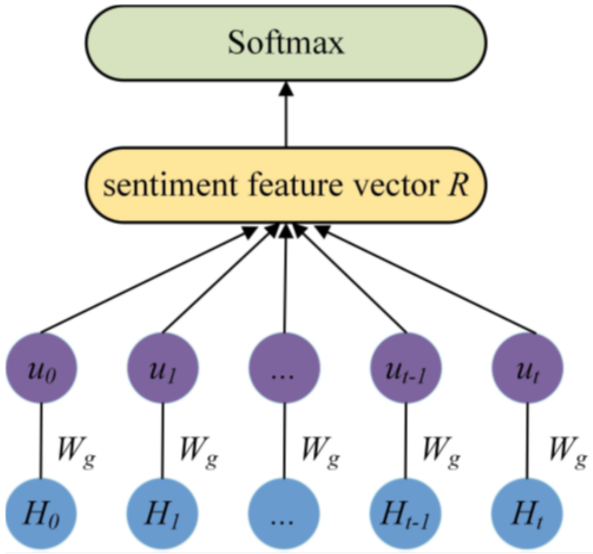

Figure 4. Attention mechanism model

2.4. **Output Layer.** In this section, the sentiment feature vector R is fed into the SoftMax classifier to determine sentiment polarity based on normalized values, where values closer to 1 indicate a more positive sentiment. SoftMax is a logistic regression technique commonly used in multi-class classifiers. It aggregates features that can be assigned to the same category, transforming them into probability values, thereby enhancing the precision and effectiveness of category classification [?, ?]. This method assists in better understanding and predicting relationships between categories. The specific computational equation is given as follows:

$$p(y=i) = \frac{e^{XW_i^T}}{\sum_{j=0}^{k-1} e^{XW_j^T + b_j}} \tag{12}$$

where the text feature vector is represented by X, and k denotes the number of categories. The final output is a k-dimensional row vector, with each dimension ranging between 0 and 1, and the sum of all dimensions is equal to 1. The corresponding loss function computation formula is provided as follows:

$$L(\theta) = -\log p(y=i) \tag{13}$$

3. Experimental Results and Analysis.

3.1. **Experimental Dataset and Evaluation Metrics.** To validate the effectiveness of the proposed model, we conducted tests using the SMP2020-EWECT microblog dataset. This dataset consists of six emotion categories, including happiness, anger, and sadness, and is derived from the open evaluation task released by the 9th National Social Media Processing Conference.

In the experiments, we adopted a word-based granularity, abstracting the text into nodes to construct a text hypergraph. Thus, we first performed tokenization on the text. The dataset exhibited notable class imbalance, with the angry class having the highest quantity, accounting for up to 30%, while the fear and surprise classes had the fewest instances, each less than 10%. To maintain consistent data distribution among subsets and ensure the accuracy of experimental results, we employed a stratified repeated random sub-sampling validation approach, dividing the data into training, validation, and test sets in an 8:1:1 ratio. Finally, we cleaned the text comments using HarvestText and Pyhanlp, removing web links and redundant symbols while preserving emotionally charged punctuation marks such as '?' and '!', resulting in 36,374 valid data samples. The vocabulary size was 39,483, and the average length per word was 21.

The experiments utilized accuracy, precision, recall, and F1 score as evaluation metrics. The results were averaged over five repetitions to obtain the final outcomes.

3.2. **Experimental Evaluation Metrics and Parameter Settings.** Through multiple iterations, we observed that different initial parameter configurations significantly influenced the experimental outcomes. Setting the hidden size to 32 or 64 resulted in smaller dimensions, which were unfavorable for feature extraction by BiGRU. In contrast, a dimensionality of 100 was found to be more suitable, as increasing the dimension further led to potential overfitting. Larger kernel sizes in the convolutional layer yielded fewer extracted features. Thus, combinations like (3,4) and (4,3) showed lower accuracy, while (3,2) could extract sufficient features and partially mitigate overfitting. The dropout rate represents the probability of randomly deactivating some neuron activation values. Setting this value too low may not enhance the model's generalization capability, while setting it too high may result in the loss of too many feature values. A dropout rate of 0.5 was found to be suitable in achieving a balance. For word vector dimensions, setting it to 50 or 100 was insufficient for information representation. A dimensionality of 300 was too high, leading to potential overfitting and increased training time. Considering these factors, a word vector dimension of 200 was found to be effective. The overall model parameter settings are presented in Table 1.

