WILEY | Hindawi

Retraction

Retracted: Application of Grammar Error Detection Method for English Composition Based on Machine Learning

Security and Communication Networks

Security and Communication Networks has retracted the article titled "Application of Grammar Error Detection Method for English Composition Based on Machine Learning" [1] due to concerns that the peer review process has been compromised.

Following an investigation conducted by the Hindawi Research Integrity team [2], significant concerns were identified with the peer reviewers assigned to this article; the investigation has concluded that the peer review process was compromised. We therefore can no longer trust the peer review process, and the article is being retracted with the agreement of the Chief Editor.

The authors do not agree to the retraction.

References

[1] H. He and D. Maia, "Application of Grammar Error Detection Method for English Composition Based on Machine Learning," *Security and Communication Networks*, vol. 2022, Article ID 7957365, 9 pages, 2022.

[2] L. Ferguson, "Advancing Research Integrity Collaboratively and with Vigour," 2022, https://www.hindawi.com/post/advancing-research-integrity-collaboratively-and-vigour/.

WILEY | Hindawi

Research Article

Application of Grammar Error Detection Method for English Composition Based on Machine Learning

Haiyin He[1] and Darchia Maia [2]

[1]*Qinghai University, Xining, Qinghai 810016, China*
[2]*School of Science and Technology, The University of Georgia, Tbilisi 017, GA, USA*

Correspondence should be addressed to Darchia Maia; dr.darchia@stu.zf.edu.ge

With the development of grammar-checking technology and algorithms, the grammar-checking system has been widely used in various fields. This paper designs and implements a grammar-checking system for English composition. The grammar-checking system adopts a multimodule design. The grammar-checking system is composed of a multilayer rule error-correcting module and a machine learning error-correcting module. This study aims to build a machine learning algorithm model that can detect English grammar errors by analysing and comparing different algorithm models currently applied in the field of education and then apply the trained model to the English composition grammar detection system. The results show that the system can save a lot of time and labor cost of manual marking, liberate teachers from heavy and repeated evaluation activities, and put more time and energy on teaching. At the same time, it can provide learners with more objective and timely feedback so that learners can intuitively and clearly know that they are prone to make grammatical mistakes in the process of English learning. It plays a certain assisting and guiding role in English learners' autonomous learning.

1. Introduction

Learners are often expected to create English compositions as part of the English teaching process in order to demonstrate their command of the language. English learners may increase their English level by producing English compositions and gaining a better understanding of the language. Simultaneously, English instructors get insight into students' learning situations by grading their English writings and identifying flaws in their own education and instruction [1–4]. Furthermore, the English composition level is often utilised as a significant criterion for assessing English learners' English ability as shown in a variety of English examinations. As a result, composition writing in English plays an essential part in English instruction. China has a sizable population, and each year, a huge number of students learn English. At the same time, China has limited educational resources, and its teacher-to-student ratio continues to lag behind that of affluent nations. Every year, a

big number of English examinations are required to correct English writings, putting a significant amount of pressure on instructors. How to assess and feed back the learning impact of language learners has long been a focus of study in the field of language instruction [5, 6]. According to current studies, the writing level is often regarded as the most accurate predictor of a learner's language competency. As a result, instructors and educational researchers place a high value on the writing portion of language tests. Manual composition correction, on the other hand, requires a lot of human assistance and takes a long time, making it impossible to give language learners with fast and efficient feedback and analysis [7]. For language learners, a natural language processing system that can identify and correct grammatical faults is critical.

Theoretically speaking, machine learning is a method to give machine learning ability so that the machine can improve its performance by using experience; in practice, it is a method to build a model through data collection training

and then use the model to predict the results [8–10]. Support vector machine algorithm (SVM), random forest algorithm, and artificial neural network algorithm are all commonly used algorithms in machine learning [11–13]. Machine learning is one of the means to realize artificial intelligence, which is considered to be the most intelligent branch in the field of artificial intelligence, and also one of the fastest developing branches. The grammar detection task involved in this research belongs to the natural language processing (NLP) direction, which studies how to realize the theory and method of using natural language to communicate effectively between machines and humans. It is a discipline integrating computer science, linguistics, and mathematics. The two core tasks of natural language processing are natural language understanding and natural language generation. Typical applications include machine translation, speech recognition, sentiment analysis, and chatbots. Natural language processing (NLP) is difficult and challenging because of the complexity and free combination of language as well as its knowledge and context dependence. Grammar error detection is an important branch of natural language processing tasks [14]. Grammar error detection task is simply to use the computer to identify the location of grammatical errors in human writing and give the classification of grammatical errors or to correct grammatical errors. It can be used for automatic correction of language learners' compositions, proofreading of writers' writing content, and grammar correction in daily writing scenes, etc., and it is very useful. Because of its extensive application, language grammar check has been widely studied by researchers at home and abroad. At present, it has achieved good results. For example, today's TOEFL test in the United States has realized the automatic evaluation of English composition by computer, with practical application. However, practice and exploration have also revealed that the current technology has not yet met the requirements of users [15, 16]. At present, the grammar checker is mainly based on the rule model. Due to the complexity of English grammar, it is impossible for the rule grammar to express all the grammar errors. However, the rule model is still the first choice for all kinds of grammar checkers because of its simple design, it is intuitive, and it is easy to use.

