Mathematical Biosciences and Engineering

MBE, 20 (2): 1981–1992.
DOI: 10.3934/mbe.2023091
Received: 06 August 2022
Revised: 02 November 2022
Accepted: 03 November 2022
Published: 09 November 2022

http://www.aimspress.com/journal/MBE

*Research article*

# DCCL: Dual-channel hybrid neural network combined with self-attention for text classification

**Li Chaofan**[1,2,†,*], **Liu Qiong**[3,†] **and Ma Kai**[4]

[1] The Yancheng School of Clinical Medicine of Nanjing Medical University, Jiangsu 224008, China
[2] Quality Management Division, Yancheng Third People's Hospital, Jiangsu 224008, China
[3] School of Medical Imaging, Jiangsu Vocational College of Medicine, Jiangsu 224005, China
[4] School of Medical Information and Engineering, Xuzhou Medical University, Jiangsu, 221004, China

† These authors contributed equally to this work.

* **Correspondence:** Email: lichaofanautism@163.com.

**Abstract:** Text classification is a fundamental task in natural language processing. The Chinese text classification task suffers from sparse text features, ambiguity in word segmentation, and poor performance of classification models. A text classification model is proposed based on the self-attention mechanism combined with CNN and LSTM. The proposed model uses word vectors as input to a dual-channel neural network structure, using multiple CNNs to extract the N-Gram information of different word windows and enrich the local feature representation through the concatenation operation, the BiLSTM is used to extract the semantic association information of the context to obtain the high-level feature representation at the sentence level. The output of BiLSTM is feature weighted with self-attention to reduce the influence of noisy features. The outputs of the dual channels are concatenated and fed into the softmax layer for classification. The results of the multiple comparison experiments showed that the DCCL model obtained 90.07% and 96.26% F1-score on the Sougou and THUNews datasets, respectively. Compared to the baseline model, the improvement was 3.24% and 2.19%, respectively. The proposed DCCL model can alleviate the problem of CNN losing word order information and the gradient of BiLSTM when processing text sequences, effectively integrate local and global text features, and highlight key information. The classification performance of the DCCL model is excellent and suitable for text classification tasks.

**Keywords:** text classification; convolutional neural networks; long short-term memory networks;

self-attention; feature fusion

## 1. Introduction

Text classification is modelling the relationship between text features and text categories to perform text category determination [1]. Unlike English grammar, Chinese text classification has character-based [2] and word-based [3] methods. The character-based method reduces the impact of unfamiliar words, but individual characters contain insufficient semantic information. The word-based method first faces the problem of accurate word segmentation, which directly affects the effectiveness of the model. However, the text classification task based on text feature words is still the most widely used method at present.

The main algorithm models for text classification can be divided into rule and template-based, statistical and machine learning-based, and deep learning-based methods.

Rule-based methods draw on the help of professionals to develop many decision rules for predefined categories, with the degree of match to particular rules serving as feature representations of the text. Limited by subjectivity, the comprehensiveness and scalability of rule templates, and most notably the complete lack of portability of rule templates, text classification models based on rule formulation have not progressed effectively.

Machine learning-based text classification algorithms [4–6] mainly include Decision Tree, Naive Bayesian Model, Support Vector Machine, and K-Nearest Neighbors. Kanish [7] used TF-IDF to convert the news corpus into digital vectors and compared KNN, RF, and LR on a specific dataset, with LR being the best and KNN the worst for classification. Chen [8] constructed the overall correlation factor of different categories, and obtained the calculation method of the optimal correlation factor by balancing the deviation and variance, which improved the classification accuracy of NBM. Liu [9] proposed an improved KNN text classification algorithm based on Simhash, which solves the computational complexity and data imbalance of traditional KNN by calculating the average Hamming distance of neighboring texts. Although the above improved machine learning model improves the effect of text classification to a certain extent, it still needs artificial feature selection and feature extraction. Limited by the size of the text dataset, the accuracy of feature extraction, and ignoring the correlation between text features, it has poor generality and scalability.

