

Research Article

Zhu Xiao*

Analysis of deep classification grammar error correction algorithm for online English grammar teaching

<https://doi.org/10.1515/comp-2025-0050>

Received March 3, 2025; accepted December 9, 2025;

published online February 4, 2026

Abstract: To solve the insufficient educational resources in offline classroom English teaching, the research focuses on online teaching and designs an online grammar correction algorithm to help students realize automatic online error correction. The algorithm is based on the Transformer model algorithm based on multi-head attention mechanism (MHAM), and integrates word order information into the encoding process. Three pseudo-parallel corpora are used to expand the number of training data. Finally, Adam is used to optimize the model parameters to improve the model performance. The accuracy, recall, and $F_{0.5}$ of the algorithm formed after two one-way optimization are the highest values in the same type of optimization model. The $SP + Pre_{human} + TF_{copy}$ algorithm formed after double optimization has the best comprehensive performance. The accuracy rate reached 68.53 %, and the $F_{0.5}$ value reached 58.26 %, both of which were the highest values in the comparison model. Moreover, this method can also provide certain technical support for the establishment of multilingual interaction platforms and high-quality natural language generation.

Keywords: deep classification; attention mechanism; transform; Adam; grammar; error correction

1 Introduction

With the development and popularization of internet technology, students can now learn English through online courses, applications, and other digital resources. This not

only provides personalized and flexible learning methods, but also greatly reduces the limitations on learning time and location [1]. Although online English grammar teaching has brought many conveniences to students, there are still shortcomings in practical applications. Traditional teaching methods often rely on fixed exercises and simple question banks to test students' grammatical competence [2]. These methods cannot simulate real language usage environments well, nor do they take into account the diversity and complexity of language usage [3]. In addition, these systems often lack personalized feedback for individual students' specific problems, which cannot effectively improve students' language errors such as speech and grammar [4]. This requires the classification and correction of English grammar. In traditional online classification grammar correction, the recognition and correction of grammatical errors are usually realized through the predefined rules, which cannot cover all errors and adapt to new language usage habits and error patterns [5]. This leads to a significant decrease in the efficiency and accuracy of the system's error correction when encountering errors outside of the rules [6]. The grammar error correction model is mainly reflected in the computer-based processing of natural language texts, which achieves automatic text recognition and classification error correction through a large amount of text training [7]. In large texts, the model needs to extract text information features based on contextual information [8]. However, the ultimate value of an efficient grammar correction system lies not only in the efficiency of technical indicators, but also in how it can be transformed into actual learning outcomes [9]. Research suggests that providing direct, immediate, and precise grammar corrections can improve the effectiveness of online English learning. This improvement occurs through the following mechanisms: First, immediate feedback can avoid the solidification of erroneous memories and help students quickly correct cognitive biases. Second, precise classification and error correction can provide targeted explanations for individual weaknesses. Third, continuous tracking of error types can provide teaching references for teachers.

*Corresponding author: **Zhu Xiao**, School of International Studies, Chengdu Jincheng College, Chengdu, 610000, China, E-mail: xiaozhu@cdjcc.edu.cn

However, there are still limitations in current related research, mainly reflected in three aspects. First, existing rule-based error correction systems have difficulty covering complex grammatical phenomena and dynamically changing language habits. These systems have a weak ability to recognize context-dependent errors, such as preposition collocation and semantic coherence errors, and lack the ability to adapt to students' individual error patterns. Second, although mainstream deep learning error correction models can alleviate the problem of excessive error correction, they highly rely on large-scale high-quality manually annotated corpora. However, in real teaching scenarios, data on student homework with error-correction annotations is scarce. This results in the model having insufficient generalization ability in practical teaching applications and being ineffective at handling irregular errors. Thirdly, existing research primarily focuses on optimizing technical model indicators. However, it neglects the connection between technology and teaching. This results in a failure to consider differences in error characteristics among students of different grades and English proficiency levels. It also results in a failure to provide suitable functions for teaching scenarios. Consequently, it becomes difficult to effectively transform technological achievements into teaching effectiveness.

Therefore, a deep learning based automatic grammar correction model is developed, and a deep classification grammar correction model combining multi-head attention mechanism (MHAM) and Transformer model is designed. Through this advanced grammar correction method, the goal is to achieve high efficiency and precision at the technical level. Additionally, it is intended to effectively enhance the effectiveness, sustainability, and personalization of students' online English learning by leveraging the aforementioned mechanisms.

2 Related works

2.1 Research on rule based grammar correction

Rule based grammar correction methods are the earliest developed technological path, which rely on manually writing grammar rule libraries to match error patterns in text. Typical representatives include the LanguageTool basic module and early Grammarly systems [10]. This method's core advantage lies in its strong interpretability. It can provide accurate positioning and correction suggestions for clear rules, such as subject-verb agreement and tense changes. It can also be quickly deployed without the need

for large-scale data training. However, its limitations can no longer meet the needs of modern online teaching. One reason is that the rules are not fully covered, with a recognition rate of less than 40 % for complex contextual errors, such as preposition placement and semantic coherence. It is also unable to adapt to the dynamic evolution of language usage habits [11]. Second, the generalization ability is weak, and the personalized error handling effect on non-native learners due to negative transfer of their mother tongue is extremely poor [12]. Third, maintenance costs are high. Adding new types of errors requires linguistic experts to manually update the rule library. This makes it difficult to address the variety of errors in online teaching [13].

2.2 Research on grammar correction based on traditional machine learning

With the development of statistical learning techniques, grammar correction methods based on classification models are gradually replacing rule-based systems, transforming error correction tasks into classification problems through feature engineering. For example, the error correction system based on the maximum entropy model proposed by Rafique A et al. identified article errors by extracting features such as part of speech and syntactic structure, achieving an accuracy rate of 58 % on the dataset [14]. The SVM classifier constructed by Kamel J et al. was optimized for preposition errors. Moreover, the recall rate was increased to 45 % [15].

2.3 Research on grammar correction based on deep learning

The breakthrough of deep learning technology has brought new ideas to grammar correction tasks, and its mainstream research can be divided into three categories. Convolutional/recurrent neural network model: The multi-layer convolutional neural network model proposed by Voloshina T et al. identified basic syntax errors through local feature extraction, with an $F_{0.5}$ value of 49.7 % on the test set. However, due to limitations in its network structure, it was unable to effectively capture long-range dependencies. Its error correction accuracy for complex sentences is only 55 % [16]. Recurrent neural networks and their variants, such as LSTM, could address sequence dependency issues. However, they experienced gradient vanishing and substantial performance degradation when processing lengthy texts [17].

Basic transformer model: The Transformer architecture proposed by Hossain N et al. broke through the bottleneck of long-distance dependencies with multi-head

attention mechanism and was widely used in syntax correction [18]. However, this type of model relied heavily on manually annotated parallel corpora. In real teaching scenarios, student homework data with error correction labels was scarce. This resulted in the model having an insufficient ability to generalize.

Fine tuning of pre-trained language models: In recent years, fine-tuning methods based on pre-trained models such as BERT and GPT have become a research hotspot. Tinn R et al. used BERT for error detection tasks and achieved an F1 value of 72 % on the W&I-dev dataset [19]. Kortemeyer G et al.'s study showed that GPT-4 had a Kendall correlation coefficient of 0.662 when evaluated using human judgment, which far exceeded traditional automatic evaluation metrics. However, when directly used for error correction tasks, there was an issue of over generation, with a modification rate of up to 18 % for error free sentences [20].