This paper's goal is to build and construct a grammar-checking tool that can be utilised as a part of an English composition auxiliary correction system to check and fix English grammar in writing. The conventional grammar-checking method is strict, with much of it being a single design, and it often has poor mistake checking accuracy and recall rate. To construct a new grammar-checking module, the new system should use the multimodel fusion design idea and merge numerous mistake-checking models. The entire design should be adaptable and expandable. As a result, in order to satisfy the overall design goal, a grammar-checking module suited for the essay assistance grading system must be built.

2. Related Work

With the advancement of machine learning algorithms, an increasing number of academics are turning their attention to grammar checking in the direction of machine learning algorithms. García-Díaz et al. [17] investigated a grammar-checking system for English based on the N-gram language model. Chen Chaocai conducted research on how to rectify collocation mistakes. It was categorised into four categories by Clarke et al. [18]: categorization and decision-making, intelligent education, traditional business automation, and typical applications for future development. Artificial intelligence technology is used to examine the data of learners on the online platform in the area of analysis and decision. Intelligent teaching, according to Park [19], employs adaptive teaching methodologies; integrates cognitive, learner, recommendation, and other technologies; and suggests tailored learning routes and learning materials for learners based on their learning habits and personality traits. Zhao et al. [20] developed a unified framework for decoding, with the major body of the system using the standard N-gram language model. In this system, a language model is utilised to correct most grammatical problems, while articles and prepositions are processed using a maximum entropy classifier. To address subject-verb agreement issues, Kumar and Boulanger [21] employed a tree language model. However, since the nodes of this tree model must include extra grammatical information in addition to the words in the sentence, data sparsity will result, affecting the model's real application impact. Around the year 2000, a statistical language model based on the oracle network was proposed [22]. The training model has not been extensively employed in the area of grammatical mistake correction since it demands a lot of computer resources and takes a long time to learn. However, several teams have begun to employ RNN (Recurrent Neural Network) in the building of grammatical error-correcting decoders because of the high abstract ability of neural networks, the quick growth of deep learning technology in the last two years, and the increase in hardware matching performance. However, training such a huge model in an acceptable amount of time in an average laboratory [23] is a significant problem; hence, this technological strategy was not adopted in this research. The goal of this paper is to investigate a common statistical language model in the field of English grammar error correction, analyse and improve it, design a hierarchical language model, and apply this language model to the design of an English grammar error detection and correction module, as well as to verify its application effect.

3. A Grammar Detection Method for English Composition

3.1. Total Framework. This study intends to complete an English composition grammar-testing system design and development, to apply before the trained model and system, and its main functions are to detect the students' upload composition of syntax errors; to mark the location of the grammar mistakes; to categorize error according to the types of errors, such as word order error, error of word choice, and all kinds of error statistics of the whole article on the number of errors; and to help students understand their grammatical weaknesses in English writing. The technical route of this
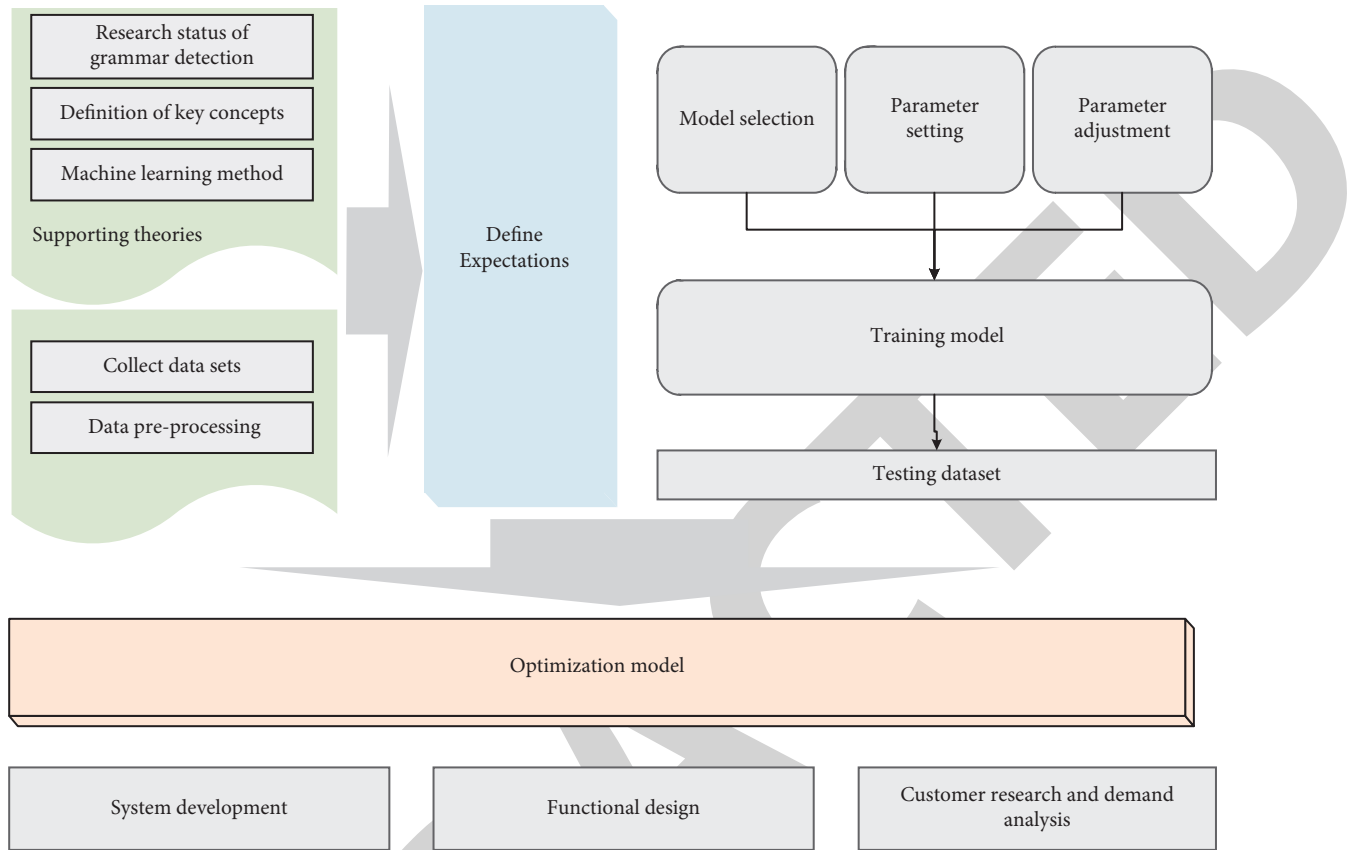
Figure 1: Technical route of this study.

