Grammatical Error Correction Detection of English Conversational Pronunciation Under a Deep Learning Algorithm

Hang Yu

Department of College English, Zhejiang Yuexiu University, Shaoxing 312000, Zhejiang, China
E-mail: yh_hyu@hotmail.com

Received 26 June 2024; Accepted 17 September 2024

Abstract

Grammar correction in spoken English can enhance proficiency. This paper briefly introduces the gate recurrent unit (GRU) algorithm and its application in English speech recognition and grammatical error correction of speech recognition results. The GRU algorithm was firstly used to recognize English speech, then transform it into a text, and finally correct the English grammar of the text. Additionally, the attention mechanism was incorporated to enhance the performance of grammatical error correction. Subsequently, simulation experiments were conducted. Firstly, speech recognition and grammatical error correction were independently verified. The performance of the proposed algorithm in correcting grammatical errors in spoken English was evaluated using a self-built speech database. The results demonstrated that the proposed GRU-based algorithm yielded the best performance in independent speech recognition, independent grammatical error correction, and the overall spoken grammatical error correction. The contribution of this study lies in using the GRU algorithm to convert speech into text and perform grammar correction on the text, providing an effective reference for grammar correction in English communication.

Keywords: English, speaking, grammar correction, gate recurrent unit.

Journal of ICT Standardization, Vol. 12_2, 229–242.
doi: 10.13052/jicts2245-800X.1225

1 Introduction

Globalization has resulted in increased communication among individuals from different countries. As different countries have their languages, there is a need for an international common language to facilitate communication [1]. Fluency and accuracy in English conversation are crucial for effective cross-cultural communication, whether in everyday life or formal settings [2]. However, non-native English speakers often encounter mispronunciation and grammatical errors. Timely correction of these errors can enhance conversation fluency and accuracy. Deep learning algorithms have gained attention with their expanding application areas, including natural language processing, for instance, speech recognition and grammatical error correction [3]. Utilizing deep learning algorithms for grammatical correction of spoken English pronunciation is more efficient than manual error correction. They also assists users in independently training their spoken language skills. Wang et al. [4] put forward a multilayer perceptron-based approach for automatic grammatical error correction in English composition. The results demonstrated that the method achieved a grammatical error detection time of under 6 minutes, a recall ratio above 90%, and a detection error rate below 6%. Qin [5] aimed to improve the performance of the Transformer model in grammatical error correction by optimizing it using a generative adversarial network (GAN). Experimental findings confirmed the reliability of the improved Transformer model in automatically correcting English grammatical errors. Zhu et al. [6] designed a machine-learning-based method for detecting grammatical errors in English compositions and enhanced the detection algorithm's generalization through the utilization of the bidirectional encoder representations from transformers model. The aforementioned studies have analyzed English grammar correction and used different algorithms to detect grammatical errors. However, these studies focus on detecting grammatical errors in written text. This article, on the other hand, places emphasis on correcting English grammar in spoken language, aiming to improve the level of spoken English.

2 GRU Algorithm

Natural language is a kind of sequence data, and recurrent neural network (RNN) is a kind of deep learning algorithm suitable for processing sequence data [7], which utilizes data from historical moments in addition to input

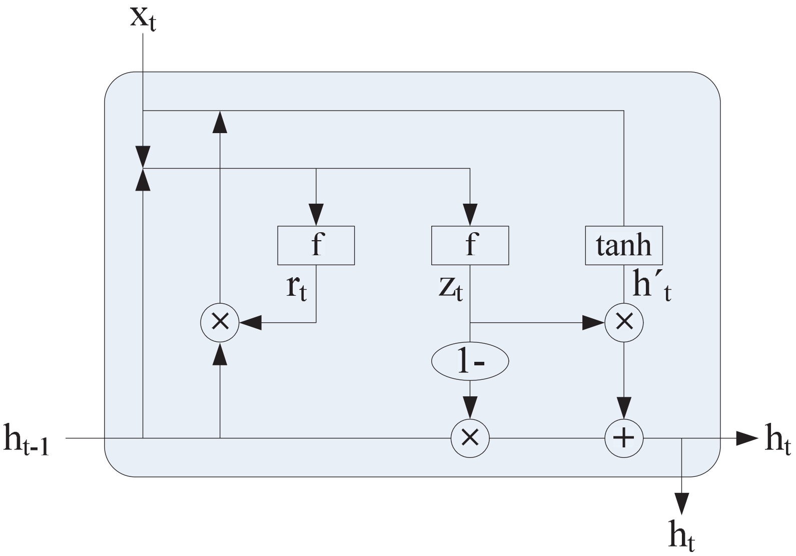

Figure 1 Basic structure of the GRU model.

