

Research Article

Machine Learning-Based Grammar Error Detection Method in English Composition

Jianbin Zhu,^{1,2} Xiaojun Shi ,¹ and Shuanghua Zhang²

¹School of International Studies, University of International Business and Economics, Beijing 100105, China

²College of Foreign Studies, Guilin University of Technology, Guilin 541006, China

Correspondence should be addressed to Xiaojun Shi; 2017135@glut.edu.cn

Received 5 November 2021; Accepted 2 December 2021; Published 18 December 2021

Academic Editor: Rahman Ali

Copyright © 2021 Jianbin Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The detection of grammatical errors in English composition is an important task in the field of NLP. The main purpose of this task is to check out grammatical errors in English sentences and correct them. Grammatical error detection and correction are important applications in the automatic proofreading of English texts and in the field of English learning aids. With the increasing influence of English on a global scale, a huge breakthrough has been made in the task of detecting English grammatical errors. Based on machine learning, this paper designs a new method for detecting grammatical errors in English composition. First, this paper implements a grammatical error detection model based on Seq2Seq. Second, this paper implements a grammatical error detection and correction scheme based on the Transformer model. The Transformer model performs better than most grammar models. Third, this paper realizes the application of the BERT model in grammar error detection and error correction tasks, and the generalization ability of the model has been significantly enhanced. This solves the problem that the forward and backward cannot be merged when the Transformer trains the language model. Fourth, this paper proposes a method of grammatical error detection and correction in English composition based on a hybrid model. According to specific application scenarios, the corresponding neural network model is used for grammatical error correction. Combine the Seq2Seq structure to encode the input sequence and automate feature engineering. Through the combination of traditional model and deep model, the advantages are complemented to realize grammatical error detection and automatic correction.

1. Introduction

English is the most widely used language in the world. Most people use English to communicate, which makes English an indispensable part of people's lives. The Chinese learn English as an effective way to adapt to the globalization of information and the internationalization of the market, which makes the importance of English learning more prominent. English is a second language for Chinese learners. It is not easy to master English well. They need a lot of practice, and they can make timely discovery of grammatical errors and know how to correct them, so that they can better improve their writing skills [1–5].

Most researchers at home and abroad believe that English writing is an important part of learning English and is the most effective way to evaluate the English

proficiency of English learners. However, checking for grammatical errors in English composition is undoubtedly a very time-consuming and laborious task. According to statistics, the teacher-student ratio in my country is now far below 1%. Therefore, there is an urgent need for teaching software that can automatically check composition to relieve the teaching pressure of teachers. Therefore, we can use natural language processing technology to automatically check for grammatical errors in the writing process of English learners and give the corresponding revision opinions, as well as tips on grammar knowledge points. It can not only reduce the work pressure of teachers but also allow students to get feedback on their writing as soon as possible. Students can fully understand the grammatical errors in the composition and corrective opinions. And you can improve your

knowledge deficiencies in a targeted way through grammar knowledge points, which will also be of great help to improving students' writing skills [6–10].

At present, there have been some good achievements in the research of grammar checking and correction technology for English composition, and some mature systems have been known and used by the majority of people. However, most of these systems are constructed and developed through rule-based analysis methods. Its applicable population is mainly for learners whose mother tongue is English, and it is not suitable for English learners whose mother tongue is Chinese. Chinese English has its own unique characteristics, and some systems developed abroad are not suitable for Chinese students. For a good English grammar checker, the context information and lexical and semantic information of the text should be fully considered for grammar checking. And its own robustness should be good enough. All these make some current grammar checkers need further improvement and perfection. This topic is mainly based on machine learning strategies to explore a grammatical error detection and correction model suitable for English learners and check and correct the grammatical errors that English learners often make in English writing. Combining natural language processing-related technologies to realize automatic checking and correction of English texts is very beneficial for improving students' English proficiency. At the same time, it greatly reduces the burden of the teacher to check the students' composition, so that the teacher has more time and energy to pay attention to the structure and semantic expression of the composition. At the same time, grammar checking and error correction, as an important indicator of automatic composition scoring, can improve the objective accuracy of automatic scoring of English composition [11–15].

2. Related Work

Regarding the related research on the methods of grammatical error correction in English composition, it can be concluded that there are two main types: rule-based grammatical error correction method and statistics-based grammatical error correction method. The former relies on a set of hand-written grammar rules. The latter depends on the establishment of statistical models, such as n -gram, to correct grammatical errors in English composition.

Rule-based methods have been used in early grammar correction tools [16–18]. Since articles are not used in Japanese and there are no singular and plural nouns, these are very common in English expressions. Therefore, this issue must be considered when translating from Japanese to English. In order to solve this problem, literature [19] establishes rules based on the context to estimate the singular and plural nouns that should be added to the translated sentence. Finally, the test found that the correct rate of this method can reach 89%. The rule-based approach has many advantages: grammar rules can be easily added, modified, or deleted. Each rule can add a corresponding grammatical

explanation, which can give users more specific and targeted feedback information. The system is easy to debug because the corresponding rules can be found according to the prompt information of each rule. The rules in the rule base are easier to write, so they can be written by linguists with limited or no programming skills. However, as pointed out in literature [20], there are often exceptions to the method of rules, and it is difficult to use corpus statistics in hand-written rules. There is another statistical-based grammatical error correction method.