Deep learning-based text classification algorithms mainly include convolutional neural networks, recurrent neural networks, long short-term memory networks, and the fusion of various types of neural network models. With the introduction of the word2vec [10,11] model, word sequences can be converted into low-dimensional dense word vectors with rich semantic information, making neural network models widely used in text classification tasks. Kim [12] proposed to use convolutional neural networks for text classification, setting different weights through convolutional kernels to obtain richer local features and extracting key information through max-pooling operations. The network structure is simple, efficient and robust due to its unique weight-sharing strategy, which allows the training model to have fewer parameters. Rehman [13] constructed a CNN-LSTM model to evaluate the movie review dataset and obtained good results. Gao [14] constructed a hybrid CNN-BiGRU model, ignoring the effect of the loss of word order information caused by CNN on sequence modelling with BiGRU. Although the method of model fusion improves the classification effect to a certain extent, it cannot represent the importance of text features to the classification effect. The introduction of the attention mechanism effectively solves this problem [15]. Wang [16] used the attention mechanism to assign weights to the deep-level information of text extracted by BiGRU to filter effective text features and reduce the interference of noisy features, effectively improving the effectiveness of the model. Deng [17] proposed the attention-based BiLSTM fused CNN model for Chinese extended text classification, by introducing a gating mechanism to assign weights to BiLSTM and CNN output to obtain text fusion features. In addition, the related neural network fusion models also include MTL-LC [18], CNN-BiLSTM-Attention [19], AC-BiLSTM [20], and Attention-BiLSTM [21]. Although the fusion model effectively improves the model prediction, it mainly adopts a recursive network structure. The extracted information is prone to gradient disappearance and explosion problems when transmitting backward. Meanwhile, the recursive network structure only uses the advantages of a single network when extracting text features. It cannot fuse the advantages of CNN and RNN to extract text features, so the classification effects need to be improved.

Pre-training is performed by training the language model through a large amount of original text to obtain an initialized model with parameters. Then fine-tuning is performed based on the pre-trained language model according to the specific task [22]. Pre-training methods have shown better results in classification and labeling tasks in NLP [23,24]. Currently, the popular pre-training methods include ELMo, OpenAI GPT, BERT [25], and XLNet [26]. However, such models are particularly complex in structure and require tremendous arithmetic support.

In order to solve the problems of sparse text features, loss of key feature information, low model performance and poor classification results when processing text classification tasks with CNN and RNN. This study constructed a dual-channel neural network model combining CNN and LSTM with self-attention for text classification. The main contributions are as follows:

(1) N-Gram information of different word windows is extracted using multilayer CNN to enrich the local feature representation of the text.

(2) Using BiLSTM for feature representation of sentence sequences and adding attention mechanism for weighting the hidden layer states to complete effective feature screening.

(3) A dual-channel neural network text classification model is constructed, which can effectively integrate the local and global features by the fusion of extracted text feature information, alleviating the problem that CNN will lose word order information and the gradient of BiLSTM when processing text sequences.

## 2. Materials and methods

DCCL: text classification model based on self-attention combined with CNN and LSTM is shown in Figure 1.

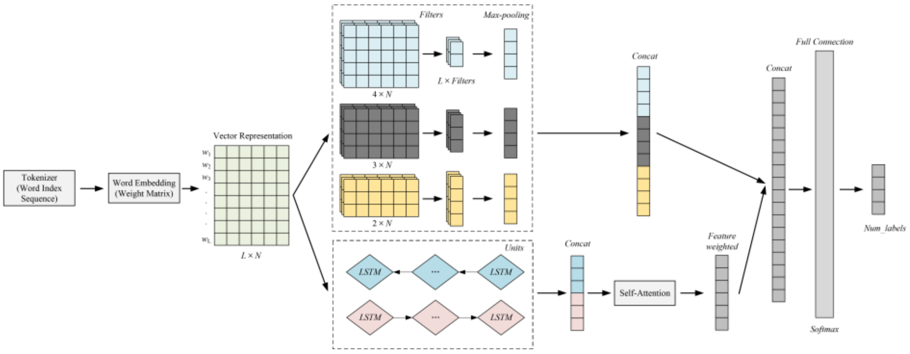

**Figure 1.** Structure of DCCL text classification model.

### 2.1. Word embedding

Pre-processing operations are performed on the text dataset, including word segmentation and removal of stop words, to form the original corpus. Training word vectors using word2vec, default skip-gram. Tokenizer converts text sequences into word index sequences based on word lists and automatically pads them to a fixed length. The word vectors trained by word2vec are used as the weight matrix for the word embedding layer, and the text sequence is vectorized and used as the neural network input. The pre-processing process and vectorized representation for the dataset is shown in the following Figure 2.

