Research Article

Research and Implementation of English Grammar Check and Error Correction Based on Deep Learning

Xiuhua Wang[1] **and Weixuan Zhong** [2]

[1]*Science and Technology College of Gannan Normal University, Ganzhou 341000, China*
[2]*School of Foreign Language, Hechi University, Yizhou 546300, China*

Correspondence should be addressed to Weixuan Zhong; 03016@hcnu.edu.cn

English as a universal language in the world will get more and more attention, but English is not our mother tongue, and there exist differences in culture and thinking. English grammar is the most difficult problem to solve. There are many English learners, and the number of English teachers is limited, and it is inevitable to use Internet technology to solve the problem of lack of resources. The article uses deep learning technology to propose an ASS grammar detection model, which can quickly and efficiently detect grammatical errors. The research results show the following. (1) This study selects data from the GEC evaluation task and analyzes the four modules of article, noun, verb, and preposition through algorithms under different models. The results indicate the accuracy of the four modules. The recall rate has been improved to a certain extent, the accuracy rate of nouns is the highest, which can reach 63.99%, the accuracy rate of prepositions is improved to a lesser extent, and the inspection accuracy rate after improvement is 12.79%. (2) In the experiment to verify the effectiveness of the ASS grammar detection model, compared with the detection effect of the ordinary model, the accuracy of the ASS comprehensive inspection has been greatly improved. The comprehensive accuracy of the ordinary detection model is 28.01%, and the ASS model's comprehensive accuracy rate of the inspection was 82.82%, and the accuracy rate was increased by 54.81%. The result shows that the performance of the ASS inspection model has been improved by leaps and bounds compared with the traditional model. (3) After transforming and upgrading the ASS model, the three models and other models obtained were run on the test set and the mixed test set, respectively. The results show that the accuracy, precision, recall, and F1 score of ASS model are the highest in the test set, which are 98.71%, 98.83%, 98.64%, and 98.73%, respectively, the Bayesian network check model has the lowest accuracy rate of 51.74%, and the ROC curve value and AUC value of the ASS model are both the largest. The accuracy of the ASS model on the mixed test set is also the highest, reaching 98.01%. The JaSt model on the mixed test set has a significant downward trend, with the accuracy rate dropping from 92.16% to 56.68%. It can be concluded that the ASS model can accurately and efficiently monitor grammatical errors.

1. Introduction

With the increasing update of computers and the Internet, tens of thousands of users tend to write and communicate in English in their daily work. For users whose native language is not English, writing in English is a major obstacle for them. Grammar checking technology originated from the application of natural language understanding. Clément et al. [1] proposed an open grammar checking system under

the deep learning model to analyze and train the grammar in depth. The standard of grammar directly affects the fluency of sentences. The grammar checking system introduced in this article can efficiently check out grammatical errors in sentences and automatically generate correct sentences to replace the wrong ones. Xu [2] improved the algorithm and accuracy of grammar checking and designed and developed a grammar checking system. Sankaravelayuthan [3] proposed an MS-Word tool to check spelling errors in text.

Because a word is composed of many English letters, after we enter the English word, there will inevitably be input errors. The tools proposed in the article will help us solve these problems and will automatically check the spelling errors in the article. Jacobs and Rodgers [4] discussed the use of French computer grammar checker as a learning and teaching resource. They conducted an experiment in which students use a screen checker or other methods to check for grammatical errors in English articles. Lüthy et al. [5] studied the method of segmenting offline cursive handwritten text lines into individual words. Prins [6] found some of the most common mistakes made by Taiwanese students in writing and provided some strategies that teachers use in ESL classrooms. Keong [7] wrote an expert system for English grammar teaching for personal computer users. The system uses a parser to implement a grammatical checking tool, and the system can check for grammatical errors in the text. You can also create files to store grammatical errors and corresponding information from the text. Kann [8] realized the writing method of long text in computer and determined the writing process through the relevant model. Xie [9] implemented the rules of grammar checking according to the principles of practicality and validity. The article simplifies the analysis of the algorithm and expands the coverage of errors. Grammar error checking is a very important task in correcting the text. For people with poor English, writing in English is a relatively difficult task. If English is not good, you cannot use the grammar in English correctly, thus increasing the demand for grammar checking software. The purpose of the literature [10] is to examine the existing literature, highlight current issues, and propose potential directions for future research. The article observes and analyzes the error analysis program, summarizes the error experience, and finds the correct program. The development of computer technology helps enrich the content of English education and provides more convenience for English learning. Pan and Zhou [11] realized personalized inspection and diagnosis of college students' English grammar. Amrhein [12] discussed the importance of correct use of conjunctions and semicolons in the preparation of policy tables to avoid misunderstanding of the intended meaning. Shepheard [13] introduced a method of self-study English grammar, through which you can formulate grammar rules for yourself. Using this learning method can help us learn and understand grammar without the guidance of a teacher. The system will list students' common grammatical errors, and students can conduct intensive training based on the grammatical errors listed by the system. Mondal and Mondal [14] introduced a proprietary software application. The program can provide many services, including helping us detect grammatical errors in English articles and automatically generate correct sentences. Richards et al. [15] introduced a two-level general English course for Italian middle school students. The course mainly emphasizes the communication methods of accuracy and fluency. The course includes three parts: communication check, grammar check, and study check. The above research methods are based on artificial intelligence or modern technology applied to English grammar detection

Table 1: Classification of grammatical errors [17].

