Research Article

Xiaojie Huang*

Research on business English grammar detection system based on LSTM model

https://doi.org/10.1515/jisys-2023-0309
received December 14, 2023; accepted February 20, 2024

Abstract: In order to solve the problems that the current English grammar correction algorithms are not effective, the error correction ability is limited, and the error correction accuracy needs to be improved, this study proposes an automatic grammar correction method for business English writing based on two-way long short-term memory (LSTM) and N-gram. First, this study considers article and preposition errors as a special sequence labeling task, and proposes a Grammar error checking (GEC) method for sequence labeling based on bidirectional LSTM. During training, english as a second language (ESL) corpus and supplementary corpus are used to label specific articles or prepositions. Second, for noun simple-plural errors, verb form errors, and subject-verb inconsistency errors, a large number of news corpora are used to count the frequency of N-gram, and a GEC method based on ESL and news corpora N-gram voting strategy is proposed. Experimental results show that the overall F_1 value of the method designed in this study on the GEC data of CoNLL2013 is 33.87%, which is higher than the F_1 value of UIUC. The F_1 value of article error correction is 38.05%, and the F_1 value of preposition error correction is 28.89%. It is proved that this method can effectively improve the accuracy of grammar error correction and solve the gradient explosion problem of traditional error correction model, which is of great significance to further strengthen the practicality of automatic grammar error correction technology.

Keywords: deep learning, syntax detection, LSTM

1 Introduction

Grammar error checking (GEC) is a very important and long-standing problem in the field of computer-aided linguistics, and it is also an important problem that must be overcome in natural language processing. It uses the dependency structure and coherence between the components to verify, repair, and correct the grammar. For second language acquisition, students not only need to quickly identify and correct the wrong use of vocabulary, but also to ensure that the grammatical structure is correct. Only by conforming to the rules of language structure can our wording and expression be more precise and persuasive. The use of GEC AIDS can effectively improve the language skills of english as a second language (ESL) learners. This tool can provide them with grammar, spelling corrections, and suggestions, in order to strengthen their listening, speaking, reading, writing, and communication skills, so that grammar is no longer a barrier in ESL learning.

In recent years, GEC has become a research hotspot in the field of natural language processing. After Ng et al. opened the CoNLL-2013 and CoNLL-2014 shared tasks, along with the continuous improvement of machine learning algorithms and models, the research related to grammatical error correction in the English department has been greatly promoted. Wang et al. proposed a labeling model based on recurrent neural networks to address the impact of grammatical errors on sequence information extraction in English writing corpora, which was verified by experiments and achieved a high accuracy [1]. Chen et al. used multi-

* **Corresponding author: Xiaojie Huang,** Department of Foreign Languages, Hubei University of Technology Engineering and Technology College, Wuhan 430068, China, e-mail: huangxjxj1988@outlook.com

layer forward feedback neural network to build a language model, complete English grammar correction, and designed the overall architecture of the system, which achieved high accuracy and fast error correction speed [2]. Zhou and Liu described in their study that they modify and adjust for grammatical errors by applying algorithms similar to human language processing neural networks [3], which opens a new way for us to solve such problems more effectively in the future. Wang and Gu developed a multilayer perceptron-based algorithm for automatically correcting English writing grammar [4]. The algorithm takes the feature set as the input of the multi-layer perceptron, and completes grammar correction by setting penalty parameters and deviation parameters, and improves the timeliness and recall rate of English grammar correction. Solyman et al. proposed a GEC model based on seq2seq Transformer [5], which uses expectation maximization routing algorithm to dynamically synthesize cross-layer information in Arabic GEC, and introduces bidirectional regularization terms using Kullback–Leibler divergence to improve the accuracy of syntax error correction.

Although the accuracy and recall rate of the above models have been significantly improved, there is still a long way to go before intelligent grammar error correction is fully realized. At the same time, the error correction task requires supervised corpus for model training, which often leads to insufficient corpus for text error correction in the training process [6]. N-gram model solves this problem well, and can use a large number of unsupervised corpus to train statistical language models. N-gram is the simplest and most commonly used model, but the model does not consider the semantic information of the context when replacing the characters inside the text using the puzzle set, which often results in the situation that although the adjacent words are correct, they are not logical in the whole sentence, resulting in a low score [7]. With the gradual rise of machine learning, more and more scholars apply deep learning methods to error correction tasks [8]. The artificial neural network of long short-term memory (LSTM) solves the problem of gradient disappearance and long-term dependence of information by setting input gate, output gate, forgetting gate, etc. The effect is more prominent in processing sequence problems. LSTM artificial neural network model can learn the deep information of text, strengthen the understanding of text semantics, deepen the learning of text structure, and have a better help to obtain the recommended modification options [9].

Therefore, this study proposes an automatic grammar correction method for business English writing based on two-way LSTM and N-gram. Aiming at the errors of articles and prepositions in business English writing, this study treats them as sequential labeling problems and uses bidirectional LSTM to train the model. In this study, an automatic English grammar correction method based on N-gram voting strategy is proposed for noun singular and plural errors, verb form errors, and subject-verb inconsistency errors.