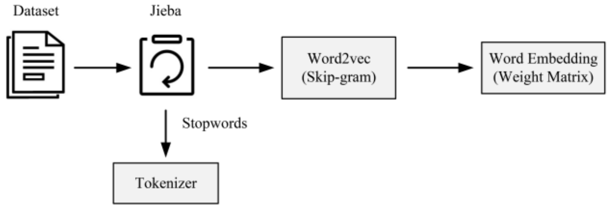

**Figure 2.** Pre-processing process for text dataset.

### 2.2. Multi-layer CNN structure

For text input sequence $S=(w_1,w_2,w_3\cdots w_n)$, $w_i \in R^d$, $d$ is the word vector dimension. The width of the convolution kernel is the same as the word embedding dimension, and the number of words taken in the window for each convolution operation is $h$, so the convolution kernel $\omega \in R^{h*d}$. For each window slide, the convolution result $c_i$:

$$c_i = ReLU(\omega \cdot w_{i:i+h-1}) + b \qquad (1)$$

where $ReLU$ is the nonlinear activation function, $w_{i:i+h-1}$ is the number of words taken in each convolution operation, and $b \in R$ is the bias term.

The length of the sequence $S$ is $n$, the padding parameter is set to the same mode, the stride size is $s$, and the convolution summary result $c=[c_1,c_2,c_3\cdots,c_{n/s}]$. The pooling layers then perform MaxPooling operation on the convolutional layer results, increasing the perceptual field of the upper convolutional kernel, preserving the main features of the word vector sequence, reducing the parameters and computation of the next layer, and preventing overfitting.

For the input of a sequence of word vectors $S$, the outputs of each layer in the parallel structure are $O_1$, $O_2$, $O_3$, respectively, and the overall output $O$ for the TextCNN is expressed as:

$$O = concatenate\left(\left[O_1,O_2,O_3\right], axis=-1\right) \qquad (2)$$

where *concatenate* denotes the *concatenate* () function and *axis* denotes the way of dimension splicing.

### 2.3. BiLSTM-Attention

Sepp Hochreiter [27] proposed LSTM to solve the problem of RNNs with long-term dependencies arising from processing too much information, leading to gradient disappearance or explosion. The structure of the LSTM unit is shown in Figure 3. By linking the memory cell, the input gate, the forgetting gate, and the output gate, the relevant parameters of the gate are controlled and updated to learn and train the model, that is, to adjust the degree of information update and forget. Therefore, the memory cell can preserve the semantic information of longer sequences effectively.

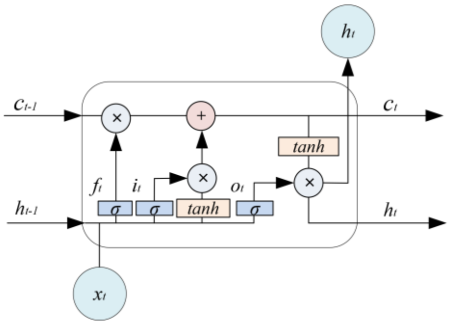

**Figure 3.** LSTM unit structure.

For the moment $t$, the input of the LSTM unit includes the current moment input vector $x_t$, the previous moment memory cell information $c_{t-1}$, and the previous moment hidden layer output information $h_{t-1}$. The specific implementation of the LSTM unit is as follows.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \qquad (3)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \qquad (4)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \qquad (5)$$

$$\overline{c_t} = tanh(W_c x_t + U_c h_{t\text{-}1} + b_c) \qquad (6)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \overline{c_t} \qquad (7)$$

$$h_t = o_t \cdot tanh(c_t) \qquad (8)$$

where $\sigma$ is the *sigmoid* function, $W_i$, $W_o$, $W_f$, $W_c$ are the weight matrix on the input vector $x_t$, $U_i$, $U_o$, $U_f$, $U_c$ are the weight matrix on the hidden layer state $h_{t-1}$, and $b_i$, $b_o$, $b_f$, $b_c$ are the bias vector. $i_t$, $o_t$, $f_t$ represent the input gate, output gate, and forget gate, respectively.

Finally, the splicing of the output vectors of the forward and backward LSTM units is performed, and the feature vectors with bidirectional semantics are the output of the BiLSTM neural network layer.

$$H_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \in R^n \qquad (9)$$

BiLSTM cannot show the importance of key information in context during computation, and it causes information redundancy when dealing with long sequence tasks. The introduction of self-attention to weight the hidden layer states of the BiLSTM can effectively highlight essential text features. The input of self-attention consists of $Q$(*Query*), $K$(*Key*), and $V$(*Value*). First, linearly transform $Q$, $K$, and $V$.

$$Q = W^Q H_t, \quad K = W^K H_t, \quad V = W^V H_t \qquad (10)$$

where $Q = K = V = H_t$, $W^Q$, $W^K$, $W^V$ are the weight matrix of $Q$, $K$, $V$ respectively.

