Chinese Automatic Readability Assessment Using Adaptive Pre-training and Linguistic Feature Fusion

Xusheng Yang
School of Computer Science
Faculty of Artificial Intelligence in Education
Central China Normal University
yaxnlp@outlook.com

Jincai Yang*
School of Computer Science
Central China Normal University
jcyang@ccnu.edu.cn

Xiao Li
Chengdu Yandaojie Primary School
1059511176@qq.com

Abstract

Chinese Automatic Readability Assessment (ARA) aims to classify the reading difficulty of Chinese texts. To address the issues of insufficient high-quality training data and underutilization of linguistic features in existing methods, we propose a method that combines adaptive pre-training with feature fusion based on an interactive attention mechanism. First, we enhance the model's ability to capture different text difficulties through domain- and task-specific adaptive pre-training. Then, we propose an Adaptive Task-guided Corpus Filtering (ATCF) method, utilizing embeddings generated by the pre-trained model and applying nearest-neighbor search along with a sample balancing mechanism to ensure comprehensive learning across various difficulty levels. Finally, we propose an Interactive Attention-Driven Feature Fusion method that integrates linguistic and deep features, providing rich difficulty information to the model. Experiments on Chinese textbook dataset demonstrate that our method achieves state-of-the-art (SOTA) performance. Transfer learning experiments further indicate that our approach generalizes well to extracurricular reading and Chinese as a Foreign Language (CFL) ARA tasks.

1 Introduction

Text difficulty, often referred to as readability, is a measure of how challenging a text is to read. ARA aims to evaluate and categorize the difficulty levels of texts based on various features, including lexical, syntactic, and semantic characteristics. This task is crucial for leveled reading and also has significant applications in areas such as essay evaluation (Wang, 2017) and book recommendation (Pera and Ng, 2014).

Early research on ARA primarily focused on the development of readability formulas grounded in pedagogical heuristics and psychological theories (Klare, 2000; Davison and Kantor, 1982). These formulas are known for their interpretability and computational efficiency, yet they mainly consider surface-level text features, failing to capture deeper structural and semantic information, which limits their overall effectiveness.

With advancements in natural language processing, traditional machine learning approaches that leverage a broader spectrum of features have been applied to train ARA models (Sung et al., 2015; Denning et al., 2016). These models incorporate not only surface features but also lexical, semantic, and syntactic analyses. While they outperform readability formulas, these methods heavily depend on labor-intensive feature engineering and selection processes, which often fall short in capturing the intricate internal characteristics of texts.

The emergence of deep learning techniques has brought new opportunities for ARA research (Martinc et al., 2021; Sun et al., 2020). By harnessing the powerful deep feature extraction capabilities of deep learning models, the performance of ARA has seen significant improvement.

Research on Chinese native language ARA started relatively late, but recent studies have begun exploring deep learning methods. These studies typically use multiple versions of Chinese language textbooks as their corpora and primarily employ two approaches: the first approach involves using end-to-end neural networks to integrate deep features extracted by pre-trained language models with linguistic features at the character, word, and sentence levels, followed by training a classification model (Cheng et al., 2020); the second approach focuses on constructing customized neural network architectures tailored to the characteristics of leveled corpora (Li and Wu, 2023). These deep learning-based ARA models effectively extract and

*Corresponding Author

Proceedings of the 31st International Conference on Computational Linguistics, pages 9013–9024
January 19–24, 2025. ©2025 Association for Computational Linguistics

utilize semantic and structural information from texts, thereby achieving promising performance.

However, existing methods still face certain limitations. Although the first approach leverages BERT (Devlin et al., 2019), which performs exceptionally well in Chinese language processing, BERT struggles with processing long texts and fails to extract effective features from the lengthy texts found in higher-grade Chinese textbooks. Additionally, the neural network models used in both approaches require a large amount of high-quality training data, yet the available textbook data is limited in scale, preventing the models from fully realizing their potential.

To address the aforementioned issues, we propose a Chinese ARA method leveraging adaptive pre-training and an interactive attention mechanism.

First, we collect a corpus of Chinese reading materials and perform Domain-Adaptive Pre-training (DAPT) on the long-sequence pre-trained model BIGBIRD (Zaheer et al., 2020).

Then, we propose an **ATCF** method, which embeds both domain-specific corpora and task-specific corpora (training datasets) into a vector space using an adaptive pre-trained model. By employing nearest neighbor search, the method retrieves the k-nearest domain samples most similar to the task samples. Additionally, the number of selected samples is controlled based on the ratio of sample distribution across different difficulty levels, aiming to obtain high-quality pre-training data that is highly relevant to the ARA task.

Finally, we present the **Interactive Attention-Driven Feature Fusion** method. This method employs interactive computations and mapping to derive the attention weights and interaction information necessary for merging linguistic features with deep features extracted by the adaptive pre-trained model. The fusion of these features provides the model with more detailed difficulty-related information.

Experiments conducted on the Chinese textbook dataset demonstrate that the proposed method achieves SOTA performance. Transfer learning experiments indicate that the proposed method also outperforms the SOTA models on both Chinese extracurricular reading ARA task and CFL ARA task.

We will release our code[1].

2 Related Work

2.1 Traditional Machine Learning Methods

Traditional machine learning approaches treat ARA as a classification task, utilizing linguistic features of the text as inputs and outputting the difficulty level. Compared to readability formulas, these models not only consider surface-level features but also analyze deeper features such as syntactic complexity and grammatical structures, leading to better performance in identifying challenging texts (Schwarm and Ostendorf, 2005; Heilman et al., 2008; Feng, 2010). For instance, Xia et al. (2016) achieved 80.3% accuracy on the Weebit dataset by training a SVM classifier with features such as lexical, syntactic, sentence length, language model, and discourse features. Vajjala and Lucic (2018) manually extracted six categories of features, including n-grams, parts of speech, psycholinguistic features, syntax, discourse, and traditional features, achieving 78.1% accuracy with an SVM classifier on the OneStopEnglish dataset. These studies demonstrate that leveraging a rich set of linguistic features can significantly enhance the accuracy of text difficulty assessment.

