Research Article

Fan Xiao and Shehui Yin*

English grammar intelligent error correction technology based on the n-gram language model

https://doi.org/10.1515/jisys-2023-0259
received November 14, 2023; accepted April 16, 2024

Abstract: With the development of the Internet, the number of electronic texts has increased rapidly. Automatic grammar error correction technology is an effective safeguard measure for the quality of electronic texts. To improve the quality of electronic text, this study introduces a moving window algorithm and linear interpolation smoothing algorithm to build a Cn-gram language model. On this basis, a syntactic analysis strategy is introduced to construct a syntactic error correction model integrating Cn-gram and syntactic analysis, and English grammar intelligent error correction is carried out through the model. The results show that compared with the Bi-gram and Tri-gram, the precision of the Cn-gram model is 0.85 and 0.91% higher, and the *F*1 value is 0.97 and 1.14% higher, respectively. Compared with the results of test set Long, the Cn-gram model has better performance on verb error correction of the Short test set, and the precision rate, recall rate, and *F*1 value are increased by 0.86, 3.94, and 1.87%, respectively. The comparison of the precision, recall rate, and *F*1 value of the proposed grammar error correction model on the complete test set shows that the precision of the study is 19.10 and 5.41% higher for subject–verb agreement errors. The recall rate is 9.55 and 10.77% higher, respectively; *F*1 values are higher by 12.65 and 10.59%, respectively. The above results show that the error-correcting technique of the research design has excellent error-correcting performance. It is hoped that this experiment can provide a reference for the relevant research of automatic error correction technology of electronic text.

Keywords: grammar error correction, move the window, n-gram algorithm, linear interpolation smoothing algorithm

1 Introduction

With the growing development of Internet technology, the number of electronic texts in the network is increasing, but the quality of their texts is declining [1]. The text existing in the network often contains various types of errors, and the workload of error correction is relatively large. Traditional manual error correction methods are no longer suitable for the rapidly growing number of electronic texts [2]. Therefore, faster and more efficient text error correction methods urgently need to be proposed [3]. Meanwhile, the continuous progress of intelligent technology has brought the intelligent processing of natural language into people's vision [4]. Computer automatic text correction has become a new direction for electronic text correction, and grammar correction algorithms have also attracted the attention of many scholars. Common English grammar errors include non word errors, true word errors, and grammar errors, among which grammar errors are

*** Corresponding author: Shehui Yin**, Fundamental Teaching Section, Henan Polytechnic Institute, Nanyang, 473000, China, e-mail: 2007006@hnpi.edu.cn
Fan Xiao: College of International Education and Cultural Tourism, Henan Polytechnic Institute, Nanyang, 473000, China, e-mail: xy071018@126.com

caused by word errors [5]. Grammar errors themselves have a certain level of complexity, and their correction work faces significant challenges. Although there have been studies that have optimized its methods, it generally has drawbacks such as complex operation and low precision, so there is still significant room for development. In view of this, to further improve the performance and precision of error correction models, this study conducts relevant discussions on grammar errors such as articles, prepositions, nouns, verbs, and active consistency in grammar and optimized common n-gram models. The innovation of this study lies in (1) Considering the impact of n values on model precision, an n-ary model of clauses is established. (2) A smart English grammar error correction technique based on the Cn-gram language model is designed by introducing the moving window algorithm and linear interpolation smoothing algorithm. (3) Grammar analysis strategies are introduced to correct complex errors such as long sentences.

This study is divided into four parts. The first part is related work, which mainly introduces the research status of the English grammar error correction model. The second part is the construction of the English grammar intelligence error correction technical model. The third part is the performance verification of the experimental model. The last part summarizes and prospects the full text.

2 Related work

With the increasing popularity of English in the world, the study of English has attracted more attention. Grammar is a difficult problem in English learning. Intelligent error correction has brought a new opportunity for it, and many researchers have discussed it. Based on the actual needs of English grammar correction, Hu et al. built a neural network-based English grammar correction model. The innovation of this model lay in the clustering method used to compress the article features. After feature selection in the proposed model, a logical regression model was applied to analyze the influence of different features on grammatical error correction. The validity of the model was finally verified by experiments [6]. Huang et al. paid attention to the important role of artificial intelligence in language education and, based on this background, investigated the relevant research studies on the integration of artificial intelligence into language education in the past 20 years. Artificial intelligence was often used in writing, reading, vocabulary, grammar, and other aspects of language learning to help students learn better. This research laid a foundation for intelligent innovation in language learning [7]. From the perspective of artificial intelligence speech recognition, Duan et al. proposed to apply this technology to correct teachers' spoken pronunciation. In this study, the traditional speech recognition technology was analyzed and improved, that was, a phoneme-level speech error correction method was introduced, and the basic flow of speech cutting was explained in detail. The proposed method could effectively correct spoken English pronunciation and had a certain reference value for the research content of this article [8]. Zhou et al. proposed an English grammar error correction model based on classification models. The model structure and model optimizer of the syntactic error correction algorithm were analyzed in this model to realize the syntactic error correction function of the whole model better. The classification precision of the proposed model could be increased with the increase of its training samples, and it required less overall running time and memory. The successful construction of this model provided a certain reference value for the innovation of English grammar correction algorithms [9]. Park et al. proposed a new indicator to address the problem of excessive correction in English grammar correction so as to more comprehensively consider the correction performance in the process of grammar correction. The proposed model could effectively improve the problem of over-correction and provide a new development perspective for the task of grammar correction [10].

Based on the intelligent review of English compositions, He proposed an algorithm for detecting grammatical errors in English verbs based on recurrent neural networks. Based on the advantage that the model could effectively retain the context-valid information during the training process, the algorithm modeled the labeled training corpus. Finally, the proposed model used the word embedding method to encode the text and mapped the text information to the low-dimensional vector space to avoid information loss. The proposed model showed good advantages in English verb error detection [11]. Wang conducted research on syntactic error correction, and in the process of developing machine translation systems, the generalized maximum

likelihood ratio algorithm was improved, and finally, an English parser was designed. Character mapping function was introduced to realize automatic recognition of sentence boundaries. Through the analysis of example sentences, this study effectively verified the effective performance of its syntax error correction [12]. Aiming at the limitations of current neural machine translation methods, Zhao and Wang proposed a syntactic error correction model optimized by dynamic masking. In the process of training, the model dynamically added random masks to the source sentences to generate more diverse sentence instances, thus improving the generalization ability of the error correction model. The excellent performance of the proposed syntax error correction was finally verified by experiments [13]. To make up for the shortcomings of the existing grammar error correction framework, Li et al. proposed a new strategy. This strategy combined the traditional "sequence-to-edit" and "sequence-to-sequence" frameworks for syntax error detection and correction, respectively. The proposed model used consistency learning to enhance the consistency of predictions between different blocks. This method demonstrated its effectiveness and robustness in grammar error correction, which had good potential [14]. Acheampong and Tian proposed a grammar error correction model enhanced by neural cascade architecture and different techniques, aiming at the disadvantage of the relatively high computational cost of neural network-based grammar error correction models. The proposed model showed excellent syntax error correction performance similar to that of high-configuration machine translation systems in low-resource machine translation systems [15].