2.4 Research on pseudo parallel corpus generation

To address the scarcity of annotated corpora, researchers propose generating pseudo-parallel corpora to expand training data. The mainstream methods can be divided into two categories. One is regularized noise injection, which generates erroneous sentences by randomly deleting, replacing, and inserting words. The second is model driven error generation. It uses a small amount of annotated data to train error generation models and then “contaminates” correct sentences.

According to the literature analysis above, the primary objective of existing research is to address the three core limitations of grammar correction models: corpus adaptation, error correction accuracy, and teaching practicality. The existing rule-based models rely on manual rule libraries, resulting in incomplete error coverage and weak generalization ability. The basic Transformer model overly relies on scarce annotated corpora and is prone to excessive error correction. A single pseudo corpus optimization model is difficult to meet the practical needs of “precise error correction + personalized support” in online English teaching due to insufficient corpus authenticity and lack of adaptability to teaching scenarios. In response to these shortcomings, the research method employs a collaborative design consisting of an improved Transformer architecture, multi-strategy pseudo corpus generation, and a dual optimization module. This design enhances the ability to capture semantic and word order through a replication mechanism and position encoding. Additionally, it solves the problem of weak long-distance dependency processing in traditional models. Combining three types of pseudo corpus fusion

(noise injection, real error simulation, and an error generation model) with real student homework data improves the corpus's authenticity and the model's generalization ability. Compared with a single reverse translation pseudo corpus model, the real error matching degree is increased by more than 40 %. Balancing the generation probability through spell-checking, preprocessing, and replication mechanisms reduces the error-free sentence modification rate from 18 % to below 5 %. Overall accuracy improved by about 10.5 % points compared to the basic Transformer model. At the same time, the research method designs a statistics panel for error types and a grade adaptation function for teaching scenarios. It covers five types of high-frequency errors and provides instant rule feedback. This overcomes the limitations of existing models that focus on technical indicators and neglect teaching value. It transforms “technical correction” into “teaching empowerment” and provides more accurate, practical technical support for online English teaching.

3 Construction of deep classification syntax correction algorithm

3.1 Transformer core algorithm

The self-attention mechanism (SAM) is the core of the Transformer model, which enables the model to perform parallel calculations and capture long-distance dependencies when processing sequences [21]. In this structure, each word of a sentence corresponds to three matrices of query quantity q , key vector k , and value vector v . The query quantity q is responsible for calculating the correlation information between the current coded word and other words in the sentence. The key vector k is responsible for querying and encoding the information of word relevance. The value vector v is the information about the words to be encoded. These three vectors are obtained by multiplying the word embedding vector of the word and three different weight matrices W_Q , W_K , and W_V . The vectors corresponding to each word are combined to generate Q , K , and V matrices. Then the dot product results of the query matrix Q and the key matrix K are scaled and normalized. The final processing result is obtained by point multiplication of the result and the value matrix. The process of SAM is shown in formula (3.1).

$$\text{Att}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{b_k}}\right)V \quad (3.1)$$

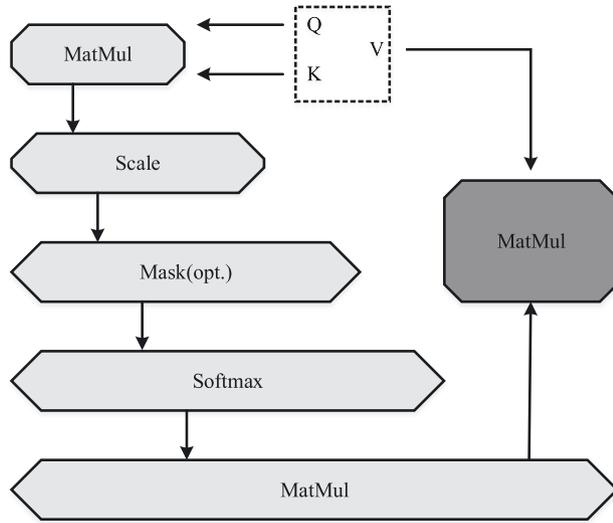


Figure 3.1: The overall structure of SAM.

The size of b_k in formula (3.1) is the dimension of q and k . The overall structure of SAM is shown in Figure 3.1.

Taking the attention score of the word ‘Thinking’ as an example, it first needs multiply the word embedding vector x_1 of the word by three weight matrices to obtain the corresponding query vector q_1 , key vector k_1 , and value vector v_1 . Then it needs to multiply the obtained q_1 , k_1 , and k_2 to get the corresponding attention score, and divide the score by the root of the q_1 and k_1 dimensions to scale the attention score. Finally, the softmax function is used to normalize and scale it, and the final result is obtained by weighted sum with the value vector. The concrete SAM calculation method is shown in Figure 3.2.

MHAM is equivalent to the combination of multiple SAMs [22]. The MHAM encoding ‘Thinking Machines’ is studied for interpretation. The attention mechanism of eight heads is selected in the study. When encoding the word ‘Thinking’, at first, the word is converted into the word embedding vector x . Then it needs to multiply the word embedding vector by eight different weight matrices to obtain eight different sets of query vectors q_i , key vector k_i , and value vector v_i ($i = 0, 1, \dots, 7$). Next it needs to operate each head in the same way as self-attention to get the vector z_i ($i = 0, 1, \dots, 7$). Finally, these vectors are spliced together with the weight matrix to get the encoded vector z . The calculation is generally carried out in the form of matrix. The matrix operation process of MHAM is shown in Figure 3.3.

MHAM provides more possibilities for the code model to acquire multiple semantic information. Each header can be responsible for different semantic information, such as capturing local semantic information around the coded word, or capturing long-distance semantic information.

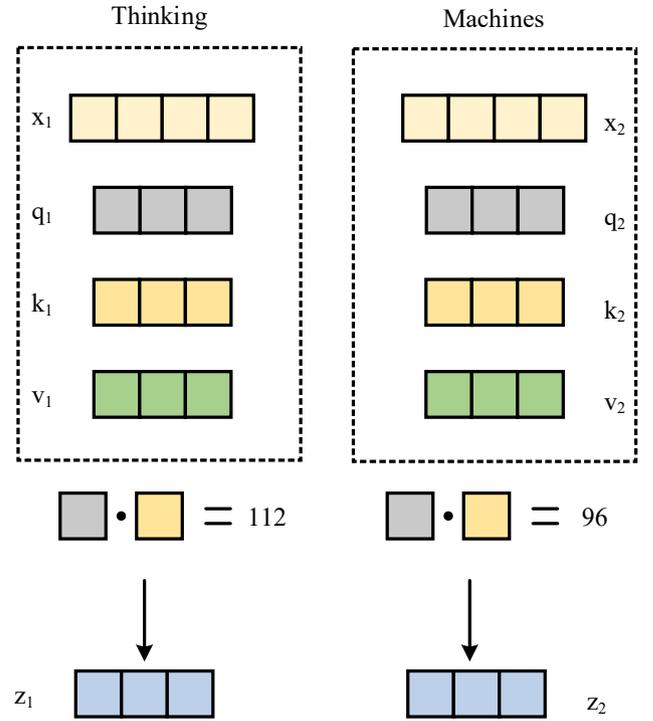


Figure 3.2: SAM calculation method.