In the context of Chinese, Chen et al. (2011) employed mutual information to select keywords and constructed an SVM model based on the TF-IDF values of these words. Sung et al. (2014) built an SVM model using 31 linguistic features across lexical, semantic, and syntactic levels as predictors. Wu et al. (2020) developed a text feature system with more layers and dimensions, achieving accurate predictions of difficulty levels in lower-grade texts. These studies further validate the effectiveness of multi-level linguistic features in Chinese ARA.

2.2 Deep Learning Methods

In contrast to traditional machine learning methods, deep learning approaches can automatically learn and extract deep features from texts, avoiding the complex process of manual feature extraction while significantly improving the accuracy of ARA. Research by Deutsch et al. (2020) demonstrated that Hierarchical Attention Networks (HAN) could extract detailed information from texts, outperforming SVM classifiers, highlighting the importance of using deep learning models with strong generalization capabilities in text difficulty assessment tasks. Lee et al. (2021) enhanced classification accuracy by combining the deep features output by

[1] https://github.com/YaxNLP/ChineseARA_ATCF_FF

the RoBERTa model with manually extracted linguistic features and feeding them into a Random Forest classifier. Martinc et al. (2021) found that truncating texts could lead to the loss of difficulty-related information. When datasets contain a large number of long texts, HAN performs best; however, BERT excels on datasets with a predominance of shorter texts.

In the Chinese context, Liu et al. (2017) combined CNN and LSTM (Hochreiter and Schmidhuber, 1997) to capture both short-range features and long-range dependencies, significantly outperforming SVM in experiments on 345 Chinese language textbook texts. Cheng et al. (2020) used the BERT model to extract sentence features from texts, then applied BiLSTM to concatenate sentence features to obtain deep document-level features, finally integrating these deep features with linguistic features at the character, word, and sentence levels, achieving an accuracy of 46% in fine-grained classification for 12th-grade texts. However, concatenating features can lead to the loss of contextual information, which may impact model performance. Li and Wu (2023) employed a variable-length convolutional layer to extract deep features from texts, achieving an accuracy of 56% in an eight-level classification task, validating the importance of deep features. However, this method did not utilize linguistic feature information. These studies underscore the potential of deep learning models in Chinese text difficulty assessment tasks, while also pointing out the challenges in effectively extracting useful information with current methods.

3 Method

3.1 Adaptive Pre-training

We collected recommended readings from the "Compulsory Education's Chinese Curriculum Standards (2022 Edition)" (of Education of the People's Republic of China, 2022) and the "General High School Chinese Curriculum Standards (2017 Edition, Revised in 2020)" (of Education of the People's Republic of China, 2020), along with Chinese National College Entrance Examination (Gaokao) reading materials from 2010 to 2022. Each text was annotated with the recommended learning stage as specified by the curriculum standards, forming the Chinese primary and secondary school reading corpus (hereafter referred to as the domain corpus). To prevent the adaptive pre-training model from prematurely learning text from the Chinese textbook dataset, we employed fuzzy matching and cosine similarity calculations to remove duplicate content between the domain corpus and the dataset. The final corpus consists of 20,939 texts totaling 28.03 million characters, with an average length of 1,338 characters per text.

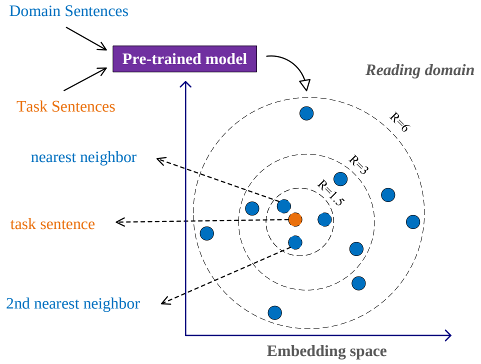


Figure 1: An illustration of ATCF. We map documents from both the domain corpus and the training set of the dataset into a shared vector space using the DAPT+TAPT model. For each document in the training set, we identify $R(l) \times k$ nearest neighbors from the reading domain. In this figure, k=2

We used the domain corpus to perform Masked Language Modeling (MLM) on BIGBIRD, enabling the model to acquire domain-specific knowledge from the large-scale reading corpus, thus achieving DAPT. We then performed small-scale task-specific MLM on BIGBIRD using the training set of the Chinese textbook dataset to achieve task-adaptive pre-training (TAPT). Finally, we applied TAPT on the model obtained from DAPT, achieving comprehensive adaptive pre-training (DAPT+TAPT) to further enhance the model's performance and applicability.

To address the challenges of small and imbalanced datasets, we propose **ATCF** (See Figure 1). This method is designed to tackle the characteristics of the domain corpus, which is relatively small in scale, consists of long texts, and has coarse-grained classification labels. It improves upon the task-specific data filtering method based on a lightweight bag-of-words model and KNN (Gururangan et al., 2020). The specific calculation process is as follows:

Text Embedding: Equations 1 and 2 represent the process of embedding the texts from the domain corpus and the training set into a vector space. D denotes the domain corpus, which includes both texts and their corresponding difficulty labels. T refers to the training set, also consisting of texts and their difficulty labels.

$$domain_{embeddings} = \text{DAPT+TAPT}(D) \quad (1)$$

$$task_{embeddings} = \text{DAPT+TAPT}(T) \quad (2)$$

Sample Count and Selection Ratio Calculation: Equation 3 calculates the number of samples for each difficulty level within the training set. Here, t_j represents the jth text in the training set, $l_t(t_j)$ indicates the difficulty label of t_j, l denotes the difficulty level, and $\mathbb{I}$ equals 1 if the condition is true, otherwise 0. Equation 4 identifies the maximum sample count, while Equation 5 calculates the selection ratio for each difficulty level.

$$N(l) = \sum_{j=1}^{|T|} \mathbb{I}(l_t(t_j) = l) \quad (3)$$

$$N_{max} = max_l N(l) \quad (4)$$

$$R(l) = \frac{N_{max}}{N(l)} \quad (5)$$

Nearest Neighbor Search: For each training sample, Equation 6 defines the function find_neighbors, which employs KNN to identify the nearest neighbor samples. This function returns the k domain samples that are most similar to the training sample within the vector space.

$$\text{neighbors}(t_j) = \text{find\_neighbors}(t_j, k) \quad (6)$$

Sample Selection and Ratio Control: Equation 7 is used to select samples from the domain corpus, ensuring that the difficulty level of these domain samples matches that of the training samples. Here, nei(t_j) corresponds to Equation 6, and s_texts(l) represents the selected set of domain samples. Equation 8 adjusts the number of selected domain samples to align with the expected selection ratio $R(l)$. Finally, Equation 9 defines final_selected_texts, which represents the final selection results.