The main contributions of this study are as follows: (1) In the N-gram method of this study, a large number of news corpus are used to perform N-gram frequency statistics, so as to identify and correct subject-predicate errors of nouns and verbs; (2) In the bidirectional LSTM model, manual error generation is carried out on the corpus without grammatical errors to balance the differences between the corpus and supplement the corpus for model training. The innovations of this study are as follows: the traditional GEC method uses N-gram or rule-based method to correct syntax errors, and only uses context information with fixed window size for correction, which is not enough. Moreover, when the window size becomes larger, it is difficult to train the model. In this study, the bidirectional LSTM network model can learn the long-term dependent information that determines the use of prepositions or articles, and can avoid the problems such as gradient disappearance that may occur in traditional recurrent neural networks.

The automatic detection of business English grammar can bring better development opportunities for artificial intelligence, enhance its knowledge and skills of natural language analysis, and improve its semantic search ability. In addition, the results of this study have some enlightening significance for foreign language teaching and native language teaching. Through this research, our results can help students to have a deeper understanding of the meaning of English words, reduce the incidence of grammar errors, and enhance their confidence and motivation in learning, and finally achieve a multiplier effect in learning with half the effort.

2 Related work

2.1 Word segmentation technology

After dividing a paragraph into single sentences, we also need to extract the words from a sentence, which is the smallest unit of language study. Therefore, word segmentation is a very critical step, which is related to the problem analysis of the whole sentence, and is an important step in text processing.

English word segmentation is relatively simple compared with Chinese, because English itself has more obvious characteristics, there are spaces or other symbols between the words. However, English word segmentation is also difficult, and there are many ambiguous scenes. For example, abbreviations of words: it's and couldn't, it's can be used as a word or split into it is, couldn't can be split into could not or as a separate word. Similarly, the representation of the number can also affect the segmentation, for example, 222333 can be written as 222,333 or 222.333, and the use of different punctuation marks to represent the number causes some ambiguity. There are also certain words, such as New York and Los Angeles, which should be treated as a complete word in word segmentation, and should not be separated to facilitate part of speech tagging [10,11].

At present, the commonly used word segmentation methods mainly include the following types: rule model, which defines a series of word segmentation rules combined with regular expressions to divide words in sentences. The other is the word segmentation technology based on statistical probability model. Based on the corpus and some commonly used language models such as N-gram and Hidden Markov model, the segmentation results are compared and analyzed to select the most reasonable way of word segmentation [12].

2.2 Part-of-speech tagging

Part-of-speech tagging is a common technique in natural language processing. Part-of-speech tagging is to label parts of speech for each word unit on the basis of word segmentation, such as verbs, nouns, adjectives, adverbs, and so on. The grammatical correctness of a sentence is closely related to the part-of-speech of a word. Some grammatical errors are caused by the wrong tense of a word. The biggest difficulty in part-of-speech tagging is to solve various ambiguity problems. Words and parts of speech are not all one-to-one correspondences, and some words have different parts of speech in different scenes. For example, here are two examples:

"I bought a book yesterday."

"I have booked a seat at the restaurant."

The book in the first sentence is the noun "book" and the verb "book" in the second sentence. It is not reasonable to distinguish the parts of speech only according to morphology, and the general processing method will label the parts of speech according to the context. The above example is only a typical example of ambiguity in part-of-speech tagging. Brown corpus is one of the most famous corpora in natural language processing. According to the statistics of part-of-speech tagging results, there are 4,100 words with multiple tagging and 35,340 words with only one tagging, which shows that ambiguity in English part-of-speech tagging is very common. The important task of part-of-speech tagging is to deal with these ambiguous scenarios.

There are two types of part-of-speech tagging: regular model and statistical model. Most of the early adopters were rule models, in which linguists defined the part-of-speech candidate set and formulated a series of part-of-speech tagging rules according to their understanding and familiarity with the language [13]. This method relies heavily on the linguist's knowledge of the language, and English is a very complex language, and thousands of rules may be formulated for some large corpora. The work is cumbersome, consumes a lot of manpower and material resources, and after too many rules, it is difficult to ensure correctness and consistency, and there may be consistency and inconsistency. The rule model has to do special processing for each language, and the work is extremely tedious. Therefore, the annotation method of the rule model is rarely used at present, at most as an auxiliary method.

Compared with the rule model, statistical method is much simpler. The development of statistical model is closely related to the construction and improvement of corpus [14]. According to the corpus and related algorithms, a tagger is first trained and learned, and then annotated. Although there is also the problem of manual initial labeling of data, the labeling here is much easier than making rules, and inconsistency before and after labeling is allowed because probability is used. Commonly used methods include Viterbi algorithm, CLAWS algorithm, and improved VOLSUNGA parts-of-speech tagging algorithm [15]. They all use Markov hypothesis principle, and apply dynamic programming technology to reduce the complexity of the algorithm and improve the speed of annotation. On the whole, the statistical model has lower requirements on the original labeled data, and the labeling process is simple, which has greater advantages than the regular model.

2.3 Syntax error correction algorithm

Syntax analysis is a common technique in natural language processing, and open source tools can perform syntax analysis and build corresponding syntax trees. Generally speaking, if the construction of a sentence's grammar tree fails, then the sentence should exist. Finally, according to the voting importance results, the grammar corrected results were obtained. Of course, syntactic analysis also has limitations and shortcomings [16]. The current syntactic analysis ability is limited, and the analysis ability of complex clauses is limited. For some special sentences, it is necessary to add parsing rules. One of the most important problems with syntax-based language is to establish a complete grammatical system for the language, which requires a lot of work by language experts. At present, the theories of grammar system construction by language experts are not consistent, and there are great controversies on some difficult points. This also leads to different results in the syntactic analysis of some sentences, and there is ambiguity.