$Q$ and $K$ are computed using the scaled dot-product function, normalized to probability distribution by softmax to obtain the vector of self-attention weights, which is then multiplied by $V$ to obtain the final weighted output $A$.

$$A = Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (11)$$

Where $d_k$ denotes the dimensions of Q, K, V, and $\sqrt{d_k}$ is the scaling factor.

### 2.4. Classification prediction

In order to take into account both local and global features of the text sequence, the overall output of the dual-channel neural network is obtained by concatenating the individual channel output. Then, the fully connected layers are connected for dimensionality reduction and used as input to the softmax classifier. Finally, directly output the probability of the text categories.

$$Output = Concatenate\left(\left[O, A\right]\right) \qquad (12)$$

$$\hat{y} = softmax(W_f \cdot Output + b_f) \qquad (13)$$

Where, $\hat{y}$ is the probability of the text category predicted by the model, and $W_f$ and $b_f$ are the weight and bias matrix of the fully connected layer, respectively.

Set the softmax cross-entropy as the loss function for the overall training of the model.

$$Loss(\hat{y}, y) = -\sum_{i=1}^{k} y_i \cdot \log \hat{y}_i \qquad (14)$$

Where, $y$ is the k-dimensional one-hot encoded vector of true labels.

## 3. Results

### 3.1. Experimental dataset

The experimental datasets are drawn from the open news corpus, Sougou and THUCNews, and the sample sizes for the two types of datasets are shown in Table 1 below.

**Table 1.** Sample size distribution of the dataset.

| Dataset | Category | Training set | Test set | Total |
|---|---|---|---|---|
| Sougou | 5 | 4000 | 500 | 4500 |
| THUNews | 10 | 50000 | 10000 | 60000 |

### 3.2. Evaluation indicators

Macro average precision (MAP), Macro average recall (MAR), and Macro average F1-score (MAF1) are used as the evaluation indicators of the text classification models. The macro-average is the arithmetic average of precision, recall, and F1-score for each category. The calculation of each type of evaluation indicator is as follows.

$$MAP=\frac{1}{k}\sum_{i=1}^{k}P_i \qquad (15)$$

$$MAR=\frac{1}{k}\sum_{i=1}^{k}R_i \qquad (16)$$

$$MAF1=\frac{1}{k}\sum_{i=1}^{k}F1_i \qquad (17)$$

Where $k$ is the number of label categories, $P_i$, $R_i$ and $F1_i$ represent the precision, recall and F1-score of the $i_{th}$ category respectively.

### 3.3. Classification results

To better verify the superiority of the proposed model for text classification tasks in public domains, we introduced five sets of comparison experiments, including TextCNN, BiLSTM, BiLSTM-Attention, TextCNN-BiLSTM-Attention (SCA-CL), and DCCL.

In constructing model experiments, especially for hybrid dual-channel neural network models, channel-based ablation experiments effectively determine model parameters. For the TextCNN model processing text classification tasks, it is essential to determine the size of the convolutional kernel used to extract N-Gram information. Experiments were conducted using a single-layer CNN structure, as shown below.

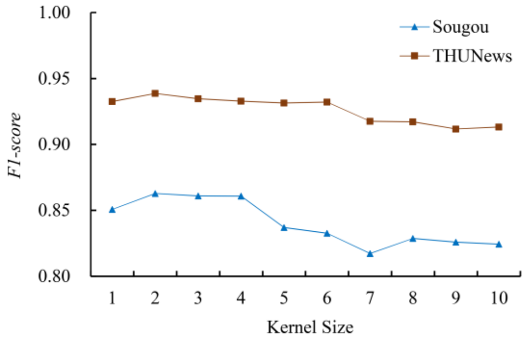

**Figure 4.** Classification results for different convolutional kernel sizes.

**Table 2.** Text classification results for each model.

| | Sougou | | | THUNews | | |
|---|---|---|---|---|---|---|
| Algorithm | MAP | MAR | MAF1 | MAP | MAR | MAF1 |
| TextCNN | 0.8839 | 0.8830 | 0.8834 | 0.9473 | 0.9472 | 0.9470 |
| BiLSTM | 0.8744 | 0.8661 | 0.8683 | 0.9507 | 0.9501 | 0.9503 |
| BiLSTM-Attention | 0.8849 | 0.8729 | 0.8768 | 0.9578 | 0.9577 | 0.9577 |
| SCA-CL | 0.8919 | 0.8882 | 0.8885 | 0.9526 | 0.9527 | 0.9524 |
| DCCL | 0.9103 | 0.8983 | 0.9007 | 0.9627 | 0.9626 | 0.9626 |

