Towards Better Utilization of Multi-Reference Training Data for Chinese Grammatical Error Correction

Yumeng Liu[1], **Zhenghua Li**[1*], **Haochen Jiang**[1], **Bo Zhang**[2], **Chen Li**[2], **Ji Zhang**[2]
[1] School of Computer Science and Technology, Soochow University, China
{ymliu14,hcj22}@stu.suda.edu.cn, zhli13@suda.edu.cn
[2] Alibaba Group, China
{klayzhang.zb,puji.lc,zj122146}@alibaba-inc.com

Abstract

For the grammatical error correction (GEC) task, there usually exist multiple correction ways for an erroneous input sentence, leading to multiple references. Observing the high proportion of multi-reference instances in Chinese GEC training data, we target a systematic study on how to better utilize multi-reference training data. We propose two new approaches and a simple two-stage training strategy. We compare them against previously proposed approaches, on two Chinese training datasets, i.e., Lang-8 for second language learner texts and FCGEC-Train for native speaker texts, and three test datasets. The experiments and analyses demonstrate the effectiveness of our proposed approaches and reveal interesting insights. Our code is available at https://github.com/ymliucs/MrGEC.

1 Introduction

Given a potentially erroneous sentence, the goal of the grammatical error correction (GEC) task is to generate an error-free sentence of the same meaning (Wang et al., 2021b; Bryant et al., 2023). Table 1 gives an example. There are two references, corresponding to two ways of correcting the sentence.

Similar to the machine translation (MT) task, GEC is a typical task for which multiple reference phenomena are ubiquitous, due to the great inherent flexibility of natural languages. For the sake of reliable evaluation, the test set is usually of multi-reference in both GEC (Ng et al., 2014; Bryant et al., 2019; Zhang et al., 2022a; Xu et al., 2022; Zhang et al., 2023) and MT research.

In contrast, in terms of training data, GEC differs from MT in that a large proportion of GEC training data is multi-reference. In the most widely used training data for second-language learner (L2) texts, i.e., Chinese Lang-8, about 48% sentences (274K among 566K) have multiple references.[1] As a recently constructed dataset for native speaker (NS) texts, FCGEC-Train contains about 23% (8K among 36.2K) multi-reference instances.

Input	小明的学习成绩是班级中最好的同学。 Xiao Ming's academic performance is the best classmate in the class.
Ref. 1	小明的学习成绩是班级中最好的~~同学~~。 Xiao Ming's academic performance is the best in the class.
Ref. 2	小明~~的学习成绩~~是班级中学习成绩最好的同学。 Xiao Ming is the classmate with the best academic performance in the class.

Table 1: A multi-reference example from NaSGEC (Zhang et al., 2023).

Despite the ubiquitousness of multi-reference training instances, there lacks a systematic investigation of the question, i.e., how to better utilize multi-reference data for training GEC models. The most common way is to treat a multi-reference instance as multiple independent single-reference instances. Ye et al. (2022) propose to statically determine a single reference from multiple ones before training, and discard other references during training. They compare several ways for reference selection.

This work aims to conduct a comprehensive investigation of this issue. We compare several approaches, including two newly proposed ones, i.e., 1) averaging the loss of multiple references at the same training step, and 2) dynamically determining and using the minimum loss among multiple references. We also propose a two-stage training strategy. We conduct experiments on two Chinese training datasets, i.e., Lang-8 for L2 texts and FCGEC-Train for NS texts, and three test datasets. The results and analyses reveal interesting insights.

[1] English Lang-8 data contains a tiny proportion of multi-reference instances, which is about 7% (67K among 943K) after merging duplicate input sentences (Mizumoto et al., 2012; Tajiri et al., 2012). Besides Lang-8, all other manually constructed training datasets for English are single-reference (Bryant et al., 2023). Partially due to the lack of multi-reference training data, our experiments avoid English but focus on Chinese.

*Corresponding Author.

Findings of the Association for Computational Linguistics ACL 2024, pages 3044–3052

2 The Basic Seq2Seq GEC Model

Given an input sentence $\boldsymbol{x} = x_1 \cdots x_n$, a GEC model aims to output another sentence $\boldsymbol{y} = y_1 \cdots y_m$ that has the same meaning but is free of grammatical errors. In this work, we adopt the sequence-to-sequence (Seq2Seq) model, since it is consistently superior in performance and more flexible in handling complex errors, compared with the other mainstream model, i.e., sequence-to-edit (Seq2Edit) (Zhang et al., 2023; Bout et al., 2023).