$$\text{s\_texts}(l) = \bigcup_{j=1}^{|T|} \{d_i | d_i \in nei(t_j) \text{ and } l_d(d_i) = l\} \quad (7)$$

$$|\text{s\_texts}(l)| \approx k \times R(l) \quad (8)$$

$$\text{final\_selected\_texts} = \bigcup_{l} \text{s\_texts}(l) \quad (9)$$

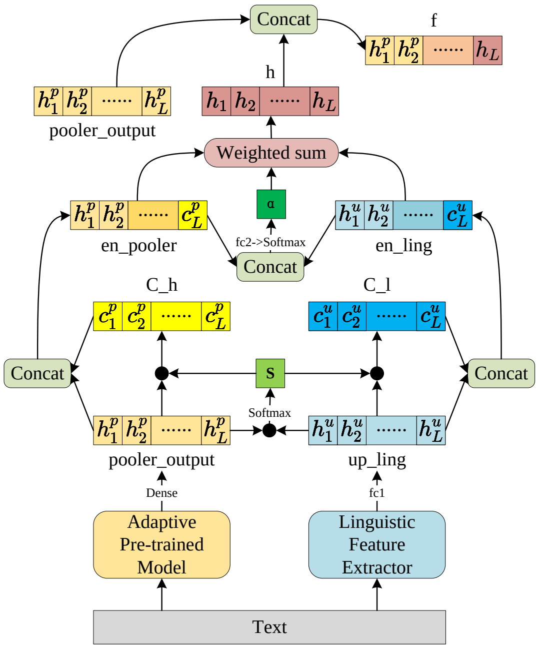

Concat
f
h_1^p h_2^p ...... h_L
h
h_1^p h_2^p ...... h_L^p
pooler_output
h_1 h_2 ...... h_L
Weighted sum
α
fc2->Softmax
h_1^p h_2^p ...... c_L^p
en_pooler
Concat
h_1^u h_2^u ...... c_L^u
en_ling
C_h
C_l
c_1^p c_2^p ...... c_L^p
c_1^u c_2^u ...... c_L^u
Concat
S
Concat
Softmax
h_1^p h_2^p ...... h_L^p
h_1^u h_2^u ...... h_L^u
pooler_output
up_ling
Dense
fc1
Adaptive Pre-trained Model
Linguistic Feature Extractor
Text

Figure 2: The process of Interactive Attention-Driven Feature Fusion.

3.2 Linguistic Features

Based on the standards outlined in the "Compulsory Education's Chinese Curriculum Standards (2022 Edition)," we extend the 50-dimensional linguistic features proposed by Cheng et al. (2020) by adding 31 new features. Additionally, the original 24-dimensional part-of-speech features were condensed into 12-dimensional word class features. The MinMax method was applied to normalize the different features. A total of 69 linguistic features were used in this study, and the specific definitions and implementation methods of the new features are in Appendix A.

3.3 Interactive Attention-Driven Feature Fusion

To effectively integrate features extracted by pre-trained language models and linguistic features, we propose an Interactive Attention-Driven Feature Fusion method. The core of this method lies in leveraging an adaptive interactive attention mechanism to enhance the interaction and representation capabilities between features. The feature fusion process is illustrated in Figure 2. The specific steps are as follows:

Feature Dimension Expansion: First, the linguistic features are expanded to the same dimensionality as the pre-trained model features to facilitate subsequent feature fusion, as shown in Equation 10. The term fc denotes a fully connected layer.

$$up\_ling = \text{fc1}(ling) \tag{10}$$

Attention Calculation: The attention weights S between the pre-trained model features ($pooler\_output$) and the expanded linguistic features (up_ling) are computed:

$$pooler\_output = \text{dense}(cls\_output) \tag{11}$$

$$S = \text{softmax}(pooler\_output \cdot up\_ling^T) \tag{12}$$

In Equation 11, cls_output is the representation of [CLS] extracted from the pre-trained language model's output, which contains information about the entire sequence. The term dense refers to a linear layer that maps to the same dimension. The $pooler\_output$ is typically used for downstream classification tasks.

Based on the attention weights S, content-related representations are computed using Equations 13 and 14:

$$C_h = S \cdot pooler\_output \tag{13}$$

$$C_l = S^T \cdot up\_ling \tag{14}$$

As shown in Equations 15 and 16, the content-related representations are concatenated with the original features to obtain the enhanced representations:

$$en\_pooler = [pooler\_output; C_h] \tag{15}$$

$$en\_ling = [up\_ling; C_l] \tag{16}$$

Attention Fusion: The attention weights α for the fused features are calculated as shown in Equation 17:

$$\alpha = \text{softmax}(\text{fc2}([en\_pooler; en\_ling])) \tag{17}$$

Using the attention weights α, the fused features h are obtained according to Equation 18:

$$h = \alpha \cdot en\_pooler + (1 - \alpha) \cdot en\_ling \tag{18}$$

Finally, the fused features h are concatenated with the pre-trained model features pooler_output, supplementing the pre-trained model features with additional information to derive the final features f in Equation 19:

$$f = [pooler\_output; h] \tag{19}$$

4 Experiments

The experimental setup is detailed in Appendix B.

4.1 Dataset

We integrate the graded Chinese language textbook corpus for primary and secondary schools (Cheng et al., 2020) and the gold standard corpus of primary school Chinese textbooks (Liu et al., 2021). Additionally, the study includes the recently revised primary school Chinese textbooks. The final dataset comprises textbooks from 13 different editions covering primary, middle, and high school levels, with poetry and classical Chinese texts removed for the experiments. For duplicate texts, the latest revised edition was prioritized.

To ensure comprehensive experimentation, two granularity levels for difficulty classification were adopted: an eight-level classification and a five-level classification (Li and Wu, 2023). The dataset was divided into training, validation, and test sets in an 8:1:1 ratio. The eight-level classification is defined as follows: 0-5 for grades 1-6 (primary school), 6 for middle school, and 7 for high school. The five-level classification is defined as follows: 0 for grades 1-2, 1 for grades 3-4, 2 for grades 5-6, 3 for middle school, and 4 for high school. Dataset details are provided in Appendix C.