The method based on statistics mainly treats grammar checking as a classification task, in which the checking of article and checking of preposition errors are the two main topics [21–23]. Features are usually used for classification, and a variety of vocabulary and part-of-speech features are used in literature [24]. This includes neighboring words, part-of-speech marks, and language model scores. Analytical features are also added in literature [25], and the results show that, in the preposition error correction, the accuracy rate and recall rate have been appropriately improved. Among them, the classification algorithm uses the maximum entropy algorithm, the voting perceptron algorithm [26], and the naive Bayes algorithm [27].

In addition to the error checking of prepositions and articles, verb form error checking has also attracted some attention [28, 29]. The advantage of classification algorithms is that they have a strong ability to correct certain types of errors. Recently, people have done some work to correct different types of errors in a comprehensive manner. Literature [30] uses a high-order sequential labeling model to detect various errors. Literature [31] uses a method based on rules and syntactic n -grams. Syntactic n -grams in this document are different from traditional n -grams in that they contain syntactic information.

In the past grammatical error correction, most of them focused on the use of articles and prepositions because these are common mistakes made by some nonnative English learners. At the same time, as an important grammatical item in English writing expression, clauses are also a difficult point in grammar learning. Based on the study of the Chinese learner corpus, among the various types of clause grammatical errors, the frequency of related word errors is the highest, and it is difficult to correct them. However, there are relatively few researches on automatic error correction of English clause-related words. Literature [32] uses a hybrid model based on rule and statistical machine translation. Then use the language model to filter to correct all errors of CoNLL-2014, including clause errors. In the field of machine translation, automatic grammar correction is also involved. The output produced by the machine translation system is often grammatically incorrect, so the establishment of an automatic editing system is needed to solve this problem, for example, because Japanese does not contain articles. When translating Japanese to English, the output sentence needs to be corrected to choose the correct article. Literature [33] solves the problem of article selection in machine translation, and these types of systems can also be used for grammatical correction.

3. Proposed Methodology

With the rapid rise of machine learning, the methodology of deep models has achieved impressive results in many fields such as image processing and speech recognition. With the landing of many mature technology products, people consider trying to bring deep learning technology into the field of English composition grammar error detection and correction. Practice has proved that the position of deep learning technology in the field of English grammar error correction has become very important. Its main advantage lies in the elimination of the complicated manual feature extraction in traditional machine learning, and the automatic feature extraction of corpus makes deep learning widely praised. And many shortcomings of the shallow model are solved by the deep model.

3.1. Research on Deep Neural Network Model

3.1.1. RNN. Recurrent Neural Network (RNN) is a recurrent neural network that expands and analyzes based on the time dimension. Usually, a text sequence is a piece of data based on time series, and the cyclic neural network model is mainly constructed for processing time series data. Therefore, RNN is better at processing text data and is used to store historical information by setting state variables. At this time, the historical information and the input data in the current state together determine the output result at the moment. The use of RNN structure and other types of deep learning models is still used to simulate language models. Therefore, the language model is the foundation of the development of natural language processing technology. The n -gram discussed before, based on the current word, can only consider the historical information of the previous limited words, and the lack of historical information cannot be avoided. If n is increased, the number of parameters of n -gram is invisibly increased, and its complexity increases exponentially. Therefore, RNN abandons this rigid memory method but adds a hidden state method to store historical information.

For the multilayer perceptron model, it only considers the mapping of input information to output information but ignores historical information. Historical information plays a vital role in the output results, which is precisely the area where RNN is good at, that is, to model the text sequence containing historical word messages. The structure of RNN is illustrated in Figure 1.

If the input sequence at the input is represented as X_t , H_t is the hidden state at time t , and RNN represents the historical output at the previous time as H_{t-1} . The calculation paradigm is shown in the following formula:

$$\begin{aligned} H_t &= f(X_t W_t + X_{t-1} W_{t-1} + b_1), \\ O_t &= H_t W + b, \end{aligned} \quad (1)$$

where W_{t-1} , W_t , and W are the weights and b_1 and b are the bias.

The hidden state can well record the historical record messages of the sequence data entered up to the present, and these calculations are calculated in a loop. In this way, RNN

has memory, and the structure of RNN is rich and diverse and can be constructed on demand.

3.1.2. LSTM. In order to solve the gradient problem of RNN, many improved algorithms have appeared. Here is a detailed introduction to the LSTM algorithm. Literature [34] proposed a calculation model of LSTM. The LSTM model only adds three thresholds on the basis of RNN. It is essentially a recurrent neural network. Compared with RNN, the gating management mechanism of LSTM takes into account the context and the order characteristics of sentences and other pieces of important information to a greater extent.

These three thresholds are called forget gates, input gates, and output gates. In addition, it also includes the same memory cells as the hidden state, which can also be called a special hidden state. This series of operations can filter historical information and determine whether information can be transmitted and retained through threshold control, screen out useful information, and discard unnecessary information. The LSTM model can not only memorize contextual information over a long period of time. At the same time, it also relieves to a certain extent the pain points that the gradient tends to zero and tends to infinity due to the chain derivation rule when the RNN is located in the hidden layer to calculate the gradient. A longer memory can be obtained. The structure of LSTM is illustrated in Figure 2.