The rule-based approach is to build the rule base first, and then match the input word sequence according to the rules. If the sentence has a syntax error, it will not match the rules in the rule base. Some rule bases are also built based on the parts of speech sequence of words, but the basic principle is no different from the word sequence principle. The word check feature of Microsoft Word uses the rule base method and can check multiple languages. And its error detection function is more powerful, for common word spelling errors, phrase errors and conjunction errors can be recognized, and can give suggestions for modification. Rule based syntax error correction is very convenient to implement, search and match according to the rules on the line, from the perspective of system implementation is simple and fast. In CoNLL2014, many papers mentioned the use of rules for grammar error correction, which is more used as an auxiliary function [17]. The biggest limitation of rule grammar correction is that rules are limited, grammar errors are complicated, English grammar errors are difficult to count, and with the expansion of the rule base, the cost of adding rules is getting higher and higher. It becomes more difficult to conclude new rules, there may be conflicts between rules, and the proliferation of rules cannot be stopped.

2.4 LSTM model

When tagging sequence data, the tagging information of the current word generally depends on the context information. Traditional sequence labeling methods rely on statistical or fusion methods, while recurrent neural networks can extract sequence information well by establishing sequence relationships of hidden layers. By setting threshold units and cells, LSTM can effectively avoid the problems such as gradient disappearance and gradient explosion that may occur in the training of traditional recurrent neural networks [18].

Compared with one-way LSTM model, which only accumulates the information before the current moment, two-way LSTM can accumulate the context information of the current moment, so that the model can synthesize the context information for sequence annotation. Bidirectional LSTM processes data from both

directions by using two separate hidden layers, both of which are then fed forward to the same output layer. The whole network structure is shown in Figure 1.

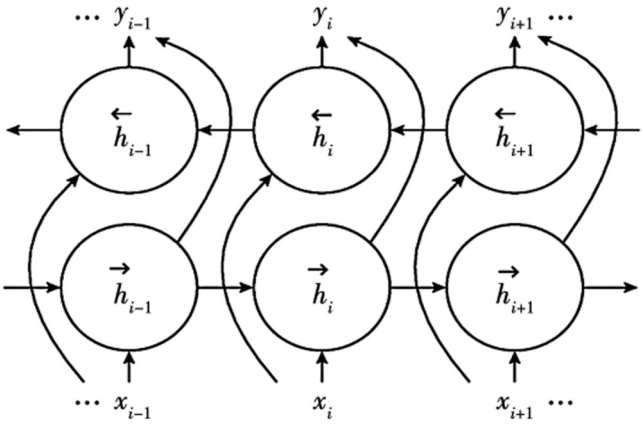

Figure 1: Schematic diagram of bidirectional LSTM network.

Through time T to 1 and time 1 to T, the bidirectional LSTM network uses the formulas (1)–(3) and the input vector sequence $x = (x_1, \cdots, x_N)$ to calculate the feedforward hidden layer sequence $\overrightarrow{h} = (\overrightarrow{h}_1, \cdots, \overrightarrow{h}_N)$, feedback hidden layer sequence $\overleftarrow{h} = (\overleftarrow{h}_1, \cdots \overleftarrow{h}_N)$, output sequence $y = (y_1, \cdots, y_N)$, and network output layer, respectively.

$$\overrightarrow{h} = f(W_{x\overrightarrow{h}}x_i + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{i-1} + b_{\overrightarrow{h}}), \tag{1}$$

$$\overleftarrow{h} = f(W_{x\overleftarrow{h}}x_i + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{i-1} + b_{\overleftarrow{h}}), \tag{2}$$

$$y_i = f(W_{\overrightarrow{h}y}\overrightarrow{h}_i + W_{\overleftarrow{h}y}\overleftarrow{h}_i + b_y), \tag{3}$$

where W represents the weight matrix; b is the bias vector; and f is the hidden layer function, specifically $f(u) = \frac{1}{1+e^{-u}}$.

3 GEC method based on N-gram and bidirectional LSTM

3.1 N-gram search service and knowledge base

The syntax error correction strategy is based on N-gram frequency statistics, so it is necessary to establish N-gram search service first. The N-gram sources used were all news for 2020 from about 12,000 news sites, with statistics shown in Table 1.

Table 1: N-gram details

Window size	Number of articles/million	Sentences/billion	Number of words/billion
1–5	1,460	1.26	34

In order to improve its query efficiency, the open source search engine is used to establish inverted index and provide search services.

Both articles and prepositions have limited variations and are in closed sets. Build a finite confusion set for articles and prepositions. The article confusion set contains three cases: the, a/an, and null. Null indicates that no article is used. The preposition confusion set contains the common 17 prepositions: on, about, into, with, as, at, by, or, from, in, of, over, to, among, between, under, and within. Unlike articles and prepositions, nouns and verbs are open sets. Therefore, the variation tables are established for noun errors, verb forms, and subject-verb inconsistency errors, respectively.

