System Report for CCL23-Eval Task 7: THU KELab (sz) - Exploring Data Augmentation and Denoising for Chinese Grammatical Error Correction

Jingheng Ye[1], **Yinghui Li**[1], **Hai-Tao Zheng**[1,2] *
[1]Shenzhen International Graduate School, Tsinghua University
[2]Peng Cheng Laboratory
{yejh22,liyinghu20}@mails.tsinghua.edu.cn

Abstract

This paper explains our GEC system submitted by **THU KELab (sz)** in the CCL2023-Eval Task 7 CLTC (Chinese Learner Text Correction) Track 1: Multidimensional Chinese Learner Text Correction. Recent studies have demonstrate GEC performance can be improved by increasing the amount of training data. However, high-quality public GEC data is much less abundant. To address this issue, we propose two data-driven techniques, data augmentation and data denoising, to improve the GEC performance. Data augmentation creates pseudo data to enhance generalization, while data denoising removes noise from the realistic training data. The results on the official evaluation dataset YACLC demonstrate the effectiveness of our approach. Finally, our GEC system ranked second in both close and open tasks. All of our datasets and codes are availabel at https://github.com/THUKElab/CCL2023-CLTC-THU_KELab.

1 Introduction

The CCL2023-CLTC Track 1 (Multidimensional Chinese Learner Text Correction) is a subtask of Grammatical Error Correction (GEC) (Ye et al., 2023), aiming to correct sentences written by Chinese learners through a two-dimensional annotation scheme, namely grammar and fluency (Wang et al., 2021). Adhering to the minimal edits principle, the former ensures that the structure of the original sentence is maintained as much as possible with the smallest number of revisions. Conversely, the latter emphasizes fluency-based correction, where annotators strive to make the sentences more fluent and native-sounding.

Numerous studies (Stahlberg and Kumar, 2021; Kiyono et al., 2020; Koyama et al., 2021b) have shown that the performance of GEC can be improved by increasing the volume of training data. However, obtaining publicly-available and high-quality data for GEC is a challenge (Ma et al., 2022; Ye et al., 2022). Training GEC models with limited data could lead to the fact that GEC models are very likely to overfit and make predictions based on spurious patterns (Tu et al., 2020), owing to the huge gap between the number of model parameters and limited data available for GEC.

This paper attempts to alleviate the aforementioned problem using two techniques, namely data augmentation and data denoising. Thanks for the ease of constructing pseudo grammatical errors, various GEC data augmentation methods have been widely explored, including *noise injection* (Kiyono et al., 2020; Grundkiewicz et al., 2019; Xu et al., 2019), *pattern noise* (Choe et al., 2019), *back-translation* (Sennrich et al., 2016; Xie et al., 2018; Stahlberg and Kumar, 2021) and *round-trip translation* (Zhou et al., 2020). Inspired by the success of GEC data augmentation, we first generate synthetic parallel data from clean monolingual corpora, which is used for pre-training GEC models [0]. Then, we introduce *Cutoff* in the fine-tuning stage to encourage GEC models to make consistent predictions regardless of random noise applied to the sentences (Shen et al., 2020).

Furthermore, we observe that the provided official Lang8 training set contains a significant amount of noise due to low-quality annotation, which could harm the performance of GEC models. To address this

*Corresponding author: Hai-Tao Zheng. (E-mail: zheng.haitao@sz.tsinghua.edu.cn)

[0]We introduce extra corpora only in the open task.

issue, we reconstruct the denoised training set using a well-trained GEC model or ensemble. Specifically, we use a GEC model/ensemble to further correct the target sentences from Lang8, which effectively removes some of the noise present in the data. We then replace the original noisy target sentences with the new corrected target sentences based on the assumption that the outputs of the well-trained model/ensemble can denoise the original dataset caused by low-quality annotation.