The parameter update for three gates is illustrated in the following formulas:

$$\begin{aligned} F_t &= \sigma(W_f[H_{t-1}; X_t] + b_f), \\ I_t &= \sigma(W_i[H_{t-1}; X_t] + b_i), \\ O_t &= \sigma(W_o[H_{t-1}; X_t] + b_o). \end{aligned} \quad (2)$$

The update method for the hidden state is

$$H_t = O_t * \tanh(C_t). \quad (3)$$

The update for the entire node is

$$C_t = F_t * C_{t-1} + I_t * \tanh(W_s[H_{t-1}; X_t] + b_s). \quad (4)$$

3.1.3. ELMO. ELMO is a stacked deep neural network based on the Bi-LSTM model [60]. It superimposes the LSTM in the vertical direction. When using ELMO to train a language model, it must be one-way, and it is impossible to train a language model from a two-way perspective. However, in order to achieve a two-way effect, it is necessary to train two LSTM models from two directions. At this time, the word vector representation is actually the forward and backward embedding for splicing. The main contribution of ELMO is to solve the shortcomings of shallow neural networks. ELMO extends a layer of neural networks to a deep stack structure. This can learn hierarchical language features, as shown in Figure 3 for the ELMO model. This learns from the word feature at the lowest level, syntactic feature related to grammar, and finally semantic feature related to semantics at the highest level. The upper-level features can be understood as context-based features. In different contexts, through

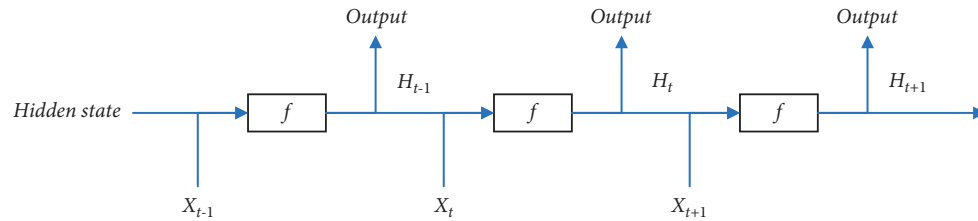


FIGURE 1: The structure of RNN.

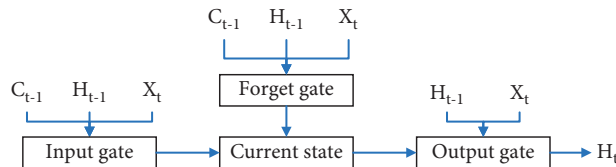


FIGURE 2: The structure of LSTM.

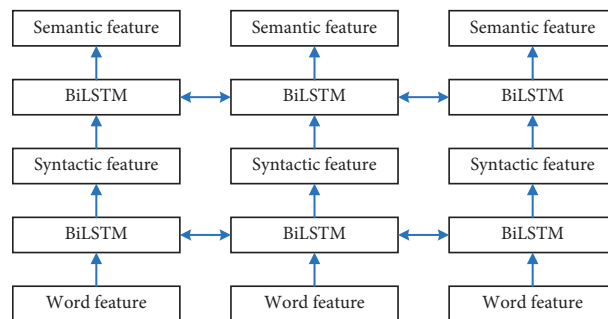


FIGURE 3: The structure of ELMO.

learning the word vectors of words, they represent the characteristics of these words in different contexts.

Although ELMO is very powerful, it still exposes the following shortcomings: (1) When the time series model processes a text sequence, it must be calculated word by word. This makes any time series model unable to process data in parallel. This wastes a lot of GPU resources, and the inability to parallelize computing is its pain point. (2) The problem of gradient disappearance: although there has been progress from RNN to LSTM, the problem of gradient disappearance has been alleviated to a certain extent by setting the threshold. However, LSTM still cannot fundamentally solve the problem of gradient disappearance, which causes the loss of important historical information at a long distance. (3) The training of the ELMO model is separately processed forward and backward. It is a pseudo-two-way neural network model without fusing it into one. This is the problem that the forward, backward, and backward cannot be merged.

3.2. Error Detection and Correction Model Based on Seq2Seq. What the RNN family realizes is that the input end is a fixed-length sequence, and the output end is a fixed-length sequence or a single tag value. In many practical applications of NLP, the variable-length expression of the input string sequence is realized to make the output requirements more

general. Both the input and output string sequences need to realize variable length characterization. In this way, more practical problems can be solved. In the field of machine translation, in English-Chinese translation, the input and output ends are often strings of variable length. The model used here is a typical sequence-to-sequence model (Seq2-Seq). Its name, encoder-decoder, can better reflect the essence of the model. Similarly, through the idea of analogy, the text sequence to be corrected is input as the variable length, and the corrected text sequence is returned as the variable length of the output end. Therefore, we can consider applying this model to the task of grammatical error detection and correction. This regards the error correction task as a machine translation type task. Practice shows that the model has an excellent error correction effect after integrating the attention mechanism. The proposed Seq2Seq model with attention for English error detection and correction (SSA-ERDC) is illustrated in Figure 4.

The design of the Seq2Seq model has a high degree of freedom and is convenient and flexible, which is a major breakthrough. The design of the encoder and decoder determines the realization of the core functions of the model. The encoder is used to process and respond to the input text sequence, and the decoder is used to express the output sequence. The construction of the more classic Seq2Seq model is essentially two RNN architectures or LSTMs of the input string sequence and the output string sequence

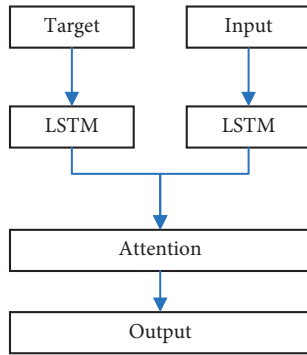


FIGURE 4: The structure of SSA-ERDC.

