Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
 Volume 25, Issues 6, pp. 5688–5700, DOI 10.12694/scpe.v25i6.2171

ENGLISH GRAMMAR AUTO-CORRECTION ROBOT BASED ON GRAMMATICAL ERROR GENERATION MODEL

ZHU GONG*

Abstract. The traditional English grammar error correction system has problems such as poor error recognition precision and error correction success rate to be improved. Therefore, in this study, data augmentation techniques were used to transform and process error correcting English texts, and a rule-based and shallow neural network-based English text grammar correction model was constructed. The experimental results showed that the grammar error generation (GEG), rule-based (RB), classification-based (CB), and recurrent neural network (RNN) models achieved accuracy rates of 93.92%, 82.17%, 79.41%, and 88.09% in correcting grammar errors on 1702641 test sentences in the One Billion word corpus, respectively. The experimental results showed that the English grammar error correction model designed in this study had a strong error correction ability, but the computational efficiency was low. The research results significantly improved the accuracy and generalization ability of English grammar correction, optimized learning costs, and brought positive impacts to educational applications, providing strong support for the development of intelligent English grammar correction.

Key words: English; Grammatical error correction; Data augmentation; Natural language

1. Introduction. Grammar error correction (GEC) is a computer-assisted language task that is often applied in natural language processing scenarios. The essence of GEC work is to analyze the internal logic and grammatical dependencies of input language text information, to detect and correct grammar errors in language information [1]. After entering the 21st century, the rapid development of deep learning technology has greatly promoted the commercial implementation of natural language work. However, automatic detection and correction products for English grammar errors in natural language processing still have high difficulty in marketization, which is the main background of this study. This is related to the complexity of English grammar, the variety of grammar error types, the scarcity and high production cost of annotated data samples, and the dependency between semantics. With the development of online education, a large number of English learners are starting to learn English on the internet. Beginners are prone to making mistakes in English grammar, which affects the learning desire of many students [2]. In the context of the rapid development of computer technology in the past decade, there has been an increasing number of studies using advanced computer technology for intelligent detection and correction of English grammar. However, most of the error correction models proposed in these studies often struggle to accurately identify and correct grammar errors in text, especially when faced with complex sentence structures and diverse types of grammar errors, resulting in a significant decrease in recognition accuracy. Meanwhile, many existing error correction models are prone to introducing new errors or failing to fully correct existing errors when attempting to correct syntax errors, resulting in a low success rate of error correction and affecting user experience. In addition, due to the complexity of English grammar and the scarcity of annotated data, the generalization ability and error correction effect of these error correction models are relatively low. Existing research showed that mixed attention mechanisms can focus on different parts of text, enabling models to more accurately capture key information in the text and improve the accuracy of grammar error recognition. Rule-based (RB) models can provide a stable grammar checking foundation, while shallow neural networks can compensate for the shortcomings of RB models by learning patterns from large amounts of data, enhancing the flexibility and adaptability of the models. In light of these considerations, the study advanced an innovative approach by integrating RB GEC models, attention mechanisms, and shallow neural networks to develop an enhanced English GEC model. This approach effectively addressed the limitations of existing solutions in error correction accuracy, generalization ability, data dependency, and flexibility. Moreover,

*School of Foreign Languages, Sichuan Vocational and Technical College, Suining 629000, China (Zhuu_Gong@163.com)

it offered novel insights and methodologies for the advancement of intelligent English grammar correction.

The structure of this study is as follows. The first part introduces the basic concepts of GEC, as well as the development background and implementation technology route of this task. The second part provides a detailed design of a new hybrid English GEC model. The third part conducts experimental verification on the model and compares it with other models. Finally, the fourth part summarizes the entire article and provides prospects for future research directions.