To sum up, scholars all over the world have devoted themselves to the study of English grammar correction models and have made some achievements. However, the above models are more or less limited in complex operation and have poor generalization ability, which makes it difficult to adapt to the variable syntax error recognition. Therefore, this study proposes an English grammar intelligent error correction technology based on the Cn-gram language model, aiming at better English grammar error correction.

3 Research on intelligent error correction technology in English grammar

3.1 Construction of error correction model based on n-gram

The situation of English grammar errors is complicated, and the intelligent error correction technology designed by the research is mainly aimed at article errors, preposition errors, noun errors, and verb errors in English grammar [16]. If n words form a sentence S, then the string of S can be expressed as $S = w_1w_2w_3 \ldots w_{n-1}w_n$, and the probability of string S is $P(S)$

$$P(S) = p(w_1)p(w_2|w_1)p(w_3|w_2|w_1)\ldots p(w_n|w_1w_2 \ldots w_{n-1}). \tag{1}$$

In equation (1), the probability of occurrence of word w_i is jointly determined by word $w_1w_2w_3 \ldots w_{i-1}$ before word w_i, so these words are the preamble of word i [17]. When the vocabulary set size is L and the length of the preamble is $i - 1$, the word at i will have L^{i-1} different preambles, which is too large to calculate. Map $w_1w_2w_3 \ldots w_{i-1}$ above to the equivalence class $E(w_1w_2w_3 \ldots w_{i-1})$ with certain rules, so that $P(w_1w_2w_3 \cdots w_{i-1}) = P[w_i|(w_1w_2w_3 \ldots w_{i-1})]$ reduces the number of free parameters.

Partition equivalence class method: Two preambles $w_{i-n+2} \ldots w_{i-1}w_i$, $w_{k-n+2} \ldots w_{k-1}w_k$ can be mapped to the same equivalence class if and only if the nearest $n - 1$ word is the same [18]. Then, $E(w_1w_2w_3 \ldots w_{i-n+2} \ldots w_{i-1}w_i) = E(v_1v_2v_3 \ldots v_{k-n+2} \ldots v_{k-1}v_k)$, if and only if $(w_{i-n+2} \ldots w_{i-1}w_i) = (v_{k-n+2} \ldots v_{k-1}v_k)$. When a language model meets the above conditions, it is called an n-gram language model [19]. The related phrases before and after a given word can be obtained by window movement in a sentence, and its mathematical description is

$$MW_{i,k}(w) = \{W_{i-j} \ldots W_{i-j+(k+1)}, j = 0, k - 1\}. \tag{2}$$

In equation (2), i indicates the position of a given word in a sentence; k is the window size; w refers to the word in position i.

Table 1 provides examples of moving window values. n-gram fragments have different lengths when window sizes are different. An error candidate set is a set composed of two parts, such as an error in a sentence and the corresponding candidate modification answer [20]. The zero probability problem occurring in n-gram is solved by linear interpolation smoothing technique; that is, the higher-order model is combined with the lower-order model in a linear way, and the higher-order n-gram model is linearly interpolated by the lower-order n-gram model, so as to estimate the probability of the higher-order model [21]

Table 1: Examples of window values

Window size	n-gram fragment
2	Actual situation
	Situation of
3	The actual situation
	Actual situation of
	Situation of building
4	Understand the actual situation
	The actual situation of
	Actual situation of building
	Situation of building energy
5	To understand the actual situation
	Understand the actual situation of
	The actual situation of building
	Actual situation of building energy
	Situation of building energy consumption

$$\hat{P}(w_n|w_{n-1}\dots w_{n-i+1}) = \lambda_1 P(w_n|w_{n-1}\dots w_{n-i+1}) + \lambda_2 P(w_n|w_{n-1}\dots w_{n-i+2}) + \dots + \lambda_i P(w_n). \tag{3}$$

In equation (3), $\sum_i \lambda_i = 1$ and λ_i are calculated by the expectation maximization (EM) algorithm. First, training data and hold-out data are determined, the initial language model is built from the training data, and the initial λ_i is defined. The EM algorithm is used for iterative optimization λ_i to ensure that the probability of Held out data is maximized [22]

$$\log P[|w_1\dots w_n|\mathrm{Max}(\lambda_1\dots\lambda_k)] = \sum_i \log P_{\mathrm{Max}(\lambda_1\dots\lambda_k)}(w_i|w_{i-1}). \tag{4}$$

Equation (4) is a probability calculation equation for Held out data. The n-gram combination model (Combine N-gram, Cn-gram) of (An = 1, 2, 3, 4, 5) is obtained by the linear interpolation smoothing algorithm. The probability calculation equation of this model is shown in equation (5)

$$P(S) = \prod_{i=1} \lambda_N P(w_i|w_{i-N+1}^{i-1}). \tag{5}$$

λ_N in equation (5) corresponds to λ_i in equation (3), both of which are calculated by the EM algorithm. The moving window is combined with the error candidate set to design an English syntax error correction technique based on the error candidate set. The results of error correction technology are evaluated by precision, recall, and $F1$ values

$$\mathrm{Precision} = \frac{N_{\mathrm{correct}}}{N_{\mathrm{predicted}}}, \tag{6}$$

$$\mathrm{Recall} = \frac{N_{\mathrm{correct}}}{N_{\mathrm{target}}}, \tag{7}$$

$$\mathrm{F1} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{8}$$

$N_{correct}$ in equations (6)–(8) refers to the number of grammatical corrections that are correct; $N_{predicted}$ refers to the number of grammatical correction errors; and N_{target} refers to the number of errors originally present in the grammar. The syntactic error correction model based on the Cn-gram model can effectively locate the possible errors in sentences and has shown good performance for processing simple sentences. However, when dealing with long sentences, the model's precision, recall rate, and $F1$ value all decrease. Therefore, syntactic analysis is introduced in this study to make the model better able to deal with errors in English long sentences.