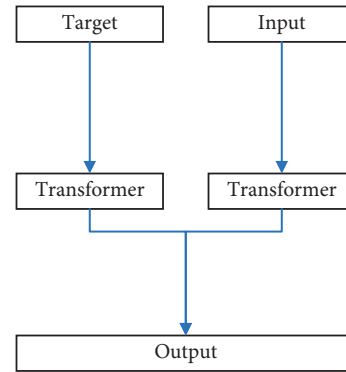


FIGURE 5: The structure of T-ERDC.

corresponding to the encoder and decoder. At present, the more mature designs of encoders and decoders are based on CNN and based on LSTM. The neural network architecture that matches the task can be flexibly designed according to the needs of the business scenario and the characteristics of the model itself. Seq2Seq also has its shortcomings. It usually uses fixed-length vectors when designing encoders and decoders. As a result, the model’s ability to capture long-distance dependencies is significantly weakened, and it is unable to fully characterize the information of longer Chinese texts. Therefore, there will be deviations in the understanding of the semantic information of the longer Chinese text.

3.3. Error Detection and Correction Model Based on Transformer. This paper builds a Transformer-based grammatical error detection and correction model, which can well solve the shortcomings of the traditional Seq2Seq model. This model discards the structure of the time series model and can capture long-distance semantic information and process data in parallel. At the same time, it can simulate the function of a time series model, which completely breaks the barriers that classic time series models such as RNN cannot achieve parallel computing. The self-attention mechanism is good at capturing the related information between words and words at a longer distance, getting rid of the constraint of text distance. This really solves the long-distance problem of missing historical information and has an excellent performance in grammatical error correction. The shadow of attention can be found in today’s deep learning models. In the early years, the composition of the traditional Seq2Seq model must be integrated into LSTM or CNN. However, the Transformer model that incorporates the attention mechanism breaks this backward construction model, and it shines in the field of neural network machine translation. The proposed Transformer model for English error detection and correction (T-ERDC) is illustrated in Figure 5.

As shown in Figure 6, Transformer is composed of three major packages. Divided into two from the middle, the information transmission between encoder component on the left and decoder component on the right is realized through the connection layer.

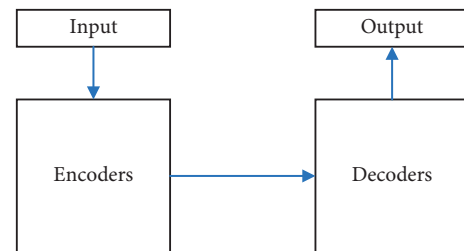


FIGURE 6: The mechanism of Transformer.

The encoder component is responsible for converting the input to the encoding vector. Figure 7 is a schematic diagram of the Transformer’s hierarchical structure. Encoders are composed of 5 encoder units of head end and tail end based on stack structure connection. Decoders are also constructed by stacking the decoder unit stack, accepting the vector from encoders, and combining the vector to make predictions to obtain output.

Transformer is different from the traditional Seq2Seq model based on LSTM components. Transformer composition is divided into two parts from the middle because there are many hierarchical structures superimposed. Therefore, the words or sentence features in each encoder layer learned by the Transformer model may have different semantics. That is to learn the characteristics of a hierarchical structure. For example, the encoder layer at the top may represent the most intuitive meaning of a sentence, and other layers may learn the deeper language features of a sentence. In this way, these outputs are passed to the decoder layer as input through the connection layer, and the output results are predicted with the help of these features.

3.4. Error Detection and Correction Model Based on BERT. When Transformer trains the language model, the problem that the forward and backward cannot be merged still exists. The Masked Language Model used in the BERT model can solve this problem well, so consider building a BERT model for English grammatical error detection and correction (B-ERDC). BERT is based on the previously used word2vec model and ELMO model. This makes the generalization ability of the word vector model significantly enhanced, enabling it to accurately describe

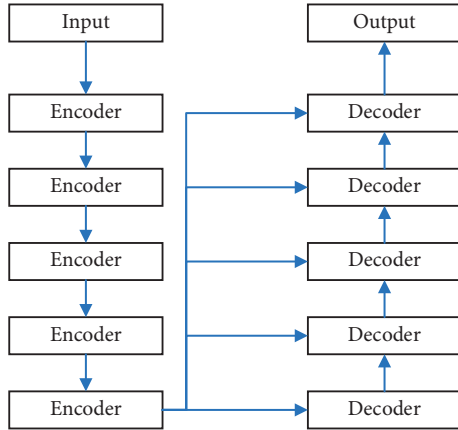


FIGURE 7: Transformer hierarchy diagram.

the relationship features at the character level and sentence level. The structure of B-ERDC is illustrated in Figure 8.

BERT uses Masked LM in the process of training the language model, that is, in the process of inputting a sentence. Randomly select some words as the words to be predicted, and replace them with special symbols. During training, it can avoid the problem that CNN is not suitable for serializing text and RNN has no way to achieve parallel computing.

3.5. Error Detection and Correction Based on Hybrid Model. The grammatical error correction scheme based on n -gram and CRF is effective in correcting surface grammatical errors. However, in the face of high-level grammatical error correction, it still needs to be resolved by the deep model. The end-to-end deep model can avoid manual feature extraction and reduce manual workload. The Seq2Seq model uses the encoder-decoder structure to solve the sequence conversion problem, one of the most widely used and best-effect models in sequence conversion tasks (such as machine translation, dialogue generation, text summarization, and image description). The RNN sequence model has a strong ability to fit text tasks, and the RNN that incorporates the attention mechanism has a better error correction effect on longer texts. The Transformer model uses a full attention structure instead of LSTM, which is used to solve the sequence-to-sequence issue, and the effect of semantic feature extraction is better. The BERT model uses a fine-tuned strategy and uses mask features to correct typos.

