Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105
Volume **8**, Number **1**, February **2023**

Multi-label Text Classification Combining BERT and Bi-GRU Based on the Attention Mechanism

Ying Tian

Continuing Education School
Ji Ning University
Jining, ShanDong, 273155, China
suning010101@163.com

Received July 20, 2022, revised August 21, 2022, accepted November 16, 2022.

Abstract. *Aiming at the problem of complex correlation between multiple labels in text classification tasks, a multi-label text classification (MLTC) method that combines BERT and label semantic attention is proposed. Firstly, the context vector representation of the input text is learned by fine-tuning the self-encoding pretrained model BERT (Bidirectional Encoder Representations from Transformers). Then, a Bidirectional Gated Recurrent Unit (Bi-GRU) neural network is used to specifically encode the labels. Finally, the attention mechanism is used to explicitly highlight the contribution of the text to each label to predict multi-label sequences. The experimental results show that on the book cataloging dataset, the performance of the proposed method for single-label and multi-label text classification is better than the advanced Fasttext and SGM algorithms, and the accuracy of single-label classification is about 4.8% and 1.9% higher than that of Fasttext and SGM models, respectively. In the multi-label text classification (MLTC) task on the APPD public dataset, the proposed method also achieves the highest F1-score, outperforming other state-of-the-art methods, indicating the effectiveness of the proposed method in fusing the intrinsic relationship between labels and text through the attention mechanism.*
Keywords: MLTC; BERT; Label semantic information; Bi-GRU; Attention mechanism

1. **Introduction.** Multi-Label Text Classification (MLTC) is one of the most important sub-topics in the field of Natural Language Processing (NLP) field [1]. Taking the book classification of digital library as an example, with the rapid development of information technology and automation technology, the digitization of book resources and the automation of library workflow have become the primary tasks of smart library construction [2]. In the daily work of the library, book cataloging is the basis for the digitization of book resources, and it is also a complicated task. It is necessary to determine the category numbers of the books under the existing knowledge-based classification system, such as the "Chinese Library Classification (CLC)", so as to realize the effective management of huge amount of book resources.

With the increase of marginal subjects and interdisciplinary subjects, book and document cataloging becomes more and more difficult. In the past, book category numbers were generally given by authors or manually determined by book catalogers. There are some problems with traditional approaches: Firstly, the determination of book category numbers requires high professionalism, and book authors have a certain degree of subjectivity, their professionalism in bibliographic classification may be insufficient, so category

numbers determined from this are not necessarily conducive to book management; Secondly, catalogers judge the classification numbers by reading the book content, which is time-consuming and labor-intensive,, resulting in a waste of human resources.

With the development and progress of deep learning methodology, MLTC techniques have gradually matured and can be applied to many scenarios in real life, such as text classification [3], opinion extraction [4], and recommendation systems [5]. Different from the Multi-Class Text Classification (MCTC) task, in which each data is only associated with a single label, MLTC assigns multiple labels to each text data, and the topics of the each document are summarized and refined based on multiple labels. Under the limited computing resources, MLTC tasks pose a huge challenge for constructing efficient classification models, among which there are problems such as a large number of labels and an unbalanced sample space. At present, deep learning methods have made great progress in addressing the scalability and label sparsity of MLTC problems [6].

The primary problem of MTLC is how to extract features from text efficiently. In the early days, NLP researchers mainly used the bag-of-words model to form a vector representation by counting the number of words appearing in the text. However, due to the huge number of texts nowadays, manual text classification is obviously not advisable. With the development and application of machine learning and deep learning, related methods are more and more applied to text classification in which training data consisting of vectors and labels are input into machine learning algorithms to train classification models.

With the development of neural network [7,8] and word embedding technology, deep learning has achieved great success by learning the vector representation of text. The widely used models include Word2vec [9] which effectively combines contextual information, GloVe (Global Vector) [11], ELMo (Embeddings from Language Models) based on Bidirectional Long Short-Term Memory (Bi-LSTM) neural network model [12], and Transformer with attention mechanism. In recent years, the field of NLP research is undergoing a milestone development, that is, BERT (Bidirectional Encoder Representations from Transformers), a pre-trained deep language representation model [13]. BERT has achieved state-of-the-art performance in many NLP tasks, such as reading comprehension, text classification, sentiment analysis, information extraction, etc. However, in the MLTC task, the BERT model mainly suffers from the following difficulties: existing models are difficult to capture the dependencies and correlations of labels from different sources, and lack the generalization ability to extend to a large amount of labeled data.

For the above problems, this paper proposes a BERT-Bi-GRU model combined with an attention mechanism to solve the MLTC problem with a scalable deep learning method. The main innovations of this paper are listed as follows:

1) The texts are input to the BERT module to learn the encoding vector containing the textual context information; at the same time, the labels are input to the label encoding layer, and the Bi-GRU (Bidirectional Gated Recurrent Unit) is used to obtain the intrinsic relationship between the labels;

2) The BERT output vector is combined with the Bi-GRU output vector for attention mechanism operation to obtain the specific association of each label with the text, and the attention mechanism can explicitly show the prominent relationship between each text and each label. Then the Sigmoid function is used to predict the independent distribution of each label, and the predicted multi-label sequence is obtained.

The rest of this paper is organized as follows. Section II introduces the related research. Section III explains the proposed method in detail. Section IV gives the experimental results and analysis. Finally, Section V summarizes the full text.