We evaluate our data-driven ideas on the official evaluation dataset YACLC using two backbone models: BART (Lewis et al., 2020) and GECToR (Omelianchuk et al., 2020). In the close task, our best single model achieves 71.88 $F_{0.5}$ for minimal correction and 42.02 $F_{0.5}$ for fluent correction (with an average of 56.95 $F_{0.5}$). Our best BART + GECToR ensemble secured the 2nd position in the close task with 74.92 $F_{0.5}$ for minimal correction and 43.89 $F_{0.5}$ for fluent correction (with an average of 59.41 $F_{0.5}$), and also secured the 2nd position in the open task with 76.14 $F_{0.5}$ for minimal correction and 44.17 $F_{0.5}$ for fluent correction (with an average of 60.16 $F_{0.5}$).

In words, the contributions of our paper are three folds:

(1) We showcase the effectiveness of GEC data augmentation methods, including pattern noise (PN), back-translation (BT) and Cutoff.

(2) We observe that the noise present in Lang8 harms the performance of GEC models. By denoising the dataset using a well-trained GEC model/emsemble, we significantly improve the GEC performance.

(3) The evaluation results confirm the effectiveness of our proposed approach. Our system achieves the 2nd place in both close task and open tasks.

2 Background

Generally, GEC models learn the monolingual translation probability $P(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta})$, where $\mathbf{x}$ denotes an ungrammatical source sentence and $\mathbf{y}$ represents a grammatically correct target sentence. Given a parallel training dataset $\mathcal{D}$, the standard training objective is to minimize the empirical risk:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[\mathcal{L}_{\mathrm{CE}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})], \tag{1}$$

where $\mathcal{L}_{\mathrm{CE}}$ denotes the cross entropy loss, $\mathcal{D}$ can either be a realistic dataset $\mathcal{D}_r$ in a standard supervised learning setting or a pseudo dataset $\mathcal{D}_p$ commonly used for GEC data augmentation. In the latter, source sentences are often generated from monolingual corpora as seen in (Kiyono et al., 2020).

Recent studies have attempted to improve the performance of GEC models by incorporating various data augmentation techniques. To this end, we examine and compare the effectiveness of two data augmentation methods that aim to improve generalization through increased training data scale.

Pattern Noise (PN). PN introduces in-distribution grammatical errors to sentences (Choe et al., 2019). Specifically, it first identifies error patterns in GEC datasets using an automated error annotation toolkit (e.g., ERRANT (Bryant et al., 2017)). Then, it applies a noising function to sentences by randomly replacing text segments with pre-extracted grammatical errors.

Backtranslation (BT). BT generates more genuine grammatical errors by learning the distribution of human-written grammatical errors using noisy Seq2Seq models (Kiyono et al., 2020; Koyama et al., 2021a; Xie et al., 2018). The noisy model is trained with the inverse of the GEC parallel dataset, where the ungrammatical sentence is treated as the target and the grammatical sentence as the source. Several variants of BT were proposed by (Xie et al., 2018), and their study revealed that the variant **BT (Noisy)** achieved the best performance. Consequently, we focus on this variant in our work. During decoding of ungrammatical sentences, BT (Noisy) adds $r\beta_{\text{random}}$ to the score of each hypothesis in the beam for each time step, where r is drawn uniformly from the interval $[0, 1]$, and β_{random} is a hyper-parameter that controls the noise degree.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 262-270, Harbin, China, August 3 - 5, 2023.

Original Target	他们有两个孩子，一男一女 They have two children, one boy and one girl	
Denoised Target	他们有两个孩子，一男一女。 They have two children, one boy and one girl.	
Original Target	妈妈在银行工作，她今年自己买了一个公寓房间 My mother works in a bank and she bought an apartment room on her own this year	
Denoised Target	妈妈在银行工作，她今年自己买了一个公寓。 My mother works in a bank and she bought an apartment on her own this year.	
Original Target	我去年十二月开始住在上海。 I have been living in Shanghai December last year.	
Denoised Target	我从去年十二月开始住在上海。 I have been living in Shanghai since December last year.	