research is shown in the following figure, which is divided into two modules, theory and practice, as shown in Figure 1.

The theoretical part includes the definition of core concepts, the statistics of algorithm models commonly used in educational applications of natural language processing, and the summary of common strategies for grammar detection. The practice part is the core part of this paper, including the collection and processing of data sets, model selection and adjustment parameters, model evaluation, and system design and development. In particular, the selection of model and parameter adjustment is a process that needs multiple iterations and continuous optimization, which is the focus of this study.

3.2. Hierarchical Language Model Construction.

In order to describe the information of English collocation phrases more effectively, we consider to introduce Dependency Parsing to capture the dependency of words in sentences and then integrate the advantages of the traditional N-gram model to describe the syntactic tree, that is, to retrieve n-tuples from the syntactic tree rather than directly obtaining n-tuples in the order in which words appear in the text.

In the process of constructing a hierarchical language model, this paper first analyses Dependency Parsing of corrective sentences and then generates hierarchical clauses by using the analysis results, that is, Stanford uses Parser to analyse the dependency groups obtained from sentences that are Stanford typed. The process of generating hierarchical clauses is as follows: firstly, a series of associative relation groups with the same dependent words are selected, the words in these relations are extracted, and then put into a new sentence according to the sequence of the original position of these words so as to form a clause with the associative word as the head node. Hierarchical clause is constructed from the dependency tree obtained from dependency analysis, and the specific hierarchical clause diagram is shown in Figure 2.

Starting from the root node of the sentence, the root node and all the direct children of the root node form a clause in the original order. The first level clause is the trunk of the whole sentence, which removes all the modifiers of the children, concentrates the sentence content, and retains the semantic information. All the immediate children of the root node are traversed, recursively creating clauses in the same way if the children have children. These low-level clauses are the embellishing parts of the upper level clauses.

3.3. Grammar Error Correction Module.

The general grammar error correction module designed in this paper mainly includes dependency analysis, hierarchical clause generation, substitution word generation, language model decoding, and error correction result generation. The entire module flow is shown in Figure 3.
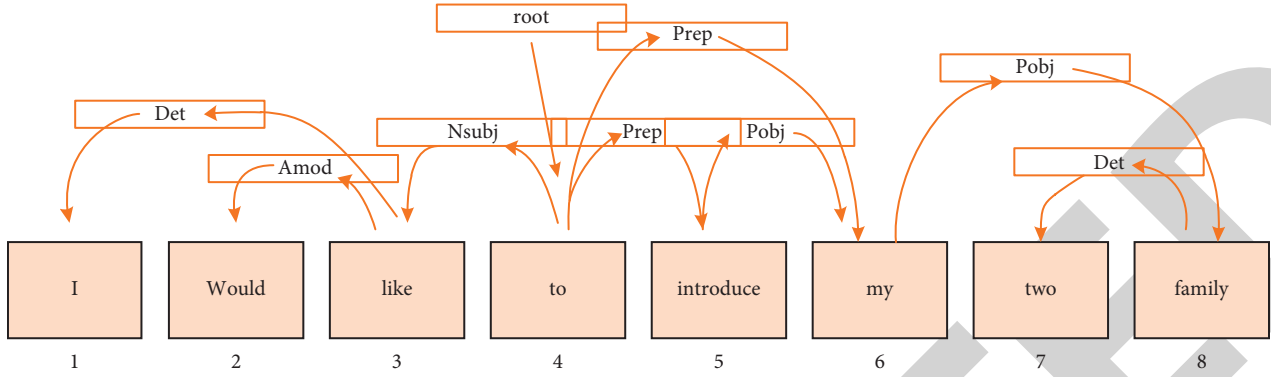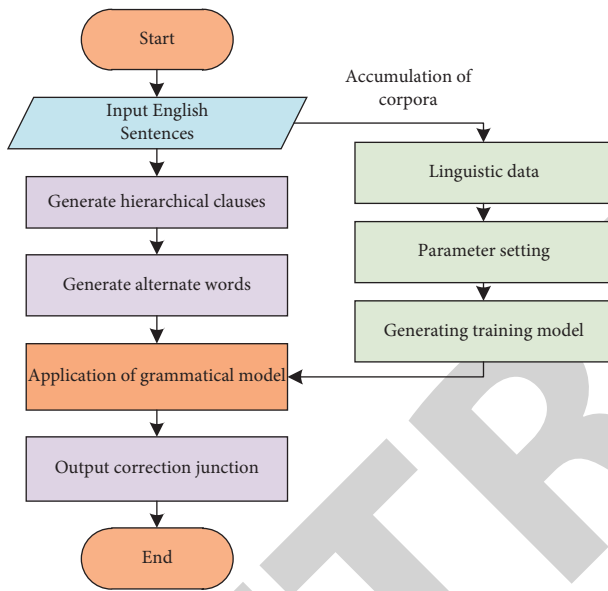
Figure 2: Dependency tree for parsing.

