Are Pre-trained Language Models Useful for Model Ensemble in Chinese Grammatical Error Correction?

Chenming Tang **Xiuyu Wu** **Yunfang Wu***
National Key Laboratory for Multimedia Information Processing, Peking University
MOE Key Laboratory of Computational Linguistics, Peking University
School of Computer Science, Peking University
tangchenming@stu.pku.edu.cn
{xiuyu_wu, wuyf}@pku.edu.cn

Abstract

Model ensemble has been in widespread use for Grammatical Error Correction (GEC), boosting model performance. We hypothesize that model ensemble based on the perplexity (PPL) computed by pre-trained language models (PLMs) should benefit the GEC system. To this end, we explore several ensemble strategies based on strong PLMs with four sophisticated single models. However, the performance does not improve but even gets worse after the PLM-based ensemble. This surprising result sets us doing a detailed analysis on the data and coming up with some insights on GEC. The human references of correct sentences is far from sufficient in the test data, and the gap between a correct sentence and an idiomatic one is worth our attention. Moreover, the PLM-based ensemble strategies provide an effective way to extend and improve GEC benchmark data. Our source code is available at https://github.com/JamyDon/PLM-based-CGEC-Model-Ensemble.

1 Introduction

Grammatical Error Correction (GEC) is the task of automatically detecting and correcting errors in text (Bryant et al., 2022). Nowadays, there are two mainstream GEC approaches. The first is treating GEC as a low-resource machine translation task (Yuan and Briscoe, 2016), where sequence-to-sequence models like BART (Lewis et al., 2020) are used. This approach simply inputs the incorrect text to the encoder and gets the corrected result from the decoder. The second is treating GEC as a sequence tagging task, where the incorrect text is still taken as the input, but the output is edit tags (keep, delete, add, replace, etc.) for each token. After applying all the edits to the input text, the corrected result is then generated. The model used in this approach is also known as sequence-to-edit models and GECToR (Omelianchuk et al., 2020) is a typical one.

However, most researches on GEC focus on English while Chinese GEC (CGEC) has just started up. The Chinese language is different from English in many ways and its GEC is thus much harder. Instead of word inflection in many Western languages, the Chinese grammar is expressed by function words and word order, making CGEC more difficult and complex for that we can't take word form as a handle. In addition, unlike English, we have very few datasets for training and testing CGEC, which sets us exploring training-free methods like model ensemble to further improve the performance of CGEC systems.

Because of the nature of GEC that corrections can be represented as several independent edits, model ensemble has been a popular way to improve GEC systems. In CGEC, Li et al. (2018), Liang et al. (2020) and Zhang et al. (2022) ensemble their models by majority voting on edits and achieve considerable improvement. Besides, Xie et al. (2016) adopt language models to improve neural language correction, following whom Junczys-Dowmunt et al. (2018) ensemble their GEC models using a language model probability. Today, transformer-based (Vaswani et al., 2017) Pre-trained Language Models (PLMs) have been in predominant use in NLP. However, we find few works on model ensemble using PLMs in CGEC.

In this work, we hypothesize that choosing the best ensemble output with the help of perplexity (PPL) computed by PLMs should boost the final performance of CGEC. We experiment on ensemble of four CGEC models, including two sequence-to-sequence ones and two sequence-to-edit ones. We try four ensemble strategies: traditional voting, sentence-level ensemble, edit-level ensemble, and edit-combination ensemble, the last three exploiting the power of PLMs.

To our surprise, the results of model ensemble

* Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 893–901

with PLMs do not exceed those of traditional voting and are even worse than most of the single models. To find out why a low PPL cannot lead to a better GEC performance, we carry out a detailed analysis on the ensemble results and get some insights on GEC:

1) In the test data, human references are insufficient, while PLM-based ensemble strategies produce valuable candidates, after being human checked, which may be considered as necessary complement to human references.

2) When facing an erroneous sentence, a human expert corrects it with the minimal effort, while PLM-based ensemble strategies generate more natural and idiomatic text, which is of great help for oversea language learners.