The experimental software environment is Windows 10, Python 3.6, Tensorflow 1.14.0, Keras 2.2.5, jieba 0.42. The model parameters are determined after several rounds of experimental comparison, where vocab size is 8000, lstm units is 256, num of filter is 128, kernel sizes are 2, 3, and

4, window size is 5, word embedding dimension is 200, max length of sentence sequence is 256, batch size is 64, dropout is 0.3 to prevent the model from overfitting, learning rate is 0.001, epoch is 50, and Adam is used to optimize the model parameters. The application effects of various text classification algorithms are shown in the following Table 2.

Also, to further validate the superiority of the DCCL model for the text classification task, Figures 5 and 6 show the evolution of the accuracy and loss values for each type of comparison model on the Sougou and THUNews training sets, respectively.

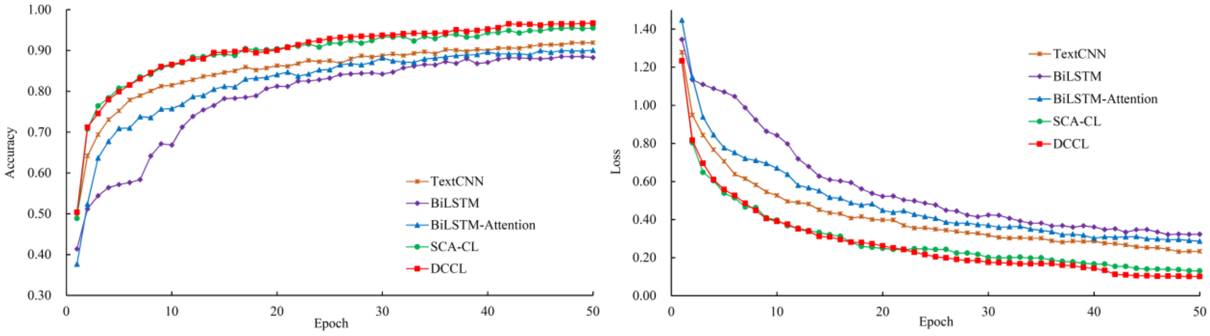

**Figure 5.** Evolution of accuracy and loss for the Sougou.

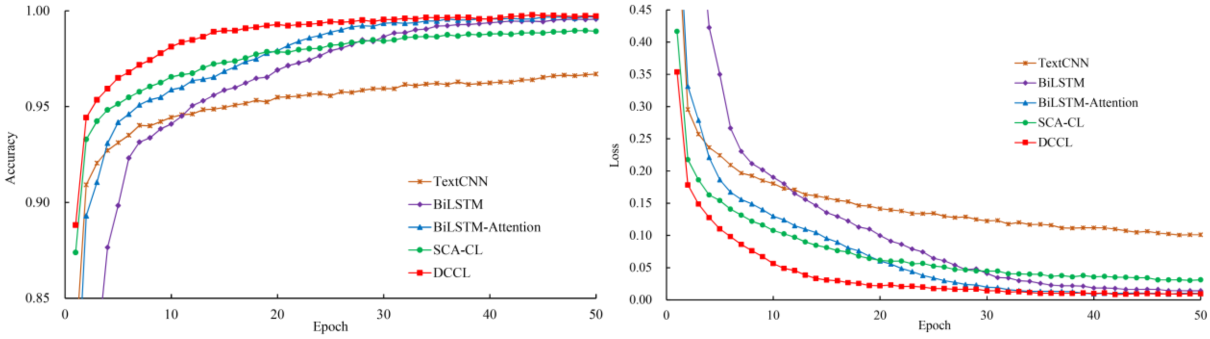

**Figure 6.** Evolution of accuracy and loss for the THUNews.

For the multi-category text classification experiments, Figures 7 and 8 show the results of the comparison models for each category in the Sougou and THUNews test sets, respectively.