The Seq2Seq GEC model adopts the standard encoder-decoder framework of Transformer (Vaswani et al., 2017). Following (Zhang et al., 2023), we employ the pre-trained BART (Lewis et al., 2020) as the model backbone.

Inference. At the τ-th timestamp, based on the encoded information of $\boldsymbol{x}$ and partially generated tokens $\boldsymbol{y}_{<\tau}$, the model generates the probability distribution in an auto-aggressive manner for the next position.

$$\boldsymbol{p}_\tau(t) = \text{Transformer}(\boldsymbol{x}, \boldsymbol{y}_{<\tau}). \quad (1)$$

where t refers to a token in the output vocabulary.

Training. Given a single-reference training dataset, $\mathcal{D} = \{(\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(j)})\}_{j=1}^{N}$, the model applies local cross-entropy loss to maximize $\boldsymbol{p}_\tau(y_\tau)$ in the teacher forcing manner.

3 Two Baseline Approaches

Multi-reference training data. As discussed in Section 1, for GEC it is ubiquitous that an erroneous sentence may have multiple ways for correction, leading to multiple references, especially when the sentence contains more complex errors (vs. spelling errors). We denoted a multi-reference data as $\mathcal{M} = \{(\boldsymbol{x}^{(j)}, \boldsymbol{Y}^{(j)})\}_{j=1}^{N}$, where $\boldsymbol{Y}^{(j)} = \{\boldsymbol{y}_1^{(j)}, \ldots, \boldsymbol{y}_{k_j}^{(j)}\}$ represents a set of reference sentences for $\boldsymbol{x}^{(j)}$.

Undoubtedly, existing GEC models cannot directly utilize $\mathcal{M}$ as training data. Previous works propose two approaches to handle this issue.

The concatenation (CAT) approach. The CAT approach splits a multi-reference training instance into several *independent* single-reference instances, each corresponding to an input/reference pair. Formally, $(\boldsymbol{x}^{(j)}, \boldsymbol{Y}^{(j)})$ becomes $\{(\boldsymbol{x}^{(j)}, \boldsymbol{y}_1^{(j)}), \ldots, (\boldsymbol{x}^{(j)}, \boldsymbol{y}_{k_j}^{(j)})\}$. During training, instances for the same $\boldsymbol{x}^{(j)}$ may appear in different batches due to shuffling.

The minimum Levenshtein distance (MLD) approach. Intuitively, the CAT approach may make the training unstable, as the model swings among multiple references.

To mitigate the uncertainty, (Ye et al., 2022) proposes to select a single reference from multiple ones before training. After comparing several strategies, they find that it is best to select the reference that is closest to the input according to Levenshtein distance (Levenshtein, 1965), as follows.

$$\text{Closeness}(\boldsymbol{x}, \boldsymbol{y}) = \frac{-LD(\boldsymbol{x}, \boldsymbol{y})}{n+m}. \quad (2)$$

where $LD(\cdot)$ is the Levenshtein distance,[2] and $n = |\boldsymbol{x}|$, $m = |\boldsymbol{y}|$ is used for eliminating the sentence length factor.

4 Our Approaches

In this section, we propose two approaches for better utilization of multi-reference training data. The basic idea is to treat a multi-reference instance as a whole, and evaluate each reference after calculating losses of all references. Moreover, we propose a two-stage training strategy to combine the power of two approaches.

4.1 Average Training Loss (AvgL)

We use the average loss of all references as follows. Multiple references function as a single reference.

$$Loss(\boldsymbol{x}^{(j)}, \boldsymbol{Y}^{(j)}) = \frac{1}{k_j} \sum_r Loss(\boldsymbol{x}^{(j)}, \boldsymbol{y}_r^{(j)}). \quad (3)$$

Compared with CAT, AvgL allows the model to consider all references at the same time and to update parameters in a compromised fashion, which, intuitively, can alleviate the training instability issue to some extent.