Table 1: Examples of denoised samples. We mark the correction part.

3 System Overview

3.1 Denoising Data

As shown in Table 1, we observe significant noise in the training set, most of which is primarily due to under-correction resulting from low-quality annotation. We hypothesize that such noise data is not useful for providing teaching signals to the model, and eventually harms its performance. As a solution, we employ a well-trained GEC model to correct the original target sentences. However, we have observed instances where the GEC model over-corrects the target, which can be problematic. To address this issue, we also explore denoising the dataset using a GEC ensemble.

3.2 Dynamically Noising Data

We introduce *Cutoff* (Shen et al., 2020), a simple yet efficient data augmentation approach that adds dynamic noise during training. The central idea behind Cutoff is to promote consistent predictions across various sentence views, each containing only partial information, to enhance the model's generalization capabilities and reduce prediction errors. Specifically, given a text sequence $\mathbf{x}$, Cutoff constructs augmented samples $\mathbf{x}'$ by randomly removing the information from the input embedding. In our implementation, we randomly convert the input token embeddings of both the encoder and decoder to 0. By imposing constraints on the input views, the learned model is taught to be robust against random noise. The training objective of Cutoff can be described as follow:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(\mathbf{x}, \mathbf{y}) + \alpha \mathcal{L}_{\mathrm{CE}}(\mathbf{x}', \mathbf{y}) + \beta \mathcal{L}_{\mathrm{KL}}(\mathbf{x}, \mathbf{x}', \mathbf{y}), \tag{2}$$

where $\mathbf{y}$ refers to the target sentence, while α and β are weights used to balance the contribution of learning from the original data and augmented data. $\mathcal{L}_{\mathrm{CE}}$ denotes the cross-entropy loss, and $\mathcal{L}_{\mathrm{KL}}$ is the KL divergence, which is defined as:

$$\mathcal{L}_{\mathrm{KL}}(\mathbf{x}, \mathbf{x}', \mathbf{y}) = \mathrm{KL}\left[P(\mathbf{y} \mid \mathbf{x}') \parallel P_{avg}\right], \tag{3}$$

where P_{avg} represents the average prediction probability across realistic and augmented samples. We only apply Cutoff in Seq2Seq models and leave the exploration of its effectiveness for Seq2Edit models to future work.

4 Experiments

4.1 Experimental Settings

We participate in both the close and open tasks of CCL2023-CLTC Track 1. The only distinction between the experimental settings of these tasks lies in their training sets, while we utilize the same GEC backbone models. We introduce additional pseudo and realistic data in the open task, which has been proven effective in improving the $\mathrm{F}_{0.5}$ score.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 262-270, Harbin, China, August 3 - 5, 2023.

GEC backbone model. Inspired by the complementary power in dealing with different error types of the Seq2Seq and Seq2Edit model in the filed of CGCC (Zhang et al., 2022a), we train both models separately. For the Seq2Seq model, we employ Chinese BART [1] as our backbone model (Shao et al., 2021), which has been proven a strong baseline in GEC (Zhang et al., 2022b; Zhang et al., 2022a). We do **not** modify the vocabulary since the updated version of Chinese BART has replaced the old vocabulary with a larger one. We adopt the Dropout-Src mechanism (Junczys-Dowmunt et al., 2018) for source-side word embeddings, following (Zhang et al., 2022b). The training of Seq2Seq models is conducted using the `Fairseq` (Ott et al., 2019) public toolkit. For the Seq2Edit model, we employ the GECToR model (Omelianchuk et al., 2020) initialized with the weights of StructBERT (Wang et al., 2020; Zhang et al., 2022a). We train the Seq2Edit model using the open-source project (Zhang et al., 2022a). The primary hyperparameters for both models are provided in Table 2.