Table 1. Model Parameter Information

Parameters	**Value**
Batch size	64
Learning rate	0.001
Epoachs	10
Dropout	0.5
Word vector dimension	200
Bigru size	2
Hidden size	100
Kernel size	(3,2)

3.3. **Comparative Experiments.** To evaluate the performance of the proposed AM_DNN model in microblog text sentiment analysis, we conducted comparative experiments on the SMP2020-EWECT dataset. We compared the proposed AM_DNN model with CNN, BiLSTM, Transformer, BERT models, BERT_CNN, TextGCN, and BERT_GCN. The experimental results are presented in Table 2.

The experimental results are visually represented in the bar chart shown in Figure 5, providing a more intuitive display of the performance differences among the various models.

As evident from Table 2 and Figure 5, compared to the other seven models, the proposed model in this study achieved superior classification performance on the SMP2020-EWECT dataset, reaching an accuracy of 85%. Precision, recall, and F1 metrics all outperformed

Table 2. Experimental results of 8 models

Model	Accuracy	Precision	Recall	F1
Cnn	0.69	0.65	0.62	0.64
Bilstm	0.69	0.66	0.63	0.65
Transformer	0.73	0.69	0.65	0.67
Bert	0.75	0.71	0.66	0.68
Bert_cnn	0.78	0.74	0.70	0.72
Textgcn	0.72	0.69	0.64	0.67
Bert_gcn	0.83	0.77	0.74	0.76
Am_dnn	0.85	0.81	0.81	0.79

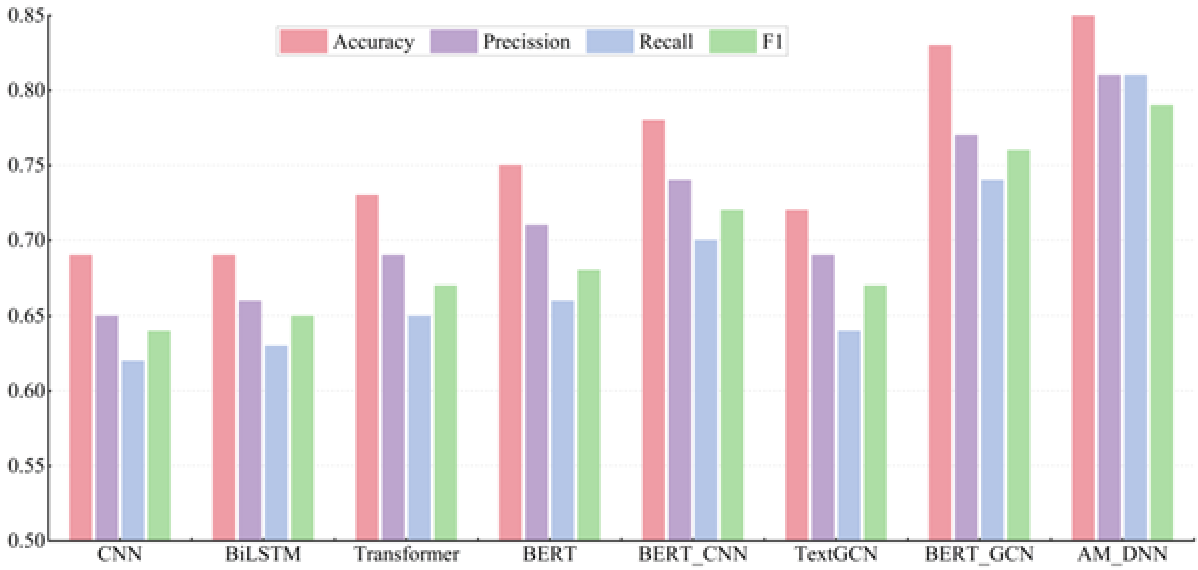

Figure 5. Comparison of experimental results of different models

the other seven models, demonstrating the effectiveness of the improved AM_DNN model proposed in this paper. In contrast, CNN, constructed with a shallow neural network, exhibited limitations in learning capacity, resulting in a significantly lower classification accuracy compared to other neural network models.

