EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

# An Automatic Grammar Error Correction Model Based on Encoder-Decoder Structure for English Texts

Jiahao Wang[1], Guimin Huang[1,2*], and Yabing Wang[1]

[1] School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, Guangxi, China

[2] Guangxi Key Laboratory of Image and Graphic Intelligent Processing

## Abstract

The role of information transmission in social life is irreplaceable, and language is a very important information carrier. Among all kinds of languages, English always occupies an important position. In the process of English learning, grammar error has become a difficult problem for most learners. In this paper, we propose an automatic grammar error correction model based on encoder-decoder structure. Different from traditional encoders, we design a dual-encoder structure to capture the information of source sentence and context sentence separately. The decoder is designed with a gated structure, it can effectively integrate output information of encoders. At the same time, the self-attention mechanism is combined to better solve the problem of long-distance information extraction. In addition, we propose a dynamic beam search algorithm to improve the accuracy of the word prediction process, and achieve dynamic extraction of the decoder output by combining kernel sampling techniques. We add a penalty factor to reduce the probability of generating repeated words, while suppressing the model's preference for generating shorter sentences. Finally, the proposed method is validated on the official English grammar error correction dataset. Experiments show that the dual encoder model in this paper has a good performance.

**Keywords:** Encoder-decoder, Grammar Error Correction (GEC), deep neural network, attention mechanism, beam search

Received on 17 July 2022, accepted on 10 September 2022, published on 12 September 2022



doi: 10.4108/eetsis.v9i5.2011

*Corresponding author. Email: sendhuang@126.com

## 1. Introduction

English is one of the most commonly used languages in social life. At present, the number of ESL(English as a Second Language) learners has far exceeded 1.5 billion in the world[1]. In the process of English learning, English writing practice is one of the most common methods, and writing ability is also an important indicator of English proficiency. However, affected by native language transfer, English texts created by ESL learners tend to have more grammar errors. Compared with such a large group of ESL learners, the number of English teachers is seriously insufficient. Therefore, it is a difficult task for teachers to manually correct all students' errors in their writing. In order to relieve the pressure of English teachers and solve the problem of the shortage of teaching resources, it is a good solution to use computer for auxiliary correction[2].

With the development of natural language processing (NLP) technology, the method of grammar error correction has made continuous progress. The first to emerge is the rule-based error correction method, which uses grammar rules established by linguistic experts for error matching. The results obtained by this method are relatively accurate, but the expansion of the rule base requires a lot of effort by professionals, and it is difficult to cover all types of grammar errors. There are also statistical-based grammar error correction methods that model language through relevant features of text. This

method usually selects a fixed size window to calculate the probability of sentence fragments, and the calculation amount increases exponentially with the window size, so it cannot cover a wider range of context information. In recent years, with the rapid development of deep learning technology, grammar error correction methods based on deep neural networks have gradually become the mainstream. However, most methods pay more attention to the internal information of the source sentence, ignoring the context in which the sentence is located. In fact, each sentence of an article is semantically interrelated rather than completely independent, so contextual sentences are also crucial for grammatical error correction. In order to solve this problem, we try to design a deep structure to extract sentence features in a wider range. Through the dual-encoder structure, the context information of adjacent sentences is considered in the error correction process, thereby improving the accuracy of the source sentence, which is the main topic of this paper. The main contributions of this paper are as follows:

The main contributions of this paper are as follows:

- We make improvements to the traditional encoder structure. An additional encoder is added to extract the contextual sentence information based on the Transformer, which we call the contextual information encoder. This encoder is able to extract sentence information in a wider range, thus providing more useful references for the process of decoding the source sentence. The other is the Bi-GRU encoder. The bidirectional structure can extract the source sentence information from the forward and backward directions, avoiding information omission caused by unidirectional encoding.
- Instead of using a traditional neural network as the decoder, we design a gated structure to process the textual information of the two decoders. Different weights are assigned to each part through the attention mechanism and processed in a gated structure. Therefore, the decoding process of the source sentence can maximize the reference to its context.
- We improved the traditional beam search algorithm based on nucleus sampling method and designed a dynamic beam search algorithm. This decoding method can calculate the probability values of different outputs, so as to dynamically select the number of optimal inferences, instead of always selecting a fixed number of outputs. In this way, our automatic grammar error correction model has higher inference efficiency and more accurate outputs.

The remainder of this paper as follows. Section 2 presents the related research on grammar error correction; Section 3 describes the grammar error automatic correction model we propose in detail; Section 4 analyses the comparative experiments on the official data set of grammar error correction; Section 5 summarizes the works of this paper and discusses future research ideas.

## 2. Related Work

Rule-based grammar error correction methods have appeared in the 1980s[3]. Linguists write some linguistic features as grammar rules, which are matched by a parser during error correction. For example, the open source tool LanguageTool [4], and the ESL Assistant [5] developed by Microsoft in the early days are all rule-based error correction methods. However, natural language has huge complexity, flexibility and uncertainty. To achieve higher precision, it is necessary to increase the number of rules, which will increase the possibility of rule conflicts. Classifier-based methods treat a specific error type as a classification problem and train a classifier based on contextual relational features. Makarenkov et al. [6] used bidirectional long short-term memory(LSTM) for training and word selection according to the sentence context, which has obvious advantages in the learning of contextual features.