Figure 3: System flow chart of grammar error correction.

The program begins by reading a statement from the text, dividing the statement into words, then obtaining the dependencies of each word, and recombining to generate hierarchical clauses. Then, the candidate word tool is used to get the alternative word for each word, and the decoding process of the language model is started. The decoding process starts at the top level of the clause and uses the probabilities calculated by the language model to calculate and preserve the maximum probabilistic path for a series of words at the current position. Since each word may have a modifier, the process is recursive.

3.4. N-Gram Algorithm Strategy.
In addition to formulating grammar rules, English grammar detection mainly adopts dictionary search method 14 and N-gram algorithm. The dictionary search method scans the string in the detected text and compares it with the dictionary. Failure to match is judged to have a syntax error. This method is a more accurate and commonly used method for English grammar detection. Its disadvantage is that it is difficult to eliminate the ambiguity of natural language. N-gram method is a

statistics-based method, which divides the text to be detected into N-element strings and then calculates the frequency of the string in the whole corpus through statistical model. If the frequency is lower than the preset threshold, it will be marked as having grammatical errors. This method relies on a very large corpus to obtain more accurate results.

N-gram algorithm can reflect the context relations well. In principle, the greater the order of each chip is, the stronger its ability to reflect the context relations is. If the sparse problem of the corpus is taken into consideration, too high order is detrimental. In practical use, binary grammar and ternary grammar are often adopted. Meta-grammar-checking systems are usually divided into two phases: the training phase and the checking phase. The system calculates and saves the corpus information according to the requirements of the model. In the checking stage, the information obtained from the input sentence is counted out to judge the grammatical errors. Different checking systems have different ways of using meta-grammars to check grammar, which can be generally divided into two kinds.

First, for the input sentence, the model will first calculate the probability of the occurrence of binary grammar in the sentence and then calculate the product to get the probability of the sentence. The formula is as follows:

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

In the second, the system finds all the binary grammars for the sample text input. In order to unify the calculation process, the beginning and end marks are generally added at the beginning and the end of the sentence respectively, so the probability of the sentence is calculated as follows:

$$p(s) = \prod_{i=1}^{n} \frac{p(w_n|w_1 \cdot w_2 \cdot w_3 \cdot \cdots \cdot w_{n-1})}{p(w_i|w_{i-1})},$$

$$p(w_i|w_{i-1} \cdot w_{i-2}) = \frac{c(w_{i-1} \cdot w_{i-2})}{c(w_{i-1} \cdot w_{i-2} \cdot w_i)}. \tag{2}$$

In the formula, $c(w_{i-2}w_{i-1}w_{ii})$ represents the number of triples in the training corpus, and $C(WI-2WI-1)$ represents the number of tuples.
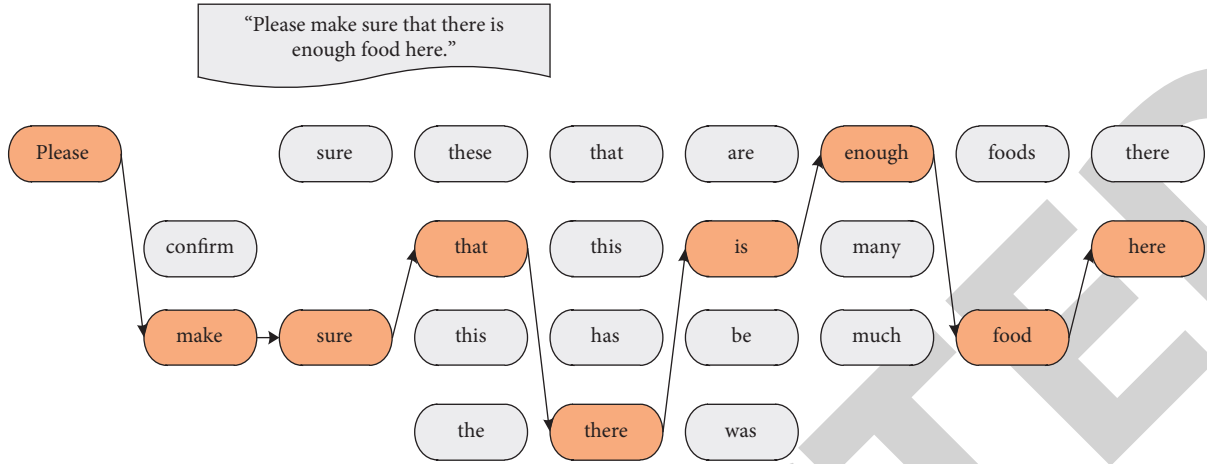
Figure 4: An example of language model error correction.