4.2 Minimum Training Loss (MinL)

We use the minimum loss of all references as follows.

$$Loss(\boldsymbol{x}^{(j)}, \boldsymbol{Y}^{(j)}) = \min_r Loss(\boldsymbol{x}^{(j)}, \boldsymbol{y}_r^{(j)}). \quad (4)$$

There are at least two implications for a reference to have the minimum loss. First, the model favors it among all references and will probably output it during inference. Second, minimum loss means the least parameter update. Therefore, it

[2]The costs of the three edit operations are 1/1/1 (delete/insert/substitute), respectively.

Dataset	#Inputs	Avg. Refs	Usage
Lang-8	566,482	1.8	Training
MuCGEC-Dev	1,137	2.2	Validation
MuCGEC-Test	6,000	2.2	Testing

Table 2: Statistics of Chinese GEC datasets from L2.

is reasonable to further encourage the model to converge toward the single reference, instead of swinging among several references.

MLD can be understood as a static selection strategy, in the sense one reference is selected and used all the time during training. In contrast, MinL dynamically selects one reference according to present losses, and different references may be selected at different training steps.

4.3 Two-stage Training (AvgL + MinL)

Our preliminary experiments show that under MinL, the model would often stick to a certain reference once the reference has the minimum loss when the instance is met for the first time. This is contrary to our expected dynamic selection process. We expect that all references can contribute to model training, and one reference wins gradually and at later stages.

Therefore, we propose a two-stage training strategy: 1) at the first S training steps, the model adopts AvgL; 2) afterward, the model adopts MinL.

5 Experiments

We conduct experiments on two types of datasets: L2 and NS.

L2 Data. For the L2 experiment, we utilize Lang-8 (Zhao et al., 2018) as the training dataset,[3] and MuCGEC (Zhang et al., 2022a) as both the validation and test dataset.

Following previous practice and based on our observation, we preprocess the L2 datasets to improve the performance of the baseline model. Details are given in Appendix A.1. Table 2 shows the statistics for the L2 datasets.

NS Data. For the NS experiment, we utilize FCGEC (Xu et al., 2022) as the training, validation, and test dataset. Meanwhile, NaSGEC-Exam (Zhang et al., 2023) serves as an additional test set in the NS experiment.

[3]The HSK(Cui and Zhang, 2011; Zhang and Cui, 2013) is excluded since most of the instances in it are single-reference.

Dataset	#Inputs	Avg. Refs	Usage
Original			
FCGEC-Train	36,329	1.2	Training
FCGEC-Dev	2,000	1.3	Validation
FCGEC-Test	3,000	-	Testing
NaSGEC-Exam	7,000	1.5	Testing
NaCGEC-All	6,369	1.2	Testing
After handling the data leakage problem			
FCGEC-Train	36,222	1.3	Training
NaSGEC-Exam	2,617	1.5	Testing
NaCGEC-All	3,152	1.2	Testing

Table 3: Statistics of Chinese original and leakage-processed NS GEC datasets. The average reference number of the FCGEC-Test is not counted as it has not been open-source, which can only be evaluated by submitting results online.

We find that the NS datasets have a severe **data leakage** problem. There are many same or very similar (input) sentences that are contained in both the training and test datasets, especially when we use NaSGEC-Exam as the test set, since FCGEC and NaSGEC-Exam are from the same data source.

This data leakage problem introduces uncertain factors and may affect the reliability of the experiment results. Therefore, we propose a preprocessing procedure to deal with the issue. Appendix A.2 gives the details. We strongly suggest that future research can follow our practice or even improve our procedure.

Besides FCGEC and NaSGEC-Exam, our preprocessing procedure also considers NaCGEC (Ma et al., 2022), which has the same data source, to facilitate future experiments, though we have not conducted experiments on NaCGEC in this work.

Table 3 gives the statistics of the NS datasets, both before and after handling the data leakage problem.

Evaluation metrics. In all the experiments, we employ P/R/$F_{0.5}$ scores calculated by ChERRANT (Zhang et al., 2022a) for evaluation.

5.1 Implementation Details

Our code is based on the PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020) frameworks. We use `fnlp/bart-large-chinese` and `HillZhang/pseudo_native_bart_CGEC` for L2 and NS experiments, respectively. For the settings of hyperparameters, please refer to Appendix B. We complete all the experiments on two Tesla V100-SXM2-32GB GPUs.

