IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 June 2024, accepted 6 August 2024, date of publication 12 August 2024, date of current version 26 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3442432

RESEARCH ARTICLE

An Intelligent Error Detection Model for Machine Translation Using Composite Neural Network-Based Semantic Perception

YAOXI WU[1] AND QIAO LIANG[2]
[1]School of Foreign Studies and Trade, Chongqing College of International Business and Economics, Chongqing 401520, China
[2]School of Mathematics and Computer Science, Chongqing College of International Business and Economics, Chongqing 401520, China
Corresponding author: Qiao Liang (liangqiao1011@ccibe.edu.cn)

ABSTRACT Although machine translation has received great progress in recent years, machine translation results usually existed some errors due to the complex relationship between sentence structure and semantics. Currently, the automatic error detection techniques towards machine translation errors have not been deeply investigated. To deal with the challenge, this paper proposes an intelligent error detection model for machine translation using composite neural network-based semantic perception. Firstly, integrating attention mechanism into Bi-GRU encoder can effectively learn contextual information of sentences and generate high-quality global feature representations. Then, multiscale CNN can extract local features at different scales, thereby capturing finer grained semantic information. Experiments are conducted on datasets containing a large amount of English text and machine translation errors, in which the proposed model is compared several benchmark methods. The experimental results indicate that the proposal has achieved significant improvements in machine translation error detection tasks. It comparison, it can more accurately identify common problems such as grammar errors, semantic errors, and word spelling errors in translation results, verifying its effectiveness and practicality.

INDEX TERMS Error detection, semantic modeling, intelligent perception, composite neural network.

I. INTRODUCTION

In today's globalized society, effective communication between languages has become particularly important, and English machine translation, as a key technology for cross-cultural and multilingual communication, has been widely applied in various fields [1]. However, even the most advanced machine translation systems are inevitably prone to some human or automated errors. These translation errors may lead to distortion, misunderstanding, or even loss of important details, causing unnecessary inconvenience and misleading for users [2]. Therefore, in order to solve this problem and better automatically detect and correct machine translation errors, scholars from various countries continue to explore and innovate, seeking a new method for detecting errors in English machine translation to improve the quality and reliability of machine translation systems.

In past research, a common method was to use statistical machine translation error detection. These models use phrase based methods to infer the best translation and use various features to determine whether the translation is accurate [3]. However, traditional machine translation error detection methods often rely on manually set features and rules when identifying errors. The limitation of this method is that it is difficult to cope with the complexity and diversity of language, which in turn limits its overall performance [4]. With the rise of deep learning, researchers have begun to use neural networks to improve the performance of machine translation error detection.

One important research direction is to use deep semantic learning frameworks to extract global and local features [5]. Global features help capture the overall semantic information of a sentence, while local features can better represent

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

the details and structure of the sentence [6]. The Bi-GRU encoder can effectively capture bidirectional contextual information of sentences and transform it into representations with rich semantics. Multi scale CNN can extract local features at different scales, thereby better capturing subtle differences in sentences [7]. By integrating Bi-GRU encoder and multi-scale CNN, this deep semantic learning framework can simultaneously consider global and local features, thereby more accurately detecting errors in English machine translation. Experimental results have shown that this method achieves excellent performance on multiple datasets and has significant improvements compared to traditional statistical machine translation models. This method has the potential to improve the quality of machine translation and can provide useful references for research in related fields. This study has made contributions to the literature in the following areas:

(1) This study uses twin RNNs to train non English word vectors, map them to the basic semantic space, learn word vectors from other languages, and detect errors in English machine translation.

(2) This study proposes a deep semantic learning framework that integrates Bi-GRU encoder and multi-scale CNN. Integrating global and local features to improve the performance of machine translation error detection.

(3) By integrating global and local features, it is possible to more comprehensively represent the semantic information of sentences, better capture the semantic information in English translation, and thus improve the accuracy and performance of error detection.

(4) The research results of this article provide new ideas and methods, which can provide new research directions for scholars and engineers in the field of English machine translation detection, and also promote the development of this field and improve the accuracy and efficiency of machine translation error detection.

II. RELATED WORK AND PRELIMINARIES

A. APPLICATION OF DEEP SEMANTIC LEARNING IN MACHINE TRANSLATION ERROR DETECTION

Deep semantic learning has important applications in error detection in English machine translation. Traditional machine translation systems mainly rely on statistics and rules for translation, usually performing well in sentence structure and grammar rules, but there are certain limitations in semantic understanding and translation accuracy [8]. Deep semantic learning can capture the semantic information of sentences through neural network models, thereby improving the accuracy and naturalness of machine translation. Specifically, in English machine translation error detection, deep semantic learning can have the following applications:

(1) Error type recognition: Deep learning models can learn feature representations of different types of errors, such as lexical errors, grammatical errors, or semantic errors. By training the model, errors in translated sentences can be automatically classified and corresponding correction suggestions can be provided [9].

(2) Error localization: By learning to capture contextual information between different words or phrases in a sentence, it is possible to accurately locate the location of the error. By locating errors, it is easier to correct them and improve the translation quality of machine translation [10].

(3) Error correction: Learn to translate the semantic representation of sentences in order to generate more accurate and natural translation results. By training the model, incorrect translation parts can be automatically corrected to more accurate expressions [11].

Overall, the application of deep semantic learning in English machine translation error detection is mainly reflected in improving word sense disambiguation, solving long-distance dependencies, and correcting structural errors. By training and optimizing deep learning models, the translation quality of machine translation systems can be significantly improved.

B. BUILDING THE BASIC SEMANTIC SPACE OF TWIN RNNs

To achieve information sharing and mapping across language spaces, it is necessary to establish a basic semantic space that contains rich language elements from multiple languages [12]. Firstly, traditional word vector models such as Word2vec's Skip gram model are used to obtain English word vector representations by training English monolingual corpora [13]. Using twin RNNs to train non-English word vectors to generate word vector representations for other languages, and projecting them into the basic semantic space, can learn and generate word vector representations for other languages within the constraint space, thereby detecting errors in English machine translation [14]. This article takes bilingual Chinese and English as an example, using a twin RNN to train the binary classification task of bilingual sentence pairs to learn cross linguistic word vectors. The structure of the twin RNN is shown in Figure 1.

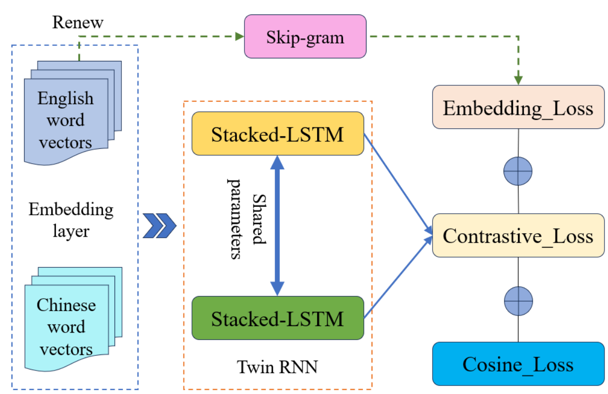

FIGURE 1. Twin RNN network structure.