If the number of occurrences of a tuple is directly counted, the probability is estimated to be zero if the number of occurrences of a tuple is zero. The phenomenon of zero frequency in the corpus cannot reflect the real statistical information but will bring calculation errors to the subsequent applications. So, it needs to be smoothed. Taking the binary language model as an example, if the number of occurrence of each binary language model is once more than the actual number of occurrence in the corpus, then the zero-frequency problem will not appear. The specific formula is as follows:

$$p(w_i|w_{i-1}) = \frac{c(w_i w_{i-1} w_{i-2})}{\sum_{wi} c(w_{i-1} w_i)} = \frac{c(w_i w_{i-1} w_{i-2})}{\sum_{wi} c(w_{i-1} w_i)} + |V|. \quad (3)$$

The smoothing method is modified as shown in the following formula:

$$p(w_i|w_{i-1}) = \frac{\max\{A - c(w_{i-n}^i)\}}{c(w_{i-1} w_i)}. \quad (4)$$

Since the smoothing effect of this method is superior to other methods in the second- and third-order language models, the smoothing method adopted in this paper is Kneser–Ney smoothing method.

4. Application of Grammatical Error Detection in English Composition

4.1. Correction of Grammatical Errors. There are many ways to correct grammatical errors by using hierarchical language model. A simple method is to generate a series of alternative words for each word of a sentence to be tested and select the combination with the highest "score" from the original word and all alternative words to form the final alternative sentence. The "score" here is calculated by the language model.

As shown in Figure 4, the top line is the original sentence to be tested. Each word is located in the right-angle box, and the rounded corner box just below the right-angle box is the alternative word corresponding to the original word. The error correction process of the original sentence can be seen

Table 1: Data set distribution statistics.

	All the sentences	Correct sentence	Wrong sentence
Training set	98045	67021	31024
Test set	6702	3400	4302

as the process of finding a path with the largest score according to the language model, and the path connected by the arrows at the end is the final error correction result, which is the decoding process of the original sentence. This method can be used to correct a variety of grammatical errors including spelling errors and subject-verb agreement errors.

In this study, there are 10071 sentences in the training set, 24797 of which have grammatical errors, and 9033 sentences in the test set, 3316 of which have grammatical errors. For example, a study published by Alibaba in 2017 found that adding grammatically correct sentences can improve the model results to a certain extent. Therefore, in this study, the author added a higher proportion of correct sentences in the training.

This is shown in Table 1. Every grammatically incorrect sentence contains at least one grammatical error. The proportion of the training set is slightly higher than the normal proportion because the training set contains a higher proportion of grammatically correct sentences. Therefore, in this study, the author added a higher proportion of correct sentences in the training set.

In order to make the test set better evaluate the effect of the model, the proportion of training set and test set is basically the same in terms of the distribution of different types of syntax errors as shown in Figure 5. In the training set, there were a total of 49594 grammatical errors, which were divided into 11076 redundant word errors, 13246 missing word errors, 21898 word selection errors, and 3374 word sequencing errors. In the test set, there were a total of 11085 grammatical errors, including 2406 redundant word errors, 2973 missing word errors, 4860 word selection errors, and 846 word sequencing errors.
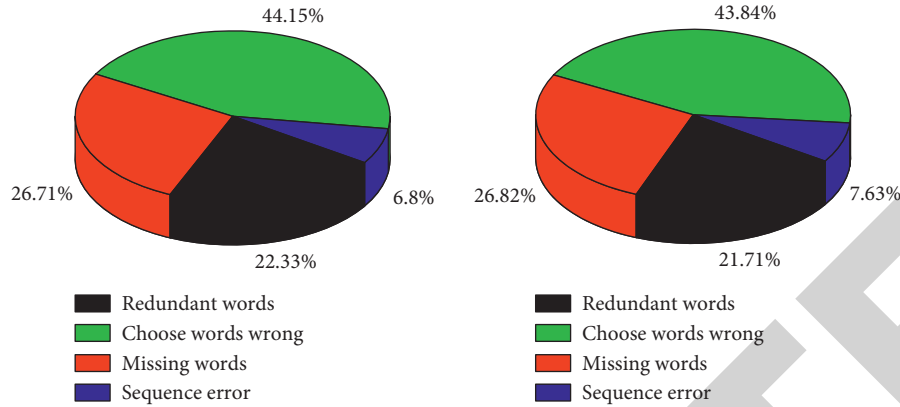
Figure 5: Classification of data set error types.

4.2. Evaluation and Verification of Error Detection Effect. In order to ensure the rationality of the evaluation, this paper selectively selects the most representative English grammar errors from many English grammar problems as the main content of the evaluation in order to judge the effect of the system error detection ability. Due to the low proportion of other grammatical contents in corpora, the proportion of different corpora is not stable, so it can be summarized into other contents. The evaluation content selected in this paper is shown in Table 2.

The syntax error detection effect is carried out on this basis, and the main syntax error distribution statistics are shown in Figure 6.

To answer the issue of English grammar identification, three distinct algorithm models were utilised in this study: the CRF model based on statistics, the LSTM-CRF model based on neural network, and the multitask learning model. In its particular area, each model has its own benefits and peculiarities (see the previous chapter for detailed description). As a result, this research evaluates and analyses the performance of the three models using the same training set and evaluation index in order to choose the best algorithm model for this job. The confusion matrix from the preceding section was used as the assessment standard, and the model was assessed on three levels: detection, identification, and location. Figure 7 shows the evaluation findings.