3.2 Construction of grammar error correction model based on Cn-gram and syntactic analysis

In this section, a Parsing and Cn-gram Grammatical Error Correction (PCGEC) is proposed, which combines Cn-gram and syntactic analysis. In essence, syntactic structure is a process of interrelation between words [23]. A dependency ties two words together, one being a core word and the other a modifier. Dependency can be used to describe the grammatical relationship between two words and can be further subdivided into various types [24]. The entered sentence corresponds to a dependency graph $G = (V, A)$, which is a directed multi-graph. Wherein, $V = \{0, 1,...,n\}$, each node i corresponds to an input word w_1. Dependency tree $T = (V, A)$ is a dependency graph, whose specific expression is

$$d = \{(A, B, C) : 0 \le A \le n, 1 \le B \le n, C \in L\}. \tag{9}$$

In equation (9), A, B, C represents a dependency arc from w_A to w_B; w_C represents core words; w_B represents a modifier; C is the type of relationship of the arc of dependence; and L is a set of dependency arc relationship types. The following assumptions exist in the dependency tree corresponding to the sentence: (1) The interaction and correlation between dependency arcs only occur in some specific structures (sub-trees) [25]. (2) The other arcs of dependence are independent of each other. Then, the fractional value of a dependency tree can be decomposed into the sum of the fractional values of several sub-trees, and the calculation equation is as follows:

$$\text{Score}(x, p) = w \cdot f(x, d) = \sum_{p \subseteq d} \text{Score}_{\text{subtree}}(x, p). \tag{10}$$

In equation (10), x represents the sentence; p represents a subtree permitted by an independent hypothesis; d represents the feature vector of a sentence; p contains one or more arcs of dependency in d; w is the characteristic weight vector; $f(x, d)$ is the aggregated syntactic feature vector corresponding to (x, d); and $\text{Score}(x, p)$ denotes the score value contributed by subtree p. The syntactic analysis model proposed in this study is introduced into the three-seed tree structure, which consists of one dependency arc, two sibling dependency arcs adjacent in the same direction, and two dependency arcs of the grandparent relationship [26]. The calculation equation of the dependency tree is

$$\begin{aligned} \text{Score}(x) = & \sum_{\{h,m,l\} \subseteq d} \text{Score}_{\text{dep}}(x, h, m, l) + \sum_{\{(h,m),(h,s)\} \subseteq d} \text{Score}_{\text{sib}}(x, h, s, m) \\ & + \sum_{\{(h,m,l),(m,g)\} \subseteq d} \text{Score}_{\text{grd}}(x, h, m, l, g). \end{aligned} \tag{11}$$

In equation (11), $\text{Score}_{\text{dep}}$, $\text{Score}_{\text{sib}}$, and $\text{Score}_{\text{grd}}$, respectively, represent the scores corresponding to the three seed trees, and their specific expressions are

$$\begin{aligned} \text{Score}_{\text{dep}}(x, h, m, l) &= W_{\text{dep}} \cdot f_{\text{dep}}(x, h, m, l) \\ \text{Score}_{\text{sib}}(x, h, s, m) &= W_{\text{sib}} \cdot f_{\text{sib}}(x, h, s, m) \\ \text{Score}_{\text{grd}}(x, h, m, l, g) &= W_{\text{grd}} \cdot f_{\text{grd}}(x, h, m, l, g). \end{aligned} \tag{12}$$

In equation (12), f_{dep}, f_{sib}, and f_{grd}, respectively, represent the eigenvectors corresponding to the three seed trees. W_{dep}, W_{sib}, and W_{grd} represent corresponding feature weight vectors, respectively. The flow of the syntax error correction algorithm based on syntax analysis is shown in Figure 1.

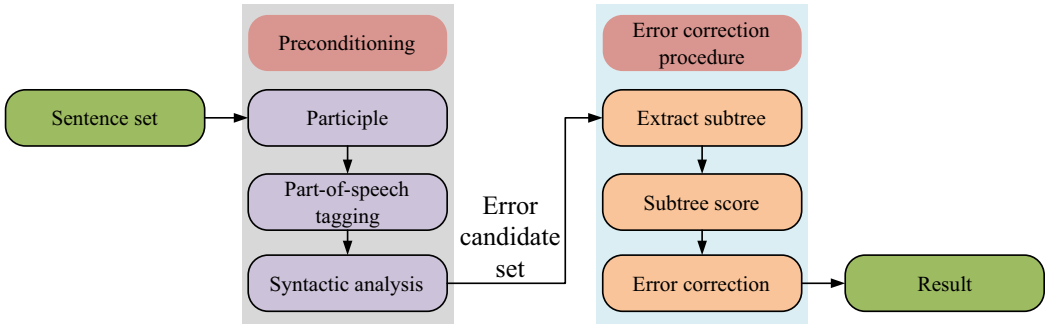

Figure 1: Syntax error correction algorithm flow based on syntax analysis.

In Figure 1, the first step is to present the set of sentences to be corrected and perform word segmentation on the sentences, that is, to decompose an input text stream into words, phrases, symbols, or some meaningful elements. Afterward, the data obtained from word segmentation are annotated with part of speech, which determines whether each word is a noun, verb, adjective, or other parts of speech. Syntactic analysis mainly refers to performing dependency syntactic analysis on the data annotated with part of speech to obtain the dependency tree corresponding to the sentence. Afterward, based on the instances in the incorrect candidate set C, first-order and second-order sub-trees are extracted from the complete dependency tree. The frequency of sub-trees is calculated based on the tree library and converted into scores. The error candidate set is taken with the highest score in the sub-tree, the error item is replaced, and the sentence is outputted after the completion of error correction.

The syntax error correction model based on parsing has the following shortcomings: (1) It relies too much on a dependency tree. Currently, the dependency tree database is not enough, and the workload required to establish the dependency tree database is large, and the corresponding relationship of each dependency tree also has different differences. (2) The local error rate and recall rate in sentences are not too high. (3) The precision of syntactic analysis has a great influence on error correction performance. Therefore, in this study, the n-gram algorithm and syntactic analysis model are integrated to improve the error correction performance of the overall model. This study focuses on the automatic correction of English compound sentence grammar. Currently, there has been a lot of research on compound sentences, such as the hierarchical study of long and difficult sentences and the study of related words. When building a compound sentence model, we should give full play to the advantages of the Cn-gram word model and avoid the shortcomings of the Cn-gram word model for long-distance constraints. This article adopts a new way of thinking, that is, by splitting the compound sentences, analyzing the syntax, and then combining the results. First, the sub-sentences of compound sentences are classified according to semantic relations, and the related words and guiding words are used as the basis of classification. On this basis, Cn-gram is used to model each segment independently. Finally, the model is combined, that is, the final complex sentence modeling result. Among them, the connective words and guiding words in the compound sentences serve as the link to connect each clause, so that they can complete the Cn-gram model independently. The probability calculation process of the combined model is

$$\begin{aligned} P_1(W_l|W_{i-1}) &= \frac{\text{count}(W_{i-1}W_i)}{\text{count}(W_{i-1}W)} \\ P_2(W_l|W_{i+1}) &= \frac{\text{count}(W_lW_{i+1})}{\text{count}(WW_{i+1})} \\ P_3(W_l|W_{i-1}, W_{l+1}) &= \frac{\text{count}(W_{i-1}W_lW_{i+1})}{\text{count}(W_{i-1}WW_{i+1})}. \end{aligned} \tag{13}$$

In equation (13), P_1 represents the probability that $W_i(w)$ occurs after knowing the previous word of the specified word $W_i(w)$; P_2 represents the probability that a word before $W_i(w)$ occurs when the word after $W_i(w)$ is known; and P_3 represents the probability of $W_i(w)$ appearing in the middle of a given word $W_i(w)$ with one word before and one word after it. Assuming that the words before and after the candidate words have the same influence on the candidate words, the score calculation equation is

$$\text{Score}(W_i) = \lambda(P_1 + P_2) + \mu P_3, 2\lambda + \mu = 1. \tag{14}$$

In equation (14), λ, μ are the values determined for a large number of training through the training set. The probability calculation equation of the entire complex sentence is

$$P(S) = \prod_{i=1} \alpha_i P(S_t). \tag{15}$$

In equation (15), S_i represents a clause or clause and α_i denotes the weight representing the probability of each clause is trained in the training set. In a pun complex sentence, the conditional possibility of two connective words is α_i. For single related words and clauses, after synthesizing the components in each clause, select α_i as the conditional probability of the sentence components of the guiding words or related words and clauses. The overall operation flow of the error correction algorithm based on PCGEC is shown in Figure 2.

