ISSN 1330-3651(Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20240720001865
Original scientific paper

A PLM-LCN Network-Based Model for e-Library Automatic Classification

Ke LU, Bei ZHENG*, Jingjing SHI

Abstract: Efficient and accurate categorization of Chinese books in digital libraries is still a challenge, and traditional manual methods are difficult to cope with the huge number of books. In this study, a novel Chinese book classification model based on an enhanced BERT architecture is proposed, which contains a pre-trained language model (PLM) and a long-short-time convolutional neural network (LCN) for improved feature extraction. Experimental results showed that the model achieved up to 93.6% for Micro F1, 95.3% for Macro *F*1, 90% for Mac-P, and 91% for Mic-P with an input text length of 256 and a batch size of 32. The results illustrate the model's efficacy in Chinese book classification, offering theoretical advancements in natural language processing applications and practical enhancements in library resource management and user services.

Keywords: BERT model; book classification; informatization; LCN; PLM

1 INTRODUCTION

In today's era of rapid development of informatization and digitization, libraries, as an important institution for knowledge management and dissemination, are facing unprecedented challenges. The exponential growth of information and the accelerated pace of knowledge creation in the modern era present significant challenges for traditional library management models, particularly manual classification methods, in effectively managing the vast electronic resources now available [1]. Existing automatic classification methods still have some significant challenges in dealing with Chinese book classification (BC), including the lack of accuracy of Chinese word separation, the complexity of semantic understanding, and the limited ability to capture the structure and local features of books [2].

The pre-trained language model (PLM) has made substantial advances in the domain of natural language processing, demonstrating the capacity to discern the profound semantics of language across expansive corpora [3, 4]. However, the application of PLM in Chinese BC still faces many challenges, especially in dealing with complex text structures and semantic understanding. Long short-term memory and convolutional networks (LCN) represent another effective feature extraction tool, demonstrating proficiency in the handling of data with spatio-temporal relationships. However, when employed in isolation, its capacity to fully capture the nuanced semantic features inherent to Chinese text is limited. Yildirim M. proposed a novel classification method by combining LCN and Mel frequency cepstrum coefficients. This method is an effective means of improving the detection of heart sounds in cases of cardiac disease. However, there is a need to enhance its complexity and error tolerance.

To solve these problems, this study proposes a novel automatic Chinese BC model combining PLM and LCN, aiming to improve the accuracy and efficiency of classification by enhancing the feature extraction capability. This approach not only theoretically expands the application of natural language processing in Chinese BC, but also provides a more efficient solution for e-libraries, helping to enhance the intelligence of book resource management and improve readers' search and borrowing experience.

This research is divided into four parts. The first part is an analysis and summary of others' research. The second part introduces the design ideas and implementation details of the PLM-LCN model. The third part shows the performance of the model validated by experiments and compares it with the existing methods. The last part summarizes the article and discusses the possible directions of improvement in the future.

2 RELATED WORKS

The informational library model represents a framework for managing, servicing, and operating library functions through the use of contemporary information technology tools. A key area of research within this model is the development of automated BC techniques. In recent years, the rise of deep learning has driven significant technological advancements in BC. Nguyen T. T. S. et al. emphasized the importance of BC rating prediction in capturing readers' attention. Their research team proposed a knowledge-optimized joint feature selection method, incorporating neural networks, which resulted in significant improvements in book classification. The experimental outcomes indicated that this method not only enhanced classification accuracy but also yielded highly accurate rating predictions [5]. However, this approach may exhibit constraints when confronted with a multitude of text types, particularly when dealing with voluminous and intricate datasets. It may also prove inadequate in maintaining the same degree of efficiency. Saraswat M. et al. observed that traditional classification algorithms often overlook the combined factors of book reviews and books, such as word vector correlations in the text. In response, their research team developed a novel BC recommendation model that integrated recurrent neural networks with semantic analysis. Experimental results demonstrated that the model achieved an average accuracy of 84% when tested on several standard datasets [6]. To address the growing volume of unstructured text data in digital

libraries, Mohammed S. H. et al. proposed a digital book subject term classification model incorporating unsupervised Bayesian networks. The experiments showed that this model could efficiently retrieve target texts from large-scale digital data samples while maintaining excellent stability [7]. Although the model demonstrates robust stability in large-scale data processing, the intrinsic nature of unsupervised learning may impose constraints on its deployment in high-precision classification tasks, particularly in scenarios where explicit labelling is a prerequisite. In such cases, its performance may not be optimal. To further optimize data preservation in libraries, Silalahi R. M. P. et al. introduced a novel BC model that combines the decision tree algorithm with data mining techniques. The experimental findings revealed that the model achieved a mean classification prediction accuracy of 89.3% across three different book datasets, a performance that significantly surpasses that of similar classification algorithms [8]. To improve the efficiency of convolutional model in book classification, Watanobe Y. et al. proposed a novel classification model after combining the structural features of codes and programming language classification. Experimental results showed that the model significantly outperformed the traditional sequential model in terms of classification accuracy, especially when the book distinguished between different algorithms in the code [9]. However, the model may need further optimization and improvement when facing more complex contextual semantic understanding.

The PLM is trained on large-scale corpora, enabling it to learn the probability distribution of a language. This capability allows it to be effectively employed across a wide range of domains, including sentiment analysis, text categorization, machine translation, and question-answering systems. Wang B. et al. identified challenges in applying PLMs to biomedicine due to the presence of interdisciplinary features that hindered their effectiveness. To address this, the research team proposed a new biomedical text classification model that integrates cluster analysis methods. Their findings indicated that the improved method outperformed traditional PLM models in effectiveness and is better suited to the demands of modern biomedicine [10]. Given the exponential growth of academic literature in the social sciences, Shen S. et al. emphasized the need for researchers to quickly identify existing methods for investigating related issues. In response, the team enhanced the SciBERT model and developed a novel pre-selection training model. Experimental results demonstrated that this model excelled in tasks such as discipline classification, function identification, and named entity recognition within social science literature [11]. Li P. et al. found that various potential security risks made deep neural networks face great challenges in intellectual property protection. Therefore, the study proposed a secure and reliable black-box watermarking framework for PLM after combining it with PLM technology. Experimental results showed that the framework was robust to watermark removal attacks on intellectual property and secure enough to resist forgery attacks [12]. The LCN architecture combines the strengths of LSTM networks and CNNs, making it particularly adept at handling data with spatio-temporal relationships, such as video and moving images. Paiva E. et al. highlighted the vulnerabilities of conventional electronic information systems, particularly in terms of data loss and delays in processing information requests. To mitigate these issues, the study proposed an optimization model incorporating LCN architecture. The research outcomes showed that this model significantly improved data transmission efficiency within electronic information systems while stabilizing overall system operation [13]. However, in high-complexity text processing tasks, LCN may not perform as well as other more specialized models. In the domain of runoff prediction, Pan H. et al. in order to solve the problem of low feature classification accuracy in brain-computer interface systems for Chinese character speech images, proposed a novel classifier after combining the optical gradient enhancer algorithm and LCN. The experimental results showed that the average classification accuracy of subjects reading Chinese characters silently and simultaneously silently in this new model increased by 5.24% and 12.44%, respectively [14]. This combination does improve the predictive power of the model, but its computational cost and efficiency in dealing with larger data sizes are still issues to be considered.

To summarize, recent years have seen extensive research into the BC of modern informational libraries, with numerous researchers proposing targeted methods. While PLM and LCN have been widely applied in their respective fields, existing automatic BC methods still face challenges in Chinese book classification, particularly regarding the accuracy of Chinese word segmentation and the complexity of Chinese semantic understanding. In response to these challenges, this study aims to combine PLM and LCN to enhance the efficiency of book management and utilization.

