IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 December 2022, accepted 9 January 2023, date of publication 18 January 2023, date of current version 23 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3237829

RESEARCH ARTICLE

Unsupervised Grammatical Correction With Optimized Layer Normalization and Dynamic Embedding Enhancement

YIN WANG[1] **AND ZHENGHAN CHEN**[2,3]
[1]School of Journalism and Communication, Jinan University, No. 601 Huangpudadaoxi Rd, Guangzhou, China
[2]Knowmeta Company Ltd., Shanxi 030000, China
[3]School of Software and Microelectronics, Peking University, Beijing 100091, China

Corresponding author: Zhenghan Chen (1979282882@pku.edu.cn)

ABSTRACT Grammatical error correction aims to detect and correct grammatical errors with all types of mistaken, disordered, missing, and redundant characters. However, most existing methods focus more on detecting errors than correcting them. This paper proposes a domain-adaptive model with Interoperable Layer Normalization (ILN) and dynamic word embedding enhancement to optimize the error correction capability. To further improve the Chinese correction capability, we introduce multiple rounds of error correction to refine the sequence tagging model's ability to fix mistakes. In addition, we propose a data augmentation method based on the complex tag to represent textual error correction traces more completely. We also explore a migration training method based on multiple training datasets. Further, we offer a unique unsupervised domain adaptation technique based on ILN, an innovative channel fusion approach that can significantly improve models' domain adaptability. Finally, experimental results show that our proposed method substantially outperforms all robust baseline methods and achieves the best results in position-level and correction-level errors on the CGED-2020 dataset.

INDEX TERMS Grammatical correction, layer normalization, dynamic embedding enhancement.

I. INTRODUCTION

The goal of grammatical error correction is to identify and fix grammatical errors in documents [1]. Many NLP applications, including writing assistants [2], search engines [3], and optical character recognition systems [4], can benefit from it. Compared with English, Chinese does not have strict grammar rules (i.e., no grammatical requirements such as tense, singular, and plural), and there is no space as a separator between words in Chinese text. The independence between various grammatical errors in Chinese is relatively substantial. The demand for Chinese grammatical error correction is rising along with Chinese popularity.

With the development of deep learning, there are two primary methodologies for Chinese grammatical error correction: machine translation-based models and sequence tagging-based models. Reference [5] use a machine translation-based approach for grammatical error correction and find that it is more effective in dealing with missing token errors and token redundancy errors. Reference [6] use a deep convolutional encoder-decoder model with attention mechanisms for grammatical error correction. Reference [7] utilize BiLSTM to build a neural translation model. Thanks to excellent neural networks like Transformer [8], and BERT [9], methods based on sequence tagging are becoming increasingly crucial for correcting Chinese grammar mistakes. Reference [10] uses a sequence tagging-based approach combined with BERT that performs well on grammatical error detection.

Furthermore, most existing methods require a separate grammatical error correction model for modification after detecting a grammatical error. Reference [11] add a mask to the erroneous token and feed it into the BERT model for text prediction. Reference [12] propose a grammatical

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

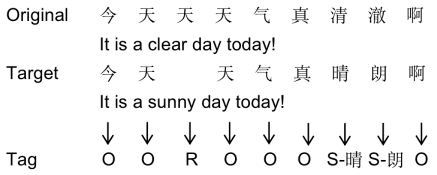


FIGURE 1. Illustration of grammatical error types. The English in the figure is a translation of the Chinese. The 'O' tag means the token is correct. 'R' indicates a word redundancy error. 'S-' indicates a word selection error and gives grammatical error correction information.

error correction model for English. The model optimizes the tag label as a combination of error type plus error correction content, which significantly improves the grammatical error correction capability of the sequence tagging method. However, the sequence detection capability of this method is significantly lower than that of other sequence tagging methods.

In this paper, we treat the grammatical error correction task as an enhanced sequence tagging task, aiming to enhance the grammatical error correction capability without degrading the detection capability. Each erroneous token is assigned with an error type, e.g., selection errors and redundant words, as shown in Figure 1. We propose a novel dynamic word embedding enhancement model with a residual connection network to improve grammatical error correction capability. The dynamic word embedding enhancement model can extensively utilize the prior knowledge gathered through pretraining approaches and is instructive for processing each word vector in the text. The residual connection network can effectively suppress network degradation and is helpful for most sequence tagging tasks with identical labels. To compensate for the shortcoming that the text input length in the sequence tagging model is strictly consistent with the label sequence output length, we utilize a multi-round error correction method. To better show the traces of model modifications, we propose a complex tagging method for Chinese text. We also propose a data augmentation method based on complex tagging due to the shortage of training data. The complex tag-based data augmentation method ensures that the style and context of the expended text are highly consistent with the original text, substantially enriching the training data.

At the same time, researchers have worked very hard to address the weak generalization performance of models which could correct Chinese grammatical errors. Since it does not require label information in the target domain, unsupervised domain adaptation (UDA), particularly, is receiving much attention [13]. Academics have recently begun examining different avenues, like creating batch normalization [14] layers using domain-specific information. These Batch Normalization (BN) based [15], or Layer Normalization (LN) based methods [16], which provide equal weight to each channel, might not be the best for domain adaptation. Therefore, we propose a novel ILN that may highlight and transfer key and transferable channels by utilizing the Layer Normalization scaling factor.

We evaluate our model on the CGED-2020 dataset, and our proposed architecture substantially improves position and correction levels. We investigate a migration training strategy that uses several training datasets. In summary, we make the following contributions:

- We propose a novel dynamic word embedding enhancement model with a residual connection network to promote the text representation to encode Chinese input text. Our approach can unify the detection and correction into one architecture according to the Chinese complex tag.
- We propose a new data augmentation method based on a complex tag to tackle the data scarcity problem, which can expand the training data and thus improve the grammatical error correction capability of the model.
- We theoretically prove that the scaling factors for some channels will come close to zero and reveal that the scaling factors of the LN can indicate the transferability of a channel. Thus, we propose ILN in place of existing LN techniques to fuse different channels.
- Experimental results on the CGED-2020 dataset show that our approach substantially outperforms several solid baselines and achieves state-of-the-art performance on position and correction levels.