The twin RNN consists of two stacked LSTM neural networks, which share weight parameters. Receive Chinese and

English word vector sequences as input, respectively. Fix the English word vector as a pre trained shared distributed expression space, and dynamically update the Chinese word vector by jointly training parallel sentences and aligned word pairs in Chinese and English, helping us learn better cross lingual word vectors [15].

This article calculates the cosine distance of distributed expressions of Chinese and English word pairs to make semantically identical words in cross lingual word vectors as close as possible to the shared embedding space. The cosine of Chinese and English word pairs and the calculation loss function are:

$$L_{\cos} = -\sum_{i=1}^{R} \frac{v'_{x_i} \cdot v'_{y_i}}{\left\| v'_{x_i} \right\|_2 \left\| v'_{y_i} \right\|_2} \tag{1}$$

In the formula, $V_X = \{v_{x1}, v_{x2}, \ldots, v_{xn}\}$, $V_Y = \{v_{y1}, v_{y2}, \ldots, v_{yn}\}$ respectively represent the word vectors corresponding to English and Chinese sequence words, $<x', y'>$ represents the alignment of a certain word x 'in sequence X with a certain word y' in sequence Y, and $<v'_x, v'_y>$ are the corresponding word vectors. Assuming that there are a total of R pairs of aligned words in parallel sentences between Chinese and English. R represents the logarithm of the maximum alignment controlled by the alignment probability threshold, where $R < n$ and $R < m$.

Two channels receive Chinese and English sentence sequences for calculation. For parallel corpora, it is necessary to ensure that the Chinese and English word vectors are similar, the sentence meanings are the same, and the network structure and parameters are the same, to ensure that the twin network learns consistent semantic features [16]. Otherwise, the output of each network will be different. The last layer of twin RNN uses a 2-class contrastive loss function. The mathematical expression of Stacked-LSTM is:

$$h_{k,t}^{(X)} = LSTM_k(X_t, h_{k,t-k}) \tag{2}$$

In the formula, k represents the number of layers in the twin network, with a value range of [1] and [3], and $h_{k,t}^{(X)}$ represents the implicit state of the nth layer in the English sequence at time t. Layer by layer LSTM can gradually extract abstract features from sentences.

Through multiple levels of processing, each layer focuses on learning features at different levels. The first layer mainly learns the vocabulary level features of words, while the second layer learns the relationships between vocabulary and sentences, such as the composition and dependency relationships of phrases. The third layer learns the features of the sentence more abstractly, such as the implicit meaning within the sentence [17]. The entire process can be seen as an abstraction of words, phrases, sentences, and semantic levels, ultimately generating a feature vector. In addition, the two outputs of the twin neural network are compared by comparing the loss function, and ultimately output a state value.

III. METHODOLOGY

The deep semantic learning framework proposed in this paper consists of two parts: Bi-GRU encoder and multi-scale CNN. Firstly, integrating the attention mechanism into the Bi-GRU encoder can effectively learn the contextual information of sentences and generate high-quality global feature representations. Multi-scale CNNS can then extract local features at different scales to capture more fine-grained semantic information, which is very effective for dealing with complex linguistic phenomena. By fusing information from these two parts, the framework is able to understand the text content more comprehensively, thus improving the accuracy of error detection.

A. GLOBAL FEATURE MODELING BASED ON Bi-GRU ENCODER

For global feature learning, we use bi-gated cycle units (BI-GRUs) as encoders. Bi-GRU is able to capture both forward and backward information of a sentence to more fully understand the context of the sentence. However, simple Bi-GRU coding may not highlight the key information in the sentence. Therefore, we introduce attention mechanisms to enhance the coding capabilities of Bi-GRU. The attention mechanism allows the model to focus on the key parts of the sentence by assigning different weights to the hidden states of each time step.

1) Bi-GRU MODEL EXTRACTS SEMANTIC INFORMATION

The Bi-GRU model is a sequence model of bidirectional gated recurrent units. In English machine translation detection, it is used to encode the input source language sentence into a fixed length hidden vector representation, capturing the semantic information of the source language sentence [18]. The advantage of the Bi-GRU model is that it can simultaneously consider both forward and backward information in sentences. Two independent GRU layers have their own hidden states, propagating information forward and backward respectively [19]. Finally, the outputs of the two GRU layers are concatenated at each time step to form a global feature representation of the source language sentence. Extracting semantic information through the Bi-GRU model can better capture contextual information in sentences, thereby improving the accuracy and fluency of machine translation [20]. Therefore, the Bi-GRU neural network calculates the state of the hidden layers before and after time t as follows:

$$h_t^+ = GRU(x_t, h_{t-1}^+) \tag{3}$$

$$h_t^- = GRU(x_t, h_{t-1}^-) \tag{4}$$

In the formula, $t = 1,2, T$. X represents the input value of the input layer, and h^+ and h^- respectively represent the forward and backward outputs obtained through forward and backward processing, which can extract semantic information of the context. So the output of the hidden layer at time t is a cascade of h^+ and h^-, which can be expressed as:

$$h_t = w_t h_t^+ + v_t h_t^- + b_t \tag{5}$$

In the formula, w_t and v_t are the weights of the forward and backward GRU hidden layer states at time t, respectively, and b_t is the bias corresponding to the hidden layer state at time t. The introduction of these parameters expands the number of free parameters in the model.

The update gate z_t of the model refers to the degree of influence of the current state at time t and the previous state t-1 on the output result. The reset gate and update gate are calculated as:

$$z_t = \sigma(w_{z,}[h_{t-1}, x_t]) \tag{6}$$

$$r_t = \sigma(w_{r,}[h_{t-1}, x_t]) \tag{7}$$

In the formula, h_{t-1} represents the hidden state of the previous moment, σ the sigmoid function can control the value of h_t within the range of (−1,1), where w_z and w_r are the weights of the reset gate and update gate, respectively [21]. These weights play a very important role in the judgment of the model. In order to obtain weights that meet the requirements, it is necessary to update these weights through training.

When the Bi-GRU model has n layers, the h_n^+ and h_i^- are the forward and reverse hidden layer states, respectively. Then, by combining these two states together, the final hidden layer state hfinal of the Bi-GRU algorithm can be obtained as:

$$h_{final} = h_n^+ \oplus h_i^- \tag{8}$$

2) INTEGRATING ATTENTION MECHANISMS TO CALCULATE ATTENTION WEIGHTS OF SOURCE SENTENCES

In order to further improve the attention modeling of the source sentence, we introduced an attention mechanism to calculate the attention weights of the source sentence. The basic structure of the Bi-GRU encoder after introducing the attention mechanism is shown in Figure 2. The attention mechanism allows the model to focus its attention on important parts of the source sentence, ignoring parts that are not important for translation. Our attention mechanism is based on the output of the bidirectional GRU encoder, and obtains the attention weight of the source sentence by calculating the similarity between the hidden state of the source sentence and the hidden state of the decoder at each time step [22].