4.2 Adaptive Pre-training, Fine-tuning and Classification of Fused Features

We conducted adaptive pre-training of the Chinese version of BIGBIRD[2] using five different data scales, applying the Whole Word Masking (WWM) strategy during masked language model training. This strategy masks entire words to improve the model's understanding of word integrity. Table 1 shows the training steps, time, dataset size, and post-training loss. We fine-tuned BIGBIRD and other adaptive pre-trained models with key hyperparameters, including batch_size=2, max_length=2048, 5 epochs, cross-entropy loss, and the AdamW optimizer (eps=1e-8). The optimizer was combined with a linear learning rate scheduler, where the learning rate increases during the first 10% of training steps, then decreases linearly to zero. A parameter search was conducted with learning rates of 8e-6, 1e-5, and 3e-5. Both the pre-training and fine-tuning of BIGBIRD were implemented using the Transformers v4.42.3 li-

[2] https://huggingface.co/Lowin/chinese-bigbird-wwm-base-4096

Pre-training Model	Loss	Steps	Time	Data
DAPT	1.274	2k	15.3h	20932
TAPT	1.682	0.7k	4.9h	2800
DAPT+TAPT	1.042	0.7k	4.9h	2800
25NN-TAPT	1.285	1k	7.5h	8463
DAPT+25NN-TAPT	1.054	1k	7.5h	8463

Table 1: Adaptive pre-training models. (1) DAPT: BIGBIRD was trained on the full domain corpus, (2) TAPT: BIGBIRD was trained on the training set texts, (3) DAPT+TAPT: BIGBIRD was first trained on the domain corpus and then on the training set texts, (4) 25NN-TAPT: using ATCF with k=25, BIGBIRD was trained on both the selected task corpus and the training set texts, and (5) DAPT+25NN-TAPT: BIGBIRD was first trained on the domain corpus and then further trained on the selected task corpus and the training set texts.

brary. For the fused features, a three-layer linear mapping was applied to predict the class labels, with each layer activated by the ReLU function. Detailed hyperparameters for adaptive pre-training and classification of fused features are provided in Appendix D.

4.3 Evaluation Metrics

Considering that ARA is an ordinal multi-class problem, the evaluation metrics used in the experiments include Accuracy, Weighted Precision, Weighted Recall, Weighted F-Measure, and Quadratic Weighted Kappa (QWK). These metrics are consistent with those used in studies by Martinc et al. (2021) and Lee et al. (2021). The SciKit-learn (Pedregosa et al., 2011) library was employed to implement these evaluation metrics.

4.4 Baseline Models

TextCNN: This model utilizes 128-dimensional Word2Vec embeddings, three convolutional layers with filter sizes of 3, 4, and 5, and 100 channels, followed by max-pooling and feature vector concatenation.

BiLSTM: This model employs 128-dimensional Word2Vec embeddings, two bidirectional LSTM layers with 50 hidden states, a fully connected layer, and two dropout layers (rate = 0.1) to prevent overfitting.

BERT+Ling (Cheng et al., 2020): The text is split into sentences, and sentence vectors from the bert-base-chinese model are fed into a BiLSTM to capture contextual relationships, forming a 512-dimensional deep feature vector. Both the deep and 69-dimensional linguistic features are projected to 32 dimensions, added, activated by Tanh, and passed to a fully connected layer for classification.

ChatGPT[3]: Based on GPT-3.5, trained with data up until January 2022.

BERT (Devlin et al., 2019) and LERT (Cui et al., 2022): Fine-tuning BERT[4] and LERT[5] using the Transformers library.

VBCNN (Li and Wu, 2023): This model uses 128-dimensional Word2Vec embeddings, a VBCNN with variable kernel sizes (1, 3, 5) and 128 channels for local feature extraction, stacked convolution blocks for downsampling, and a BiLSTM with 64 hidden units for sequential processing.

The detailed hyperparameters of the baseline models and the prompt used for ChatGPT are listed in Appendix E.

5 Results

Table 2 presents the results for the eight-class and five-class classification experiments.

Shi et al. (2023) found that since ChatGPT's parameters are inaccessible and it cannot be fine-tuned on specific datasets for task adaptation, its classification performance is poor when used directly. The results in this study show that the same conclusion applies to the Chinese ARA task.

In both classification tasks, fine-tuned BIGBIRD outperformed most baseline models. The performance improvement of BIGBIRD over VBCNN was modest. However, achieving better results through fine-tuning alone highlights the strength of pre-trained language models.

DAPT and TAPT outperformed the base BIGBIRD model in both classification tasks. D-TAPT achieved better performance than both DAPT and TAPT in the eight-class classification, demonstrating the advantages of combining domain-adaptive and task-adaptive pre-training. In the five-class classification, D-TAPT slightly underperformed compared to DAPT and TAPT, possibly due to the broader classification criteria, which made the benefits of pre-training less pronounced than in the eight-class task.

25-TAPT showed strong performance in both classification tasks, suggesting that highly task-relevant data can further improve the model's performance in ARA. D-25-TAPT achieved the best results among pre-trained models.

[3] https://chatgpt.com

[4] https://huggingface.co/google-bert/bert-base-chinese

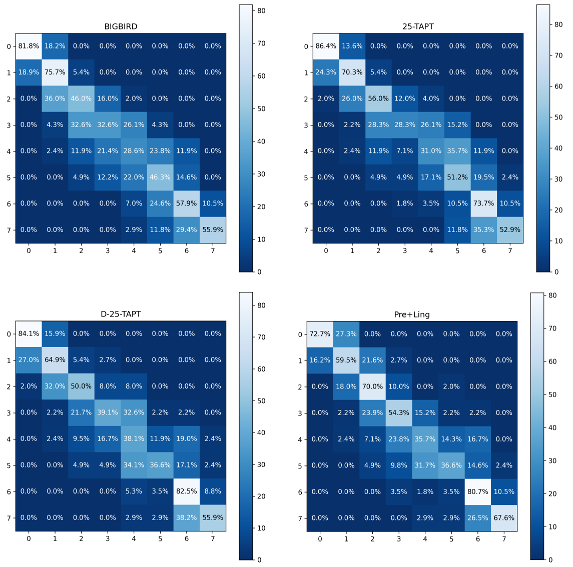
[5] https://huggingface.co/hfl/chinese-lert-base