In this paper, we propose a hybrid model (H-ERDC) to conduct error detection and correction, which consists of three designed models mentioned in previous sections. They are SSA-ERDC model, T-ERDC model, and B-ERDC model.

The model of H-ERDC proposed in this paper dynamically merges the outputs of the three submodels, SSA-ERDC model, T-ERDC model, and B-ERDC model, and then outputs the final result. This strategy can combine the advantages of the three submodels, thereby maximizing the performance of error detection and correction.

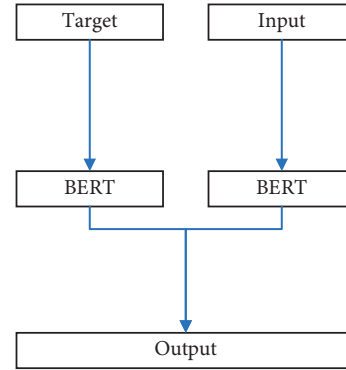


FIGURE 8: The structure of B-ERDC.

4. Experiment and Results Analysis

In this section, the details about the results generation are provided.

4.1. Dataset and Its Description. The training corpus of this paper is mainly divided into three parts. The first part comes from a student's work for a certain English writing exercise provided by an English teacher in a college, and it includes a total of 500 workbooks. At the same time, the teacher will give a prompt of grammatical errors as a comparison with the results of the machine inspection because this paper is to check common grammatical errors in college English writing and CET-4 and CET-6 essays. Therefore, this part of the corpus is the test corpus that most conforms to and best reflects the effect of the grammatical error correction model in this paper. The corpus of the second part comes from five kinds of students, including middle school students in the Chinese English Learner Corpus, college English levels 4 and 6, and professional English lower and upper grades. These assignments have varying degrees of vocabulary and grammatical errors, and manual errors have been marked, which is very beneficial to our experiments. Since each level of English learners has a different emphasis on grammatical errors in different periods, this part of the corpus is mainly used to test the grammatical test effect of this grammatical error correction model in different levels. The last part is an article from the Wall Street Journal. The corpus selected in this part is 100 pieces, all of which are correct sentences. This article manually adds some type errors to make it meet the requirements of the test corpus we need. It is mainly used to compare and analyze the influence of corpus selection on the efficiency of grammar inspection and obtain a targeted inspection model. The evaluation used in this work is precision, recall, and F-0.5 score.

4.2. Evaluation of SSA-ERDC. The SSA-ERDC model combines the sequence-to-sequence model and attention. To prove the effectiveness of this model, this paper conducts experiments on the loss during training and testing performance, and the results are shown in Figure 9.

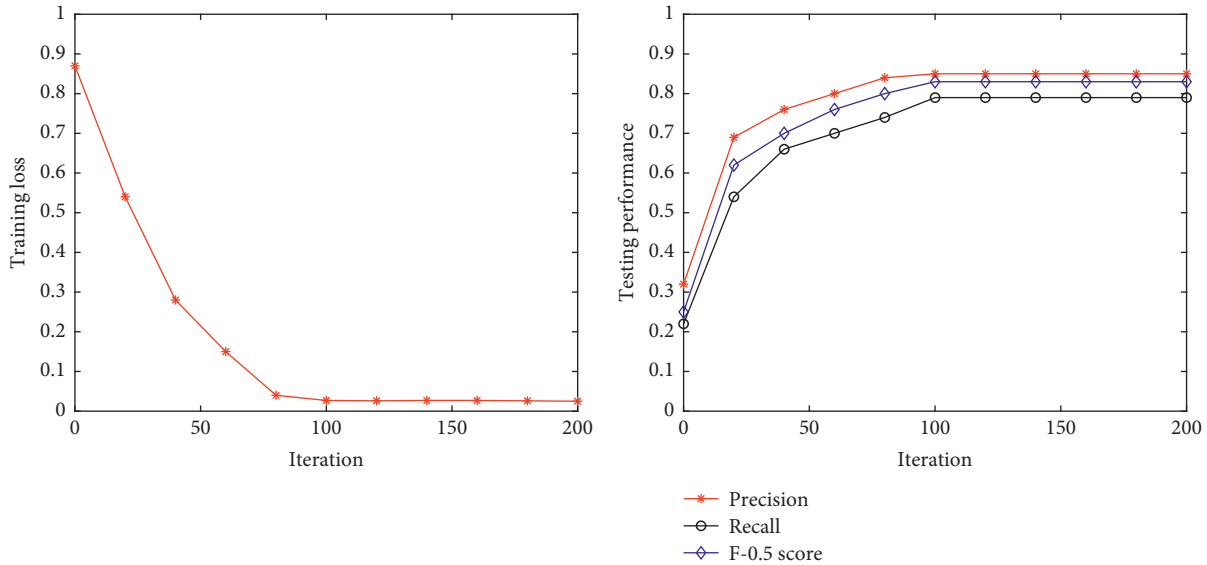


FIGURE 9: Evaluation of SSA-ERDC.

The training loss diminishes as the network training advances, and the training accuracy rate increases. At around 100 iterations, the SSA-ERDC begins to converge. Network reliability and robustness may be ensured by achieving convergent network training with our proposed algorithm. And after the network finally converges, the model can achieve 85% precision, 79% recall, and 83% F-0.5 score.