2. **Related Research.** For solving the task of MLTC, NLP researchers have proposed many effective methods. Early methods mainly focus on algorithms based on traditional machine learning, including problem transformation methods and algorithm adaptation methods. The other category is deep learning methods, deep learning models learn rich syntactic and semantic information from raw texts through deep neural network structures, and have achieved great success on different NLP tasks.

The problem transformation method handles the multi-label learning task by transforming it into one or more traditional single-label learning tasks, and there are many mature algorithms to choose from to complete the single-label classification task. Binary Relevance(BR) [14] is a typical problem transformation method that decomposes the multi-label learning problem into multiple independent binary classification problems. However, this method lacks the ability to discover the interdependence between labels, which may result in a decrease in the prediction performance. Yen et al. [15] proposed the PDSparse method to learn a separate linear classifier for each label. During training, the classifier optimizes the label distribution to distinguish between all positive labels and a small number of active negative labels for each training sample. Jain et al. [16] proposed the Slice model, which uses the negative sampling technique to train the most easily confused negative labels, and solves the problems associated with the imbalance of label samples.

Algorithm adaptation methods handle multi-label data by extending traditional single-label classification algorithms with certain modifications. The theory and practical experience of single-label learning in the traditional supervised mode provide important references for the exploration of multi-label learning methods. Ranking Support Vector Machine (Rank-SVM) [17] is a machine learning algorithm based on statistical learning theory, which extends the classic support vector machine (SVM) to multi-label learning problems. The basic idea of Multi-label Decision Tree (ML-DT) [18] is to use decision tree technique to process multi-label data, and the criterion of entropy information gain is utilized to construct a decision tree recursively. Multi-label k-Nearest Neighbor (ML-kNN) [19] uses the K-nearest neighbor algorithm to obtain the class labeling of the nearest neighbor samples, and then obtains the label set of unknown samples through reasoning that maximizes the posterior probability.

With the development of deep neural networks, researchers have proposed various MLTC models based on deep neural networks. Compared to simple bag-of-words models as text representations, the technique of encoding raw text into word vectors has begun to be effectively utilized in deep learning models. Neural networks learn high-dimensional text representation vectors to capture syntactic and semantic information of text context. XML-CNN [20] uses a convolutional neural network (CNN) to design a dynamic pool to handle text classification, but the method focuses on document representation and ignores the correlation between labels. Kurata et al. [21] proposed to use the co-occurrence matrix of labels as the initialized weights of the hidden layers and output layers of the model, thus taking into account the correlation of labels. However, the above algorithms all suffer from two problems: 1) due to the limitation of the CNN window size, the long-range dependencies between texts cannot be captured; 2) the words in the document are treated equally when the model makes predictions, including those that are redundant and the noisy ones, which does not focus on those words that contribute more to the classification results. SGM (Sequence Generation Model) [22] uses a Recurrent Neural Network (RNN) to encode the given original text in a Seq2Seq manner, and a new RNN layer is used as a decoder to sequentially generate predicted labels. The disadvantage of this kind of sequence generation is also very obvious. Whether the subsequent generated labels are correct or not depends too much on the results of the previous time series, that

is, the label prediction results will affect each other. Noticing this problem, You et al. [23] proposed to use a self-attention mechanism to learn a text representation for each label, but ignore the relevance of different labels.

In practical applications, labels in MLTC tasks have semantic information, but in many past studies, labels are only regarded as atomic symbols, ignoring the latent knowledge from the text content of labels. In MLTC, labels are in text form and consist of several words. As the most basic module of NLP, word embedding can capture the similarity and regularity between words, so there is a lot of work using word embedding representation for labels, which endow labels with specific semantic information, so as to model the context semantics and label semantics. Du et al. [24] proposed a method of interacting word representations and label representations to obtain a matching score for each word and label, but did not take into account learning different document representations for different labels at a deeper level.

3. **System Framework.** The proposed multi-label text classification framework structure combining BERT and Bi-GRU is shown in Figure 1. The main components include BERT module, label embedding layer, and multi-label attention layer. Firstly, the text is input to the BERT module, and the encoding vector containing the textual context information is learned. At the same time, the labels are input into the Bi-GRU module to obtain the intrinsic relationship between different labels. Then, the output vectors of BERT and Bi-GRU are fed into the attention layer, which effectively guides the text information for classification through the attention mechanism, thereby obtaining the specific connection between each label and the text. Finally, the independent distribution of each label is predicted by the sigmoid function, and the multi-label sequence prediction is obtained.

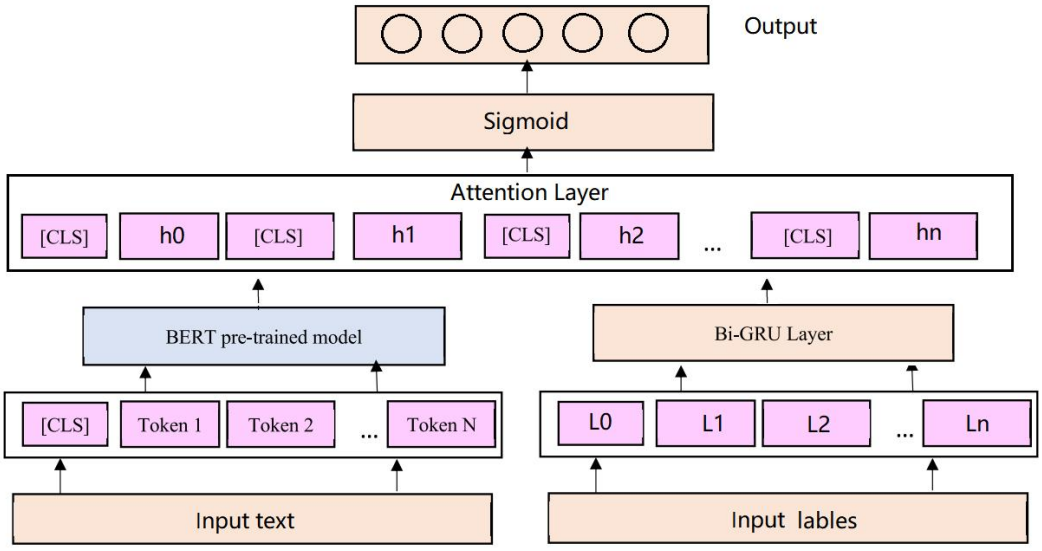

FIGURE 1. Framework Structure.