3.2 Moving windows and N-gram voting strategies

The GEC method for confounding set as open set is based on moving window and N-gram voting strategy.

The moving window is defined as follows:

$$MW_{i,k}(\omega) = \{\omega_{i-j}, \cdots, \omega_{i-j+(k-1)}, j = 0, k-1\}, \tag{4}$$

where ω_i is the ith word in the sentence, k represents the window size, and j is the distance between the first word in the window and ω_i. The selection of the window size k and the value range of j directly affect the effect of GEC. For different error types, different k and j values are selected.

The N-gram voting strategy simulates a real-life voting mechanism, with an N-gram fragment containing a syntactically incorrect candidate representing a candidate who may have the right to vote. Due to the limited corpus, the frequency of N-gram fragments may be very sparse. This policy sets a minimum effective frequency. Only when the queried frequency is higher than the minimum effective frequency, the N-gram fragment has the voting right.

In real life, the importance of votes cast by different people in different fields is different, and the importance of votes cast by experts in fields is higher than that of ordinary people. This strategy uses N-gram fragment length to simulate the degree of expertise of the domain expert, and the longer the N-gram, the higher the importance of the vote. Finally, according to the voting importance results, the grammar corrected results were obtained. The specific algorithm is as follows:

Input: N-gram fragment set Nset[window][candidates], least significant frequency set Fset[window], weight set Wset[window], N-grams Search service N-gram.

Output: Vote result set Rset[candidates].

```
from window = 3 to 5:
maxFrep = 0, vote = null
for candidate in Nset[window]:
        freq = N-gram.getFreq(candidate)
        If(freq > = Fset[window]andfreq > maxFreq):
                maxFreq = freq
                vote = candidate
Rset[vote] + = Wset[window]
return Rset.Max
```

Fset and Wset are parameters. Fset is the minimum effective frequency. Voting can be performed only when the queried frequency is greater than Fset. Wset is used to adjust the voting weights of N-gram fragments of different lengths.

This algorithm is based on the corpus, because on the one hand, the corpus size is limited and it is impossible to contain all fragments, and on the other hand, there are noise data in the corpus. Therefore, the minimum effective frequency Fset is set. Only when the frequency of the queried N-gram fragment is greater than this frequency, can it be shown that the corpus contains relevant corpus and this N-gram fragment has voting rights. According to the experimental comparison, Fset is set to 100.

The frequency of the N-gram segment with voting rights represents the probability of changing to the corresponding result. Suppose that the frequency of "have an apple" is 2, the frequency of "have the apple" is 1, and the frequency of "have apple" is 1. Then, according to the corpus, the probability of changing to "have an apple" is greater than "have the apple" and "have apple." The voting strategy is to select one as the voting object from "have an apple," "have the apple," and "have apple." This strategy selects the one with high probability as the voting object.

3.3 Annotation correction strategy based on bidirectional LSTM

Considering article and preposition errors as a special sequence labeling task, this study proposes a GEC method for sequence labeling based on LSTM. First, the training corpus with part-of-speech tagging is pre-processed, and the article part of speech is replaced by a special mark "ART" and the prepositional part of speech is replaced by a special mark "TO," and the above marks are exchanged with the positions of the articles or prepositions. Then, the model is trained using LSTM. Finally, the trained model is used to test the data. The labeling model based on LSTM is shown in Figure 2.

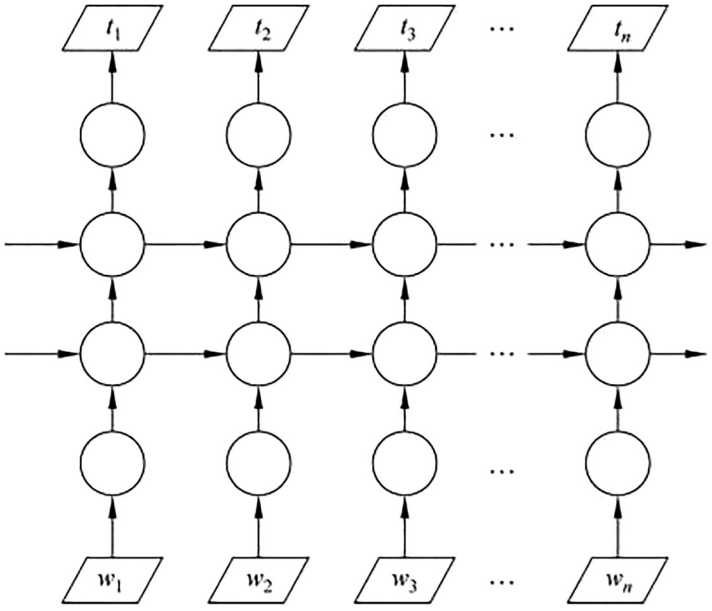

Figure 2: Labeling model based on LSTM.

The model first converts the word into a word vector as input to the model, entering the word vector at the corresponding position in the sequence at each moment. During the training process, word vectors are updated as parameters. The model uses the word vector as input to the LSTM unit and, at each moment, outputs the corresponding annotation vector. The tagging set is the union of all parts of speech sets and all confounding sets of prepositions or articles. The dimension of the annotation vector is consistent with the total number of annotation sets, and the annotation with the highest selection probability is marked by SoftMax.