Methods based on deep neural networks perform equally well in grammar error correction tasks. Hu et al.[7] designed an error correction model based on a convolutional neural network, and used the clustering of word vector features to improve the performance of the model. Xie et al. [8] designed an Encoder-Decoder architecture based on RNN, using a character-level model with an attention mechanism to process OOV words. They integrated an n-gram language model and beam search algorithm on the encoder to calculate the score of candidate prefixes. Hu et al. [9] used a logistic regression model to study the relevant features in grammar error correction, and compressed the features through a clustering algorithm. Experiments on ten prepositions and eleven grammar errors demonstrate the effectiveness of the method. Yan [10] constructed a sequence annotation model using Bi-LSTM, which provided a new idea for grammar error correction. Chollampatt et al. [11] first used convolutional neural networks in the encoder-decoder structure to correct grammar, word spelling and other errors, and used the method of minimizing the error rate in the scorer to train an n-gram language model to optimize the target matrix. Based on the back-translation method in machine translation, Xie et al. [12] used the parallel sentence pair corpus to train the grammar error generation model, and finally formed a pseudo-parallel sentence pair with the correct sentence to expand the scale of the training corpus. Zhao et al. [13] proposed a copy-enhanced model structure. Based on a large-scale unlabeled training corpus, a denoising auto-encoder was used to pre-train a transformer model with a copy mechanism [14], and achieved good results. Cheng et al. [15] designed a character-level deep learning model based on transformer and Seq2Seq, and used a model ensemble approach combined with an N-gram language model to obtain the highest-scoring output. Zhou et al. [16] used

the idea of classification model to design a grammar error correction model, and continuously optimized the model through the grammatical relationship and hierarchical structure between words. Tarnavskyi et al.[17] made an in-depth study on the sequence tagging method of pre-trained large-scale models, ensemble models by span-level edits voting algorithm, and achieved new SOTA results on the BEA-2019 test set.

NLP-related technologies have a wide range of applications in social life. In addition to grammatical error correction, it has good application prospects in database algorithms[18][19][20], auxiliary disease diagnosis [21][22], and insurance recommendation algorithms[23]. The above methods have good performance in grammar error correction, but do not fully utilize the sentence features of cross-sentence context. This paper designs a grammar error automatic correction model with dual encoder structure, which not only captures the internal information of sentences, but also pays attention to the interactive information between sentences, effectively expanding the range of text processing.

## 3. Methodology

In this paper, we design an encoder-decoder model with a dual-encoder structure named deep contextual information model (DCIM). This model utilizes the idea of auxiliary encoder, and combines the encoder structure of the Transformer model and the Bi-GRU neural network model to extract the deep contextual information of the source sentence. We use context information encoder to learn the semantic relation-ship between the source sentence and its context. Except this, Bi-GRU encoder is used to learn the features of the source sentence. The overall structure of DCIM is shown in Fig.1.

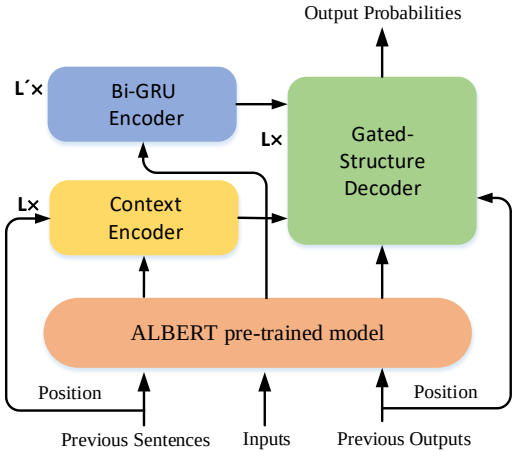


**Figure 1.** The structure of DCIM

### 3.1. Dual-Encoder Structure

The encoder of the model consists of two parts, the context information encoder and the Bi-GRU encoder. The context information encoder can focus on a wider range of contextual information and extract useful information through the Attention mechanism. The Bi-GRU encoder encodes the source sentence from both forward and reverse directions to obtain more comprehensive semantic information. The following two parts are described in detail.

#### Context Information Encoder

The context information encoder is designed based on the structure of the Transformer encoder for two reasons:

- The problem of long-distance dependency has always been a difficulty in the processing of lengthy sentences. This problem first appeared in the N-gram model in statistical methods. A window of fixed size (usually 3-5 tokens) cannot cover the semantic information of the full sentence. With the emergence of neural network models such as LSTM and GRU, the problem of long-distance dependence has been alleviated to a certain extent. But it's still powerless when dealing with sentences with too many words. Trans-former reduces the distance calculation between any two positions of the sentence to a constant through the attention mechanism, which better solves the problem of long-distance dependence.
- Parallel computing can improve the efficiency of model training and determine the performance of the model. The traditional RNN model performs sequential calculations, so the calculation at each time step depends on the output results of the previous step. A fully-connected neural network is used between the hidden layers, which does not have the ability of parallel computing. Some researchers try to improve the structure of the hidden layer of RNN to realize parallel computing. For example, the full connection be-tween hidden layers is interrupted according to the fixed time step part, and then the depth of the network is increased to obtain more distant features. The computing power of the improved RNN model still cannot exceed the CNN model. The CNN model can be computed in parallel, but for long-distance information transfer CNN requires multiple layers of convolution to expand the receptive field. Compared with the CNN model, the Transformer is completely parallelized, which greatly improves the computational efficiency.