3.1. **BERT model.** Pre-trained language models have made significant progress on NLP tasks, such models first being pre-trained on massive amounts of text in an unsupervised way, and then fine-tuned on task-specific data. Language model refers to the task of predicting the next word by the context of a given text. During the training process, the model can effectively learn the underlying deep semantic and grammatical information of

the text. BERT is a deep pre-trained language model based on Transformer architecture [25], and its main structure is shown in Figure 2. Taking the Chinese pre-training model

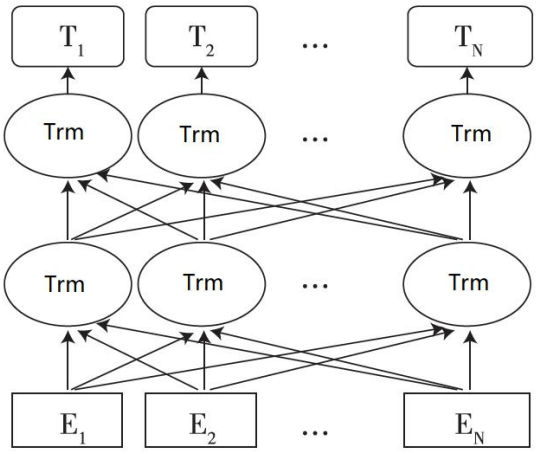

FIGURE 2. Structure of the BERT model.

as an example, in Figure 2, $E_1, E_2, \ldots, E_N$ are the words with marks [CLS] and [SEP] that's been added at the beginning and the end of the sentence segment, respectively. The segment is passed through 12 layers of bidirectional Transformer (Trm) encoder in turn, and the contextual embeddings of text words can be obtained. Transformer is an encoder-decoder based on a Self-attention mechanism. The input to the deepest Transformer encoder is the sum of word vectors, word position vectors, and the sentence fragment vector. Each layer in the model is composed of two parts: Multi-head Self-attention and Feed-forward Neural Networks. The former enables the encoder to pay attention to the information of other surrounding characters when encoding each character; the latter is used to enhance the fitting ability of the model. After each layer of the model undergoes an Add & Norm operation, a new character vector is generated, which is used as the input of the encoder of the next layer. The top-level encoder outputs an encoded vector T_1 labeled with [CLS], which can be regarded as a semantic representation of the entire sentence for subsequent text classification tasks.

In addition, in order to enhance the ability of semantic representation, BERT proposes the concepts of Masked Language Model (MLM) and Next Sentence Prediction (NSP). The essence of MLM is a cloze task. 15% of the words in the Chinese corpus will be selected, of which 80% will be replaced with [MASK], 10% will be randomly replaced with other words, and the remaining 10% will remain the original words. The model needs to go through a linear classifier to predict the selected words. In order to be consistent with the following operations, BERT needs to place the original words or random words at the position of the predicted words in a certain proportion, so that the model is more inclined to use the context information to predict the selected words. In the NSP task, the model selects several sentence pairs, of which there is a 50% probability that the two sentences are adjacent and 50% that the two sentences are not adjacent.

BERT is trained by maximizing the likelihood function of the predicted words, which is calculated as :

$$\max \ln p_\theta(\bar{x}|\hat{x}) \approx \sum_{t=1}^{T} m_t \ln p_\theta(x_t|\hat{x}) \tag{1}$$

where θ denotes the model parameters, $\bar{x}$ is the target word being predicted, $\hat{x}$ is the context words of the target word. When $m_t = 1$, it means that the word is masked and will be replaced by the "[MASK]" token.

The BERT module is applied to learn the vector representation of text, and then fine-tuned to adapt to MLTC tasks. BERT first separates sentences by "[CLS]" and "[SEP]" and inputs them into the model as E_N, the embedding dimension of each word is 768. Then, each word is converted into a T_N which is rich in syntactic and semantic features through the 12-layer Transformers Encoder structure, and the special T_{CLS} feature vector is taken to represent the global context information of the sentence. The "[CLS]" mark does not represent any special word in the text. It will perform self-attention operation with each word in the sentence, so the contextual semantic information of each word in the sentence can be learned, and it can be used as a classification basis to reflect certain fairness and rationality.