In order to verify the effectiveness of the attention strategy, this work carried out an additional set of comparative experiments. Compare it with the performance without using attention, and the experimental results are shown in Table 1.

It can be seen that when not using attention, the model can only achieve 81% precision, 75% recall, and 80% F-0.5 score. Compared with the SSA-ERDC model, there are 4%, 4%, and 3% declines on three performance evaluation metrics. This proves the correctness of the SSA-ERDC model designed in this paper, and the attention module can effectively extract more discriminative features.

4.3. Evaluation of T-ERDC. The T-ERDC model combines the Transformer model and error correction. To prove the effectiveness of this model, this paper conducts experiments on the loss during training and testing performance, and the results are shown in Figure 10.

The training loss diminishes as the network training advances, and the training accuracy rate increases. At around 80 iterations, the T-ERDC begins to converge. Network reliability and robustness may be ensured by achieving convergent network training with our proposed algorithm. And after the network finally converges, the model can achieve 88% precision, 81% recall, and 84% F-0.5 score.

TABLE 1: Evaluation of attention mechanism.

Method	Precision (%)	Recall (%)	F-0.5 score (%)
No attention	81	75	80
Attention	85	79	83

4.4. Evaluation of B-ERDC. The B-ERDC model combines the BERT model and error correction. To prove the effectiveness of this model, this paper conducts experiments on the loss during training and testing performance, and the results are shown in Figure 11.

The training loss diminishes as the network training advances, and the training accuracy rate increases. The same as T-ERDC model, when the training iteration is 80, the T-ERDC begins to converge. Network reliability and robustness may be ensured by achieving convergent network training with our proposed algorithm. And after the network finally converges, the model can achieve 90% precision, 84% recall, and 88% F-0.5 score.

4.5. Comparison of the Three Submodels. This paper proposes three submodels for the detection and correction of grammatical errors in English composition. In order to explore the performance gap between different submodels, this section conducts a comparative experiment. This compares the performance of different submodels on three performance indicators. The experimental results are shown in Table 2.

It can be seen that these three submodels have achieved relatively good performance. But compared to the bottom, the performance of the B-ERDC model is the best. Compared to the SSA-ERDC model and the T-ERDC model, it obtains 5% and 2% gains on precision, 5% and 3% gains on recall, and 5% as well as 4% gains on F-0.5 score. This is

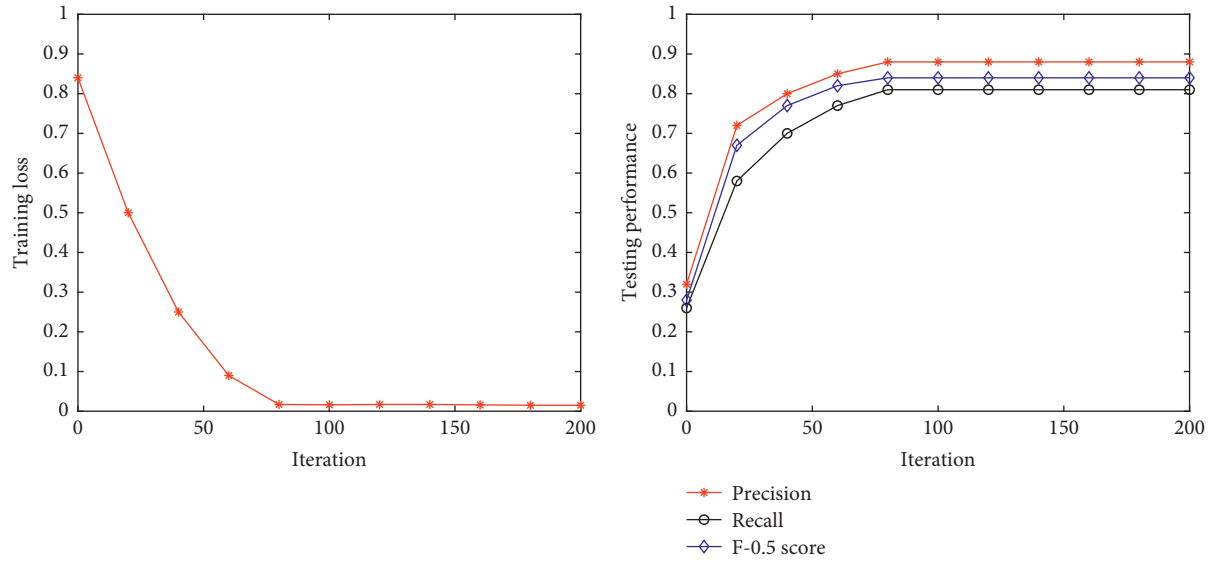


FIGURE 10: Evaluation of T-ERDC.

