Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 12, No. 1, 2025

Study on Small Sample Text Classification Based on Multi-Level Self-Attention and Multi-Feature Residual Fusion under Data Enhancement

Linlin Deng *, Xina Lu

School of Economics, North Minzu University, Yinchuan, Ningxia, China
*** Corresponding author:** Linlin Deng (Email: m19862109528@163.com)

Abstract: In commercial applications, while traditional models can achieve comparable performance to mainstream large language models, they generally necessitate extensive training data. This requirement presents a significant challenge when processing complex, lengthy Chinese texts and multi-label classification tasks with limited data availability. Furthermore, conventional data augmentation techniques frequently disrupt the original word order, thereby diminishing their efficacy for pre-trained language model applications. To overcome these limitations, we introduce the MacBERT-CNN-BiLSTM model, which incorporates a multi-level self-attention mechanism to dynamically weight the integrated features extracted from MacBERT, CNN, and BiLSTM components. Our methodology preserves the integrity of original features during the final fusion phase by combining weighted features with original features through residual connections, thus generating a comprehensive final representation. This approach culminates in our MacBERT-BiLSTM-CNN-ResAttNet model (MBCResAttNet), specifically designed for multi-label classification of small-sample Chinese literature abstracts. We conducted extensive evaluations of our model across three datasets: AEDA-augmented, EDA-augmented, and original samples, benchmarking against six alternative models. The empirical results demonstrate that incorporating pre-trained language models substantially enhances classification performance. Moreover, the multi-level self-attention mechanism combined with residual feature fusion effectively captures global textual patterns, resulting in significant performance improvements. In the context of pre-trained language models, AEDA demonstrates superior efficacy compared to EDA in maintaining original semantic integrity. Additionally, the residual feature fusion methodology preserves critical original information while markedly improving model performance. With the implementation of AEDA augmentation, all evaluated models exhibited performance gains exceeding 10%, with our MBCResAttNet model attaining 96.17% accuracy—representing a substantial 13.41% improvement over baseline methods.

Keywords: MacBERT; Data Enhancement; Self-attention Mechanism; Text Categorization; Small Samples.

1. Introduction

Currently, multimodal large language models excel in text classification, but using paid APIs is costly. Task-specific deep learning models are cheaper but depend heavily on training data, especially for complex Chinese texts. For high performance with small samples, models need strong local feature extraction, contextual relationship building, and information retention, plus appropriate data augmentation without altering original semantics.

Deep learning has advantages over traditional machine learning in text analysis, reducing manual feature engineering through pre-trained language models. While traditional methods struggle with unstructured text, deep learning excels with complex semantics. Yoon Kim's work[1] effectively applied CNNs to text classification, influencing subsequent research. Now deep learning is widely used in NLP tasks: [2] applied TextCNN to adverse drug event detection; [3] used improved multi-channel TextCNN for short texts; [4] achieved excellence in multi-label classification using CNN; [5] proposed an effective CNN-BiLSTM model with attention for depression detection.

Most research focuses on downstream models rather than upstream text processing. Google's BERT[6] uses masked language modeling, Transformers, and self-attention to capture sentence structure and context, solving Word2Vec's polysemy limitations. [7] and [8] showed BERT improves classification performance. [9] proposed FF-BERT for flash flood classification. [10] demonstrated BERT+BiLSTM +Attention outperforms Word2Vec in Chinese medicine text classification.

BERT quickly became fundamental in NLP, inspiring improved models like RoBERTa[11], ALBERT[12], DistilBERT [13], and ChineseBERT[14] which incorporates grapheme and pinyin information. MacBERT[15] introduced correction-type masked language modeling, effectively addressing pre-training and downstream task inconsistencies, outperforming original BERT in Chinese NLP tasks.

This paper uses MacBERT for upstream feature extraction and TextCNN for downstream local feature capture. Since CNN may miss global features[16], BiLSTM is added for contextual connections as suggested by [17]. [18] noted CNN-BiLSTM combinations compensate for each other's weaknesses in feature extraction.

The self-attention mechanism highlights key words and captures global features. [19] achieved accurate polarity prediction using BERT-BiLSTM with improved self-attention. [20] used multi-head self-attention for complex contextual relationships. [21] improved short text classification performance. [22] constructed a quantum self-attention neural network with excellent text classification performance. This paper implements self-attention mechanisms at multiple levels.

[23] improved classification by fusing features from CNN and LSTM models. [24] enhanced performance using feature selection fusion. With limited small-sample information, this

paper retains both original and attention-weighted features through residual connections.

For small-sample data augmentation, [25] proposed symbolic augmentation like word replacement, which risks changing original semantics. [26] introduced Easy Data Augmentation (EDA) with four operations: synonym replacement, random insertion, word swapping, and deletion. [27] achieved improvements using EDA for medical data. However, with pre-trained models like BERT, EDA may change word order or cause information loss, potentially decreasing model performance.

The AEDA method [28] randomly inserts punctuation, maintaining word order while improving generalization. This paper explores AEDA's combination with Chinese pre-trained language models.

The resulting MBCResAttNet model with AEDA data augmentation performs excellently in multi-label classification of small-sample Chinese literature abstracts, providing valuable reference for combining data augmentation with pre-trained language models in small-sample multi-label classification.