Specifically, a weighted average method based on attention mechanism was adopted, which weights all positions in the source statement according to their similarity with the target statement positions, and then introduces the weighted average vector as a global feature into the model [23]. Therefore, by comparing different parts of the source sentence with the current hidden state of the decoder, we can dynamically adjust the attention of the source sentence. By using the Bi-GRU fusion attention mechanism to calculate the attention weights of the source sentence, our deep semantic learning framework can more accurately capture the global and local features of the source sentence, thereby improving the error detection performance of English machine translation.

The attention probability distribution of the hidden state hn in the Bi-GRU model is:

$$b_n = \frac{exp(h'_n)}{\sum_{i=0}^{N} exp(h'_i)} \tag{9}$$

where,

$$h'_n = h_n^T U h_{final} \tag{10}$$

In the formula, N is the number of tokens input, and U is the weight matrix.

After calculating the attention mechanism, data normalization is also necessary, mainly to unify the numerical range of attention weights. Normalization can map attention weights onto a unified scale, which can improve the convergence speed of the model and reduce the differences in attention weights between different samples. The usual normalization method is to use the *softmax* function to convert attention weights into probability distributions, so that the sum of all weights is 1. After normalizing each layer, obtain new hidden layer states:

$$h_{new} = f[\frac{g}{\sigma_t}\Theta(a_t - \mu_t) + d] \tag{11}$$

$$b_t = W_{hh}h_{t-1} + W_{xh}x_t \tag{12}$$

In the formula, g and d represent the gain matrix and offset matrix, σ Indicates the use of the softmax function, where W_{hh} and W_{xh} are the weight matrices between the hidden layer and the input layer and the hidden layer, respectively.

B. EXTRACTING LOCAL FEATURES AND CALCULATING WEIGHTS BASED ON CNN

In local feature learning, we use multi-scale CNN to capture rich semantic information in context. CNN extracts local features effectively through convolution operations, while multi-scale convolution kernel provides comprehensive and detailed description of local features. Therefore, we design a CNN architecture integrating convolution kernels of different sizes to extract multi-scale local features in parallel. After that, the Minimum Error Rate Training (MERT) algorithm is introduced to optimize the feature weights and dynamically adjust the translation weights by minimizing the translation error rate. This strategy improves the model's understanding of the text and translation accuracy, thus improving the overall translation quality.

1) MULTI SCALE CNN EXTRACTION OF LOCAL FEATURES

CNN performs well in the field of image processing, but may face two challenges in natural language processing tasks. Firstly, natural language has word order and contextual dependencies, which are different from the spatial locality properties in image processing [24]. Secondly, sentence lengths in natural language are often not fixed, which can lead to information loss or padding issues when using traditional fixed size sliding windows for convolution. To overcome these issues, I introduced a multi-scale CNN to extract local

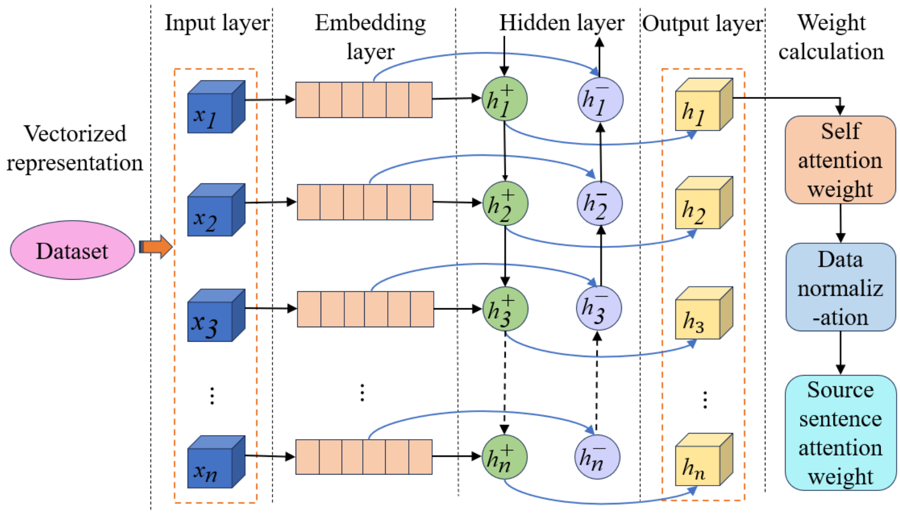

FIGURE 2. Basic structure of Bi-GRU encoder.

features. Specifically, a multi-scale convolution operation was used to process sentences of different lengths. By using convolution kernels of different sizes to capture semantic information from different ranges, I am able to better adapt to sentences of different lengths. In this way, the model can simultaneously capture local features at different scales. Due to the fact that semantic correlation often has multiple scales of expression, this multi-scale convolution operation helps to improve the model's grasp of semantic information.

The convolutional layer, pooling layer, and fully connected layer of CNN network can be used for deep level local feature extraction. Among them, the convolutional layer is one of the most basic building units, which applies multiple convolution kernels to input data for convolution operations to extract features of images or texts. The final representation of the feature map obtained through the activation function is calculated as follows:

$$c_i = f(w^* UO_{i:i+t-1} + b) \tag{13}$$

In the formula, c_i represents the operation result of the convolution kernel, w represents its weight matrix, b is the bias term of the convolution kernel, $UO_{i:i+t-1}$ represents the vector matrix, * is the operation symbol of the convolution process, $f(x)$ represents the activation function, and this article uses the Rule activation function [25]. All features obtained after convolution operation are represented as:

$$C = \{C_1, C_2, \ldots, C_{d-t+1}\} \tag{14}$$

Adding pooling layers after convolutional layers can reduce computational complexity and preserve important features. This article uses max pooling to select the maximum value within the pooling window as a new feature, reducing data dimensions and preserving salient features. The stacking of convolutional and pooling layers gradually extracts more important and abstract features from the data, which are used for subsequent fully connected layers or classifiers for classification, recognition, and other tasks. The maximum pooling operation is calculated as follows:

$$pooling_{u,v}^{max} = \frac{1}{|\Omega_{u,v}|} \sum\nolimits_{i,j \in \Omega_{u,v}} a_{i,j} \tag{15}$$

Among them, a_{ij} is the activation value of the pooling area; i,j is an index representation; $\Omega_{u,v}$ is the corresponding pooling region on the feature map.

In a fully connected layer, the output of the previous layer will be input into this layer. Multiply with the initial weight matrix of this layer and add bias, while adopting a random dropout strategy. Specifically, the dropout strategy will randomly shut down some neurons in the neural network during each iteration, using only a subset of neurons to train the model and obtain the values of the weight matrix w and bias parameter b [26]. In fact, this is equivalent to training on different neural networks, which can reduce the dependency between neurons, thereby helping to reduce the impact of overfitting problems and enhancing the network's generalization ability.

2) CONTEXT INFORMATION WEIGHT CALCULATION

After extracting local features, use MERT algorithm to calculate the weight of contextual information. The MERT algorithm is a commonly used optimization method that selects the best translation by minimizing the error rate. In this study, the MERT algorithm can dynamically adjust the weights of different contextual information in machine translation based on their importance, making the model more accurate [27].