Configuration	**Value**
Pre-training	
Backbone	BART-large
Devices	4 Tesla V100 (80GB)
Epochs	20
Batch size per GPU	4096 tokens
Optimizer	Adam
Learning rate	3×10^{-5}
Warmup updates	2000
Max source length	1024
Dropout	0.2
Dropout-src	0.2
Fine-tuning	
Epoch	3
Cutoff Weights	α=1.0, β=1.0
Learning rate	3×10^{-5}, 2×10^{-5}
Warmup updates	2000
Inference	
Beam size	12

Table 2: Hyperparameter of Seq2Seq.

Configuration	**Value**
Pre-training	
Backbone	StructBERT
Devices	1 Tesla V100 (80GB)
Epochs	10
Batch size per GPU	512 sentences
Optimizer	Adam
Learning rate	1×10^{-5}
Patience	3
Fine-tuning	
Epoch	1
Learning rate	1×10^{-5}, 5×10^{-6}
Inference	
Keep bias	0.05
Iterations	5

Table 3: Hyperparameters of Seq2Edit.

Data Augmentation. We introduce pseudo datasets to pre-train our GEC models. As the close task do not allow additional datasets, we apply PN and BT on the official training set to generate more pseudo data. Finally, we construct a combination of pseudo datasets consisting of 4 and 4 pseudo datasets respectively generated by PN and BT, where a target sentence correspond to 8 pseudo source sentences. We pre-train our Seq2Edit models using these pseudo datasets [2]. For the open task, we generate 8M pseudo data using the seed corpus *news2016zh* [3] with the same target sentences for both PN and BT.

	Dataset	**#Sentences**	**Usage**
Close	**Pseudo Lang8**	9,707,656	Pre-training (Seq2Edit)
	Official Lang8	1,213,457	Fine-tuning I
	YACLC-*dev*	19,195	Fine-tuning II
Open	**News2016zh**	8,000,000	Pre-training
	Lang8+CGED+HSK	1,423,196	Fine-tuning I
	YACLC-*dev*	19,195	Fine-tuning II
Test	**YACLC-*test*-minimal**	7,296	Testing
	YACLC-*test*-fluent	5,515	Testing

Table 4: Statistics of GEC datasets.

Datasets and evaluation. We decompose the fine-tuning of GEC models into two stages following (Huang, 2022). For the close task, we fine-tune the GEC models on 1) the official Lang8 training set, and 2) the YACLC validation set. For the open task, we fine-tune the GEC models on 1) a combination

[1] https://huggingface.co/fnlp/bart-large-chinese
[2] We also attempt to pre-train our Seq2Seq models but it fail to improve the performance.
[3] `https://github.com/brightmart/nlp_chinese_corpus`

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 262-270, Harbin, China, August 3 - 5, 2023.

	System	Backbone	YACLC-*test*-minimal P	R	$F_{0.5}$	YACLC-*test*-fluent P	R	$F_{0.5}$	Average $F_{0.5}$
Close	Huang (2022)	BART-large	76.53	54.61	70.84	48.67	24.26	40.52	55.68
	Our Seq2Seq Baseline	BART-large	75.75	55.63	70.64	47.55	24.88	40.22	55.43
	Seq2Seq (ours)	BART-large	77.01	56.75	**71.88**	49.67	26.00	**42.02**	**56.95**
	Seq2Edit (ours)	StructBERT-large	72.10	52.76	67.17	47.43	25.01	40.22	53.70
	Huang (2022)	N×BART-large	79.95	50.27	71.51	50.69	21.66	39.97	55.74
	Ensemble (ours)	5×Seq2Seq + 4×Seq2Edit	82.25	55.23	**74.92**	53.82	25.24	**43.89**	**59.41**
Open	**Seq2Seq** (ours)	BART-large	79.27	58.45	**74.00**	50.80	26.50	**42.93**	**58.47**
	Seq2Edit (ours)	StructBERT-large	74.11	52.16	68.36	49.48	23.73	40.65	54.50
	Ensemble (ours)	5×Seq2Seq + 4×Seq2Edit	83.58	56.15	76.14	54.50	25.13	44.17	60.16

Table 5: Results on YACLC-*test*.

of Lang8, CGED and HSK[4], and 2) the YACLC validation set. We evaluate the GEC models using the YACLC validation set in the first stage, and then further fine-tune them for several runs. We report the results on the official YACLC test set.