data from the current moment. However, a traditional RNN has problems of vanishing and explosive gradients, which makes it challenging to process long sequence data. In order to solve this problem, LSTM introduces memory units in RNN and uses the gating mechanism to control the flow of information to realize the screening of edge information in long sequence data. GRU is a recurrent neural network [8] developed by slightly improving the gating mechanism of LSTM and combining the forgetting and input gates from LSTM. The basic structure of GRU is displayed in Figure 1, and its corresponding calculation formula is:

$$\begin{cases} z_t = f(\omega_z(h_{t-1}, x_t)) \\ r_t = f(\omega_r(h_{t-1}, x_t)) \\ h'_t = \tanh(\omega(r_t \times h_{t-1}, x_t)) \\ h_t = (1 - z_t) \times h_{t-1} + z_t \times h'_t \end{cases}, \tag{1}$$

where z_t is the update gate output [9], r_t is the reset gate output, ω_z and ω_r are the weights in the update and reset gates, x_t is the current input, $f(\ )$ is the activation function, h_{t-1} is the hidden state of the previous moment, h'_t is the temporary hidden state of the current moment, ω is the weight at the time of calculating h_t', h_t is the hidden state of the current moment, and $tahn(\ )$ is the function for calculating the current hidden state.

3 Speech Recognition for Spoken English

During English conversations, the accuracy of pronunciation and grammar directly impacts communication effectiveness. The process generally involves two steps when utilizing deep learning algorithms for detecting grammatical errors in spoken English. Firstly, the English pronunciation is recognized and then converted into English text. Secondly, grammatical error correction is performed on the converted English text [10].

For the speech recognition of English pronunciation, this paper adopts the gated recurrent unit-connectionist temporal classification (GRU-CTC) algorithm. This algorithm incorporates a CTC layer into the GRU algorithm to calculate the training loss during the training process. The following steps are followed.

(1) The input speech samples undergo preprocessing, including pre-emphasis, windowing, and framing [11].

(2) The features of each frame of the speech samples are extracted. This study uses Mel frequency cepstral coefficient (MFCC) features as the extracted features.

(3) The MFCC features of the speech samples are fed into the hidden layer of the GRU for forward computation, following the frame order. Equation (1) is used for forward computation in the hidden layer of the GRU.

(4) The output layer employs the softmax function to calculate the hidden state output from the hidden layer, obtaining the label probability distribution of text characters. Subsequently, it is determined whether the model is in the training phase. If not, the label distribution probability of text characters is decoded using the beam search algorithm, obtaining the corresponding text sequence for the speech.

(5) If it is in the training phase, the CTC layer utilizes both the true label sequences of speech samples and the probability distributions of labels generated by the output layer to calculate loss. The computational formula is:

$$L_{CTC} = -\ln \sum_u a_t^u b_t^u, \tag{2}$$

where L_{CTC} denotes the training loss, a_t^u is the sum of the forward probabilities of label u at the moment of t in the sequence of labels corresponding to the training samples and b_t^u is the sum of the backward probabilities of label u at the moment of t in the sequence of labels corresponding to the training samples.

(6) The calculated training loss is used to reversely adjust the GRU parameters. Return to step (3) until the training loss converges to a stable level or the training reaches a preset number of times.

4 Grammatical Error Correction for Speech Recognition Results

After the English speech is converted into English text sequences using the method above, grammatical error correction is conducted on the text. Grammatical error correction can be considered a categorization task: words in English texts are classified to determine if there are grammar errors and identify the types of errors.

Grammatical errors need to be judged in the context of the text as a whole, and individual words or phrases cannot effectively reflect the grammar. On the one hand, the English text itself is sequential data; on the other hand, the judgment of grammar needs to be combined with the context. Therefore, the GRU algorithm can be used to identify grammatical errors. The steps are show below.

(1) The English text to be recognized and preprocessed is input. The purpose of preprocessing is to remove the special characters from the text and to expand the abbreviated words or phrases. The length of the text is n, and x_i is the i-th word in the text.