2. Related works. Real-time English natural language information in real application scenarios often has certain grammatical errors. To reduce such errors, intelligent GEC systems based on intelligent technology and computer technology have been developed. The task of natural language processing of text types including grammatical errors has been the focus of research by computer experts and linguists. Solyman et al. found that the GEC seq2seq model with multiple encoder and decoder layers had a key drawback, namely, the existence of exposure bias problem during inference, which led to the deletion of some previous target words and reduced the quality of grammatical correction. Therefore, the research team proposed a seq2seq Transformer-based GEC model to solve these problems. Furthermore, to overcome the problem of disclosure bias, a two-way regularization term was introduced in the training objective using the Kullback-Leibler scatter to improve the consistency between the from-right-to-left and from-left-to-right models. Experiments conducted on two standard test datasets QALB-2014 and QALB-2015 showed that the model proposed in this study obtained the best F1 marks than the existing Arabic GEC system [3]. Pajk et al. employed existing pre-trained multilingual models to address the deficiency in multilingual solutions for the GEC problem, with the objective of correcting grammatical errors. They also investigated the influence of diverse pre-training techniques on the ultimate GEC quality and conducted experiments to identify a unified GEC model capable of rectifying seven languages [4]. Wang et al. systematically analyzed the current data augmentation processing methods applied to the GEC problem and the advantages and disadvantages of various mainstream GEC methods and elucidated the future direction of the GEC industry [5]. Choi found that deep neural networks and pre-training models were positive for improving the performance of GEC models, so a Korean GEC model based on improved convolutional neural networks and pre-training methods was designed. The test results showed that the accuracy of Korean GEC of the model was significantly higher than that of the unimproved model and several classical GEC models [6]. Witteloostuijn et al. found that patients with developmental dyslexia were poor at recognizing grammatical errors in linguistic information, so they improved the data augmentation method and designed a GEC model based on the idea of categorical language. The experimental outcomes denoted that students who used this model for assisted reading showed significant improvements in reading comprehension accuracy and average reading duration [7]. Linarsih et al. found that for foreign language beginners, they were vulnerable to grammatical problems when reading linguistic information in a non-native language, so an improved GEC model incorporating recurrent neural networks (RNNs) was designed. The performance of the designed model was tested using several datasets commonly used in the GEC industry. The outcomes indicated that the accuracy of the model in correcting grammatical errors in each selected dataset was on average 10.24% higher than that of the comparison method. However, in terms of computational speed, the former did not have a significant advantage over the latter, which was mainly brought about by the complexity of the RNN's own structure. Further improvements were expected in subsequent studies [8]. Jiang F et al. found that the grammar correction module could greatly affect the quality of speech recognition and response in speech intelligent question and answer systems. Therefore, an improved grammar correction module for speech intelligent question and answer systems was constructed using neural machine translation technology and RNN-based language model. The experimental analysis findings expressed that the speech intelligent question and answer system installed with the improved model proposed in this study showed better speech response capability [9]. Putra's team designed an improved GEC system based on the ternary language model to cope with the problem of unstable recognition in the GEC model and asked several volunteers to use and try this system. It rated the experience of using this system higher than the preim proved GEC system [10]. Koyama S addressed the issue of insufficient training data leading to poor correction results in neural GEC. This paper proposed designing multiple error generation rules for different grammar categories and combining these rules for data augmentation. The results showed that the method proposed in this article could also train high-performance models under unsupervised settings, and could more effectively correct writing errors compared to models based on round-trip translation [11]. Zhang J proposed a hybrid method combining bidirectional encoder representation from transformers (BERT) model (utilizing syntactic information and context embedding) and dictionary-based graph neural network (utilizing lexical information) to automatically detect grammar errors in Chinese grammar error diagnosis (CGED) tasks. The results showed that in the CGED 2020 task, the proposed system achieved the highest F1 score in both error detection and recognition [12]. Tlonaen et al. found a considerable number of grammatical errors in students writing academic papers. The authors' team first analyzed in detail each of the main grammatical problems that exist in student-written material in this case, and used this as training data to design and train a model that can assist teachers in changing grammatical errors in students' academic papers. The experimental data denoted that the model's recall and accuracy rates of correcting grammatical errors on students' academic materials were 91.34% and 89.24%, respectively, which were higher than those of traditional intelligent grammatical error recognition methods [13]. Li et al. found that adding a certain attention mechanism to the GEC model could optimize the algorithm's operational process, and the designed model had significantly higher recognition accuracy in English natural text than traditional models [14]. Wang et al. attempted to use bidirectional long short-term memory for GEC, but they did not perform more refined processing on the dataset and established a model for grammar recognition in Chinese text, which was relatively ordinary [15]. The GEC model construction approach proposed in this study had certain novelty compared to other studies.

In summary, although a lot of previous studies have been conducted to improve the computational accuracy and performance of grammatical error detection and correction systems, most of them have failed to fully consider both the defects of the raw data itself and the advantages of deep learning tools. Therefore, this study attempts to design an RB and shallow neural network-based English grammar error generation (GEG) model based on the characteristics of English grammatical error texts, and use it as the core to design an improved English GEC model.

3. Design of English GEC system integrating grammar error generation model.

3.1. Design of English grammar error generation model based on data augmentation. The English text is a natural language information with time series and non-linear characteristics, so if it needs to detect and correct English grammatical errors, it will develop a model to describe English grammatical errors. Here, the way of data augmentation is chosen to cope with the lack of data in the training corpus, and then the idea of attention mechanism, neural network back propagation and cross entropy loss function are integrated to build an English GEG model. Due to its strong data processing compatibility, this model can handle various forms of English text, including business and communicative English.

Firstly, the performance evaluation index of English GEG model is determined. Most of the current evaluation methods need to compare the manually annotated target sentences with the output sequence and complete the word alignment operation of both. So, the performance is mostly measured by precision, max match score (MMS) and other indexes in the industry before [16]. MMS method is commonly used in English GEC problems. This method takes the corrected sentences output by the error generation model and the manually annotated sentences for word alignment operation. The number of word operations needed to convert the sentences is calculated, and the model is evaluated using the recall *Rec*, F0.5, and precision $\Pr e$ metrics. The three calculations are shown in equations (3.1), (3.2), and (3.3).

$$Rec = \left(\sum_{i=1}^{n} |e_i \cap g_i|\right) \Big/ \left(\sum_{i=1}^{n} |g_i|\right) \tag{3.1}$$

$$F0.5 = \left(P \times R\left(0.5^2+1\right)\right) / \left(R + \left(0.5^2 + P\right)\right) \tag{3.2}$$

$$\Pr e = \left(\sum_{i=1}^{n} |e_i \cap g_i|\right) \Big/ \left(\sum_{i=1}^{n} |e_i|\right) \tag{3.3}$$

In equations (3.1), (3.2) and (3.3), e_i and g_i represent the set of candidate edits output by the GEG model and the set consisting of standard corrective edits, respectively. Moreover, the following relationship needs to be satisfied between these two sets: $e_i \cap g_i = \{e \in e_i | \exists g \in g_i, e = g\}$. The generalized language evaluation understanding (GLEU) metric can be used to evaluate the fluency of an utterance, and its expression is given in equation (3.4).

$$GLEU(C,R,S) = BP \cdot \exp\left(\sum_{n=1}^{4} W_n \log p_n^{'}\right) \tag{3.4}$$

where C, R and S mean the corrected sentence, the standard sentence, and the initial input error sentence, respectively; W_n and P_n denote the weights distributed in a uniform manner and the weighting precision of the corrected sentence compared with the standard sentence respectively. TheBP in equation (3.4) is the mapping function between the target sentence and the output sentence, and the calculation method is shown in equation (3.5).

$$BP = \begin{cases} e^{\left(\frac{c-r}{c}\right)}, c \leq r \\ 1, c > r \end{cases} \tag{3.5}$$

In equation (3.5), r is the length of the target sentence and c indicates the length of the model output sentence. The indicators provide corresponding scores for both error detection and correction, which is different from the MMS method, in which each marker is classified as false positive (FP), false negative (FN), true positive (TP), and true positive (TN). Then, the recall Rec, F_β, and precision $\Pr e$ indicators can be calculated according to equations (3.6) to (3.8). References [17] and [18] also use the same evaluation index calculation method [15-16].