Model-8c	**Acc**	**QWK**	**Pre**	**Rec**	**F1**
TextCNN	47.0	85.5	46.3	47.0	46.4
BiLSTM	49.3	88.5	50.4	49.3	49.1
BERT+Ling	47.6	87.8	45.1	47.6	45.2
ChatGPT	24.2	58.1	27.4	24.2	23.6
BERT	51.6	89.2	53.5	51.6	52.0
LERT	49.0	89.1	50.3	49.0	49.5
VBCNN	**52.4**	**90.8**	**53.7**	**52.4**	**52.8**
Ling	43.9	85.1	45.6	43.9	41.9
BIGBIRD	52.7	92.0	53.5	52.7	52.5
DAPT	55.3	92.3	55.6	55.3	54.6
TAPT	53.3	92.0	51.4	53.3	51.9
D-TAPT	57.0	92.3	57.2	57.0	55.7
25-TAPT	56.7	92.4	56.7	56.7	55.8
D-25-TAPT	57.3	92.3	58.5	57.3	56.7
C(Pre;Ling)	59.2	92.4	60.0	59.3	59.1
Pre+Ling	**60.7**	**92.6**	**61.0**	**60.7**	**60.2**
Model-5c	**Acc**	**QWK**	**Pre**	**Rec**	**F1**
TextCNN	66.7	86.1	66.6	66.7	66.6
BiLSTM	**70.4**	**88.3**	**72.4**	**70.4**	**69.8**
BERT+Ling	63.5	85.1	62.8	63.5	62.3
ChatGPT	32.8	52.1	33.4	32.8	31.0
BERT	64.4	84.3	64.3	64.4	64.2
LERT	64.1	86.2	64.1	64.1	64.1
VBCNN	68.7	86.5	71.1	68.7	66.7
Ling	59.3	84.0	58.8	59.3	57.7
BIGBIRD	70.1	87.8	70.6	70.1	69.4
DAPT	70.4	88.6	70.7	70.4	69.9
TAPT	70.4	88.2	69.8	70.4	69.9
D-TAPT	69.2	88.2	70.6	69.2	69.0
25-TAPT	71.2	88.9	72.4	71.2	71.1
D-25-TAPT	74.1	89.5	74.6	74.1	74.0
C(Pre;Ling)	75.2	90.2	75.5	75.2	74.8
Pre+Ling	**75.8**	**90.9**	**76.3**	**75.8**	**75.6**

Table 2: Results of both eight-class (8c) and five-class (5c) experiments. Ling: Uses two fully connected layers to project 69-dimensional linguistic features, followed by ReLU activation. D-TAPT: DAPT+TAPT. 25-TAPT: 25NN-TAPT. D-25-TAPT: DAPT+25NN-TAPT. C(Pre;Ling): Concatenate features extracted by D-25-TAPT with linguistic features and classify using two fully connected layers. Pre+Ling: Combines D-25-TAPT with linguistic features using the Interactive Attention-Driven Feature Fusion method.

Pre+Ling excelled in both classification tasks, achieving 60.7% accuracy in the eight-class classification and 75.8% in the five-class classification, outperforming all other models. Compared to BIGBIRD, it improved accuracy by 8.0% in the eight-class task and 5.7% in the five-class task, demonstrating that integrating pre-trained model features with linguistic features offers more comprehensive information.

On the test set, we constructed confusion matrices for the eight-class classification of BIGBIRD, 25-TAPT, D-25-TAPT, and Pre+Ling, where the diagonal values represent the accuracy of each category. The results are shown in Figures 3.



BIGBIRD

	0	1	2	3	4	5	6	7
0	81.8%	18.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
1	18.9%	75.7%	5.4%	0.0%	0.0%	0.0%	0.0%	0.0%
2	0.0%	36.0%	46.0%	16.0%	2.0%	0.0%	0.0%	0.0%
3	0.0%	4.3%	32.6%	32.6%	26.1%	4.3%	0.0%	0.0%
4	0.0%	2.4%	11.9%	21.4%	28.6%	23.8%	11.9%	0.0%
5	0.0%	0.0%	4.9%	12.2%	22.0%	46.3%	14.6%	0.0%
6	0.0%	0.0%	0.0%	0.0%	7.0%	24.6%	57.9%	10.5%
7	0.0%	0.0%	0.0%	0.0%	2.9%	11.8%	29.4%	55.9%

25-TAPT

	0	1	2	3	4	5	6	7
0	86.4%	13.6%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
1	24.3%	70.3%	5.4%	0.0%	0.0%	0.0%	0.0%	0.0%
2	2.0%	26.0%	56.0%	12.0%	4.0%	0.0%	0.0%	0.0%
3	0.0%	2.2%	28.3%	28.3%	26.1%	15.2%	0.0%	0.0%
4	0.0%	2.4%	11.9%	7.1%	31.0%	35.7%	11.9%	0.0%
5	0.0%	0.0%	4.9%	4.9%	17.1%	51.2%	19.5%	2.4%
6	0.0%	0.0%	0.0%	1.8%	3.5%	10.5%	73.7%	10.5%
7	0.0%	0.0%	0.0%	0.0%	0.0%	11.8%	35.3%	52.9%

D-25-TAPT

	0	1	2	3	4	5	6	7
0	84.1%	15.9%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
1	27.0%	64.9%	5.4%	2.7%	0.0%	0.0%	0.0%	0.0%
2	2.0%	32.0%	50.0%	8.0%	8.0%	0.0%	0.0%	0.0%
3	0.0%	2.2%	21.7%	39.1%	32.6%	2.2%	2.2%	0.0%
4	0.0%	2.4%	9.5%	16.7%	38.1%	11.9%	19.0%	2.4%
5	0.0%	0.0%	4.9%	4.9%	34.1%	36.6%	17.1%	2.4%
6	0.0%	0.0%	0.0%	0.0%	5.3%	3.5%	82.5%	8.8%
7	0.0%	0.0%	0.0%	0.0%	2.9%	2.9%	38.2%	55.9%

Pre+Ling

	0	1	2	3	4	5	6	7
0	72.7%	27.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
1	16.2%	59.5%	21.6%	2.7%	0.0%	0.0%	0.0%	0.0%
2	0.0%	18.0%	70.0%	10.0%	0.0%	2.0%	0.0%	0.0%
3	0.0%	2.2%	23.9%	54.3%	15.2%	2.2%	2.2%	0.0%
4	0.0%	2.4%	7.1%	23.8%	35.7%	14.3%	16.7%	0.0%
5	0.0%	0.0%	4.9%	9.8%	31.7%	36.6%	14.6%	2.4%
6	0.0%	0.0%	0.0%	3.5%	1.8%	3.5%	80.7%	10.5%
7	0.0%	0.0%	0.0%	0.0%	2.9%	2.9%	26.5%	67.6%

Figure 3: Confusion matrices.