Specifically, first compare and count the number of errors between the source statement and the output sentence of the English machine translation system. Use function $E(p_1, p)$ to represent the number of errors calculated and counted. The purpose of this algorithm is to obtain the target sentence with the minimum total number of errors from a series of candidate corrections, and the algorithm's parameters are the optimal ones [28]. When calculating the number of errors, the system considers the candidate sentence with the highest score for correction and the standard sentence for calculation. The calculation is as follows:

$$\lambda_1^M = argmin\left\{\sum\nolimits_{s=1}^{s} E(r_s, e(f_s; \lambda_1^M))\right\} \tag{16}$$

In the formula, r_s represents the output sentence, and t represents the standard sentence. F_s represents the s-th input sentence to be corrected (translated). Where $e(\cdot)$ is:

$$e(f_s; \lambda_1^M) = argmax\left\{\sum\nolimits_{m=1}^{M} \lambda_m h_m(e \mid Of_s)\right\} \tag{17}$$

$$score(T, S) = \sum\nolimits_{m=1}^{|M|} \lambda_i f_i(T, S) \tag{18}$$

In the formula λ_m represents the obtained local feature weights, $|M|$ represents the number of features.

C. ENGLISH MACHINE TRANSLATION ERROR DETECTION MODEL INTEGRATING GLOBAL AND LOCAL FEATURES

In order to make error detection in English machine translation more accurate, the global and local features extracted based on Bi-GRU encoder and CNN in the previous text are fused to obtain more comprehensive features for accurate and effective error detection. This article constructs a new English machine translation error detection model that integrates global and local features by integrating two information attention weights using a gated unit structure based on the deep semantic learning framework. The basic structure of the model is shown in Figure 4, and the relevant pseudocode is shown in Algorithm 1.

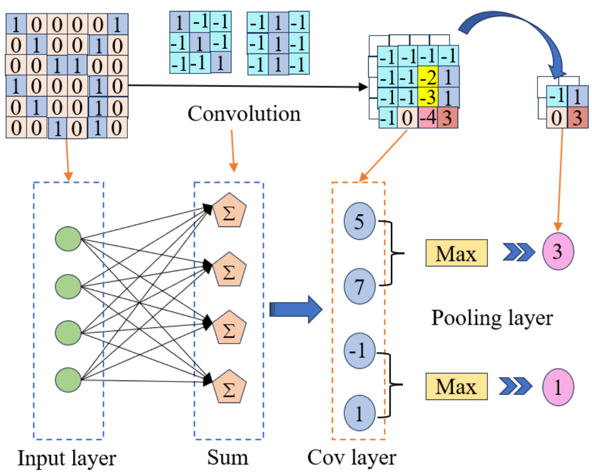

FIGURE 3. CNN Network Structure Diagram.

In English machine translation error detection, gating units are used to integrate the attention weights of the source sentence and context. By linearly combining the attention weights of the source sentence and context, a weighted sum can be obtained to represent the comprehensive information. Then, the two attention weights are classified using the softmax function and the probability of candidate sentences is

---

Algorithm 1 The Intelligent Error Detection Model For Machine Translation Results Using Composite Neural Network

---

Input: The number of layers in the twin network k, the implicit state $h_{k,t}^{(X)}$ of the English sequence, the forward and backward outputs h_n^+ and h_i^- of Bi- GRU, the number of tokens input N, the attention weight of the source sentence c, the attention weight of the context C, and the integrated feature output Y_t at time t

Constructing Basic Semantic Space Using Twin RNNs

Calculate the cosine and loss function of bilingual word pairs using eq-1

for all $t = 1$ to T **do**

  $h_{k,t}^{(X)} = LSTM_k(X_t, h_{k,t-k})$

  The value of k introduced is between [1], [3]

  Calculate h_{final} using eq-8

    for $i = 1 : N$

      $h_{final} = h_n^+ \mathring{A} h_i^-$

    if integrating features using a gate control unit structure

      $Q_t = Y_t + c_t + L_t QC_t + c_{t-1}$

      Obtain the source sentence and contextual features

    else

      Output the probability of candidate sentences

    end for

end for

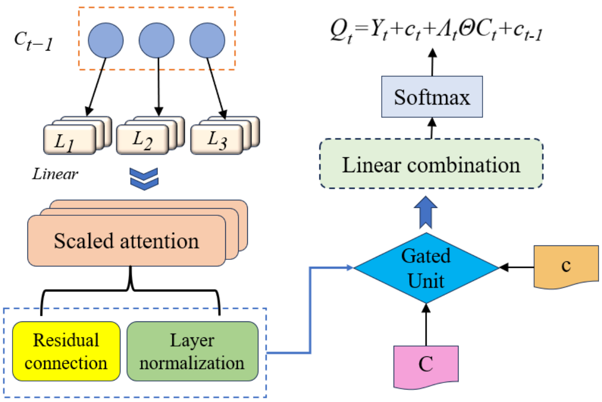

FIGURE 4. English machine translation error detection model integrating global and local features.

obtained. This process ensures the integration of source sentence and contextual information, and reasonably classifies the weights of the two types of information. The final output calculation is as follows:

$$Q_t = Y_t + c_t + \Lambda_t \Theta C_t + c_{t-1} \tag{19}$$

In the formula, c represents the attention weight of the source sentence, C represents the attention weight of the context, and Y_t and c_{t-1} represent the outputs at time t and t-1, respectively; Θ for Hadamard product, $\Lambda_t \Theta C_t$ represents the impact of contextual information on the current moment.

By integrating global and local features into an English machine translation error detection model, the following optimization effects can be achieved:

(1) By integrating global and local features, this model can better understand the semantic information of the source and target languages, thereby effectively detecting errors in the translation process.

(2) Due to the powerful learning ability of deep learning frameworks, this model can better capture the relationship between the source language and the target language, and make more accurate choices of phrases and vocabulary in translation.

(3) In translation detection, the model can automatically correct errors such as verb tenses and subject verb consistency through learned grammar rules, thereby reducing grammar and structural issues.

(4) It can adaptively learn translation patterns that are suitable for different corpora and text types. This allows the model to perform well in different translation tasks and provide higher quality translation output.

D. TESTING SCORE MEASUREMENT STANDARDS

The results of error detection in English machine translation require some evaluation indicators to objectively and accurately evaluate the performance of the model. The experiment selected common evaluation indicators such as Precision (P), Recall (R), and $F_{0.5}$ value to evaluate the performance of the machine translation error model. Although F_1 score is widely used as an evaluation index, in the task of machine translation error detection, considering the importance of machine translation error detection and sensitivity to wrong translation, more weight is tended to be placed on missed reports, that is, in the hope of minimizing the omission of wrong translation. Therefore, $F_{0.5}$ is chosen, where a beta value of 0.5 means that we give a higher weight to recall (the proportion of incorrect translations correctly detected). By increasing the weight of recall rate, we can reduce the situation of missing report to a certain extent and improve the accuracy of error detection. The specific calculation formula is as follows:

Precision is the proportion of true positive examples output by the evaluation model. In English machine translation error detection, positive examples represent the translation results that the model determines to be translation errors. Therefore, P measures how much of the translation results identified by the model as translation errors are correct. A higher P-value indicates a higher accuracy of the model's judgment. In the formula, *TP* represents the samples correctly predicted as positive in the positive class samples, and *FP* represents the samples incorrectly predicted as positive in the negative class samples [29].