However, it does not take the word order of the input sentence into account in the calculation process. Therefore, sentences with different word sequences have the same coding results. However, in the actual natural language processing process, the encoding process is generally required to incorporate the word order information in the sentence. Therefore, the Transformer model introduces location coding into the SAM coding process. The position vector corresponding to each word in the sentence is different. The position vector is determined manually, and the design rules of the position vector are shown in formula (3.2).

$$p(i, 2w) = \sin\left(\frac{i}{10000 \frac{2w}{d}}\right) \quad (3.2)$$

In formula (3.2), w is the word vector corresponding to the current word, and d is the dimension of the word vector. The Transformer model includes encoder and decoder sections, each consisting of 6 layers. Each encoder layer consists of a MHAM and a forward fully connected network, with residual connections and normalization. The decoder layer adds an additional attention layer on top of the encoder. The overall model structure processes sequential data through modularity and repetition layers. The overall structure of the Transformer model is shown in Figure 3.4, where N is the number of layers of the encoder and decoder.

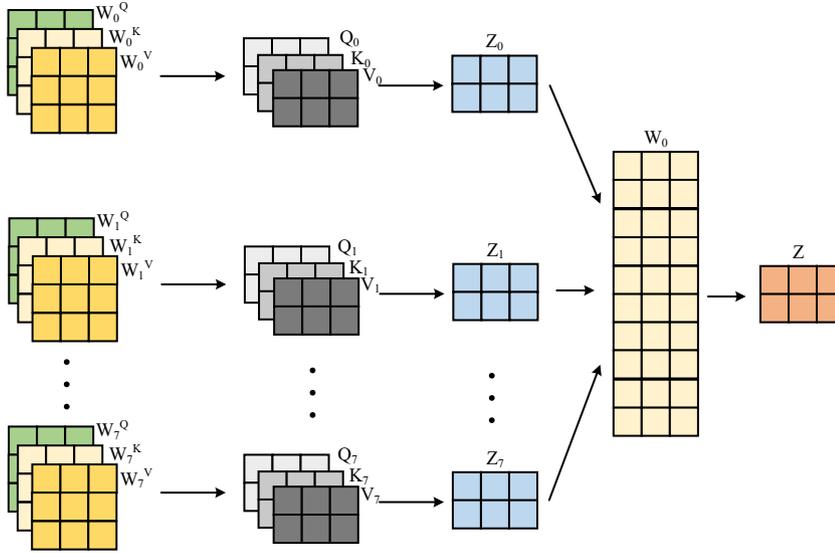


Figure 3.3: Matrix operation process of multi-head attention mechanism.

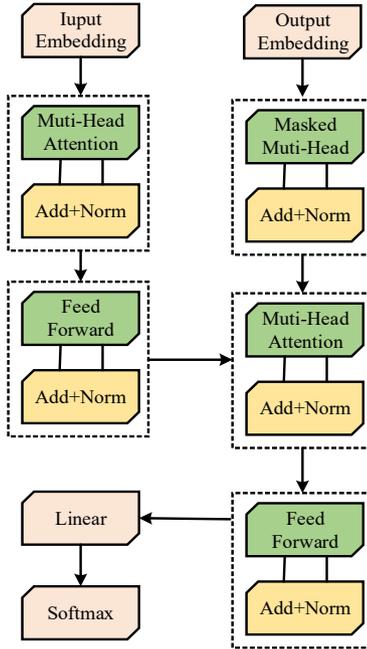


Figure 3.4: Hierarchy of encoder and decoder.

When correcting grammatical errors, they are usually reflected in the use of individual words in sentences. The rest is usually grammatically correct. To avoid the disturbing effect of grammar correction algorithm on sentences without grammatical errors, a Transformer model based on replication mechanism is proposed. It can directly copy words without grammatical errors into the target sentence. The research takes an English grammar correction task as a machine translation task. For the input of source statement (x_1, \dots, x_N) containing syntax errors, the target statement

(y_1, \dots, y_T) is corrected accordingly. Formula (3.3) is the sequence generation process of the correction sentence.

$$\begin{cases} h_{1,\dots,N}^{\text{src}} = \text{encoder}(L^{\text{src}}x_{1,\dots,N}) \\ h_t = \text{decoder}(L^{\text{trg}}y_{t-1,\dots,1}, h_{1,\dots,N}^{\text{src}}) \\ S_t(g) = \text{softmax}(L^t h_t) \end{cases} \quad (3.3)$$

In formula (3), L represents the word embedding matrix. $h_{1,\dots,N}^s$ represents the implicit state after the source statement is encoded by the encoder. h_t indicates the implicit state of the prediction target word. $S_t(g)$ is the probability distribution of the target word. At present, the number of sentences with grammatical error correction and manual tagging is small, which greatly limits the research progress of the algorithm. The research will increase the training data by creating pseudo-parallel sentence pairs, and explore the impact of different generation methods of pseudo-parallel sentence pairs on model performance.

3.2 Copy mechanism and its application

The study uses three methods to generate pseudo parallel corpora to expand training data for grammar error correction, as shown below. (1) Noise is added directly to the correct sentence. By performing BPE word segmentation on the sentence and performing deletion, replacement, and occlusion operations with a certain probability are performed to create grammar errors. (2) Real language learning errors are simulated. Common errors are collected in the English learning. Moreover, a word modification dictionary is created, which is used to perform specified probability

substitutions, synonym substitutions, or part of speech perturbations on words in high-quality corpus. (3) A grammar error generation model is trained. Utilizing existing manually annotated data, a model is trained to generate incorrect sentences and expand the dataset. A correct sentence sequence $x = (x_1, x_2, \dots, x_n)$ is processed and the corresponding wrong sentence $y = (y_1, y_2, \dots, y_m)$ is generated are assumed. The probability of sentence syntax error generation and the loss function of the model are shown in formulas (3.4) and (3.5). The parameter σ is trained by minimizing the loss function.

$$s(x|y) = \prod_{t=1}^m s(x_t|y, x_{1:t-1}; \sigma) \quad (3.4)$$

$$L(\sigma) = \sum_{t=1}^m -\log s(x_t|y, x_{1:t-1}; \sigma) \quad (3.5)$$

Through the method of formula (3.4), a grammar error generation model can be trained. English corpus without grammar errors can be input into the model. Moreover, the English text with grammar errors can be obtained, thus generating a large number of parallel sentence pairs. The Transformer model based on the replication mechanism adopted in the study directly copies the correct words into the target sentence. The probability distribution of words in the target sentence is the combination of the probability distribution S_t^{ge} generated by the error correction model and the probability distribution S_t^{co} copied by the source sentence. The specific principle is shown in formula (3.6).

$$S_t(g) = (1 - \beta_t^{co}) \cdot s_t^{ge}(g) + \beta_t^{co} \cdot s_t^{co}(g) \quad (3.6)$$

In formula (3.6), $\beta_t^{co} \in [0, 1]$, and β_t^{co} represents the balance parameter of each time step t used to control the generation probability and replication probability. Figure 3.5 shows

the structure of Transformer model based on replication mechanism.

Figure 3.5 shows the structure of the model in detail. The probability distribution of the target word is generated by the basic Transformer model. The implicit state $h_{1, \dots, N}^{src}(H^{src})$ of the input word of the source statement and the implicit state of the target word are used to solve the copy score. The solution of copy-based attention score is consistent with that of conventional Transformer attention score, as shown in formulas (3.7) and (3.8).

$$q_t, K, V = h_t^{trg} W_q^T, H^{src} W_k^T, H^{src} W_v^T \quad (3.7)$$

$$S_t^{co}(g) = \text{softmax}(R_t) \quad (3.8)$$

In formula (3.7), q_t, K, V are the query, key, and value vectors used to calculate the attention distribution and copy the hidden layer. The attention distribution obtained by normalization is taken as the copy score, and the balance parameter β_t^{co} is estimated by copying the hidden layer, as shown in formula (3.9).

$$\beta_t^{co} = \text{sigmoid}\left(W^t \sum \left(\frac{R}{t} \cdot V\right)\right) \quad (3.9)$$

The loss function of the model is shown in formula (3.10).