Combining the above two points, we design a context information encoder based on the encoder structure of Transformer. The grammatical features of sentences usually depend on key contextual information. The context encoder can effectively extract the key information in the sentence through the multi-head self-

attention mechanism. The structure of the context information encoder is shown in Fig.2.

The vector representation of the sentence is obtained through ALBERT, and then the Multi-Head Atten-tion mechanism is used to calculate the grammatical and semantic connections that exist between each word and each sentence. This can improve computational efficiency through parallel computing. In the Add & Norm module, add means residual connection and norm means layer normalization. The layer normalization operation is performed after adding the output of the encoded position information to the out-put of the multi-head self-attention layer.

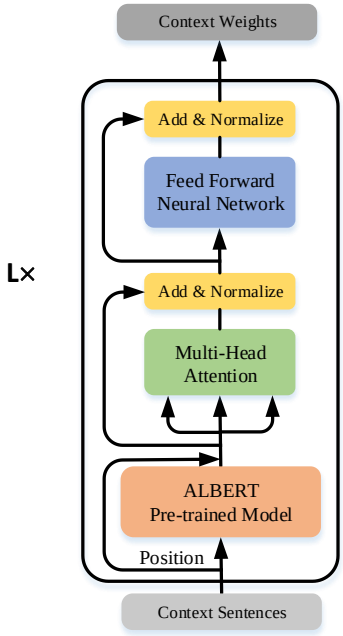


**Figure 2.** The structure of context information encoder

Such an approach can only focus on the differences and make the model easier to train. The output of the Add & Norm layer is passed to the Feed Forward neural network layer. Then it is output after the Add & Norm layer calculation again. The output is the value of the context attention in the Decoder. When the number of historical sentences is less than 3 sentences, use "<SOS> <SEP>" to complete.

Assume that the input of the contextual information encoder contains N sentences: $S=S_1, S_2, \ldots, S_N$, , for the kth sentence, the number of tokens is $|S_k|$ , that is, $S_k = s_{k,1} \ldots s_{k,|S_k|}$. Suppose the probability of correcting the target is $T_k = t_{k,1} \ldots t_{k,|T_k|}$, when the grammatical error of a single sentence is corrected without considering its context, the probability of sentence correction can be expressed by the conditional probability formula (1):

$$P(T_k \mid S_k, \Theta) = \prod_{i=1}^{|T_k|} P(t_{k,i} \mid T_{k,<i}, S_k, \Theta) \quad (1)$$

In the above formula, $T_{k,<i}$ is the historical target word $t_{k,1}, t_{k,2}, \ldots, t_{k,i-1}$ , $\Theta$ is the parameter of the model as shown in (2).

$$\Theta = (W^K, W^Q, W^V, W^{FFN}, W^{LAYERNORM}; b^K, b^Q, b^V, b^{FFN}, b^{LAYERNORM}) \quad (2)$$

Considering the information interaction between sentence and context, the probability of sentence correction is updated as (3):

$$P(T_k \mid S_k, \Theta) = \prod_{i=1}^{|T_k|} P(t_{k,i} \mid T_{k,<i}, S_k, S_{doc}, \Theta) \quad (3)$$

$S_{doc}$ represents the context of sentence $S_k$, formally written as $S_{doc}=(S_{k-1}, S_{k-2}, S_{k-3})$. It should be noted that the context sentence $S_{doc}$ is the original context, not the target sentence generated by the model. The format of $S_{doc}$ when entering the context information encoder is as follows: "<SOS>The electric cars invented in 1990 did not have a powerful battery.<SEP>Due to the limitation of its weigh, size and the battery technology, the battery used in the electric cars at that time was limited to a range of 100 miles.<SEP>As a result, they are not convenient enough.<EOS>"

For ease of understanding, we use x to represent each token in sentence S. We use the ALBERT pretrained model to obtain word vector representations for each word. After vectorizing the input sentence, the first hidden layer state h0 of the model is obtained (4):

$$h_0 = xW_c + pW_p + qW_Q \quad (4)$$

$W_c$ is the word embedding matrix, which converts the input tokens into word embeddings. $W_P$ is the position embedding matrix. p is the position encoding, which is used to provide the position information of the word for the non-sequential encoder. $W_Q$ is the segment embedding matrix, and q is the segment encoding, which is used to distinguish the source sequence from the target sequence. In our model, a 2-layer context information encoder structure is adopted, and the hidden layer state representation is shown in (5):

$$h_n = \text{transformer}_{enc}(h_{n-1}) \quad \forall n \in [1, 2] \quad (5)$$

## Bi-GRU Encoder

While considering the global information of the context, this paper also considers the local information of the source sentence. Compared with the RNN model and language models such as N-gram in statistical methods, Bi-GRU expands the scope of feature extraction through the gated structure. Moreover, com-pared with the large parameter model, the structure of the GRU model is simpler, and it is more convenient to adjust the key parameters. Therefore, the Bi-GRU encoder is adopted in this paper to obtain the internal features of the source sentence. The Bi-GRU encoder encodes according to the

order of sentences in the document and it encodes sentences in both forward and backward directions in order to obtain a more comprehensive feature representation. The Bi-GRU encoder structure is shown in Fig.3.