3 RESEARCH METHODS

To develop a Chinese BC model for information automation, the study first selects the BERT pre-training model as the foundational framework, using it to construct a conventional automatic classification model for Chinese books. Next, PLM-LCN is introduced to enhance the extraction of book text features, leading to the construction of an optimized Chinese BC model. Additionally, the key modules of the model are thoroughly detailed.

3.1 An Automatic Classification Algorithm for Chinese Books Based on BERT Pre-Training Model

In the field of natural language processing, pre-trained models have shown exceptional performance. Among these, the BERT model stands out for its superior effectiveness across a variety of natural language processing tasks [15, 16]. The structure of the BERT model is illustrated in Fig. 1.

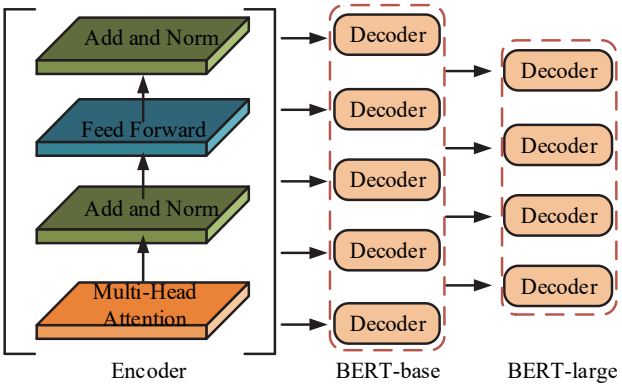

Figure 1 BERT model structure

In Fig. 1, the BERT model is divided into two distinct components: the encoder and the decoder. The encoder part consists of a multi-head attention mechanism and a feed-forward neural network, where each layer includes residual connections and normalization operations [17-19]. The multi-head attention mechanism allows the model to process the input text in parallel in different sub-spaces. Moreover, the feed-forward neural network further processes this information. BERT-base and BERT-large represent two distinct scales of the model, respectively. BERT-large has a greater number of layers of encoders, thereby providing a more nuanced semantic understanding and suitability for more complex text categorization tasks. In the context of BC, the BERT model initially processes the input text data through the encoder, whereby deep semantic representations are extracted. These are then passed to the decoder for subsequent processing and classification. The process of converting text input is illustrated in Fig. 2.

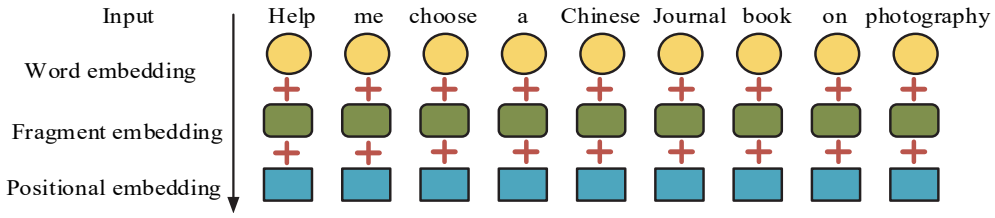

Figure 2 The text input conversion process of BERT

In Fig. 2, the text input conversion process involves three key steps: word embedding, segment embedding, and positional embedding. Word embeddings are responsible for converting each word in the text into a vector representation to capture its semantic information. Fragment embeddings are used to distinguish different sentence fragments to help the model understand the text structure. Positional embeddings provide information about the position of words in a sentence to ensure that the model is able to perceive word order and language. First, word embeddings are used to convert the input text into a vector representation of each word. Next, fragment embeddings are used to distinguish between different sentence fragments to ensure that the model can correctly understand sentence boundaries and paragraph structure. Position embeddings provide information about the position of words in the sentence, allowing the model to perceive word order and structure. The three embeddings are summed to form the final input representations that are fed into the BERT model for processing. This enables the model to capture the overall semantics and structure of the text, and ultimately to classify the book into appropriate categories through the classification layer, e.g., to determine the subject classification of the book based on the title, abstract, or content. The relationship between the three types of embeddings is represented in Eq. (1).

$$Q = Q_t + Q_s + Q_p \tag{1}$$

In Eq. (1), Q, Q_t, Q_s and Q_p denote the text input vector, word vector, segment vector, and position vector, respectively. This combination enables the model to capture the semantic information of the text, the sentence structure, and the relative position of the words in the BC, thereby ensuring that every part of the text is adequately considered. The representation of word vectors is similar to word embeddings in neural networks, where the text sequence to be processed is converted into vector representations of specified dimensions. Assuming the initial text sequence is x, the corresponding word vector representation is provided in Eq. (2).

$$Q_{t,x} = w^t \times P^t \tag{2}$$

In Eq. (2), w denotes the dimension of the word vector, while P represents the word vector matrix. With this formula, the model is able to convert each word in the text into a numerical representation for further processing. Fragment vectors are used to differentiate between various sentences or text fragments, aiding the model in understanding the relationships between individual sentences within a multi-sentence input. Position vectors provide positional information within the sequence, ensuring that the model comprehends the order and relative positions of words in a sentence. When performing text categorization tasks, the workflow of BERT can be divided into three main components: the input layer, the feature extraction layer, and the output layer [20-22]. The input layer is responsible for labeling the initial sequence and mapping it according to the three embedding modules. This process is illustrated in Eq. (3).

$$X = (x_{1-\Diamond}, x_{2-\Diamond}, x_{3-\Diamond}, ..., x_{n-\Diamond}) \tag{3}$$

In Eq. (3), ◊ denotes the labeling symbol, $x_1, x_2, x_3, ..., x_n$ all belong to the initial sequence X, and n represents the length of the sentence. With this mapping, the model is able to recognize sentence boundaries and important tokens in the input text, ensuring the accuracy of text classification. The model mapping at this stage is then represented in Eq. (4).

$$V = Input\,\mathrm{Re}\,presentation(X) \tag{4}$$

In Eq. (4), V denotes the mapped sequence. The BERT model captures the relationships between individual words in a text by mapping the sequence of contextual representations generated. The feature extraction layer is responsible for understanding and extracting features from the input text. Typically, feature extraction can involve 12 to 24 layers, with an increased number of layers enhancing the strength of feature extraction. The formula for feature extraction is provided in Eq. (5).

$$H = BERT(V) \tag{5}$$

In Eq. (5), H represents the contextual representation of the input text. The more layers of the model, the more detailed the extracted features are, providing richer information for classification. The output layer is responsible for generating classification results based on the features extracted by BERT. This process involves recognizing the markup symbols associated with word features, then inputting the recognized text features into a fully connected layer. Finally, the probability distribution of the labeled features is calculated, as shown in Eq. (6).

$$p = Soft\max(h_0 P^0 + b_0) \tag{6}$$

In Eq. (6), h_0 represents the feature of the input to the fully connected layer, P^0 denotes the vector matrix of that feature, and b_0 represents the linear weights. Through this process, BERT translates the feature representations into concrete classification decisions, and in the BC task, this output determines the classification result for each book. In conclusion, this study proposes an informative BC model integrated with the BERT pre-training model. The classification flow of the model is illustrated in Fig. 3.