It can be found from the above experimental results that the multitasking learning model has the best performance in all three evaluation levels. The CRF model has poor performance because it largely relies on feature engineering. Due to the sparsity of the training set, it is difficult to carry out feature engineering. Even if the part-of-speech features and semantic features are added and many feature template pages are designed, the model effect cannot be improved. Also, manual feature extraction makes it difficult to find specific features to catch specific types of errors. The performance of the LSTM-CRF model is slightly better than that of the CRF model because it can automatically extract the features of the CRF model rather than manually. But it still has the problem of sparse data. At all three levels, the performance of the multitasking learning model is better than that of LSM-CRF because the sequential tag model is

Table 2: Syntax error types.

Type	Content
Vt	Wrong tense
V_0	Error of missing verbs
Prep	Wrong use of prepositions
SVA	Subject-verb inconsistency
Nn	Singular or plural error
Others	—

optimized only based on the information contained in the tag, whereas in the test set, more than 70% of the sentences contained no errors in the test set. Therefore, many tags in the data set made little contribution to the training process. However, the multitasking learning model with auxiliary tasks does not rely entirely on obtaining information from tags, so the model can be fully trained under such uneven tags, and has a better performance in grammar error detection task than other models.

4.3. English Event Pronoun Resolution. Unlike entity reference, event reference cannot be applied to event reference disambiguation because its antecedent candidate is an event, which has a completely different semantic classification system from nominal reference pronoun. This paper presents an event pronoun reference disambiguation platform based on machine learning and introduces the example generation and feature selection process of the platform in detail. Anaphora can be roughly divided into two categories: (1) entity anaphora, the antecedent and anaphora of which are both concrete entities of objective existence, and (2) event reference: anaphora refers to events, facts, propositions, and other event and abstract objects. This paper selects effective features for event reference resolution from three aspects. The basic principle of selection is that it is easy to obtain and effective for this task. The so-called minimum extension tree is a structured syntactic tree that only retains the shortest path between the antecedent candidate and the reference. Since the anaphora and the antecedent may not be in the same syntactic tree, we can link the two syntactic trees by adding a virtual node TOP to form a discourse tree and
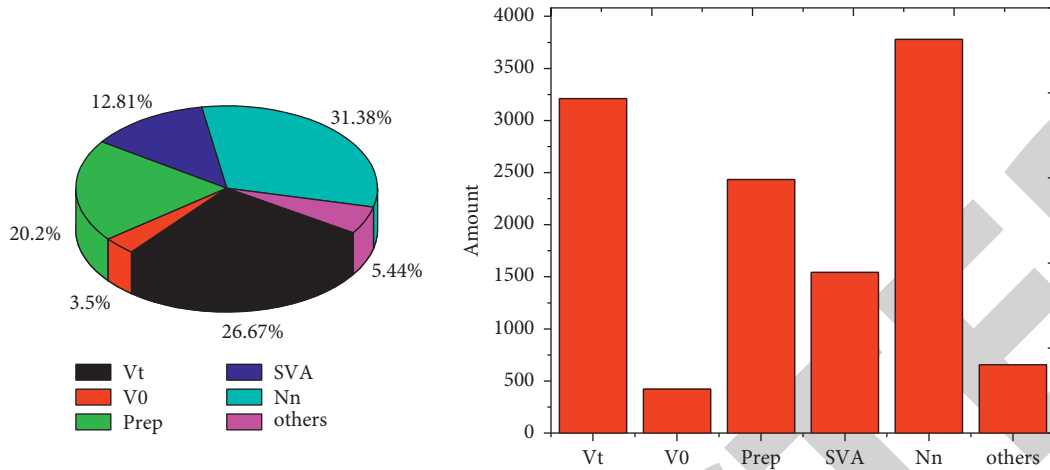
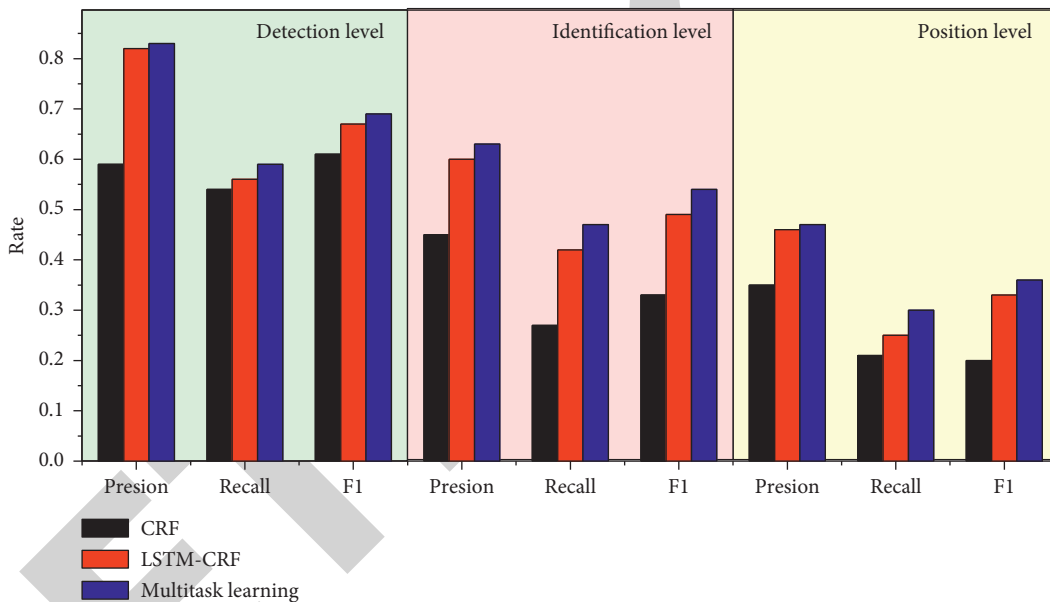FIGURE 6: Distribution statistics of major syntax errors.