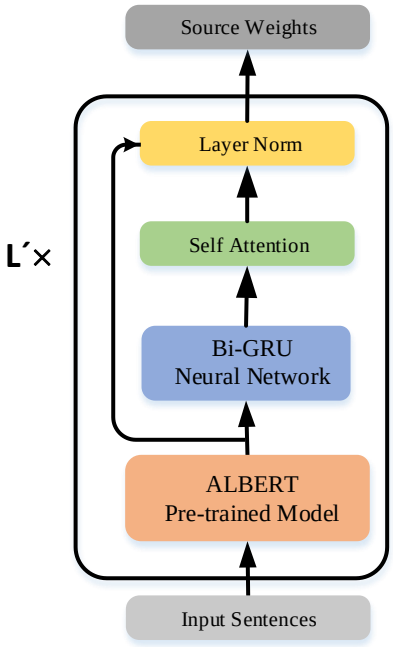


**Figure 3.** The structure of Bi-GRU encoder

In the Bi-GRU encoder, the state $h_t$ of the hidden layer at time t is the concatenation of the hidden layer states in the forward and reverse directions as shown in (6):

$$h_t = h_t^+ \oplus h_t^- \tag{6}$$

Assuming that the number of layers of Bi-GRU is n, then the final state of the forward hidden layer is $h_n^+$, and the final state of the reverse hidden layer is $h_n^-$. The two states are concatenated and the final hidden layer state of Bi-GRU is obtained as shown in (7):

$$h_{final} = h_n^+ \oplus h_1^- \tag{7}$$

Using $h$ to denote the state of the hidden layer at each moment. Using $a=(a_1, a_2,\ldots a_n)$ to denote the attention probability distribution of the final state of Bi-GRU. Then at time n, the probability distribution of the hidden layer state to attention can be written as (8):

$$a_n = \frac{\exp(h_n^{'})}{\sum_{i=0}^{N}\exp(h_i^{'})} \tag{8}$$

$$h_n^{'} = h_n^T U h_{final} \tag{9}$$

In the above formula, $N$ is the token number of the input sentence, and $U$ is the weight matrix. Finally, perform layer normalization to obtain the hidden layer state of the Bi-GRU encoder as shown in (10~13):

$$h_{t\_enc} = f[\frac{g}{\sigma_t} e\ (a_t - \mu_t) + b] \tag{10}$$

$$\mu_t = \frac{1}{H}\sum_{i=1}^{H} a_t^i \tag{11}$$

$$\sigma_t = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(a_t^i - \mu_t)^2} \tag{12}$$

$$a_t = W_{hh} h_{t-1} + W_{xh} x_t \tag{13}$$

In the formula (10), $g$ and $b$ are the gain matrix and the offset matrix, respectively. $H$ is the number of hidden layer units, $W_{xh}$ and $W_{hh}$ represent the weight matrix between the input and the hidden layer and between the hidden layer and the hidden layer, respectively.

## 3.2. Decoder

In the process of generating the target sequence word by word, the decoder needs to make full use of more contextual information of the sentence to obtain more accurate prediction results. The decoder can better solve the variable length problem of the output. In addition, more historical information can be used as decoding features, so that the output at the current moment can be more accurately predicted. In the decoding process, Masked Multi-Head Attention is first performed on the input of the decoder to obtain the attention weight after the mask. Then combined with the results of Bi-GRU encoder and context information encoder, the gating mechanism is used to ex-tract the correlation information between the source sentence and the context to obtain the final target sequence.

### Context-Referenced Decoder

We design a context-referenced decoder based on a gated structure. This method can better integrate the contextual attention weight and the source sentence attention weight. Therefore, when decoding the source sentence, the coding information of the context can be effectively used. The context-referenced decoder structure is shown in Fig.4.