5.2 Main Results and Analyses

The main result is shown in Table 4. Benefiting from our preprocessing strategy, the CAT approach

Model	MuCGEC-Test			FCGEC-Dev			FCGEC-Test			NaSGEC-Exam		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
SynGEC (Zhang et al., 2022b)	54.69	29.10	46.51	-	-	-	-	-	-	-	-	-
NaSGEC-BART (Zhang et al., 2023)	53.84	29.77	46.34	-	-	-	-	-	-	24.81	11.16	19.93
STG-Joint (Xu et al., 2022)	-	-	-	50.00	39.21	47.39	48.19	37.14	45.48	-	-	-
CAT	53.19	30.56	46.31	52.33	30.21	45.45	64.46	36.45	55.79	60.81	30.09	50.35
MLD	55.30	26.79	45.78	54.84	28.66	46.15	64.72	33.51	54.37	62.07	28.03	49.70
AvgL	56.37	29.12	**47.47**	50.45	34.09	46.03	63.06	39.94	56.52	58.36	33.06	50.61
MinL	54.92	29.02	46.60	54.01	32.17	47.50	65.62	36.79	56.68	60.90	30.40	50.68
AvgL + MinL	56.25	28.35	47.00	54.11	32.66	**47.80**	65.71	37.78	**57.22**	61.38	30.78	**51.17**

Table 4: Main experimental results. SynGEC and NaSGEC-BART use Lang-8 and 5 × HSK as training data for all experiments, while STG-Joint uses original FCGEC-Train as training data. We retest the performance of NaSGEC-BART on the NaSGEC-Exam after handling the data leakage. CAT means concatenation approach, and MLD stands for minimum Levenshtein distance approach. AvgL and MinL stand for average and minimum training loss approachs, respectively. All results are the average of three random seeds.

Dataset	#Input	Avg. Refs	Overlap %
Lang-8	274,120	2.69	70.91
FCGEC-Train	7,035	2.16	58.80

Table 5: Statistics of multi-reference instances in Lang-8 and FCGEC-Train, and the percentage of multi-reference instances in which MLD and MinL select the same reference. The best-performing AvgL + MinL model is used to compute the minimum loss.

Dataset	MLD	MinL	Superb	Poor
Lang-8 (300)	79	69	86	66
FCGEC-Train (300)	13	58	221	8

Table 6: Manual evaluation of MLD/MinL selected references. MLD shows the percentage where MLD picks high-quality and MinL picks poor-quality references, and vice versa for MinL. Superb/Poor indicates the percentage of instances where both approaches select high/poor quality references.

achieves performance on par with NaSGEC-BART without using the HSK dataset in the L2 experiment. Besides, the CAT approach significantly outperforms NaSGEC-BART in the NS experiment, responding to the vast differences between GEC datasets sourced from L2 and NS.

Compared to the CAT approach, the MLD approach improves the precision, but the recall decreases accordingly, resulting in a drop of $F_{0.5}$ score. This shows that only learning the reference closest to the input and ignoring valuable GEC knowledge in other references causes the model to become excessively cautious, which will lead to a decrease in model performance.

In comparison with the CAT approach, our AvgL + MinL approach achieves an increase of 0.69, 2.35, 1.43, and 0.82 in the $F_{0.5}$ score on MuCGEC-Test, FCGEC-Dev, FCGEC-Test, and NaSGEC-Exam datasets, respectively. It's noteworthy that although AvgL + MinL achieves the best performance as expected in the NS experiment, AvgL obtains the top performance (+1.16 compared to CAT) in the L2 experiment. This may be because Lang-8 contains many annotations with incomplete corrections. Although MinL is capable of considering multiple references, it gradually focuses on one single reference during the training process. Therefore, despite an improvement in performance over MLD, it remains prone to the effects of noise.

In contrast, AvgL can combine the advantages of each reference for training. Unlike Lang-8, FCGEC-Train comes from high-quality manual annotation, and the MinL model can pick out the reference with the best quality for training. We also conduct experiments on the original NS datasets. The performance of the models is much higher than those after handling the data leakage problem, which is intuitively reasonable. More importantly, the overall trends are consistent regarding the effectiveness of our proposed approach. Appendix C gives the detailed results.