Type	Content
Vt	Verb tense
Vm	Verb modality
V0	Missing verb
V form	Verb form
SVA	Subject-verb agreement
Art or Det	Article
Nn	Noun (singular and plural)
Npos	Word possessive
Pform	Pronoun form
Pref	Pronoun reference

and verification. However, the detection efficiency of grammar is low, the error rate is high, and the prediction efficiency is not ideal. In this paper, an ASS grammar detection model is proposed by using deep learning technology, which can detect grammar errors quickly and efficiently.

2. Research and Realization of English Grammar Check

2.1. The Significance of Grammar Checking Research. Due to the great flexibility and uncertainty of natural language itself, English is a typical representative of many vocabulary, complex grammar, and extensive usage scenarios, which increases the difficulty of computer automatic error detection and correction. Another important reason that affects the development of grammatical error correction is the lack of relevant corpus. It is very difficult to construct a corpus marked with grammatical errors. The current mainstream research methods of grammatical error correction are all based on statistical machine learning, which requires a large amount of corpus for model training and testing [16]. However, with the attention of universities and research institutions on this issue, the problem of lack of corpus has been greatly improved, laying a solid foundation for further research. Therefore, this article will study deep learning technology and use it to solve the problem of English grammar error correction. Based on the proposed error correction algorithm, the algorithm model experiment and verification are carried out, and considering the application of the algorithm, a grammatical error correction system similar to Google translation is constructed, providing a simple and convenient way for English learners to use. The combination of theory and practice has a certain promotion effect on solving the problem of English grammar error correction and improving the grammar level of English learners. The classification of common grammatical errors is shown in Table 1.

2.2. The Overall Framework Design of English Grammar Check. According to the results of the functional analysis of business requirements, the architecture of the grammar error correction system model [18] is shown in Figure 1.

The core module of grammar error correction mainly includes three functional modules: data processing, model
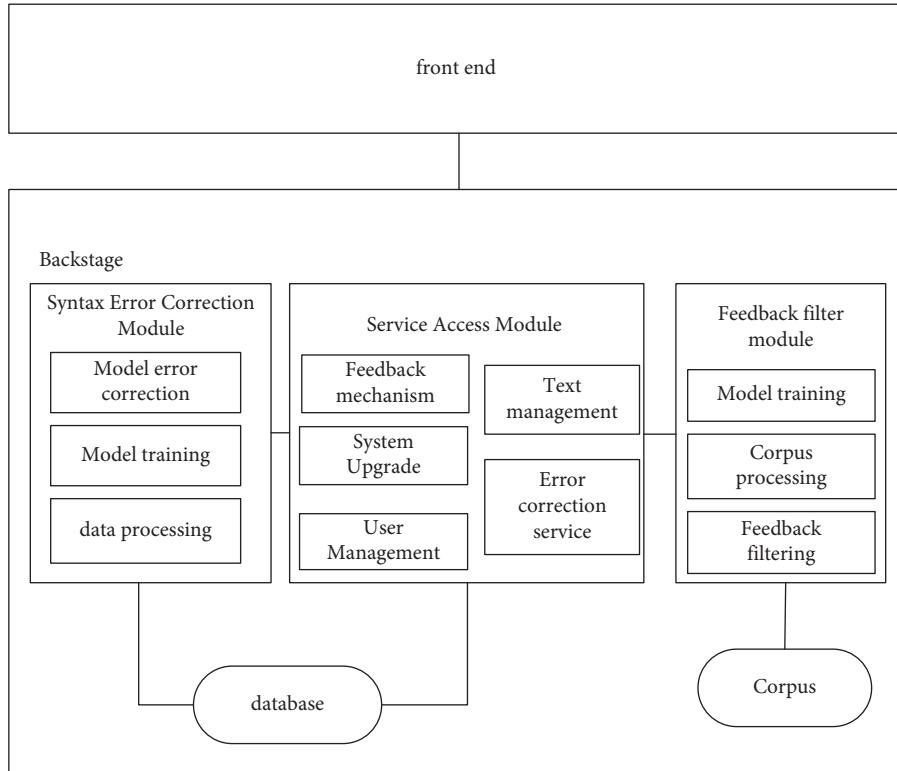
FIGURE 1: Framework diagram of grammatical error correction model.

training, and model error correction, and model error correction is the core function of the whole algorithm [19]. The main function of data processing is to preprocess the original corpus data, to store the processed corpus data in a structured way, and to learn the module and get the standard dataset. Model training is to train the data in the corpus, save the trained features in the database, and apply them in the later test and matching [20]. Model error correction is to use the error correction model stored in the training library to match the input sentence grammar and output the correct sentence. The error correction service model can accept the error correction request from the user in real time, analyze it through the error correction model of the corpus, and return the correct content to the user.