II. RELATED WORK

A. MACHINE TRANSLATION-BASED METHODS

The machine translation-based method can be regarded as a sequence-to-sequence (seq2seq) method, which can translate incorrect sentences into correct ones. In the past few years, several methods have been proposed to improve the performance of machine translation-based grammatical error correction models. Reference [17] incorporate a pretrained masked language model like BERT into a seq2seq model. Reference [18] propose a copy-augmented architecture for grammatical error correction. Reference [19] focuses on constructing additional synthetic data for pretraining using translation models. Reference [20] directly adds noise to standard sentences to back-translation. For Chinese text, [6] construct a seq2seq model with a multi-layer convolution and attention mechanism. Reference [7] propose a BiLSTM-based machine translation model to capture long-distance interdependency. Despite the above enhancements, the machine translation-based models still suffer from generating results from scratch, which unavoidably leads to overcorrection and generation errors.

B. SEQUENCE TAGGING-BASED METHODS

Another line of research takes a different angle by framing the grammatical error correction task as a sequence tagging task. These models generally predict a predefined set of tags based on the source sentence and make an edition to erroneous tokens.

Reference [21] predicts editions between keeping, deleting, or adding a new token/phrase from a predefined dictionary. Meanwhile, [22] predicts token-level editions sequentially for a fixed number of iterations in a non-autoregressive way. Reference [23] generate span-level tags to generate more compact editions. Reference [12] advance the approach further by designing finer-grained editions based on English lexical rules. While performing reasonably well in English grammatical error correction tasks, these models are not suitable for Chinese grammatical error correction tasks. Besides, most models mentioned above focus on identifying grammatical errors rather than correcting them. For Chinese grammatical error correction, [11] add a mask to the text positions labeled as missing errors and use BERT for text prediction. However, this approach still requires a separate correction model for grammatical error correction.

To further improve the Chinese grammatical error correction capability, we transpose the English grammatical error correction model GECToR [12] to Chinese grammatical error correction, incorporating a novel dynamic word embedding enhancement with a residual connection network. In the meantime, we improve the model's ability to deal with complex tags by training it on an augmented dataset.

III. METHODOLOGY

A. NORMALIZATION OPTIMIZATION

The most well-known is Batch Normalization (BN) [24]. Every layer's inputs are converted to zero in terms of mean and variance before being scaled and shifted using two trainable parameters. The BN-based or LN-based approach has recently been applied in specific landmark works to solve this issue. In order to minimize the domain shift, AdaBN [25] makes use of target statistics at inference. At the training step, AutoDIAL [14] introduces a linear combination of the source and target features into BN. Each BN layer involves an additional parameter resulting from a parameter trade-off between the source and target domains. Using BN parameters, DSBN [26] gathers domain-specific data and converts it into domain-invariant representations. TransNorm [27] replaces the current BN-based architecture by assuming that the lack of transferability is mainly caused by the inherent limits of CNN's architecture design.

B. PROBLEM FORMULATION

Our goal is to generate and correct grammatical errors in Chinese text. Given a sequence of n tokens $X = (x_1, x_2, \ldots, x_n)$, the goal is to transform it into an m-character sequence $Y = (y_1, y_2, \ldots, y_m)$, where n and m could be the same. We aim at correcting X to Y through a multi-round correction method.

C. MODEL

Our model can be viewed as a sequence tagger with dynamic word embedding enhancement through a residual connection network. An illustration of the proposed framework is shown in Figure 2. The sequence tagging model with complex tags generates grammatical error types and their corresponding correction information. Benefiting from dynamic word embedding enhancement through residual connection network, the prior knowledge obtained by pre-trained models can help the model to get better word embedding representation. We utilize a multi-round correction method to further improve the grammatical error correction ability. Besides, we design a data augmentation method based on complex tags, which ensures a high degree of consistency between the style of the expanded text and the original text. The following sections describe the sequence tagging task, the dynamic word embedding enhancement approach through the residual connection network, and the data augmentation method.

1) SEQUENCE TAGGING TASK

Sequence tagging methods are commonly used in NLP and have been widely used in tasks such as Chinese word separation, lexical annotation, and named entity recognition. Sequence tagging-based grammatical error correction methods require tagging each text word with an error type, which is a word-level classification process. By estimating, these models generally predict a tag sequence Y based on the source sentence X.

$$p(Y|X) = \prod_{i=1}^{m} p(y_i|X) \tag{1}$$

As shown in Figure 2, $(e'_{\cdot 1}, e'_{\cdot 2}, \ldots, e'_{\cdot m})$ denotes the feature vector of each word in the sequence after the pre-trained model operates the input sequence. The feature vector is classified to obtain the label sequence Y. Unlike standard classification, each tag may be connected to the other. The algorithm must evaluate the whole sequence based on the tags' relationship to obtain the highest probability annotated sequence.

However, the general grammatical error correction sequence annotation label contains only error type information and cannot correct grammatical errors. Following [12], we add error correction information to the tags so that the model has grammatical error correction capability.

We classify Chinese grammatical errors into four categories: selection errors (S), redundancy errors (R), missing errors (M), and disordering errors (D). As shown in Figure 1, for correct words in the text and text errors that do not require further, the model is labeled in the same way as the general sequence tagging, directly labeled as 'R' (redundancy errors) or 'D' (disordering errors). Labels combine error type and grammatical error correction information for selection and missing errors. 'S-qing' means a selection error in the original text for 'qing' (transparent), which should be replaced with 'qing' (sunny). 'S-lang' means that the original text's word 'che' (clear) was incorrectly chosen and should be replaced with 'lang' (sunny). On the other hand, for missing errors, 'M-token' needs to be added after the word where the missing error appears in the original text. At last, for complex grammatical error correction, we treat it as a combination