$$Rec = TP/(TP + FN) \tag{3.6}$$

$$F_\beta = \frac{P \cdot R\left(1 + \beta^2\right)}{(\beta^2 \cdot P) + R} \tag{3.7}$$

$$\Pr e = TP/(TP + FP) \tag{3.8}$$

The indicator I is calculated using the weighted precision W_{acc}, as denoted in equation (3.9).

$$W_{acc} = \frac{TN + \omega TP}{TN + FN + \omega\,(TP + FP) - (\omega + 1)\frac{FP}{2}} \tag{3.9}$$

In equation (3.9), ω is the penalty weight coefficient, and its default setting is 2.0. Therefore, the indicator I can be calculated according to equation (3.10).

$$I = \begin{cases} \lfloor W_{acc\_s} \rfloor, W_{acc\_s} = W_{acc\_b} \\ \frac{W_{acc\_s} - W_{acc\_b}}{1 - W_{acc\_b}}, W_{acc\_s} > W_{acc\_b} \\ \frac{W_{acc\_s}}{W_{acc\_b}} - 1, W_{acc\_s} < W_{acc\_b} \end{cases} \tag{3.10}$$

Now, the English GEG model is designed again, the current mainstream GEC system in the market framework structure is shown in Figure 3.1.

In Figure 3.1, the preprocessed English data is fed into different corpora for expansion, to increase the diversity and scale of the data. The expanded data is used to train various translation model-based error correction systems. After training, the model iteratively corrects and reorders the data to find the best error correction solution. The data are pre-processed before being fed into the grammar generation model, where they are mainly subjected to de-duplication, blank line removal, special symbol processing, length control, and word separation. Special symbols do not affect grammar checking and are directly removed, and length control refers to the splitting and truncation process for sentences that are too long. When GEC work is chosen to be carried out, the amount of manual annotated language work is huge, so the data available for training are

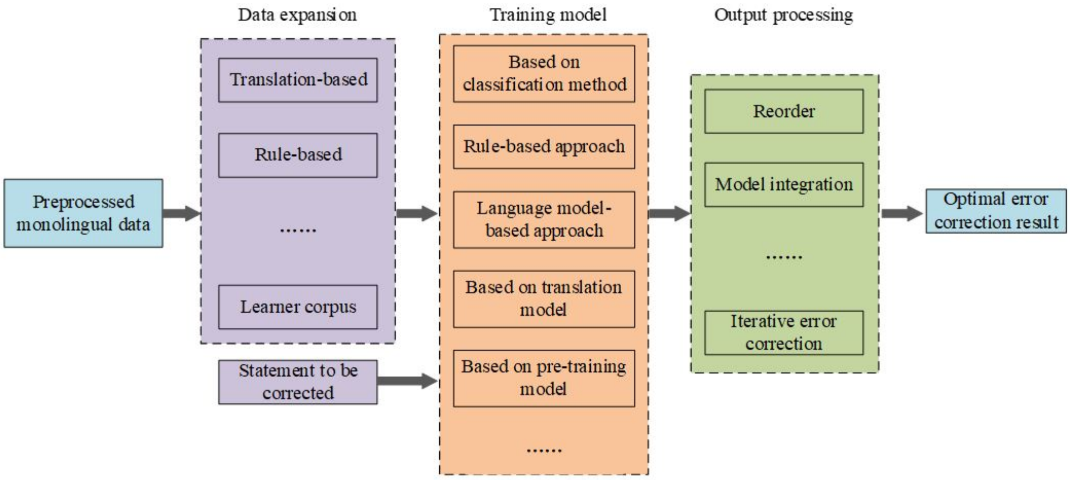

Fig. 3.1: Structure of common English GEC system

Table 3.1: Comparison of data augmentation methods

Method number	*Name of data enhancement method*	*Disadvantages*	*Advantages*
#01	RB data enrichment	Higher implementation costs and fewer rules that can be defined in advance	Low computational complexity and efficient
#02	Reverse translation-based data enrichment	Generate ungrammatical output messages	Wide range of applications
#03	Round-trip translation-based data augmentation	May produce semantic differences that differ significantly from the source sentence	Supports multi-language extensions and does not require large amounts of labeled data
#04	Data widening based on modification history	Generate data with more noise, need to filter	Output data is large in size and can be collected as error correction data
#05	Data augmentation based on fused tag types	Depends on the method of fusion	Depends on the method of fusion

often lacking to some extent. Therefore, before error generation on the data, data augmentation operations are required, and the common operations are shown in Table 3.1.

Considering the characteristics of the dataset selected for this study, an RB approach is chosen here to carry out data enrichment operations, i.e., delete, insert, swap, and replace operations are performed at the word level with introduction probabilities of 0.20, 0.30, 0.25, and 0.25, respectively [19-20]. Analysis of the learner corpus reveals that there are more grammatical errors in prepositions, spelling, punctuation, coronals, nouns, and verbs. Therefore, a derivation method based on substitution rules is proposed here to build confusion sets of these errors separately, count the words with word frequency no less than 4 in a single corpus, and compose the TOP7000 into a dictionary [21-22]. Then the above four operations will be performed randomly on the operation words during data synthesis, and the specific manual rules are defined as follows. It is supposed that the original sentence is$S = \left\{w^0, w^1, ..., w^{i-1}, w^i\right\}$, and the error statement obtained is T.