2. Materials and Methods

2.1. MBCResAttNet Model

To reduce the risk of overfitting and improve the model's generalization ability, the Dropout algorithm is introduced based on the above model. The risk of overfitting is reduced by discarding a portion of the neurons. At the same time, the model's capability is enhanced because a multi-level self-attention mechanism is introduced into the model to ensure that information is not lost during processing. In the final feature fusion stage, the comprehensive features obtained by concatenating BiLSTM_out and TextCNN_out are first saved. After these comprehensive features are processed by the final self-attention mechanism, a weighted feature representation is obtained. This weighted feature and the previously retained original comprehensive features are fused into the final feature representation by means of a residual connection, and then passed through a fully connected layer to complete the final classification task. This method ensures that the original feature information is not lost and can avoid the loss of feature information between layers. The final model structure is shown in (Fig 1)

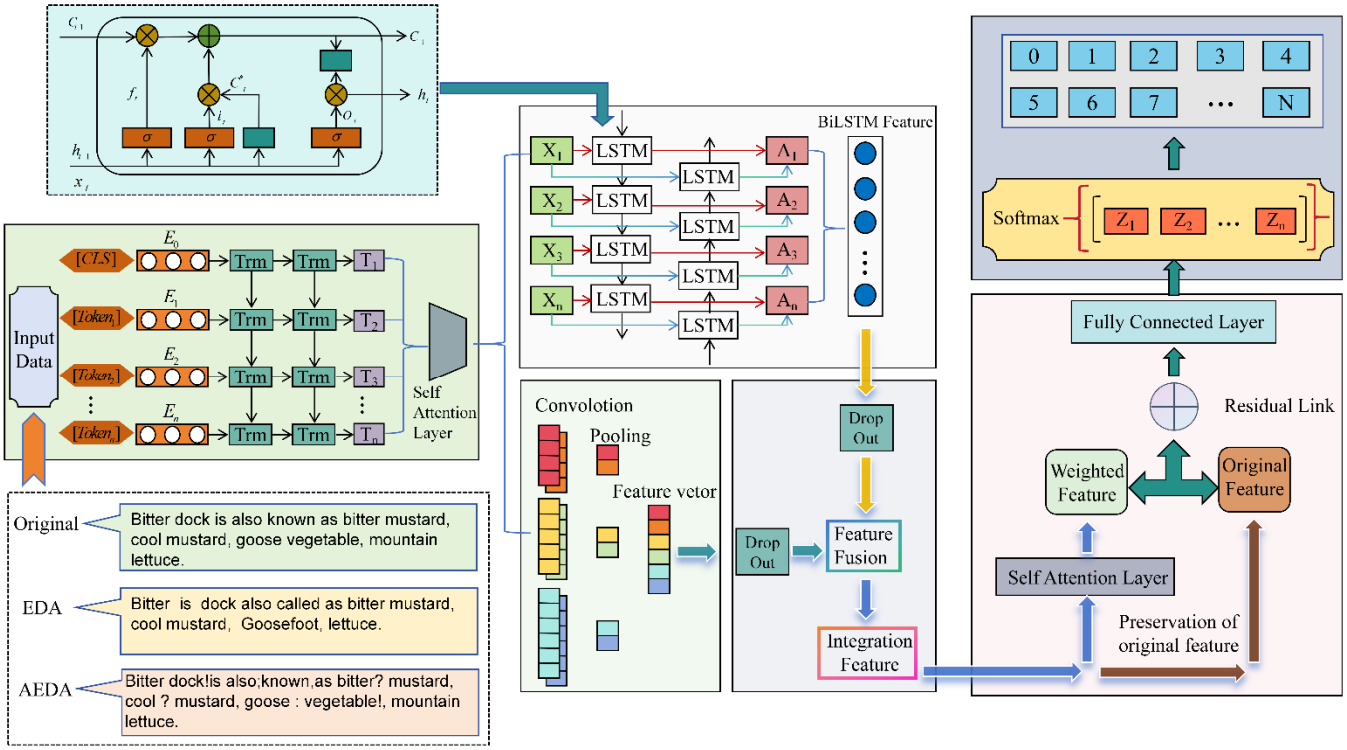

Fig 1. MBCResAttNet model structure diagram

2.2. Data Enhancement Technology

2.2.1. EDA Data Enhancement

EDA (Easy Data Augmentation), proposed by Jason Wei et al., expands datasets and improves model generalization when data is scarce. By manipulating text data through four basic operations—synonym replacement, random insertion, random swapping, and random deletion—EDA creates diverse training samples that help models learn varied expressions and improve robustness. This technique is ideal for limited-data tasks like text classification and sentiment analysis, where it reduces overfitting and enhances generalization. Its advantage lies in improving linguistic understanding without complex external resources. However, operations like synonym replacement may alter original semantics, potentially harming performance in semantically sensitive tasks.

2.2.2. AEDA Data Enhancement

AEDA is a simpler text data augmentation method compared to EDA. The paper introducing AEDA was published by the IMP Lab of the University of Parma, Italy, at the EMNLP 2021 conference. AEDA was inspired by EDA, which proposes four simple data augmentation operations: synonym replacement (replacing words in a sentence with synonyms from a synonym list), random swap (randomly swapping two words in a sentence to change the order of words), random insertion (randomly inserting a synonym for a word in the sentence), and random deletion (randomly deleting words in the sentence). However, EDA sometimes yields poor results when using a pre-trained language model

and may even worsen the outcomes. This may be because operations like synonym replacement, random swap, random insertion, and random deletion alter the sequence information of the original text. In contrast, the AEDA method only inserts punctuation marks, resulting in minimal modification to the sequence information of the original data. Modifying words may lead to semantic changes and thus negative effects. AEDA, on the other hand, only adds punctuation marks. Although it introduces noise, it does not change the order of the original text, and it does not insert too many punctuation marks, so the added noise is minimal.

The specific method of AEDA data enhancement is as follows: in sentence length 1, a number is randomly selected as the number of punctuation marks to be inserted, and the positions of the punctuation marks to be inserted in the sentence are randomly inserted. There are mainly six types of punctuation marks to be inserted: [“.”, “;”, “?”, “:”, “!”, “,”]

2.3. Experimental Data

The data used in this paper comprises the abstract texts of collected literature. The sample content has been manually cleaned, resulting in 2,532 samples. There are ten categories in total: law, engineering, education, management, economics, science, history, agriculture, literature, and medicine. Table 1 presents the category and label information of the original samples. Table 2 provides an example of a sample in the EDA data augmentation section, and Table 3 presents an example of a sample in the AEDA data augmentation section.