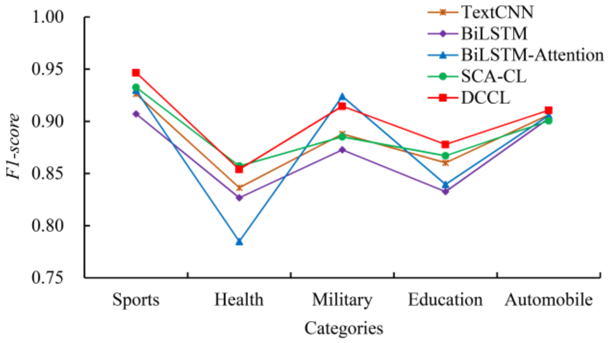

**Figure 7.** Classification results for each category in the Sougou.

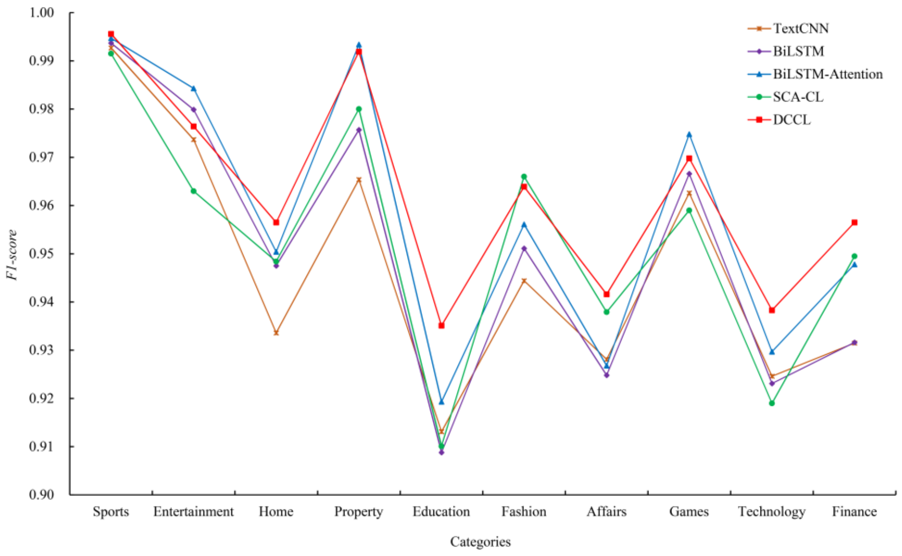

**Figure 8.** Classification results for each category in the THUNews.

## 4. Discussion

The experimental results in Table 2 show that the constructed DCCL model achieved the most excellent results in the classification experiments for both datasets, with MAP, MAR, and MAF1 of 91.03%, 89.83%, and 90.07% respectively for the Sougou dataset, and 96.27%, 96.26%, 96.26% respectively for the THUNews dataset.

The classification results of the DCCL model for the two types of datasets are quite different. The major impacts we consider include two points: (1) The difference in sample size between the two datasets is large, and the word vectors obtained from word2vec training on a large-scale corpus are more closely matched to the actual distribution. Hence, the THUNews has better classification results. (2) The THUNews dataset was filtered by the Natural Language Processing and Social Humanities Computing Laboratory of Tsinghua University, and the text data of each category differed significantly.

For the classification effect of the Sougou, self-attention is introduced to weight the output of the BiLSTM hidden layer to reduce the influence of redundant features on the classification effect. Therefore, the MAF1 of BiLSTM-Attention is 0.85% higher than that of BiLSTM. However, BiLSTM-Attention still performs slightly worse than TextCNN, which uses multiple word windows to extract the N-Gram information of the text, followed by max-pooling operation, similar to the attention mechanism for feature highlighting. For the SCA-CL model, the CNN causes a loss of word order information, and the error is passed to the BiLSTM for text feature reconstitution. However, based on the experimental results, it can be concluded that the SCA-CL model with the application of the tandem structure plays a positive role [28], making MAF1 higher than BiLSTM-Attention by 1.17%. For the ablation experiments of the DCCL model, the classification effect was improved by 1.73% and 2.39% respectively compared to the single channel.

For the THUNews, the difference in classification results achieved by the various comparative experimental models is not too significant. Due to the apparent differences in text data between categories in the THUNews dataset, the sentence-level high-level features constructed by BiLSTM and then features weighted by the attention mechanism, making the BiLSTM-Attention significantly more effective than TextCNN and SCA-CL. For the ablation experiments of the DCCL model, the classification effect was improved by 1.56% and 0.49% respectively compared to the single channel.