2.3. Implementation of Syntax Error Correction. Firstly, learning is carried out according to the characteristics of grammar error correction, and an English error correction application submitted by a user is received. First determine whether the submitted parameters are valid and then enter the next step to split the sentence. Then, the error correction model trained before is used for grammar error correction. When the error correction of the last sentence is completed, the error correction sentences returned to the segmentation are merged. If the sentence is simple, the error correction model can be used directly without sentence segmentation. Feedback suggestion means that when the user is not satisfied with the grammatical error correction given by the system or there is a better way to modify it, the modification suggestion is fed back to the

system. As mentioned above, we will filter the modification suggestions submitted by users, so the previous feedback suggestion filtering model will be used in the feedback suggestion function. Similar to grammatical error correction, we also carry out the design of feedback suggestions from two aspects. One is the feedback filtering interface itself, and its work flowchart is given; the other is the call flow between modules, which is explained using sequence diagrams. First, we introduce the feedback filtering interface. According to the syntax error correction process, first determine whether the request parameters are legal; if not, directly end [21]. The probabilities of the error correction statement and the original system modification statement are calculated, respectively, for the error correction model.

3. Error Correction Model

3.1. Deep Learning Technology. The seq2seq model is composed of an encoder and is a cyclic neural network; in the encoding stage, a semantic vector is generated according to the input sequence $(x_1, x_2, \ldots)$ and conversion rule f [22], and the calculation formula is

$$h_t = f(h_{t-1}, x_t). \tag{1}$$

Summarize the semantic vector:

$$C = q(h_1, h_2, h_3, \ldots, h_t). \tag{2}$$

Decoding:

$$y_t = \arg\max P(y_t) = \prod_{t=1}^{T} p(y_t | \{y_1, y_2, \ldots, y_{t-1}\}, C). \quad (3)$$

The calculation formula of the hiding algorithm:

$$h_t' = f(h_{t-1}', y_{t-1}, C). \quad (4)$$

3.2. Evaluation Criteria for English Grammar Error Correction.
The most commonly used evaluation algorithm for grammatical error correction is Max Match (M^2). The principle of the algorithm Max Match is introduced below [23]; correction rate P:

$$P = \frac{\sum_{i=1}^{n} |g_i \cap e_i|}{\sum_{i=1}^{n} |e_i|}. \quad (5)$$

Correction rate R:

$$R = \frac{\sum_{i=1}^{n} |g_i \cap e_i|}{\sum_{i=1}^{n} |e_i|}. \quad (6)$$

The key evaluation index in Max Match is $F_{0.5}$, and the formula is defined as follows:

$$F_{0.5} = \frac{(1 + 0.5^2) * R * P}{R + 0.5^2 * P}. \quad (7)$$

3.2.1. Syntax Error Correction Model.
In the Soft Attention Mechanism, the weight a_{ij} is determined by the $(i-1)$th hidden state $s_i - 1$ and each hidden state variable in the input [24]. The calculation formula is as follows:

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \quad (8)$$

$$e_{ij} = a(s_{i-1}, h_j).$$

LN is calculated by inputting the entire layer of neurons in RNN [25]:

$$\mu = \frac{1}{H} \sum_{i=1}^{H} x_i,$$

$$\sigma = \sqrt{\frac{1}{H} \sum_{i=1}^{H} (x_i - \mu)^2}. \quad (9)$$

Anti-filtering algorithm:

$$p(\omega_i | \omega_1, \ldots, \omega_{i-1}) = p(\omega_i | \omega_{i-n-1}, \ldots, \omega_{i-1}). \quad (10)$$

When $n = 2$, bigram is

$$p(\omega_i | \omega_1, \ldots, \omega_m) = \prod_{i=1}^{m} p(\omega_i | \omega_{i-1}). \quad (11)$$

When $n = 3$, bigram is

$$p(\omega_i | \omega_1, \ldots, \omega_m) = \prod_{i=1}^{m} p(\omega_{i-2}\omega_{i-1}). \quad (12)$$

Estimate the value of $p(\omega_i | \omega_{i-2}\omega_{i-1})$, and the formula is

$$p(\omega_i | \omega_{i-2}\omega_{i-1}) = \frac{c(\omega_{i-2}\omega_{i-1}\omega_i)}{c(\omega_{i-2}\omega_{i-1})}. \quad (13)$$

According to the N-gram grammar model introduced above, we can get

$$P(S) = P(\omega_1, \omega_2, \ldots, \omega_m). \quad (14)$$

Confusion:

$$PP(S) = P(\omega_1, \omega_2, \ldots, \omega_m)^{1/m}$$

$$= {}^m\sqrt{\frac{1}{P(\omega_1, \omega_2, \ldots, \omega_m)}}. \quad (15)$$

According to the chain method, it can be written as

$$PP(S) = {}^m\sqrt{\prod_{i=1}^{m} p(\omega_i | \omega_{i-n-1}, \ldots, \omega_{i-1})}. \quad (16)$$