Insertion error: a rule is inserted to$S$, a token$w'$is added and $T = \left\{w^0, w^1, ..., w', ..., w^{i-1}, w^i\right\}$ is obtained. Deletion error: it deletes a random token to$w'$and S and $T = \left\{w^0, w^1, ..., w^{i-1}, w^i\right\}$ is got. Exchange error:

any two tokens are exchanged in S and $T = \left\{w^0, w^1, ..., w^{i-1}, ..., w', w^i\right\}$ is obtained. Replacement error: it randomly replaces the selected word w'. If it exists in the list of words to be replaced in the confusion set, it will randomly select an alternative word from the corresponding candidate set list to replace it, or vice versa, it will select any word from the dictionary to replace it and is got. In summary, it can get the calculation process of English GEG model based on rule and reverse translation data augmentation, which will be described in detail below. GEG_1 and GEG_2 represent the learner corpus training generation model and the error generation model based on the data augmentation method, respectively. The first step of the calculation is to train the GEG model using the corpus GEG_1, where the injected noise probability is $P(s|t)$, and the parameters of the model α_{back} and model loss $Loss_{back}$ can be obtained by applying the great likelihood estimation method. The three calculations are shown in equations (3.11) to (3.13), respectively.

$$P(s|t) = \prod_{T=1}^{N} P\left(s\_{T}|t, s_{1:T-1}; \alpha_{back}\right) \tag{3.11}$$

$$\alpha_{back} = \arg\max \sum \log P\left(s\_{T}|t, s_{1:T-1}; \alpha_{back}\right) \tag{3.12}$$

$$Loss_{back} = \sum - \log P\left(s\_{t}|t, s_{1:t-1}; \alpha_{back}\right) \tag{3.13}$$

where $s = (s_1, s_2, ..., s_n)$ denotes the grammatically correct input English text sequence from the source and $t = (t_1, t_2, ..., t_n)$ indicates the output English text sequence containing grammatical errors from the target. The second step is to synthesize the training data using the data augmentation method chosen above and generate the optimized GEG_2.

3.2. Design of English grammatical error detection and correction model based on grammar error generation model. Before correcting for grammatical errors, grammatical errors have to be detected first. Therefore, the following then designs a model for detecting and correcting English grammatical errors based on a GEG model. In Figure 3.2, the grammar detection model is built using the BERT structure for detecting and correcting English grammar errors. Firstly, the English text to be detected is preprocessed, and the text is transformed into a format that the model can understand through methods such as word segmentation or subword partitioning, known as tokenization. This step decomposes the text into a series of tokens, each representing a word, subword, or punctuation mark in the text. Then, it is to convert the tokenized input data into tensor form, which is a multidimensional array that can be efficiently stored in a computer and subjected to mathematical operations. The next is to feed the tensor form input data into the BERT model. In the model, each sequence's token is assigned to different layers of the Transformer for processing, capturing the contextual relationships between tokens through self attention mechanisms and generating a deep representation of each token. The BERT model will ultimately generate an embedded representation for the entire sentence. After obtaining the sentence embedding, the GED linear classifier is used to classify the sentence and determine if there are any grammatical errors. A linear classifier will output one or more probability values based on the features embedded in the sentence, indicating the likelihood that the sentence belongs to different categories.

Meanwhile, to further improve the detection performance of the model, a reordering method based on multiple features is used to select the result with the highest combined score as the output. The weights of the features are obtained using the minimum error rate training (MERT) algorithm. The MERT algorithm compares the number of errors in the source statement with the GEC output $E(t_1, t)$, where t_1 is the information output from the GEC system; t and f_s represent the standard statement and the input information to be corrected at s, respectively. The purpose of the MERT algorithm is to calculate the target sentence with the smallest number of errors among multiple candidate corrections, which is also the optimal parameter of the algorithm. The algorithm compares the standard sentence with the highest scoring corrected sentence when there is an error in the statement, and the calculation expression is as in equations (3.14) to (3.16), which is also consistent with the calculation method of evaluation indicators in reference [20].

$$\lambda_1^M = \arg\min \left\{ \sum_{s=1}^{S} E(r_s, e(f_s; \lambda_1^M)) \right\} \tag{3.14}$$

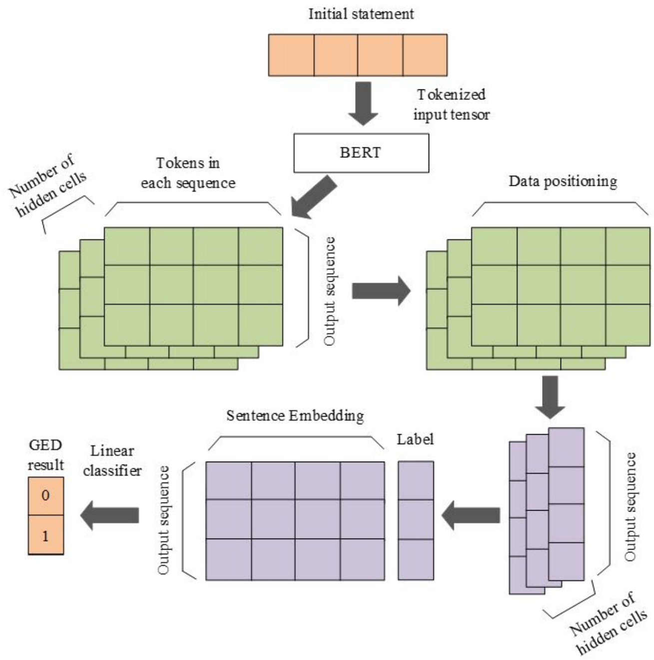

Fig. 3.2: English grammar error detection model

$$e(f_s;\lambda_1^M) = \arg\max\left\{\sum_{m=1}^{M}\lambda_m h_m(e|f_s)\right\} \tag{3.15}$$

$$score(T,S) = \sum_{m=1}^{|M|}\lambda_i f_i(T,S) \tag{3.16}$$

In equations (3.14) to (3.16), t_1, t, and f_s indicate the output information of the GEC model, the standard sentence, and the corresponding input of the s corrected sentence, respectively. λ_m and $|M|$ mean the feature weights and the number of features, respectively. In the following, the GEC model is designed. Grammatical errors generally refer to errors in sentences that do not conform to grammatical rules, and are divided into structural and non-structural errors. The former being the type of errors that can only be corrected by moving, deleting or inserting a number of words, and the latter being errors that can be corrected by replacing some words. Moreover, semantic errors are those that exist in the text, which basically do not belong to spelling or grammatical errors and are difficult to identify. The above-mentioned multiple error types often appear in English text at the same time. Therefore, a correction model is needed to be designed that mainly deals with grammatical errors, but can also incidentally deal with a part of semantic and spelling errors. In the correction model, it is first necessary to generate training data using a data augmentation strategy, which will be used to train the GEG model together with the learner corpus. Then the GEC model is trained using the learner corpus together with the synthesized training data, the workflow of which is shown in Figure 3.3. In Figure 3.3, the idea of alternate training is incorporated, and the GEC model is used to correct the source sentences. The corrected data and the reference sentences in the learner corpus together form a parallel corpus and are added to the GEG, and this process will be repeated until the error correction needs are satisfied.