The labeling model relies on back propagation through time algorithm and uses stochastic gradient descent for supervised training.

In this study, the grammatical errors with fixed confusion sets, namely, articles and prepositions, are marked and corrected. Before labeling, the article or preposition in the sequence is represented by a uniform identifier. When marking, the uniform identifier is marked with specific prepositions or articles to realize the correction of article or preposition errors in grammar.

3.4 Article error identification and correction

Article errors mainly include: misuse of articles, redundancy of articles, and absence of articles. In this study, article error correction is regarded as a special sequence labeling task, which involves three sub-modules: article error preprocessing module, article error recognition and correction module, and article error post-processing module.

(1) Article error preprocessing module.

The part of speech is replaced by a special mark "ART" to swap the part of speech with the position of the article. In this way, all places in the sentence where the article appears are replaced with "ART," and the corresponding part of speech is modified to indicate the article that should appear there.

(2) Article error recognition and correction module.

According to the given sentence, it determines the position where there may be incorrect use of articles in the sentence, makes part-of-speech tagging of the sentence, and then identifies all parts of speech labeled as articles (a, an, the). Article errors are identified and corrected using LSTM-based sequence annotation method. First, each word in the input sentence is converted into a word vector representation. Then, through the two-layer LSTM model, the annotation results are obtained.

(3) rticle error post-processing module

Swap the word with the special mark "ART" in the result of the previous step with the part-of-speech mark, and then delete the part-of-speech mark to get the final output result.

3.5 Identification and correction of preposition errors

Preposition errors mainly include: misuse of prepositions, redundancy of prepositions, and absence of prepositions. Preposition error correction is regarded as a special sequence labeling task, which involves three sub-modules: preposition error preprocessing module, preposition error recognition and correction module, and preposition error post-processing module.

(1) Preposition error preprocessing module

Replace the prepositional part of speech with a special mark "TO," switching the part of speech with the position of the preposition. In this way, all places in the sentence where the preposition occurs are replaced with "TO," and the corresponding part of speech is modified to indicate the preposition that should occur there.

(2) Preposition error recognition and correction module

According to the given sentence, it determines the possible positions in the sentence where the preposition is used incorrectly, tags the parts of speech in the sentence, and then identifies all the parts of speech labeled as prepositions. The sequence annotation method based on LSTM was used to correct preposition errors.

(3) Preposition error post-processing module

Swap the word with the special mark "TO" in the result of the previous step with the part-of-speech mark, and then delete the part-of-speech mark to get the final output result.

3.6 Noun single and plural error correction

Based on the noun-single and plural change table and N-gram voting strategy, the module mainly corrects the misuse of noun-single and plural. The specific correction process of this module is described as follows:

(1) The part-of-speech sequence of the example sentence is obtained by part-of-speech tagging, and the error candidate set is obtained by extracting the words labeled NN and NNS.

(2) The corresponding set of correction candidates is obtained by using the table of single and plural noun changes.

(3) Based on the correction candidate set, the N-grams fragment set is obtained using a moving window with a size of 3–5. Use the N-gram voting strategy to get the correct candidate with the highest number of votes, and replace the original sentence.

3.7 Verb and subject-verb inconsistency error correction

The verb error correction module mainly corrects the verb form misuse and subject-verb inconsistency. This module relies on verb form conjugation table, verb single and plural conjugation table and N-gram voting strategy. The specific correction process is as follows:

(1) The parts of speech sequence is obtained by marking the parts of speech of the words in the sentence. For verb form errors, the words labeled as VB, VBD, VBG, and VBN are extracted as error candidates. For subject-verb inconsistency errors, the words labeled VBP and VBZ are extracted as candidates for errors.
(2) The correction candidate set of the error candidate is obtained according to the error candidate and verb form change table/verb single and plural change table.
(3) For the correction candidate, the collection of N-grams fragments is obtained using a moving window with a size of 3–5. Use the N-gram voting strategy to get the correct candidate with the highest number of votes and replace it in the original sentence.

The GEC method system architecture based on LSTM and N-gram is shown in Figure 3.

