MuCGEC: a Multi-Reference Multi-Source Evaluation Dataset for Chinese Grammatical Error Correction

Yue Zhang[1†], **Zhenghua Li**[1*], **Zuyi Bao**[2], **Jiacheng Li**[1],
Bo Zhang[2], **Chen Li**[2], **Fei Huang**[2], **Min Zhang**[1]
[1]Institute of Artificial Intelligence, School of Computer Science and Technology,
Soochow University, China; [2]DAMO Academy, Alibaba Group, China
[1]{yzhang21,jcli20}@stu.suda.edu.cn, [1]{zhli13,minzhang}@suda.edu.cn
[2]{zuyi.bzy,klayzhang.zb,puji.lc,f.huang}@alibaba-inc.com

Abstract

This paper presents MuCGEC, a multi-reference multi-source evaluation dataset for Chinese Grammatical Error Correction (CGEC), consisting of 7,063 sentences collected from three Chinese-as-a-Second-Language (CSL) learner sources. Each sentence is corrected by three annotators, and their corrections are carefully reviewed by a senior annotator, resulting in 2.3 references per sentence. We conduct experiments with two mainstream CGEC models, i.e., the sequence-to-sequence model and the sequence-to-edit model, both enhanced with large pretrained language models, achieving competitive benchmark performance on previous and our datasets. We also discuss CGEC evaluation methodologies, including the effect of multiple references and using a char-based metric. Our annotation guidelines, data, and code are available at https://github.com/HillZhang1999/MuCGEC.

1 Introduction

Given a potentially noisy input sentence, grammatical error correction (GEC) aims to detect and correct all errors and produce a clean sentence. Recently, GEC has increasingly gained attention for its vital value in various downstream scenarios (Grundkiewicz et al., 2020; Wang et al., 2021).

To support GEC research, high-quality manually labeled evaluation data is indispensable. For English GEC (EGEC), such datasets are abundant (Yannakoudakis et al., 2011; Dahlmeier et al., 2013; Ng et al., 2014; Napoles et al., 2017; Bryant et al., 2019; Napoles et al., 2019; Flachs et al., 2020). However, Chinese GEC (CGEC) evaluation datasets are relatively scarce. The two publicly available CGEC evaluation datasets are NLPCC18 and CGED, contributed by the NLPCC-2018 (Zhao et al., 2018) and the series of CGED shared tasks (Rao et al., 2018, 2020), respectively.

Source	我不知道他何时返回回来。 I don't know when he will return back.
Ref. 1	我不知道他何时返回~~回来~~。 I don't know when he will return.
Ref. 2	我不知道他何时~~返回~~回来。 I don't know when he will be back.

Table 1: A CGEC example with two references.

Most EGEC evaluation datasets provide multiple references for each input sentence, such as CoNLL14-test (Ng et al., 2014) and BEA19-test (Bryant et al., 2019). In contrast, sentences in existing CGEC evaluation datasets usually have only one reference (i.e., 87% of the sentences in NLPCC18 and all in CGED). This is probably due to the different annotation workflows adopted.

As suggested by Bryant and Ng (2015), enforcing multi-reference annotation is crucial for both GEC model evaluation and GEC data annotation, because there usually exist more than one acceptable reference with similar meanings for an incorrect sentence, as illustrated by the example in Table 1. On the one hand, if a GEC model outputs a correct reference, which is yet different from the one given in the evaluation data, then the model performance will be unfairly underestimated. To mitigate this issue, a straightforward solution is increasing the number of references (Sakaguchi et al., 2016; Choshen and Abend, 2018). On the other hand, imposing a single-reference constraint makes data annotation problematic. If annotators submit different equally acceptable corrections, which is very common, it will be difficult for the senior annotator to decide which one is the best.

Besides the lack of multiple references, existing CGEC datasets collect sentences from a single text source, which may be insufficient for robust model evaluation (Mita et al., 2019). In addition, we be-

† This work was partially done during the first author's internship at Alibaba DAMO Academy.
* Corresponding author.

lieve that it is beneficial for improving data quality to compile comprehensive annotation guidelines.

To fill these gaps, this paper presents a multi-reference multi-source evaluation dataset for CGEC, named MuCGEC. After investigating previous works on constructing GEC datasets, we compile comprehensive annotation guidelines. Based on a browser-based online annotation tool, each sentence is assigned to three annotators for independent correction, and one senior annotator for final review. An annotator may submit multiple references, and the senior annotator may also supplement new references besides rejecting incorrect submissions. In this way, we aim to produce as many references as possible. In summary, this work makes the following contributions.

(1) Our newly constructed MuCGEC consists of 7,063 sentences from three representative sources of Chinese-as-a-Second-Language (CSL) learner texts. Each sentence obtains 2.3 references on average. We conduct detailed analyses on our new dataset to gain more insights.