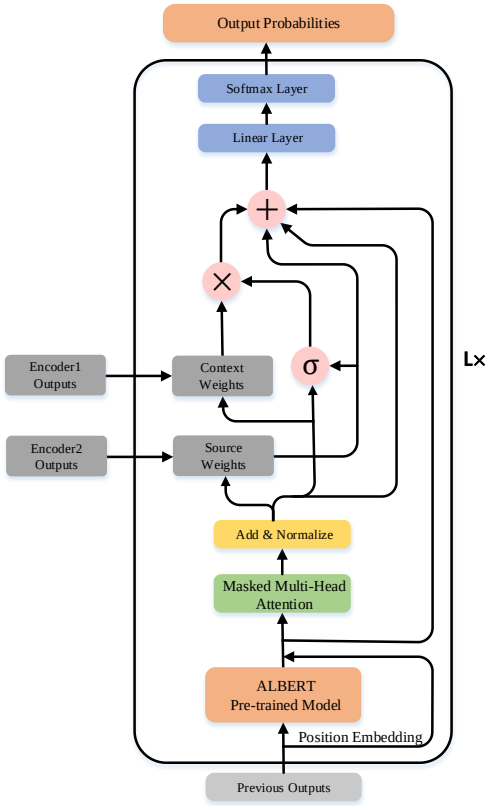


**Figure 4.** The structure of context-referenced decoder

The calculation steps in the model are described below in combination with the formula. Considering the Masked Multi-Head Attention and Add & Norm in the decoder as a whole, the output $Y_t$ of this part at time $t$ depends on the output of the decoder before t-1 time, as shown in (14):

$$Y_t = \text{LayerNorm}(\text{MultiHead}(G_{t\text{-}1}) + G_{t\text{-}1}) \tag{14}$$

Combining the source sentence attention weight $C_t$ and the contextual attention weight $\hat{C}_t$, the final output $G_t$ of the decoder is obtained after the gating unit as shown in (15~16)

$$G_t = Y_t + C_t + \Lambda_t \, e \, \hat{C}_t + G_{t-1} \tag{15}$$

$$\Lambda_t = \sigma(LIN(Y_t) + LIN(C_t)) \tag{16}$$

Symbol e represents the Hadamard product, and the part $\Lambda_t$ e $\hat{C}_t$ represents the influence of the above information on the current moment.

### Dynamic Beam Search

In order to obtain more suitable candidate words in the smallest sampling range, this paper designs a dynamic beam search algorithm. The algorithm uses Nucleus Sampling technology to dynamically select the sampling interval of word probability distribution to adapt to diverse probability distributions. This method does not fix the number of elements in the candidate set, but fixes the proportion of the sum of the probability distributions of each element in the candidate set to the overall probability. The formulation is described as follows, given a probability distribution to construct a minimum candidate set and make the formula (17) true:

$$\sum_{x \in V^{(p)}} P(x \mid x_{1:i-1}) \geq p \tag{17}$$

The $p$ is the threshold of probability. After the accumulated probability exceeds the threshold, it will be truncated and the following candidate words will no longer be used. Therefore, kernel sampling is also called top-p sampling.

In this paper, the method of Keskar et al. [24] is combined and improved on the top-p method. Finally, we design a dynamic beam search method with penalty factor. Penalty factors are introduced for two purposes. One reason is to reduce the probability that words that have appeared before will appear again, so as to avoid repeated words always appear in the inferred sentences. Another reason is that the length penalty factor is used to suppress the bias of the model to generate sentences with shorter lengths, thereby improving the accuracy of error correction results. After adding the penalty factor, the probability distribution of words is shown in (18~19):

$$p_i = \frac{\exp(x_i / T \cdot I(i \in g))}{\sum_j \exp(x_j / T \cdot I(j \in g))} \tag{18}$$

$$\begin{cases} I(c) = 1.2 \ , & \text{when c is true} \\ I(c) = 1.0 \ , & \text{when c is false} \end{cases} \tag{19}$$

In the above formula, $g$ is the set of historically generated words. Through continuous training and optimization, the hyperparameter $T$ in this formula is set to 0.7, and the parameter in the length penalty factor is set to 0.6. Assuming that the target sentence is generated from the source sentence, the score function after introducing the penalty factor is shown in (20~21):

$$\text{score}(Y, x) = \frac{\log(p(Y \mid x))}{\text{LP}(Y)} \tag{20}$$

$$\text{LP}(Y) = \frac{(5 + |Y|)^{\alpha}}{(5+1)^{\alpha}} \tag{21}$$

## 4. Experiment

We validate the performance of our proposed dual-decoder-based grammar error automatic correction model on the official English grammar error correction dataset.

## 4.1. Dataset

Our training set includes NUCLE, CLEC and ICNALE corpora, and the validation set includes CoNLL-2013 Test Set and JFLEG Dev Set. We adopt the CoNLL-2014 Test Set and JFLEG Test Set to test the overall performance of our model. Then, we extract 1000 essays from the CLEC corpus to test the performance of DCIM on different types of grammar errors. The details of the dataset are shown in Table 1.

NUCLE was jointly created by the NLP team of the National University of Singapore (NUS) and Dahlmeier et al.[25] It is the official training corpus for the grammar error correction tasks CoNLL-2013 and CoNLL-2014, and contains 57,151 parallel sentence pairs.

CLEC mainly includes college English CET-4 and CET-6 and professional English CET-4 and CET-6 students' essays, as well as some middle school students' English essays. It covers Chinese students' essays of various English proficiency levels, with a total of more than 1 million words.

ICNALE is an international learner corpus developed by Dr. Ishikawa of Kobe University in Japan, which contains about 1.3M tokens, and the corpus contents are all from ESL learners in Asia.