3.2.2. ASS Model Design.
The objective function of a single-layer neural network is

$$\text{vec}(p) \approx \tanh\left(\sum_{i=1}^{n} \frac{\Omega(P)}{\Omega(c_i)} W_{\text{code},i} \cdot \text{vec}(c_i) + b\right), \quad (17)$$

$$W_{\text{code},i} = \eta_l W_{\text{code}}^l + \eta_r W_{\text{code}}^r.$$

The coefficients η_l and η_r are

$$\eta_l = \frac{\Omega(p) - i}{\Omega(p) - 1},$$

$$\eta_r = \frac{i - 1}{\Omega(p) - 1}. \quad (18)$$

The matrix parameters of the only child node are

$$W_{\text{code},i} = \frac{1}{2} W_{\text{code}}^l + \frac{1}{2} W_{\text{code}}^r. \quad (19)$$

Square of Euclidean distance:

$$d = \left\| \text{vec}(p) - \tan h\left(\sum_{i=1}^{n} l_i W_{\text{code},i} \cdot \text{vec}(c_i) + b\right) \right\|_2^2. \quad (20)$$

The error function of training sample $x^{(i)}$ and its corresponding negative example $x_c^{(i)}$ is

$$J(d^{(i)}, d_c^{(i)}) = \max(0, \Delta + d^{(i)} - d_c^{(i)}). \quad (21)$$

The final training objective function is

TABLE 2: Data statistics of evaluation tasks.

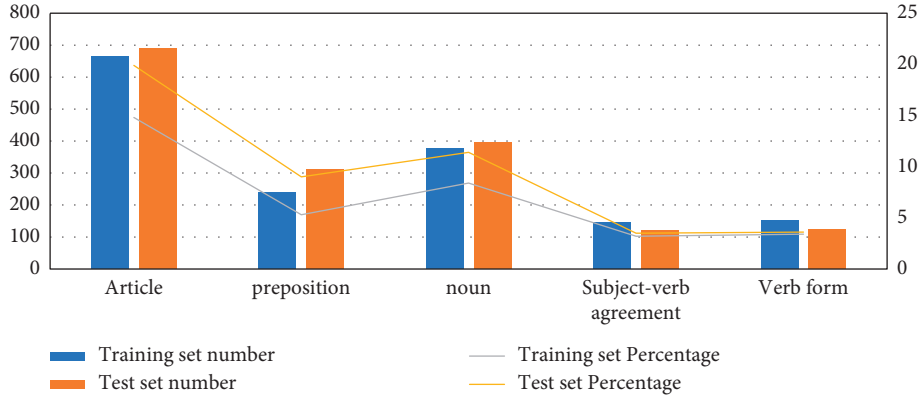| Type of error	Training set number	Training set percentage	Test set number	Test set percentage
Article	665	14.8	690	19.9
Preposition	240	5.3	312	9.0
Noun	377	8.4	396	11.4
Subject-verb agreement	145	3.2	122	3.5
Verb form	152	3.4	124	3.6
Five types	582	35.1	1644	47.4
All types	651	100	3470	100.0

FIGURE 2: Statistical graph of experimental results.
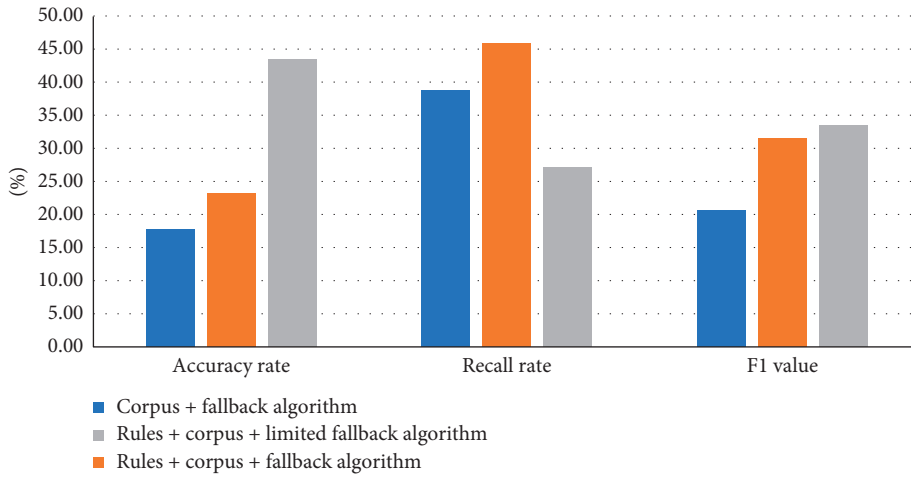
FIGURE 3: Statistics of article inspection results.

$$\min_{W^l_{\text{code}}} \frac{1}{2N} \sum_{i=1}^{N} J\left(d^{(i)}, d_c^{(i)}\right) + \frac{\lambda}{2M} \left(\left\| W^l_{\text{code}} \right\|_F^2 + \left\| W^r_{\text{code}} \right\|_F^2 \right).$$

(22)

4. Simulation Experiment

4.1. Data Analysis. In order to effectively analyze the data, we select the data in the GEC evaluation task and analyze the algorithms of the four modules of article, noun, verb, and preposition under different models. The experiment compares the results by setting whether to use the law library, and in order to enhance for the persuasiveness of the experiment, we have added a comparative experiment with the common algorithm. The statistical results of various error types in the training data and test data are shown in Table 2 and Figure 2.