Furthermore, in deep learning tasks, the choice of optimizer profoundly influences the model's convergence speed, loss rate variation, and ultimately, the classification performance [31]. Therefore, this study conducted extensive comparative experiments on the SMP2020-EWECT dataset, delving into the profound impact of four widely used optimizers, namely SGD, RMSProp, AdaDelta, and Adam, on model performance. The effects on accuracy and loss values are illustrated in Figure 6 and Figure 7, respectively.

According to the experimental results in Figures 6 and 7, it can be observed that SGD exhibits a relatively slow convergence speed and experiences oscillations in loss during training. This is attributed to the incorrect calculation of gradient directions during the gradient descent process. In contrast, RMSProp, Adam, and AdaDelta, as adaptive learning rate optimizers, demonstrate superior performance compared to SGD. RMSProp initially performs poorly due to the absence of a correction factor. AdaDelta shows good performance in the early and middle stages of training but exhibits some precision fluctuations during the later stages of convergence. Adam optimizer demonstrates a faster convergence speed on both datasets, with overall stable changes in loss values and minimal

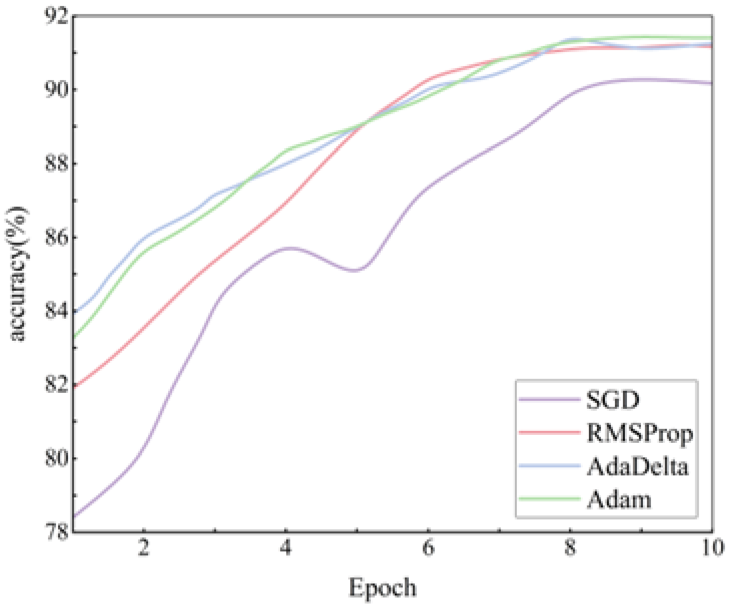

Figure 6. The impact of different optimizers on accuracy

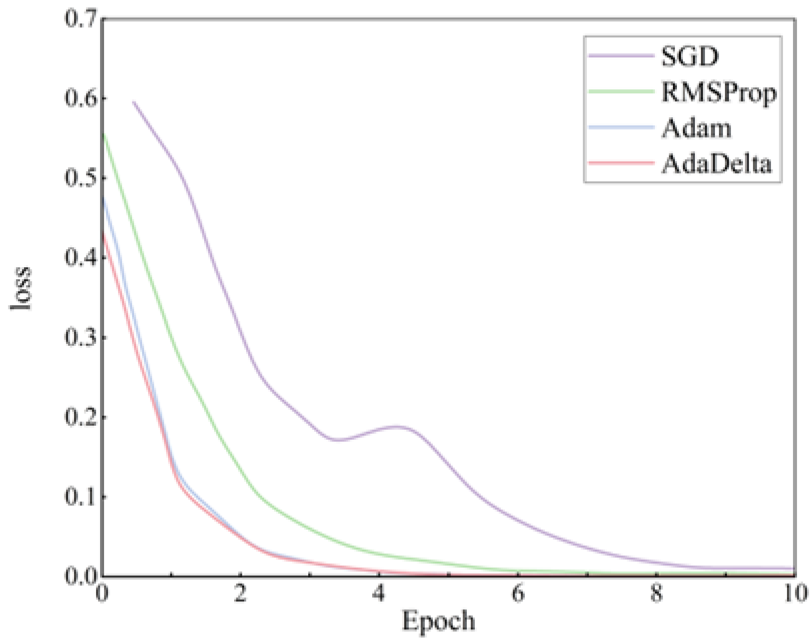

Figure 7. Comparison of experimental results of different models

oscillations. Additionally, Adam is more robust in terms of hyperparameter selection, avoiding precision fluctuation issues and achieving the best overall performance.

The above experimental results clearly indicate that the improved AM_DNN model proposed in this study has achieved significant effectiveness in sentiment analysis tasks for Sina Weibo texts, outperforming existing models. Generally speaking, Weibo's emotional analysis is helpful to improve the accuracy and pertinence of education, help educators better understand social emotional trends, and provide positive ideological guidance for individuals.

4. **Conclusions.** To address the limitations in feature extraction and key information identification in traditional text sentiment analysis, this paper proposes a text sentiment analysis model applicable to Sina Weibo. The model employs a dynamic deep neural network based on attention mechanisms and bidirectional gated recurrent units (BiGRU). Initially, wide convolutional kernels are used to extract edge features, and dynamic k-max pooling is employed to preserve sequential features related