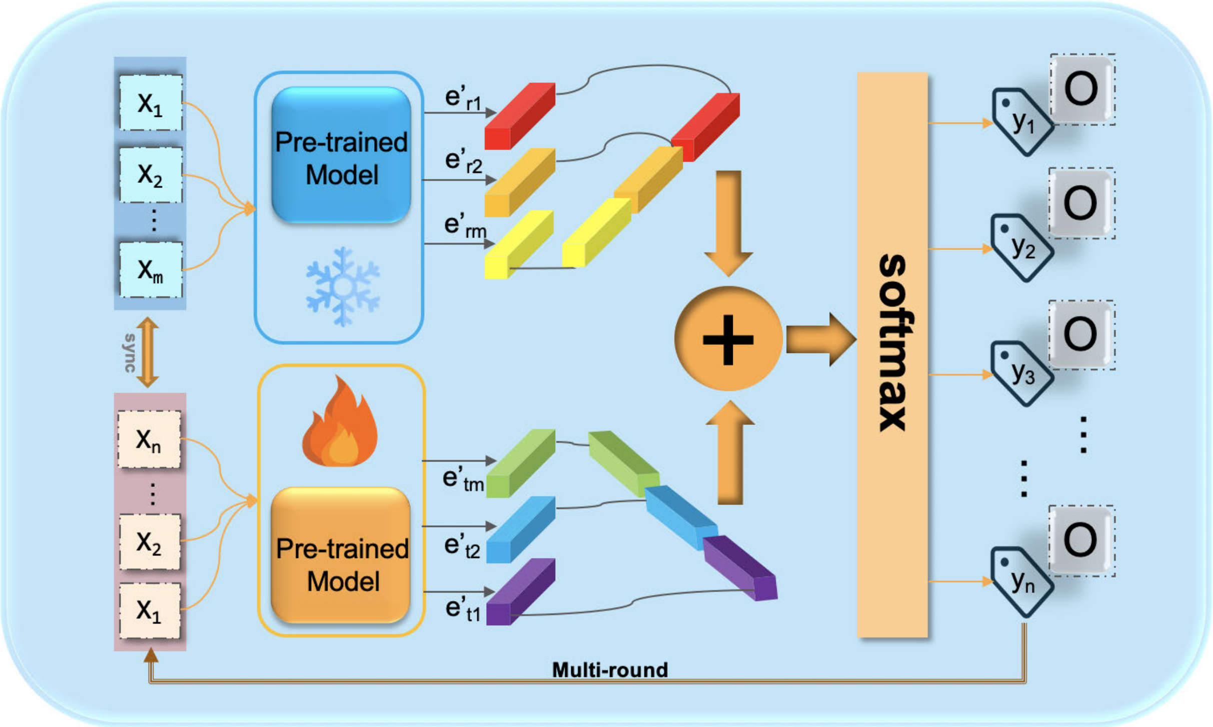


FIGURE 2. The figure shows the overall architecture of the proposed model in this paper. We treat the grammatical error correction task as a sequence tagging task. The error correction capability is improved by multi-round error correction until the labels change to 'O'. The middle part of our model is dynamic word embedding enhancement with a residual connection network, which helps the model better represent the words. The blue part indicates the pre-trained model with frozen parameters, which introduce prior knowledge to the model. The red part indicates the fine-tuned pre-trained model. The model robustness can be improved by connecting the two parts through a residual network.

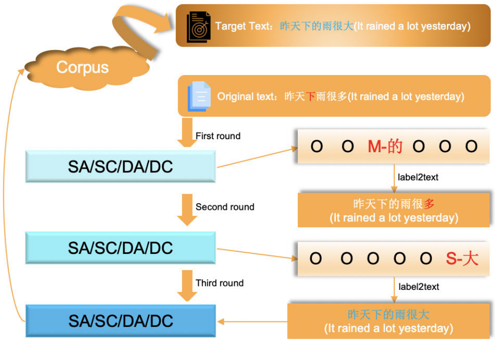

Target Text：昨天下的雨很大(It rained a lot yesterday)
Corpus
Original text：昨天下雨很多(It rained a lot yesterday)
First round
SA/SC/DA/DC
O O M-的 O O O
label2text
昨天下的雨很多
(It rained a lot yesterday)
Second round
SA/SC/DA/DC
O O O O O S-大
Third round
label2text
SA/SC/DA/DC
昨天下的雨很大
(It rained a lot yesterday)

FIGURE 3. Example of multi-round grammatical error correction based on multiple error types. In addition, the English in the figure is a translation of the corresponding Chinese sentence.

of different error types. For example, correcting 'ta' (he) to 'ni men' (you), ordinary sequence tagging can only annotate 'ta' as 'S-B'. However, our proposed model can label 'ta' as 'S-ni|M-men', which means replacing 'ta' with 'ni' and adding 'men' after it. Therefore, our proposed model can directly perform textual error correction by sequence tagging and has rich error correction capability.

Models for sequence tagging tasks often consist of a feature extraction network and a classification network. The computation process of the pre-trained model can be seen as a word vector feature extraction process, and standard pre-trained models are BERT [9], RoBERTa [28]. However, the pre-training and fine-tuning process of the above-pre-trained models could be more consistent, and there are few corresponding [MASK] tokens in the input text for the model to use during the fine-tuning process of the actual task. To increase the consistency of the two processes, we utilize MacBERT [29] as the pre-trained model for our proposed method.

Due to the limitation of the sequence tagging model, the length of the output tag sequence is the same as the length of the input text, so the model can only add, delete and change a single word for each prediction. The model cannot complete the error correction task simultaneously for more complex correction needs. To solve this problem, we introduce multiple rounds of error correction to improve the error correction capability of the sequence tagging model. As shown in Figure 3, for the original text "zuo tian xia yu hen duo" (It rained a lot yesterday) and the target text "zuo tian xia de yu hen da" (It rained a log yesterday),[1] there is a word missing error and a word selection error. If the model cannot identify multiple errors simultaneously, all errors can be found by correcting them multiple times.

[1] Although the English translation is identical, there are grammatical errors in Chinese, including missing error and selection error.

TABLE 1. S (static) and D (dynamic) mean that the network is based separately on static word embedding and dynamic word embedding. A (add) means that the residuals of this network are realized as vector accumulation or difference-making. C (concatenate) means that the network residuals are realized as vector concatenation.