$$P = \frac{TP}{TP + FP} \tag{20}$$

Recall measures the model's ability to accurately detect translation errors in translation results. A higher R value indicates a stronger detection ability of the model. In the formula, FN represents the samples in the positive class that were incorrectly predicted as negative classes [30].

$$R = \frac{TP}{TP + FN} \tag{21}$$

The $F_{0.5}$ value is an evaluation indicator that comprehensively considers Precision and Recall, and adjusts the importance of both by weighting them. In some cases, there is a greater emphasis on the recall of the model, and the larger the F-value, the better the overall performance of the model [31].

$$F_{0.5} = \frac{(0.5^2 + 1) \times P \times R}{0.5^2 \times P + R} \tag{22}$$

To more accurately evaluate errors in English machine translation, the GLEU metric is used to evaluate sentence fluency. The GLEU metric evaluates the quality of machine translation by calculating the weighted accuracy P_n between sentence C corrected by the system and the manually annotated standard sentence R. Among them, S represents incorrect sentences, R represents correct standard sentences, and W_n represents a uniform distribution of weight values [32]. So, the calculation of the GLEU indicator is:

$$GLEU(C, R, S) = BP \cdot exp(\sum_{n=1}^{4} W_n log\, p'_n) \tag{23}$$

The BP calculation formula is:

$$\mathrm{BP} = \begin{cases} 1 & if\ c > r \\ e^{(1-\frac{r}{c})} & if\ c \leq r. \end{cases} \tag{24}$$

In the formula, c is the length of the system's output sentence, and r is the length of the target sentence.

IV. SIMULATION TESTING AND ANALYSIS

A. TESTING ENVIRONMENT AND DATA

1) EXPERIMENTAL ENVIRONMENT AND SETTINGS

The experimental data for this study is large and requires the use of computers with higher configurations to complete. Using a server equipped with four high computing hardware - RTX2080Ti, with Ubuntu 18.04 CPU, it can effectively support high-performance computing and data processing. 32GB DDR4 3000MHz memory to support large-scale data loading and processing. High frequency DDR4 memory can also provide faster data transfer speed, accelerating computer processing speed. And with 11GB x 4 of graphics memory, its powerful computing power can accelerate tasks such as deep learning and scientific computing. Meanwhile, the entire experimental program was written in Python language with TensorFlow1.14 as the deep learning framework. TensorFlow is a widely used deep learning framework with highly flexible graph computation and automatic differentiation capabilities. It supports various types of neural networks and deep learning models, and provides rich optimization and debugging tools. Through TensorFlow, deep learning models can be easily constructed, trained, and deployed for accurate translation error detection.

In order to obtain accurate English machine translation error detection results, multiple parameters in the model were repeatedly tested and appropriate values were set to improve the performance and effectiveness of the error detection model. The parameter settings for the Bi-GRU model and CNN model used in the proposed model are shown in Table 1. The Bi-GRU model is optimized during training using the Adam optimizer.

TABLE 1. Parameter settings of the research model.

Parameter Name	Parameter values
Optimizer	Adam
Decay learning rate	1
Hidden_Num	256
Attention_Length	64
Num_Layer	2
Window_Length	10
Learning rate	0.001
Multi-scale	4
Number of epoches	30
Word vector dimension	256

2) SAMPLE SET AND PROCESSING

To evaluate the performance of the proposed deep semantic learning framework that combines global and local features for detecting errors in English machine translation, it is necessary to select an appropriate dataset. In this field of research, selecting and processing experimental datasets is crucial. The data set should contain correct translations and wrong English target language translations, which can be obtained from different sources, such as specially collected machine translation research data sets, open translation tasks (such as WMT translation contest data sets) or translation samples obtained on the Internet. When selecting a dataset, it is also important to ensure that it can cover various types and levels of translation errors, and has sufficient sample size and diversity. Therefore, this article used the following two datasets:

(1) CoNLL-2014 Dataset: The CoNLL-2014 Official Grammar Correction Task Corpus is a dataset widely used in natural language processing and machine translation, containing 60000 parallel sentence pairs. This dataset covers various types of grammatical errors, including noun errors, subject verb inconsistency errors, and verb errors. This dataset can be used to train machine translation models and other multiple tasks.

(2) WMT Dataset: is an international machine translation conference whose official website provides a series of parallel corpora for English and other languages, which can be used for machine translation evaluation. This dataset covers sentences from multiple languages and different topics.

To further verify the performance of the translation error detection method proposed in this article and ensure the effectiveness and reliability of the model. The experiment divides each dataset into training, validation, and testing sets, which can evaluate the performance of the model in real-world scenarios. The training set is used to train the parameters of the model, the validation set is used to adjust the hyperparameters of the model and select the optimal model, and the test set is an independent sample for the final evaluation of model performance. Testing an independent test set on data that the model has never seen before can objectively evaluate the generalization performance of the model. Therefore, in order to train and test the constructed model, both datasets were divided into training, testing, and validation sets in a 6:3:1 ratio for experimentation and evaluation of our algorithm results.

B. MODEL DETECTION PERFORMANCE RESULTS

In order to verify the effectiveness and superiority of the English machine translation error detection model proposed in this article that integrates global and local features, experiments were conducted based on the CoNLL-2014 dataset and WMT dataset mentioned above, and the two datasets were processed and partitioned accordingly to ensure the quality of the data, and to display the experimental results with appropriate visual charts, it will be more intuitive to understand the relationship and trend between the data. At the same time, the equipment, parameters, and word vector training methods used in each model were uniformly processed in the experiment to ensure the accuracy of the results. This model was trained and compared with CNN, Bi LSTM, Bi-GRU, BERT, and Transformer for error detection. The comparison results are shown in Figure 5-7. Figures 5 and 6 show the comparison results of English sentence syntax correction between the CoNLL-2014 dataset and the WMT dataset, respectively. In order to further compare the differences between the results of this model and professional English translation, the GLEU indicator shown in Figure 7 is used to evaluate the overall coherence and fluency of sentences.