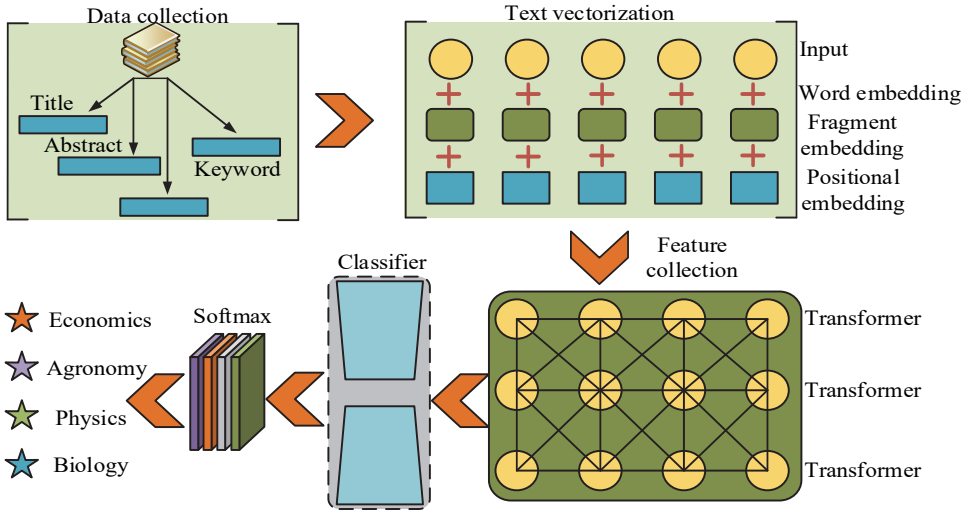

Figure 3 BERT Chinese book automatic classification model

In Fig. 3, the entire classification process is divided into six key steps. First, data collection and organization of library book information are carried out to extract metadata and content summaries of books from library databases and to ensure a relatively balanced sample size for each category. In the pre-processing stage, the text is standardized. This includes the removal of stop words, punctuation marks and special characters, followed by word splitting and stemming extraction. The BERT model begins by inputting a portion of this data for training, converting it into a machine-understandable vector format. This vectorized data is then processed through several layers of the Transformer encoder to refine and extract relevant features. Once these features are extracted, they are passed to the classifier, which computes probability values for different labels. After normalization, the category with the highest probability is selected as the topic label for the data.

3.2 PLM-LCN Optimization of BERT Skeleton for Fine-Grained Book Classification Model Construction

The study addresses the primary need of modern informational libraries: to meet the detailed daily reading classification requirements of teachers and students. To achieve this, the study focuses on optimizing and improving BERT in the direction of fine-grained enhancement. The PLM-LCN feature enhancement method is introduced, leading to the development of a Chinese book automatic classification model that integrates PLM-LCN. The structure of this model is depicted in Fig. 4.

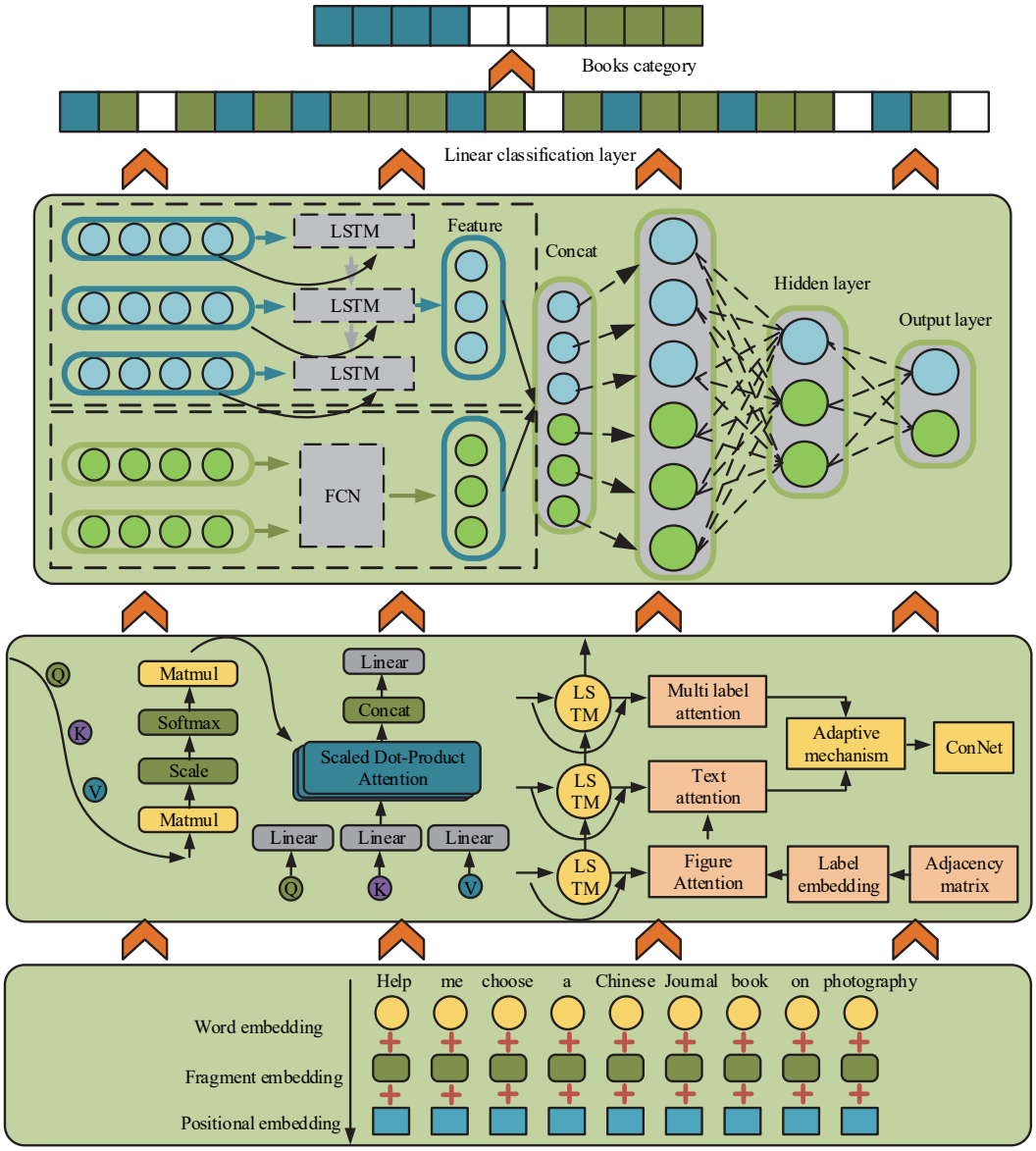

Figure 4 PLM-LCN based automatic classification model for Chinese books

In Fig. 4, the entire PLM-LCN architecture is built upon the BERT model as the core structural foundation. Additional auxiliary modules, such as convolution, pooling, recurrent units, multi-head self-attention, and label attention, are integrated to enhance its capabilities [23-25]. The PLM first encodes the input text to generate a high-dimensional contextual semantic representation. Then, the LCN extracts local features from the semantic representation generated by the PLM and recognizes fine-grained patterns in the text through convolutional operations. Among them, PLM is responsible for generating global semantic features covering the topics of the book content, while LCN refines these features to improve the classification accuracy. Furthermore, PLM is equipped with a multi-language support feature. Through the training of a multi-language corpus, PLM is capable of capturing the semantic relationships and common features between disparate languages. In addition, LCN does not depend on a specific language structure when capturing local features of text, and thus can adapt to different writing systems, such as Latin alphabet, Arabic alphabet, or Chinese characters. For the hyper-parameters, the learning rate is set to 0.001 to balance the training speed of the model with the convergence stability. It ensures that the model can converge in a reasonable time without premature stagnation. The regularization parameter is set to 0.01 to effectively prevent the model from overfitting and improve its generalization ability. The batch size is set to 32 to ensure the stability of the gradient update during training. The optimizer is chosen as Adam, which allows the model to better adapt to the dynamically changing learning environment during training. Throughout this process, the convolution, pooling, and recurrent units primarily serve to optimize parameters and enhance the model's stability. The convolution operation is defined by the formula presented in Eq. (7).

$$C_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I(i+m, j+n)K(m,n)) \tag{7}$$