The five types include errors in articles, errors in prepositions, errors in names, grammatical errors in subject-verb agreement, and errors in verb forms, while the all types include errors in verb tenses, missing verbs, verb forms, subject-verb agreement, articles, singular and plural nouns, possessive words, and pronoun forms. Table 2 shows an experimental analysis of five common types of grammatical errors.

TABLE 3: Comparison of article check module results.

Model	Accuracy rate (%)	Recall rate (%)	F1 value (%)
Corpus + fallback algorithm	17.79	38.70	20.71
Rules + corpus + fallback algorithm	23.25	45.88	31.44
Rules + corpus + limited fallback algorithm	43.49	27.12	33.45

TABLE 4: Comparison of noun check module results.

Model	Accuracy rate (%)	Recall rate (%)	F1 value (%)
Corpus + fallback algorithm	63.99	30.05	40.90
Corpus + limited fallback algorithm	4.06	51.26	45.3

TABLE 5: Comparison of verb check module results.

Model	Accuracy rate (%)	Recall rate (%)	F1 value (%)
Corpus + fallback algorithm	12.79	22.36	16.27
Corpus + limited fallback algorithm	19.37	22.36	20.27

TABLE 6: Comparison of results of preposition checking module.

Model	Accuracy rate (%)	Recall rate (%)	F1 value (%)
Corpus + fallback algorithm	8.01	25.32	12.17
Corpus + limited fallback algorithm	13.46	18.83	15.70

4.1.1. Article Inspection Module. The accuracy and recall rate of adding the rule library to the article check module have been significantly improved, indicating that the automatic extraction of the rule library is effective for the entire inspection and correction process as shown in Figure 3 and Table 3. At the same time, the fallback algorithm is improved. After the limited back-off algorithm, the accuracy rate has also been greatly improved, and the correction process is more accurate, thereby increasing the final F1 value.

4.1.2. Noun Check Module. In the noun checking module of Table 4, after using the model algorithm, the accuracy and recall rate have been greatly improved, and the accuracy rate is up to 63.99% because nouns account for the highest proportion of sentences. When using the grammar check module, you can correct more noun errors, thereby increasing the F1 value of the noun check module.

4.1.3. Verb Check Module. 4.1.4. Preposition Checking Module. As shown in the data in Tables 5 and 6, in the grammar detection module, the accuracy and recall rate of verbs and prepositions have been improved, but the accuracy of prepositions has been improved to a lesser extent. The improved inspection accuracy rate is 12.79%. After using the fallback algorithm, the model's judgment on grammar detection is stricter. The accuracy measure is the ratio of the number of correct samples to the number of all samples in the test set. The larger the index value is, the more accurate the recommendation result is. The F1 measurement index can effectively balance the precision and recall by biasing the objects with small values. The larger the index

value is, the more accurate the recommendation result is. The improved recall of the verb and preposition checking modules of Table 5 and Table 6 illustrates that the results of the verb and preposition checking modules are more precise.

4.2. Comparison of Test Results. We compared the inspection effect of the ASS model with the inspection effect of the ordinary model; the inspection effect of the grammar inspection module under the ordinary algorithm is shown in Table 7, and the comprehensive grammar inspection result of the ASS model is shown in Table 8.

From the data in Figures 4 and 5, we can conclude that the accuracy of the ASS comprehensive inspection has been greatly improved compared with the inspection effect of the ordinary model. The comprehensive accuracy of the ordinary inspection model is 28.01%, and the ASS model inspection is better. The overall accuracy rate is 82.82%, the accuracy rate is increased by 54.81%, and the overall recall rate of the ASS model is also increasing, indicating that the performance of the ASS inspection model has been improved by leaps and bounds, and the efficiency of grammar detection and the correctness of grammar detection have been improved.

4.3. Model Performance Testing. We run each model on the test set and the mixed test set and record the experimental data. In the process of using the ASS model to detect grammar, the grammar needs to be converted into a mathematical representation that the model can handle, as shown in Figure 6.

According to the data in Table 9 and Figure 7, we can conclude that the accuracy of the ASS model is the highest among several models, reaching 99.71%, indicating that the

TABLE 7: Common model detection effect.

	Accuracy rate (%)	Recall rate (%)	F1 value (%)
Verb error detection	21.67	21.21	30.41
Subject-verb consistency error detection	29.39	33.84	31.46
Noun singular and plural error detection	31.32	30.61	30.96
Article error detection	27.12	18.26	21.83
Preposition error detection	27.62	36.58	31.47
Overall error detection	28.01	37.19	31.96

TABLE 8: ASS comprehensive inspection results.