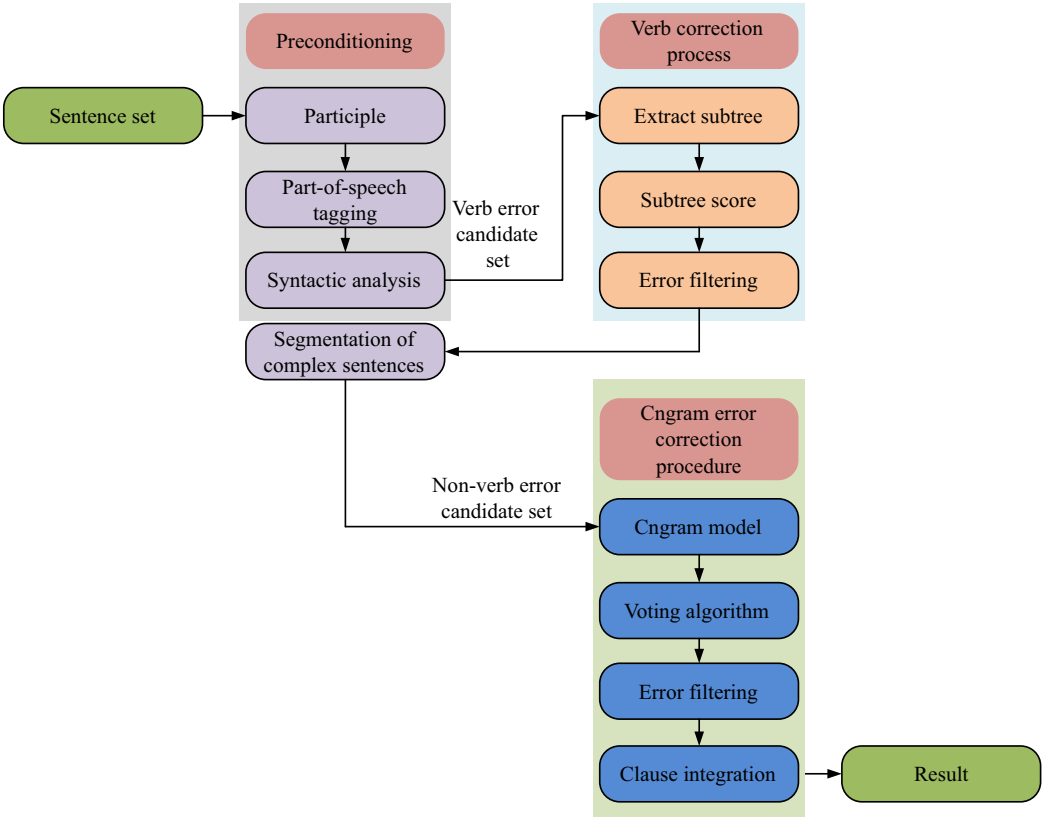

Figure 2: Overall operation flow of error correction algorithm based on PCGEC.

In Figure 2, first, the corrected sentences are segmented based on the set of sentences to be corrected, and the data obtained from the segmentation are annotated with part of speech. Afterward, dependency syntactic analysis is performed on the data annotated with part of speech to obtain the dependency tree corresponding to the sentence. Based on instances in the incorrect candidate set, first-order and second-order sub-trees are extracted from a complete dependency tree. Sub-tree frequencies are calculated based on the tree library and converted into scores. The error item is replaced with the instance in the error candidate set that corresponds

to the sub-tree with the highest score. Errors in the N-gram error correction process are corrected in the order of nouns, articles, and prepositions, and a voting strategy is used to rate the N-gram. The probability of the corrected sentence is calculated in the N-gram model, and the sentence with a higher probability is selected to output the corrected sentence.

4 Application effect of grammar intelligent error correction technology

4.1 Performance verification of syntax error correction algorithm based on Cn-gram

The training data and test data as shown in Table 2 are studied and selected, and the practical application effect of the designed syntax error correction technology is verified by the Windows 7 operating system. In Table 2, error types include article error, preposition error, noun error, subject–verb agreement error, and verb form error. According to the length of sentences in the test data, it is divided into two parts with the same number, the Long part and the Short part.

Table 2: Details of training data and test data

Error type	Training set		Test set	
	Number	Proportion (%)	Number	Proportion (%)
Article	6,655	14.7	692	19.7
Preposition	2,411	5.3	314	8.9
Noun	3,780	8.4	397	11.3
Subject predicate agreement	1,451	3.2	124	3.5
Verb form	1,533	3.4	129	3.7
Total number of errors	15,831	35.1	1645	46.8
Total	45,123	100	3516	100

In the n-gram model, when n is 2, the word w_i in the position i is only affected by the previous historical word w_{i-1}, which is denoted as the Bi-gram model. In the n-gram model, when n is 3, the word w_i in the position i is only affected by the first two historical words w_{i-1} and w_{i-2}, which is recorded as the Tri-gram model. The comparison experiment of the Bi-gram model, Tri-gram model, and n-gram model-based error correction algorithm Cn-gram is set up.

In Figure 3, the correct rate of Bi-gram and Tri-gram is 0.2935 and 0.2899, respectively, both of which are lower than that of Cn-gram (0.3011). The recall of Bi-gram and Tri-gram is 0.4342 and 0.4401, both lower than Cn-gram (0.4832). The $F1$ value of Cn-gram is 0.3710, 2.08% higher than that of Bi-gram and 2.15% higher than that of Tri-gram. Figure 3(b) shows that in the correction of prepositions in English grammar, the correct rate of Bi-gram is 0.0482, and the correct rate of Tri-gram is 0.0476, both lower than that of Cn-gram (0.0567). The recall of Bi-gram is 0.1690, slightly higher than Cn-gram (0.1678); The recall of Tri-gram is 0.1589. The $F1$ value of Cn-gram is 0.0750, which is 0.97% higher than Bi-gram and 1.14% higher than Tri-gram. To sum up, Cn-gram has good error correction performance for articles, that is, the Cn-gram model is suitable for the local description of sentences.