In Eq. (7), $C_{i,j}$ represents the convolutional output at position i, j, K denotes the convolutional kernel, M and N denote the height and width of the convolution kernel,

respectively, while (m, n) denotes the size of the convolution kernel, and (m, n) indicates the size of the convolutional kernel and I represents the convolutional input data. Localized feature extraction of the input data is achieved by applying the convolutional kernel, which slides over the input data and computes the dot product of the local region. This method effectively reduces the number of parameters while preserving the spatial structure of the data. Additionally, it is highly effective in capturing local dependencies and patterns. The formula for maximum pooling is presented in Eq. (8).

$$P`(i,j) = \mathrm{Max}(W`) \tag{8}$$

In Eq. (8), P' (i, j) denotes the output at position i, j after the pooling operation. W' represents the pooling window. This process ensures that the most significant features are retained by selecting the maximum value within the pooling window. The formula for average pooling, which calculates the average value within the pooling window, is provided in Eq. (9).

$$\bar{P`}(i,j) = \frac{1}{N}\sum W` \tag{9}$$

In Eq. (9), N denotes the number of elements within the pooling window and $P`(i, j)$ represents the average of all values within that window. Average pooling directly reflects the average feature strength of the region, providing a balanced representation of the features within the pooled area. The convolution and pooling processes on a 2D plane are illustrated in Fig. 5.

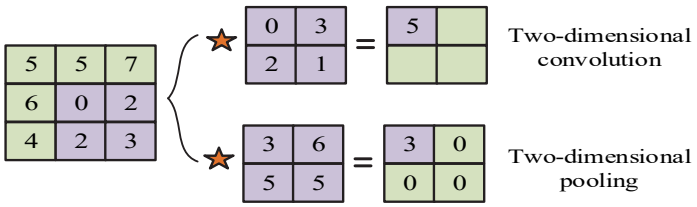

Figure 5 Rolling and pooling operations in two-dimensional planes

In Fig. 5, the initial data size is 3 × 3, with the convolutional kernel having dimensions of 2 × 2, and a sliding step length of 1. The process begins with the convolutional kernel performing feature extraction and dimensionality reduction on the initial data. Next, pooling is applied to screen the feature data, selectively retaining the most important features while discarding redundant information. Additionally, multi-head attention and labeled attention mechanisms further enhance the focus on key features, making the model more targeted and reducing its computational complexity. The linear mapping formula for the multi-head attention mechanism is presented in Eq. (10).

$$\begin{cases} Q_i = QW_i^Q \\ K_i = KW_i^K \\ V_i = VW_i^V \end{cases} \tag{10}$$

In Eq. (10), Q, K and V denote the query task (Query), Q_i, K_i and V_i denote the query vector, key vector and value vector respectively. The corresponding key value (Key) and the task purpose (Value), and W_i^Q, W_i^K and W_i^V denote the learning weights of Q, K and V, respectively. The formula for calculating the multiple fusion is provided in Eq. (11).

$$MultiHead(Q,K,V) = Concat(head_1, head_2, \cdots, head_h)W^T \tag{11}$$

In Eq. (11), W^T denotes the learnable matrix and $head_n$ denotes the n th attention head. The use of multi-head attention allows the model to process a larger volume of data, thereby improving its ability to extract and utilize information effectively. Moreover, traditional average word embedding can cause damage to the labeling of features, leading to the loss of semantic information. To address this, the study incorporates labeled attention with fine-grained processing as a key component. The formula for fine-grained processing is provided in Eq. (12).

$$\begin{cases} \overrightarrow{a}_{i,j} = H_{c,j} \cdot H_{ei}^T \\ \overleftarrow{a}_{i,j} = H_{c,j} \cdot H_{ei}^T \end{cases} \tag{12}$$

In Eq. (12), $\overrightarrow{a}_{i,j}$ and $\overleftarrow{a}_{i,j}$ denote the positive and negative attention scores of the j th word vector in the label and the i th word vector in the book document, respectively. $H_{c,j}$ denotes the semantic representation of the j th word vector in the label and H_{ei}^T denotes the semantic representation of the i th word vector in the document. The semantic component of each word vector in the document is then calculated as shown in Eq. (13).

$$\begin{cases} \overrightarrow{u} = \overrightarrow{A} \cdot H_{ei}^T \\ \overleftarrow{u} = \overleftarrow{A} \cdot H_{ei}^T \end{cases} \tag{13}$$

In Eq. (13), $\overrightarrow{u}$ and $\overleftarrow{u}$ represent the positive and negative semantic components of each word vector in the tag within the document, while $\overrightarrow{A}$ and $\overleftarrow{A}$ denote the positive and negative attention score matrices, respectively. The schematic diagrams illustrating multiple and labeled attention are presented Fig. 6.

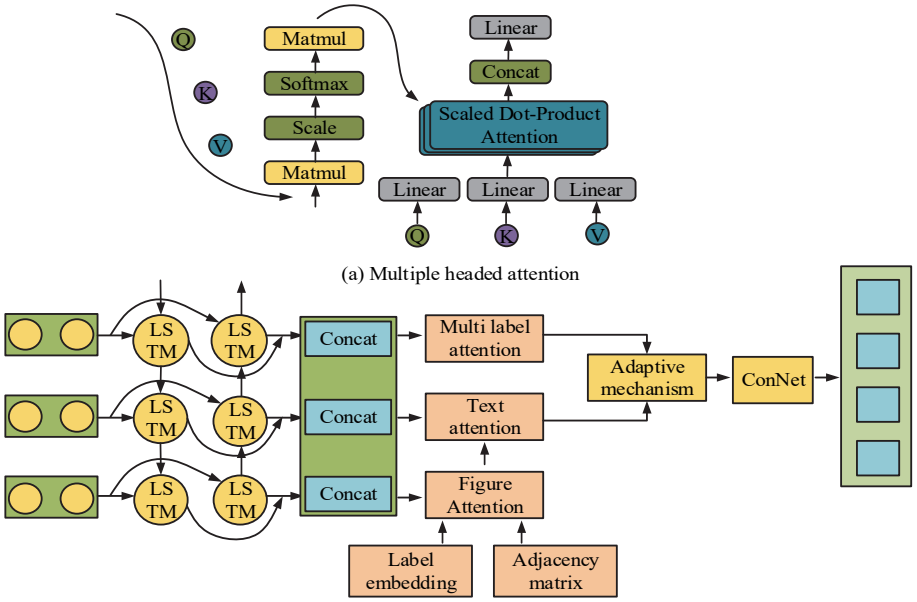

Figure 6 Schematic diagram of multi head attention and label attention

Fig. 6a illustrates the concept of multi-head attention, while Fig. 6b depicts the process of labeling attention. Multi-head attention enhances the model's ability to understand complex data by processing inputs independently across multiple attention heads, allowing the model to focus on different aspects of the data simultaneously. This approach is especially effective for sequence data processing. Label attention, on the other hand, optimizes tasks such as multi-label classification by aligning the attention weights with specific labels and input features, focusing the model's processing on data elements most relevant to particular categories. The text features derived from the attention mechanism are then fed into the classifier, where they are normalized using the Softmax function. At this stage, the loss function is represented in Eq. (14).

$$HLoss = -\sum_{i \in N^*} \sum_{j \in M^*} q` \ln p` \tag{14}$$

In Eq. (14), N^* denotes the number of training samples and M^* denotes the number of labels. N represents the number of training samples and M represents the number of labels. q' denotes the number of true labels and p' denotes the number of book documents.