According to the comparison results of English sentence syntax errors under the CoNLL-2014 data set in Figure 5 and Table 2, it can be seen that the deep semantic learning framework model based on the CoNLL-2014 data set has the best performance in the English machine translation error detection task, and has the highest accuracy and superiority. Its prediction accuracy is 87.01%, recall is 48.74%, and the $F_{0.5}$ score is as high as 76.22%. This means that the model, after balancing accuracy and recall, assigns a high weight to the importance of detecting errors. The performance of other models is relatively low. The accuracy of the BERT model is 79.97%, which is relatively high, but the recall rate is only 43.21%, and the $F_{0.5}$ score is 55.34%. The Bi-GRU and Bi LSTM models performed well in terms of recall, reaching levels of 45.66% and 42.32%, respectively, but with lower accuracy. The performance of the Transformer model is at a moderate level in terms of accuracy and recall. The

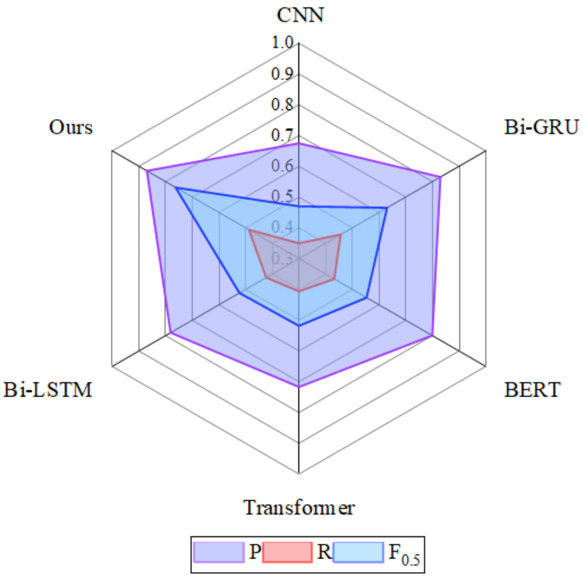

FIGURE 5. Comparison of english sentence grammar errors on the CoNLL-2014 Dataset.

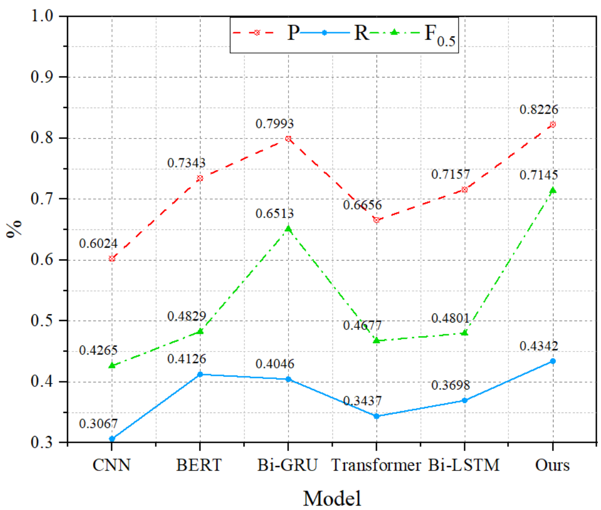

FIGURE 6. GLEU results for each model.

advantage of this model compared to other models may lie in its ability to extract features from different perspectives. Bi-GRU can learn global semantic information, while CNN can better capture local features. This fusion can provide a more comprehensive understanding of the semantics and structure in sentences, and effectively detect errors.

Figure 6 shows the comparison results of English sentence syntax errors on the WMT dataset. From the overall performance of the results, the Ours model performs the best in accuracy, recall, and $F_{0.5}$, with values of 82.26%, 43.42%, and 71.45%, respectively. This indicates that the model can achieve high accuracy and superior performance in English machine translation error detection tasks. The Bi-GRU and Bi LSTM models have relatively high Precision and Recall values of 79.93% and 71.57%, respectively, but

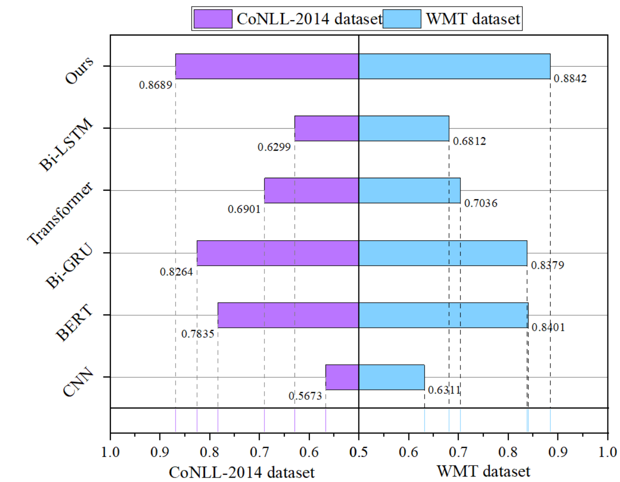

FIGURE 7. Precision and recall results of the model in terms of syntax error types in this article.

TABLE 2. Comparison results of syntax errors in English sentences in the CoNLL-2014 dataset.

Model	Pre	Re	$F_{0.5}$
CNN	67.44%	37.98%	46.98%
BERT	79.97%	43.21%	55.34%
Bi-GRU	83.01%	45.66%	63.02%
Transformer	71.74%	40.65%	51.86%
Bi-LSTM	78.11%	42.32%	52.32%
Ours	87.01%	48.74%	76.22%

are relatively low in the $F_{0.5}$ metric. This may indicate that these two models have some balance issues in comprehensive evaluation, namely the trade-off between accuracy and recall. The BERT model performs relatively well in Precision and Recall, at 73.43% and 41.26%, respectively, but slightly lower than the Bi-GRU and Bi LSTM models at 48.29% in the $F_{0.5}$ metric, indicating that the model has some shortcomings in balancing accuracy and recall. The Transformer model achieved 66.56% and 34.37% in Precision and Recall, respectively, and 46.77% in $F_{0.5}$ metric.

Compared to other models, its performance is relatively low. Compared to the results of the CoNLL-2014 dataset, the English machine translation error detection results of this dataset are significantly lower, which may be due to the fact that the CoNLL-2014 dataset is mainly used for syntactic and semantic dependency analysis tasks. The sentences in the dataset are usually short and annotated with grammatical and semantic information such as part of speech and dependency relationships. The WMT dataset is a dataset used for machine translation tasks, which contains bilingual sentence pairs from different languages. WMT datasets typically contain a large number of sentences and texts, each of which has a corresponding translation. From this, it can be seen that the selection of the dataset will have a certain impact on the experiment, but the accuracy and superiority of the model in

English machine translation error detection are the highest, and its error detection performance is the best.

From Figure 7 and Table 3, it can be observed that different models have different scores on the GLEU indicator on the CoNLL-2014 dataset and the WMT dataset. The overall GLEU results of each model on the WMT dataset are better than those on the CoNLL-2014 dataset. On the CoNLL-2014 dataset, the highest GLEU score is 0.8689, corresponding to the model in this paper. On the WMT dataset, the highest GLEU score is 0.8842, which also corresponds to the model we proposed. Further verification confirmed that the difference between the grammar correction results of our model on two datasets and the translation results of professional English personnel is relatively small, which means that our model performs better in grammar correction tasks compared to other models. In addition, it can be observed that the BERT model has relatively high GLEU scores on both datasets, with values of 0.7835 (CoNLL-2014) and 0.8401 (WMT), respectively. This indicates that the BERT model also performs well in grammar correction tasks. However, CNN and Bi LSTM models have lower GLEU scores on both datasets and are located in lower positions, respectively. From this, it can be seen that the model presented in this article exhibits high grammar correction ability, and has great potential to achieve better results in natural language processing tasks, with small differences compared to professional English translation results.