3.3 Summary of deep classification grammar correction algorithm process

Based on the above description, the overall process of the deep classification based grammar correction algorithm model designed for research is shown in Figure 3.6.

As shown in Figure 3.6, the deep classification syntax correction model proposed by the research is built around

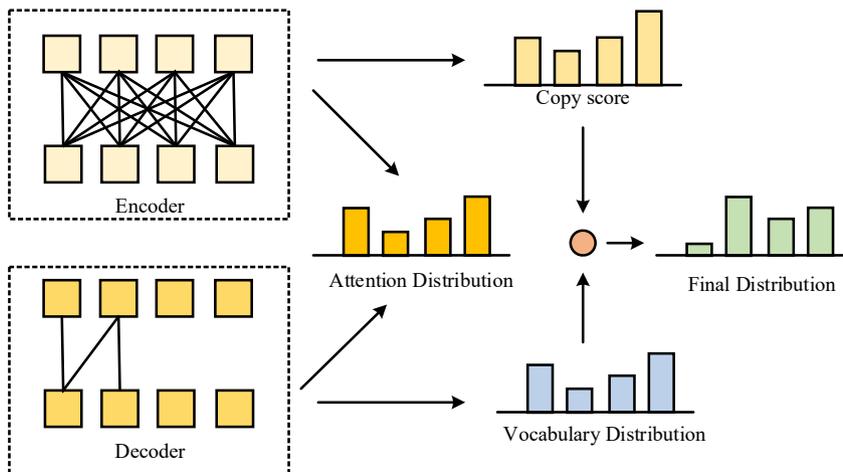


Figure 3.5: Transformer model structure based on replication mechanism.

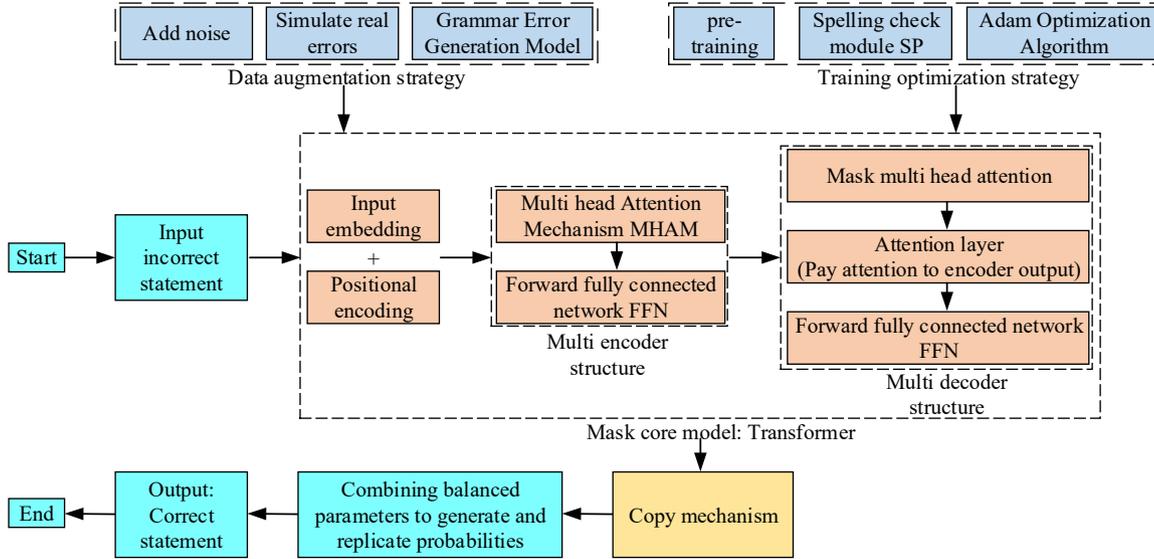


Figure 3.6: The overall process of syntax correction algorithm model based on deep classification.

a Transformer model based on multi-head attention mechanism (MHAM). First, the SAM is utilized to achieve sequence parallel computing and long-distance dependency capture. Attention scores are generated through matrix operations of query vectors (q), key vectors (k), and value vectors (V). Multiple SAMs are then combined into an MHAM to obtain multidimensional semantic information. Additionally, positional encoding is introduced to address the Transformer model's insensitivity to word order by incorporating sentence order information. The Transformer model includes 6 layers of encoders (including MHAM and forward fully connected network) and decoders (with additional attention layers). Second, three methods are used to generate a pseudo-parallel corpus to expand the training data. A replication mechanism is also introduced to enable the model to directly replicate grammatically correct words in the target sentence. By balancing parameters and combining the error correction model to generate probabilities and source sentence replication probabilities, the final probability distribution of the target word is obtained. Finally, the Adam algorithm is used to optimize the model parameters, while a dual optimization strategy is employed to further improve the model performance, resulting in the final SP + humans + TF_{copy} algorithm.

3.4 Model evaluation metrics

To comprehensively and objectively assess the performance of the grammar error correction model, three core metrics and one auxiliary metric are adopted.

1. Accuracy: Measuring the proportion of correctly corrected errors among all error correction attempts made by the model is shown in formula (3.10).

$$\text{Accuracy} = \frac{TP}{TP + FP} \quad (3.10)$$

TP (true positive) represents the number of errors that are actually present and correctly corrected by the model, and FP (false positive) represents the number of non errors that are incorrectly identified and modified by the model. This metric focuses on reducing “excessive error correction” to ensure the reliability of the model's outputs.

2. Recall: Measuring the proportion of correctly corrected errors among all actual errors in the test data is shown in formula (3.11).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.11)$$

FN (false negative) represents the number of errors that are actually present but not detected by the model. This metric reflects the model's ability to identify errors, especially edge and complex errors.

3. $F_{0.5}$ -Score: A weighted harmonic mean of Accuracy and Recall, with a weight of 0.5 assigned to Recall and 2 to Accuracy, is shown in formula (3.12).

$$F_{0.5} = \frac{(1 + 0.5^2) \times \text{Accuracy} \times \text{Recall}}{0.5^2 \times \text{Accuracy} + \text{Recall}} \quad (3.12)$$

This metric prioritizes Accuracy over Recall, which aligns with the research goal of avoiding excessive

error correction in online teaching scenarios while maintaining the ability to identify basic errors.

4 Algorithm performance analysis

4.1 Definition of basic parameters and optimization strategies for experimental design

The experimental content is implemented on the Python 1.0 platform, and the fairseq framework is used to train the model, while the Tesla V100 is used to accelerate the model training process. In this study, “SP + Rehuman + TF_{copy}” represents a complete system that integrates spell-checking modules and uses pre-trained Transformer models based on replication mechanisms that simulate human error-generated corpora. “SP” represents the spell-checking module, which detects and corrects spelling errors before the input text enters the main grammar correction model. “Pre” stands for pre-training, which refers to pre-training the model using the three methods described in Section 3.2 (directly adding noise, simulating real errors, and training error generated models) to generate large-scale pseudo parallel corpora. The suffixes “man” and “human” specifically refer to the method of generating pre-trained corpora, while “Preman” indicates the use of artificially defined rules to generate pseudo parallel corpora for pre-training. “Prehuman” refers to pre-training pseudo parallel corpora generated by simulating real language learning errors or using error generation models trained on manually annotated data, which are closer to real errors. “TF_{copy}” refers to the Transformer model based on the replication mechanism proposed in this study. “Double optimization” refers to the simultaneous use of two optimization strategies. One is to introduce a spell checking module, and the other is to use pseudo parallel corpus for pre-training.