BIGBIRD showed stable performance across most categories but exhibited noticeable confusion in middle categories, such as categories 3 and 4. 25-TAPT achieved significantly higher accuracy in the upper-grade levels, indicating the necessity of increasing pre-training data. D-25-TAPT showed a more balanced performance, with clearer distinction in categories 3 and 4. The Pre+Ling model, by integrating linguistic features, was able to capture textual characteristics more comprehensively, resulting in a notable improvement in recognizing finer-grained categories.

6 Transfer Learning

The CMER[6] dataset, compiled by Zeng et al. (2022), consists of 2,260 extracurricular reading texts for children and adolescents from mainland China, divided into 12 levels. The HSK[7] dataset, constructed by Tan et al. (2024) based on the official Test of CFL documents, includes 5,721 texts across six levels. We fine-tuned our adaptive pre-trained models using both the CMER and HSK datasets. The fine-tuning results on the CMER dataset are shown in Table 3, and the results on the HSK dataset are presented in Table 4. During fine-tuning, all hyperparameters remained consistent with those used in the textbook dataset, except for the learning rate on CMER, which was set to 7e-6, 4e-6, and 3e-6, respectively.

The CMER dataset primarily targets fine-grained classification across 12 semesters from grades 1

[6] https://github.com/JinshanZeng/DTRA-Readability

[7] https://github.com/CocoTan1020/MLF-BERT

Model	Acc	QWK	Pre	Rec	F1
HAN	23.4	**72.1**	15.4	-	18.4
BERT	22.3	65.3	30.1	-	13.4
DTRA	**26.5**	70.5	25.3	-	**25.1**
PromptARA	**26.5**	68.7	**24.2**	-	23.9
Ling	23.6	64.6	14.0	23.6	15.1
BIGBIRD	28.9	72.8	22.4	28.9	23.3
DAPT	27.6	76.0	27.7	27.6	24.8
TAPT	27.4	77.6	**31.8**	27.4	**27.4**
D-TAPT	27.8	76.6	27.3	27.8	23.8
25-TAPT	**30.5**	78.3	27.8	**30.5**	26.2
D-25-TAPT	26.3	77.2	27.8	26.3	22.3
C(Pre;Ling)	28.0	77.0	26.7	28.0	24.8
Pre+Ling	29.1	**79.2**	26.2	29.1	27.2

Table 3: Performance of models on the CMER dataset. Pre is 25-TAPT. DTRA (Zeng et al., 2022) is a model based on BERT embeddings with a structure similar to HAN. It learns ordinal information between texts by predicting the relative difficulty of paired texts and utilizing distance-related soft labels. PromptARA (Zeng et al., 2023) enhances deep feature representations extracted by BIGBIRD with a prompt mechanism and fuses deep features with linguistic features through an orthogonal projection layer.

to 6, whereas the DAPT dataset spans texts from grades 1 through 12, covering more advanced content, which introduces classification interference.

As shown in Figure 3, the 25-TAPT model outperforms BIGBIRD in overall performance for stages 0-2 (grades 1-6) in the textbook dataset. Due to the shorter pre-training steps and smaller data size of 25-TAPT, the interference with BIGBIRD is less significant than that of DAPT, resulting in superior performance and achieving new SOTA results.

On the HSK dataset, adaptive pre-trained models underperformed compared to BERT, suggesting that models pre-trained on native Chinese and task-specific data have lower adaptability to CFL corpora. Thus, we performed task-adaptive pre-training on BERT with the following hyperparameters: max_length=512, batch_size=32, gradient_accumulation_steps=3, max_grad_norm=1, and steps=385, resulting in the model B-TAPT. The label mapping between HSK and the domain corpus is as follows: 0-1 for stage 1, 2-3 for stage 2, and 4-5 for stage 3. We replaced DAPT+TAPT with B-TAPT in ATCF and set k=50, resulting in the model B-50-TAPT.

In Table 4, B-50-TAPT significantly improves B-TAPT's performance, indicating that the ATCF approach is also highly applicable to CFL corpora. While B-50-TAPT's accuracy is slightly lower than that of MLF-BERT, it reduces the demand for linguistic feature extraction, thus conserving resources. Furthermore, with the adoption of the Interactive Attention-Driven Feature Fusion method, Pre+Ling surpasses MLF-BERT in the F1 score, achieving new SOTA.

Model	Acc	QWK	Pre	Rec	F1
ELECTRA	88.1	-	-	-	87.0
MLF-ELECTRA	89.7	-	-	-	89.4
BERT	91.1	-	-	-	90.9
MLF-BERT	**94.2**	-	-	-	**93.9**
Ling	61.8	90.1	62.2	61.8	60.5
BIGBIRD	90.2	97.6	90.8	90.2	89.5
DAPT	88.8	97.2	89.8	88.8	87.4
TAPT	88.3	97.1	88.9	88.3	87.5
D-TAPT	89.3	97.3	90.2	89.3	88.0
25-TAPT	88.3	97.1	89.3	88.3	87.3
D-25-TAPT	90.5	97.7	90.8	90.5	90.1
BERT*	90.9	97.8	91.0	90.9	90.8
B-TAPT	92.5	98.2	92.5	92.5	92.4
B-50-TAPT	94.0	**98.6**	94.0	94.0	94.0
C(Pre;Ling)	93.5	98.5	93.5	93.5	93.5
Pre+Ling	**94.2**	97.6	**94.2**	**94.2**	**94.1**

Table 4: Performance of models on the HSK dataset. * denotes experiments conducted on our setup. Pre is B-50-TAPT. MLF-ELECTRA / MLF-BERT (Tan et al., 2024) incorporates linguistic features into the embedding and self-attention layers of the ELECTRA / BERT models.