4 RESULTS AND DISCUSSION

To demonstrate the efficacy of the Chinese BC model proposed in this study, it is first necessary to validate its underlying skeleton model, BERT, to determine the optimal values for its hyper-parameters. Once these optimal values are identified, subsequent tests of the classification model can be conducted. Additionally, the classification models, along with their counterparts, are tested and compared using refined evaluation metrics to verify the superior performance of these models.

4.1 BERT Training Model Performance Test

The experimental setup for this study is based on PyTorch, with the processor being an Intel i7-9700K, the operating system Ubuntu 19.1, 64GB of RAM, and an NVIDIA GeForce RTX 1060s graphics card. Since no public Chinese books dataset is available, the study uses a web crawler to collect data from Douban Books. After data validation, cleaning, and segmentation, a custom dataset containing information on approximately 20000 Chinese books is created, including details such as book titles, publishers, and subject categories. The study begins by optimizing the selection of hyper-parameters for the BERT model. Common metrics in multi-text classification, such as Precision, Recall, and $F1$ score, are used as primary reference metrics. Additionally, two refined Micro-average metrics, Micro $F1$ and Macro $F1$, are employed, considering the characteristics of the dataset. The Micro $F1$ method is a classification approach that prioritizes the evaluation of high-frequency categories. This is achieved by calculating the total precision and total recall for all categories, and subsequently determining the $F1$ value. It is suitable for situations where the distribution of

categories is unbalanced, as it can reflect the overall classification accuracy. In contrast, the Macro $F1$ metric calculates the $F1$ values for each category separately and then averages these $F1$ values. Therefore, it focuses more on the equal treatment of each category, and is more reflective of the model's performance, especially on low-frequency categories. For the experiments, the learning rate is fixed at 0.05, the batch size for input text is set to 12, and the number of training epochs is 12. The study tests the maximum sequence lengths for different input text lengths, with the results presented in Fig. 7.

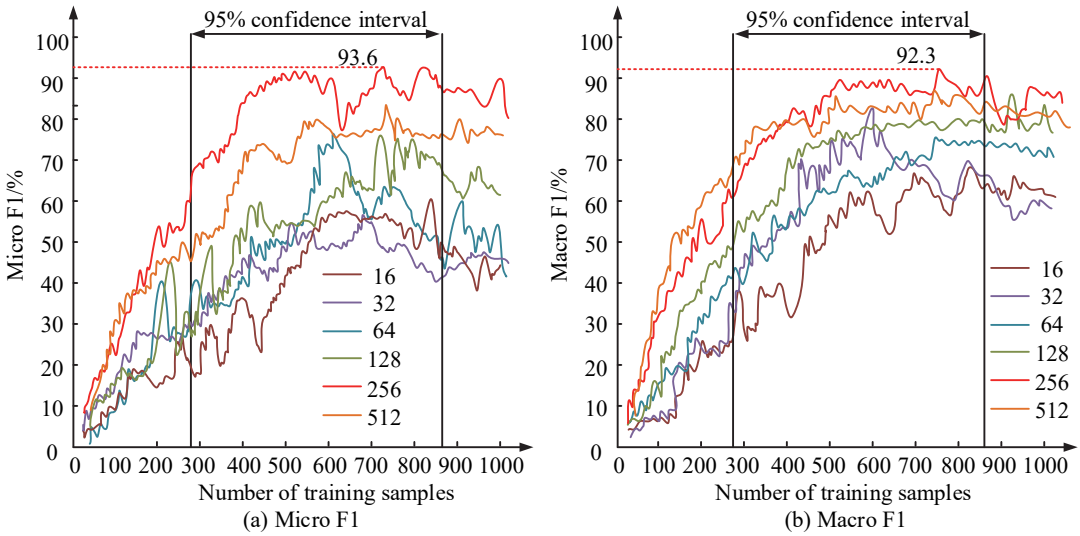

Figure 7 Performance results of indicators with different input text lengths

Fig. 7 presents the performance results of Micro $F1$ values for different input text lengths. In the 16 - 256 interval, the detection performance improves linearly as the input text length increases. However, in the 256 - 512 interval, the detection value begins to decrease. Therefore, the BERT model achieves its best performance when the input text length is set to 256, with Micro F1 reaching a maximum of 93.6% and Macro F1 peaking at 92.3%. This decline beyond 256 is likely due to the truncation of text, which results in the loss of important information. Furthermore, with a fixed learning rate of \(1\times 10^{-5}\), an input text length of 256, and 12 training epochs, both Micro $F1$ and Macro $F1$ are also used as indicators for evaluating different text batch sizes. The results of these tests are shown in Fig. 8.

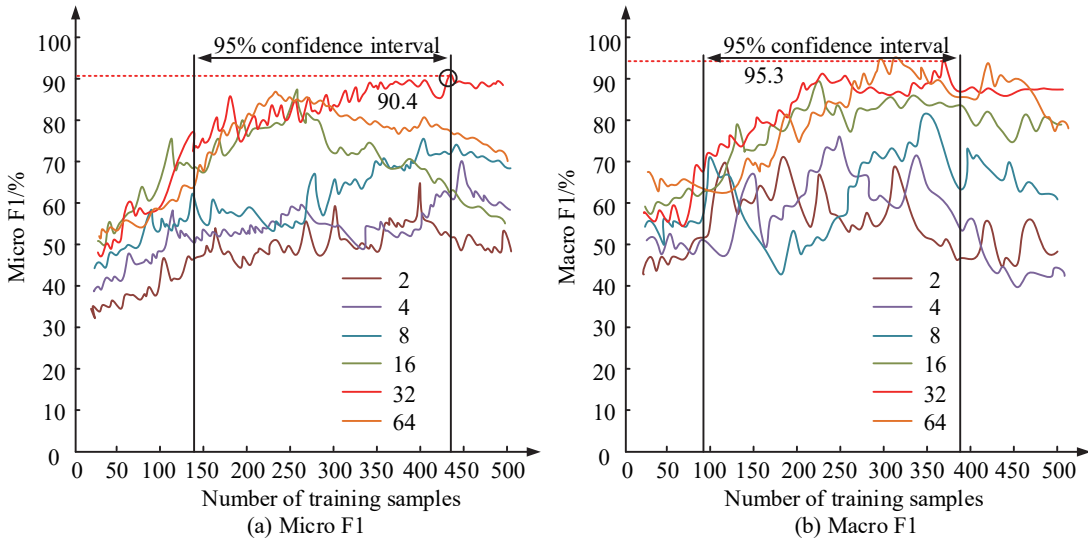

Figure 8 Test results of indicators with the same batch size as the text

Fig. 8 shows the performance results of Macro $F1$ values for different text batch sizes. The results indicate that when the text batch size is 32, the model achieves the highest Micro $F1$ at 90.4% and the highest Macro $F1$ at 95.3%. Although there is a small increase in the Macro $F1$ value with a batch size of 64, its arithmetic stability is poor, and a continuous downward trend follows. This instability occurs because both larger and smaller batch sizes can negatively impact model convergence, leading to increased running time and a decrease in gradient efficiency. While

the model's performance becomes more stable as the number of training iterations increases, this outcome may still be influenced by the unequal distribution of cultural and linguistic elements in the training data, particularly when dealing with cross-cultural content or less prevalent writing systems. Based on these findings, the study fixes the input text length at 256 and the text batch size at 32 as the optimal model hyperparameters. The study then introduces other similar multi-text PLMs, such as continuous bag of words (CBOW), skip-gram (SG), global vector (GV), Word2vec, and the hierarchical Softmax (HS) model. Taking accuracy, recall, F1 value, loss function and number of iterations as reference indexes, the test results are tested statistically and the results are shown in Tab. 1.