Specifically, the specific steps of using the alternate training model board in Figure 3 are as follows. First, the GEC model with the strongest performance is selected to compute the source sentences in each learner's corpus to avoid the error correction due to the low performance of GEC. The standard reference sentences are

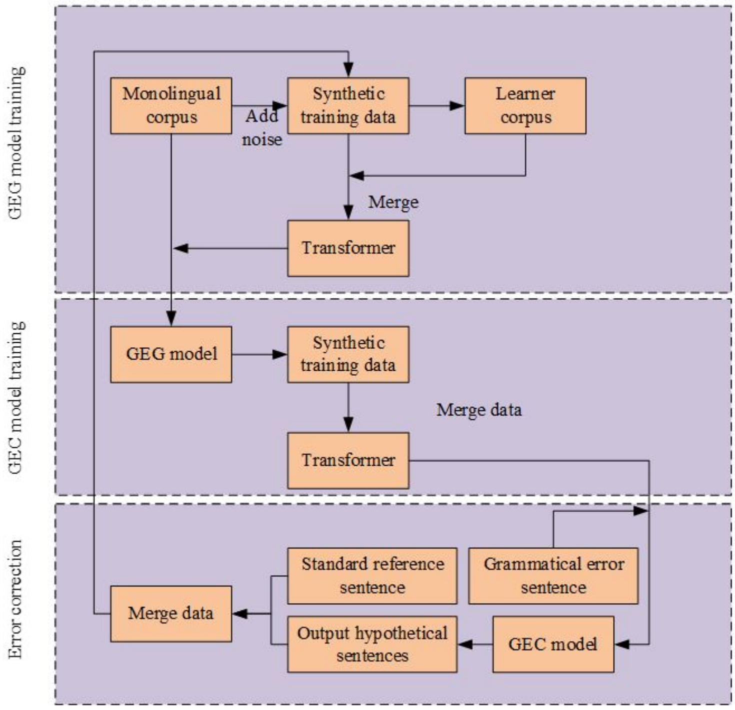

Fig. 3.3: Computational flow of error correction model incorporating the idea of alternate training

then combined with the candidate sentences output by the model to form the training set, and the data are mixed after the data augmentation operation and re-input into the corresponding mode. The next step is to fine-tune the selected learner corpus to obtain GEG3. The GEG3 is used to process the monolingual corpus and to generate synthetic datasets of different sizes. Finally, after the synthetic data is formed, the pre-trained data is further scaled up with the aim of improving the performance of these designed grammar generation models.

Finally, it analyzes the potential shortcomings of the model designed in this study. Due to the integration of various computational structures such as BERT, shallow neural networks, and attention mechanisms in the designed model, the overall structure of the model is relatively complex. There may be issues with slow computation speed and high computational complexity in the computer. Subsequent experiments will also address these issues.

In summary, in the method model designed in this study, it is first necessary to perform data augmentation processing on the input data to expand the data scale. Then, the amplified data will be input into the hybrid BERT algorithm, where the MERT method is used to obtain the weight coefficients of features, and the MERT processed data will be input into the attention network to adjust these weights based on recognition performance. Subsequently, the data will be inputted into a GEC model with alternating training characteristics for error correction. Here, the corrected and the reference sentences in the learner's corpus together form a parallel corpus and are added to the GEG. This process will be repeated until the error correction requirements are met, and the final error correction result will be output.

4. Performance testing of an automatic English GEC model that is based on a grammatical error generation model. To verify the performance of the English GEC system designed in this research, a validation test was now designed and conducted, in which a model was built using the open-source Sequence-to-Sequence toolkit. The operating environment and parameter settings of the model are shown in Table 4.1.

The study adopted the method of sampling decoding. The models were trained using FCE, W&I+LOCNESS,

Table 4.1: Operating environment and parameter settings of the model

Project	*Parameter*
System	Window 10
GPU	NVIDIA Tesla H800
CPU	AMD Ryzen 9 7950X3D
Memory	DDR5 6400 32GB(16GBx2
Development language	Python
Vector dimension	256
The number of layers in the network where the decoder and encoder are located	6th floor
Dimension of hidden layers in forward neural networks	2024
Dropout ratio	0.22
Adam type optimizer	\
Initial learning rate	0.0015
Label smoothness rate	0.25
Preheating step size	13000
Model training corpus	FCE, W&I+localization, NUCLU corpus, and Lang-8 corpus

Table 4.2: Specific information of training and testing corpus

Number	*Type*	*Name*	*Number of statements*	*Statement size*	*Marker size*
#001	Learner corpus	NUCLU	57426	58K	1.21M
#002		FCE	28668	29K	463K
#003		W&I+LOCNESS	34629	35K	637K
#004	Social media data	Lang-8	1048853	1.05M	11.83M
#005	General corpus	One Billion word	1702641	1.70M	19.05M

NUCLU corpus, and the Lang-8 corpus. Moreover, the GEC model was needed to be fine-tuned. Then, the completed models were tested using the general-purpose corpus One Billion word. The specific information of these models is shown in Table 4.3. In addition, the recall *Rec*, F0.5, precision $\Pr e$, and computation time consuming were chosen as as the evaluation index of the computation results.