3.2. **Bi-GRU layer.** The Bi-GRU model consists of two independent GRU models, and the model structure of GRU is shown in Figure 3. Compared with the LSTM model,

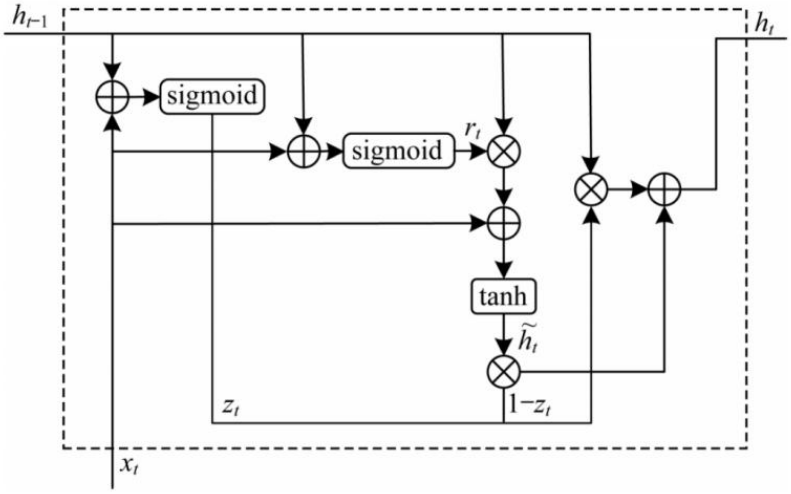

FIGURE 3. GRU model structure.

GRU is simpler in structure and has fewer model parameters, which can reduce the risk of overfitting in the training process. At the same time, the time required for model training is shorter. The calculations are as follows:

$$z_t = \sigma(\boldsymbol{W}_z x_t + \boldsymbol{U}_z h_{t-1}) \tag{2}$$

$$r_t = \sigma(\boldsymbol{W}_r x_t + \boldsymbol{U}_r h_{t-1}) \tag{3}$$

$$\tilde{h}_t = \tanh(\boldsymbol{W} x_t + \boldsymbol{U}(r_t \odot h_{t-1}) \tag{4}$$

$$h_t = z \odot \tilde{h}_t + (1 - z_t) \odot h_{t-1} \tag{5}$$

where $\boldsymbol{W}_z, \boldsymbol{W}_r, \boldsymbol{W}, \boldsymbol{U}_z, \boldsymbol{U}_r$ and $\boldsymbol{U}$ are the weight matrices of the GRU. ht is the current hidden state of the model, and h_{t-1} is the input of the previous state. $\odot$ represents element-wise multiplication, $\tilde{h}_t$ is the candidate state. z_t and r_t denote update gate and reset gate, respectively. x_t is the input state of the model at instant t. σ and tanh are sigmoid and tanh activation functions, respectively.

In the Bi-GRU model, the two GRUs use the same word vector list, but the parameters of the two are independent of each other. The input label vector can be understood as the input sequence. The input sequence passes through the forward GRU and the backward

GRU in the forward and reverse order, respectively. The label feature information obtained at each moment contains contextual relations. The model structure of Bi-GRU is shown in Figure 4. The hidden output of the Bi-GRU at an instant t is jointly determined

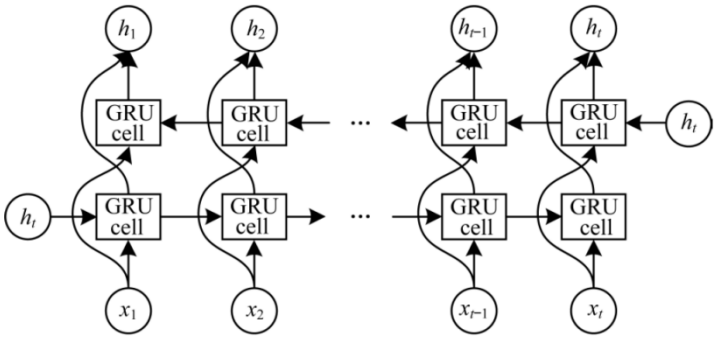

FIGURE 4. Bi-GRU structure.

by two independent GRUs, calculated as follows:

$$\vec{h}_t = GRU(\vec{h}_{t-1}, x_t) \tag{6}$$

$$\overleftarrow{h}_t = GRU(\overleftarrow{h}_{t-1}, x_t) \tag{7}$$

$$h_t = \boldsymbol{W}_t\vec{h}_t + \boldsymbol{V}_t\overleftarrow{h}_t + \boldsymbol{b}_t \tag{8}$$

where $\vec{h}_t$ and $\overleftarrow{h}_t$ are the output states of the forward GRU and the reverse GRU at time t, respectively. $\boldsymbol{W}_t$ and $\boldsymbol{V}_t$ are weight matrices. $\boldsymbol{b}_t$ denotes the bias vector.
In the Bi-GRU-based label encoding layer, first, the label text information is encoded to obtain the vector representation of the label:

$$E = W^e x + b \tag{9}$$

where $E \in \mathbf{R}^{l\times e}$, l represents the number of labels, W is the word embedding, and e is the dimension of the word embedding. After that, the association between labels is obtained through the memory mechanism of the Bi-GRU structure. The function of label coding is to extract all label features through Bi-GRU, expand the vector dimension of labels, and provide computational convenience and interpretability for the subsequent attention mechanism; at the same time, Bi-GRU is used to learn intrinsic continuous features between different labels.

3.3. **Multi-label attention layer.** The proposed framework performs a separated attention fusion operation on the outputs of the BERT model and Bi-GRU, the weights of the CLS vector and each label is obtained, and then the weights are assigned to the CLS vector. The calculation is as follows:

$$C_l = \sum_{i=1}^{i} T_{ti} H_i \tag{10}$$

$$O_l = \text{Sigmoid}(C_l) \tag{11}$$

Where $T_{ti} \in \boldsymbol{R}^h$ is the CLS output of the BERT module, H_i is the output fo the Bi-GRU layer. Through the attention operation of T_{ti} and H_i, the corresponding relationship between "[CLS]" and each label code can be obtained. The attention mechanism can effectively highlight the explicit expression between text features and labels. Finally, the Sigmoid function is used to map the representation to the label dimension, and the prediction result of each label is obtained. The structure of the attention layer is shown in

Figure 5. The advantage of this structure is that the model can predict the independent distribution of each label separately, and at the same time the probability distribution is mapped to the interval of 0∼1, reducing the computational burden caused by a large number of labels.