Model	Description	Residual Connection Network
SA	Accumulation residual network based on static word embedding	$E_{out} - E_{in}$
SC	Concatenation residual network based on static word embedding	$concat(E_{in}, E_{out} - E_{in})$
DA	Accumulation residual network based on dynamic word embedding	$E_{fine-tuning} - E_{freezed}$
DC	Concatenation residual network based on dynamic word embedding	$concat(E_{freezed}, E_{fine-tuning} - E_{freezed})$

2) DYNAMIC WORD EMBEDDING ENHANCEMENT WITH RESIDUAL CONNECTION

The ideal annotation result for a text $T_n = \{w_1, w_2, \ldots, w_n\}$ that does not contain any word errors should be $\{O, O, \ldots, O\}$. For the input word embedding $E_{in} = \{e_1, e_2, \ldots, e_n\}$ and the pre-trained model output (without fine-tuning) of dynamic word embedding $E_{out} = \{e'_1, e'_2, \ldots, e'_n\}$, $E_{out} - E_{in} = \{e'_1 - e_1, e'_2 - e_2, \ldots, e'_n - e_n\}$. Moreover, in texts of tens of words long, the error-related words are often only a few, and the word retention label 'O' is predominant. Therefore, such a residual structure is of practical importance to improve the robustness of the model.

$$\begin{aligned} E_{out} &= E_{fine-tuning} - E_{freezed} \\ &= \{e'_{t1} - e'_{r1}, e'_{t2} - e'_{r2}, \ldots, e'_{tm} - e'_{rm}\} \end{aligned} \quad (2)$$

For the input word embedding $E_{in} = \{e_1, e_2, \ldots, e_n\}$ and the dynamic word embedding $E_{out} = \{e'_1, e'_2, \ldots, e'_n\}$ output by the pre-trained model (without fine-tuning), the dynamic word embedding of a Chinese character w in the text is very different from the static word embedding e.

The blue part is the MacBERT without fine-tuning, which is used to output the dynamic word embedding $E_{freezed} = \{e'_{r1}, e'_{r2}, \ldots, e'_{rm}\}$ parsed by preliminary MacBERT. The red part is the MacBERT with fine-tuning. Finally, the residual results E_{out} are classified to obtain the final error correction information.

The model combines the dynamic word embedding output from the original MacBERT (without fine-tuning), which can exceptionally extensively exploit the prior knowledge obtained from the original model trained on a massive amount of text, which is a guideline for the processing of each word embedding in the text.

There are two main ways to combine residual connections, concatenation or accumulation. Suppose X is the residual layer input, X' is the original input features, Y is the residual layer output, and $F(X)$ is the residual term. The basic idea of the concatenation residual-based connection is to highly retain the features of each dimension of the existing input X and learn the X' separately. For the grammatical error correction task, the dynamic word embedding X without fine-tuning can be used to retain the prior knowledge acquired by the original MacBERT model. Then the fine-tuned MacBERT model can be used to obtain the dynamic word embedding X that fits the actual error correction task, which can effectively expand the word feature extraction of the model and improve the grammatical error correction effect of the model.

$$Y = concat(X', X) \quad (3)$$

The basic idea of the cumulative residual-based connection is that the model trusts more in its input X and tries to perform feature learning based on the input; $F(X)$ is the incremental change of the input in each dimension. This can effectively prevent model degradation, and the least desirable case is the residual term, at which time the output of the current layer $Y = X$.

$$Y = F(X) + X \quad (4)$$

According to two types of residuals, namely, accumulation and concatenation, and two types of word feature extraction, namely, static word embedding and dynamic word embedding, four grammatical error correction models are constructed in this paper, and the differences of each model are shown in Table 1.

3) DATA AUGMENTATION

Data augmentation based on complex tags can expand the text data while highly preserving the style and content of the original text, effectively expanding the size of the training dataset. After using our data enhancement method proposed below, the dataset size is enlarged by nearly ten times compared to the original dataset.

Ideally, each tag in a tagging sequence should be modified once per word. However, due to the occurrence of a word missing error, when the text is missing multiple words, the word at a particular tag position modification step will be greater than once. This paper refers to such tags with more than one modification step as **complex tags**.

There are two general English grammatical error correction approaches when dealing with complex tags. The first method is to take only the first step of the modification. The second approach is to treat the complex tag as a phrase-based modification, i.e., the complex tag is treated as a separate unique tag. Although both methods reduce the pressure of parsing text, they reduce the possibility of recovering the target text and are unsuitable for Chinese grammatical error correction.

Complex tags are mainly formed because of missing multi-word phrases and may co-occur with word selection errors. Therefore there are two types of complex tags:

$$label_M = M\_word_1|M\_word_2|\cdots|M\_word_n \quad (5)$$

$$labels_S = S\_word_1|M\_word_2|\cdots|M\_word_n \quad (6)$$

where *word* is the missing word or the word to be replaced, for $label_M$, $text = \{word_1, word_2, \ldots, word_n\}$ needs to be added at the corresponding position of the original text. For $label_S$, use $word_1$ to replace the word at the corresponding position in the original text, followed by $text = \{word_2, \ldots, word_n\}$.

In order to reduce the difficulty of the model in performing grammatical error correction, we optimize the complex tags into a combination of text and base tags. We propose a step-by-step reduction approach that expands the corresponding text and labels to the reduction steps required for each step of a complex tag. For complex tag $label_M$, we expand it to

$$out_{1\leq m\leq n} = (\sum_{1}^{m-1} word_i) + M\_word_m \tag{7}$$

That is, the short text $\sum_1^{m-1} word_i$ is added to the original text in order, the last word in the short text is marked as M_word_m, and the other words in the short text are marked as O. As long as circular error correction is introduced to the model, the model may restore the target text entirely in each error correction of one word at a time.

For complex tag $label_S$, we expand it to:

$$out_{1\leq m\leq n} = \begin{cases} S\_word_m & m = 1 \\ (\sum_1^{m-1} word_i) + M\_word_m & 2 \leq m \leq n \end{cases} \tag{8}$$

To further enrich the error correction capability of the model, an attempt is made to expand the complex tags to contain only one under the complete text. In other words, each error in the complex tag is regarded as an item to be filled in the complete text. For complex tag $label_M$, we expand it to:

$$out_{1\leq m\leq n} = (\sum_{1}^{m-1} word_i) + M\_word_m + (\sum_{m}^{n} word_{i+1}) \tag{9}$$

Similarly, for complex tag $label_S$, we expand it to:

$$out_{1\leq m\leq n} = \begin{cases} S\_word_m & m = 1 \\ (\sum\limits_{1}^{m-1} word_i) + M\_word_m & \\ \qquad +(\sum\limits_{m}^{n} word_{i+1}) & 2 \leq m \leq n \end{cases} \tag{10}$$

The step-by-step reduction strategy attempts to restore the target text via a circular error correction method. The complex tag simplification approach allows the model to restore the target text in a single step by changing the original text suitably. The original text, including complex tags, may be enlarged with multiple pairs of the parallel corpus by using both approaches above in the data augmentation. The complex tag-based data augmentation approach may guarantee that the extended text's style and content are compatible with the original text.