TABLE 3. **GLEU results for each model.**

Model	CoNLL-2014 Dataset	WMT Dataset
CNN	0.5673	0.6311
BERT	0.7835	0.8401
Bi-GRU	0.8264	0.8379
Transformer	0.6901	0.7036
Bi-LSTM	0.6299	0.6812
Ours	0.8689	0.8842

1) STATISTICAL ANALYSIS OF ERROR RESULTS

In order to further evaluate the effectiveness and superiority of this model in grammar correction, 1200 English compositions from students from other majors who took the English proficiency test were selected from the corpus as the test data for this study. Conduct experiments on this test set using the model proposed in this article, and calculate the grammar error correction accuracy, recall, and $F_{0.5}$ value of this model based on the experimental results. The specific results are shown in Figure 8.

According to the results in Figure 8, it can be observed that the increase in the number of sentences in the test data has resulted in a stable improvement in the grammar correction performance of the model in this paper. When the number of articles is 200, the accuracy of the model reaches 79.45%, the recall rate is 60.25%, and $F_{0.5}$ is 68.63%. When the number of articles increased to 1200, the accuracy rate increased to 81.8%, the recall rate increased to 65.76%, and $F_{0.5}$ further improved to 72.91%. Based on a comprehensive analysis of these data, it can be concluded that as the number of articles increases, the effectiveness and superiority of the model in grammar correction have improved. Although there is some fluctuation in accuracy, recall, and $F_{0.5}$ score on the entire test dataset, in most cases, the model can effectively correct grammar errors, and for different numbers of English grammar errors, the accuracy is maintained at around 80%, the average recall is 62%, and $F_{0.5}$ is around 70%. Overall, its error correction effect is generally good. Therefore, this model has certain effectiveness and superiority in grammar correction.

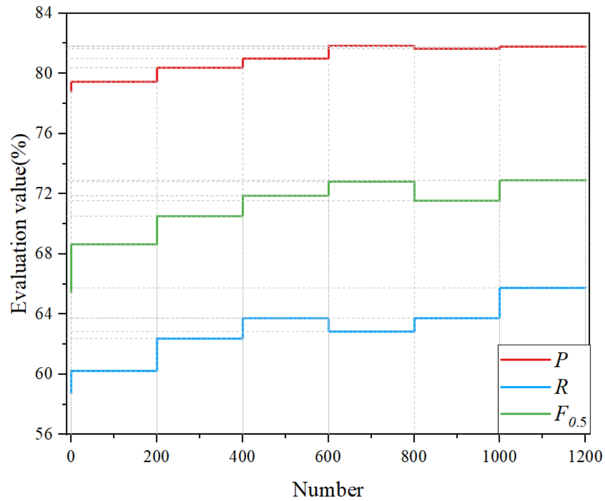

FIGURE 8. **The grammar error correction evaluation values of this article's model under different article numbers.**

To verify the effectiveness of the model in correcting grammar errors in this article. We have compiled and categorized the types of grammar errors annotated: tense errors (Ten), subject verb consistency errors (SVC), article, preposition, and pronoun errors (APP), part of speech errors (PS), verb errors (Ver), coordinate structure errors (CS), sentence structure errors (SS), incomplete sentence errors (IS), and word order errors (WO). Calculate the accuracy and recall of each error type based on the statistical results, as shown in Figure 9.

As shown in Figure 9, considering both precision and recall, it can be seen that the overall evaluation results of error types are relatively balanced and high. The model achieved good results in detecting errors in articles, prepositions, and pronouns (APP), with accuracy and recall rates exceeding 90%. This means that the model can accurately and comprehensively identify errors of this type. Parallel structure errors (CS) and word order errors (WO) are the worst performing types of errors in the model, with low accuracy and recall. The results of verb errors (Ver) and sentence structure errors (SS) are both around 80%, belonging to the moderate level. The results of tense errors (Ten) and incomplete sentence errors (IS) reached high levels of 89.45% and 91.24%, respectively.

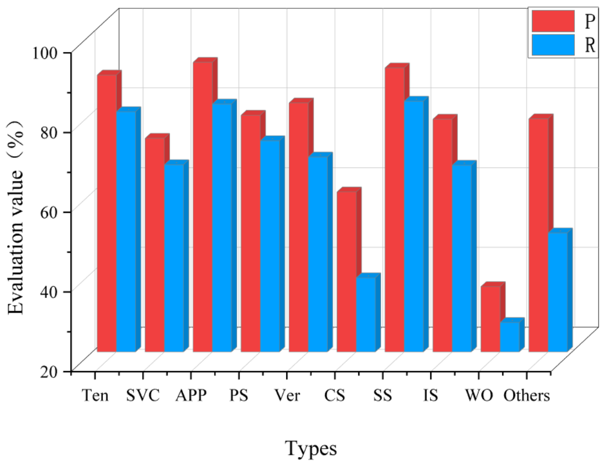

FIGURE 9. Precision and recall results of the model in terms of syntax error types in this article.

V. CONCLUSION

Due to differences between languages and limitations of artificial intelligence technology, errors in machine translation remain a common problem, hindering its further promotion. This study aims to develop an effective deep learning framework to improve the error detection capability in machine translation. Establishing a basic semantic space through twin RNNs for information sharing and mapping across linguistic spaces. And integrate attention mechanism into Bi-GRU encoder for global feature modeling and semantic information extraction. And integrate the obtained source sentence features with the local features extracted by multi-scale CNN using a gating unit structure, integrating the advantages of these two structures. This framework can consider both global and local information simultaneously, thereby more accurately detecting and correcting errors in machine translation.

Studies have shown that Bi-GRU encoders are excellent at capturing source language context information and translating it into global feature representations, which is essential for understanding the deep semantics of source language sentences. At the same time, we use multi-scale CNN to capture local semantic information at different levels in sentences. By combining these two features - global and local - effectively, we can show the meaning of the source language more fully. In English machine translation error detection tasks, our model shows higher accuracy and wider applicability than traditional methods. Through the verification of BLEU score, accuracy rate and recall rate, our model has made remarkable progress in the task of automatic translation error correction. Especially in dealing with long sentences, complex grammatical structures and ambiguities, our model shows stronger performance and robustness. This research not only provides an effective solution for the field of machine translation, but also provides important implications for the further application of deep semantic learning in natural language processing tasks. However, there are some limitations and challenges in this model. The introduction of multi-scale CNN increases the model parameters, which may lead to overfitting problems in the case of limited data. In addition, the model currently focuses on error detection in English machine translation, and its applicability to other language pairs has yet to be verified. Future research will explore ways to further enhance the generalization ability of the model by introducing cross-language data and advanced pre-training techniques.

REFERENCES

[1] S. Wexler, J. Shaffer, and A. Cotgreave, *The Big Book of Dashboards: Visualizing Your Data Using Real-world Bus. Scenarios*. Hoboken, NJ, USA: Wiley, 2017.

[2] J. Huang, X. Wu, J. Wen, C. Huang, M. Luo, L. Liu, and Y. Zheng, "Evaluating familiarity ratings of domain concepts with interpretable machine learning: A comparative study," *Appl. Sci.*, vol. 13, no. 23, p. 12818, Nov. 2023.

[3] O. I. Hammadi, A. Dawah Abas, and K. Hammad Ayed, "Face recognition using deep learning methods a review," *Int. J. Eng. Technol.*, vol. 7, no. 4, pp. 6181–6188, May 2019.