FIGURE 7: Evaluation of experimental test results.

then select the minimum extension tree on the discourse tree.

To evaluate our platform with a larger specification and a wider range of OneNotes 3.0 data. Figure 8 shows the number of anaphora.

It can be seen that the corpus has detailed annotation of lexical, syntactic, semantic, and other information, especially the annotated event reference relation, which can be well used by our digestion platform. Our experiment will use the English portion of its news corpus (approx. 500K, WSJ300K, Radio News 200K). The proportion of event pronoun is relatively low (about 4% of the total number of proxy words and 14% of the reference words of proxy words), and the pronoun itself contains less information, so the resolution of event pronoun is more difficult than the resolution of other event noun phrases or entity reference words. Here, it refers to the digestion similar, and the real event refers to the pronouns recognition. This paper discusses the event that refers to the first language of pronouns recognition task, in order to not to be affected by event that refers to the pronouns recognition performance, and here, we assume that the event refers to the pronouns and known exactly right. Figure 9 shows the distribution of the distance between the antecedent and the event pronoun.

As can be seen from Figure 9, the distance between the event pronoun and its antecedent is not large, whether measured by the sentence unit or by the number of central verbs. Therefore, in this paper, the search space of the antecedent is limited to the current sentence and the first two sentences. In addition, according to the central theory, the focal point of two adjacent utterances should be smooth and
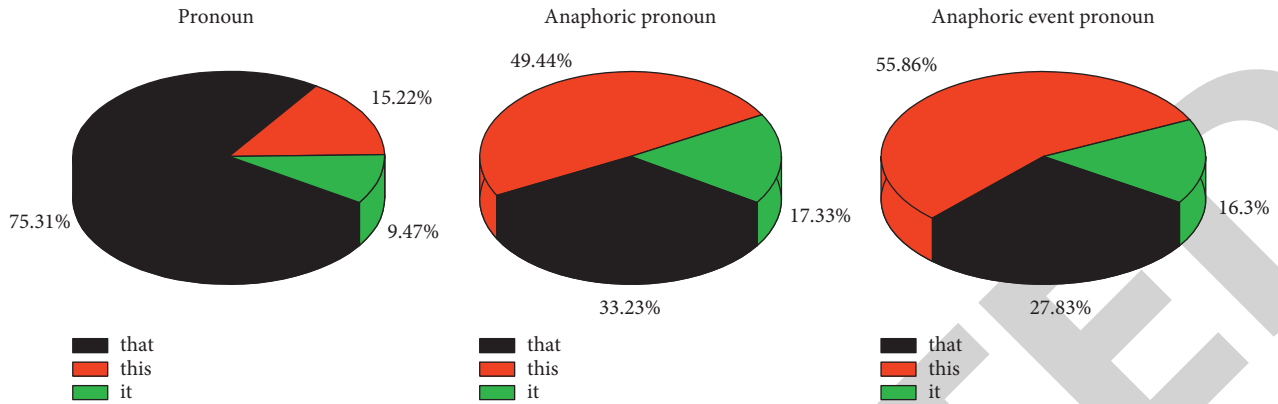
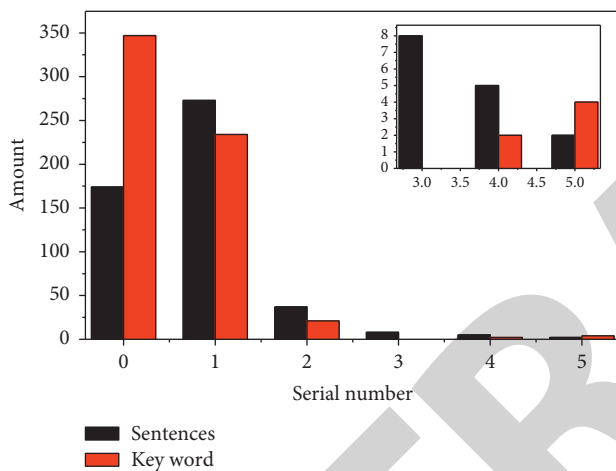FIGURE 8: The distribution of various anaphora in the composition.

FIGURE 9: The distribution of the distance between the antecedent and the event pronoun.

cross, and the sentence in which the anagram and the antecedent are located should also have a high degree of semantic similarity or correlation.