When performing data augmentation, we expand the parallel corpus and remove complex tags that are too long in the text, which are not utilized as training data. For $label_M$ or $label_S$, when $n > 4$, the parallel corpus is removed from the training data.

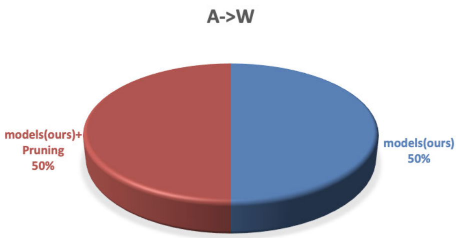


FIGURE 4. A-distance after domain adaption is used to quantify the discrepancy in the distribution.

4) ILN FOR GENERALIZATION

We initially used A-distance on the A→W problem with models proposed to demonstrate that channels with tiny scaling factors have little impact on domain adaptation. When we prune several channels with scaling factors near zero, we plot the A-distance on task A→W in Figure 4. The A-distance of models(SA/SC/DA/DC)+Pruning is nearly identical to the original one, as shown in Figure 4.

When λ is getting close to zero, $\frac{\partial \mathcal{H}}{\partial x}$ will likewise be. In reality, there could not be a lot of LN channels with λ values close to zero.

$$\mathcal{H} = \mathcal{H}(g(x, W), y) + \alpha \sum_{h=1}^{H} s(\lambda) \tag{11}$$

$s(\alpha)$ represents a scaling factor penalty brought on by sparsity, and an equalizing term, α, balances the two components.

Assume that λ is the domain adaptation loss function specified in Eq. 11 and that $\lambda_c^{(j)}$ is the LN scaling factor for the various domain networks. After that, the domain adaptation process' gradient of the loss function λ concerning $\lambda_c^{(j)}$ is as follows (for this proof, we select $|\lambda| \geq 1$). Please refer to the supplemental file for more information on the specific proving procedure in this case.) When $|\lambda| < 1$, we may reach the same result.

$$\frac{\partial \mathcal{H}}{\partial \lambda_c^{(j)}} = \frac{\partial \mathcal{H}}{\partial x_c^{\prime(j)}} \frac{x_c^{(j)} - \mu_c^{(j)}}{\sqrt{\beta_c^{2(j)} + \varepsilon^{(j)}}} + \alpha \frac{\partial \lambda_c^{(j)}}{\partial \left|\lambda_c^{(j)}\right|} \tag{12}$$

Additionally, we can have the probability distribution shown below between the range mentioned above:

$$Q = \gamma\left(\alpha\left(\frac{\partial \mathcal{H}}{\partial x_c^{(j)}}\right)^{-1}\right) - \gamma\left(-\alpha\left(\frac{\partial \mathcal{H}}{\partial x_c^{(j)}}\right)^{-1}\right) \tag{13}$$

Convergence, $\frac{\partial \mathcal{H}}{\partial x_c^{(j)}} \rightarrow 0^+$, and $\left(\frac{\partial \mathcal{H}}{\partial x_c^{(j)}}\right)^{-1} \rightarrow +\infty$ training can result in $2\gamma\left(\alpha\left(\frac{\partial \mathcal{H}}{\partial x_c^{(j)}}\right)^{-1}\right) - 1 \rightarrow 1$. This demonstrates that some channels have a probability of 1 or approaching zero. As a result, we have demonstrated that when a sparsity

TABLE 2. Main results on the CGED-2020 dataset. The best results are in bold. ∗ denotes the results from [30]. † and ‡ represent the first-ranked and second-ranked results, respectively.

Model	Detection	Identification	Position	Correction
Seq2Seq*	86.76	58.29	31.40	-
BackTranslation*	87.03	59.89	31.92	-
ADV*	87.11	60.20	32.81	-
CNEG*	88.12	62.00	33.99	-
BERT	79.52	53.10	35.46	22.67
MacBERT	81.43	52.33	37.68	23.13
S2A Model	90.57	64.82	38.94	20.41
SE-CGED	90.36	67.99	46.45‡	21.97
Alignment-Agnostic	81.32	63.93	40.61	20.17
CGED2020_best	91.22	67.36	40.41	18.91
SA (ours)	90.23	66.43	40.60	22.97
SC (ours)	91.14	63.84	40.26	23.01
DA (ours)	92.16	67.96	39.82	24.12
DC (ours)	91.59	68.37	42.03	24.91‡
SA (+ILN)	91.67	67.71	41.52	23.07
SC (+ILN)	92.63	65.26	40.89	23.92
DA (+ILN)	93.22‡	68.67‡	41.57	24.75
DC (+ILN)	**93.74**†	**70.13**†	**47.48**†	**26.15**†

regularization is applied, some channels' scaling factors will be very close to zero.

Thus, this channel fusion can simultaneously decrease ineffective feature transfer in one domain. So, the LN to ILN is improved as follows:

$$x_c^{(s)} = \begin{cases} \lambda_c^{(s)} \frac{x_c^{(s)} - \mu_c^{(s)}}{\sqrt{\beta_c^{2(s)} + \varepsilon}} + \beta_c^{(s)}, & \text{if } \lambda_c^{(s)} > k \\ \frac{1}{C} \sum_{c' \neq c}^{C} \lambda_{c'}^{(t)} \frac{x_{c'}^{(t)} - \mu_{c'}^{(t)}}{\sqrt{\beta_{c'}^{(t)} + \varepsilon}} + \beta_{c'}^{(t)}, & \text{if } \lambda_c^{(s)} < k \& \lambda_c^{(t)} > k \end{cases} \tag{14}$$

s indicates the source domain or target domain. If the current channel c scaling factor is less than a predetermined threshold k, the current channel c is replaced with the mean of other channels in the associated target domain or source domain.

IV. EXPERIMENTS

A. DATASETS

We experiment with the latest released CGED-2020 dataset [1], which is a competition dataset. CGED-2020 dataset incorporates foreign Chinese learners' writing and contains four error types mentioned above. We also experiment with the writing section of the HSK dataset. It has 1,457 text units, each containing 1-5 sentences. More than 20% of the sentences do not contain grammatical errors. It contains 769 cases of redundancy errors, 864 missing errors, 1,694 selection errors, and 327 disordering errors. In addition, we used the same dataset processing method as the baseline models to complete the splitting of the training, validation, and test sets.