Table 1. Sample Label Names and Distribution

Category Name	Sample Size	Label
Law	230	0
Engineering	267	1
Management	248	2
Pedagogical	226	3
Economics	245	4
Science	261	5
History	295	6
Agronomy	266	7
Literary	267	8
Medical Science	227	9

2.4. Experimental Environment and Model Parameters

The environmental configuration is shown in Table 2.

Table 2. Experimental environment configuration

Related Environment	Specific Version
Operating System	Windows10
GPU	GeForce RTX4090(24GB)
CPU	16vCPU Intel(R) Xeon(R) Platinum 8352V CPU@2.10GHz
Development Language	Python3.10.8
Development Framework	Pytorch2.1.0
Development Tool	PyCharm Professional Edition

The relevant parameters of the deep learning model are shown in Table 3.

Table 3. Related parameters

Parameter Name	Specific Meaning	Value
Epoch	number of training sessions	35
Learning_rate	Learning Rate	1e-5
Train_batch_size	Number of samples included in each batch during training	28
Test_batch_size	Number of samples per batch included in the model Evaluation or Testing	16
Filter_size	Convolutional Kernel Size	[2,3,4]
Weight_Decay	Weight Attenuation	0.015
Drop_out	Discard Rate	0.5
Max_Length	Maximum truncation length of the model (see the following paragraph for analysis)	235

2.5. Selection of Evaluation Indicators

Evaluation metrics the accuracy and F1 value are used as evaluation criteria, and the calculation equation is:

$$\begin{aligned} Accuracy &= \frac{(TP+TN)}{N} \\ Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \\ F1 &= \frac{2\times Precision\times Recall}{Precision+Recall} \end{aligned} \tag{1}$$

TP (True Positive) is the number of samples correctly predicted by the classification model as belonging to a specific subject category. TN (True Negative) is the number of samples correctly predicted by the classification model as not belonging to that subject category. FP (False Positive) is the number of samples incorrectly predicted by the classification model as belonging to that subject category. FN (False Negative) is the number of samples incorrectly predicted by the classification model as not belonging to that subject category.

3. Results and Discussion

3.1. Word Length Analysis

The maximum input length of the MacBERT model is 512 tokens. Here, we set Max_length, the maximum input length of the text, to x^*, which represents the word length used by MacBERT. Assuming that the summary text is set to $Text_i$, it contains x_i tokens. If each token is represented as an n-dimensional vector, the text matrix can be expressed as follows, as shown in the following matrix:

$$W_i = \begin{pmatrix} word_1 \\ word_2 \\ \cdots \\ word_{x_i} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{x_i 1} & a_{x_i 2} & \cdots & a_{x_i n} \end{pmatrix}_{x_i \times n} \tag{2}$$

If $x_{i \geq} x^*$, the first x^*row of the summary text matrix is taken as input; the result is as follows：

$$\begin{pmatrix} word_1 \\ word_2 \\ \dots \\ word_{x^*} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{x^*1} & a_{x^*2} & \dots & a_{x^*n} \end{pmatrix}_{x^* \times n}, x_i \geq x^* \quad (3)$$

If $x_{i <} x^*$, a n-dimensional zero vector is added below the original summary text matrix to supplement the number of rows to x^*; the result is as follows:

$$\begin{pmatrix} word_1 \\ word_2 \\ \dots \\ word_{x_i} \\ 0 \\ 0 \\ \dots \\ 0 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{x_i1} & a_{x_i2} & \dots & a_{x_in} \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{pmatrix}_{x^* \times n} x_i < x^* \quad (4)$$

According to general signal processing principles, the method of padding with zero vectors can lead to the sparsification of text data in text classification, thereby reducing classification efficiency. In other words, the zero vectors added here are essentially data white noise, and the whiter noise there is, the more negatively it impacts subsequent classification processing.

In summary, the choice of x^*will significantly affect the model's performance. If x^* is greater than the text data length and the difference is large, it will result in too much content being deleted. Similarly, if $x^*$is less than the text data length and the difference is large, it will result in more white noise being added. Both situations will cause the model to have a biased understanding of the original meaning, resulting in poor performance.

(Fig 2) shows the skewed distribution of the number of samples across different word ranges for each sample type. From the distribution in the Fig, it can be seen that the characteristics of the three types of data are all right-skewed—that is, the right tail of the data is longer, and most of the data is concentrated on the left. Additionally, from the distribution of the text in different intervals, it can be seen that, for the three sample types, the text word range of 100–300 accounts for the largest proportion of the overall samples. Taking the original sample as an example, the total number of samples with 100–300 words is 1,806, accounting for 71% of the total samples. Therefore, when selecting the word length, we choose the middle word count of 200 words as the benchmark and use a range of ±50 words to conduct model testing. It is found that when the truncation length is 150–200 words, the model performance is generally low. This is because the negative impact of excessive content deletion is often greater than the impact of adding white noise. The accuracy of the model is higher when the truncation length is 200–250 words, reaching its highest at 230–250 words, generally exceeding 90%. Within this range, we choose 5 as the step size to divide the interval, and we select truncation lengths of 230,235,240,245 and 250 words to test and determine the final word length. Table 4 and (Fig 3) show the test results and corresponding training metrics.