to the relative positions of text. Subsequently, a parallel hybrid structure of deep neural networks and BiGRU is established to avoid partial feature loss while retaining both local features and global contextual information, thereby enhancing the model's feature extraction capabilities. Following feature fusion, an attention mechanism is introduced to globalize its impact, enhancing the model's ability to identify key information.

Finally, the proposed AM_DNN model is tested on the SMP2020-EWECT Weibo dataset, and its performance is compared with CNN, BiLSTM, Transformer, BERT models, BERT_CNN, TextGCN, and BERT_GCN. The results demonstrate that, across the metrics of accuracy, precision, recall, and F1, the improved AM_DNN model outperforms other models, showcasing superior classification performance in Weibo sentiment analysis. Additionally, the paper investigates the impact of different optimizers on the model, with experimental results indicating that Adam, due to its robustness in hyperparameter selection, does not induce precision fluctuation issues and exhibits the best overall performance.

Acknowledgment. This work supported by the Project of Counselor's Studio Construction in 2022 of Chongqing Water Resources and Electric Engineering College (No. FDYGZS-2022-02), and the "14th Five-Year Plan" project of Chongqing Education and Science (No. K23ZG3170189).

REFERENCES

[1] H. Li, Y. Ma, Z. Ma, and H. Zhu, "Weibo text sentiment analysis based on bert and deep learning," *Applied Sciences*, vol. 11, no. 22, 10774, 2021.

[2] Y. Li, and N. Li, "Sentiment analysis of Weibo comments based on graph neural network," *IEEE Access*, vol. 10, pp. 23497-23510, 2022.

[3] X. Yang, S. Xu, H. Wu, and R. Bie, "Sentiment analysis of Weibo comment texts based on extended vocabulary and convolutional neural network," *Procedia Computer Science*, vol. 147, pp. 361-368, 2019.

[4] D. Xinhui, Z. Yunqiu, and Y. Kaixi, "Fine-grained sentiment analysis based on Weibo," *Data Analysis and Knowledge Discovery*, vol. 1, no. 7, pp. 61-72, 2017.

[5] W. Jia, and J. Peng, "The public sentiment analysis of double reduction policy on weibo platform," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[6] Z. Jiang, F. Di Troia, and M. Stamp, "Sentiment analysis for troll detection on Weibo," *Malware Analysis Using Artificial Intelligence and Deep Learning*, vol. 11, pp. 555-579, 2021.

[7] M. E. Basiri, S. Nemati, M. Abdar, S. Asadi, and U. R. Acharya, "A novel fusion-based deep learning model for sentiment analysis of COVID-19 tweets," *Knowledge-Based Systems*, vol. 228, 107242, 2021.