5.3 Overlap and Quality Analyses

We extract multi-reference instances from Lang-8 and FCGEC-Train, and use MLD and MinL to select a reference per instance. The results are shown in Table 5. The references selected by the two approaches show notable differences, especially on FCGEC-Train which has fewer average references.

We conduct manual annotation on 600 instances to compare the quality of references selected by the two approaches, and the details are presented in Appendix D. The results are shown in the Table 6. In Lang-8, we find that MLD tends to select higher-quality references more frequently, whereas

FCGEC shows the opposite trend. we suggest that this discrepancy might stem from MinL's propensity to favor higher-fluency references. As a result, the chosen references are prone to overcorrection, this phenomenon is also observed in the case study in Appendix E. Concurrently, many references only partially correct the errors in the input and are marked as "Poor", making MLD more susceptible to noise. In contrast, MinL can dynamically learn from other references, thereby being less affected by such disturbances. In FCGEC, where references are derived from manual annotations, the quality of references selected by both approaches is generally high. Under these circumstances, MLD lacks the abundance of useful knowledge, leading to a lower recall compared to MinL. Moreover, in instances of selection inconsistency, MinL predominantly chooses references of higher quality, while MLD prefers references with minimal modifications, resulting in insufficiently thorough corrections.

6 Related Works

As the most closely related work with ours, Zheng et al. (2018) investigate multi-reference training for the MT task. They compare three methods, all of which are similar to the CAT method in this work. Unlike the circumstances for GEC, the proportion of multi-reference instances is extremely low in training data for MT. Therefore, they first train the model on large-scale single-reference data, and then continue training on small-scale multi-reference data that is actually composed of several other evaluation datasets. Another interesting contribution of their work is to generate more references given a multi-reference instance.

Lin et al. (2022) target the diverse MT task, which requires that a single model outputs multiple diverse but correct translations. They generate pseudo multi-reference training data using an off-the-shelf neural MT model, and during training adjust the loss function according to the quality of each reference and the diversity among references.

Similar to the MT task, the GEC task also involves multi-reference instances, and the GEC community has paid attention to this issue. Wang et al. (2021a) annotate YACLC, a multi-reference GEC dataset, which contains 14.6 references per input sentence on average. For a more accurate evaluation, CLEME (Ye et al., 2023) combines all references of the same input and evaluates the model performance at the chunk level.

Among these works, the question of how to better train models with multiple references has not been fully explored. The only previous work that we are aware of is Ye et al. (2022), which proposes selecting a reference from multiple references for each input sentence based on handcrafted rules during the preprocessing phase. However, this approach may not fully utilize the information contained in multiple references. In contrast, our MinL approach dynamically selects references in real time during the training phase.

7 Conclusion

This paper focuses on optimizing the utilization of multi-reference GEC training data. We introduce two new methods and a simple two-stage training strategy. The experiments on the L2 and NS datasets reveal that our method significantly improves the performance of the model training on multi-reference data, and outperforms previous proposed methods in the quality of selected references.

8 Limitations

Although the proportion of multi-reference training data is much higher for GEC than for MT, the average reference number for each instance is quite low. Most multi-reference instances have only two references to be considered during training. With more references to consider, different approaches may display greater potential in improving GEC performance. To this end, we may consider increasing the number of references for multi-reference training instances, similar to the work of Zheng et al. (2018), which we think is possible, especially for sentences having complex grammatical errors.

Acknowledgements

We thank the anonymous reviewers for their insightful feedback and constructive suggestions. We thank Yu Zhang for developing and maintaining `Supar`, on which we built our code.[4] Our gratitude extends to Houquan Zhou for his vigilant discovery and invaluable assistance in the issue of NS data leakage, and the help in perfecting our paper.

This work was supported by the National Natural Science Foundation of China (Grant No. 62176173 and 62336006), Alibaba Group through the Alibaba Innovative Research Program, and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

[4] https://github.com/yzhangcs/parser

References

Andrey Bout, Alexander Podolskiy, Sergey Nikolenko, and Irina Piontkovskaya. 2023. Efficient Grammatical Error Correction Via Multi-Task Training and Optimized Training Schedule. In *Proceedings of EMNLP*, pages 5800–5816, Singapore.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 Shared Task on Grammatical Error Correction. In *Proceedings of BEA*, pages 52–75, Florence, Italy.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical Error Correction: A Survey of the State of the Art. *CL*, pages 643–701.