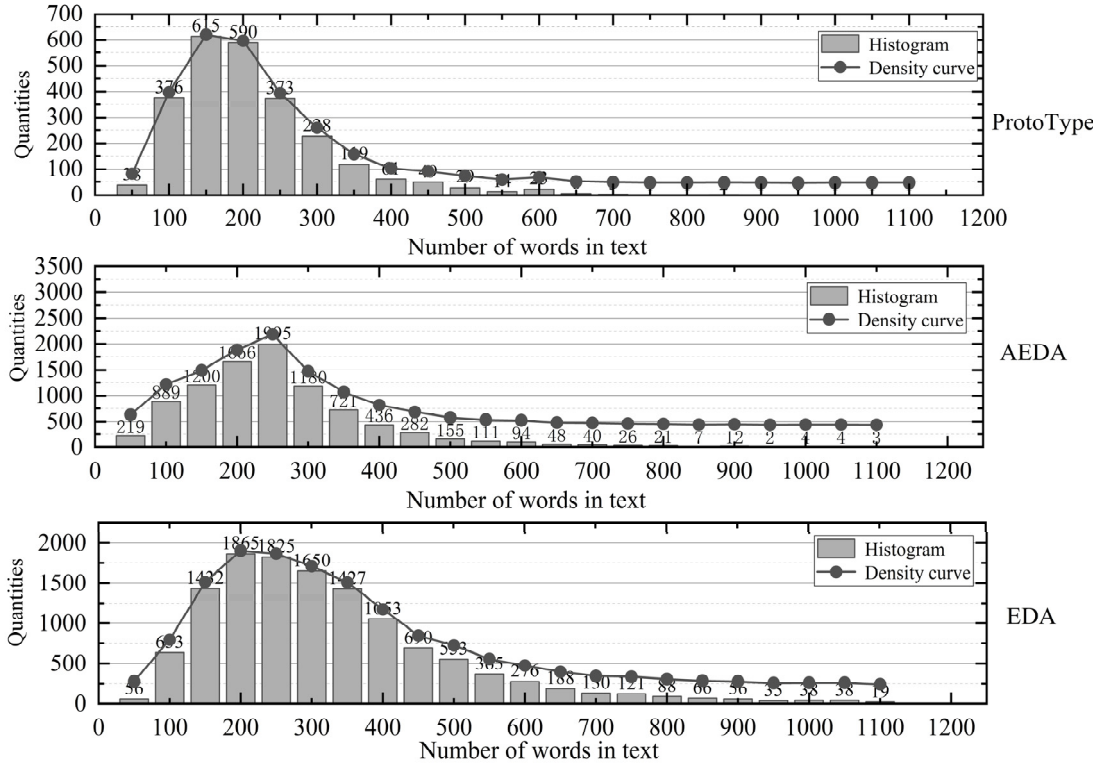

Fig 2. Three samples Number of samples in different sample word ranges

Table 4. Test results for different word length models

Test Model	MacBERT-SA-BiLSTM-TextCNN-SA	
Word Length	**Accuracy**	**F1**
230	94.09%	94.09%
235	94.82%	94.80%
240	93.08%	93.08%
245	94.26%	94.22%
250	94.05%	94.05%

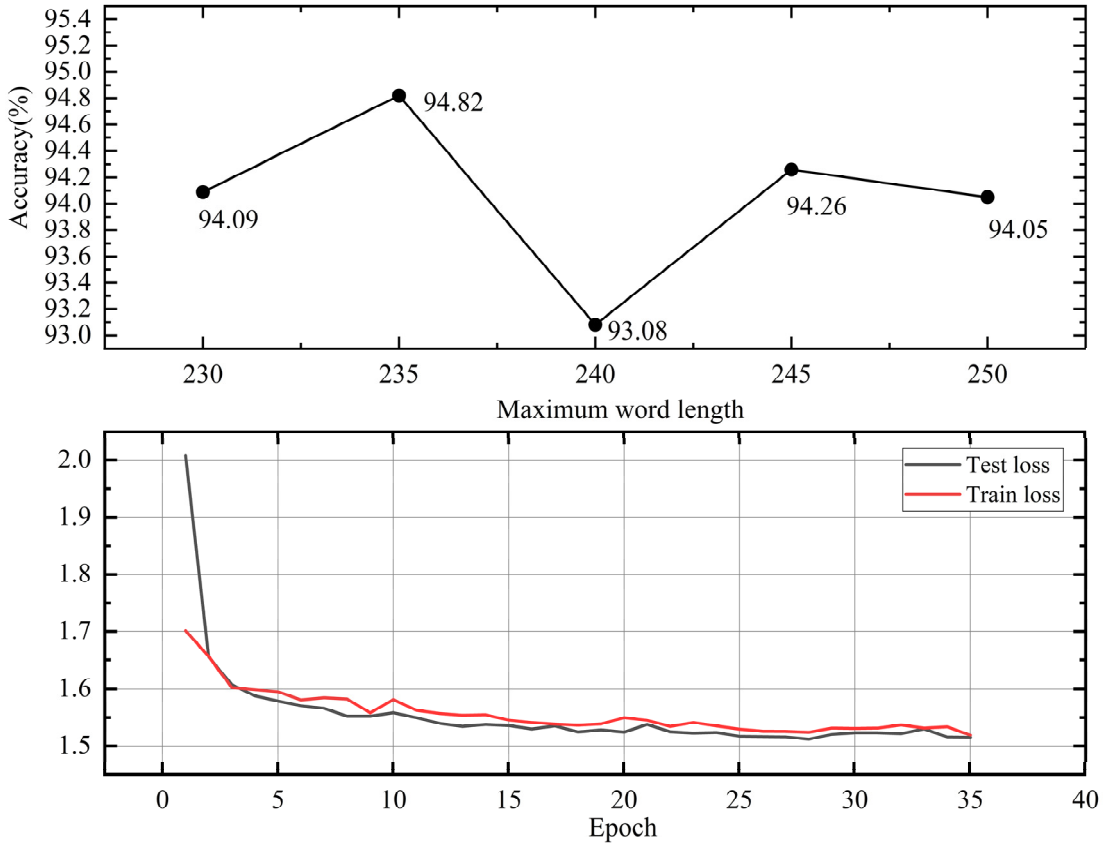

Fig 3. Performance trend diagram for different word length models

The test results show that when the word length is 235, both the Accuracy and F1 values reach their highest levels, with an accuracy rate of 94.82%. A high accuracy and F1 value mean that the model performs well, but it is also important to pay attention to overfitting. If the training loss is much lower than the test loss, it may indicate that the model performs well on the training data but may not generalize well to new, unseen data. The bottom half of (Fig 3) shows the values of training loss and test loss when the word length is 235. From this Fig, it can be seen that the difference between the training loss and test loss is very small and they essentially overlap, which indicates that the experimental results are good and there is no overfitting problem.