4.2 Experiment 1: comparison of different transformer network structures

For experiment 1, three different network structures based on Transformer are used to train grammatical error correction tasks, and the error correction results are compared. The three methods are the basic TF_{base}, the TF_{more} that expands the dimensions and increases the number of attention heads at each level on this basis, and the Transformer model based on the replication mechanism proposed by the research, namely TF_{copy} model. Table 4.1 shows the structural parameters of three different models.

Table 4.1: Structural parameters of three different Models.

	TF _{base}	TF _{more}	TF _{copy}
Word embedding dimension	512	1,024	512
Number of attention heads	8	16	8
Hidden layer dimension of feedforward network	2,048	4,096	4,096
Number of network layers of encoder and decoder	6	6	6

The dropout value is 0.3 and the initial learning rate is 1×10^{-4} , the minimum learning rate value is 1×10^{-9} , and the number of max-tokens is 4,000. Adam algorithm is used to optimize the model so that the adjustment of learning rate can be automated. $\chi_1 = 0.9$. $\chi_2 = 0.99$.

4.3 Experiment 2: training and fine tuning of TF_{copy} model

For experiment 2, TF_{copy} is used for training and fine-tuning of the model, and the parameter settings of other experimental environments and training stages are the same as experiment 1. The hidden layer dimension of the feed-forward network of the reverse syntax error generation model is 1,024, and the number of attention heads is 4. Lang-8, NUCLE, W&I_dev, and FCE_dev are used for model training, including one million sentences. There are two datasets: W&I_dev, FCE_dev for verification and adoption, including 7,000 sentences. The test set uses the CoNLL-2014 dataset, which includes 5,000 sentences. The results of the error correction model after the introduction of the spelling check module on the CoNLL-2014 testing dataset are compared with those of the other two models, as shown in Figure 4.1.

Sub-fig. (c) in Figure 4.1 shows the comparison results of the bias model formed by the designed model under different optimization settings. The accuracy value of TF_{copy} + SP model is 68.76 %, the recall value is 33.48 %, and the F_{0.5} value is 53.31 %. All values are the highest in the same basic optimization setting model. The second accurate is 62.18 % of TF_{copy}, the second recall is 32.33 % of TF_{more} + SP, and the second accurate is 51.65 % of TF_{more} + SP. The same basic optimization model that is close to TF_{copy} + SP in terms of overall indicators does not exist, and TF_{more} + SP is the only model with the second performance. TF_{copy} + SP with the best optimized setting is compared with other models, and the peak accuracy of the best model under the MCNN and SMT model cluster is 67.06 %. The peak recall rate of the best model is 23.17 %, and the peak F_{0.5} value of the best model is 49.73 %, both of which are smaller than the relevant values of the TF_{copy} + SP model. At the same time, in the classifier model, the peak accuracy rate of the best model is 60.19 %, and the peak recall rate of the best model is 23.17 %, and the peak F_{0.5} value of the best model is 49.73 %, both of which are smaller than the relevant values of the TF_{copy} + SP model.

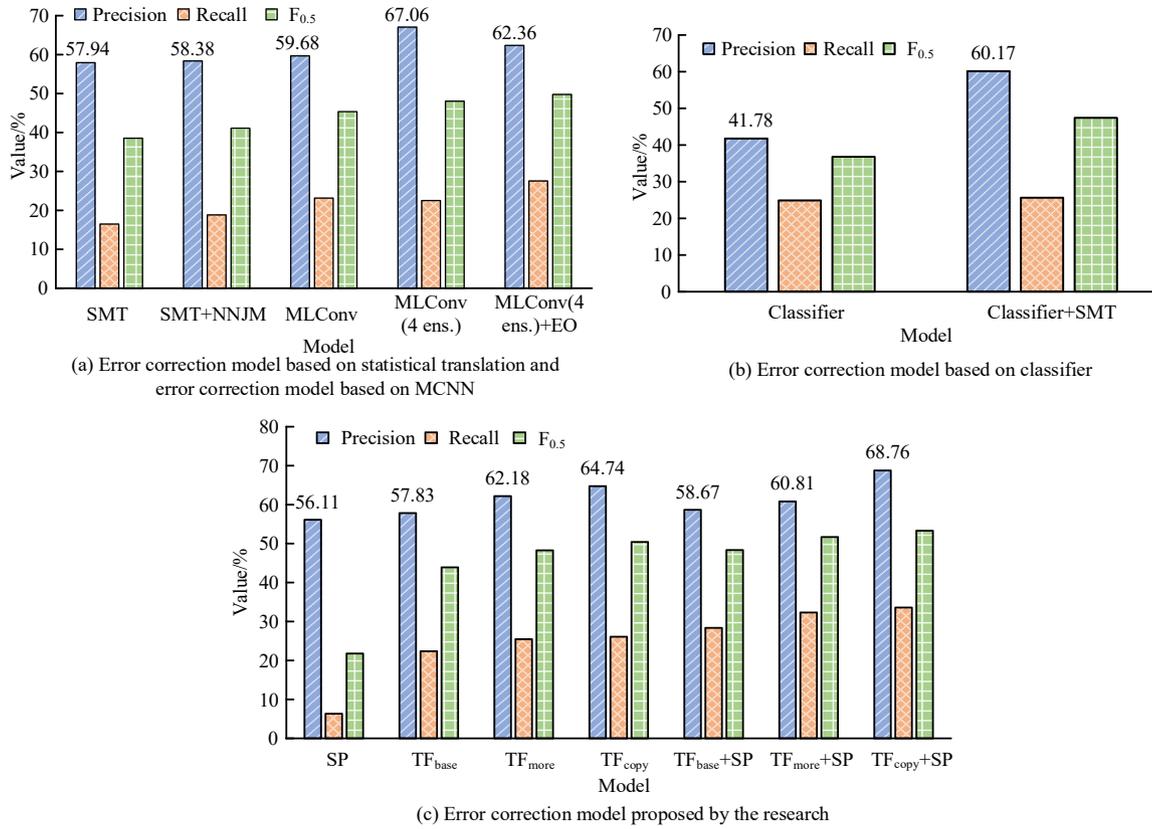


Figure 4.1: Results of different error correction models on the CoNLL-2014 training dataset.

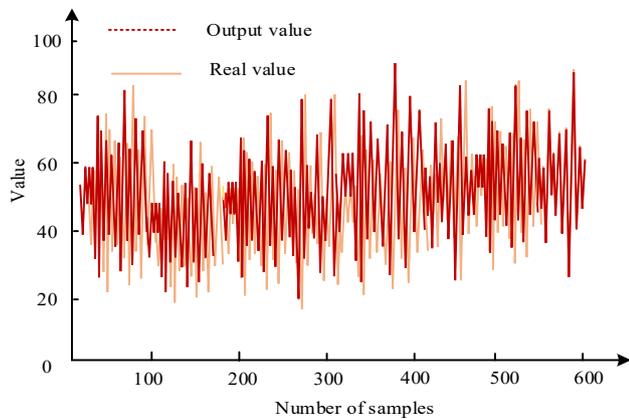


Figure 4.2: Output status.

the peak recall rate of the best model is 25.68 %, and the peak F_{0.5} value of the best model is 17.45 %. Both of them are less than the relevant values of the TF_{copy} + SP model. The output effect of the model is shown in Figure 4.2.