Table 1. Details of the dataset.

| Corpus type | Source | Number of tokens |
|---|---|---|
| Train Set | NUCLE | 3,835,212 |
| | CLEC | 2,704,309 |
| | ICNALE | 3,328,625 |
| Dev Set | CoNLL-2013 Test Set | 29,207 |
| | JFLEG Dev Set | 14,000 |
| Test Set | CoNLL-2014 Test Set | 30,144 |
| | JFLEG Test Set | 13,000 |
| | CLEC（ST3） | 78,397 |
| | CLEC（ST4） | 124,463 |

During model training, the number of hidden units of transformer encoder is 512, the number of heads of multi-head attention is 6, the number of hidden layer units of Bi-GRU is 256, the dropout is set to 0.2, and the probability threshold p in dynamic beam search is 0.95, and the length penalty parameter is 0.6.

## 4.2. Baseline Model

We test and compare our model with existing models on the above datasets, and the selected baseline models are as follows.

- Ji et al. [26] build a nested attention layer based on the seq2seq structure, using word-level and character-level attention. Error correction for grammar and sentence fluency through word-level attention, and correct spelling errors through character-level attention. GRU units are used in both encoder and decoder.
- Chollampatt et al. [11] used multi-layer CNN and attention mechanism to correct grammar and collocation errors, initialized word embeddings through pre-training methods, and introduced a multi-model integration strategy.
- Stahlberg et al. [27] adopted a hybrid model of SMT and NMT, and a neural language model is added to the method based on the Finite State Transducer (FST).
- Grundkiewicz et al. [28] designed a method for unsupervised generation of parallel sentence pairs via edit distance, fine-tuned the model using annotated error data, and integrated language model and reranking method.

## 4.3. Experimental results and analysis

We first test the overall performance of DCIM on the CoNLL-2014 Test Set and JFLEG Test Set. And then we test the performance of DCIM on different types of grammar errors on CLEC. Finally, the optimal probability threshold of the model is selected through experiments.

### Overall Performance of the Model

The performance comparison between our model and the baseline model on the CoNLL-2014 Test Set is shown in Table 2. It can be seen that DCIM achieves superior results compared to other models. It is worth noting that the model marked with an asterisk was trained using an additional non-public corpus (Cambridge Learner Corpus, CLC), and DCIM still exceeds it by more than 5 percentage points at the F0.5 value. Compared with the SOTA(state-of-the-art) hybrid model of SMT and NMT (Stahlberg et al.), although our model has a slightly lower recall of 0.7 percentage points, it surpassed its accuracy and F0.5 value by 3.7 and 1.8 percentage points, respectively. The effectiveness of the method we proposed in this paper is proved. Compared to the current SOTA models with large parameters (Grundkiewicz et al.), DCIM still falls short. This is because there are gaps that cannot be ignored in terms of corpus resources and computing resources. However, DCIM is smaller and has fewer parameters by comparison. We initialize the weights through the ALBERT pretrained model to make the model more generalizable. The performance of the

model will be further improved when there are sufficient training corpora.

Table 2. The performance of DCIM on the CoNLL-2014 test set.

| Refs. | Model | P | R | F |
|---|---|---|---|---|
| Ji et al. | Nested-GRU * | 55.3 | 26.0 | 45.1 |
| Chollampatt et al. | MLConv (4 ens.) +EO | 62.3 | 27.5 | 49.7 |
| | MLConvembed (4 ens.) +EO +LM + SpellCheck * | 65.4 | 33.1 | 54.7 |
| Stahlberg et al. | SMT+NMT+FST-LM | 66.9 | 38.6 | 58.3 |
| Grundkiewicz et al. | DataAugmentation +Pre-training +Transformer | — | — | 64.1 |
| DCIM | Bi-GRU+ Transformer+ALBERT | 70.7 | 37.8 | 60.2 |

The error correction examples of this model on the CoNLL-2014 test set is shown in Table 3. We list the source sentence, the base correction results without considering the context, the results with context information, and the official reference. Markers in italics indicate that the model made a change to the word at that location. It can be seen that considering the context information of sentences plays an important role in correcting article or determiner errors (ArtOrDet) and subject-verb agreement errors (SVA).

Table 3. The performance of DCIM on the JFLEG test set.

| Type | Output |
|---|---|
| source | As a result, government need more taxed from companies in or-der to have enough money to provide healthcare to elderly people. |
| base | As a result, *governments* need more *taxes* from companies in order to have enough money to provide healthcare to elderly people. |
| DCIM | As a result, *the government needs* more *taxes* from companies in order to have enough money to provide healthcare to elderly people. |
| Refer. | As a result, the government needs more taxes from companies in order to have enough money to provide healthcare to elderly people. |

The performance comparison between DCIM and the baseline model on the JFLEG Test Set is shown in Table 4. Based on previous work, we introduce the GLEU score of human performance to evaluate the adequacy and fluency of the model. As can be seen in Table 4, our model outperforms other baselines except for the large-scale SOTA model of multi-model ensemble by Grundkiewicz et al. While DCIM is less computationally demanding than the large-scale model of Grundkiewicz et al.