Table 1 Test results of indicators for different pre trained models

Model	*P*	*R*	*F*1	Loss	Iterations	Processing time/s	*P*
CBOW	85.30%	88.63%	87.03%	8.93	782	2.11	0.003
SG	89.91%	90.58%	90.27%	6.77	679	2.04	0.002
GV	93.44%	92.13%	92.88%	4.62	648	1.25	0.001
Word2vec	93.56%	91.42%	92.37%	7.68	789	1.63	0.003
HS	90.48%	92.79%	91.73%	9.83	592	1.59	0.006
BERT	97.86%	96.61%	96.97%	1.51	325	1.07	0.004

Tab. 1 presents the results of the six multi-text pre-training models. The BERT model proposed in this study achieved the highest overall evaluation scores and demonstrated the most favorable performance across all metrics. Specifically, the BERT model exhibited the highest precision (*P*) value at 97.86%, the highest recall (*R*) value at 96.61%, the highest F1 value at 96.97%, the lowest loss value at 1.51, and the fewest iterations at 325. In particular, the processing time of the proposed model is found to be the shortest at 1.07 s, which still represents a reduction of 0.16 s compared to the GV model. It is the best performer among the other five classes of models. It indicates that the study of the proposed model has a faster image recognition and processing speed among many models. Other models such as GV and Word2vec, despite showing significance in *P*-value ($P = 0.001$ and $P = 0.003$), their *F*1 values, 92.88% and 92.37% respectively still lag behind BERT. It indicates that BERT performs better in terms of balancing precision and recall. These results indicate that the BERT model, as the foundational framework of the Chinese BC model, offers significant superiority and reliability. This strong performance suggests that the BERT model has the potential to directly influence and enhance the outcomes of subsequent classification tasks. It is also possible that these results are influenced by biases present in the training dataset. For instance, the training data may exhibit a high prevalence of books from specific subject areas or cultural backgrounds, accompanied by a relatively low representation of books from other areas or backgrounds. This imbalance may result in the model under-performing in real-world applications when confronted with these less common categories.

4.2 PLM-LCN Chinese Book Classification Optimization Model Performance Test

Based on the previous tests, the input text length is set to 256, the text batch size to 32, the iteration parameter to 325, the learning rate to \(1\times 10^{-5}\), and the optimizer selected as AdamW. Ablation tests are then performed on the PLM-LCN model, which includes testing the individual PLM module, CNN module, LSTM module, as well as the combined PLM-CNN, PLM-LSTM, and PLM-LCN models. To address the potential non-uniformity in the distribution of discipline data within the homemade dataset, the study introduces Macro-average, Micro-average, and Weight-average metrics to further subdivide the precision (*P*) and recall (*R*) values. This approach aims to eliminate any bias caused by the uneven distribution of disciplines in the initial dataset. The outcomes of these tests are presented in Tab. 2.

Tab 2 Results of ablation tests for each module of PLM-LCN model

Module	Mac-*P*	Mac-*R*	Mic-*P*	Mic-*R*	Loss	*P*
BERT	80.43%	81.17%	82.47%	83.67%	2.17	0.002
PLM	81.43%	81.41%	84.79%	83.52%	3.67	0.003
CNN	85.31%	83.47%	82.52%	80.79%	5.18	0.001
LSTM	80.27%	82.91%	83.11%	83.68%	5.03	0.004
PLM-CNN	86.79%	86.37%	85.17%	86.68%	2.83	0.002
PLM-LSTM	87.66%	84.58%	88.94%	87.74%	2.47	0.003
PLM-LCN	92.63%	93.43%	92.18%	90.18%	0.97	0.003

In Tab. 2, the performance tests for each module of the PLM-LCN model show that most individual modules achieve test values above 80%. However, the PLM-LCN model stands out with the highest macro-average precision (Mac-*P*) at 92.63%, the highest macro-average recall (Mac-*R*) at 91.43%, the highest micro-average precision (Mic-*P*) at 92.18%, the highest micro-average recall (Mic-*R*) at 90.18%, and the lowest loss value at 0.97. Statistical analysis shows that the P-values of all modules are less than 0.05, indicating that the performance differences between different modules are statistically significant. In particular, the performance of the PLM module is relatively deficient. While it demonstrates moderate proficiency in terms of micro-accuracy, it exhibits a loss value of 3.67, indicating that a degree of error accumulation occurs during the model training process. Moreover, the combination of PLM and LCN (i.e., the PLM-LCN model) has been demonstrated to not only

effectively reduce loss value but also significantly improve performance indexes. This suggests that LCN is an effective complement to PLM in capturing local features, particularly in enhancing feature expression ability and improving classification robustness. These results indicate that the feature extraction optimization functions of the PLM-CNN and PLM-LSTM modules significantly enhance the overall performance of the model. To further evaluate the effectiveness of the PLM-LCN model, the study introduces other popular Chinese BC methods for comparison, including linear discriminant analysis (LDA), support vector machine (SVM), and extreme learning machine (ELM). Additionally, three long-text datasets are incorporated as data sources: the Super Star eBook (SSReader) database, the China Basic Ancient Books Library, and the Peking University Law database. The results of these comparisons are presented in Fig. 9.