Finally, to compare the performance of the model designed in this study with other models, RB, classification-based (CB), and RNN-based algorithm approaches were chosen to construct the comparison models.

After the experiments, the changes of the loss function of each error correction model during the training process were counted and shown in Figure 4.1. It needs to note that the horizontal axis in Figure 4 stands for the iteration number and the vertical axis stands for the value of the loss function, and different types of curves represent different GEC models. GEG, RB, CB, and RNN mean the models designed in this study. Because of the large range of order-of-magnitude variation of the loss function values during the training process, the vertical axis was used to display in a multi-segment manner. Observing Figure 4.1, with the growth of the number of iterations, the overall trend of the training loss function value of each model first decreased and then stabilized. When the iteration times was small, the rate of decline was generally fast, but the rate of decline was also rapidly decreasing. When the iteration times exceeded a certain value, the loss function completed convergence. After the number of iterations reached 200, all models were trained, and the loss function values of GEG, RB, CB, and RNN models were 0.04, 0.12, 0.11, and 0.07 respectively at this time.

The statistical outcomes are presented in Figure 4.2. The meaning of the horizontal axis in Figure 4.2 is the same as that in Figure 4.1, and the vertical axis indicates the correction accuracy of the correction models on the test set after training to the corresponding degree in %. From Figure 4.2, the correction accuracy of each model had an opposite trend with the increase of iteration times, which first increased rapidly and then

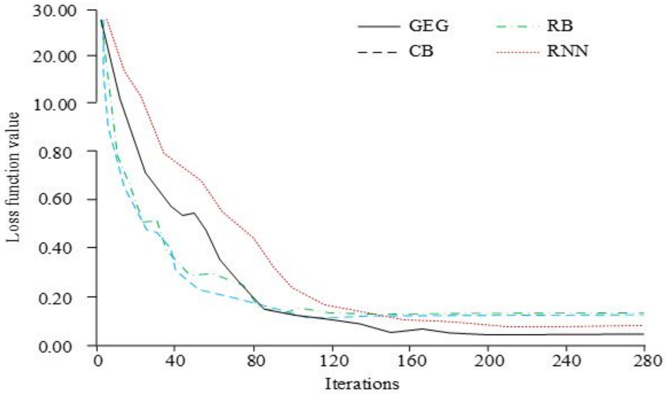

Fig. 4.1: Loss function change curve of training process

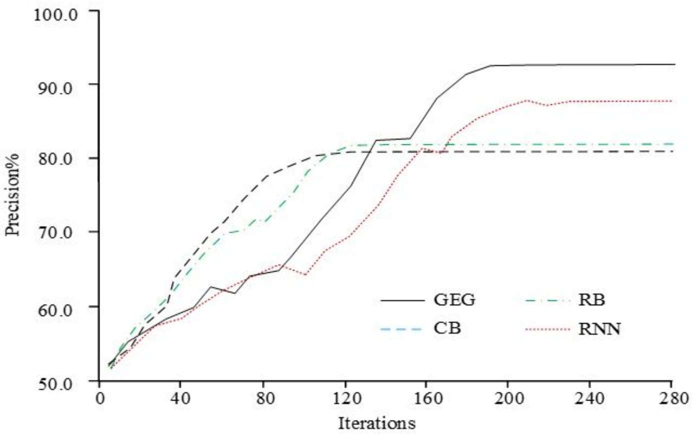

Fig. 4.2: Precision rate change curve during training

gradually converged. At the same time, there was a small amount of repeated fluctuation in the rising value, which was brought by the randomness of the selection of the training batch data. When the iteration times reached 200, each correction model also completed convergence, at which time the test set grammar correction precision rates of GEG, RB, CB, and RNN models were 93.6%, 82.4%, 80.7%, and 88.2%, respectively.

The statistics of grammar correction precision and recall for different sample numbers in the test set after the training of each correction model are shown in Figure 4.3. The horizontal axis in Figure 4.3 indicates the number of samples used to test the models in the test set, and the vertical axis means the correction precision and correction recall of each model under the selected test set scenarios in %, and different lines represent different correction models. Observing Figure 6, when the number of test samples was small, the fluctuation of both the precision rate and the recall rate of each model was larger. As the number of test samples grew, the fluctuation became less and less floating, and the trend of these two metrics was generally consistent. Moreover, the precision and recall rates of the models designed in this study were significantly higher than all the comparison models, but the recall rate values were less different from the second-ranked corrected model. Specifically, when the number of test samples reached the maximum, the fluctuation in precision and recall of each correction model was the smallest, and precision and recall of GEG, RB, CB, and RNN models were 93.92%, 82.17%, 79.41%, 88.09% and 95.81%, 86.92%, 83.46%, 94.37%, respectively.

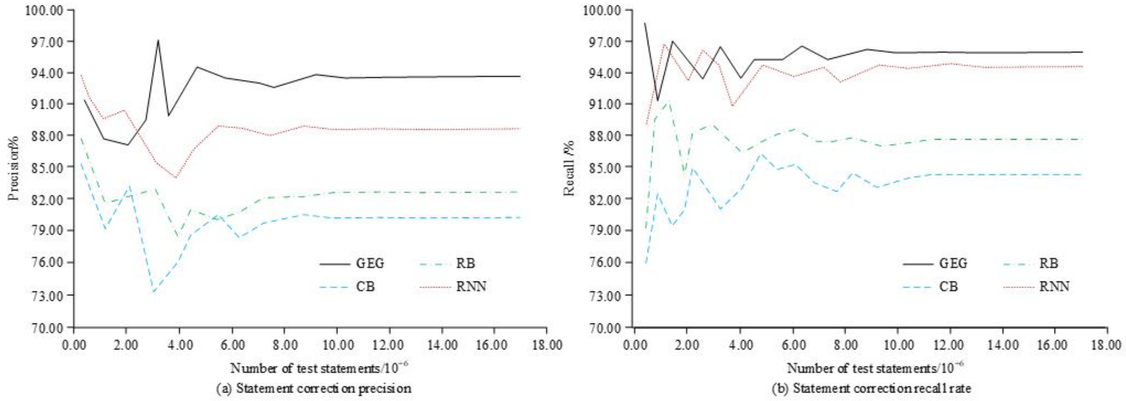

Fig. 4.3: Precision and recall variation curves of each model on the test set after the training