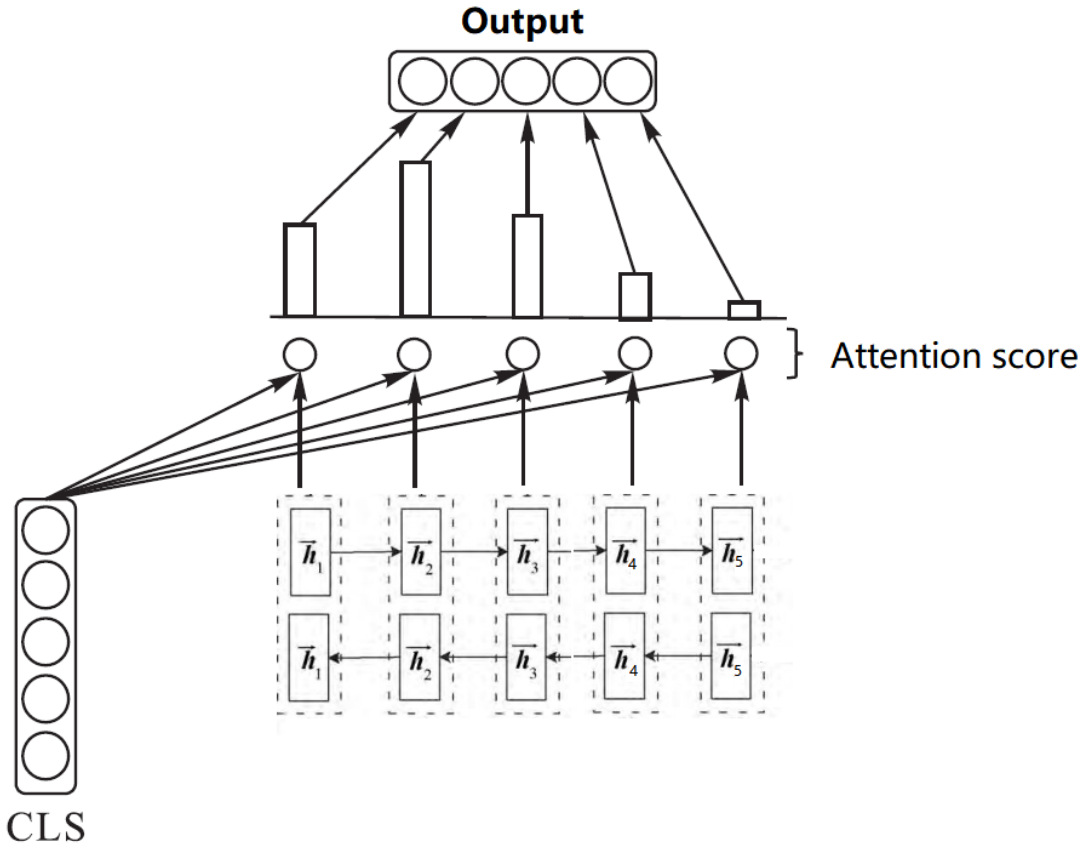

FIGURE 5. Attention layer structure.

3.4. **Loss function.** Binary cross entropy loss is used as the loss function, which is widely used in neural network classification training tasks [26]. The loss function is defined as follows

$$L_{loss} = -\sum_{i=1}^{N}\sum_{j=1}^{l} y_{ij} \log\log(\hat{y}_{ij}) + (1 - y_{ij}) \log\log(1 - \hat{y}_{ij}) \tag{12}$$

where N is the number of documents. l is the number of labels. $\hat{y}_{ij} \in [0, 1]$ and $y_{ij} \in \{0, 1\}$ are the predicted label and the ground-truth label of the j-th label of the i-th instance.

4. **Experiment and Analysis.** The performance of the proposed model is evaluated on the public dataset AAPD and the self-built book cataloging dataset. In this section, the experiment datasets, evaluation metrics and implementation details will be introduced. The experimental results will be analyzed and discussed, then the performance of the proposed method will be compared with other advanced models.

4.1. **Dataset.** Firstly, the single-label and multi-label text classification performance of the proposed method is analyzed through the self-built book cataloging dataset. The data of this dataset is widely collected from Duxiu academic website, in which texts such as titles, subject headings, abstracts, and CLC numbers of 21 categories of B-X and a total of 115,307 books are collected. Among these books, 110,003 books have only one CLC number, accounting for about 95.4%; 3,585 books have two CLC numbers, and 1,719 books have three or more CLC numbers. The dataset is divided into training set, validation set and test set according to 8:1:1 ratio.

Afterwards, the performance of the proposed method and other advanced methods in multi-label text classification is compared using the AAPD (Arxiv Academic Paper Dataset) [22], a public dataset provided by the Big Data Research Institute of Peking University. The dataset mainly consists of abstracts and corresponding topics of 55,840 papers in the field of computer science collected from websites. A paper abstract may contain multiple topics, with a total of 54 topic headings. The goal of MLTC task is to predict the topics corresponding to academic papers based on the abstract content.