Post-processing. In our pilot experiments, we observe that GEC models tend to make unnecessary edits to numbers and letters, which adversely affected performance. Therefore, we filter out the edits involving numbers and letters, resulting in an improvement of 0.5∼1.0 point in the average $F_{0.5}$ score.

Ensemble. Following previous works (Zhang et al., 2022a; Huang, 2022), we ensemble heterogeneous models by edit-wise majority voting mechanism. Specifically, we first extract edits of system hypotheses using the open-source evaluation tool ChERRANT (Zhang et al., 2022a), and then preserve the edits that appear more than $N/2$ times, where N represents the number of models.

4.2 Main Results

The main results are listed in Table 5. When training only on the official Lang8 dataset, our single Seq2Seq baseline model using cutoff achieves an average of 55.43 $F_{0.5}$ score in both close and open tasks, which is comparable to the previous best result. If data denoising are available, our Seq2Seq model improve the $F_{0.5}$ score by approximately 1.5 points, achieving an average of 56.95 $F_{0.5}$. However, there is a huge gap of performance between the Seq2Edit and Seq2Seq model, possibly because the cutoff technique is not applicable for the Seq2Edit model. Considering the performance gap between the Seq2Seq and Seq2Edit models, we ensemble them with imbalance numbers. Our best ensemble, which is composed of 5×Seq2Seq + 4×Seq2Edit, achieves an average of 59.41 $F_{0.5}$ score in the close task.

For the open task, both models perform better since they are trained using pseudo data generated from additional monolingual corpora and more realistic data. An interesting finding is the improvement of the Seq2Seq model is more significant in comparison to the Seq2Edit model, even though the performance of the former is better. This suggests the enormous potential of Seq2Seq models when massive data is available. Finally, our best ensemble achieves an average of 60.16 $F_{0.5}$ score in the open task.

4.3 Analysis

In this section, we conduct several ablation studies to highlight the contribution of our proposed techniques. We mainly report the performance of our Seq2Seq model since it has been shown that Seq2Seq models outperform Seq2Edit models in Table 5.

Effectiveness of denoising. We explore the effectiveness of denoising the training data using multiple strategies. Considering the strong performance of a single Seq2Seq model, we first denoise the datasets using a single Seq2Seq model. As shown in Table 6, it improves the GEC model by 0.66 $F_{0.5}$ in the close task. However, it does not benefit the GEC model in the open task. We suspect the extra high-quality training data offset the negative effects of the noise in Lang8. Furthermore, we adopt to denoise the datasets with a GEC ensemble, which is composed of 5× Seq2Seq and 5 × Seq2Edit models. We tune the majority voting number M in the close task, where M is the threshold for controlling the edit

[4]We filter out the sentences that already exist in the YACLC dataset.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 262-270, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

	Denoise	YACLC-*test*-minimal P	R	$F_{0.5}$	YACLC-*test*-fluent P	R	$F_{0.5}$	Average $F_{0.5}$
Close	-	75.75	55.63	70.64	47.55	24.88	40.22	55.43
	Seq2Seq	75.35	57.67	71.00	47.78	26.51	41.18	56.09
	5×Seq2Seq + 5×Seq2Edit							
	M=4	75.95	57.93	71.50	48.58	26.76	41.77	56.64
	M=5	75.65	57.96	71.30	48.19	26.79	41.55	56.43
	M=6	77.01	56.75	**71.88**	49.67	26.00	**42.02**	**56.95**
	M=7	76.69	56.82	71.68	49.50	25.95	41.90	56.79
Open	-	79.42	57.05	73.65	50.95	25.16	42.29	57.97
	Seq2Seq	78.78	57.31	73.35	50.88	25.53	42.45	57.90
	5×Seq2Seq + 5×Seq2Edit							
	M=6	79.27	58.45	**74.00**	50.80	26.50	**42.93**	**58.47**

Table 6: Effect of data denoising. We report the performance of Seq2Seq models trained with different datasets.