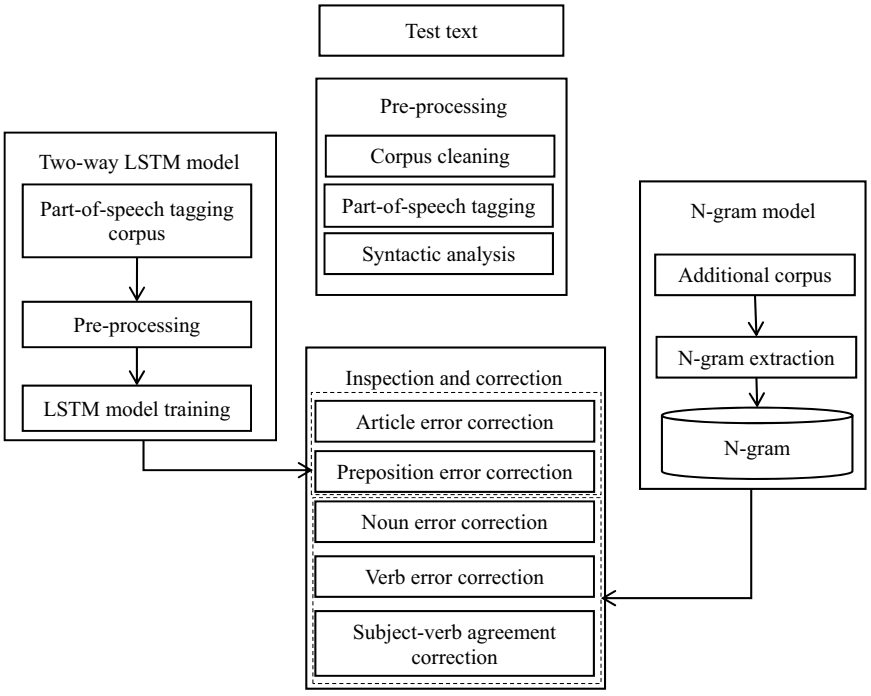

Figure 3: GEC architecture diagram.

4 Experiment

4.1 Experimental data

The experimental data came from the GEC evaluation task of CoNLL2013, and the statistical results are shown in Table 2. The CoNLL2013 corpus does not have correct parts-of-speech tagging, and the CoNLL2013 training corpus is smaller than PIGAI parts-of-speech tagging corpus and Brown corpus. Therefore, when the PIGAI parts-of-speech tagging corpus, Brown corpus, and labeled CoNLL corpus were used to expand the LSTM training corpus tagging, the Stanford tagging tool was used for parts-of-speech tagging. Among them, CoNLL2013 and PIGAI corpus are used as ESL corpus, and Brown corpus is used as supplementary news corpus to participate in the training of the model.

Table 2: GEC evaluation task data statistics of CoNLL2013

Error types	Training set		Test set	
	Number/piece	Proportion/%	Number/piece	Proportion/%
Articles	6,658	14.8	690	19.9
Prepositions	2,404	5.3	312	9.0
Noun	3,779	8.4	396	11.4
Subject-verb agreement	1,453	3.2	122	3.5
Verb form	1,527	3.4	124	3.6
Five types	15,821	35.1	1,644	47.4
All types	45,106	100.0	3,470	100.0

The GEC evaluation task data of CoNLL2013 marked a variety of error types, but the evaluation task mainly targeted at the five error types with a relatively high proportion: article error, preposition error, noun error, subject-verb agreement, and verb form error.

4.2 Evaluation method

The evaluation standard of CoNLL2013 is F_1, which is defined as follows:

$$F_1 = \frac{2 \times P \times R}{P + R}, \tag{5}$$

where P and R represent the accuracy rate and recall rate, respectively, defined as follows:

$$P = \frac{N_{\text{correct}}}{N_{\text{predicted}}}, \tag{6}$$

$$R = \frac{N_{\text{correct}}}{N_{\text{target}}}, \tag{7}$$

where N_{correct} refers to the correct number of syntax errors that the system modifies, $N_{\text{predicted}}$ refers to the number of syntax errors that the system modifies, and N_{target} refers to the number of errors in the corpus itself.

4.3 Experimental results and analysis

The experimental results of the GEC method based on LSTM and N-gram on the GEC evaluation data of CoNLL2013 are shown in Tables 3–5, and are compared with the corp-based methods of GEC and correction of English articles and UIUC methods.

Table 3: Results of article and preposition errors correction

Error types	Method comparison	P	R	F_1
Article error	UIUC	0.4784	0.2565	0.3340
	Corpus GEC	0.4349	0.2712	0.3345
	LSTM GEC	0.3315	0.4464	**0.3805**
Preposition error	UIUC	0.2653	0.0418	0.0722
	Corpus GEC	0.1346	0.1883	0.1570
	LSTM GEC	0.1921	0.5820	**0.2889**

Bold values represent the best performing values in each experimental result.

Table 4: Results of noun and verb errors correction

Error types	Method comparison	*P*	*R*	F_1
Noun error	UIUC	0.5223	0.3838	0.4425
	N-gram + vote	0.3266	0.5732	0.4161
Verb error	UIUC	0.3894	0.1789	0.2451
	N-gram + vote	0.1602	0.2195	0.1852

Table 5: Results of correcting all types of errors

Error types	This article			UIUC
	P	*R*	F_1	F_1
Articles	0.3315	0.1875	**0.2395**	0.18
+Preposition	0.2614	0.2976	**0.2783**	0.19
+Noun	0.2790	0.4358	**0.3402**	0.29
Subject-verb agreement	0.2753	0.4601	**0.3445**	0.30
Verb form	0.2652	0.4687	**0.3387**	0.31

Bold values represent the best performing values in each experimental result.

Table 6: Analysis of model syntax error correction effect

Models	*P*	*R*	F_1
RNN-Sep2Sep model	39.87	30.11	34.31
LSTM-Sep2Sep model	48.97	34.09	40.20
CNN-Sep2Sep model	61.08	33.27	43.08
Nested neural model	54.69	25.31	34.60
Depth context model	53.67	21.37	30.57
This study	**66.78**	**35.09**	**46.00**

Bold values represent the best performing values in each experimental result.