Xiliang Cui and Baolin Zhang. 2011. The Principles for Building the "International Corpus of Learner Chinese". *Applied Linguistics*, pages 100–108.

Vladimir Iosifovich Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. In *Doklady Akademii Nauk*, pages 845–848.

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 ACL*, pages 7871–7880, Online.

Huan Lin, Baosong Yang, Liang Yao, Dayiheng Liu, Haibo Zhang, Jun Xie, Min Zhang, and Jinsong Su. 2022. Bridging the Gap between Training and Inference: Multi-Candidate Optimization for Diverse Neural Machine Translation. In *Findings of NAACL*, pages 2622–2632, Seattle, United States.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of ICLR*.

Shirong Ma, Yinghui Li, Rongyi Sun, Qingyu Zhou, Shulin Huang, Dingchao Zhang, Li Yangning, Ruiyang Liu, Zhongli Li, Yunbo Cao, Haitao Zheng, and Ying Shen. 2022. Linguistic Rules-Based Corpus Generation for Native Chinese Grammatical Error Correction. *ArXiv preprint*, abs/2210.10442.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. The Effect of Learner Corpus Size in Grammatical Error Correction of ESL Writings. In *Proceedings of COLING*, pages 863–872, Mumbai, India.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of CoNLL*, pages 1–14, Baltimore, Maryland.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in NeurIPS*, pages 8024–8035.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and Aspect Error Correction for ESL Learners Using Global Context. In *Proceedings of ACL*, pages 198–202, Jeju Island, Korea.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in NeurIPS*, pages 5998–6008.

Yingying Wang, Cunliang Kong, Liner Yang, Yijun Wang, Xiaorong Lu, Renfen Hu, Shan He, Zhenghao Liu, Yuxiang Chen, Erhong Yang, and Maosong Sun. 2021a. YACLC: A Chinese Learner Corpus with Multidimensional Annotation. *ArXiv preprint*, abs/2112.15043.

Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021b. A Comprehensive Survey of Grammatical Error Correction. *TIST*, pages 1–51.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of EMNLP*, pages 38–45, Online.

Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Jiayu Fu, and Ming Cai. 2022. FCGEC: Fine-Grained Corpus for Chinese Grammatical Error Correction. In *Findings of EMNLP*, pages 1900–1918, Abu Dhabi, United Arab Emirates.

Jingheng Ye, Yinghui Li, Shirong Ma, Rui Xie, Wei Wu, and Haitao Zheng. 2022. Focus Is What You Need For Chinese Grammatical Error Correction. *ArXiv preprint*, 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. In *Proceedings of EMNLP*, pages 6174–6189, Singapore.

Baolin Zhang and Xiliang Cui. 2013. Design Concepts of "the Construction and Research of the Interlanguage Corpus of Chinese from Global Learners". *Language Teaching and Linguistic Study*, pages 27–34.

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 NAACL*, pages 3118–3130, Seattle, United States.

Yue Zhang, Bo Zhang, Haochen Jiang, Zhenghua Li, Chen Li, Fei Huang, and Min Zhang. 2023. NaSGEC: a Multi-Domain Chinese Grammatical Error Correction Dataset from Native Speaker Texts. In *Findings of ACL*, pages 9935–9951, Toronto, Canada.

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 EMNLP*, pages 2518–2531, Abu Dhabi, United Arab Emirates.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the NLPCC 2018 Shared Task: Grammatical Error Correction. In *Proceedings of NLPCC*, pages 439–445.

Renjie Zheng, Mingbo Ma, and Liang Huang. 2018. Multi-Reference Training with Pseudo-References for Neural Translation and Text Generation. In *Proceedings of EMNLP*, pages 3188–3197, Brussels, Belgium.

Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023. Improving Seq2Seq Grammatical Error Correction via Decoding Interventions. In *Findings of EMNLP*, pages 7393–7405, Singapore.

A Data Preprocessing

We merge and deduplicate the references of the same input for all L2 and NS datasets.

A.1 Preprocessing L2 Data

We remove references unchanged from the input in Lang-8 and then clean it for denoising. First, we delete samples whose reference number is inconsistent with the manual annotation. Then, we remove references that differ by more than 1.5 times in length from the input sentence following (Zhou et al., 2023). Besides, we use regular expressions to reduce punctuation errors in references. For example, we retain only one punctuation when continuous Chinese punctuations (commas, periods, etc.) appear. Please refer to our code for all cleaning techniques. Additionally, we strictly removed all input sentences that coincide with MuCGEC from Lang-8 following Zhang et al. (2022a).