3) With the powerful ability, PLM-based models try to generate fluent sentences but sometimes ignore the original meaning of the source sentence, resulting in over-correction that should be addressed in future work.

2 Basic Models

2.1 Single CGEC Models

We implement four single models as baselines, with two seq2seq models and two seq2edit ones. All the models use the Lang-8 [1] dataset for training.

Sequence to Sequence Models. The two seq2seq models are both based on BART-base-Chinese (Shao et al., 2021), and are implemented using fairseq [2](Ott et al., 2019). Besides Lang-8, the HSK data [3] is also used for training. One seq2seq model adopts the "dropout-src" strategy, where each token in input sentences is replaced with "[PAD]" with a probability of 10%. The other one is pre-trained on the synthetic data constrcted on THUCNews [4] (Sun et al., 2016) before the normal training.

Sequence to Edit Models. We apply GECToR-Chinese [5] (Zhang et al., 2022) as our seq2edit models, with the pre-trained Structbert-large-Chinese [6] (Wang et al., 2019) as backbone. Our two seq2edit models only differ in random seeds.

2.2 Pre-trained Language Models

We adopt three PLMs to carry out model ensemble.

BERT-base-Chinese [7]**.** It is pre-trained on two tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). In MLM, each token has a chance of 15% to be replaced with a "[MASK]" (80%), a random word (10%), or itself (10%). Please refer to Devlin et al. (2019) for details.

MacBERT-base-Chinese [8]**.** It is similar to BERT, but employs whole word masking, N-gram masking and similar word replacing in MLM. Besides, Sentence-Order Prediction (SOP) is exploited instead of NSP. Please refer to Cui et al. (2020) for details.

GPT2-Chinese [9]**.** It is an unofficial Chinese version of GPT-2 (Radford et al., 2019). It employs generative pre-training, by predicting the next word in a sentence with only previous words provided.

3 Ensemble Strategy

With the source sentence and the outputs of four single models as the input, we present four ensemble strategies. The diagram of our PLM-based ensamble strategies is shown in Figure 1.

3.1 Traditional Voting

Different models vote for the final results. For each sentence, we consider edit operations suggested by no less than T models as the correct one. In our work, we experiment on T from 2 to 4. We implement the original code provided by Zhang et al. (2022) to carry out this voting strategy.

3.2 Sentence-level Ensemble

Using different PLMs, we compute the perplexities (PPLs) of the source sentence and the outputs of four single models. Specifically, given a sentence $S = (w_1, w_2, ..., w_n)$ and the probability of the word w_i computed by a PLM denoted as p_i, then $PPL = (\prod_{i=1}^{n} \frac{1}{p_i})^{1/n}$. The sentence with the lowest PPL is chosen to be the final output.

3.3 Edit-level Ensemble

Given a source sentence S, all the edits suggested by single models constitute a candidate set A, and the number of edit spans is denoted as m. An edit span means the start-end pair of an edit's position in the sentence. The set of all the edits (from different single models) on the i-th edit span (including

[1] http://tcci.ccf.org.cn/conference/2018/taskdata.php
[2] https://github.com/facebookresearch/fairseq
[3] https://github.com/shibing624/pycorrector
[4] http://thuctc.thunlp.org
[5] https://github.com/HillZhang1999/MuCGEC
[6] https://huggingface.co/bayartsogt/structbert-large
[7] https://huggingface.co/bert-base-chinese
[8] https://huggingface.co/hfl/chinese-macbert-base
[9] https://github.com/Morizeyao/GPT2-Chinese