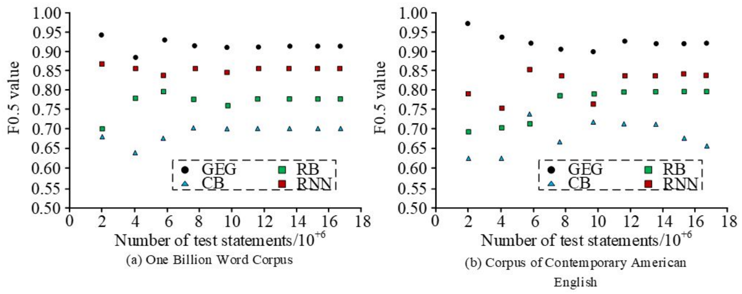

Fig. 4.4: F0.5 values of each model on the test set after training

To further improve the reliability of the statistical results, normalized F0.5 values were used to compare the performance of each model and the statistical results. At the same time, to demonstrate the adaptability of each model in the validation model, Corpus of Contemporary American English was used to test the models. The meaning of the horizontal axis in Figure 4.4 is consistent with that in Figure 6. The vertical axis represents the normalized F0.5 value for each experimental scenario on the test set. Observing Figure 4.4(a), there was no significant correlation between the number of test samples and F0.5 of the calibration model. Specifically, when there were few test samples, the F0.5 value of each calibration model fluctuated greatly, which was related to the calculation method of F0.5 value. However, as the number of test samples increased, the fluctuation of this indicator significantly decreased. When the test sample reached its maximum value, the F0.5 values of GEG, RB, CB, and RNN models were 0.92, 0.78, 0.70, and 0.86, respectively. Observing Figure 4.4 (b), the F0.5 value of the GEG model constructed in the study was not significantly different from that in Figure 4.4(a), while the RB, CB, and RNN models showed significant fluctuations and differences, indicating that the GEG model constructed in the study had superior adaptability.

Finally, the computational efficiency of each model was analyzed, and the computational elapsed time of each correction model in different test sample size scenarios was counted as an indicator. The statistical results are shown in Table 4.3. It noted that to improve the precision of the computational results, each experimental scenario was repeated five times, and the results were presented in the form of mean ± standard deviation of elapsed time. Observing Table 3, the computational efficiency of the error correction model designed in this study was lower than that of the RB and CB error correction models, but significantly higher than that of the RNN model. The RNN error correction model was built based on a neural network algorithm, so the computational speed was slower. The computational standard deviation of the RNN error correction model

Table 4.3: Comparison of computational efficiency of the models (unit: s)

Number of test statements	*GEG*	*RB*	*CB*	*RNN*
100	0.24±0.08	0.18±0.07	0.15±0.04	0.86±0.14
1000	2.03±0.17	1.65±0.14	1.35±0.11	7.51±0.92
10000	22.54±1.75	17.88±1.92	15.26±1.36	80.71±11.59
100000	196.35±15.42	169.29±12.58	142.22±12.67	765.90±93.65
1,000,000	852.56±62.78	637.25±48.39	577.51±37.02	3879.39±431.82
1702641	1433.52±87.29	1174.61±94.35	1027.95±89.15	1027.95±89.15

accounted for the largest proportion of the computational elapsed time when the number of test utterances was the same, indicating that the computational elapsed time stability of this model was also the worst. Specifically, the F0.5 values of the GEG, RB, CB, and RNN models were 1433.52 ± 87.29s, 1174.61 ± 94.35s, 1027.95 ± 89.15s, and 1027.95 ± 89.15s, respectively, when the number of test utterances reached the maximum.

5. Discussions. In today's globalized world, the importance of English is increasingly prominent. However, many non-native English learners often encounter grammar errors during the learning. Traditional English teaching methods often find it difficult to detect and correct students' grammar errors in a timely and accurate manner, which has an undeniable impact on learners' English learning. Therefore, how to improve the efficiency and effectiveness of English grammar teaching through technological means has become an urgent problem to be solved. Based on this, an English grammar automatic correction robot based on the GEG model was proposed.

The experimental results denoted that the designed GEG model had no significant advantage in training speed, but the loss function after training was significantly lower than the other comparison models, and the precision was significantly higher than the comparison models. Moreover, the precision, recall, and F0.5 values of the model on the test set were also higher due to the comparison model. This was mainly because attention neural networks could compensate for the shortcomings of the BERT model in treating input data equally, and the BERT model itself had great compatibility with text content with grammar errors. Generally, there is no need to adjust the model structure according to the characteristics of the processed data. Although the RNN series algorithms also had significant data compatibility, there was a risk of judgment and recognition due to the disappearance of gradients. The main drawback of the CB and RB models was that the core basis for GEC was too single. For example, for classification error correction models, the precision of error correction for data with multiple types of grammar errors would be greatly reduced. Therefore, the GEC model designed in this study performed better than common error correction models. However, the model designed in this study also has a drawback, which is that the fusion algorithm has a large structure, multiple internal calculation steps, and a large amount of computation, resulting in a longer processing time for samples of the same size. Subsequent research should focus on lightweight adjustments to the model while ensuring algorithm error correction and recognition accuracy.