	Pre-training	YACLC-*test*-minimal P	R	$F_{0.5}$	YACLC-*test*-fluent P	R	$F_{0.5}$	Average $F_{0.5}$
Open	-	**79.67**	54.97	73.10	**52.71**	23.87	42.45	57.78
	PN	79.28	57.09	73.56	52.06	25.69	43.19	58.38
	BT	78.87	**58.52**	**73.74**	51.64	26.18	**43.23**	**58.49**

Table 7: Effect of pre-training using 8M pseudo data. We report the performance of Seq2Seq models in the open task.

	Cutoff Ratio	YACLC-*test*-minimal P	R	$F_{0.5}$	YACLC-*test*-fluent P	R	$F_{0.5}$	Average $F_{0.5}$
Close	0.05	75.75	55.63	**70.64**	47.55	24.88	40.22	55.43
	0.10	75.21	56.70	70.60	48.00	26.07	**41.09**	**55.85**
	0.15	72.63	58.90	69.40	44.95	28.05	40.12	54.76
	0.20	71.84	60.02	69.12	44.55	28.97	40.22	54.67
	0.25	74.11	57.76	70.14	45.85	26.55	40.03	55.09
	0.30	73.36	56.51	69.23	45.86	26.10	39.83	54.53

Table 8: Effect of Cutoff ratios. We report the performance of Seq2Seq models in the close task.

preservation. It is observed that GEC models achieve the peak of average $F_{0.5}$ when $M = 6$. Training with denoised datasets is also helpful in the open task. The results demonstrate the effectiveness of data denoising, particularly for GEC datasets with considerable noise.

Effectiveness of pre-training. Pre-training GEC models using pseudo data has been proven effective in previous works (Kiyono et al., 2020; Stahlberg and Kumar, 2021). We compare data augmentation methods, PN and BT, in terms of constructing pseudo data. We train Seq2Seq models with an additional pre-training stage on 8M pseudo data. The results, reported in Table 7, demonstrate that both methods can significantly improve the Recall and $F_{0.5}$ scores of GEC models, with a slight decrease in Precision.

The effect of Cutoff ratios. One important hyperparameter with the Cutoff approach is the ratio of tokens to be removed. We attempt to investigate how GEC models perform with varying cutoff ratios in {0.05,0.10,0.15,0.20,0.25,0.30}. As shown in Table 8, various cutoff ratios significantly impact the $F_{0.5}$ score, where the model achieves the highest $F_{0.5}$ score at a ratio of 0.10. The decreased performance of a higher cutoff ratio may be attributed to the assumption that more noise could not necessarily lead to better generalization ability.

5 Conclusion

In this CCL2023-CLTC Track 1 Open&Close Task, we improve GEC models by adopting two data-driven techniques, namely data augmentation and data denoising. Our experiments on YACLC evaluation

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 262-270, Harbin, China, August 3 - 5, 2023.

datasets annotated with two principles demonstrate the effectiveness of our proposed methods. Our best ensemble, which is consisting of Seq2Seq and Seq2Edit models, achieves an average $F_{0.5}$ of 59.41 in the close task and 60.16 in the open task, ranking second in both tasks. In the future, we will explore the effectiveness of our approach in other languages and datasets.

Limitations

First, despite the improvement of data denoising, it requires extra computational costs, particularly when denoising large-scale datasets using well-trained ensembles. It is also promising to develop a dynamic denoising strategy during training of GEC models. Secondly, our Seq2Edit models lag far behind our Seq2Seq model, which could lead to a mismatch in ability when used for model ensemble. Given the inference efficiency of Seq2Edit models, extra improvements should have been considered.