From Figure 4, the correct rate of Cn-gram is 0.2597, higher than that of Bi-gram (0.2409) and Tri-gram (0.2403), respectively, increasing by 1.88 and 1.94%. The recall of Cn-gram is 0.5088, higher than Bi-gram (0.4538). The recall of Tri-gram is 0.4608, which increased by 5.5 and 4.8%, respectively. The $F1$ value of Cn-

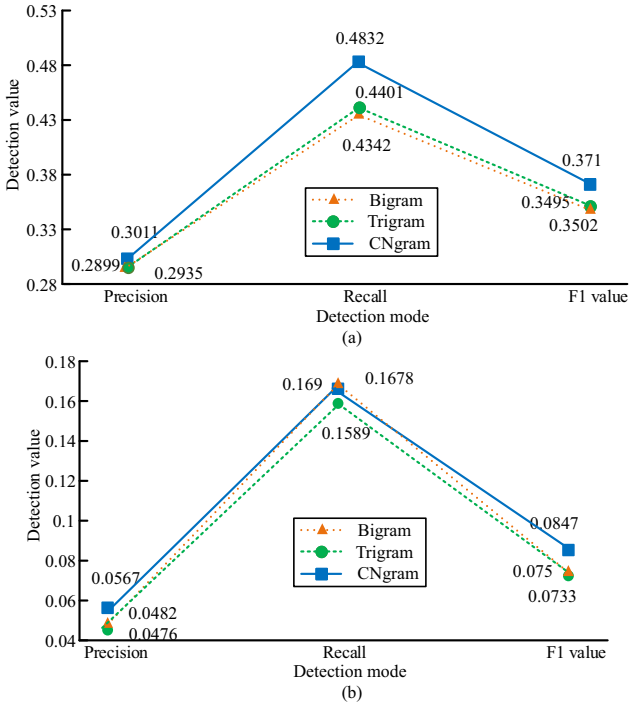

Figure 3: A comparative study on the correction results of (a) article errors and (b) preposition errors.

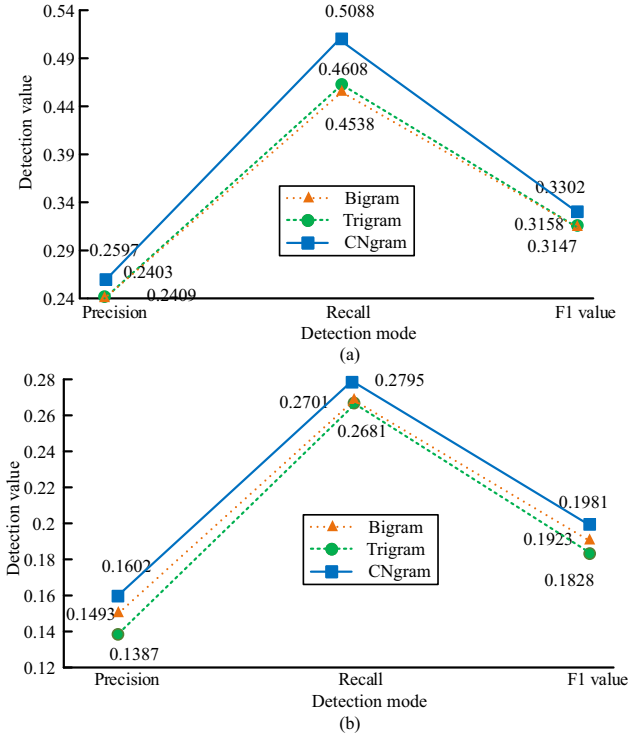

Figure 4: A comparison of the results of correcting (a) noun errors and (b) verb errors.

gram is 0.3302, higher than Bi-gram (0.3147) and Tri-gram (0.3158), respectively, higher by 1.55 and 1.44%. Figure 4(b) shows that in verb error correction of English grammar, the correct rate of Bi-gram is 0.1493, and the correct rate of Tri-gram is 0.1387, both lower than the correct rate of Cn-gram is 0.1602. The recall of Bi-

gram and Tri-gram is 0.2701 and 0.2681, both lower than Cn-gram (0.2795). The *F*1 value of Cn-gram is 0.1981, which is 0.58% higher than Bi-gram and 1.73% higher than Tri-gram. The above results show that the Cn-gram algorithm has better error correction performance for nouns than for verbs. Short test set and Long test set are processed by the Cn-gram model, and the influence of sentence length on the error correction performance of the Cn-gram model is compared. The specific results are shown in Figure 5.

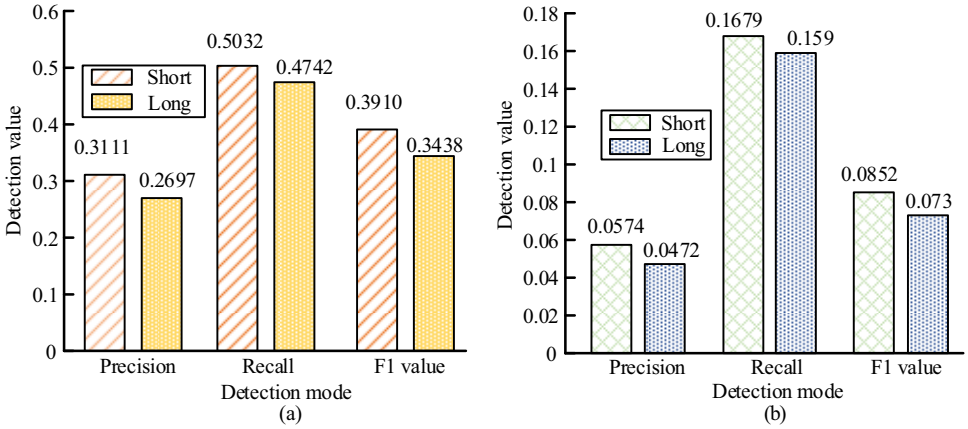

Figure 5: A comparison of the (a) correction results of article errors and (b) preposition errors under different sentence lengths.

From Figure 5, when correcting English grammar articles in the test set with a Long sentence length (Long), the precision rate of the Cn-gram model is 0.2697, the recall rate is 0.4742, and the *F*1 value is 0.3438. The precision of the Cn-gram model is 0.3111, the recall rate is 0.5032, and the *F*1 value is 0.3910 when correcting the articles of English grammar in a Short test set with short sentence length. Figure 5(b) shows that when correcting prepositions of English grammar in the Long test set, the precision rate of the Cn-gram model is 0.0472, the recall rate is 0.1590, and the *F*1 value is 0.0730. When correcting prepositions of English grammar in the Short test set, the precision rate of the Cn-gram model is 0.0574, the recall rate is 0.1679, and the *F*1 value is 0.0852. These results show that the Cn-gram model has better performance in article and preposition correction for English with short sentence length.