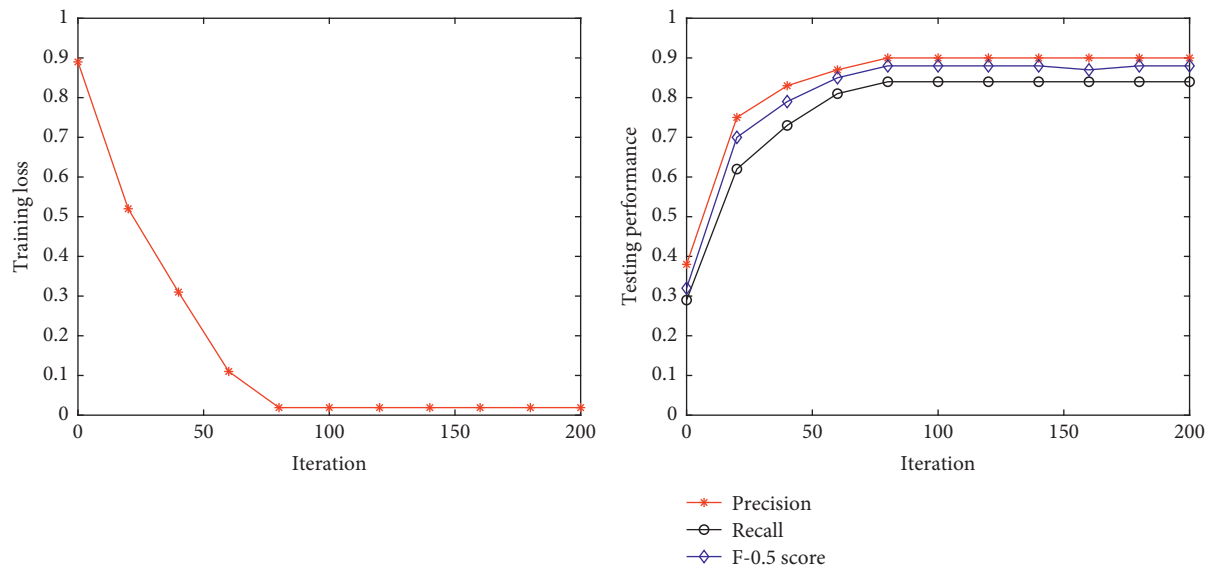


FIGURE 11: Evaluation of B-ERDC.

TABLE 2: Comparison of the three submodels.

Model	Precision (%)	Recall (%)	F-0.5 score (%)
SSA-ERDC	85	79	83
T-ERDC	88	81	84
B-ERDC	90	84	88

mainly because the BERT model overcomes the problem that the forward and backward cannot be merged during training.

4.6. Evaluation of H-ERDC. This work combines the SSA-ERDC model, T-ERDC model, and B-ERDC model to construct H-ERDC, which can fuse different information from different submodels. To verify the effectiveness of this

strategy, we compare our method with other methods for grammatical error detection and correction in English composition. The compared methods contain SVM, Decision Tree, RNN, LSTM, Transformer, and BERT. The experimental result is illustrated in Table 3.

Compared with other methods, the H-ERDC model designed in this paper can achieve the best performance improvement, surpassing all the methods listed in the table. And compared with the data in Table 2, the performance of

TABLE 3: Evaluation of H-ERDC.

Method	Precision (%)	Recall (%)	F-0.5 score (%)
SVM	76	71	74
Decision Tree	79	74	76
RNN	83	76	81
LSTM	85	77	81
Transformer	89	81	83
BERT	90	83	86
H-ERDC	92	87	89

the hybrid model on the three performance indicators is better than the other three submodels. This verifies the validity and reliability of the method in this paper.

5. Conclusion and Future Work

In the context of globalization, English has become one of the most popular languages in the world. For English learners, the learners do not have a wealth of systematic grammar knowledge, and under the influence of their mother tongue, it is difficult to accurately identify and correct grammatical errors when writing English. Therefore, it is necessary to detect and correct grammatical errors in English composition. To begin with, a Seq2Seq-based syntax error detection model is used in this paper. Second, the Transformer model is used to create a grammatical error detection and correction system. This model outperforms the majority of other grammatical models. Third, the BERT model's generalization ability has been greatly improved as a result of this paper's realization of its use in grammar error detection and correction tasks. This eliminates the inability of the Transformer to integrate the forward and backward directions when training the language model. Last, this research presents a hybrid approach for grammatical error identification and correction in English writing. Massive experiments can verify the efficiency of our proposed method.

In future, we plan to work on the use of deep learning methods for the subject problem and achieve higher accuracy.

Data Availability

The datasets used during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was sponsored in part by the "2018 Guangxi Philosophy and Social Science Research Program (Grant no. 18BYY005)."