(2) For grammatical error recognition of word x_i in an English text with a length of n, the text is divided into a left text sequence $(x_1, x_2, \ldots x_{i-1})$ and a right text sequence $(x_{i+1}, x_{i+2}, \ldots x_n)$ using x_i as the segmentation point.

(3) Word to vector (Word2vec) is used to vectorize the English text.

(4) The word text vectors $(x_1, x_2, \ldots x_{i-1})$ are input into the left-text GRU in sequential order, and the word text vectors $(x_{i+1}, x_{i+2}, \ldots x_n)$ are input into the right-text GRU in reverse order. Both the left-text GRU and the right-text GRU compute the input text vectors according to Equation (1) to obtain the hidden state sequences of the left and right texts.

(5) In order to enhance the recognition of grammatical errors and highlight the critical information in the context, this paper introduces the attention

mechanism [13] to assign weights to the hidden state sequences of the left and right texts. The formulas are:

$$\begin{cases} s(h_{l,t}) = h_{l,t}^T \cdot W_l \cdot h_{l,i-1} \\ s(h_{r,t}) = h_{r,t}^T \cdot W_r \cdot h_{r,i+1} \\ a_l(t) = \dfrac{\exp(s(h_{l,t}))}{\sum_{j=1}^{i-1} \exp(s(h_{l,j}))} \\ a_r(t) = \dfrac{\exp(s(h_{r,t}))}{\sum_{j=i+1}^{n} \exp(s(h_{r,j}))} \\ st_l = \left(\displaystyle\sum_{t=1}^{i-1} a_l(t) \cdot h_{l,t} \right) \oplus h_{l,i-1} \\ s_r = \left(\displaystyle\sum_{t=i+1}^{n} a_r(t) \cdot h_{r,t} \right) \oplus h_{r,i+1} \end{cases}, \tag{3}$$

where $h_{l,t}^T$ and $h_{r,t}^T$ are the hidden states of the left and right texts under the input of moment t (sequence), respectively, $h_{l,i-1}$ and $h_{r,i+1}$ are the last hidden state outputs of the of the left- and right-text GRU, respectively, $s(h_{l,t})$ and $s(h_{r,t})$ are the corresponding attention function scores, W_l and W_r are the weight matrices used to compute the attention scores, $a_l(t)$ and $a_r(t)$ are the weights of the hidden states under the input of the corresponding moments, st_l is the hidden state of the left text after the assignment of the weights, and st_r is the hidden state of the right text after the assignment of the weights.

(6) st_l is combined with st_r and input into the multilayer perceptron [14] for computation to obtain the computation results. The calculation formula is:

$$\begin{cases} M(x) = \omega \cdot x + b \\ o(x) = g(M(f(M(x)))) \end{cases}, \tag{4}$$

where $M(x)$ is the fully connected layer in the multilayer perceptual machine, ω is the weight of the fully connected layer, b is the bias of the fully connected layer, $f()$ is the activation function, the relu function in this paper, and $g()$ refers to the softmax function [15]. After the forward computation in the multilayer perceptual machine, the distribution probability of grammatical error types can be obtained, from which the type with the highest probability is selected as the output result.

5 Simulation Experiments

5.1 Experimental Data

Simulation experiments were conducted in three parts: independent validation of the speech recognition part, independent validation of the grammatical correction part, and overall validation of the combined two parts. The TIMIT dataset (https://catalog.ldc.upenn.edu) was used to validate the speech recognition part. This dataset has a sampling parameter of 16 kHz and provides manual segmentation and labeling at the phoneme level. The independent validation of the grammatical conjugation part was performed using the CoNLL-2014 grammatical error correction dataset (https://levyomer.wordpress.com). The overall validation of the combined two parts was conducted using a self-constructed dataset. This dataset was created by selecting English texts with different grammatical errors from the CoNLL-2014 dataset. The selected texts were read aloud, and the corresponding speech was recorded in a studio. The speech sampling parameter for this dataset was set at 16 kHz.

5.2 Experimental Setup

This paper used the GRU algorithm for grammatical error correction and recognition in English pronunciation. The algorithm was divided into two parts, speech recognition and grammatical error correction. The relevant parameters for these two parts obtained after conducting orthogonal testing are presented in Tables 1 and 2. For the grammatical error correction part, two GRUs namely left and right GRUs, were utilized, and the parameters for both GRUs were identical and set as Table 2.