Figure 6 shows that when correcting nouns in English grammar in the Long test set, the precision rate of the Cn-gram model is 0.2315, the recall rate is 0.4533, and the *F*1 value is 0.3064. When correcting nouns in English grammar in the Short test set, the precision rate of the Cn-gram model is 0.2757, the recall rate is 0.5268,

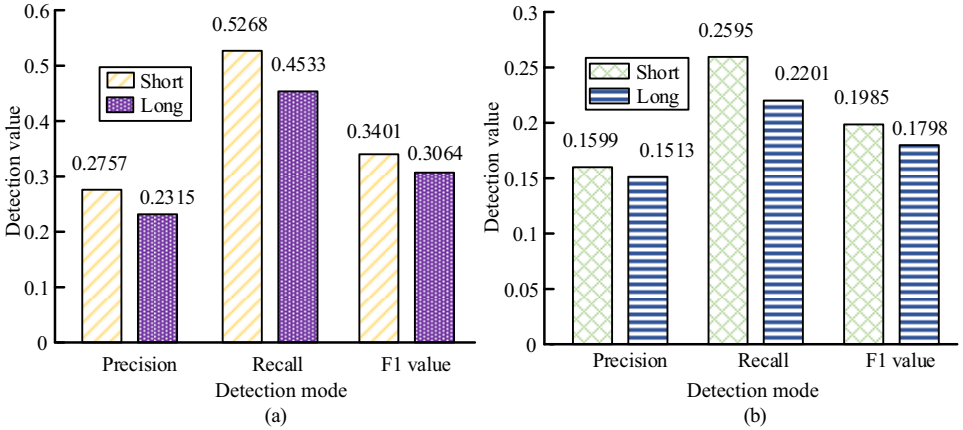

Figure 6: A comparison of the correction results of (a) noun errors and (b) verb errors under different sentence lengths.

and the *F*1 value is 0.3401. It can be seen that in noun error correction, the precision rate of the Short test set is 4.42% higher, the recall rate is 7.35% higher, and the *F*1 value is 3.37% higher than the results of the Long test set. Figure 6(b) shows that when correcting verbs of English grammar in the Long test set, the precision rate of the Cn-gram model is 0.1513, the recall rate is 0.2201, and the *F*1 value is 0.1798. When correcting English grammar verbs in the Short test set, the precision rate of the Cn-gram model is 0.1599, the recall rate is 0.2595, and the *F*1 value is 0.1985. It can be seen that in verb error correction, the precision of Short test set is 0.86% higher, the recall rate is 3.94% higher, and the *F*1 value is 1.87% higher than the results of Long test set.

4.2 Performance verification of the syntax error correction model based on the PCGEC model

Next, the performance of the syntactic error correction model based on the PCGEC model is verified. The experimental environment and training data sets used in this study are the same as those used in the previous section. In Figure 7, for article errors, PCGEC's precision, recall rate, and *F*1 value are higher than Cn-gram and syntactic analysis model (SAM). The precision of PCGEC is 2.33 and 5.1% higher than that of Cn-gram and SAM, respectively. The recall rate is higher by 0.69 and 3.73%; *F*1 values are higher by 1.29 and 6.88%. For preposition errors, PCGEC is 2.07 and 3.27% more accurate than Cn-gram and SAM. The recall rate is higher by 1.68 and 0.02%, respectively. *F*1 values are 2.11 and 3.33% higher.

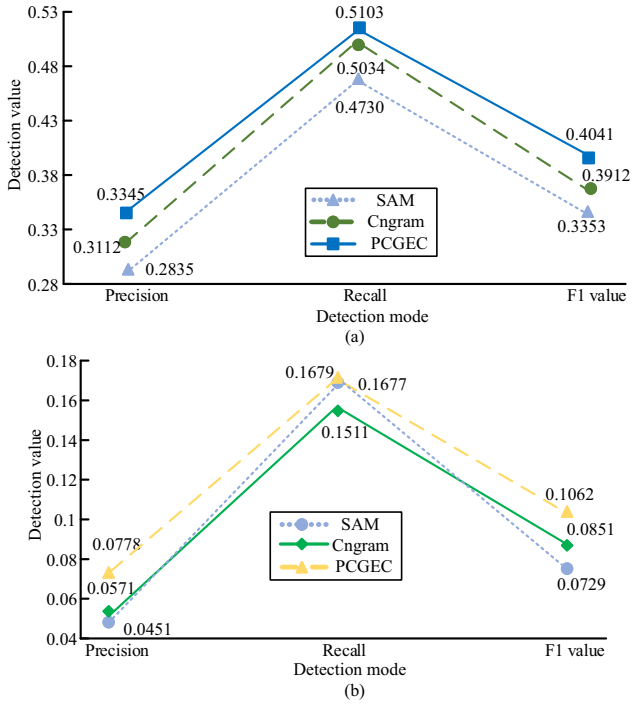

Figure 7: A comparative study on the correction results of (a) article errors and (b) preposition errors.

In Figure 8, for noun errors, PCGEC has higher precision, recall rate, and *F*1 values than Cn-gram and SAM. The precision of PCGEC is 2.17 and 4.58% higher than that of Cn-gram and SAM, respectively. The recall rate is higher by 0.21 and 12.03%, respectively. *F*1 values are 6.74 and 7.31% higher, respectively. For verb errors, PCGEC's precision, recall rate, and *F*1 value are also higher than Cn-gram and SAM. The precision of PCGEC is 11.07 and 4.03% higher than that of Cn-gram and SAM, respectively. The recall rate is 2.18 and 6.13% higher, respectively. *F*1 values are 7.73 and 5.07% higher, respectively.

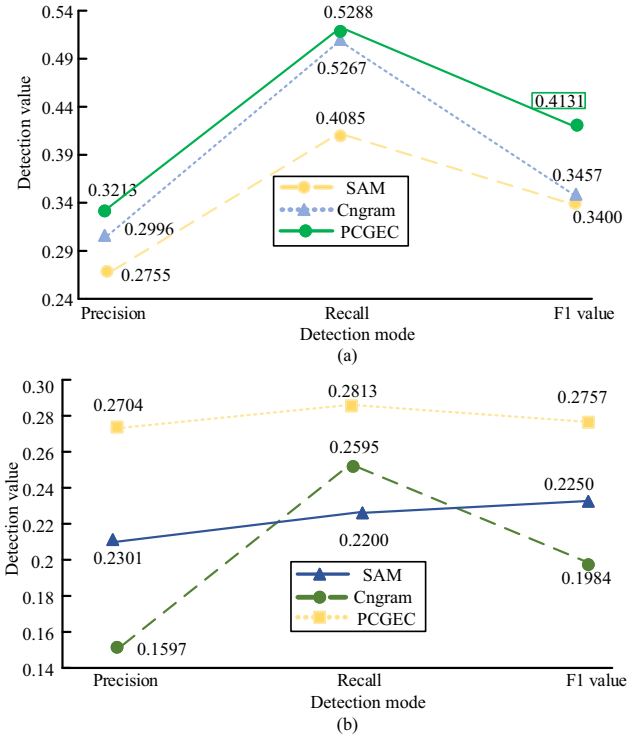

Figure 8: Comparison of the results of correcting (a) noun errors and (b) verb errors.