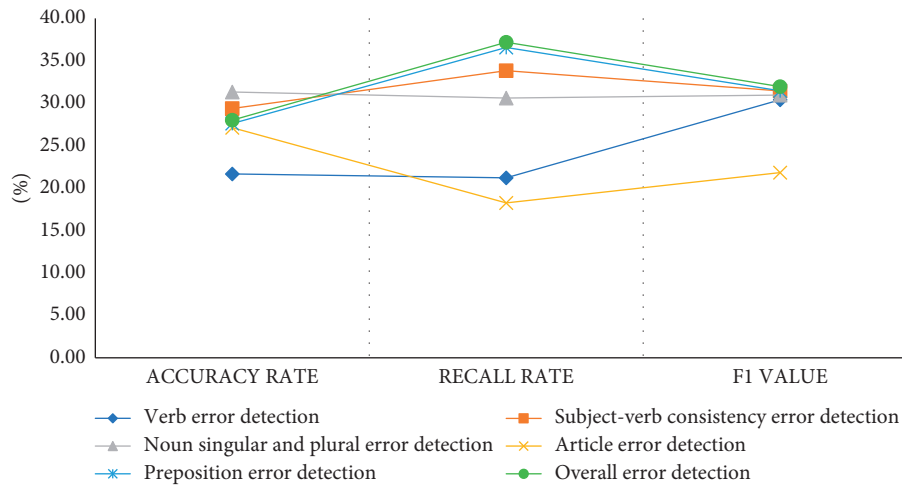|	Accuracy rate (%)	Recall rate (%)	F1 value
Verb error detection	30.12	20.12	34.21%
Subject-verb consistency error detection	40.00	21.33	31.71%
Noun singular and plural error detection	45.51	28.77	35.50%
Article error detection	67.39	42.84	52.38
Preposition error detection	72.13	37.87	49.66%
Overall error detection	82.82	25.57	39.07%

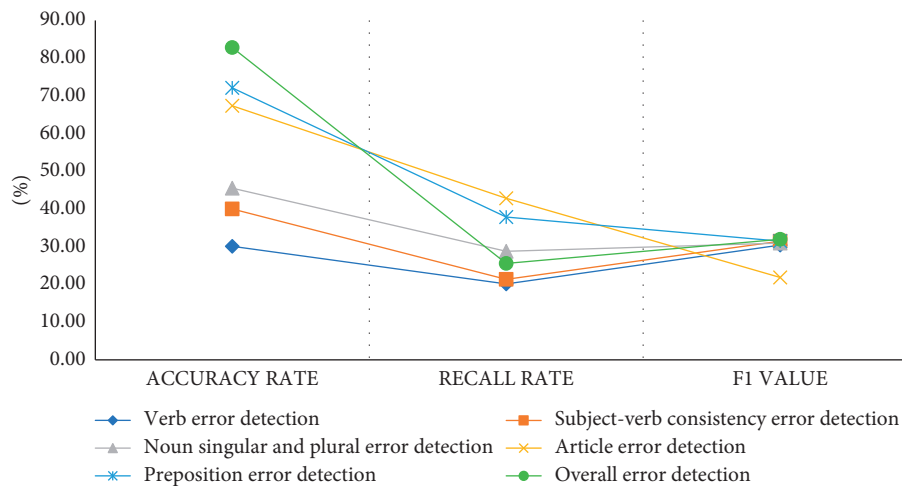FIGURE 4: Statistics of general inspection results.

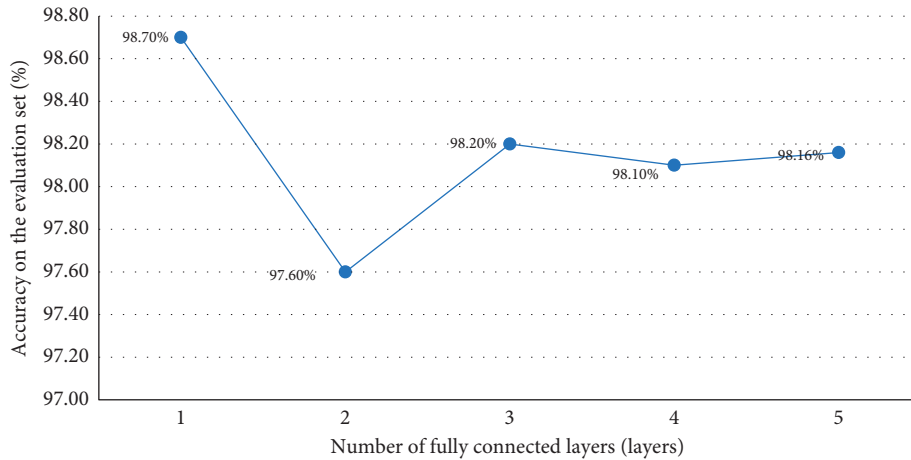FIGURE 5: Statistics of ASS inspection results.

FIGURE 6: The influence of the number of fully connected layers on the model.

TABLE 9: Performance of each model on the test set.