7 Conclusion

This paper proposes a method that combines adaptive pre-training with linguistic feature fusion to enhance the accuracy of Chinese text readability classification. By embedding both domain-specific and task-specific corpora into a shared vector space through adaptive pre-training, our approach supplements task corpora with the most similar domain texts. Additionally, a quantity-filtering mechanism based on class proportions ensures a relatively balanced distribution of samples across difficulty levels. The high-quality pre-trained corpus improves the model's ability to distinguish text difficulty. The Interactive Attention-Driven Feature Fusion captures the complex relationships between multi-level linguistic and deep features, generating fused features that provide additional textual information to the adaptive pre-trained model. Experimental results demonstrate that the proposed method performs excellently on datasets of Chinese native language textbooks, extracurricular readings, and CFL texts.

In future work, we will explore more diverse pre-training strategies and data augmentation techniques, and investigate whether this method can be applied to large language models. Additionally, since the linguistic features relevant to readability may vary across different types of datasets, identifying the optimal combination of linguistic features for specific domains represents another promising direction for future research.

Limitations

When applying our method to other languages or domains, a considerable amount of domain-specific data must be collected and organized, which requires substantial resources. When the domain data is particularly large, using adaptive pre-trained language models for high-quality data filtering can place heavy demands on computational resources. One challenge we face is how to automatically analyze the dataset to select appropriate linguistic features. Too few features may have little impact on the model's performance, while too many could introduce noise, and feature extraction itself can be resource-intensive.

References

Wanxiang Che, Yunlong Feng, Libo Qin, and Ting Liu. 2021. N-LTP: An open-source neural language technology platform for Chinese. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 42–49, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yaw-Huei Chen, Yao-Hung Hubert Tsai, and Yu-Ta Chen. 2011. Chinese readability assessment using tf-idf and svm. In *2011 International Conference on Machine Learning and Cybernetics*, volume 2, pages 705–710.

Yong Cheng, Dekuang Xu, and Jun Dong. 2020. Automatic grading of chinese text reading difficulty based on multiple linguistic features and deep features. *Journal of Chinese Information Processing*, 34(04):101–110.

Yiming Cui, Wanxiang Che, Shijin Wang, and Ting Liu. 2022. Lert: A linguistically-motivated pre-trained language model. *ArXiv*, abs/2211.05344.

Alice Davison and Robert N. Kantor. 1982. On the failure of readability formulas to define readable texts: A case study from adaptations. *Reading Research Quarterly*, 17:187–209.

Joel Denning, Maria Soledad Pera, and Yiu-Kai Ng. 2016. A readability level prediction tool for k-12 books. *Journal of the Association for Information Science and Technology*, 67.

Tovly Deutsch, Masoud Jasbi, and Stuart M. Shieber. 2020. Linguistic features for readability assessment. In *Workshop on Innovative Use of NLP for Building Educational Applications*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages "4171–4186", Minneapolis, Minnesota. Association for Computational Linguistics.

Lijun Feng. 2010. *Automatic Readability Assessment*. Ph.D. thesis, City University of New York.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Michael Heilman, Kevyn Collins-Thompson, and Maxine Eskenazi. 2008. An analysis of statistical models and features for reading difficulty prediction. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 71–79, Columbus, Ohio. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

George R. Klare. 2000. The measurement of readability: useful information for communicators. *ACM J. Comput. Documentation*, 24:107–121.

Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenbiao Li and Yunfang Wu. 2023. Chinese readability assessment based on deep neural networks. *Journal of Chinese Information Processing*, 37(2):158.

Hao Liu, Si Li, Jianbo Zhao, Zuyi Bao, and Xiaopeng Bai. 2017. Chinese teaching material readability assessment with contextual information. In *2017 International Conference on Asian Language Processing (IALP)*, pages 66–69. IEEE.

Miaomiao Liu, Yan Li, Xinmeng Wang, Linlin Gan, and Hong Li. 2021. Leveled reading for primary students:construction and evaluation of chinese readability formulas based on textbooks. *Applied Linguistics*, (2):11.

Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. Supervised and unsupervised neural approaches to text readability. *Computational Linguistics*, 47(1):141–179.

Ministry of Education of the People's Republic of China. 2020. *General High School Chinese Curriculum Standards (2017 Edition, Revised in 2020)*. People's Education Press.

Ministry of Education of the People's Republic of China. 2022. *Compulsory Education's Chinese Curriculum Standards (2022 Edition)*. Beijing Normal University Publishing Group.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. 12:2825–2830.

Maria Soledad Pera and Yiu-Kai Ng. 2014. Automating readers' advisory to make book recommendations for k-12 readers. In *ACM Conference on Recommender Systems*, RecSys '14, page 9–16, New York, NY, USA. Association for Computing Machinery.

Sarah E. Schwarm and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Annual Meeting of the Association for Computational Linguistics*.

Y. Shi, H. Ma, W. Zhong, Q. Tan, G. Mai, X. Li, T. Liu, and J. Huang. 2023. Chatgraph: Interpretable text classification by converting chatgpt knowledge to graphs. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 515–520, Los Alamitos, CA, USA. IEEE Computer Society.

Yuxuan Sun, Keying Chen, Lin Sun, and Chenlu Hu. 2020. Attention-based deep learning model for text readability evaluation. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Yao-Ting Sung, Ju-Ling Chen, Ji-Her Cha, Hou-Chiang Tseng, Tao-Hsing Chang, and Kuo-En Chang. 2014. Constructing and validating readability models: the method of integrating multilevel linguistic features with machine learning. *Behavior Research Methods*, 47:340 – 354.

Yao-Ting Sung, Wei-Chun Lin, Scott Benjamin Dyson, Kuo-En Chang, and Yu-Chia Chen. 2015. Leveling l2 texts through readability: Combining multilevel linguistic features with the cefr. *The Modern Language Journal*, 99:371–391.

Keren Tan, Yunshi Lan, Yang Zhang, and Anqing Ding. 2024. Chinese text readability grading via multi-level linguistic feature fusion. *Journal of Chinese Information Processing*, 38(5):41.