For complex tag-based data augmentation, we choose the HSK dataset [31], lang8 dataset [32] and CTC dataset.[2]

- HSK dataset, a dynamic composition corpus for the Chinese Proficiency Test created by Beijing Language and Culture University. The dataset's content is mainly from the HSK essay exams from 1992 to 2005, which were answered by international students from various countries who took the exams. The HSK dataset contains 156,870 lines of the parallel corpus, and the entire HSK dataset is retained for the grammatical error correction task in this paper.
- The content of the lang8 dataset comes from the lang-8 Language Learning Exchange Community,[3] which an NLPCC 2018 GEC Shared Task published. This paper is based on processing the lang8 dataset into a parallel corpus and retaining the textual similarity between the target text and the original text with greater than 70% of the sentence pairs. After processing, the final lang8 parallel corpus of 701,364 sentences was obtained.
- CTC dataset is provided by the Chinese Text Correction Track of the 3rd China AI Innovation and Entrepreneurship Competition in 2021. Unlike the HSK and lang8 datasets, both texts written by foreign language learners, the CTC dataset contains pseudo data. The CTC dataset content was selected from Chinese Internet web texts,

[2] https://2021aichina.caai.cn/

[3] https://lang-8.com

and errors were artificially added to the correct texts. The CTC dataset has a total of 317,634 sentences of the parallel corpus, which features a small editing distance. The error types and the number of errors in the text are very suitable for the Chinese text correction task. However, a small part of the parallel corpus has error correction of English words, and this part of the text is excluded in this paper when using the CTC dataset.

B. IMPLEMENTATION DETAILS

We use Adam optimizer with an initial learning rate $\alpha = 0.00001$, momentum $\beta_1 = 0.9$, $\beta_2 = 0.999$ and weight decay $\epsilon = 10^{-5}$. We use a size of 16, and the length of truncated sentences is 128. For MacBERT that requires fine-tuning, we set cold_step as 1; that is, MacBERT is then added to the training fine-tuning after one round of training in the classification network. The initial learning rate of cold_step is 10^{-3}.

We use a migration training method for multiple training datasets following [12]. First, we train on the CTC dataset to learn the Chinese grammatical error correction task. Then we perform fine-tuning training on the lang8 dataset to further learn Chinese grammatical error correction. Finally, the final fine-tuning is done in the HSK dataset. We only use text in the dataset that contains grammatical errors.

C. EVALUATION METRICS

We use the evaluation metric used in [1]. It contains four levels:

- **Detection-level.** It is a binary classification task to determine whether the text is correct. For a given text, it can only be marked as *Wrong* or *Correct*, strictly according to the test set. If a text contains a grammatical error in the task, it is incorrect.
- **Identification-level.** It is a multiclassification task to determine the type of grammatical errors. For a given text, the type of textual error contained in that text needs to be marked. Even if the text contains more than one grammatical error of a certain type, it is still marked once.
- **Position-level.** It is a character-based multiclassification task that indicates where a grammatical error occurs. For a given text, the starting and ending positions of that grammatical error must be pointed out based on correctly determining the type of grammatical error.
- **Correction-level.** For word selection and missing errors, the corresponding corrections need to be given based on the correct labeling of the error type and location.

D. BASELINES

We compare our augmentation method with several baseline methods.

- **Seq2Seq** [33]. The model uses the right sentences as inputs and the incorrect ones as predictions.
- **BackTranslation** [34]. The model transforms the original sentence into a bridge language, then back into the source language. English is the bridging language in this experiment.
- **ADV** [35]. The approach creates adversarial instances by identifying weak spots and replacing them with correction-to-error mapping.
- **CNEG** [30]. The model masks a span in a correct text and then predicts an erroneous span conditioned on both the masked text and the correct span.
- **BERT** [36]. The model is designed to pre-train deep bidirectional representations from the unlabeled text by joint conditioning on both the left and right context in all layers.
- **S2A Model** [39]. Sequence-to-Action (S2A) module jointly takes the source and target sentences as input, and is able to automatically generate a token-level action sequence before predicting each token.
- **SE-CGED** [40]. SE-CGED requires less training data by using a unified workflow to handle various types of grammatical errors. Two measures are proposed in this model to enhance the performance of CGED.
- **Alignment-Agnostic** [41]. Alignment-Agnostic is a novel alignment-agnostic detect-correct framework that can handle both text aligned and non-aligned situations and can serve as a cold start model when no annotation data are provided.
- **Competition Results** [1]. We take the detection-level result of NJU-NLP_run1, the identification-level result of Flying_run2, the position-level result of Flying_run3, and the correction-level result of UNIPUS-Flaubert as the best result combination.

E. AUTOMATIC EVALUATION

The experimental results on the CGED-2020 dataset are shown in Table 2. The experimental results indicate that the seq2seq-based methods could be better than the sequence tagging-based methods. The ADV method achieves relatively good results because the contextual information is considered. With ILN's optimization, DA and DC have achieved the best performance in detection-level, identification-level, position-level, and correction-level. The results demonstrate that our proposed model outperforms other models regarding its grammatical correction capability.

F. ABLATION STUDY

For further analyzing the effectiveness of the components of our proposed model, we conduct ablation studies as follows:

- **Effects of Data Augmentation and Residual Connection.** We test the effect of the models before and after data augmentation on the HSK dataset. As shown in Table 3, for the same model, after being augmented by the data based on the complex tags, the model effect is improved under all evaluation metrics. The experimental phenomenon is consistent in the lang8 dataset and

TABLE 3. Comparison of the effect of five models of gector*/SA/SC/DA/DC before and after data enhancement. Gector* represents MacBERT replaced gector model. hsk_multi refers to the HSK dataset augmented with complex tag-based data.