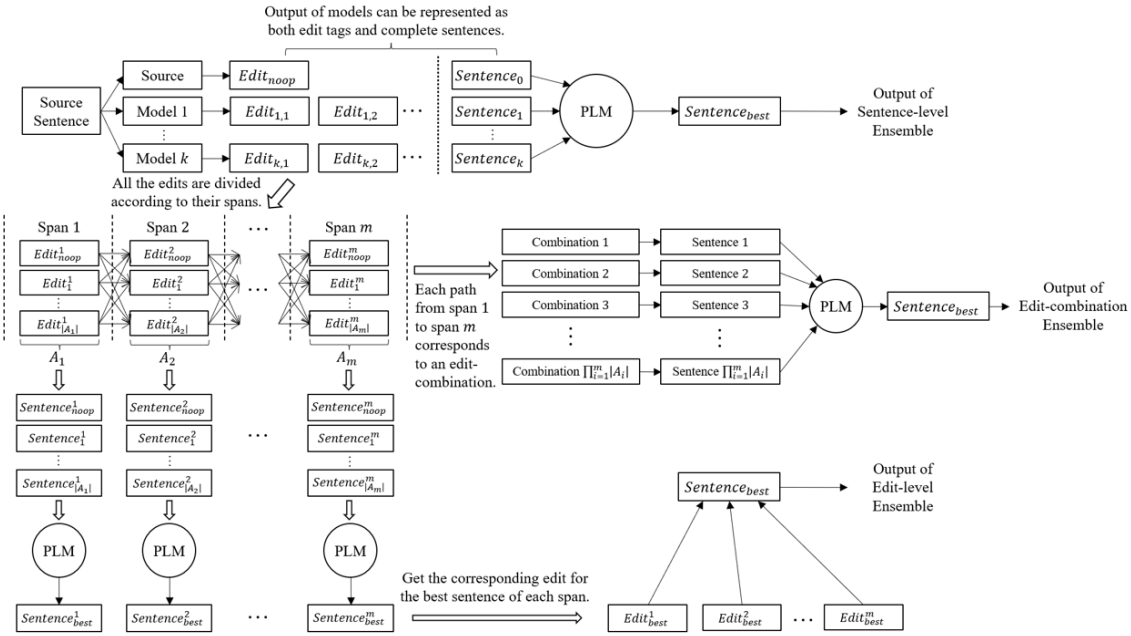

Figure 1: Diagram of our PLM-based ensemble strategies.

"noop") is denoted as A_i. Thus, we can divide $A = \bigcup_{i=1}^{m} A_i$, where $A_i = \{e_j^i \mid j = 1, 2, ..., |A_i|\}$, and e_j^i means the j-th edit on the i-th edit span.

For each edit span (A_i in A), we generate $|A_i|$ new sentences, each corresponding to a single edit in A_i. Then we consult PLMs about PPLs of these new sentences and accept the edit corresponding to the sentence with the lowest PPL, which we mark as e_{best}^i. In other words, e_{best}^i is the best edit (decided by PLMs) in A_i, or on span i.

With each span's best edit, the final edit set E_{final} combines these best edits, described as:

$$E_{final} = \{e_{best}^i \mid i \in \{1, 2, ..., m\}\}, \quad (1)$$

The final hypothesis sentence is then produced on the basis of E_{final}.

3.4 Edit-combination Ensemble

One source sentence may contain more than one errors. For each sentence, this strategy applies all edit combinations to the source sentence and generates many new sentences.

To be specific, given a source sentence S, the edit candidates A are still divided as $A = \bigcup_{i=1}^{m} A_i$, and then we get all possible edit-combinations by:

$$U = \{\{e_{j_1}^1, e_{j_2}^2, ..., e_{j_m}^m\} \mid j_i \in \{1, 2, ..., |A_i|\}\}. \quad (2)$$

Thus we generate $(\prod_{i=1}^{m} |A_i|)$ new sentences, each corresponding to an edit-combination in U. The sentence with the lowest PPL will be accepted as the final output.

Taking the computational complexity into consideration, we only apply this strategy on sentences whose number of edit-combinations is no more than 300. Such simple sentences make up 95.15% of MuCGEC-test and 98.90% of NLPCC-test. We do nothing to the left not-so-simple sentences.

4 Experiments

4.1 Dataset and Evaluation Metrics

We carry out experiments on MuCGEC test data (Zhang et al., 2022) and NLPCC test data (Zhao et al., 2018). MuCGEC contains 7063 sentences and each have at most three references, but is not available at present. NLPCC contains 2000 sentences, each with one or two references, and about 1.1 references on average. We carry out analysis on NLPCC test data.