From Figure 4.2, the output value of the study is close to the real value, and the overall trend is consistent, indicating that the model has certain accuracy in grammar error detection and classification.

4.4 Comparison of the effectiveness of pre-training pseudo parallel corpora

The effect of the pre-training model and the addition of the spell check module on the CoNLL-2014 training dataset is shown in Figure 4.3.

From Figure 4.3, the accuracy rate of SP + Preman algorithm reaches 60.52 % among all optimized algorithms, which is the highest accuracy rate among all comparison models. The recall rate of SP + Preman algorithm reached 31.46 %, which is not the highest value in the comparison model, but is close to the highest value. The F_{0.5} value of SP + Preman algorithm reaches 48.68 %, which is also the highest accuracy of all comparison models.

4.5 Performance verification of dual optimization model (SP + humans + TF_{copy})

In general, the SP + Preman optimization setting algorithm designed by the research has the best performance. The test results of the CoNLL-2014 test set are shown in Figure 4.4.

In Figure 4.4, the SP + Prehuman + TF_{copy} algorithm with dual optimization settings performs well on the

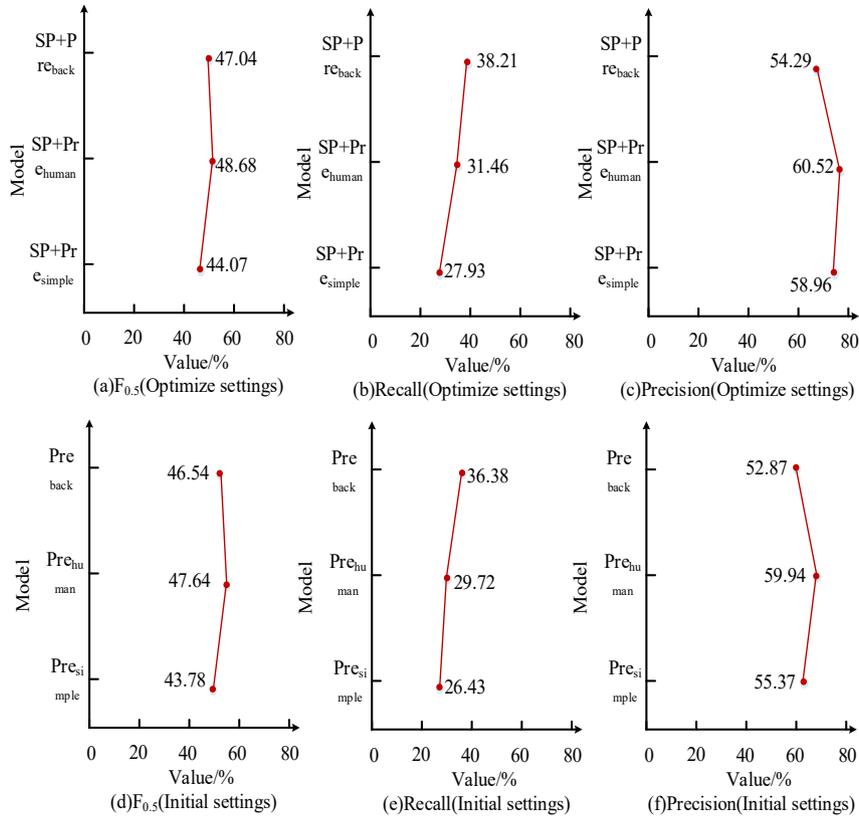


Figure 4.3: Effect of pre-training and spell check.

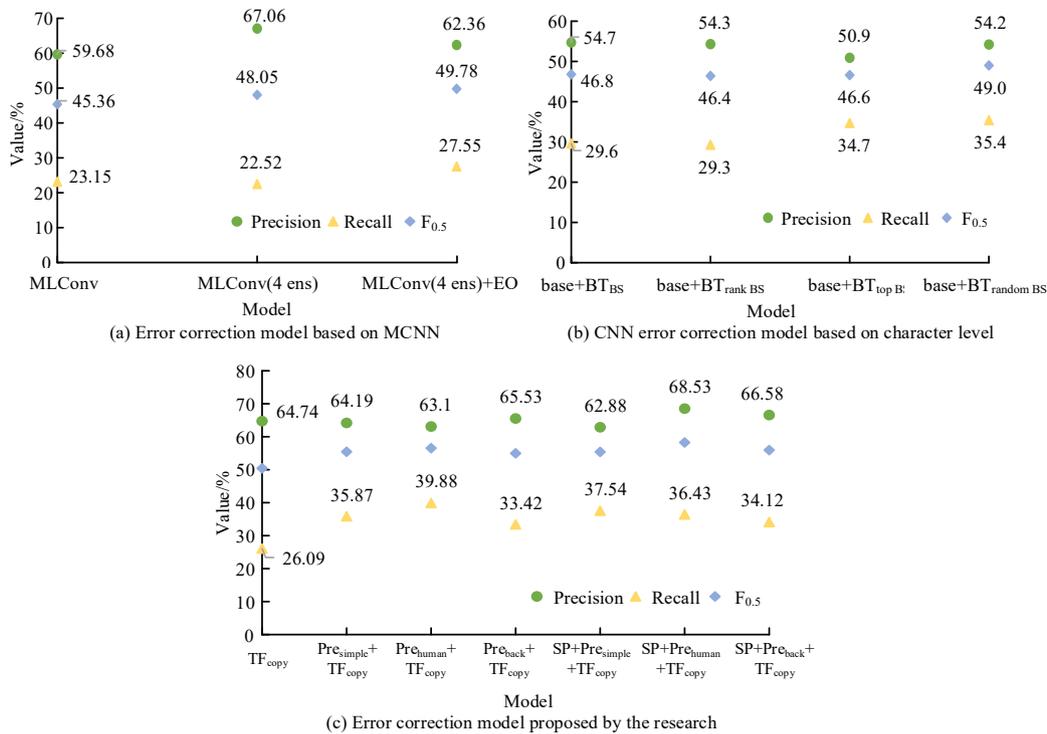


Figure 4.4: Results of different models on the CoNLL-2014 test set.

CoNLL-2014 test set in this study, with an accuracy of 68.53 %, ranking first among all compared algorithms. At the same time, the F_5 value of the algorithm also reaches 58.26 %, the highest among similar model-based optimization algorithms and other types of models. In the horizontal comparison, the highest accuracy of the error correction model based on MCNN is about 67.06 %, while the highest accuracy of the error correction model based on character level CNN is about 54.70 %. Both lower than the algorithm in this study. In terms of recall rate, the SP + Prehuman + TF_{copy} algorithm reaches 36.43 %, which is higher than the best model based on MCNN (about 27.55 %) and also better than the model based on character level CNN (about 34.70 %). Based on various indicators, the SP + Prehuman + TF_{copy} algorithm performs the best overall on the CoNLL-2014 test set. The practical performance effects for different syntax error types are shown in Figure 4.5.