Through the training process of various models, it can be seen that the iterations of BiLSTM and BiLSTM-Attention are slow, TextCNN is relatively fast, and SCA-CL and DCCL are both excellent. Due to the small sample size and noisy data of Sougou, the text features constructed by BiLSTM and BiLSTM-Attention have large information redundancy and errors, resulting in poor classification results. TextCNN has excellent performance and high accuracy during training because it captures multi-level local features. Due to the large sample size and manual pre-processing of THUNews, BiLSTM and BiLSTM-Attention can better obtain high-level text features, and ultimately achieve higher accuracy and lower Loss. TextCNN processing of text sequences suffers from word order and information loss, which can cause cumulative propagation of errors in the SCA-CL model. DCCL can complement the shortcomings of CNN and LSTM for feature extraction, thus effectively integrating local and global text features and highlighting essential information to obtain a more comprehensive text feature at multiple levels. As a result, DCCL has the best performance in the training process for both types of datasets.

For each category of the dataset, it can be seen from Figures 7 and 8 that among the five categories of the Sougou, DCCL is the best in the categories of Sports, Education and Automobile, and the next best in the categories of Health and Military. Among the ten categories of the THUNews, DCCL has the best classification effect in the six categories of Sports, Home, Education, Affairs, Technology and Finance, and the next best in the three categories of Property, Fashion, and Games. Meanwhile, for the other comparison models performed poorly in the categories of Home, Education, Affairs, and Technology, while DCCL substantially improved the classification results.

## 5. Conclusions

DCCL: dual-channel hybrid neural network combined with self-attention is proposed to solve the problems of high-dimensional sparse features, the low performance of classification models, and poor classification results in text classification tasks. DCCL complements the shortcomings of CNN and LSTM for text feature extraction, and can integrate local and global features of text and highlight key features. The results of multiple rounds of model comparison experiments with the two datasets show that DCCL can achieve excellent classification results and is suitable for Chinese text classification tasks. The application of the attention mechanism, the overall model structure, and the parameters need to be reasonably adjusted based on the experimental dataset for the DCCL model. At the same time, the effective combination of pre-trained language models and classification models can reduce the time consumption of the training process and significantly improve classification performance.

## Funding statement

This research is supported by Key R&D Program for Xuzhou Science and Technology Plan Project, with granted number KC21308. In addition, it is also supported by Innovation and Entrepreneurship Project for University Students in Jiangsu Province, under granted No. 201810313047Y and 201910313004Z.

## Data availability statement

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflict of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## References

1. S. Al, S. Andrew, Short text classification using contextual analysis, *IEEE Access*, **9** (2021), 149619–149629. https://doi.org/10.1109/ACCESS.2021.3125768
2. X. Zhang, J. B. Zhao, L. C. Yann, Character-level convolutional networks for text classification, *Adv. Neural. Inf. Process. Syst.*, **28** (2015), 649–657.
3. Y. Lin, J. P. Li, L. Yang, K. Xu, H. F. Lin, Sentiment analysis with comparison enhanced deep neural network, *IEEE Access*, **8** (2020), 78378–78384. https://doi.org/10.1109/ACCESS.2020.2989424
4. R. Sharma, M. Kim, A. Gupta, Motor imagery classification in brain-machine interface with machine learning algorithms: Classical approach to multi-layer perceptron model, *Biomed. Signal Process. Control*, **71** (2022). https://doi.org/10.1016/j.bspc.2021.103101
5. D. Kapgate, Efficient quadcopter flight control using hybrid SSVEP+P300 visual brain computer interface, *Int. J. Human-Comput. Interact.*, **38** (2021), 42–52. https://doi.org/10.1080/10447318.2021.1921482
6. A. M. Roy, A multi-scale fusion CNN model based on adaptive transfer learning for multi-class MI-classification in BCI system, (2022). https://doi.org/10.1101/2022.03.17.481909
7. K. Shah, H. Patel, D. Sanghvi, M. Shah, A comparative analysis of logistic regression, random forest and knn models for the text classification, *Augm. Human Res.*, **5** (2020), 5–12. https://doi.org/10.1007/s41133-020-00032-0
8. J. N. Chen, Z. B. Dai, J. T. Duan, H. Matzinger, I. Popescu, Improved Naive Bayes with optimal correlation factor for text classification, *SN Appl. Sci.*, **1** (2019), 1–10. https://doi.org/10.1007/s42452-019-1153-5
9. J. Liu, T. Jin, K. Pan, Y. Yang, Y. Wu, X. Wang, et al, An improved KNN text classification algorithm based on Simhash, *IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing*, (2017), 92–95.
10. T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, *Comput Sci.*, (2013). https://arxiv.org/abs/1301.3781
11. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrasesand their compositionality, *Neural Inform. Process. Syst.*, **26** (2013), 3111–3119. https://arxiv.org/abs/1310.4546v1
12. Y. Kim, Convolutional neural networks for sentence classification, *EMNLP*, (2014). https://arxiv.org/abs/1408.5882
13. A. U. Rehman, A. K. Malik, B. Raza, W. Ali, A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis, *Multimed. Tools Appl.*, **78** (2019), 26597–26613. https://doi.org/10.1007/s11042-019-07788-7
14. Z. W. Gao, Z. Y. Li, J. Y. Luo, X. L. Li, Short text aspect-based sentiment analysis based on CNN + BiGRU, *Appl. Sci.*, **12** (2022). https://doi.org/10.3390/app12052707