On MuCGEC, we submit the results of our systems to the public evaluation website [10]. On NLPCC, we implement the tools provided by Zhang et al. (2022) to compute the P (Precision), R (Recall), and $F_{0.5}$ of the output on char-level. Also, we report word-level results on NLPCC-test for reference with previous works.

[10] https://tianchi.aliyun.com/dataset/131328

Strategy	MuCGEC-test			NLPCC-test			NLPCC-test (word-level)		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Single Models									
seq2seq-1	**55.00**	28.32	**46.28**	**43.93**	28.21	**39.52**	**46.17**	29.51	**41.48**
seq2seq-2	50.62	30.40	44.68	40.79	**29.59**	37.92	43.40	31.29	40.28
seq2edit-1	45.80	28.41	40.81	38.42	26.79	35.35	43.08	30.05	39.64
seq2edit-2	45.45	**30.45**	41.37	36.19	28.15	34.24	41.41	**31.58**	38.98
Average of 4	*49.22*	*29.40*	*43.29*	*39.83*	*28.19*	*36.76*	*43.52*	*30.61*	*40.10*
Traditional Voting									
$T = 2$	52.58	**33.61**	47.25	42.71	**32.62**	40.22	45.58	**34.66**	42.88
$T = 3$	69.10	21.68	**48.07**	60.81	21.00	**44.09**	58.39	21.55	**43.52**
$T = 4$	**76.13**	15.35	42.48	**67.33**	14.96	39.61	**64.51**	15.35	39.32
Sentence-level									
BERT-base-Chinese	**48.56**	24.33	40.50	37.71	22.80	33.35	41.38	24.55	36.39
MacBERT-base-Chinese	46.83	33.35	43.33	37.62	31.30	36.16	**42.24**	34.15	40.33
GPT2-Chinese	47.36	**35.01**	**44.24**	**37.75**	**33.20**	**36.74**	41.94	**36.13**	**40.63**
Edit-level									
BERT-base-Chinese	41.31	21.79	35.04	33.19	20.59	29.57	36.69	23.24	32.89
MacBERT-base-Chinese	43.40	29.19	39.55	**35.38**	28.42	33.73	**40.07**	32.87	38.39
GPT2-Chinese	**43.93**	**33.36**	**41.31**	35.04	**31.60**	**34.29**	39.44	**36.07**	**38.71**
Edit-combination									
BERT-base-Chinese	42.90	20.18	35.01	34.25	21.56	30.64	37.56	23.94	33.72
MacBERT-base-Chinese	45.18	28.73	40.54	**36.35**	30.69	35.05	40.11	33.62	38.62
GPT2-Chinese	**46.07**	**31.92**	**42.32**	36.23	**33.29**	**35.60**	**40.50**	**36.44**	**39.62**

Table 1: Experimental results on MuCGEC-test and NLPCC-test. The relatively best results in a group are reported in **bold**, and the best results of all are listed in **underlined bold**.

4.2 Results

Table 1 shows the experimental results. The traditional voting strategy achieves the best performance, with a 44.09 $F_{0.5}$ score on char level that is significantly higher than the best single model. With the threshold T increasing, the precision rises while the recall drops. When $T = 3$, $F_{0.5}$ score reaches the peak, in line with the finding of Tarnavskyi et al. (2022).

However, the PLM-based ensemble strategies get much worse performance than the simple voting strategy, and are even lower than most of single models. In terms of precision and recall, traditional voting achieves higher precision but lower recall than single models while PLM-based strategies are on the contrary. Among three ensemble strategies, the sentence-level one performs best.

Among different PLMs, GPT2-Chinese achieves the best results in all three ensemble strategies. This may be because BERT-based models are naturally good at mask prediction rather than computing PPLs for whole sentences. Later, we base GPT2-Chinese to make further analysis.

5 Analysis and Discussion

We design three ensemble strategies to choose the sequence with the lowest PPL as the final output, but why does $F_{0.5}$ score drop? In our work, all single models are made up of their own PLMs, which means ensembling them exploiting another PLM is just like using PLMs to judge PLMs, so the performance may benefit little. This is in line with the work of Junczys-Dowmunt et al. (2018), where pre-trained single models gain little and even have worse performance after PLM-based ensemble while other simple single models benefit a lot. Besides this, are there any other reasons?