Furthermore, a comparison with other algorithms was conducted to validate the effectiveness of the algorithm proposed in this paper. In the

Table 1 Parameters related to the speech recognition part

Parameter	Setup	Parameter	Setup
Framing window	Hamming window, a frame length of 15 ms, and a frame shift of 2 ms	MFCC feature	39-dimensional
The input layer of GRU	39 nodes	GRU hidden layer	256 nodes, the sigmoid activation function
The output layer of GRU	50 nodes, the softmax activation function	Maximum number of training sessions	300

Table 2 Parameters related to the grammatical error correction part

Parameter	Setup	Parameter	Setup
Word2vec vector dimension	300-dimensional	The input layer of GRU	300 nodes
The hidden layer of GRU	512 nodes, the sigmoid activation function	Multilayer perceptron	Two fully connected layers
Training method	SGD	Maximum number of training sessions	300 times

comparison, the GRU structure used in the proposed algorithm was replaced with RNN and LSTM structures. Since all three algorithms are recurrent neural networks, they share the same basic parameters.

5.3 Evaluation Criteria

For the evaluation criteria of the speech recognition part, this paper adopted the word error rate to measure; for the evaluation criteria of the grammatical error correction part, this paper adopted the confusion matrix and maximum matching methods to measure. The evaluation indicators of the confusion matrix method are precision, recall rate, and F value. The maximum matching method can compute the editing distance between the prediction result and the actual result from the phrase level and the indicators it calculates are similar to those of the confusion matrix method. Precision P_{M^2} is calculated using the maximum matching method:

$$P_{M^2} = \frac{\sum_{i=1}^{n} |e_i \bigcap g_i|}{\sum_{i=1}^{n} |e_i|}, \tag{5}$$

where $e_i \bigcap g_i$ is the maximum match between the result forecasted by the error correction algorithm and the actual result, $e_i \bigcap g_i = \{e \in e_i | \exists g \in g_i, match(g, e)\}$, g_i is the actual result of the error correction, and e_i is the result forecasted by the error correction algorithm. Recall rate R_{M^2} is computed using the maximum matching method:

$$R_{M^2} = \frac{\sum_{i=1}^{n} |e_i \bigcap g_i|}{\sum_{i=1}^{n} |g_i|}. \tag{6}$$

Composite indicator F_{M^2} is computed using the maximum matching method:

$$F_{M^2} = \frac{2 \cdot P_{M^2} \cdot R_{M^2}}{P_{M^2} + R_{M^2}}. \tag{7}$$

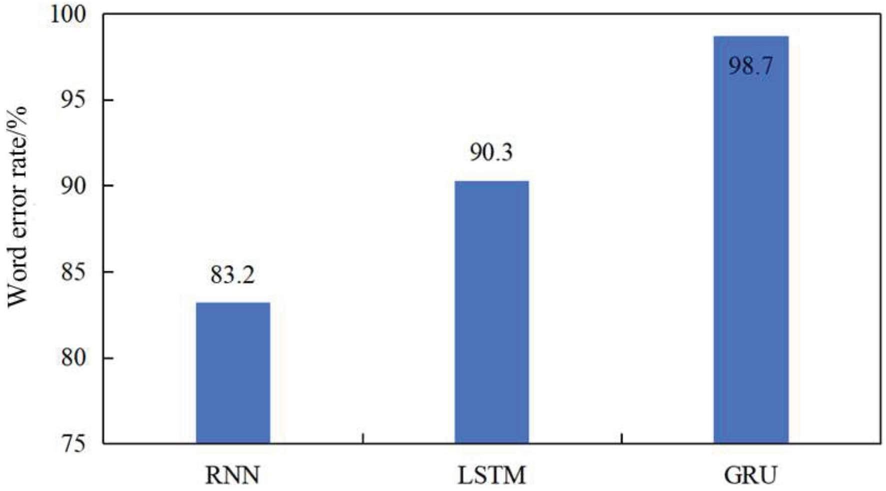

Figure 2 Performance of three algorithms for speech recognition.