A.2 Handling the Leakage Issue of NS Data

When analyzing the model performance on the NS GEC benchmarks, several peculiar cases lead us to suspect the existence of data leakage. After carefully examining the input sentences in the NaSGEC benchmarks and the training set of the FCGEC, we find many identical input sentences in both datasets. In addition to these identical input sentences, we also find numerous slightly different input sentences that contain the same erroneous portions and require the same corrections, as shown in Table 7. These cognate sentences cause data leakage issues in GEC.

After analyzing a substantial number of samples, we take 60 as the threshold of the BLEU score (after removing punctuation) to determine whether two sentences are cognate. Our statistics show that 60.8% of the sentences in NaSGEC-Exam and 50.5% of the sentences in NaCGEC-All (a concatenation of NaCGEC-Dev and NaCGEC-Test) are cognate sentences with FCGEC-Train-Filtered.

To address the data leakage issue, we propose two general strategies, which can be used alone or in combination. The first strategy is moving the cognate sentences in the test datasets to the training datasets, and the second strategy is deleting cognate sentences from the training datasets.

In this work, we use these two strategies in combination to solve the data leakage issue. Specifically, we first move the cognate sentences in NaSGEC-Exam and NaCGEC-All to FCGEC-Train-Filtered, obtaining FCGEC-Train-V2, NaSGEC-Exam-V2, and NaCGEC-All-V2. Next, we delete the cognate sentences in FCGEC-Dev, FCGEC-Test, NaSGEC-Exam-V2, and NaCGEC-All-V2 from FCGEC-Train-V2, obtaining the final version, FCGEC-Train-V3.

Our code for handling the data leakage problem is available in our GitHub repository, along with a detailed log of preprocessing FCGEC, NaSGEC, and NaCGEC.

We do not report the results of NaCGEC-All in this paper because its results are quite similar to those of NaSGEC-Exam in our preliminary experiments.

B Hyperparameters Settings

We configure the warm-up steps to be 4,000 and 200 for L2 and NS experiments, respectively. For two-stage model training, the training steps S in the first stage are designated as 10,000 for L2 exper-

Dataset		Sample	BLEU Score
FCGEC-Train-Filtered	Input	与作家不同的是，摄影家们把自己对山川、草木、城市、乡野的感受没有倾注于笔下，而是直接聚焦于镜头。 Unlike writers, photographers pour their feelings about mountains, plants, cities and countryside do not into their writings, but focus directly on the lens.	60.4
	Ref	与作家不同的是，摄影家们没有把自己对山川、草木、城市、乡野的感受~~没有~~倾注于笔下，而是直接聚焦于镜头。 Unlike writers, photographers do not pour their feelings about mountains, plants, cities and countryside into their writings, but focus directly on the lens.	
NaSGEC-Exam	Input	与作家不同的是，摄影家们把自己对大自然中山川、草木、河流的独特感受没有倾注于笔下，而是直接聚焦于镜头，用画面与读者交流。 Unlike writers, photographers pour their unique feelings about mountains, plants and rivers in nature do not into their writings, but focus directly on the lens and communicate with readers through images.	
	Ref	与作家不同的是，摄影家们没有把自己对大自然中山川、草木、河流的独特感受~~没有~~倾注于笔下，而是直接聚焦于镜头，用画面与读者交流。 Unlike writers, photographers do not pour their unique feelings about mountains, plants and rivers in nature into their writings, but focus directly on the lens and communicate with readers through images.	
FCGEC-Train-Filtered	Input	从这件平凡的小事中，说明了一个深刻的道理。 In this ordinary little thing illustrates a profound truth.	61.7
	Ref	~~从~~这件平凡的小事~~中~~，说明了一个深刻的道理。 This ordinary little thing illustrates a profound truth.	
NaCGEC-All	Input	从这件平凡的小事中，却说明了一个重大问题。 In this ordinary little thing illustrates a major problem.	
	Ref	~~从~~这件平凡的小事~~中~~，却说明了一个重大问题。 This ordinary little thing illustrates a major problem.	