5.1 Statistical Results

In order to find out the cause of the poor performance of PLM-based ensemble strategies, on NLPCC test data, we randomly select 200 samples from the results of all the three strategies along with the best single model (seq2seq-1) for comparison, and ask two graduate students to analyze the output sentences with a double-blind manner. After that, a third expert arbitrates for the inconsistency. Instructions for human annotators are shown

in Appendix A.

According to human judgement, four types are summarized. **Exact** (**E**): the output is fluent and correct, in line with the reference. **Good** (**G**): the output is fluent and correct but different with the reference, which indicates that the references are not sufficient enough. **Over-corrected** (**O**): the output is fluent but doesn't meet the original meaning of the source sentence. **Wrong** (**W**): the output has other problems that we don't care in this work.

The result of human annotation is reported in Table 2, and some examples of **G** and **O** are shown in Table 3.

	E	G	O	W
seq2seq-1 (best single)	38	42	9	111
Sentence-level	36	53	23	88
Edit-level	32	45	20	103
Edit-combination	32	59	21	88

Table 2: Human annotation of generated outputs.

G	src: 我的家附近有很多考式补习班。 out: 我家附近有很多考试补习班。 ref: 我的家附近有很多考试补习班。 There are many cram schools near my home.
G	src: 我低幼儿童的时候很想养狗。 out: 我小时候很想养狗。 ref: 我小的时候很想养狗。 I really wanted a dog when I was young.
G	src: 可它的表情是从来没看过的。 out: 可它的表情是我从来没见过的。 ref: 可它的表情是我从来没看过的。 But it has a look I have never seen before.
O	src: 我班里有很漂亮的女同学，我一见钟情。 out: 我班里有个很漂亮的女同学， 她对我一见钟情。 There was a beautiful girl in my class. She fell in love with me at first sight. ref: 我班里有位很漂亮的女同学， 我对她一见钟情。 There was a beautiful girl in my class. I fell in love with her at first sight.

Table 3: Three examples for **G** and one for **O**. Label "src", "out" and "ref" means the source sentence, the output of one of our PLM-based ensemble strategies and the reference, respectively.

5.2 Discussion

The insufficiency of GEC references. In the outputs of PLM-based ensemble strategies, about 1/4 ("**G**") are automatically judged to be wrong according to the golden references, but indeed correct after human check. Actually, if we assume class **G** is also correct, the number of sentences corrected by PLM-based ensemble strategies (except edit-level ensemble) exceeds that by seq2seq-1, the best single model.

This indicates that GEC references are not sufficient enough, even though datasets like NLPCC provide multi-references. Since artificially generating a correct sentence is much harder than judging a machine-generated sequence correct or not, continuously adding human checked results of PLM-ensemble systems to the references may be a good solution to improve the quality and diversity of the GEC test data.

The goal of GEC. This is a significant issue. Is it enough to just get a sentence rid of errors? Taking coding into example, can we say a piece of code "good" when all the "errors" are clear but pages of "warnings" are flashing? In "**Good**" samples, we compare the human references and automatically generated sentences, and find many of references are only **correct** but not so **idiomatic**. On the other hand, many output sentences of PLM-based ensemble strategies are more natural and like native speakers. If a GEC system is aimed at helping overseas students with their language learning, for example, then idiomaticity should be taken into consideration.

The over-correction of PLM-based models. About 1/10 of sentences generated in PLM-based ensemble ("**O**") are over-corrected, i.e., the model corrects a correct token and thus produces a wrong sentence. PLMs always choose the most fluent sentence with the lowest PPL, sometimes ignoring the original meaning of the source sentence. The over-correction of PLM-based generative models should be addressed in future work.

6 Conclusion

This paper introduces novel ensemble strategies for the GEC task by leveraging the power of pre-trained language models (PLMs). We compare different strategies of model ensemble in CGEC. Surprisingly, PLM-based ensemble strategies do not benefit the system. This suggests that PPL and $F_{0.5}$ have diverging goals. According to our analysis, the insufficiency of references in GEC remains a major problem, which should be continuously improved in future work.