Table 3 Performance of three algorithms for grammatical error correction

	Confusion Matrix Method			Maximum Matching Method		
Evaluation Criteria	P	R	F	P_{M^2}	R_{M^2}	F_{M^2}
RNN	65.3%	65.4%	65.3%	62.6%	63.9%	63.2%
LSTM	88.9%	87.5%	88.2%	86.8%	85.7%	86.2%
GRU	96.1%	94.3%	95.2%	97.3%	95.6%	96.4%

5.4 Experimental Results

As depicted in Figure 2, the word error rates for the RNN-based, LSTM-based, and GRU-based speech recognition algorithms were 83.2%, 90.3%, and 98.7%, respectively. It is evident that the algorithm utilizing GRU exhibited the best speech recognition performance, followed by the LSTM-based algorithm and the RNN-based algorithm.

Table 3 presents the performance of the algorithms in correcting grammatical errors in speech recognition text. The data in the table indicated that both evaluation indicators, namely the confusion matrix approach and the maximum matching approach, consistently demonstrated that the GRU-based algorithm outperformed the others in terms of grammatical error correction. The LSTM-based algorithm exhibited the second-best performance, while the RNN-based algorithm performed the least effectively.

Table 4 Partial results of the algorithms for grammatical error correction of spoken English pronunciation

Algorithm	Pronunciation Example	Type of Grammatical Error	Speech Recognition Result	Grammatical Error Recognition Result
RNN	The weather **is** good yesterday, which is suitable for outdoor sports, picnics or leisure activities at home.	Incorrect verb tense	The **weether** is good yesterday, which is suitable for **outdor** sports, picnics or leisure **activity** at home.	Article error
LSTM			The weather is **god** yesterday, which is suitable for outdoor sports, picnics or leisure activities at home.	Prepositional error
GRU			The weather **is** good yesterday, which is suitable for outdoor sports, picnics or leisure activities at home.	Incorrect verb tense

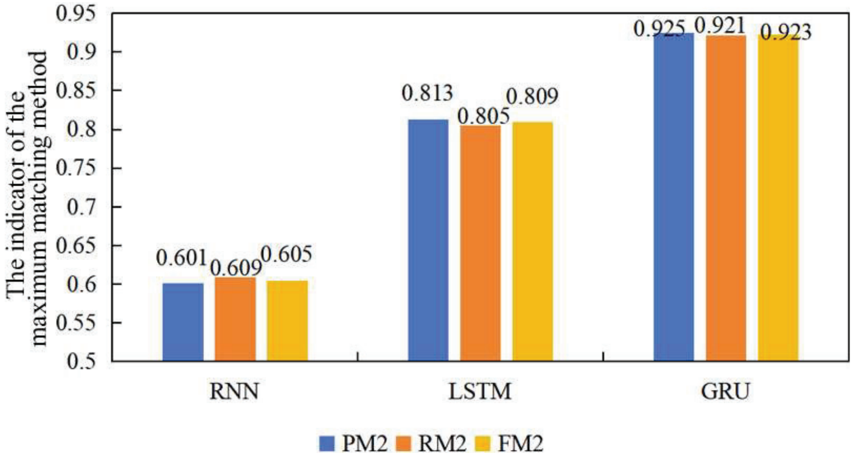

Figure 3 Grammatical error correction performance of three algorithms on a self-built speech database.

Following the independent verification of the speech recognition and grammar error correction parts, the spoken grammar error correction performance of the algorithm was verified using a self-built speech database. Table 4 shows some recognition results, while Figure 3 presents the overall performance indicator. It was seen from Table 4 that the RNN-based algorithm produced more errors in recognizing speech and made a mistake in identifying the grammatical error type. The LSTM-based algorithm produced a relatively lower number of errors in recognizing speech but made a mistake in identifying grammatical error types. The GRU-based algorithm accurately recognized pronunciation and made a precise judgment on the type of the grammatical error. Figure 3 indicates that the algorithm utilizing GRU achieved the best error correction performance, followed by the LSTM-based algorithm. In contrast, the RNN-based algorithm performed the least effectively. These findings aligned with the results presented in Table 4.