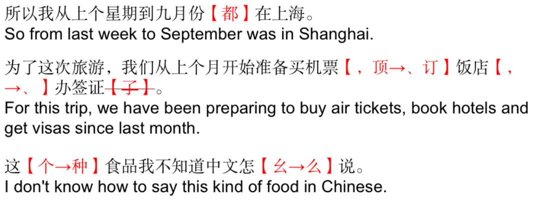
Model	Training data	Detection	Identification	Position	Correction
gector*	hsk	75.31	44.99	32.31	22.1
	hsk_multi	**77.39**	**48.16**	**33.98**	**23.27**
MacBERT	hsk	72.68	41.47	31.83	20.53
	hsk_multi	**76.74**	**47.64**	**33.12**	**22.79**
SA	hsk	83.72	58.24	35.69	21.56
	hsk_multi	**86.23**	**60.43**	**37.60**	**21.97**
SC	hsk	86.50	52.98	33.56	21.79
	hsk_multi	**88.73**	**57.94**	**35.71**	**22.01**
DA	hsk	83.47	53.15	34.63	21.95
	hsk_multi	**86.34**	**61.52**	**35.87**	**23.58**
DC	hsk	82.34	54.21	36.27	22.94
	hsk_multi	**89.47**	**59.85**	**39.62**	**23.50**
SA(+ILN)	hsk	84.26	58.97	36.05	22.19
	hsk_multi	**88.42**	**62.56**	**39.44**	**22.19**
SC(+ILN)	hsk	87.66	54.08	35.24	22.13
	hsk_multi	**89.15**	**58.72**	**36.97**	**22.81**
DA(+ILN)	hsk	85.57	55.93	37.42	23.60
	hsk_multi	**89.37**	**65.14**	**38.75**	**24.21**
DC(+ILN)	hsk	86.63	62.74	37.98	24.18
	hsk_multi	**91.26**	**66.37**	**41.86**	**25.33**

TABLE 4. Single versus multiple dataset training results. clh_multi indicates that the model is trained on the data-enhanced CTC dataset, lang8 dataset, and HSK dataset, respectively.

Model	Dataset	Detection	Identification	Position	Correction
gactor*	hsk_multi	77.39	48.16	33.98	23.27
	clh_multi	**83.21**	**54.04**	**39.91**	**24.81**
MacBERT	hsk_multi	76.74	47.64	33.12	22.79
	clh_multi	**81.43**	**52.33**	**37.68**	**23.33**
SA	hsk_multi	86.23	60.43	37.60	21.97
	clh_multi	**90.23**	**66.43**	**40.60**	**22.97**
SC	hsk_multi	88.73	57.94	35.71	22.01
	clh_multi	**91.14**	**63.84**	**40.26**	**22.97**
DA	hsk_multi	86.34	61.52	35.87	23.58
	clh_multi	**92.16**	**67.96**	**39.82**	**24.12**
DC	hsk_multi	89.47	59.85	39.62	23.50
	clh_multi	**91.59**	**68.37**	**42.03**	**24.91**
SA(+ILN)	hsk_multi	88.42	62.56	39.44	22.19
	clh_multi	**91.67**	**67.71**	**41.52**	**23.07**
SC(+ILN)	hsk_multi	89.15	58.72	36.97	22.81
	clh_multi	**92.63**	**65.26**	**40.89**	**23.92**
DA(+ILN)	hsk_multi	89.37	65.14	38.75	24.21
	clh_multi	**93.22**	**68.67**	**41.57**	**24.75**
DC(+ILN)	hsk_multi	91.26	66.37	41.86	25.33
	clh_multi	**93.74**	**70.13**	**43.48**	**26.15**

CTC dataset. After adding the residual connection for the same dataset, the four models, SA, SC, DA, and DC, have improved model effects under all evaluation metrics.

- **Effects of Multidataset.** We test the effect of a single dataset versus multiple datasets. As shown in Table 4, the training order of clh (CTC-lang8-HSK) with multiple datasets is helpful in improving the model.



所以我从上个星期到九月份【都】在上海。
So from last week to September was in Shanghai.

为了这次旅游，我们从上个月开始准备买机票【，顶→、订】饭店【，→、】办签证~~【子】~~。
For this trip, we have been preparing to buy air tickets, book hotels and get visas since last month.

这【个→种】食品我不知道中文怎【幺→么】说。
I don't know how to say this kind of food in Chinese.

FIGURE 5. Sample Chinese grammatical error correction cases. Each pair of Chinese and English sentences in the diagram corresponds.

For models SA/SC/DA/DC, they can improve on all four evaluation metrics. We also explore three datasets with different training orders; the result is weaker than clh training.

- **Effects of ILN.** We test the effect of a single dataset versus multiple datasets and an HSK dataset. As shown in Table 3 and 4, after adding the ILN method for the same dataset, the four models, SA, SC, DA, and DC, have improved model effects under all evaluation metrics.

G. CASE STUDY

In Figure 5, we present example error correction cases generated by our proposed model. For texts containing grammatical errors, it is possible to correct multiple and different types of errors in the text and correct errors in the improper use of punctuation. The results show that the model has good error correction ability for word selection, missing, and redundancy errors.

V. CONCLUSION AND FUTURE WORK

We propose a novel dynamic word embedding enhancement with residual connection for Chinese grammatical error correction in this work. The combination of static and dynamic word embedding can effectively capture text features to obtain better text representation. We also propose a data augmentation based on complex tags to improve our model's error correction capability. At the same time, in order to successfully fuse various channels for UDA, we concurrently offer an ILN, a unique module for UDA. Additionally, we demonstrate that with sparsity regularization, some channels' scaling factors will be 0. Simply substituting ILN for the LN layer during the training process will allow ILN to be included in various network backbones. According to empirical research, ILN dramatically improves the models proposed and significantly improves the generalization capability of models. Experiment results show that the proposed method significantly outperforms all robust baseline methods and achieves the best result of prediction-level and correction-level on the CGED-2020 dataset. In the future, we will add Chinese character feature information to the word embedding for further improvement. Since manually correcting text, error correctors consider contextual information and analyze the pronunciation and morphological structure of the wrong words.

REFERENCES

[1] G. Rao, Q. Gong, B. Zhang, and E. Xun, "Overview of NLPTEA-2018 share task Chinese grammatical error diagnosis," in *Proc. 5th Workshop Natural Lang. Process. Techn. Educ. Appl.*, 2018, pp. 42–51.

[2] C. Napoles, K. Sakaguchi, and J. Tetreault, "JFLEG: A fluency corpus and benchmark for grammatical error correction," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 229–234.