Acknowledgement

This work is supported by the National Hi-Tech RD Program of China (No.2020AAA0106600),

the National Natural Science Foundation of China (62076008) and the Key Project of Natural Science Foundation of China (61936012).

Limitations

First, we don't use any single models without PLMs in their structures to carry out comparative experiments, even though few advanced models nowadays can get rid of PLMs. Second, because of the wrapping of fairseq, we don't have access to all the output probabilities of the single models and thus cannot apply the strategy of using the weighted sum of single models and PLMs used in Junczys-Dowmunt et al. (2018). Third, while BERT-based PLMs are good at mask prediction, we haven't found a strategy to make use of that capacity without being embarrassed by conditional probability. Fourth, we carry out our experiments only on Chinese.

Ethics Statement

About Scientific Artifacts. Since we focus on CGEC, all the code and tools are for the Chinese language and all data is in Chinese. All the scientific artifacts are used for GEC only. The artifacts provided by Zhang et al. (2022) are publicly available based on the Apache-2.0 license, on which we base our own codes and models.

About Computational Budget. We run all the experiments of model ensemble on an Intel® Xeon® Gold 5218 CPU. Processing times are shown in table 4.

Strategy	MuCGEC-test	NLPCC-test
Traditional Voting	1~2s	<1s
Sentence-level	25min	6min
Edit-level	56min	12min
Edit-combination	2.5h	25min

Table 4: Processing times of different ensemble strategies.

About Reproducibility. All the experiments of model ensemble is completely reproducible when the PLMs are frozen (i.e., no matter how many times we run the experiments, the results are just the same).

About Human Annotators. Each of the annotators is paid $20 per hour, above the legal minimum wage. The instructions are shown in Appendix A.

References

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2022. Grammatical error correction: A survey of the state of the art. *arXiv preprint arXiv:2211.05166*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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. Association for Computational Linguistics.

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.

Chen Li, Junpei Zhou, Zuyi Bao, Hengyou Liu, Guangwei Xu, and Linlin Li. 2018. A hybrid system for Chinese grammatical error diagnosis and correction. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 60–69, Melbourne, Australia. Association for Computational Linguistics.

Deng Liang, Chen Zheng, Lei Guo, Xin Cui, Xiuzhang Xiong, Hengqiao Rong, and Jinpeng Dong. 2020. BERT enhanced neural machine translation and sequence tagging model for Chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 57–66, Suzhou, China. 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.

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 NAACL-HLT 2019: Demonstrations*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, 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*.

Maosong Sun, Jingyang Li, Zhipeng Guo, Zhao Yu, Y Zheng, X Si, and Z Liu. 2016. Thuctc: an efficient chinese text classifier. *GitHub Repository*.

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. *arXiv preprint arXiv:2203.13064*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. 2016. Neural language correction with character-based attention. *CoRR*, abs/1603.09727.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. 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. Association for Computational Linguistics.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 439–445. Springer.

A Instructions for Human Annotation

The instructions for human annotators mentioned in Section 5 are as follows:

1. You can see the data in "sample_200.txt", which contains results of 200 sentences.

2. Each sample contains several lines, including "Input" (the source sentence), "seq2seq-1", "Sentence-level", "Edit-level", "Edit-combination", and one or two "Reference" lines.

3. You need to annotate the "seq2seq-1", "Sentence-level", "Edit-level" and "Edit-combination" lines according to the input and reference(s).

4. To be specific, you should choose from the following four types. Exact (E): the output is fluent and correct, in line with the reference. Good (G): the output is fluent and correct but different with the reference, which indicates that the references are not sufficient enough. Over-corrected (O): the output is fluent but doesn't meet the original meaning of the source sentence. Wrong (W): the output has other problems that we don't care in this work.

5. Thank you for your contributions!

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4.1

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Ethics Statement

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Ethics Statement

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Ethics Statement

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Not applicable. Left blank.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Ethics Statement

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 5

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Appendix B

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Ethics Statement

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.