4.2. **Environment Configuration.** The experimental platform adopts Windows10 and Ubuntu 18.04 operating systems, and the hardware platform is configured with Intel i5-9400 2.9GHz processor, 16 GB memory and GTX1650Ti GPU. The model is implemented in the Python programming language. The Python version is 3.7. The deep learning libraries used include tensorflow2.3.0, keras2.4.3, gensim3.8.3, and numpy1.18.5. The coding work is completed by the Pycharm development tool [27].

During the training process, the BERT model structure contains 12 layers of Transformers, the hidden layer dimension is 768, and the number of Heads is 12. For the label encoding layer, the label embedding dimension is 256, and the Bi-GRU hidden layer dimension is 768. In addition, the Adam optimizer is used to speed up the training process of the neural network, the initial learning rate is $\alpha = 3E - 5$, the momentum parameter $\beta = 0.9$, and the decaying learning rate $\epsilon = 1 \times 10^{-5}$. At the same time, in the text and label encoding layer, Dropout regularization technique is used to prevent the network from overfitting, and the Dropout rate is 0.5.

4.3. **Evaluation metrics.** Accuracy rate (A), precision rate (P), recall rate (R) and F_1-score are used as the evaluation criteria of the experiment. The related confusion matrix is shown in Table 1. In the table, the rows of the confusion matrix represent the ground-truth categories of the samples, and the columns of the matrix represent the predicted results of the samples. The accuracy rate (A) is the proportion of the number of correctly

Table 1. Confusion matrix

	Positive	Negative
True	True Positive (TP)	True Negative (TN)
False	False Positive (FP)	False Negative (FN)

classified samples to the total number of samples, calculated as:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

The Precision (P) is the proportion of the number of correctly predicted positive samples to the total number of predicted positive samples:

$$P = \frac{TP}{TP + FP} \tag{14}$$

The Recall rate (R) is the proportion of the number of correctly predicted positive samples to the total number of actual positive samples:

$$R = \frac{TP}{TP + FN} \tag{15}$$

The F_1-score is the harmonic mean of precision and recall rate, calculated as:

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{16}$$

4.4. **Experiment results.**

4.4.1. ***Single label classification results***. First, on the book cataloging dataset, experiments are performed on books with only a single CLC number. The number of books in each category is shown in Table 2, where B: Philosophy, Religion, Psychology; C: General Introduction to Social Science; D: Politics, Law; E: Military; F: Economy; G: Culture, Science, Sports , Education; H: Language and Writing; I: Literature; J: Art; K: History and Geography; N: General Introduction to Nature and Sciences; O: Mathematical Science and Chemistry; P: Astronomy, Earth Science; Q: Biological Science; R: Medicine, health; S: Agricultural science; T: Industrial Technology; U: Transportation; V: Aerospace; X: Environmental science, Safety science.

Table 2. Statistics of books with a single CLC number

Catalog number	B	C	D	E	F	G	H	I	J	K
number of books	4824	6583	4975	5815	3024	7014	8752	3468	3974	5601
Catalog number	N	O	P	Q	R	S	T	U	V	X
number of books	7814	4368	2785	5975	6154	7218	8618	5975	4874	3192

As shown in Figure 6, the proposed method only uses the BERT model and does not use Bi-GRU. When the book title and subject heading are used as the input text of BERT, the classification accuracy is greatly improved by nearly 8% compared with only the book title is used as input. On this basis, adding information such as publishers and abstracts, the increase in the classification accuracy is not obvious, but the number of iterations required to achieve convergence gradually increases. Therefore, it can be considered that "book title + subject heading" can effectively represent the main content of the book. Figure 7 presents the single-label classification performance of BERT combined with Bi-GRU and compared with the benchmark methods fasttext and SGM. FastText [28] is a simple and fast text classification model published by Facebook in 2016. It takes word vectors as input, passes through an average pooling layer as a hidden layer, and finally outputs the classification results through softmax. The accuracy rate of the proposed model on the validation set is about 4.8% and 1.9% higher than that of the Fasttext and SGM models, respectively, and the proposed method requires less training epochs than other methods, proving the effectiveness of the proposed method in book cataloging tasks.

4.4.2. ***Multi-label classification results***. On the basis of single-label classification, multi-label classification experiments are carried out. In addition to the single-label data used in the previous section, 5304 book with multiple CLC numbers are added. Although the multi-labeled books account for a small proportion of the total number of books in the dataset (about 3.2%), the relationships between the books and categories are intricate, with as many as 181 different combinations. The two most common types of books with multiple CLC numbers are F&D (economics, politics and law), and R&Q (medicine and

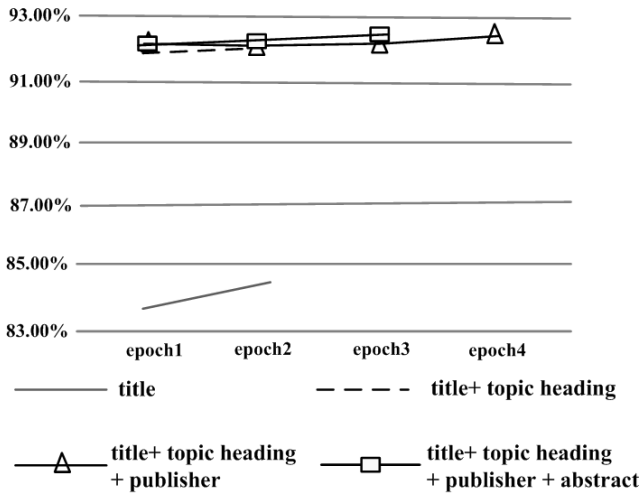

FIGURE 6. Results on the validation set of the book cataloging dataset.