As shown in Table 3, for the correction of article errors, the F_1 value of the proposed method is 5% higher than that of the UIUC method and 5% higher than that of the Corpus GEC method. For the correction of preposition errors, F_1 value of the proposed method is 21% higher than that of UIUC method and 13% higher than Corpus GEC method, indicating that the LSTM-based GEC method is effective for article and preposition syntax error correction tasks. This is because the word vector contains the main rich context information, and using LSTM better learns the long-term dependency information that determines the use of articles or prepositions, so the results are better.

As shown in Table 4, when only N-gram + vote voting strategy is used to correct noun and verb errors, there is still a certain gap between F_1 value and UIUC method. This is because the news corpus on which the N-gram + vote strategy is based is different from the article that needs to be corrected, and a large number of correct sentences are changed into incorrect ones. The table of noun and verb conjugations cannot cover all the conjugations of nouns and verbs, which leads to certain limitations in the correction of nouns and verbs.

As shown in Table 5, for the correction of all five types of errors, the proposed method is superior to the UIUC method. The total F_1 value in the GEC data of CoNLL in 2013 is 33.87%, exceeding the total F_1 value of UIUC by 31.20%. The experimental results show that the automatic correction method based on LSTM and N-gram is effective.

In order to verify the accuracy of the automatic grammar correction system based on bidirectional LSTM and N-gram, the RNN based Sep2Sep model, LSTM-based Sep2Sep model, CNN-based Sep2Sep model, nested attention neural model, and deep context model are further introduced for comparative analysis. The syntax error correction effects of the six error correction models are shown in Table 6.

As can be seen from Table 6, the automatic syntax correction system based on bidirectional LSTM and N-gram in this study has the highest detection accuracy, with F_1 value of 46.00, and P value and R value of 66.78 and 35.09, respectively, significantly superior to other error correction models.

5 Overall architecture of syntax error correction system

When developing the model and modularity of the syntax checking system, we must fully consider the functional needs of the business requirements. We will divide subsystems and sub-modules according to the principle of independent development and modification, while ensuring the low coupling degree and relative independence between modules. The syntax error correction system architecture is shown in Figure 4.
(1) Syntax error correction module

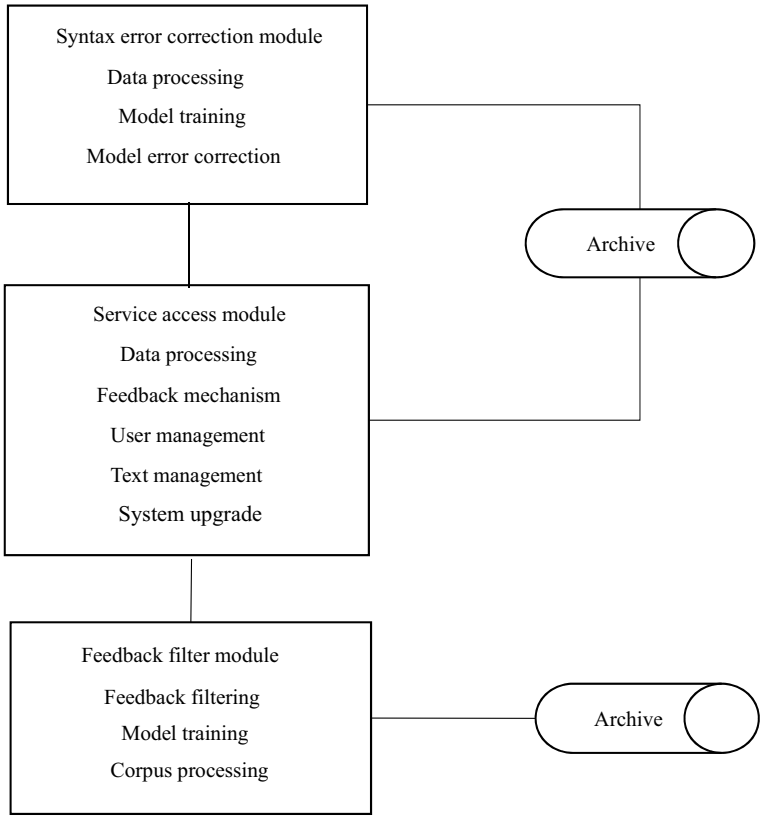

Figure 4: Architecture diagram of syntax error correction system.

The most important part of the system is the syntax error correction module, which has three pillars such as information sorting, algorithm optimization, and error correction, and its error correction function is its main focus. By filtering, selecting, and sorting out the original corpus, and generating the structure and rules, the standardized text can be obtained to meet the subsequent needs. Build and optimize machine learning methods to correct text accuracy by using large linguistic datasets to train models and check for errors. Finally, these optimized models are deployed and used as reference tools. Use a trained error correction model to check the grammatical correctness of sentences and fix sentences without spelling or grammatical errors. This component provides two sets of Thrift APIs, one to receive model training requests and the other to handle errors and perform model correction and validation.

(2) Service access module

The service access module consists of five parts: error correction service, information feedback mechanism, user operation and management, text processing and management, and system upgrade and update. The error correction service accepts error correction requests from users, uses specific algorithms and language processing techniques to correct syntax errors, and then sends the corrected version back. An information feedback mechanism that accepts the user's comments on the content of the web page and applies filters to select the most suitable solutions from the suggestions returned by the server side, and then pushes them to the user. User operation and management include identity authentication, category differentiation, and permission verification. During identity authentication, you need to check whether the user name and key are correct. Implement the role management mechanism according to the user's change suggestion action, only senior members and VIP customers are allowed to enjoy this function control. Text management allows you to store, modify, and export suggested texts in batches. The recommended processing times are recorded and a counter is set in the system update. Once accumulated to a predefined upper limit, it causes the syntax repair algorithm to be retrained to improve the self-learning ability of the whole system.