Table 4. The performance of DCIM on the JFLEG test set.

| Refs. | Model | GLEU |
|---|---|---|
| Ji et al. | Nested-GRU * | 53.4 |
| Chollampatt et al. | MLConvembed (4 ens.) +EO +LM + SpellCheck * | 57.4 |
| Stahlberg et al. | FST-LM-hybrid | 58.6 |
| Grundkiewicz et al. | DataAugmentation +Pre-training +Transformer | 61.1 |
| Human performance | — | 62.3 |
| DCIM | Bi-GRU+ Transformer+ALBERT | 61.1 |

The good performance of DCIM is due to two points. First, the pre-training model plays an important role. After fine-tuning, it can obtain a more comprehensive semantic representation with less computing resources. Second, the transformer has powerful feature extraction capabilities. Combined with Bi-GRU encoder and attention-based decoder, DCIM more comprehensively combines local and global information of sentences, so the generated sentences are more fluent.

## The Performance of DCIM on Different Types of Grammar Errors

We test the performance of DCIM on different types of grammar errors on 1000 English essays selected from the CLEC corpus. In the selected 1000 English compositions, 1561 grammatical errors were marked. In the course of the experiment, the experimental results are counted in sections, and the statistical results are shown in Table 5.

Table 5. The statistical results of correction under different number of essays.

| Number | Annotatioin | Detectioin | Correction |
|---|---|---|---|
| 100 | 136 | 102 | 81 |
| 200 | 266 | 208 | 167 |
| 500 | 694 | 544 | 441 |
| 800 | 1066 | 827 | 670 |
| 1000 | 1561 | 1226 | 994 |

As can be seen from the table 5, our model detected 1226 grammatical errors, of which 994 were accurately corrected according to the annotations. The overall precision rate is 81.08%, the recall rate is 63.68%, and the F1 value is 71.33%.

We count and organize the results, calculate the precision and recall of each grammar error type, and express them in the form of a histogram, as shown in Fig. 5. Experiments show that DCIM performs well in the correction of article and determiner errors (ArtOrDet), noun singular, plural errors (Nn), verb form errors (Vform) and modal verbs (Vm), especially for subject-verb agreement errors (SVA) and verb absence (V0). This is mainly due to the combination of transformer and Bi-GRU, which integrates local and global information of sentences, and the improved cluster search method can obtain more accurate inference results when decoding.

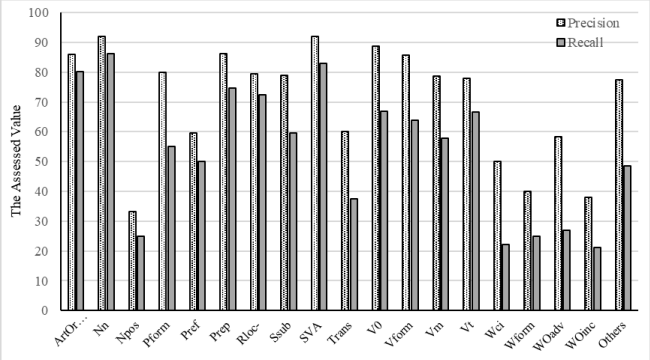


**Figure 5.** Precision and Recall for Different Grammar Errors Types

### The Influence of the Probability Threshold on the Performance of DCIM

In order to obtain the optimal probability threshold of the improved beam search, we test the influence of the probability threshold on the performance of DCIM for the precision, recall and F1 value, and select the optimal threshold p suitable for the model. The experimental results are shown in Fig. 6, it can be seen from the experimental results that when the p value is 0.95, DCIM can maximize the accuracy while considering more candidate results.

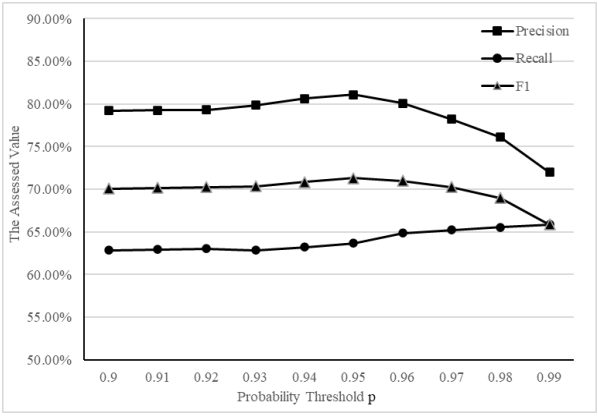


**Figure 6.** The influence of the probability threshold on the performance of DCIM

## 5. Conclusion

This paper proposes an automatic error correction model for English text based on dual encoders. The model combines the local information of the sentence and the global information of the context to correct the grammar errors of the source sentence. The experimental results show that adding context-related information of the sentence can effectively improve the accuracy. However, compared with the method of model integration of large parameter models, there is still a certain gap. The reason is that we chose a lightweight strategy in order to make the model easier to apply, and it is still difficult to surpass some large-scale models at this stage. However, grammar error correction is not limited to parameter expansion and multi-model stacking. In future research, parameters compression method for model ensemble will be considered. This can further reduce the parameters of the model, and the large-scale model will be easier to train.