[8] D. Antonakaki, P. Fragopoulou, and S. Ioannidis, "A survey of Twitter research: Data model, graph structure, sentiment analysis and attacks," *Expert Systems with Applications*, vol. 164, 114006, 2021.

[9] M. Chiny, M. Chihab, O. Bencharef, and Y. Chihab, "LSTM, VADER and TF-IDF based hybrid sentiment analysis model," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, pp. 141-152, 2021.

[10] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, pp. 279-294, 2021.

[11] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics*, vol. 9, no. 3, 483, 2020.

[12] C. C. Hyland, Y. Tao, L. Azizi, M. Gerlach, T. P. Peixoto, and E. G. Altmann, "Multilayer networks for text analysis with multiple data types," *EPJ Data Science*, vol. 10, no. 1, 33, 2021.

[13] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121-2133, 2022.

[14] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of Quantum Genetic Optimization of LVQ Neural Network in Smart City Traffic Network Prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[15] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, "Human motion recognition based on SVM in VR art media interaction environment," *Human-centric Computing and Information Sciences*, vol. 9, 40, 2019.

[16] K. Wang, X. Zhang, F. Wang, T.-Y. Wu, and C.-M. Chen, "Multilayer Dense Attention Model for Image Caption," *IEEE Access*, vol. 7, pp. 66358-66368, 2019.

[17] M.-E. Wu, J.-H. Syu, and C.-M. Chen, "Kelly-Based Options Trading Strategies on Settlement Date via Supervised Learning Algorithms," *Computational Economics*, vol. 59, no. 4, pp. 1627-1644, 2022.

[18] H. T. Phan, V. C. Tran, N. T. Nguyen, and D. Hwang, "Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model," *IEEE Access*, vol. 8, pp. 14630-14641, 2020.

[19] A. U. Rehman, A. K. Malik, B. Raza, and W. Ali, "A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis," *Multimedia Tools and Applications*, vol. 78, pp. 26597-26613, 2019.

[20] M. Singh, A. K. Jakhar, and S. Pandey, "Sentiment analysis on the impact of coronavirus in social life using the BERT model," *Social Network Analysis and Mining*, vol. 11, no. 1, pp. 33, 2021.

[21] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Tree-structured regional CNN-LSTM model for dimensional sentiment analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 581-591, 2019.

[22] A. Yadav, and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4335-4385, 2020.

[23] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment analysis for E-commerce product reviews in Chinese based on sentiment lexicon and deep learning," *IEEE Access*, vol. 8, pp. 23522-23530, 2020.

[24] F. Long, K. Zhou, and W. Ou, "Sentiment analysis of text based on bidirectional LSTM with multi-head attention," *IEEE Access*, vol. 7, pp. 141960-141969, 2019.

[25] F. Hemmatian, and M. K. Sohrabi, "A survey on classification techniques for opinion mining and sentiment analysis," *Artificial Intelligence Review*, vol. 52, no. 3, pp. 1495-1545, 2019.

[26] C. Singla, F. N. Al-Wesabi, Y. S. Pathania, B. S. Alfurhood, A. M. Hilal, M. Rizwanullah, M. A. Hamza, and M. Mahzari, "An Optimized Deep Learning Model for Emotion Classification in Tweets," *Computers, Materials & Continua*, vol. 70, no. 3, 2022.

[27] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093-1113, 2014.

[28] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522-51532, 2019.

[29] S. Kardakis, I. Perikos, F. Grivokostopoulou, and I. Hatzilygeroudis, "Examining attention mechanisms in deep learning models for sentiment analysis," *Applied Sciences*, vol. 11, no. 9, 3883, 2021.

[30] X. Wang, M. Tang, T. Yang, and Z. Wang, "A novel network with multiple attention mechanisms for aspect-level sentiment analysis," *Knowledge-Based Systems*, vol. 227, 107196, 2021.

[31] H. Sadr, M. M. Pedram, and M. Teshnehlab, "Convolutional neural network equipped with attention mechanism and transfer learning for enhancing performance of sentiment analysis," *Journal of AI and Data Mining*, vol. 9, no. 2, pp. 141-151, 2021.