(3) Feedback filter module

This functional component mainly undertakes the implementation of feedback filtering and its service, including three key steps: corpus processing, model training, and feedback classification. After using the collected language samples for lexical analysis and grammatical decomposition, the expressiveness of the language model is optimized by controlling the sentence structure. Finally, the trained model is evaluated to obtain its performance data. The Thrift service interface provides feedback and filters out proposals that do not meet the requirements, and returns the accepted or rejected responses to the user.

6 Conclusion

To sum up, the automatic grammar correction system built in this study based on two-way LSTM and N-gram is feasible and effective, which can effectively improve the accuracy of English writing grammar error detection and enhance the effect of English grammar error correction. Experimental verification results show that the F_1 value, accuracy rate, and recall rate of the proposed method are higher than that of the traditional UIUC method and Corpus GEC method, and the grammar error correction of articles and prepositions is better. At the same time, compared with other models, the accuracy, recall rate, and F_1 value of this model are significantly improved. However, the effectiveness of the method proposed in this study is still to be improved in correcting the errors of nouns and verbs. In the next step, we will further improve the rationality of data construction, so that the constructed error samples are more consistent with the actual grammatical errors made by business English students. In addition, we will improve the structure of multi-task learning of linguistic features to further improve the GED task detection effect.

Funding information: Research on Business English Practical Teaching Based on the Integration of Industry and Education (Hubei Provincial Higher Education Teaching Research Project No. 2021543).

Author contribution: The author confirms the sole responsibility for the conception of the study, presented results and manuscript preparation.

Conflict of interest: No potential conflict of interest was reported by the author.

Data availability statement: All data generated or analyzed during this study are included in this article.

References

[1] Wang W, Li YA, Ma L, Qu QQ. Research on error detection technology of english writing based on recurrent neural network. 2021 International Conference on Big Data Analysis and Computer Science; 2021 (BDACS). p. 209–14.

[2] Chen HL. Design and application of English grammar error correction system based on deep learning. Secur Commun Network. 2021;2021:1–9.

[3] Zhou S, Liu W. English grammar error correction algorithm based on classification model. Complexity. 2021;2021:1–11.

[4] Wang J, Gu F. An automatic error correction method for English composition grammar based on multilayer perceptron. Math Probl Eng. 2022;2022.

[5] Solyman A, Wang Z, Tao Q, Elhag AAM, Zhang R, Mahmoud Z. Automatic Arabic grammatical error correction based on expectation-maximization routing and target-bidirectional agreement. Knowl Syst. 2022;241:108180.

[6] Yue X, Zhong Y. On the correction of errors in English grammar by deep learning. J Intell Syst. 2022;31(1):260–70.

[7] Shang HY, Huang JF, Chen HG. Chinese Grammar error correction model based on integrated parts-of-speech features in transformer. Comput Appl. 2022;42(S02):25–30.

[8] Qin M. A study on automatic correction of English grammar errors based on deep learning. J Intell Syst. 2022;31(1):672–80.

[9] Li KS, Shen JY, Gong C, Li ZH, Zhang M. Chinese grammar error correction based on pointer network incorporating confused-set knowledge. Chin Inf J. 2022.

[10] Sun XD, Yang DQ. Application of data augmentation strategies in English grammar correction. Comput Eng Appl. 2022;07:43–54.

[11] Solyman A, Wang Z, Tao Q. Proposed model for Arabic grammar error correction based on convolutional neural network. 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering; 2019 (ICCCEEE).

[12] Zhu J, Shi X, Zhang S. Machine learning-based grammar error detection method in English composition. Sci Program. 2021;2021:1–10.

[13] Dashtipour K, Gogate M, Li J, Jiang F, Kong B, Hussain A. A hybrid Persian sentiment analysis framework: integrating dependency grammar based rules and deep neural networks. Neurocomputing. 2019;380:1–32.

[14] Khurshid A, Latif S, Latif R. Transfer learning grammar for multilingual surface realisation. 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2). IEEE; 2021.

[15] Khadilkar A, Patil HY, Sundaramali G. Context sentences to single vector compression using convolutional transformers for deep learning based NLG tasks. 2021 International Conference on Computer Communication and Informatics; 2021 (ICCCI).

[16] Venkatraman SR, Anand A, Balasubramanian S, Sarma RR. Learning compositional structures for deep learning: why routing-by-agreement is necessary. ICLR 2021 Conference, Vienna, Austria, 2020.

[17] Lin N, Lin N, Lin X, Yang Z, Jiang S. A new evaluation method: evaluation data and metrics for Chinese grammar error correction. WOODSTOCK'18, June, 2018, El Paso, Texas, USA, 2022.

[18] Liu H, Xiang MA, Zhang L, He R. Aspect-based sentiment analysis model integrating match-LSTM network and grammatical distance. J Comput Appl. 2023;43(1):45–50.