### Acknowledgements.

This work is supported by the National Natural Science Foundation of China (No. 62066009), the Key Research and Development Project of Guilin (No. 2020010308).

## References

[1] Bentley J. Report from TESOL 2014: 1.5 Billion English learners worldwide[J]. Chicago, IL: International TEFL Academy found online on December, 2014, 19: 2017.

[2] Ranalli J, Yamashita T. Automated written corrective feedback: Error-correction performance and timing of delivery[J]. Language Learning & Technology, 2022, 26(1): 1-25.

[3] Sakaguchi K. Robust Text Correction for Grammar and Fluency[D]. Johns Hopkins University, 2018.

[4] Naber D, A rule-based style and grammar checker[J]. university of Bielefeld, 2003.

[5] Gamon M, Leacock C, Brockett C, Using statistical techniques and web search to correct ESL errors[J]. Calico Journal, 2009, 26(3): 491-511.

[6] Makarenkov V, Rokach L, Shapira B, Choosing the right word: Using bidirectional LSTM tagger for writing support systems[J]. Engineering Applications of Artificial Intelligence, 2019, 84: 1-10.

[7] Hu L, Tang Y, Wu X, Considering optimization of English grammar error correction based on neural network[J]. Neural Computing and Applications, 2022, 34(5): 3323-3335.

[8] Xie Z, Avati A, Arivazhagan N, Neural language correction with character-based attention[J]. arXiv preprint arXiv,1603.09727, 2016.

[9] Hu L, Tang Y, Wu X, Considering optimization of English grammar error correction based on neural network[J]. Neural Computing and Applications, 2021.

[10] Shi Y, Research on English Grammar Error Correction Technology Based on BLSTM Sequence An-notation[J]. Asian Conference on Artificial Intelligence Technology, 2021.

[11] Chollampatt S, H. T. Ng, A multilayer convolutional encoder-decoder neural network for grammatical error

correction[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2018, 32(1).
[12] Xie Z, Genthial G, Xie S, Noising and denoising natural language: Diverse backtranslation for grammar correction[J]. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, (Volume 1): 619-628.
[13] Zhao W, Wang L, Shen K, Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data[J]. arXiv preprint arXiv, 1903.00138, 2019.
[14] Vaswani A, Shazeer N, Parmar N, Attention is all you need[J]. arXiv preprint arXiv, 1706.03762, 2017 .
[15] Cheng L, Ben P, Qiao Y, Research on Automatic Error Correction Method in English Writing Based on Deep Neural Network[J]. Computational Intelligence and Neuroscience, 2022.
[16] Zhou S, Liu W, English Grammar Error Correction Algorithm Based on Classification Model[J]. Complexity, 2021.
[17] Tarnavskyi M, Chernodub A, Omelianchuk K, Ensembling and Knowledge Distilling of Large Sequence Taggers for Grammatical Error Correction[J]. arXiv preprint arXiv, 2203.13064, 2022.
[18] Ge Y F, Orlowska M, Cao J, et al. MDDE: multitasking distributed differential evolution for privacy-preserving database fragmentation[J]. The VLDB Journal, 2022: 1-19.
[19] Ge Y F, Yu W J, Cao J, et al. Distributed memetic algorithm for outsourced database fragmentation[J]. IEEE Transactions on Cybernetics, 2020, 51(10): 4808-4821.
[20] Li J Y, Zhan Z H, Wang H, et al. Data-driven evolutionary algorithm with perturbation-based ensemble surrogates[J]. IEEE Transactions on Cybernetics, 2020, 51(8): 3925-3937.
[21] Alvi A M, Siuly S, Wang H. A long short-term memory based framework for early detection of mild cognitive impairment from EEG signals[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2022.
[22] Siuly S, Khare S K, Bajaj V, et al. A computerized method for automatic detection of schizophrenia using EEG signals[J]. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2020, 28(11): 2390-2400.
[23] Shi W, Chen W N, Kwong S, et al. A coevolutionary estimation of distribution algorithm for group insurance portfolio[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021.
[24] Keskar N S, McCann B, Varshney L R, et al. Ctrl: A conditional transformer language model for controllable generation[J]. arXiv preprint arXiv:1909.05858, 2019.
[25] Dahlmeier D, H. T. Ng, S. M. Wu, Building a large annotated corpus of learner English: The NUS corpus of learner English, Proceedings of the eighth workshop on innovative use of NLP for building educational applications. 2013: 22-31.
[26] Ji J, Wang Q, Toutanova K, et al. A nested attention neural hybrid model for grammatical error correction[J]. arXiv preprint arXiv:1707.02026, 2017.
[27] Stahlberg F, Bryant C, Byrne B. Neural grammatical error correction with finite state transducers[J]. arXiv preprint arXiv:1903.10625, 2019.
[28] Grundkiewicz R, Junczys Dowmunt M, Heafield K. Neural grammatical error correction systems with unsupervised pre-training on synthetic data[C]//Proceedings of the Fourteenth Workshop on Innova-tive Use of NLP for Building Educational Applications. 2019: 252-263.