6. Conclusion. To improve the quality of English GEC, an English GEC model based on GEG and neural network was designed in this study. An experiment was conducted to verify the error correction performance of the four error correction models, including the designed model. The experimental results indicated that in the model training phase, the convergence speed of the designed GEC model was among RNN, RB, and CB models, but the loss function after convergence was 0.04, which was significantly lower than all the comparison error correction models. The precision rate after convergence was 93.6%, which was higher than all the comparison models. The precision and recall rates of GEG, RB, CB, and RNN models on the complete test set after training completion were 93.92%, 82.17%, 79.41%, 88.09% and 95.81%, 86.92%, 83.46%, and 94.37%, respectively.The computational elapsed time of these four models on the complete test data set was 1433.52 ± 87.29s. The error correction quality of the English GEC model designed in this study was better than the traditional error correction model, but the former did not improve significantly in terms of computational efficiency. The designed error correction model had a certain potential for application in intelligent translation and English text proofreading. However, due to experimental limitations, this study was unable to develop a nearly commercial level English GEC system based on the designed model and verify its practical effectiveness. Moreover, the calculation time of the model designed in this study was also relatively long, and making fine-tune to the model is also one of the future research directions.

REFERENCES

[1] Rahmanu I, Winarta I, Ni P. An empirical study on grammatical error uttered by non-native English students. *Journal of Applied Studies in Language*, 2020, 4(2): 235-246.
[2] Nanning N, Saepuddin, Munawir. An Analysis of Grammatical Error of English Students in Writing Skill. *EDUVELOP (Journal of English Education and Development)*, 2020, 3(2): 145-160.
[3] Solyman A, Wang Z, Tao Q, Elhag A A M, Zhang R, Mahmoud Z. Automatic Arabic Grammatical Error Correction based on Expectation-Maximization routing and target-bidirectional agreement. *Knowledge-Based Systems*, 2022, 241(4): 108180.1-108180.13.
[4] Paik K, Pajk D. Multilingual fine-tuning for Grammatical Error Correction. *Expert Systems with Applications*, 2022, 200(8): 116948.1-116948.9.
[5] Wang Y, Wang Y, Dang K, Liu J, Liu Z. A Comprehensive Survey of Grammatical Error Correction. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2021, 12(5): 65.1-65.51.
[6] Choi S P, Lee M, Shin H, Lee D. Korean Grammatical Error Correction Based on Transformer with Copying Mechanisms and Grammatical Noise Implantation Methods. *Sensors*, 2021, 21(8): 2658-2659.
[7] Witteloostuijn M V, Boersma P, Wijnen F, Rispens J. Grammatical performance in children with dyslexia: the contributions of individual differences in phonological memory and statistical learning. *Applied Psycholinguistics*, 2021, 42(3): 14-31.
[8] Linarsih A, Irwan D, Putra M. The Interferences of Indonesian Grammatical Aspects into English: An Evaluation on Preservice English Teachers' EFL Learning. *IJELTAL (Indonesian Journal of English Language Teaching and Applied Linguistics)*, 2020, 5(1): 69-81.
[9] Jiang F, Chiba Y, Nose T, Ito A. Language modeling in speech recognition for grammatical error detection based on neural machine translation. *Acoustical Science and Technology*, 2020, 41(5): 788-791.
[10] Putra F P, Enda D. Model design for grammatical error identification in software requirements specification using statistics and rule-based techniques. *Journal of Physics: Conference Series*, 2020, 1450(1): 012071-012080.
[11] Koyama S, Takamura H, Okazaki N. Analysis of Data Augmentation for Grammatical Error Correction Based on Various Rules. *Journal of Natural Language Processing*, 2022, 29(2): 542-586.
[12] Zhang J. Combining GCN and Transformer for Chinese Grammatical Error Detection. *Journal of Internet Technology*, 2022, 23(7): 1663-1668.
[13] Tlonaen Z A. Grammatical Error Found in the Academic Essays Written by Students of English Education. *Lectura Jurnal Pendidikan*, 2020, 11(1): 15-30.
[14] Li Z, Parnow K, Zhao H. Incorporating rich syntax information in Grammatical Error Correction. *Information Processing & Management*, 2022, 59(3): 102891-102910.
[15] Wang H, Zhang Y J, Sun X M. Chinese grammatical error diagnosis based on sequence tagging methods. *Journal of Physics: Conference Series*, 2021, 1948(1): 012027-012033.
[16] Wang B, Hirota K, Liu C, Li Y, Zhi D, Jia Y. An Approach to NMT Re-Ranking Using Sequence-Labeling for Grammatical Error Correction. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 2020, 24(4): 557-567.
[17] Wang N, Issa R, Anumba C J. NLP-Based Query-Answering System for Information Extraction from Building Information Models. *Journal of Computing in Civil Engineering*, 2022, 36(3): 4022004.1-4022004.11.
[18] Lee J W, Choi N, Kim D, Tan N L P, Kim B J. Side Chain Engineered Naphthalene Diimide-Based Terpolymer for Efficient and Mechanically Robust All-Polymer Solar Cells. *Chemistry of Materials*, 2021, 33(3): 1070-1081.
[19] Qi L, Wang Y, Chen J, Liao M, Zhang J. Culture under Complex Perspective: A Classification for Traditional Chinese Cultural Elements Based on NLP and Complex Networks. *Complexity*, 2021, 2021(11): 6693753.1-6693753.15.
[20] Amato F, Coppolino L, Cozzolino G, Mazzeo G, Nardone O. Enhancing random forest classification with NLP in DAMEH: A system for DAta Management in eHealth Domain. *Neurocomputing*, 2021, 444(1): 79-91.
[21] Suleman R M, Korkontzelos I. Extending latent semantic analysis to manage its syntactic blindness. *Expert Systems with Applications*, 2020, 165(3): 114130.1-114130.9.
[22] Wach A, Zhang D, Nichols-Besel K. Grammar instruction through multinational telecollaboration for pre-service teachers. *ReCALL*, 2021, 34(1): 4-20.

Edited by: Achyut Shankar
Special issue on: Machine Learning for Smart Systems: Smart Building, Smart Campus, and Smart City
Received: Mar 24, 2023
Accepted: Aug 31, 2024