Figure 9 shows the comparison results of precision, recall rate, and *F*1 values of the three models on Short and Long data sets. In the Short data set, the precision of PCGEC is 16.11 and 5.11% higher than that of Cn-gram and SAM for subject-verb agreement errors. Recall rates are 8.26 and 7.19% higher; *F*1 values are 13.15 and 6.15% higher. On the Long data set, the precision of PCGEC is 21.15 and 5.05% higher than that of Cn-gram and SAM for subject–verb agreement errors, respectively. The recall rate is higher by 13.15 and 5.79%; *F*1 values are

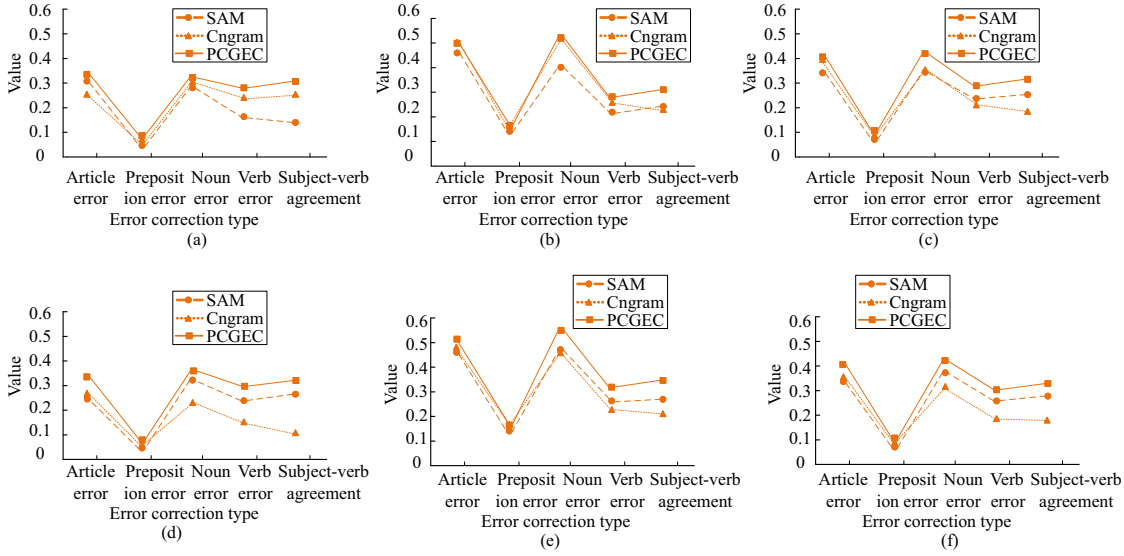

Figure 9: Comparison results of precision, recall rate, and *F*1 values of the three models on Short and Long data sets. (a) The precision of three models on the Short dataset. (b) The recall of three models on the Short dataset. (c) The *F*1 of three models on the Short dataset. (d) The precision of three models on the Long dataset. (e) The recall of three models on the Long dataset. (f) The *F*1 of three models on the Long dataset.

higher by 15.14 and 5.31%. Therefore, PCGEC performs better than Cn-gram and SAM. Among them, the difference in subject–predicate agreement is obvious. The error-correcting effect of nouns and articles is very similar. The second is the verb and subject–verb agreement, and the last is the preposition. There may be the following reasons: First, there is little difference between articles and nouns, and errors in articles are mainly manifested as lexical errors, while errors in nouns are mainly manifested as syntactic errors. Second, the form of the verb changes with the context, mainly due to syntactic errors. Third, subject–verb agreement error is a kind of grammatical error. Fourth, there are many types of prepositions, and the preposition phrases are very complicated.

Figure 10 shows the comparison results of precision, recall rate, and *F*1 value of the three methods on the complete test set. Among them, the precision of PCGEC is 19.10 and 5.41% higher than that of Cn-gram and SAM for subject–verb agreement errors, respectively. The recall rate was 9.55 and 10.77% higher, respectively; *F*1 values were higher by 12.65 and 1059%, respectively. Through the above experimental analysis, the following conclusions can be drawn: (1) The Cn-gram model is very powerful for the local description of sentences, and it is very effective for local sentence errors (lexical errors), but it is not effective for syntactic errors. (2) The SAM method can analyze the structure of the sentence and the relationship between various elements in the sentence, so it has a significant performance in grammar errors, but is weak in vocabulary. (3) The combination of these two methods can effectively improve the correction effect of vocabulary and grammar errors meanwhile.

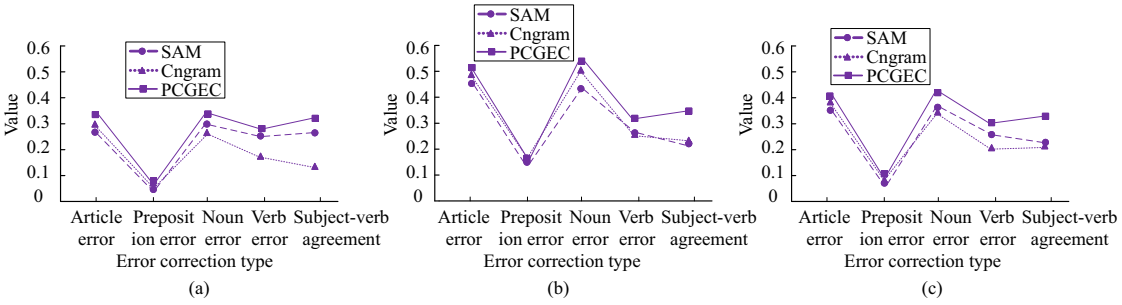

Figure 10: Comparison results of (a) precision, (b) recall rate, and (c) *F*1 value of the three methods on the complete test set.

This study used word move's distance (WMD) and improved move's distance (IMD) to measure the measurement of perturbed samples. The larger the WMD distance, the smaller the similarity. On the contrary, the deviation in word meaning is relatively small. IMD mainly considers the distance of movement between pinyin and determines the degree of semantic deviation. Figure 11 shows the line graph of the test results.

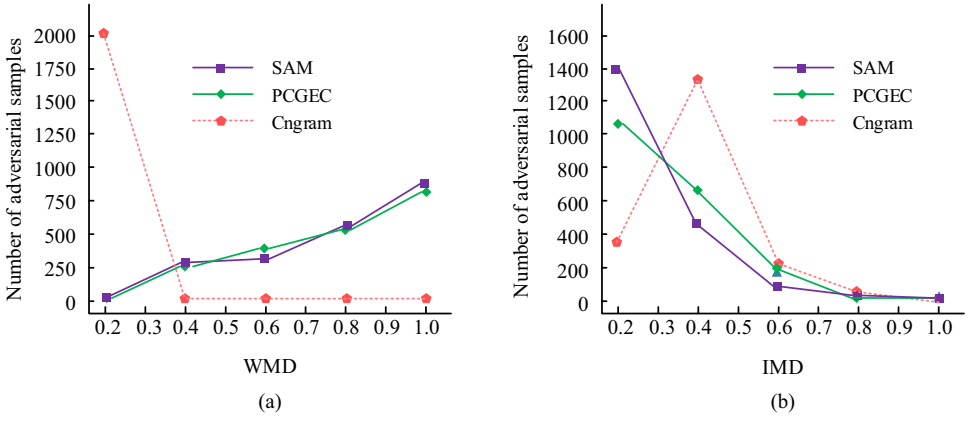

Figure 11: Generated adversarial sample test results. (a) WMD distribution of adversarial sample size generated by different methods. (b) IMD distribution of adversarial sample size generated by different methods.