[4] W. Zhang, "An automatic error detection method for machine translation results via deep learning," *IEEE Access*, vol. 11, pp. 53237–53248, 2023.

[5] Z. Guo, K. Yu, A. Jolfaei, G. Li, F. Ding, and A. Beheshti, "Mixed graph neural network-based fake news detection for sustainable vehicular social networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 1–13, Aug. 2022.

[6] S. Farzana and N. Parde, "Are interaction patterns helpful for task-agnostic dementia detection? An empirical exploration," in *Proc. 23rd Annu. Meeting Special Interest Group Discourse Dialogue*, 2022, pp. 172–182.

[7] J. Yang, Z. Guo, J. Luo, Y. Shen, and K. Yu, "Cloud-Edge-End collaborative caching based on graph learning for cyber-physical virtual reality," *IEEE Syst. J.*, vol. 117, no. 4, pp. 1–12, Jul. 2023.

[8] J. Huang, F. Yang, C. Chakraborty, Z. Guo, H. Zhang, L. Zhen, and K. Yu, "Opportunistic capacity based resource allocation for 6G wireless systems with network slicing," *Future Gener. Comput. Syst.*, vol. 140, pp. 390–401, Mar. 2023.

[9] J. Zhu, X. Shi, and S. Zhang, "Machine learning-based grammar error detection method in English composition," *Sci. Program.*, vol. 2021, pp. 1–10, Dec. 2021.

[10] L. Pan, "A study of English learning vocabulary detection based on image semantic segmentation fusion network," *Frontiers Comput. Neurosci.*, vol. 16, Jun. 2022, Art. no. 895680.

[11] D. Mouratidis, K. L. Kermanidis, and V. Sosoni, "Innovatively fused deep learning with limited noisy data for evaluating translations from poor into rich morphology," *Appl. Sci.*, vol. 11, no. 2, p. 639, Jan. 2021.

[12] R. Zhang and H. Huang, "Automatic recognition method of machine English translation errors based on multisignal feature fusion," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–9, May 2022.

[13] R. Das and T. D. Singh, "A hybrid fusion-based machine learning framework to improve sentiment prediction of assamese in low resource setting," *Multimedia Tools Appl.*, vol. 83, no. 8, pp. 22153–22172, Aug. 2023.

[14] C. I. Eke, A. A. Norman, and L. Shuib, "Multi-feature fusion framework for sarcasm identification on Twitter data: A machine learning based approach," *PLoS One*, vol. 16, no. 6, Jun. 2021, Art. no. e0252918.

[15] B. Palani, S. Elango, and V. Viswanathan K, "CB-fake: A multimodal deep learning framework for automatic fake news detection using capsule neural network and BERT," *Multimedia Tools Appl.*, vol. 81, no. 4, pp. 5587–5620, Feb. 2022.

[16] Y. Zhou, Y. Yang, H. Liu, X. Liu, and N. Savage, "Deep learning based fusion approach for hate speech detection," *IEEE Access*, vol. 8, pp. 128923–128929, 2020.

[17] A. Gandhi, K. Adhvaryu, S. Poria, E. Cambria, and A. Hussain, "Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions," *Inf. Fusion*, vol. 91, pp. 424–444, Mar. 2023.

[18] A. Mohammed and R. Kora, "An effective ensemble deep learning framework for text classification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8825–8837, Nov. 2022.

[19] D. Gibert, C. Mateu, and J. Planes, "HYDRA: A multimodal deep learning framework for malware classification," *Comput. Secur.*, vol. 95, Aug. 2020, Art. no. 101873.

[20] X. Li, Z. Zeng, M. Wu, Z. Huang, Y. Sha, and L. Shi, "An offensive language identification based on deep semantic feature fusion," in *Proc. IEEE 8th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2022, pp. 1477–1483.

[21] H. Peng, Y. Ma, S. Poria, Y. Li, and E. Cambria, "Phonetic-enriched text representation for Chinese sentiment analysis with reinforcement learning," *Inf. Fusion*, vol. 70, pp. 88–99, Jun. 2021.

[22] F. Meng, Y. Zheng, S. Bao, J. Wang, and S. Yang, "Formulaic language identification model based on GCN fusing associated information," *PeerJ Comput. Sci.*, vol. 8, p. e984, Jun. 2022.

[23] S. Paul, S. Saha, and M. Hasanuzzaman, "Identification of cyberbullying: A deep learning based multimodal approach," *Multimedia Tools Appl.*, vol. 81, no. 19, pp. 26989–27008, Aug. 2022.

[24] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-oriented multi-user semantic communications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2584–2597, Sep. 2022.

[25] Z. Zhang, W. Lin, M. Liu, and M. Mahmoud, "Multimodal deep learning framework for mental disorder recognition," in *Proc. 15th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Nov. 2020, pp. 344–350.

[26] D. Wang, J. Su, and H. Yu, "Feature extraction and analysis of natural language processing for deep learning English language," *IEEE Access*, vol. 8, pp. 46335–46345, 2020.

[27] Y. Zheng, Z. Xu, and X. Wang, "The fusion of deep learning and fuzzy systems: A state-of-the-art survey," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 8, pp. 2783–2799, Aug. 2022.

[28] Z. Guo, Y. Shen, A. K. Bashir, M. Imran, N. Kumar, D. Zhang, and K. Yu, "Robust spammer detection using collaborative neural network in Internet-of-Things applications," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9549–9558, Jun. 2021.

[29] Y. Luo, G. Jiang, T. Yu, Y. Liu, L. Vo, H. Ding, Y. Su, W. W. Qian, H. Zhao, and J. Peng, "ECNet is an evolutionary context-integrated deep learning framework for protein engineering," *Nature Commun.*, vol. 12, no. 1, p. 5743, Sep. 2021.

[30] L. Wu, Y. Long, C. Gao, Z. Wang, and Y. Zhang, "MFIR: Multimodal fusion and inconsistency reasoning for explainable fake news detection," *Inf. Fusion*, vol. 100, Dec. 2023, Art. no. 101944.

[31] Y. Li, Z. Li, Z. Guo, A. Siddique, Y. Liu, and K. Yu, "Infrared small target detection based on adaptive region growing algorithm with iterative threshold analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5003715.

[32] Z. Wen, R. Wang, X. Luo, Q. Wang, B. Liang, J. Du, X. Yu, L. Gui, and R. Xu, "Multi-perspective contrastive learning framework guided by sememe knowledge and label information for sarcasm detection," *Int. J. Mach. Learn. Cybern.*, vol. 14, no. 12, pp. 4119–4134, Dec. 2023.

YAOXI WU was born in Chongqing, China, in 1982. He received the master's degree from Chongqing Normal University, China. Currently, he works as a Lecture with Chongqing College of International Business and Economics. His research interests include applied linguistics, semantic modeling, and machine translation.

QIAO LIANG was born in Chongqing, China, in 1989. He received the master's degree from Chongqing Normal University. Currently, he works as a Lecture with the School of Mathematics and Computer Science, Chongqing College of International Business and Economics. His research interests include computer vision, natural language processing, and deep learning.