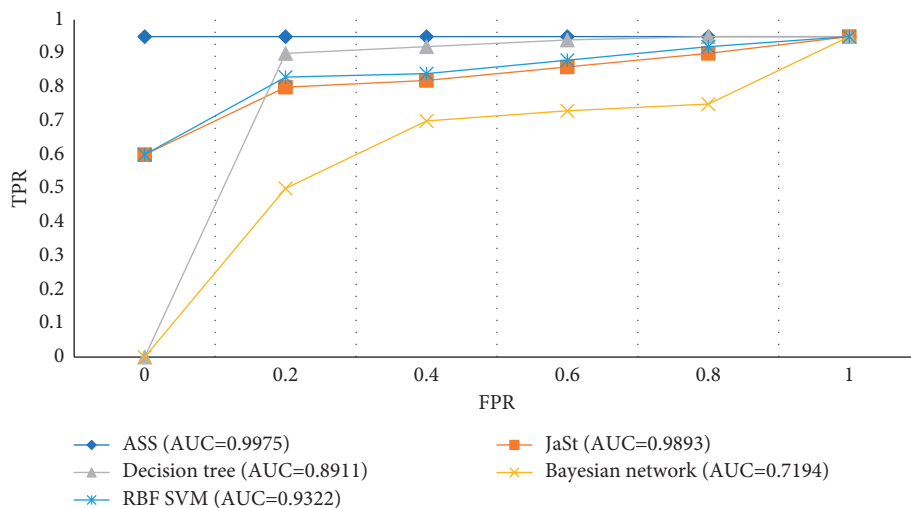| Model	Accuracy rate (%)	Precision rate (%)	Recall rate (%)	F1 score (%)
RBF core support vector machine	85.00	93.06	76.13	83.75
Bayesian network	51.74	51.29	97.83	67.30
Decision tree	89.12	88.83	89.87	89.35
JaSt	92.16	96.71	92.53	94.57
ASS-T	97.97	98.10	97.89	98.00
ASS-G	96.52	98.26	94.83	96.52
ASS-TG	99.71	98.83	97.74	98.82

FIGURE 7: ROC curve on the test set.

performance of ASS detection is the highest, and the accuracy of the Bayesian network is the lowest, which is 51.74%, indicating that the detection efficiency of the Bayesian network model is not good enough.

ASS-T is a test of the data model and the overall syntax, starting with the creation of a new window or table. For each participating object, list the different domains and syntaxes. With the help of field definitions and basic techniques, ASS-G analyzes test data, overall syntax tests, partitions, and boundary values. The ASS-TG data model is the detailed

syntax test. For the strictly syntactically controlled parts, you need to perform more detailed tests.

According to the data in Table 10 and Figure 8, it is concluded that the accuracy rate of the ASS-G model is as high as 98.01%. The JaSt model's various indicators on the mixed test set have a significant downward trend, and the accuracy rate has dropped from 92.16% to 56.68%, because that syntax information in the test set is obfuscated, and the excellence of the ASS model is also reflected in the ROC curve.

TABLE 10: Performance of each model on the mixed test set.

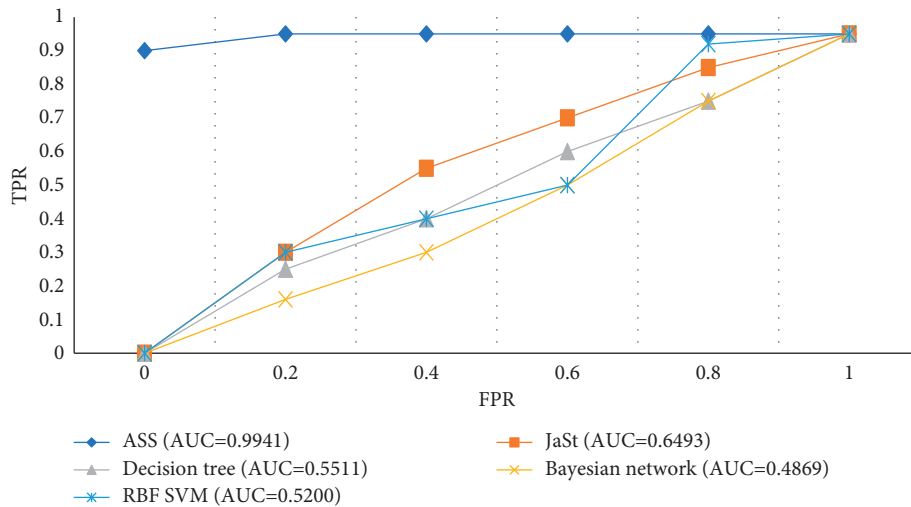| Model	Accuracy rate (%)	Precision rate (%)	Recall rate (%)	F1 score (%)
RBF core support vector machine	47.51	80.77	7.45	13.63
Bayesian network	57.70	56.37	90.75	69.55
Decision tree	54.49	61.25	49.63	54.83
JaSt	56.68	71.49	24.75	36.77
ASS-T	92.71	95.03	91.70	93.33
ASS-G	98.01	98.67	97.73	98.20
ASS-TG	96.72	98.54	95.53	97.01

FIGURE 8: ROC curve on the mixed test set.

5. Conclusion

At present, there are more and more English learners, and the English grammar module is also a very important part of the English learning process. However, due to the particularity of English teaching, the auxiliary teaching of English grammar detection is particularly important, although the current grammar-assisted teaching has been combined with computers. Technology and network technology have greatly reduced the error rate, but there are still some problems with poor user experience. There is still a lot of room for improvement in English assisted teaching. Therefore, it should combine the current problems to improve constantly and propose a more intelligent and accurate grammar detection model to make English teaching easier and more efficient.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. Clément, K. Gerdes, and R. Marlet, "A grammar correction algorithm – deep parsing and minimal corrections for a grammar checker," in *Proceedings of the 14th international conference on Formal grammar*, vol. 12, no. 6, pp. 11–17, Springer, Berlin, Germany, July 2009.