From Figure 4.5, the MLConv (4 ens)+EO model with relatively high $F_{0.5}$ value is selected for comparison. The accuracy rate, recall rate, and $F_{0.5}$ value of the designed model in singular and plural errors of nouns are 63.53 %, 53.46 %, and 66.67 %, respectively, which are higher than 57.51 %, 40.32 %, and 64.37 % of MLConv (4 ens)+EO model. The accuracy rate of prepositional errors and $F_{0.5}$ value of the designed model are 23.64 %, 8.92 %, and 40.24 %, respectively, which are higher than the 23.26 % and 36.74 % of MLConv (4 ens)+EO model. The model designed in the study is inferior in the recall rate, but the overall gap is not large, which does not affect the comprehensive performance judgment. At the same time, the accuracy rate, recall rate, and $F_{0.5}$ value of the designed model are 56.37 %, 43.68 % and 60.79 % respectively, which are higher than 55.67 %, 37.84 % and 58.2 % of MLConv (4 ens)+EO model. To sum up, the designed model is only slightly inferior in the recall rate, but the gap is not large. It shows significant advantages in other performance values. The grammar correction model designed by the research can provide an effective teaching aid for online English teaching. The low overall recall rate is

due to the fact that the research model focuses on improving error correction accuracy by introducing replication mechanisms and dual optimization strategies. In terms of parameter settings and loss function optimization, it places more emphasis on reducing the risk of “excessive error correction”. This “guarantee all” strategy sacrifices the ability to identify some edges and complex errors while improving accuracy (up to 68.53 %).

4.6 Analysis of error correction effectiveness for specific grammar error types

The effectiveness of the SP + Prehuman + TF_{copy} grammar correction model is further illustrated in practical applications. A computer with 16 GB of RAM is used for testing in the Matlab2017a and Windows 10 operating system. 1,200 online English assignments from A school’s first, second, and third year students are selected as the test subjects, which contain 322,56 English sentences. The accuracy of correcting singular and plural errors, preposition errors, article errors, adjective errors, and pronoun errors are tested in these assignments. The specific results are shown in Figure 4.6.

Figure 4.6 shows the error correction test results of online English homework in high school using the error correction model. From Figure 4.6(a), the error correction accuracy of high school students’ homework is relatively high in all parts, especially for adjective errors, reaching 95 %, while the accuracy of article errors is relatively low, at 89 %. From Figure 4.6(b), the overall error correction accuracy of high school sophomores’ assignments is slightly lower than that of high school freshmen. The error correction accuracy of preposition and adjective errors is the same, both at 90 %, while the error correction accuracy of pronoun errors is the lowest, at 86 %. From Figure 4.6(c), the error correction accuracy of high school students’ homework continues to decline, with the error correction accuracy of article errors being 88 %, which is the same as high school students’, while the error correction accuracy of adjective errors is

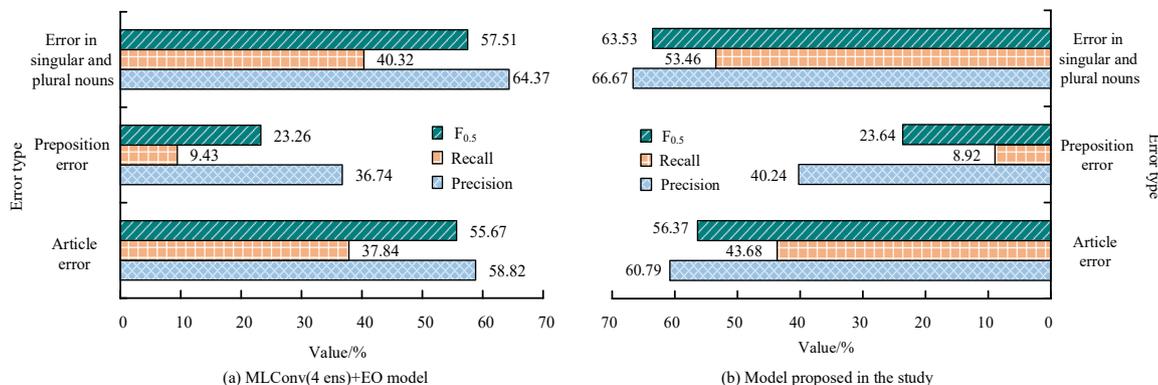


Figure 4.5: Comparison of effects of different models in specific error types.

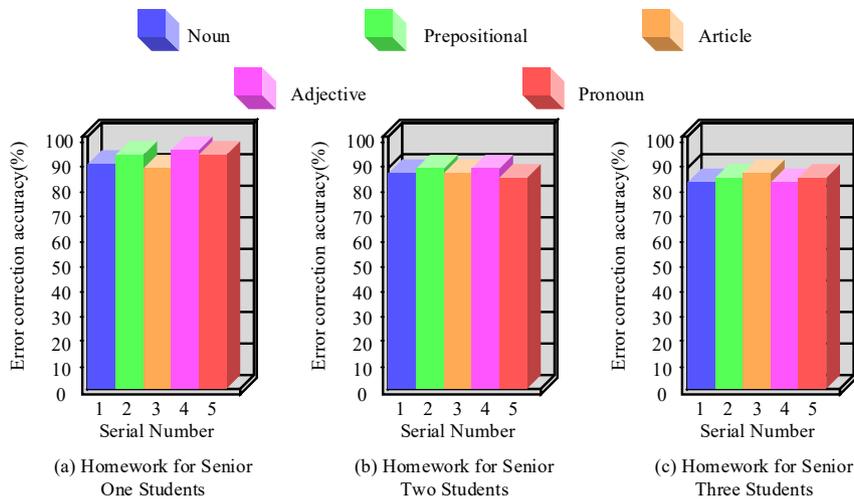


Figure 4.6: Grammar test results of English homework for different grades in high school.

the lowest, only 83 %. The accuracy slightly decreases with grade, as older students' English assignments may be more complex, using more advanced and diverse grammatical structures and vocabulary, which may lead to challenges for error recognition and correction. Second, the training data of the error correction model may not be sufficient when facing the more complex language characteristics of older students, resulting in a decrease in the model's coping ability. Although the accuracy has decreased, the error correction accuracy is still at a high level, indicating that the model is still effective in practical applications and can provide reliable error correction services for students of different grades.

Providing more immediate and accurate feedback on grammar corrections can enhance students' engagement and confidence in learning, as well as promote long-term knowledge retention. For example, when a student repeatedly makes errors in the third person singular and plural forms of the verb "He go to school" in their essay, the system can immediately highlight the error and provide a correction suggestion of "He goes to school", while giving a concise rule prompt of "adding -s to the verb after the third person singular subject". This instant feedback avoids the solidification of error patterns and improves students' accuracy of the grammar point in their next writing. Meanwhile, the system can adapt to different levels of language complexity, provide teachers with efficient teaching aids, optimize teaching processes, and offer personalized learning support. For example, when writing argumentative essays in the third year of high school, the system can recognize not only basic subject-verb agreement errors, but also redundant expressions in complex sentence structures, such as "desire the fact that", and suggest more precise "although"

structures. This helps high-level students improve the accuracy and conciseness of their language. Teachers use the "Error Type Statistics Dashboard" in the system backend to identify collective weaknesses in the use of articles in class. They then design targeted thematic training, transforming unified teaching into precise, personalized guidance.