6 Conclusion

This paper briefly introduces the GRU algorithm and its application in English speech recognition and grammatical error correction of speech recognition results. Additionally, the attention mechanism was incorporated to enhance the performance of grammatical error correction. The simulation experiments verified both the speech recognition and grammatical error correction parts and tested the overall performance of grammatical error correction of spoken English using a self-built speech database. Moreover, the designed algorithm was compared with the RNN and LSTM algorithms. The GRU algorithm demonstrated the highest performance in speech recognition, followed by the LSTM-based algorithm, while the RNN-based algorithm performed the worst. Both evaluation indicators, namely the confusion matrix and maximum matching approaches, consistently indicated that the GRU-based algorithm outperformed the others. The LSTM-based algorithm achieved the second-best performance, while the RNN-based algorithm exhibited the least effective performance. The algorithm utilizing GRU achieved the best error correction performance, followed by the LSTM-based algorithm in second place, while the RNN-based algorithm exhibited comparatively lower effectiveness. These findings aligned with the outcomes presented in Table 4. The limitation of this article is that it only focuses on correcting grammatical errors in terms of individual words in spoken English. Therefore, the future research direction is to enhance the recognition of grammatical errors in phrases and entire sentences.

References

[1] A. Rozovskaya, D. Roth, 'Grammar Error Correction in Morphologically Rich Languages: The Case of Russian', Trans. Assoc. Comput. Linguist., 7(1), pp. 1–17, 2019.

[2] S. M. Hussein, 'The Correlation between Error Correction and Grammar Accuracy in Second Language Writing', International Journal of Psychosocial Rehabilitation, 24(5), pp. 2980–2990, 2020.

[3] J. W. Lee, 'A comparison study on EFL learner and teacher perception of grammar instruction and error correction', English Teach., 73(2), pp. 139–159, 2018.

[4] J. Wang, F. Gu, 'An Automatic Error Correction Method for English Composition Grammar Based on Multilayer Perceptron', Math. Probl. Eng., 2022(Pt.29), pp. 1.1–1.7, 2022.

[5] M. Qin, 'A study on automatic correction of English grammar errors based on deep learning', Journal of Intelligent Systems, 31(1), pp. 672–680, 2022.

[6] J. Zhu, X. Shi, S. Zhang, 'Machine Learning-Based Grammar Error Detection Method in English Composition', Sci. Programming, 2021(Pt.13), pp. 4213791.1–4213791.10, 2021.

[7] G. Eckstein, 'Grammar Correction in the Writing Centre: Expectations and Experiences of Monolingual and Multilingual Writers', Can. Mod. Lang. Rev., 72(3), pp. 1–23, 2016.

[8] T. vor der Brück, 'A Probabilistic Approach to Error Detection & Correction for Tree-Mapping Grammars', Prague Bull. Math. Linguist., 111(1), pp. 97–112, 2018.

[9] C. Park, Y. Yang, C. Lee, H. Lim, 'Comparison of the evaluation metrics for Neural Grammatical Error Correction with Overcorrection', IEEE Access, 8, pp. 106264–106272, 2020.

[10] M. H. Alhumsi, S. Belhassen, 'The Challenges of Developing a Living Arabic Phonetic Dictionary for Speech Recognition System: A Literature Review', Adv. J. Soc. Sci., 8(1), pp. 164–170, 2021.

[11] H. A. Alsayadi, A. A. Abdelhamid, I. Hegazy, Z. T. Fayed, 'Arabic speech recognition using end-to-end deep learning', IET Signal Process., 15(8), pp. 521–534, 2021.

[12] C. Long, S. Wang, 'Music classroom assistant teaching system based on intelligent speech recognition', J. Intell. Fuzzy Syst., 2021(14), pp. 1–10, 2021.

[13] S. Tripathi, V. Kansal, ‘Machine Translation Evaluation: Unveiling the Role of Dense Sentence Vector Embedding for Morphologically Rich Language’, Int. J. Pattern Recogn., 34(1), pp. 2059001.1–2059001.18, 2020.

[14] R. Hos, M. Kekec, ‘Unpacking the Discrepancy between Learner and Teacher Beliefs: What should be the Role of Grammar in Language Classes?’, Eur. Educ. Res. J., 4(2), pp. 70–76, 2015.

[15] X. Chen, ‘Synthetic Network and Search Filter Algorithm in English Oral Duplicate Correction Map’, Complexity, 2021(Pt.12), pp. 9960101-1–9960101-12, 2021.

Biography

Hang Yu was born in Shaoxing, Zhejiang, P.R. China, in 1985. Now, she works in Zhejiang Yuexiu University. Her research interests include applied linguistics and English language teaching.