[3] D. Micol and C. Quirk, "A large scale ranker-based system for search query spelling correction," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 358–366.

[4] K. Mokhtar, S. S. Bukhari, and A. Dengel, "OCR error correction: State-of-the-art vs an NMT-based approach," in *Proc. 13th IAPR Int. Workshop Document Anal. Syst. (DAS)*, Apr. 2018, pp. 429–434.

[5] Y. Zhao, M. Komachi, and H. Ishikawa, "Improving Chinese grammatical error correction with corpus augmentation and hierarchical phrase-based statistical machine translation," in *Proc. 2nd Workshop Natural Lang. Process. Techn. Educ. Appl.*, 2015, pp. 111–116.

[6] H. Ren, L. Yang, and E. Xun, "A sequence to sequence learning for Chinese grammatical error correction," in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput.* Springer, 2018, pp. 401–410.

[7] J. Zhou, C. Li, H. Liu, Z. Bao, G. Xu, and L. Li, "Chinese grammatical error correction using statistical and neural models," in *Proc. 7th CCF Int. Conf. Natural Lang. Process. Chin. Comput. (NLPCC)*. Hohhot, China: Springer, 2018, pp. 117–128.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[9] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[10] J. Zhang, "Combining GCN and transformer for Chinese grammatical error detection," 2021, *arXiv:2105.09085*.

[11] S. Wang, B. Wang, J. Gong, Z. Wang, X. Hu, X. Duan, Z. Shen, G. Yue, R. Fu, and D. Wu, "Combining ResNet and transformer for Chinese grammatical error diagnosis," in *Proc. 6th Workshop Natural Lang. Process. Techn. Educ. Appl.*, Suzhou, China, 2020, pp. 36–43.

[12] K. Omelianchuk, V. Atrasevych, A. Chernodub, and O. Skurzhanskyi, "GECToR–grammatical error correction: Tag, not rewrite," in *Proc. 15th Workshop Innov. Use NLP Building Educ. Appl.*, 2020, pp. 163–170.

[13] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1–9.

[14] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulo, "AutoDIAL: Automatic domain alignment layers," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5067–5075.

[15] M. Liu, W. Wu, Z. Gu, Z. Yu, F. Qi, and Y. Li, "Deep learning based on batch normalization for P300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, Jan. 2018.

[16] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[17] M. Kaneko, M. Mita, S. Kiyono, J. Suzuki, and K. Inui, "Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4248–4254.

[18] W. Zhao, L. Wang, K. Shen, R. Jia, and J. Liu, "Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data," in *Proc. Conf. North*, 2019, pp. 156–165.

[19] Z. Xie, G. Genthial, S. Xie, A. Ng, and D. Jurafsky, "Noising and denoising natural language: Diverse backtranslation for grammar correction," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 619–628.

[20] W. Zhou, T. Ge, C. Mu, K. Xu, F. Wei, and M. Zhou, "Improving grammatical error correction with machine translation pairs," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2020, pp. 318–328.

[21] E. Malmi, S. Krause, S. Rothe, D. Mirylenka, and A. Severyn, "Encode, tag, realize: high-precision text editing," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 5054–5065.

[22] A. Awasthi, S. Sarawagi, R. Goyal, S. Ghosh, and V. Piratla, "Parallel iterative edit models for local sequence transduction," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 4260–4270.

[23] F. Stahlberg and S. Kumar, "Seq2Edits: Sequence transduction using span-level edit operations," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 5147–5159.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[25] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," 2016, *arXiv:1603.04779*.

[26] W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7354–7362.

[27] X. Wang, Y. Jin, M. Long, J. Wang, and M. I. Jordan, "Transferable normalization: Towards improving transferability of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–11.

[28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[29] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu, "Revisiting pre-trained models for Chinese natural language processing," 2020, *arXiv:2004.13922*.

[30] T. Yue, S. Liu, H. Cai, T. Yang, S. Song, and T. Yu, "Improving Chinese grammatical error detection via data augmentation by conditional error generation," in *Proc. Findings Assoc. Comput. Linguistics*, 2022, pp. 2966–2975.

[31] C. H. Yu and H. H. Chen, "Detecting word ordering errors in Chinese sentences for learning Chinese as a foreign language," in *Proc. COLING*, 2012, pp. 3003–3018.

[32] Y. Zhao, N. Jiang, W. Sun, and X. Wan, "Overview of the NLPCC 2018 shared task: Grammatical error correction," in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput.* Springer, 2018, pp. 439–445.

[33] L. Wang, W. Zhao, R. Jia, S. Li, and J. Liu, "Denoising based sequence-to-sequence pre-training for text generation," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 4003–4015.

[34] J. Lichtarge, C. Alberti, S. Kumar, N. Shazeer, N. Parmar, and S. Tong, "Corpora generation for grammatical error correction," in *Proc. Conf. North*, 2019, pp. 3291–3301.

[35] L. Wang and X. Zheng, "Improving grammatical error correction models with purpose-built adversarial examples," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 2858–2869.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[37] M. Fang, K. Fu, J. Wang, Y. Liu, J. Huang, and Y. Duan, "A hybrid system for NLPTEA-2020 CGED shared task," in *Proc. 6th Workshop Natural Lang. Process. Techn. Educ. Appl.*, 2020, pp. 67–77.

[38] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[39] J. Li, J. Guo, Y. Zhu, X. Sheng, D. Jiang, B. Ren, and L. Xu, "Sequence-to-action: Grammatical error correction with action guided sequence generation," 2022, *arXiv:2205.10884*.

[40] H. Xie, X. Lyu, and X. Chen, "String editing based Chinese grammatical error diagnosis," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 5335–5344.

[41] L. Zheng, Y. Deng, W. Song, L. Xu, and J. Xiao, "An alignment-agnostic model for Chinese text error correction," 2021, *arXiv:2104.07190*.

YIN WANG received the B.A. degree in journalism and communication from Jinan University, in 2019. Her research interests include text recommender systems, graph neural networks, and natural language processing.

ZHENGHAN CHEN received the master's degree from Peking University, Beijing, China, in 2022. He is currently a Kaggle Master and a Staff Member doing research on artificial intelligence recommendation algorithms with Microsoft and Knowmeta. His current research interests include graph representation learning and natural language processing.