[2] X. Xu, "Exploration of English composition diagnosis system based on rule matching," *International Journal of Emerging Technologies in Learning*, vol. 13, no. 7, pp. 161–172, 2018.

[3] R. Sankaravelayuthan, "Spell and grammar checker for Tamil," *Developing computing tools for Tamil*, vol. 05, no. 23, pp. 52–64, 2015.

[4] G. Jacobs and C Rodgers, "Treacherous allies: foreign language grammar checkers," *CALICO Journal*, vol. 16, no. 4, pp. 87–95, 1999.

[5] F. Lüthy, T. Varga, and H. Bunke, "Using hidden markov models as a tool for handwritten text line segmentation," in *Proceedings of the international conference on document analysis & recognition*, vol. 12, no. 8, pp. 117–123, IEEE, Curitiba, Brazil, September 2007.

[6] H. Prins, "Conquering Chinese English in the ESL classroom," *Internet Tesl Journal*, vol. 03, no. 11, pp. 25–36, 2006.

[7] C. C Keong, "An expert system for the teaching of English grammar," in *Proceedings of the IEEE Region 10 Conference on Computer & Communication Systems, IEEE Tencon*, vol. 08, no. 12, pp. 52–57, IEEE, Hong Kong, China, September 1990.

[8] V. Kann, "CrossCheck – a grammar checker for second language writers of Swedish," *KTH Nada*, vol. 23, no. 12, pp. 78–82, 2008.

[9] K. W. Xie, "An grammar check research based on instance," *Journal of Hubei University for Nationalities Natural Science Edition*, vol. 23, no. 12, pp. 22–31, 2009.

[10] M. Soni and J. S. Thakur, "A systematic review of automated grammar checking in English language," *Computation and Language*, vol. 07, no. 12, pp. 112–121, 2018.

[11] D. Pan and H. Zhou, "English learning system design for college students personalized English grammar check and diagnosis," *International Journal of Emerging Technologies in Learning*, vol. 13, no. 4, pp. 51–59, 2018.

[12] C. Amrhein, "Grammar check," *Property & Casualty*, vol. 07, no. 12, pp. 141–152, 2016.

[13] J. Shepheard, "Teach yourself English grammar as a foreign language," *Academic Leadership Journal in Student Research*, vol. 23, no. 6, pp. 12–18, 2003.

[14] H. Mondal and S. Mondal, "Free grammar check for your manuscript," *Indian Journal of Vascular and Endovascular Surgery*, vol. 11, no. 3, pp. 21–27, 2020.

[15] J. C. Richards, J. Hull, S. Proctor, and D Haines, *Changes 1 workbook Italian edition: English for International Communication*, Cambridge University Press, Cambridge, England, 2010.

[16] P. Ratanaworabhan, B. Livshits, and B. Zorn, "Nozzle: a defense against heap-spraying code injection attacks," in *Proceedings of the SSYM'09 18th conference on USENIX security symposium*, pp. 169–186, Berkeley, CA, August 2009.

[17] T. N Hwee, S. M. Wu, T. Briscoe, C. Hadiwinoto, and C. Bryant, "The CoNLL-2014 shared task on grammatical error correction," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pp. 1–14, Shared Task, Baltimore, MA, USA, January 2014.

[18] H. G. Kim, D. J. Kim, S. J. Cho, and M. j. Park, "Efficient detection of malicious web pages using high-interaction client honeypots," *Journal of Information Science and Engineering*, vol. 28, no. 5, pp. 911–924, 2012.

[19] K. A. Moore, C. Rutherford, and K. A. Crawford, "Supporting postsecondary English language learners' writing proficiency using technological tools," *Journal of International Students*, vol. 6, no. 4, pp. 857–872, 2016.

[20] Y. Guo and G. H. Beckett, "The hegemony of English as a global language: reclaiming local knowledge and culture in China," *Convergence*, vol. 40, pp. 117–132, 2006.

[21] E. S. Atwell and S. Elliot, "Dealing with ill-formed English text the computational analysis of English," *A Corpus-Based Approach*, vol. 12, pp. 120–138, 1987.

[22] R. Dale and A. Kilgarriff, "Helping our own: the HO0 2011 pilot shared task," in *Proceedings of the 13th European Workshop on Natural Language Generation*, vol. 05, no. 14, pp. 242–249, Association for Computational Linguistics, Nancy, France, September 2011.

[23] P. Bhaskar, A. Ghosh, S Pal, and S Bandyopadhyay, "May I check the English of your paper," in *Proceedings of the 13th European workshop on natural language generation*, vol. 03, no. 12, pp. 250–253, Association for Computational Linguistics, Nancy, France, September 2011.

[24] E. Ivanova, D. Bernhard, and C. Grouin, "Handling outlandish occurrences: using rulesand lexicons for correcting NLP articles," in *Proceedings of the 13th European Workshop on Natural Language Generation*, vol. 12, no. 1, pp. 254–256, Association for Computational Linguistics, Nancy, France, September 2011.

[25] K. Kukich, "Techniques for automatically correcting words in text," *ACM Computing Surveys*, vol. 24, no. 4, pp. 377–439, 1992.