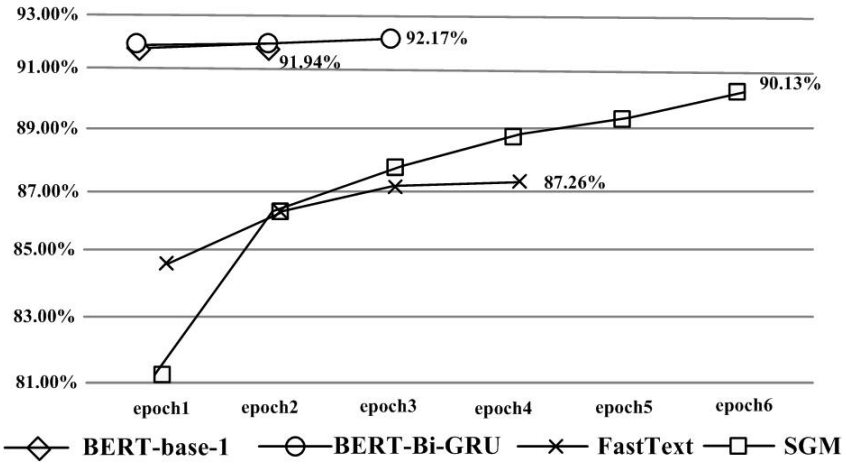

FIGURE 7. Classification accuracy comparison.

health, and biological sciences). The entire data setis divided into training set, validation set, and test set in a ratio of about 8:1:1, and a total of 2 epochs are trained to make the model to converge on the validation set. The results of the proposed model on the test set is shown in Table 3. It is worth noting that in the results shown in Table 3, some books are predicted to have more labels than the actual number. However, after manual inspection, it is found that these predictions that are inconsistent with the actual labels are also reasonable. It is proved that the proposed multi-label classification method preliminarily predicts the subject category to which the book belongs, and can supplement the cataloging numbers that some books may be missing, providing useful recommendations and references for book cataloging personnel.

TABLE 3. Multi-label classification results

Correctly predict at least one CLC number	95.71%
Correctly predict all CLC numbers	93.12%
Predict extra CLC number in addition to the actual CLC numbers	1.5%

4.5. **Comparison with other methods.** Table 4 and Table 5 show the performance comparison between the proposed method and other state-of-the-art methods on the self-built book dataset and AAPD dataset, respectively. Among them, XML-CNN uses CNN to obtain local syntactic and semantic information of text, and uses multiple convolution kernels to extract multi-dimensional features, but focuses on document representation and ignores the correlation between labels, leading to a poor performance. FastText is a simple and fast method based on word vectors, but does not capture deep semantics well. The SGM network has advantages in capturing global information, so it achieves better performance than the first two methods. However, whether the subsequently generated labels are correct or not depends too much on the results of the previous time series, which limits the overall performance. AttentionXML is designed and implemented based on the self-attention mechanism, which learns a specific document representation for each label according to the current document, achieving sub-optimal performance. In the proposed method, through the fusion of BERT and Bi-GRU based on the attention mechanism, the feature relationship between text and labels is explicitly highlighted, thereby achieving the best multi-label classification performance.

Table 4. Performance comparison on the book cataloging dataset

Models	Accuracy	Precision	Recall	F_1-score
XML-CNN [20]	85.94%	85.47%	85.99%	85.89%
FastText [28]	87.26%	87.45%	87.25%	87.27%
SGM [22]	90.13%	90.02%	90.13%	90.11%
AttentionXML [23]	91.45%	91.37%	91.74%	91.49%
Proposed method	93.13%	93.15%	93.11%	93.14%

Table 5. Performance comparison on the AAPD public dataset

Models	Accuracy	Precision	Recall	F_1-score
XML-CNN	62.34%	61.11%	62.80%	62.66%
FastText	66.25%	66.81%	60.53%	63.33%
SGM	71.84%	70.51%	65.92%	68.48%
AttentionXML	80.73%	79.12%	62.57%	71.85%
Proposed method	79.13%	78.34%	70.88%	77.67%

5. **Conclusion.** An MLTC method combining BERT and Bi-GRU is proposed. Through the pre-trained language model BERT, the deep syntactic and semantic representations of the text are learned, so as to effectively integrate the context information of the text; the attention mechanism is used to fully learn the probability distribution characteristics of multiple labels and texts, further improving the performance of the model on MLTC tasks. The experimental results show that the proposed method has achieved good results on both self-built book datasets and public datasets. The proposed method can capture the correlation between different labels, and further improve the classification effect of the model by fusing the intrinsic relationship between labels and texts based on the attention mechanism. In future, we will consider the combination of attention mechanisms of different granularities in the MLTC task. It is expected that the combination of different attention mechanisms can obtain richer semantic representations of texts and labels, so as to predict label sequence efficiently and accurately.

REFERENCES

[1] P. Pant, S.-A Sai, T. Choudhury, P. Dhingra, "Multi-label classification trending challenges and approaches," *Emerging Trends in Expert Applications and Security*, vol. 841, no. 7, pp. 433-444, 2019.

[2] B.-J. Bamgbade, B.-A. Akintola, D.-O. Agbenu, C.-O. Ayeni, O.-O. Fagbami, H.-O. Abubakar, "Comparative analysis and benefits of digital library over traditional library," *World Scientific News*, vol. 24, no. 9, pp. 1-7, 2015.