3.2. Model Comparison Experiment and Result Analysis

The experiment aims first to verify the performance of the basic model MacBERT-SA-TextCNN-BiLSTM-SA constructed in this paper, and second to validate the rationale of the summary text classification model through comparative experiments. Table 5 presents a comparison of the results of each evaluated model index, and (Fig 4) shows a comparison chart of the performance of the three samples under the accuracy metric.

As shown in (Fig 4), comparing the results of AEDA and EDA with the original sample data reveals that AEDA has a greater effect on improving model performance, with improvements exceeding 10% in all cases. This indicates that when using a pre-trained language model, AEDA data augmentation technology can effectively enhance model performance. A comparison of AEDA and EDA shows that although EDA achieves greater improvement than the original sample, its effectiveness is not as pronounced as that of AEDA across all models, and the AEDA method does not significantly impact the semantics of the original text.

(Fig 5) compares experimental results. The AEDA-enhanced data in comparison group 3 reveals a significant performance gap between TextCNN and BiLSTM, with TextCNN alone performing worst overall. This occurs because literature abstracts contain complex semantics and strong contextual connections. While TextCNN effectively captures local features, it struggles with long-distance information and contextual connections, leading to insufficient information capture. Comparison group 4 shows that combining TextCNN with BiLSTM compensates for BiLSTM's weakness in capturing local features, improving overall performance by 0.98%.

Table 5. Indicator results for the base model and each of the comparison models

Models	Type	Accuracy	F1
MacBERT-SA-TextCNN-BiLSTM-SA	AEDA	94.82%	94.80%
	EDA	92.14%	92.13%
	Original sample	81.43%	80.70%
BERT-SA-BiLSTM-TextCNN-SA (BERT-Chinese)	AEDA	94.46%	94.78%
	EDA	90.82%	90.82%
	Original sample	80.61%	79.05%
MacBERT-TextCNN-BiLSTM	AEDA	93.58%	93.12%
	EDA	90.64%	90.59%
	Original sample	78.42%	78.12%
MacBERT-TextCNN	AEDA	85.73%	83.46%
	EDA	78.60%	71.39%
	Original sample	75.42%	74.05%
MacBERT-BiLSTM	AEDA	92.60%	92.63%
	EDA	89.88%	89.85%
	Original sample	79.59%	80.61%
MacBERT-TextCNN-BiLSTM-SA	AEDA	94.11%	94.32%
	EDA	91.57%	91.52%
	Original sample	81.46%	80.59%