Sowmya Vajjala and Ivana Lucic. 2018. Onestopenglish corpus: A new corpus for automatic readability assessment and text simplification. In *BEA@NAACL-HLT*.

Yixuan Wang. 2017. The correlation between lexical richness and writing score of csl learner———the multivariable linear regression model and equation of writing quality. *Applied Linguistics*, (02):93–101.

Siyuan Wu, Dong Yu, and Xing Jiang. 2020. Development of linguistic features system for chinese text readability assessment and its validity verification. *Chinese Teaching in the World*, 34(01):81–97.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *BEA@NAACL-HLT*.

Jincai Yang, Yuxin Cao, Quan Hu, and Xuxun Cai. 2022. Automatic recognition of chinese compound sentence relation based on bert-fhan model and sentence features. *Computer Systems and Applications*, 31(9):233–240.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Jinshan Zeng, Yudong Xie, Xianglong Yu, John Lee, and Ding-Xuan Zhou. 2022. Enhancing automatic readability assessment with pre-training and soft labels for ordinal regression. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4557–4568, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinshan Zeng, Xianglong Yu, Xianchao Tong, and Wenyan Xiao. 2023. PromptARA: Improving deep representation in hybrid automatic readability assessment with prompt and orthogonal projection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15360–15371, Singapore. Association for Computational Linguistics.

A Definitions and Implementation Methods of New Features

Ind/Dim	Feature
1 / 2	Number/Proportion of Idioms
2 / 12	Word Class Proportions
3 / 5	Syntactic Structure Proportions
4 / 6	Sentence Component Proportions
5 / 12	Compound Sentence Type Proportions
6 / 6	Punctuation Proportions

Table 5: Supplementary linguistic features overview.

Number/Proportion of Idioms: Traverse all words and determine if they are idioms based on the Xinhua Dictionary Idiom Database[8].

[8] https://github.com/pwxcoo/chinese-xinhua

Word Class Proportions: Word Class Proportions: The proportions of nouns, verbs, adjectives, numerals, classifiers, pronouns, adverbs, prepositions, conjunctions, particles, modal particles, and interjections. Part-of-speech tagging was performed using the Language Technology Platform (LTP) (Che et al., 2021).

Syntactic Structure Proportions: The proportions of coordination, modification, subject-predicate, verb-object, and complement structures. Syntactic analysis was conducted using LTP.

Sentence Component Proportions: The proportions of subjects, predicates, objects, attributives, adverbials, and complements. Syntactic analysis and semantic role labeling were performed using LTP.

Compound Sentence Type Proportions: The proportions of causal, hypothetical, inferential, conditional, purposive, coordinative, coherent, progressive, alternative, adversative, concessive, and pseudo-adversative types. The CCCSRA dataset (Yang et al., 2022) was divided into an 8:2 ratio for training and testing, and BERT was fine-tuned to achieve a classification model with 98% accuracy for identifying compound sentence types.

Punctuation Proportions: The proportions of commas, periods, question marks, exclamation marks, colons, and quotation marks. These were identified using pattern matching rules.

B Experimental Setup

The experiments were conducted using PyTorch on a Windows 11 system, with an NVIDIA RTX 3090 GPU (24GB VRAM) and an AMD Ryzen 7 5700X processor. To ensure reproducibility and fair comparisons, the random seed for Python, NumPy, and PyTorch was set to 0.

C Details of the Chinese Textbook Dataset

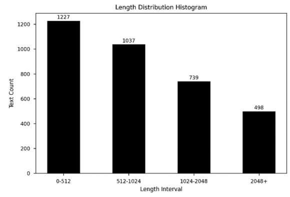


Figure 4: Chinese textbook dataset length distribution

Label	Texts	Avg. Chars	Total Chars
0	366	155	56878
1	429	320	137384
2	462	555	256440
3	497	783	389576
4	507	1038	526203
5	453	1282	580728
6	436	2231	972530
7	351	3311	1162277
Total	3501	1166	4082016

Label	Texts	Avg. Chars	Total Chars
0	795	244	194262
1	959	674	646016
2	960	1153	1106931
3	436	2231	972530
4	351	3311	1162277
Total	3501	1166	4082016

Table 6: Statistics of the eight-class and five-class classifications of the textbook dataset.

D Hyperparameters for Adaptive Pre-training and Classification of Fused Features

Adaptive Pre-training: The training used a batch_size of 2, with gradient accumulation (gradient_accumulation_steps=50) to enhance training efficiency and memory utilization. To prevent gradient explosion, the gradient norm was clipped to 1 (max_grad_norm=1). According to Figure 4, approximately 85% of the texts are shorter than 2048 tokens, max_length was set to 2048.

Fused feature classification: batch_size=32, epochs=5, learning_rate=6e-4, using cross-entropy loss and the AdamW optimizer.

E Baseline Model Hyperparameters

TextCNN: Dropout (with a rate of 0.1) is applied, followed by a fully connected layer for classification. The main hyperparameters are batch_size=64, epochs=50, learning_rate=0.0001, the optimizer is AdamW, and the loss function is cross-entropy loss.

BiLSTM: The main hyperparameters are batch_size=64, epochs=50, learning_rate=0.005, the optimizer is AdamW, and the loss function is cross-entropy loss.

BERT+Ling (Cheng et al., 2020): The main hyperparameters are batch_size=64, epochs=50, learning_rate=0.0001, the optimizer is SGD, and the loss function is cross-entropy loss.

BERT (Devlin et al., 2019) and LERT (Cui et al., 2022): The main hyperparameters are batch_size=16, epochs=5, learning_rate=8e-6, max_length=512, the optimizer is AdamW, and the loss function is cross-entropy loss.

ChatGPT: The prompt requires the text to be classified into two granularities. The first granularity divides the text into: Grade 1, Grade 2, Grade

3, Grade 4, Grade 5, Grade 6, Middle, and High School. The second granularity divides the text into: First Stage, Second Stage, Third Stage, Middle, and High School. Only the final classification results for the entire text in these two granularities are needed.

VBCNN (Li and Wu, 2023): Each block consists of two convolution layers with kernel sizes of 3, max pooling, and ReLU activation. The main hyperparameters are batch_size=64, epochs=200, learning_rate=0.0001, max_length=4500, the optimizer is AdamW, and the loss function incorporates label smoothing with cross-entropy loss.