5 Conclusions

Aiming at the problem of insufficient grammar teaching resources in classroom English teaching, a deep classification grammar correction algorithm for online English grammar teaching was researched and designed. The core achievements could be summarized into two aspects. One was to construct an architecture based on MHAM improved Transformer model as the core, which solved the problem of missing word order information by introducing position encoding. It effectively reduced the interference of incorrect sentence pairs during model training through the integration of a replication mechanism. Simultaneously, it employed three pseudo-parallel corpus generation methods to expand the volume of training data. Finally, the SP + Prehuman + TF_{copy} algorithm formed by the dual optimization of "spelling check module (SP)+pseudo parallel corpus pre-training" achieved an accuracy of 68.53 % and $F_{0.5}$ value of 58.26 % on the CoNLL-2014 test set, both of which were the highest compared models. Moreover, the error correction effect was better than the MLConv (4 ens)+EO model on typical errors such as singular and plural nouns and articles. Second, the model performed stably in real teaching scenarios. It had an error correction accuracy rate of over 83 % for 1200 English assignments across three high school grades. It could adapt to the difficulty of assignments in

different grades and could also assist teachers in achieving precise teaching through the “Error Type Statistics Panel”. Through real-time feedback, it helped students avoid fixed error patterns.

At the same time, the research has certain limitations. The model had a low recall rate of 33–36 %, which sacrificed the ability to identify edge and hidden errors because accuracy was prioritized. There were differences between pseudo parallel corpora and personalized errors in real language learning, and there were insufficient training samples for complex and difficult sentences in higher grades. The model lacked the ability to dynamically adapt to individual student error characteristics, and personalized support only stayed at the group level.

This study’s broad significance lies in providing a new approach to online English teaching: “technology-empowered precision teaching.” This approach solves the problems of feedback lag and uneven resources in traditional teaching. In the field of NLP, the effectiveness of the Transformer model when combined with a replication mechanism and a pseudo-parallel corpus for grammar correction tasks has been verified. This provides a reference for text correction research in scenarios with limited resources.

Therefore, future research can be improved from three aspects. One is to optimize the design of the loss function, introduce a recall accuracy balance mechanism, and enhance the ability to identify implicit and edge errors. The second is to construct a multimodal corpus that integrates erroneous data from real students, and enhances the authenticity and pertinence of the data by combining tags such as native language background and learning stage. The third is to add a student portrait module. This module can adjust personalized error correction strategies by tracking individual error trajectories. This further enhances the model’s adaptability in teaching scenarios.

Acknowledgments: Not applicable.

Research ethics: Not applicable.

Informed consent: Confirming that informed consent was obtained from all subjects and/or their legal guardian(s); this includes information regarding informed consent obtained from the study participant’s parent or legal guardian for any participant below the age of consent.

Author contribution: Zhu Xiao wrote the main manuscript text, collected the data and prepared tables, did the data analysis, designed and conducted the research.

Use of Large Language Models, AI and Machine Learning Tools: None declared.

Conflict of interest: The authors declare no competing interests.

Funding information: This research received no external funding.

Data availability statement: Not applicable.

References

- [1] Z. Zhao and H. Wang, “Maskgrec: Improving neural grammatical error correction via dynamic masking,” *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, pp. 1226–1233, 2020.
- [2] B. Chen and J. Zhang, “Pre-training-based grammatical error correction model for the written language of Chinese hearing impaired students,” *IEEE Access*, vol. 10, pp. 35061–35072, 2022.
- [3] J. H. Min, S. J. Jung, S. H. Jung, Y. Seongmin, and S. C. Jun, “Grammatical error correction models for Korean language via pre-trained denoising,” *Quant. Bio-Sci.*, vol. 39, no. 1, pp. 17–24, 2020.
- [4] Y. Zhong and X. Yue, “On the correction of errors in English grammar by deep learning,” *J. Intell. Syst.*, vol. 31, no. 1, pp. 260–270, 2022.
- [5] S. Li *et al.*, “Chinese grammatical error correction based on convolutional sequence to sequence model,” *IEEE Access*, vol. 7, pp. 72905–72913, 2019.
- [6] Y. Zhang, “Application of intelligent grammar error correction system following deep learning algorithm in English teaching,” *Int. J. Grid Util. Comput.*, vol. 13, nos. 2-3, pp. 128–137, 2022.
- [7] K. N. Acheampong and W. Tian, “Toward perfect neural cascading architecture for grammatical error correction,” *Appl. Intell.*, vol. 51, no. 6, pp. 3775–3788, 2021.
- [8] S. Koo *et al.*, “K-NCT: Korean neural grammatical error correction gold-standard test set using novel error type classification criteria,” *IEEE Access*, vol. 10, pp. 118167–118175, 2022.
- [9] J. Lichtarge, C. Alberti, and S. Kumar, “Data weighted training strategies for grammatical error correction,” *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 634–646, 2020.
- [10] Z. Gong, “English grammar auto-correction robot based on grammatical error generation model,” *Scalable Comput.: Pract. Exp.*, vol. 25, no. 6, pp. 7780–7788, 2024.
- [11] F. Xiao and S. Yin, “English grammar intelligent error correction technology based on the n-gram language model,” *J. Intell. Syst.*, vol. 33, pp. 28–30, 2024.
- [12] S. Kasmaiee, S. Kasmaiee, and M. Homayounpour, “Correcting spelling mistakes in Persian texts with rules and deep learning methods,” *Sci. Rep.*, vol. 13, no. 1, pp. 45–50, 2023.
- [13] A. N. Golubinsky, A. A. Tolstykh, and M. Y. Tolstykh, “Automatic generation of scientific paper abstracts based on large language models,” *Inform. Autom.*, vol. 24, no. 1, pp. 275–301, 2025.
- [14] A. A. Rafique, M. Gochoo, A. Jalal, and K. Kim, “Maximum entropy scaled super pixels segmentation for multi-object detection and scene recognition via deep belief network,” *Multimed. Tool. Appl.*, vol. 82, no. 9, pp. 13401–13430, 2023.

- [15] J. Kamel and P. Alexander, "Sentiment analysis of Arabic tweets using SVM classifier with POS tagging features," *Int. J. Open Inf. Technol.*, vol. 11, no. 6, pp. 29–37, 2023.
- [16] T. G. Voloshina, A. A. Mustafaeva, and E. A. Bocharova, "Lexis and grammar specifics of the English Language in Africa," *Proc. Southwest State Univ.*, vol. 14, no. 1, pp. 57–64, 2024.
- [17] S. Lei and Y. Li, "English machine translation system based on neural network algorithm," *Procedia Comput. Sci.*, vol. 228, pp. 409–420, 2023.
- [18] N. Hossain, M. H. Bijoy, S. Islam, and S. Shatabda, "Panini: A transformer-based grammatical error correction method for Bangla," *Neural Comput. Appl.*, vol. 36, no. 7, pp. 3463–3477, 2024.
- [19] R. Tinn *et al.*, "Fine-tuning large neural language models for biomedical natural language processing," *Patterns*, vol. 4, no. 4, pp. 63–68, 2023.
- [20] G. Kortemeyer, "Performance of the pre-trained large language model GPT-4 on automated short answer grading," *Discov. Artif. Intell.*, vol. 4, no. 1, pp. 47–48, 2024.
- [21] S. Leem and H. Seo, "Attention guided CAM: Visual explanations of vision transformer guided by self-attention," *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 4, pp. 2956–2964, 2024.
- [22] Y. Wang *et al.*, "A new stable and interpretable flood forecasting model combining multi-head attention mechanism and multiple linear regression," *J. Hydroinform.*, vol. 25, no. 6, pp. 2561–2588, 2023.