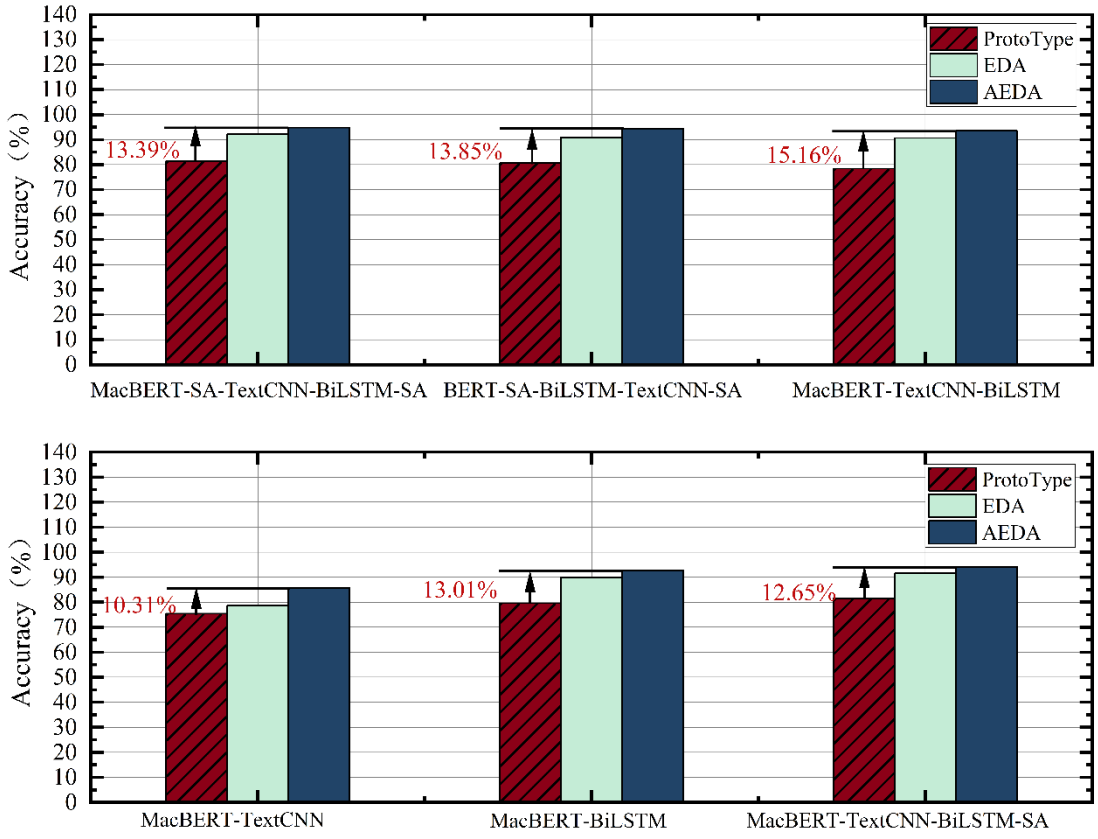

Fig 4. Comparison of model performance under three sample conditions

In comparison group 5, adding a self-attention mechanism at the end of MacBERT-TextCNN-BiLSTM to weight combined feature representations before classification significantly enhances performance by 0.53%. Comparison group 2 demonstrates that adding a self-attention mechanism at the end of the pre-trained language model improves performance by 0.71% by enabling better capture of long-distance dependencies. Comparison group 1 shows MacBERT outperforms traditional BERT in processing summary texts by 0.36%. The MacBERT-SA-TextCNN-BiLSTM-SA model achieves 94.82% accuracy, outperforming all comparison models. The confusion matrix (Fig 6) shows few incorrect predictions with no instances of completely wrong category predictions. By introducing multi-layer Dropout and optimizing feature fusion, we developed the MBCResAttNet model for small-sample text summary classification. Table 8 presents MBCResAttNet's performance compared to MacBERT-SA-TextCNN-BiLSTM-SA across three samples.

(Fig 7) shows the training metrics of the MBCResAttNet model. The top half is a locally zoomed-in plot of the Accuracy and F1 values, and the bottom half is a line graph of the complete training data containing the training loss and test loss. This Fig shows that the difference between the training loss and test loss is very small and essentially overlaps, indicating that the model has good performance and

does not have overfitting problems.

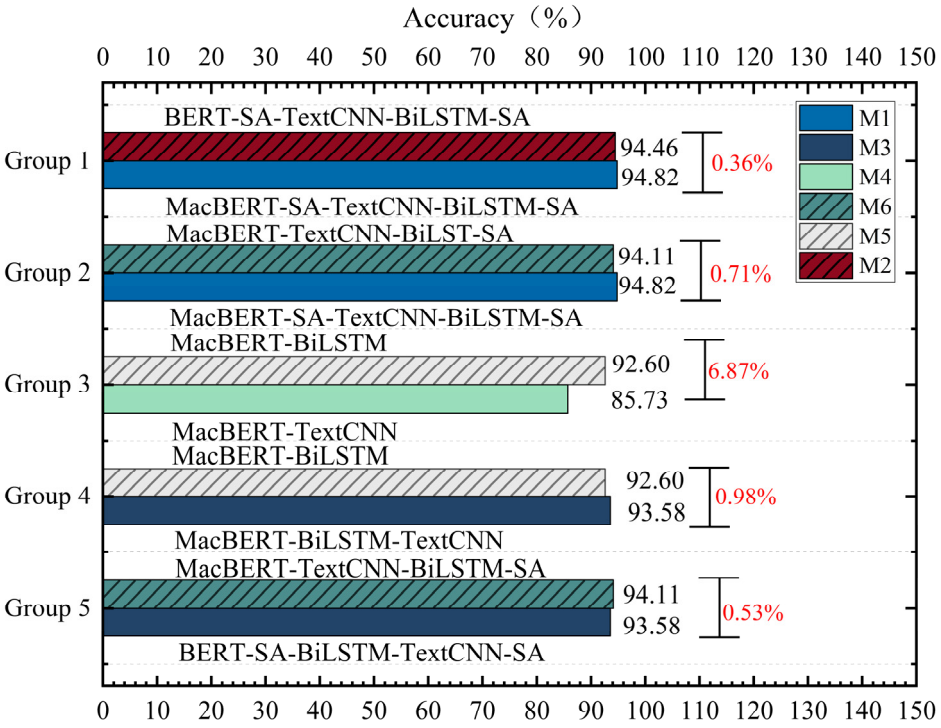

Fig 5. Comparison of models

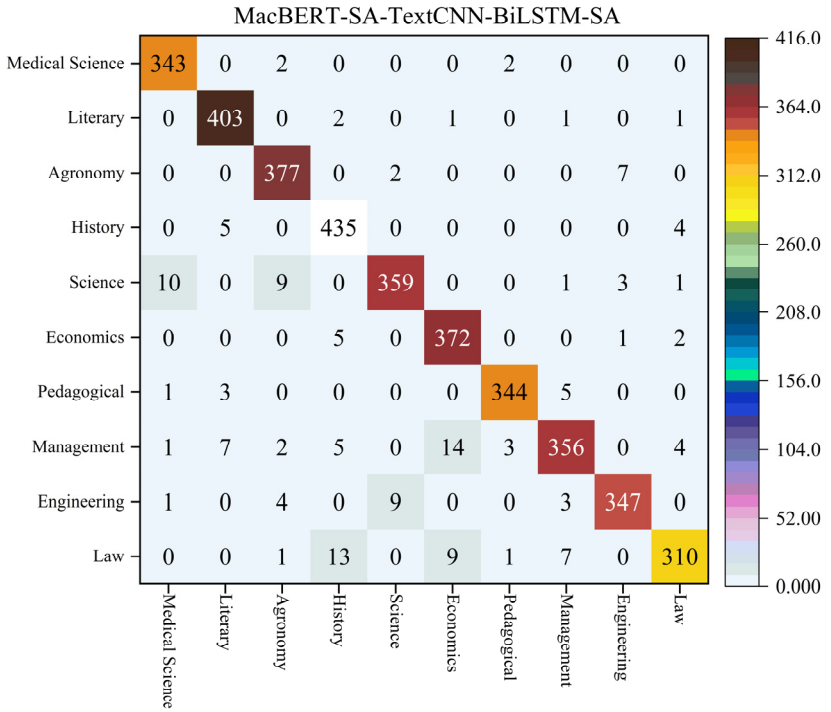

Fig 6. MacBERT-SA-TextCNN-BiLSTM-SA Confusion matrix

Table 6. MBCResAttNet Different sample performance test results

Models	Type	Accuracy	F1
MBCResAttNet	AEDA	96.17%	96.15%
	EDA	93.19%	93.18%
	Original sample	82.76%	81.20%
MacBERT-SA-TextCNN-BiLSTM-SA	AEDA	94.82%	94.80%
	EDA	92.14%	92.13%
	Original sample	81.46%	80.59%