When the sample size reaches 2,000, WMD is used to measure the generated sample size, and the obtained sample sizes are all between 0 and 0.2, while other methods are all between 0.4 and 0.6. When calculating IMD offset, the proposed method also has better performance than other methods.

5 Conclusion

With the expansion of the application of the Internet, the number of electronic texts has increased sharply, and the importance of automatic error correction technology for electronic text grammar has increased. To realize intelligent error correction of English grammar, this article proposes a grammar intelligent error correction model (PCGEC model) based on the n-gram algorithm and syntax analysis method and verifies the practical application effect of the grammar intelligent error correction technology through experiments. First, the Cn-gram algorithm was verified. In article correction, the precision of the Cn-gram model was 0.76 and 1.12% higher than Bi-gram and Tri-gram, respectively. The recall rate was higher by 4.9 and 4.31%, respectively. *F*1 values were higher by 2.08 and 2.15%, respectively. In terms of noun error correction, compared with the Bi-gram error correction method and Tri-gram, the precision of the Cn-gram model was 1.88 and 1.94% higher, and the *F*1 value was 1.55 and 1.44% higher. In verb error correction, the precision of the Cn-gram model was 1.09 and 2.15% higher, and the *F*1 value was 0.58 and 1.73% higher. The error correction performance of the Cn-gram model for the Short test set was due to the error correction performance for the Long test set. The performance verification results of the PCGEC model showed that the precision of the PCGEC model was 19.10 and 5.41% higher than that of Cn-gram and SAM in the complete test set, respectively. The recall rate was 9.55 and 10.77% higher, respectively; *F*1 values were higher by 12.65 and 10.59%, respectively. Although the research has made some achievements, there are still shortcomings. For example, the corpus used in the research covers too single a field and has certain limitations. In the future, more fields of corpus will be introduced to continue to optimize intelligent error correction technology.

Funding information: This project was supported by Key Scientific and Technological Projects in Henan Province (Grant no. 23210222002).

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and consented to its submission to the journal, reviewed all the results and approved the final version of the manuscript. Fan Xiao gather data and wrote original draft preparation. Shehui Yin reviewed the manuscript and provided financial support.

Conflict of interest: Authors state no conflict of interest.

Data availability statement: The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

[1] Wang Y, Wang Y, Dang K, Liu J, Liu Z. A comprehensive survey of grammatical error correction. ACM Trans Intell Syst Technol (TIST). 2021;12(5):1–51.
[2] Fitria TN. Grammarly as AI-powered English writing assistant: Students' alternative for writing English. Metathesis: J Engl Lang Lit Teach. 2021;5(1):65–78.
[3] Fitria TN. Artificial intelligence (AI) technology in OpenAI ChatGPT application: A review of ChatGPT in writing English essay. J Engl Lang Teach. 2023;12(1):44–58.
[4] Aouragh SL, Yousfi A, Laaroussi S. A new estimate of the n-gram language model. Procedia Computer Sci. 2021;189:211–5.
[5] Wang H, He J, Zhang X, Liu S. A short text classification method based on N-Gram and CNN. Chin J Electron. 2020;29(2):248–54.

[6] Hu L, Tang Y, Wu X, Zeng J. Considering optimization of English grammar error correction based on neural network. Neural Comput Appl. 2022;34:3323–35.
[7] Huang X, Zou D, Cheng G, Chen X, Xie H. Trends, research issues and applications of artificial intelligence in language education. Educ Technol Soc. 2023;26(1):112–31.
[8] Duan R, Wang Y, Qin H. Artificial intelligence speech recognition model for correcting spoken English teaching. J Intell Fuzzy Syst. 2021;40(2):3513–24.
[9] Zhou S, Liu W. English grammar error correction algorithm based on classification model. Complexity. 2021;2021:1–11.
[10] Park C, Yang Y, Lee C, Lim H. Comparison of the evaluation metrics for neural grammatical error correction with overcorrection. IEEE Access. 2020;8:106264–72.
[11] He Z. English grammar error detection using recurrent neural networks. Sci Program. 2021;2021:1–8.
[12] Wang X. Translation correction of English phrases based on optimized GLR algorithm. J Intell Syst. 2021;30(1):868–80.
[13] Zhao Z, Wang H. Maskgec: Improving neural grammatical error correction via dynamic masking. Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34, No. 1, 2020. p. 1226–33.
[14] Li Y, Liu X, Wang S, Gong P, Wong D, Gao Y, et al. Templategec: Improving grammatical error correction with detection template. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Vol. 1, 2023. p. 6878–92.
[15] Acheampong K, Tian W. Toward perfect neural cascading architecture for grammatical error correction. Appl Intell. 2021;51:3775–88.
[16] Wu LQ, Wu Y, Zhang XY. L2 learner cognitive psychological factors about artificial intelligence writing corrective feedback. Engl Lang Teach. 2021;14(10):70–83.
[17] Kiyono S, Suzuki J, Mizumoto T, Inui K. Massive exploration of pseudo data for grammatical error correction. IEEE/ACM Trans Audio Speech Lang Process. 2020;28:2134–45.
[18] Anbukkarasi S, Varadhaganapathy S. Neural network-based error handler in natural language processing. Neural Comput Appl. 2022;34(23):20629–38.
[19] Chen Y, Wang X, Du X. Diagnostic evaluation model of English learning based on machine learning. J Intell Fuzzy Syst. 2021;40(2):2169–79.
[20] Zhang G. A study of grammar analysis in English teaching with deep learning algorithm. Int J Emerg Technol Learn (iJET). 2020;15(18):20–30.
[21] Tang J, Qian K, Wang N, Hu X. Exploring language learning and corrective feedback in an eTandem project. J China Computer-Assisted Lang Learn. 2021;1(1):110–44.
[22] Kholis A. Elsa speak app: automatic speech recognition (ASR) for supplementing English pronunciation skills. Pedagogy: J Engl Lang Teach. 2021;9(1):1–14.
[23] Mashoor BBN, Abdullah ATH. Error analysis of spoken English language among Jordanian secondary school students. Int J Educ Res. 2020;8(5):75–82.
[24] Zhang H, Qiu D, Wu R, Ji D, Li G, Niu Z, et al. Novel model to integrate word embeddings and syntactic trees for automatic caption generation from images. Soft Comput: Fusion Found Methodol Appl. 2020;24(2):1377–97.
[25] Syamala M, Nalini NJ. A speech-based sentiment analysis using combined deep learning and language model on real-time product review. Int J Eng Trends Technol. 2021;69(1):172–8.
[26] Martinez A, Sudoh K, Matsumoto Y. Sub-subword n-gram features for subword-level neural machine translation. J Nat Lang Process. 2021;28(1):82–103.