Acknowledgements

This research is supported by National Natural Science Foundation of China (Grant No.62276154), Research Center for Computer Network (Shenzhen) Ministry of Education, Beijing Academy of Artificial Intelligence (BAAI), the Natural Science Foundation of Guangdong Province (Grant No. 2023A1515012914), Basic Research Fund of Shenzhen City (Grant No. JCYJ20210324120012033 and JSGG20210802154402007), the Major Key Project of PCL for Experiments and Applications (PCL2021A06), and Overseas Cooperation Research Fund of Tsinghua Shenzhen International Graduate School (HW2021008).

References

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada, July. Association for Computational Linguistics.

Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. 2019. A neural grammatical error correction system built on better pre-training and sequential transfer learning. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, Florence, Italy, August. Association for Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy, August. Association for Computational Linguistics.

Rong Huang. 2022. Ccl2022-cltc track3: Technical reports of kk team. *CCL2022-CLTC Technical Reports*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana, June. Association for Computational Linguistics.

Shun Kiyono, Jun Suzuki, Tomoya Mizumoto, and Kentaro Inui. 2020. Massive exploration of pseudo data for grammatical error correction. *IEEE/ACM transactions on audio, speech, and language processing*, 28:2134–2145.

Aomi Koyama, Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2021a. Comparison of grammatical error correction using back-translation models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 126–135, Online, June. Association for Computational Linguistics.

Shota Koyama, Hiroya Takamura, and Naoaki Okazaki. 2021b. Various errors improve neural grammatical error correction. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 251–261.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 262-270, Harbin, China, August 3 - 5, 2023.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July. Association for Computational Linguistics.

Shirong Ma, Yinghui Li, Rongyi Sun, Qingyu Zhou, Shulin Huang, Ding Zhang, Yangning Li, Ruiyang Liu, Zhongli Li, Yunbo Cao, Haitao Zheng, and Ying Shen. 2022. Linguistic rules-based corpus generation for native chinese grammatical error correction. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 576–589. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online, July. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August. Association for Computational Linguistics.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Hang Yan, Fei Yang, Li Zhe, Hujun Bao, and Xipeng Qiu. 2021. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *arXiv preprint arXiv:2109.05729*.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*.

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47, Online, April. Association for Computational Linguistics.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. Structbert: Incorporating language structures into pre-training for deep language understanding. In *International Conference on Learning Representations*.

Yingying Wang, Cunliang Kong, Liner Yang, Yijun Wang, Xiaorong Lu, Renfen Hu, Shan He, Zhenghao Liu, Yun Chen, Erhong Yang, et al. 2021. Yaclc: A chinese learner corpus with multidimensional annotation. *arXiv preprint arXiv:2112.15043*.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 619–628, New Orleans, Louisiana, June. Association for Computational Linguistics.

Shuyao Xu, Jiehao Zhang, Jin Chen, and Long Qin. 2019. Erroneous data generation for grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158, Florence, Italy, August. Association for Computational Linguistics.

Jingheng Ye, Yinghui Li, Shirong Ma, Rui Xie, Wei Wu, and Hai-Tao Zheng. 2022. Focus is what you need for chinese grammatical error correction. *CoRR*, abs/2210.12692.

Jingheng Ye, Yinghui Li, Qingyu Zhou, Yangning Li, Shirong Ma, Hai-Tao Zheng, and Ying Shen. 2023. CLEME: debiasing multi-reference evaluation for grammatical error correction. *CoRR*, abs/2305.10819.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022a. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3118–3130, Seattle, United States, July. Association for Computational Linguistics.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022b. SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC-oriented parser. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2518–2531, Abu Dhabi, United Arab Emirates, December. Association for Computational Linguistics.

Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. 2020. Improving grammatical error correction with machine translation pairs. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 318–328, Online, November. Association for Computational Linguistics.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 262-270, Harbin, China, August 3 - 5, 2023.