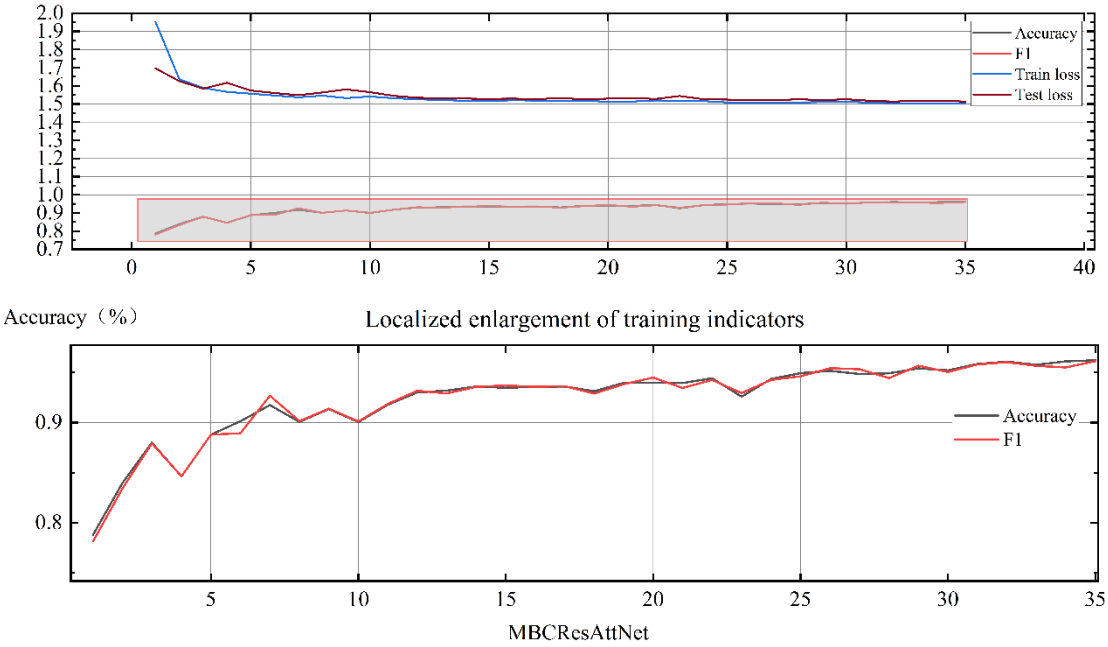

Fig 7. MBCResAttNet model training indicator chart

(Fig 8) illustrates the results of the MBCResAttNet model. The left side shows the corresponding confusion matrix, from which it can be seen that the model predicts few labels incorrectly, indicating high overall accuracy on the classification task. The right side displays the performance comparison under different samples, revealing that across the three types of samples, retaining the original features by residual linkage with the weighted features can effectively improve the model's performance by 1.35% under the AEDA data augmentation technique.

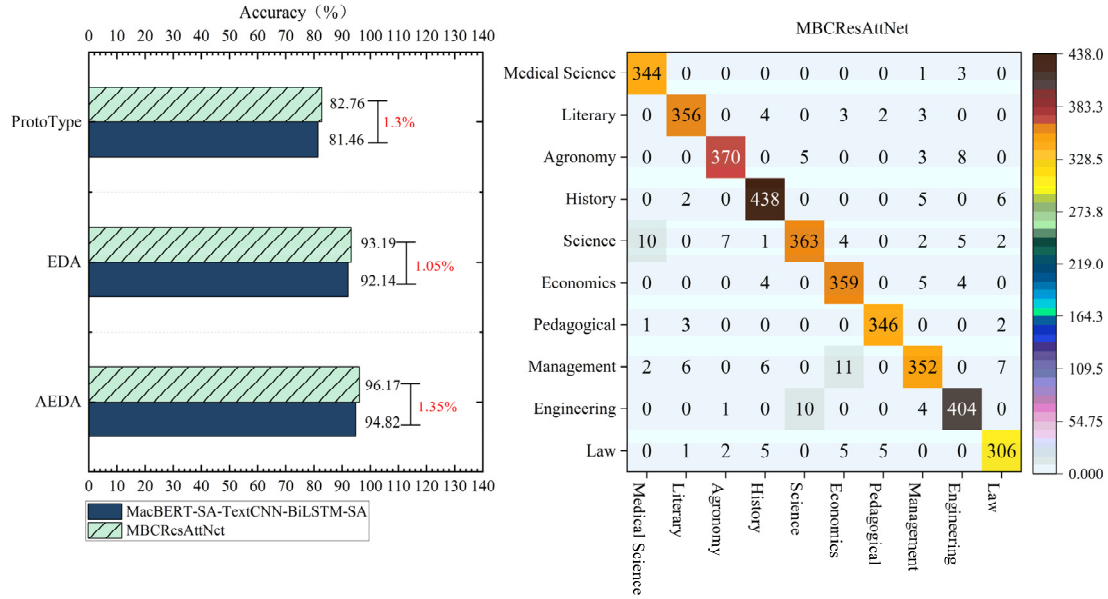

Fig 8. Comparison of MBCResAttNet performance under three sample conditions and confusion matrix diagram