[3] A. Pal, M. Selvakumar, M. Sankarasubbu, "Multi-label text classification using attention-based graph neural network," in *12th International Conference on Agents and Artificial Intelligence (ICAART 2020)*, IEEE, pp.11644-11653, 2020.

[4] R. Irfan, C.-K. King, D. Grages, S. Ewen, S.-U. Khan, S.-A. Madani, J. Kolodziej, L. Wang, D. Chen, A. Rayes, "A survey on text mining in social networks," *The Knowledge Engineering Review*, vol. 30, no. 2, pp. 157-170, 2015.

[5] A. Fiallos, K. Jimenes, "Using reddit data for multi-label text classification of twitter users interests," in *Sixth International Conference on eDemocracy & eGovernment (ICEDEG 2019)*, IEEE, pp. 324-327, 2019.

[6] W. Liu, H Wang, X. Shen, I.-W. Tsang, "The emerging trends of multi-label learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. [Online] Available: https://doi.org/10.1109/TPAMI.2021.3119334

[7] K.-W. Eric, X. Zhang, F. Wang, T.-Y. Wu, C.-M. Chen, "Multilayer dense attention model for image caption," *IEEE Access*, vol. 7, no.5, pp. 66358-66368, 2019.

[8] F.-Q Zhang, T.-Y. Wu, Y. Wang, R. Xiong, L. Liu, "Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, no. 7, pp. 104555-104564, 2020.

[9] K.-E. Daouadi, R.-Z. Reba, I. Amous, "Optimizing semantic deep forest for tweet topic classification," *Information Systems*, vol. 101, no. 2, pp. 101801-101813, 2021.

[10] W.-K. Sari, D.-P. Rini, R.-F. Malik, "Text classification using long short-term memory with GloVe," *Journal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 5, no. 1, pp. 85-100, 2019.

[11] W. Liu, B. Wen, S. Gao, C. Wang, "A multi-label text classification model based on ELMo and attention," in *Ninth MATEC Web of Conferences* (EDP Science 2020). IEEE, pp. 309-318, 2020.

[12] M. Tezgider, B. Yildiz, G. Aydin, "Text classification using improved bidirectional transformer," *Concurrency and Computation: Practice and Experience*, vol. 34, no.9, pp. e6486-e6499, 2022.

[13] S. González-Carvajal, E.-C. Garrido-Merchán, "Comparing BERT against traditional machine learning text classification", 2020. [Online] Available: https://doi.org/10.48550/arXiv.2005.1301

[14] M.-R. Boutell, J. Luo, X. Shen, C.-M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757-1771, 2004.

[15] I.-E.-H Yen, X. Huang, W. Dai, D. Chen, "Ppdsparse: A parallel primal-dual sparse method for extreme classification," in *23rd International Conference on Knowledge Discovery and Data Mining* (ICKDDM 2017). IEEE, pp.545-553, 2017.

[16] H. Jain, V. Balasubramanian, B. Chunduri, M. Varma, "Slice: scalable linear extreme classifiers trained on 100 million labels for related searches," in *Twelfth ACM International Conference on Web Search and Data Mining* (WSDM 2019). IEEE, pp. 528-536, 2019.

[17] A. Elisseeff, J. Weston, "A kernel method for multi-labelled classification," in *14th Advances in Neural Information Processing Systems* (NIPS 2001). IEEE, pp.1-14, 2001.

[18] A. Clare, R.-D. King, "Knowledge discovery in multi-label phenotype data," in *Sixth European Conference on Principles of Data Mining and Knowledge Discovery* (DMKD 2001). IEEE, pp. 42-53, 2001.

[19] M.-L. Zhang, Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038-2048, 2007.

[20] J. Liu, W.-C. Chang, Y. Wu, Y. Yang, "Deep learning for extreme multi-label text classification," in *Fortieth International ACM SIGIR Conference on Research and Development in Information Retrieval (RDIR 2017)*, IEEE, pp. 115-124, 2017.

[21] G. Kurata, B. Xiang, B. Zhou, "Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence," in *Fifteenth Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)*, pp. 521-526, 2016.

[22] P. Yang, X. Sun, W. Li, S. Ma, W. Wei, "SGM: sequence generation model for multi-label classification," in *Twenty-seventh International Conference on Computational Linguistics (ICCL 2018)*, IEEE, pp. 3915-3926, 2018.

[23] R. You, Z. Zhang, Z. Wang, S. Dai, H. Mamitsuka, S. Zhu, "Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification," in *20th Advances in Neural Information Processing Systems (NIPS 2019)*, IEEE, pp. 1-32, 2019.

[24] C. Du, Z. Chen, F. Feng, Z. Lei, L. Nie, "Explicit interaction model towards text classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no, 1, pp. 6359-6366, 2019.

[25] C. Sun, X. Qiu, Y. Xu, X. Huang, "How to fine-tune bert for text classification?" in *19th China National Conference on Chinese Computational Linguistics (CCL 2019)*, IEEE, pp. 194-206, 2019.

[26] S. Baker, A.-L. Korhonen, "Initializing neural networks for hierarchical multi-label text classification," in *Fourth Biomedical Natural Language Processing Shared Task Workshop (Bio NLP 2017)*, IEEE, pp. 307-315, 2017.

[27] M. Dilhara, A. Ketkar, D. Dig, "Understanding Software-2.0: A study of machine learning library usage and evolution," *ACM Transactions on Software Engineering and Methodology*, vol. 30, no. 4, pp. 1-42, 2021.

[28] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, "Bag of tricks for efficient text classification," 2016. [Online]. Available: https://doi.org/10.48550/arXiv.1607.01759.