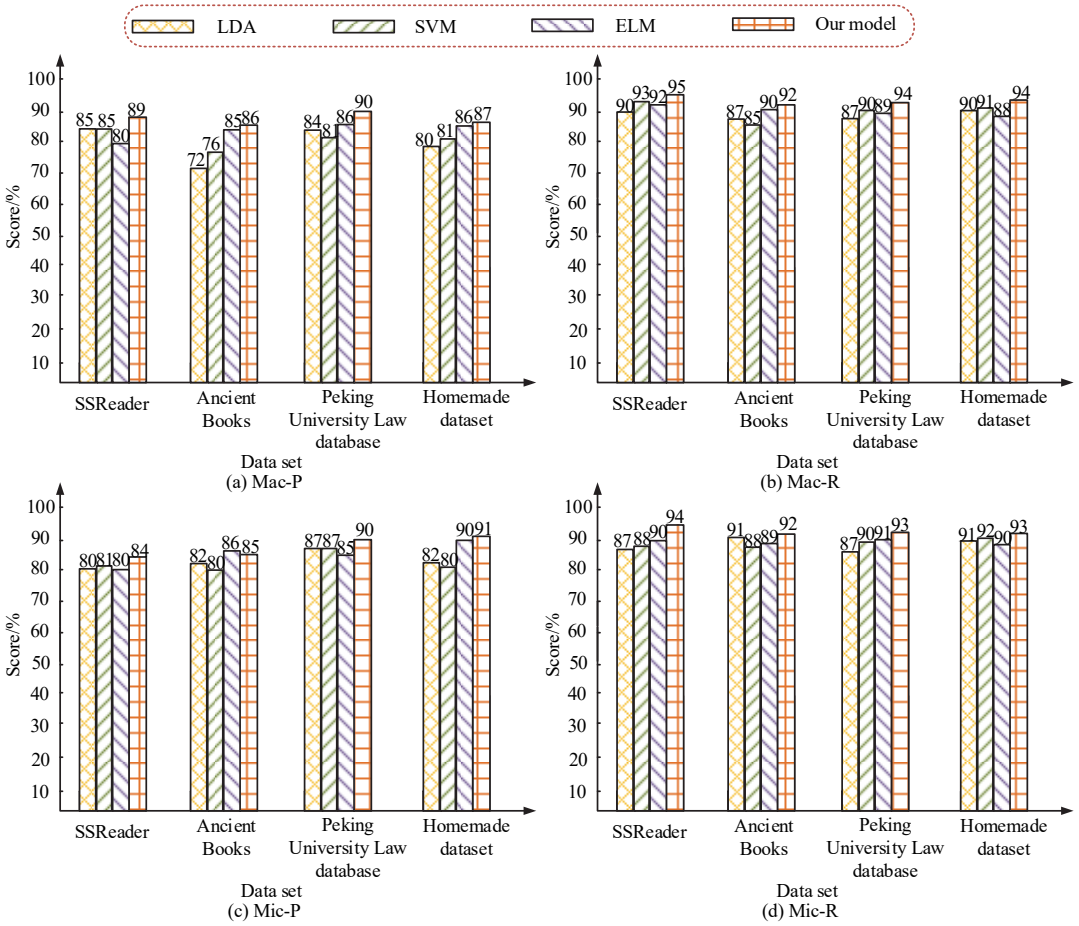

Figure 9 Performance test results of different multi text classification models

Fig. 9a, Fug. 9b, Fig. 9c, and Fig. 9d illustrate the test results for Mac-*P*, Mac-*R*, Mic-*P*, and Mic-*R* values across four models under four different datasets. The proposed models in the study consistently demonstrate superior performance, with general test values exceeding 85 across all metrics. Specifically, the highest Mac-*P* is 90% in the NLF dataset, the highest Mac-*R* is 95% in the SSReader dataset, the highest Mic-*P* is 91% in the Homemade dataset, and the highest Mic-*R* is 94% in the SSReader dataset. This shows that the proposed model of the study is more advantageous than the state-of-the-art LDA, SVM, and ELM methods in image classification tasks, and exhibits higher accuracy and stability especially under complex conditions. Given these results, the SSReader dataset, which showed the highest overall performance, is selected as the primary data source. From this dataset, core subject journals related to four disciplines—management, finance, mathematics, and computer science—are chosen. Classification confusion matrices are then constructed for each of the four models. The results of these confusion matrices are presented in Fig. 10.

Fig. 10a, Fig. 10b, Fig. 10c, and Fig 10d depict the confusion matrices for the LDA model, SVM model, ELM model, and the proposed model, respectively. The LDA model has 58 correct classifications in the finance and economics category, but performs poorly in the computer science category, correctly classifying only 54 books. This indicates that the LDA model has a more limited classification ability when dealing with categories with large differences. The SVM model demonstrates marginal superiority in the Mathematics category, with 56 books correctly classified. However, its performance in the Computer Science category is markedly inferior, with only 42 correctly classified books. This illustrates the constraints of SVM in addressing complex data sets. The model proposed in the study performs well in all categories, especially in the computer science category with a correct classification number of 59, which is the best

performance of all models. For the management and finance and economics categories, the research model also achieved 57 and 59 correct classifications respectively, which is slightly higher than the other models. Therefore, it can be shown that the proposed model of the study is more suitable for categorizing Chinese books in information technology libraries.

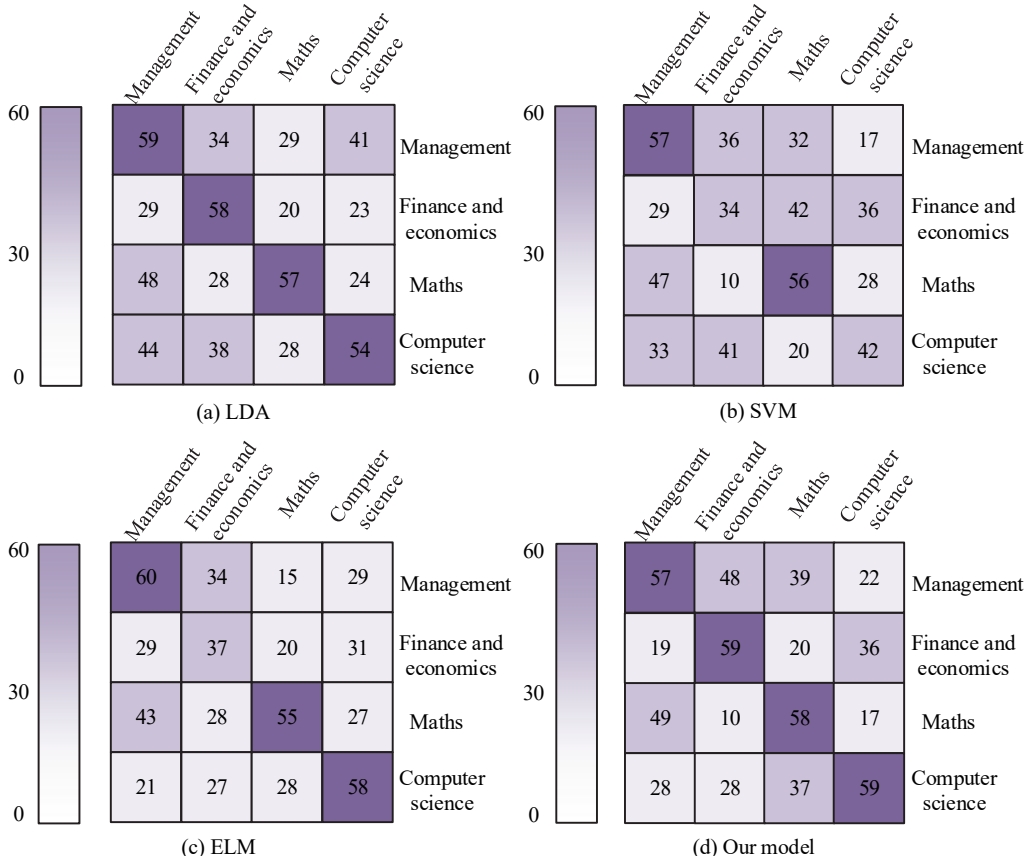

Figure 10 The results of different subject book classification by four models

5 DISCUSSION

The proposed model demonstrated significant advantages in the BC task. For example, in the BC task, the study's method in achieved 85.3% accuracy compared to the traditional method, which was an improvement of 7.1 percentage points compared to the current state-of-the-art model. This enhancement not only improved macro *F*1 by 6.8% and micro *F*1 by 6.5%, but also significantly enhanced the model's classification robustness. Many methods in the existing literature suffered from dataset bias problems. For example, Datta D. et al. argued that the study performed unevenly in the task of book categorization across different cultural backgrounds or time periods, whereas the study proposed to mitigate these problems effectively by integrating linguistic properties and contextual information [26]. First, the improved classification accuracy would enhance the resource management efficiency of libraries, enabling users to retrieve the desired books more quickly and accurately. In addition, the robustness of the model enhanced the stability of the system when dealing with diverse book data, which was especially important for expanding digital library resources. However, although the model showed good adaptability in different languages and cultural contexts, there may still be some special cases that are not covered. Further extensions of the model may be undertaken in the future, incorporating additional features and data sources, particularly when dealing with different book types and complex writing systems. This will enhance the model's generalizability and application scope.

6 CONCLUSION

In this study, an innovative approach to the automatic classification of Chinese books in digital libraries is proposed as a means of addressing the limitations of traditional manual methods. By integrating a PLM and a long-short-term convolutional neural network with the BERT architecture, a model with exemplary performance in the BC task has been developed. The model shows significant improvements in classification accuracy and efficiency across a variety of metrics. The present study makes a significant contribution to both the theoretical advancement of natural language processing in librarianship and the practicality of library resource management. However, despite the remarkable results, the study still has some limitations. For example, the performance of the model in processing books from different periods or different cultural backgrounds may be affected by potential biases in the training data. Moreover, the model has been trained primarily on a single language,

and its capacity for generalization in a multilingual setting has yet to be fully validated. It is recommended that future research directions include extending the model to other language environments and exploring the incorporation of more diverse data, such as author, publication year, and geographic region, in the classification process. This approach may enhance the accuracy and applicability of the classification.