4. Conclusion

Abstract texts with complex linguistic features often demand sophisticated models and extensive training data. Many researchers utilize pre-trained language models like BERT to improve performance. However, when applying EDA for data augmentation on small samples, results can deteriorate as original word order may change. This paper presents a method combining AEDA and MacBERT that preserves word order during augmentation. The proposed MBCResAttNet model significantly outperforms alternatives by integrating self-attention mechanisms at multiple levels—after the pre-trained model and after the downstream task—enabling better comprehension of long Chinese texts with complex contexts. Adding dropout layers prevents local optima and overfitting. By retaining and combining original features with weighted features during fusion, the method ensures complete utilization of feature information while avoiding information loss between layers, achieving strong performance in multi-label small-sample classification. Future improvements could involve domain-specific MacBERT fine-tuning.

Acknowledgments

I would like to extend special thanks to the Graduate Innovation Project of North Minzu University (Project Grant

No.: YCX24177) for funding this project.

References

[1] Kim Y. Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv:14085882. 2014.

[2] Rawat A, Wani MA, ElAffendi M, Imran AS, Kastrati Z, Daudpota SM. Drug adverse event detection using text-based convolutional neural networks (TextCNN) technique. Electronics. 2022;11(20):3336.

[3] Guo B, Zhang C, Liu J, Ma X. Improving text classification with weighted word embeddings via a multi-channel TextCNN model. 2019;363:366-74.

[4] Yang Z, Emmert-Streib F. Optimal performance of Binary Relevance CNN in targeted multi-label text classification. Knowledge-Based Systems. 2024; 284:111286.

[5] Thekkekara JP, Yongchareon S, Liesaputra V. An attention-based CNN-BiLSTM model for depression detection on social media text. Expert Systems with Applications. 2024; 249: 123834.

[6] Devlin J. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.

[7] Deng S, Li Q, Dai R, Wei S, Wu D, He Y, et al. A Chinese power text classification algorithm based on deep active learning. Applied Soft Computing. 2024;150:111067.

[8] Nithya K, Krishnamoorthi M, Easwaramoorthy SV, Dhivyaa C, Yoo S, Cho J. Hybrid approach of deep feature extraction using BERT–OPCNN & FIAC with customized Bi-LSTM for rumor text classification. Alexandria Engineering Journal. 2024;90: 65-75.

[9] Wilkho RS, Chang S, Gharaibeh NG. FF-BERT: A BERT-based ensemble for automated classification of web-based text on flash flood events. Advanced Engineering Informatics. 2024;59:102293.

[10] Hui Y, Du L, Lin S, Qu Y, Cao D, editors. Extraction and classification of tcm medical records based on bert and bi-lstm with attention mechanism. 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM); 2020: IEEE.

[11] Liu Y. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:190711692. 2019.

[12] Chi P-H, Chung P-H, Wu T-H, Hsieh C-C, Chen Y-H, Li S-W, et al., editors. Audio albert: A lite bert for self-supervised learning of audio representation. 2021 IEEE Spoken Language Technology Workshop (SLT); 2021: IEEE.

[13] Sanh V. DistilBERT, A Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. arXiv preprint arXiv:191001108. 2019.

[14] Sun Z, Li X, Sun X, Meng Y, Ao X, He Q, et al. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. arXiv preprint arXiv:210616038. 2021.

[15] Cui Y, Che W, Liu T, Qin B, Yang Z. Pre-training with whole word masking for chinese bert. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2021;29:3504-14.

[16] Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J. Deep learning--based text classification: a comprehensive review. ACM computing surveys (CSUR). 2021; 54(3):1-40.

[17] Reusens M, Stevens A, Tonglet J, De Smedt J, Verbeke W, Vanden Broucke S, et al. Evaluating text classification: A benchmark study. Expert Systems with Applications. 2024:124302.

[18] Taha K, Yoo PD, Yeun C, Homouz D, Taha A. A comprehensive survey of text classification techniques and their research applications: Observational and experimental insights. Computer Science Review. 2024;54:100664.

[19] Shobana J, Murali M. An improved self attention mechanism based on optimized BERT-BiLSTM model for accurate polarity prediction. The Computer Journal. 2023;66(5):1279-94.

[20] Jia C, He H, Zhou J, Li K, Li J, Wei Z. A performance degradation prediction model for PEMFC based on bi-directional long short-term memory and multi-head self-attention mechanism. International Journal of Hydrogen Energy. 2024;60:133-46.

[21] Cai Z, Zhang H, Zhan P, Jia X, Yan Y, Song X, et al. Multi-schema prompting powered token-feature woven attention network for short text classification. Pattern Recognition. 2024;156:110782.

[22] Li G, Zhao X, Wang X. Quantum self-attention neural networks for text classification. Science China Information Sciences. 2024;67(4):142501.

[23] Liu C, Xu X. AMFF: A new attention-based multi-feature fusion method for intention recognition. Knowledge-based systems. 2021;233:107525.

[24] Liu J, Li D, Shan W, Liu S. A feature selection method based on multiple feature subsets extraction and result fusion for improving classification performance. Applied Soft Computing. 2024; 150:111018.

[25] Shorten C, Khoshgoftaar TM, Furht B. Text data augmentation for deep learning. Journal of big Data. 2021;8(1):101.

[26] Wei J, Zou K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:190111196. 2019.

[27] Xia F, Weng Y, Xia M, Yu Q, He S, Liu K, et al., editors. Does BERT Know Which Answer Beyond the Question? China Conference on Knowledge Graph and Semantic Computing; 2021: Springer.

[28] Karimi A, Rossi L, Prati A. AEDA: an easier data augmentation technique for text classification. arXiv preprint arXiv:210813230. 2021.