15. P. Bhuvaneshwari, A. N. Rao, Y. H. Robinson, M. N. Thippeswamy, Sentiment analysis for user reviews using Bi-LSTM self-attention based CNN model, *Multimed. Tools Appl.*, **81** (2022), 12405–12419. https://doi.org/10.1007/s11042-022-12410-4
16. W. Wang, Y. X. Sun, Q. J. Qi, X. F. Meng, Text sentiment classification model based on BiGRU-attention neural network, *Appl. Res. Comput.*, **36** (2019), 3558–3564. https://doi.org/10.19734/j.issn.1001-3695.2018.07.0413
17. J. F. Deng, L. L. Cheng, Z. W. Wang, Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification, *Comput. Speech Lang.*, **68** (2021). https://doi.org/10.1016/J.CSL.2020.101182
18. J. B. Xie, J. H. Li, S. Q. Kang, Q. Y. Wang, Y. J. Wang, A multi-domain text classification method based on recurrent convolution multi-task learning, *J. Electron. Inform. Technol.*, **43** (2021), 2395–2403. https://doi.org/10.11999/JEIT200869
19. H. Y. Wu, J. Yan, S. B. Huang, R. S. Li, M. Q. Jiang, CNN-BiLSTM-Attention Hybrid Model for Text Classification, *Computer Sci.*, **47** (2020), 23–27. https://doi.org/10.11896/jsjkx.200400116
20. G. Liu, J. B. Guo, Bidirectional LSTM with attention mechanism and convolutional layer for text classification, *Neurocomputing*, **337** (2019), 325–338. https://doi.org/10.1016/j.neucom.2019.01.078
21. G. X. Xu, Z. X. Zhang, T. Zhang, S. A. Yu, Y. T. Meng, S. J. Chen, Aspect-level sentiment classification based on attention-BiLSTM model and transfer learning, *Knowledge-based Syst.*, **245** (2022). https://doi.org/10.1016/j.knosys.2022.108586
22. P. Kumar, B. Raman, A BERT based dual-channel explainable text emotion recognition system, *Neural Networks*, **150** (2022), 392–407. https://doi.org/10.1016/j.neunet.2022.03.017
23. C. Yan, J. H. Liu, W. Liu, X. H. Liu, Research on public opinion sentiment classification based on attention parallel dual-channel deep learning hybrid model, *Eng. Appl. Artif. Intell.*, **116** (2022). https://doi.org/10.1016/j.engappai.2022.105448
24. F. Zhao, X. N. Li, Y. T. Gao, Y. Li, Z. Q. Feng, C. M. Zhang, Multi-layer features ablation of BERT model and its application in stock trend prediction, *Expert Syst. Appl.*, **207** (2022). https://doi.org/10.1016/j.eswa.2022.117958
25. J. Devlin, M. W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, (2018). https://arxiv.org/abs/1810.04805v1
26. A. Alhanouf, A. Abdulrahman, AraXLNet: Pre-trained language model for sentiment analysis of Arabic, *J. Big Data*, **9** (2022). https://doi.org/10.1186/s40537-022-00625-z
27. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9** (1997), 1735–1780. https://doi.org/10.1162/NECO.1997.9.8.1735
28. Q. N. Zhu, X. F. Jiang, R. Z. Ye, Sentiment analysis of review text based on BiGRU-attention and hybrid CNN, *IEEE Access*, **9** (2021), 149077–149088. https://doi.org/10.1109/ACCESS.2021.3118537