5. Conclusion

We propose and implement a grammar-checking module for an English composition grading system in this work. In English composition, the module identifies and corrects grammatical problems. The experiment shows that the system has a high accuracy and recall rate as well as meeting precise design criteria. The goal of this study is to increase the grammar check's accuracy and recall rate. This work tries to combine models and introduces separate article and preposition grammar modules to achieve this goal. The specific work of this study is mainly to understand the requirements of the system through research, and from the two aspects of system implementation and effect promotion, study the grammar check work of all aspects, combine the requirements with the design, and finally determine the technical scheme of the system design. Adopt the rule model first and then integrate a number of models into the overall design scheme. The author gathers and organises a huge number of English grammar rules from the Internet and English grammar monographs in order to develop the rule-based grammar-checking module. Nearly a thousand grammatical rules have been categorised and kept for future use by the system. Some outcomes have been accomplished as a consequence of the system design, but English grammar check still has a long way to go.

Data Availability

The data used to support the findings of this study are included in the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] Z. He, "English grammar error detection using recurrent neural networks," *Scientific Programming*, vol. 2021, pp. 1–8, Article ID 7058723, 2021.

[2] Y. Wang, P. Jia, L. Liu, C. Huang, and Z Liu, "A systematic review of fuzzing based on machine learning techniques," *PLoS One*, vol. 15, no. 8, Article ID e0237749, 2020.

[3] N. Ballier, S. Canu, C. Petitjean et al., "Machine learning for learner English," *International Journal of Learner Corpus Research*, vol. 6, no. 1, pp. 72–103, 2020.

[4] S. M. Lee, "The impact of using machine translation on EFL students' writing," *Computer Assisted Language Learning*, vol. 33, no. 3, pp. 157–175, 2020.

[5] Y. Jiang, Y. Yang, L. Nie, H. Liu, and J. Zheng, "Identification and effective connections of core networks in patients with temporal lobe epilepsy and cognitive impairment: g," *International Journal of Neuroscience*, pp. 1–12, 2021.

[6] B. Settles, G. T LaFlair, and M. Hagiwara, "Machine learning–driven language assessment," *Transactions of the Association for computational Linguistics*, vol. 8, pp. 247–263, 2020.

[7] A. Al-Ahdal, "Using computer software as a tool of error analysis: giving EFL teachers and learners a much-needed impetus[J]," *International Journal of Innovation, Creativity and Change*, vol. 12, no. 2, pp. 418–437, 2020.

[8] M. Allamanis, E. T. Barr, P. Devanbu, and C Sutton, "A survey of machine learning for big code and naturalness," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–37, 2019.

[9] J. Shin and M. J. Gierl, "More efficient processes for creating automated essay scoring frameworks: a demonstration of two algorithms," *Language Testing*, vol. 38, no. 2, pp. 247–272, 2021.

[10] S. L. Marie-Sainte, N. Alalyani, S. Alotaibi et al., "Arabic natural language processing and machine learning-based systems[J]," *IEEE Access*, vol. 7, pp. 7011–7020, 2018.

[11] S. M. Lee, "An investigation of machine translation output quality and the influencing factors of source texts[J]," *Re-CALL*, vol. 34, pp. 1–14, 2021.

[12] H. Yannakoudakis, Ø E. Andersen, A. Geranpayeh, T. Briscoe, and D Nicholls, "Developing an automated writing placement system for ESL learners," *Applied Measurement in Education*, vol. 31, no. 3, pp. 251–267, 2018.

[13] F. Ullah, J. Wang, M. Farhan, M. Habib, and S. Khalid, "Software plagiarism detection in multiprogramming languages using machine learning approach[J]," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 4, Article ID e5000, 2021.

[14] Z. R. Liu and Y. Liu, "Exploiting unlabeled data for neural grammatical error detection," *Journal of Computer Science and Technology*, vol. 32, no. 4, pp. 758–767, 2017.

[15] F. J. Ryan, "Application of machine learning techniques for creating urban microbial fingerprints," *Biology Direct*, vol. 14, no. 1, p. 13, 2019.

[16] N. Madi and H. S. Al-Khalifa, "A proposed Arabic grammatical error detection tool based on deep learning," *Procedia Computer Science*, vol. 142, pp. 352–355, 2018.

[17] J. A. García-Díaz, M. Cánovas-García, R. Colomo-Palacios, and R Valencia-Garcia, "Detecting misogyny in Spanish tweets. An approach based on linguistics features and word embeddings," *Future Generation Computer Systems*, vol. 114, pp. 506–518, 2021.

[18] N. Clarke, P. Foltz, and P. Garrard, "'How to do things with (thousands of) words: c," *Cortex*, vol. 129, pp. 446–463, 2020.

[19] J. Park, "'Implications of AI-based grammar checker in EFL learning and testing: Korean high school students' writing," *The Korea English Language Testing Association*, vol. 14, no. 1, pp. 11–39, 2019.

[20] H. Zhao, Z. Liu, X. Yao, and Q. Yang, "A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach," *Information Processing & Management*, vol. 58, no. 5, Article ID 102656, 2021.

[21] V. S. Kumar and D. Boulanger, "Automated essay scoring and the deep learning black box: how are rubric scores determined?[J]," *International Journal of Artificial Intelligence in Education*, pp. 1–47, 2020.

[22] F. V. Lim and J. Phua, "Teaching writing with language feedback technology," *Computers and Composition*, vol. 54, Article ID 102518, 2019.

[23] P. C. Bermant, M. M. Bronstein, R. J. Wood, S. Gero, and D. F. Gruber, "Deep machine learning techniques for the detection and classification of sperm whale bioacoustics," *Scientific Reports*, vol. 9, no. 1, Article ID 12588, 2019.