Table 7: Examples of cognate but different sentences. BLEU scores of two input sentences are calculated after removing punctuation. The underlined words in the input sentences are different but do not affect the correction.

Model	FCGEC-Dev			FCGEC-Test			NaSGEC-Exam		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
CAT	56.18	35.97	50.49	65.47	38.49	57.40	63.42	38.69	56.19
MLD	57.75	34.02	50.37	65.63	35.26	55.81	65.37	35.26	55.44
AvgL	57.66	35.21	51.11	66.97	36.84	57.54	64.90	36.53	56.14
MinL	58.27	37.56	52.47	66.42	38.24	57.88	65.58	39.74	58.01
AvgL + MinL	59.91	36.35	**53.00**	67.94	36.81	**58.09**	67.11	38.50	**58.39**

Table 8: Experimental results with original NS data. The steps of the first stage are set to 525 for the AvgL-MinL approach. All results are the average of three random seeds.

iments and 460 for NS experiments. The learning rate is set to 3×10^{-5}, and the dropout rate is set to 1×10^{-2}. We set the gradient clip to 1.0 and our model has a max input length 1024. In addition, we use the batch size 16384, and the update frequency is set to 5. The number of total epochs is set to 60, and we use the early stopping strategy whose patience is 3 and 5 for L2 and NS experiments, respectively. We use the AdamW (Loshchilov and Hutter, 2019) with default parameters as the optimizer. In the inference stage, we use batch size 1024. The beam search strategy is conducted with a beam size of 12.

Dataset	MLD	MinL	Superb	Poor
Lang-8 (100)	30	20	37	13
FCGEC-Train (100)	3	19	76	2

Table 9: Results of annotating initial 200 instances of MLD/MinL selected references.

C Original NS Data Experiments

In addition to utilizing the leakage-processed NS data, we also conduct experiments with the original NS datasets. The resultant findings, depicted in Table 8, mirror the trends observed in Section 5.2.

Lang-8	Input	我自己思考思考了，然后做了检查在网上。 I think think by myself, and then on the internet do a check.
	Ref. MLD	我自己思考~~思考~~了，然后做了~~检查~~在网上调查。 I think by myself, and then on the internet do a check.
	Ref. MinL	我自己想~~思考思考~~了想，然后~~做了检查~~在网上做了检查。 I think it over by myself, and then do a check on the internet.
FCGEC-Train	Input	我们经过一个冬天的奋战，一道拦河大坝终于建成了。 We after work hard all winter, a big dam is finally completed.
	Ref. MLD	我们~~经过~~一个冬天的奋战，一道拦河大坝终于建成了。 We work hard all winter, a big dam is finally completed.
	Ref. MinL	经过我们~~经过~~一个冬天的奋战，一道拦河大坝终于建成了。 After we work hard all winter, a big dam is finally completed.

Table 10: Case study of references selected by MLD and MinL.

D Manual Quality Annotation

When the training of the "AvgL+MinL" model is completed, we randomly sampled a total of 600 instances from Lang-8 and FCGEC-Train (300 for each) that receive different references according to the MLD and MinL metrics. Then we manually evaluate the quality of the references, in order to understand which metric can select better references.

Before the annotation process, we ensure that the annotators cannot identify which method selects each reference, and we shuffle the order of the references for all instances to maintain impartiality. Then we assigned them to two graduate students (Yumeng and Haochen) who specialized in GEC for independent annotation. The two annotators subsequently engage in discussions to resolve disagreements and reach a consensus to get the final annotation result.

At first, we have annotated 200 instances. As suggested by an anonymous reviewer, and in order to make our analysis findings more reliable, we increased the number of annotation instances from 200 to 600. Table 9 shows the annotation results for the initial 200 instances, which are consistent with the results for Table 6.

E Case Study

The case study to compare the selection of MLD and MinL is shown in Table 10. In Lang-8, the references selected by both methods fail to correct all the errors in the input. Concurrently, MinL prefers to select more fluent references than the MLD, allowing the model to correct more errors but may lead to over-correction. This could be why MinL has a higher recall rate but a lower precision rate than MLD in Table 4. In contrast, the references for FCGEC come from manual annotations. As shown in Table 9, most instances have at least one qualified reference, and MinL is more capable of finding the optimal reference than MLD.