7 REFERENCE

[1] Benítez-Andrades, J. A., Alija-Pérez, J. M., & Vidal, M. E. (2022). Traditional machine learning models and bidirectional encoder representations from transformer (BERT)-based automatic classification of tweets about eating disorders: Algorithm development and validation study. *JMIR medical informatics, 10*(2), 34492-34493. https://doi.org/10.2196/34492

[2] Elmitwally, N. S. & Alsayat, A. (2020). The multi-class classification for the first six surats of the Holy Quran. *International Journal of Advanced Computer Science and Applications, 11*(1), 327-332. https://doi.org/10.14569/IJACSA.2020.0110141

[3] Tuba, I., Veinovic, M., Tuba, E., Hrosik, R. C., & Tuba, M. (2022). Tuning convolutional neural network hyperparameters by bare bones fireworks algorithm. *Studies in Informatics and Control, 31*(1), 25-35. https://doi.org/10.24846/v31i1y202203

[4] Yildirim, M. (2022). Automatic classification and diagnosis of heart valve diseases using heart sounds with MFCC and proposed deep model. *Concurrency and Computation: Practice and Experience, 34*(24), 7232-7233. https://doi.org/10.1002/cpe.7232

[5] Nguyen, T. T. S. & Do, P. M. T. (2022). Classification optimization for training a large dataset with Naïve Bayes. *Journal of Combinatorial Optimization, 40*(1), 141-169. https://doi.org/10.1007/s10878-020-00578-0

[6] Saraswat, M. & Srishti (2022). Leveraging genre classification with RNN for Book recommendation. *International Journal of Information Technology, 14*(7), 3751-3756. https://doi.org/10.1007/s41870-022-00937-6

[7] Mohammed, S. H. & Al-augby, S. (2020). Lsa & lda topic modeling classification: Comparison study on e-books. *Indonesian Journal of Electrical Engineering and Computer Science, 19*(1), 353-362. https://doi.org/10.11591/ijeecs.v19.i1.pp353-362

[8] Watanobe, Y., Rahman, M. M., & Amin, M. F. I. (2023). Identifying algorithm in program code based on structural features using CNN classification model. *Applied Intelligence, 53*(10), 12210-12236. https://doi.org/10.1007/s10489-022-04078-y

[9] Wang, B., Xie, Q., Pei, J., Chen, Z., Tiwar, P., Li, Z., & Fu, J. (2023). Pre-trained language models in biomedical domain: A systematic survey. A*CM Computing Surveys, 56*(3), 1-52. https://doi.org/10.1145/3611651

[10] Shen, S., Liu, J., Lin, L., Huang, Y., Zhang, L., Liu, C., Feng, Y., & Wang, D. (2023). SsciBERT: A pre-trained language model for social science texts. *Scientometrics, 128*(2), 1241-1263. https://doi.org/10.1007/s11192-022-04602-4

[11] Paiva, E., Paim, A., & Ebecken, N. (2021). Convolutional neural networks and long short-term memory networks for textual classification of information access requests. *IEEE Latin America Transactions, 19*(5), 826-833. https://doi.org/10.1109/TLA.2021.9448317

[12] Li, P., Cheng, P., Li, F., Du, W., Zhao, H., & Liu, G. (2023). Plmmark: a secure and robust black-box watermarking framework for pre-trained language models. *Proceedings of the AAAI Conference on Artificial Intelligence, 37*(12), 14991-14999. https://doi.org/10.1609/aaai.v37i12.26750

[13] Song, C., Sun, Z., Wang, H., & Tian, Y. (2023).Event-Triggered Piecewise Continuous Tracking Control of Networked Control Systems Using Linear Perturbed System Models with Time Delays. *Studies in Informatics and Control, 32*(4), 17-26. https://doi.org/10.24846/v32i4y202302

[14] Pan, H., Li, Z., Tian, C., Wang, L., Fu, Y., Qin, X., & Liu, F. (2023). The LightGBM-based classification algorithm for Chinese characters speech imagery BCI system. *Cognitive Neurodynamics, 17*(2), 373-384. https://doi.org/10.1007/s11571-022-09819-w

[15] Preethi, P. & Mamatha, H. R. (2023).Region-Based Convolutional Neural Network for Segmenting Text in Epigraphical Images. *Artif. Intell, 1*(2), 119-127. https://doi.org/10.47852/bonviewAIA2202293

[16] Cîrnu, C. & Georgescu, A. (2023).Complex System Governance Theory and Conceptual Links to Cyber Diplomacy. *Studies in Informatics and Control, 32*(2), 127-136. https://doi.org/10.24846/v32i2y202312

[17] Ay, Ş., Ekinci, E., & Garip, Z. (2023).A comparative analysis of meta-heuristic optimization algorithms for feature selection on ML-based classification of heart-related diseases. *The Journal of Supercomputing, 79*(11), 11797-11826. https://doi.org/10.1007/s11227-023-05132-3.

[18] Wu, Z., Xie, J., Shen, S., Lin, C., Xu, G., & Chen, E. (2023).A confusion method for the protection of user topic privacy in Chinese keyword-based book retrieval. *ACM transactions on asian and low-resource language information processing, 22*(5), 1-19. https://doi.org/10.1145/3571731.

[19] Attou, H., Guezzaz, A., & Benkirane, S. (2023). Cloud-based intrusion detection approach using machine learning techniques. *Big Data Mining and Analytics, 6*(3), 311-320. https://doi.org/10.26599/BDMA.2022.9020038

[20] Mahbobi, M., Kimiagari, S., & Vasudevan, M. (2023). Credit risk classification: an integrated predictive accuracy algorithm using artificial and deep neural networks. *Annals of Operations Research, 330*(1), 609-637. https://doi.org/10.1007/s10479-021-04114-z

[21] Cunha, W., Viegas, F., França, C., Rosa, T., & Rocha, L. (2023). A Comparative Survey of Instance Selection Methods applied to Non-Neural and Transformer-Based Text Classification. *ACM Computing Surveys, 55*(13), 1-52. https://doi.org/10.1145/3582000

[22] Watanabe, K. & Baturo, A. (2024). Seeded sequential LDA: A semi-supervised algorithm for topic-specific analysis of sentences. *Social Science Computer Review, 42*(1), 224-248. https://doi.org/10.1177/08944393231178

[23] Goel, L. & Nagpal, J. (2023). A systematic review of recent machine learning techniques for plant disease identification and classification. *IETE Technical Review, 40*(3), 423-439. https://doi.org/10.1080/02564602.2022.2121772

[24] Hossin, M. M., Shamrat, F. M. J. M., & Bhuiyan, M. R. (2023). Breast cancer detection: an effective comparison of different machine learning algorithms on the Wisconsin dataset. *Bulletin of Electrical Engineering and Informatics, 12*(4), 2446-2456. https://doi.org/10.11591/eei.v12i4.4448

[25] Rajwar, K., Deep, K., & Das, S. (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. *Artificial Intelligence Review, 56*(11), 13187-13257. https://doi.org/10.1007/s10462-023-10470-y

[26] Datta, D., Bhattacharya, M., Rajest, S. S., Shynu, T., Regin, R., & Priscila, S. (2023).Development of predictive model of diabetic using supervised machine learning classification algorithm of ensemble voting. *International Journal of Bioinformatics Research and Applications, 19*(3), 151-169. https://doi.org/10.1504/IJBRA.2023.133695

Contact information:

Ke LU
Zhejiang University of Water Resources and Electric Power,
Hang Zhou, China, 310018
E-mail: luke@zjweu.edu.cn

Bei ZHENG
(Corresponding author)
Zhejiang Tongji Vocational College of Science and Technology,
Hang Zhou, China, 311231
E-mail: zhengbei@zjtongji.edu.cn

Jingjing SHI
Taizhou Vocational & Technical College,
Taizhou, China, 318000
E-mail: janejjshi@tzvtc.edu.cn