References

- [1] P.-Y. Wang and H.-C. Yang, "Using collaborative filtering to support college students' use of online forum for English learning," *Computers & Education*, vol. 59, no. 2, pp. 628–637, 2012.
- [2] S. Atef-Vahid and A. F. Kashani, "The effect of English learning anxiety on Iranian high-school students' English language achievement," *BRAIN: Broad Research in Artificial Intelligence and Neuroscience*, vol. 2, no. 3, pp. 29–44, 2011.
- [3] J. M. Carreira, K. Ozaki, and T. Maeda, "Motivational model of English learning among elementary school students in Japan," *System*, vol. 41, no. 3, pp. 706–719, 2013.
- [4] A. I. Fageeh, "EFL learners' use of blogging for developing writing skills and enhancing attitudes towards English learning: an exploratory study," *Journal of Language and Literature*, vol. 2, no. 1, pp. 31–48, 2011.
- [5] T.-Y. Kim, "Korean elementary school students' English learning demotivation: a comparative survey study," *Asia Pacific Education Review*, vol. 12, no. 1, pp. 1–11, 2011.
- [6] S.-W. Yeh, J.-J. Lo, and H.-M. Chu, "Application of online annotations to develop a web-based Error Correction Practice System for English writing instruction," *System*, vol. 47, pp. 39–52, 2014.
- [7] L. Shi and L. Yang, "A community of practice of teaching English writing in a Chinese University," *System*, vol. 42, pp. 133–142, 2014.
- [8] B. Maryam, K. S. Hosein, and N. M. Sharafi, "Peer feedback in learning English writing: advantages and disadvantages," *Journal of Studies in Education*, vol. 3, no. 4, pp. 91–97, 2013.
- [9] J.-Y. Chang, "The use of general and specialized corpora as reference sources for academic English writing: a case study," *ReCALL*, vol. 26, no. 2, pp. 243–259, 2014.
- [10] H. Y. Cho and J. A. Shin, "Cohesive devices in English writing textbooks and Korean learners' English writings," *English teaching*, vol. 69, no. 1, pp. 41–59, 2014.
- [11] G. Adriaens, "Simplified English grammar and style correction in an Mt framework: the Ire secc project," *ASLIB Proceedings*, vol. 47, no. 3, pp. 73–82, 2013.
- [12] L. Hu, Y. Tang, X. Wu et al., "Considering optimization of English grammar error correction based on neural network," *Neural Computing & Applications*, vol. 17, 2021.
- [13] G. Sidorov, "Syntactic dependency based N-grams in rule based automatic English as second language grammar correction," *International Journal of Computational Linguistics & Applications*, vol. 4, no. 2, pp. 169–188, 2013.
- [14] Y.-L. Wu, "The impact of technology on language learning," *Future Information Technology*, Springer, Berlin, Heidelberg, pp. 727–731, 2014.
- [15] L. Aversano, G. Canfora, A. De Lucia et al., "Evolving ispell: a case study of program understanding for reuse," in *Proceedings of the International Workshop on Program Comprehension*, pp. 197–206, IEEE, Paris, France, June 2002.
- [16] F. Bond and S. Ikehara, *When and How to Disambiguate? Countability in Machine Translation*, CiteSeer, in *Proceedings of the International Seminar on Multimodal Interactive*

- Disambiguation*, pp. 29–40, Kanagawa-ken, JAPAN, June 1996.
- [17] J. E. Heine, *Definiteness Predictions for Japanese Noun Phrases*, pp. 519–525, Annual Meeting of the Association for Computational Linguistics and International Conference on Computational Linguistics, 1998.
- [18] R. Mudge, “The design of a proofreading software service,” *NAACL HLT 2010 Workshop on Computational Linguistics and Writing*, pp. 24–32, Writing Processes and Authoring Aids, 2010.
- [19] M. Murata and M. Nagao, “Determination of referential property and number of nouns in Japanese sentences for machine translation into English,” *Tmi*, vol. 12, no. 3, pp. 218–225, 2000.
- [20] N. R. Han, M. Chodorow, and C. Leacock, “Detecting errors in English article usage by nonnative speakers,” *Natural Language Engineering*, vol. 12, no. 02, pp. 115–129, 2006.
- [21] K. Knight and I. Chander, “Automated postediting of documents,” *AAAI*, pp. 779–784, 1994.
- [22] J. R. Tetreault and M. Chodorow, “The ups and downs of preposition error detection in ESL writing,” *International Conference on Computational Linguistics*, pp. 865–872, 2008.
- [23] D. Dahlmeier and H. T. Ng, “Grammatical error correction with alternating structure optimization,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 915–923, ACL, Stroudsburg PA 18360, USA, June 2011.
- [24] M. Gamon, *Using Mostly Native Data to Correct Errors in Learners’ Writing: A Metaclassifier Approach*, pp. 163–171, Human Language Technologies, Columbia, Canada, 2010.
- [25] J. Tetreault, J. Foster, and M. Chodorow, “Using parse features for preposition selection and error detection,” in *Proceedings of the ACL conference short papers. Association for Computational Linguistics*, pp. 353–358, ACL, New York, June 2010.
- [26] R. De Felice and S. G. Pulman, “Automatically acquiring models of preposition use,” in *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions - SigSem ’07*, pp. 45–50, ACL, New York, June 2007.
- [27] A. Rozovskaya and D. Roth, “Algorithm selection and model adaptation for ESL correction tasks,” in *Proceedings of the Conference The 49th Annual Meeting of the Association for Computational Linguistics*, pp. 924–933, Human Language Technologies, Portland, Oregon, USA, June, 2011.
- [28] X. Liu, B. Han, K. Li et al., “SRL-based verb selection for ESL,” in *Proceedings of the Conference on empirical methods in natural language processing*, pp. 1068–1076, ACL, New York, October 2010.
- [29] T. Tajiri, M. Komachi, and Y. Matsumoto, “Tense and aspect error correction for ESL learners using global context,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 198–202, Association for Computational Linguistics, Stroudsburg, Pennsylvania, United States, July 2012.
- [30] M. Gamon, “High-order sequence modeling for language learner error detection,” in *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 180–189, April 2011.
- [31] G. Sidorov, A. Gupta, M. Tozer et al., *Rule-based System for Automatic Grammar Correction Using Syntactic N-Grams for English Language Learning*, pp. 96–101, CoNLL Shared Task, 2013.
- [32] M. Felice, Z. Yuan, E. Andersen Ø et al., *Grammatical Error Correction Using Hybrid Systems and Type Filtering*, pp. 15–24, CoNLL Shared Task, 2014.
- [33] K. Knight and I. Chander, “Automated postediting of documents,” in *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pp. 779–784, Seattle Washington USA, June 1994.
- [34] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.