(2) We conduct benchmark experiments using two mainstream and competitive CGEC models, i.e., the sequence-to-edit (Seq2Edit) and sequence-to-sequence (Seq2Seq) models, both enhanced with pretrained language models (PLMs). We also experiment with an extremely effective ensemble strategy. Moreover, we investigate the effect of multiple references on model evaluation, and propose to use a char-based evaluation metric, which we believe is simpler and more suitable than previous word-based ones for CGEC.

2 Data Annotation

2.1 Multi-Source Data Selection

This work focuses on CSL learner texts. In order to investigate diverse types of Chinese grammatical errors, we select data from the following three sources.

(1) We re-annotate the NLPCC18 test set (Zhao et al., 2018), which contains 2,000 sentences from the Peking University (PKU) Chinese Learner Corpus.

(2) We select and re-annotate sentences from CGED-2018 and CGED-2020 test datasets (Rao et al., 2018, 2020), which come from the writing section of the HSK exam (Hanyu Shuiping Kaoshi, translated as the Chinese level exam), an official Chinese proficiency test. After removing sentences marked as correct from the total 5,006 ones, we obtain 3,137 potentially erroneous sentences for re-annotation.

(3) Lang8[1] is a language learning platform, where native speakers voluntarily correct texts written by second-language learners. The NLPCC-2018 shared task organizers collect about 717K Chinese sentences with their corrections from Lang8 and encourage participants to use them as the training data. We randomly select 2,000 sentences with 30 to 60 characters for re-annotation.

In the end, we obtain 7,137 sentences. For simplicity, we discard all original corrections and directly perform re-annotation from scratch following our new annotation guidelines and workflow.

2.2 Annotation Paradigm: Direct Rewriting

There are mainly two types of annotation paradigms for constructing GEC data, i.e., *error-coded* and *direct rewriting*. The *error-coded* paradigm requires annotators to explicitly mark the erroneous span in the original sentence, then choose its error type, and finally make corrections. Ng et al. (2013, 2014) adopt the *error-coded* paradigm for constructing data for the CoNLL-2013/2014 EGEC shared tasks. For CGEC, the original NLPCC18 and CGED datasets both follow the *error-coded* paradigm as well.

As discussed by Sakaguchi et al. (2016), the *error-coded* paradigm poses two challenges. First, it is extremely difficult for different annotators to agree upon the boundaries of the erroneous spans and their error types, especially when there are many categories to consider (Bryant et al., 2017). This inevitably leads to an increase in annotation effort and a decrease in annotation quality. Second, under such a complex annotation paradigm, annotators would pay less attention to the fluency of the resulting reference, sometimes even leading to unnatural expressions.

Instead, the *direct rewriting* paradigm asks annotators to directly rewrite the input sentence and produce a corresponding grammatically correct one, without changing the original meaning. In order to

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

Major Types	Minor Types
Punctuation	Missing; Redundancy; Misuse
Spelling	Phonetic confusion; Glyph confusion; Character disorder
Word	Missing; Redundancy; Misuse
Syntax	Word order; Mixing syntax patterns
Pragmatics	Logical inconsistency; Ambiguity; Commonsense mistake

Table 2: The 5 major and 14 minor error types adopted by our guidelines for organizing the content.

Source	我的爸爸经常我。 My dad usually me.
Previous	我的爸爸经常骂我。 My dad usually scolds me.
Ours	我的爸爸经常[MC]我 My dad usually [MC] me.

Table 3: An example for handling context-dependent missing components. The inserted tokens are underlined. "Previous" means annotation in previous datasets, and "Ours" refers to our annotation.

evaluate model performance, edits can be extracted automatically from parallel sentences by additional tools (Bryant et al., 2017).

This annotation paradigm has proven to be efficient and cost-effective (Sakaguchi et al., 2016), and has been adopted by several recent GEC data construction works (Napoles et al., 2017, 2019; Syvokon and Nahorna, 2021; Náplava et al., 2022).

In this work, we adopt the *direct rewriting* paradigm. Besides above-mentioned advantages, we believe this paradigm can help improve the diversity of references since annotators can correct errors more freely.

2.3 Annotation Guidelines

After an extensive survey of previous work on GEC data construction, we compiled 30 pages of comprehensive guidelines for CGEC annotation. During the course of the annotation process, we gradually improved our guidelines according to feedback from annotators.

To facilitate learning, our guidelines adopt a two-tier hierarchical error taxonomy, including 5 major error types and 14 minor types, as shown in Table 2. Our guidelines describe in detail how to handle each minor error type and provide typical examples. We will release our guidelines along with the dataset, which we hope can benefit future research.

For dealing with word-missing errors, we found that it was unreasonable to simply insert certain words when the missing words are context-dependent, which means the missing words are related to context beyond the given sentence. Table 3 shows a sentence in which a verb is missing. However, under sentence-level GEC annotation, annotators are unable to decide the specific missing verb. According to our observation, previous CGEC datasets directly insert specific words like "scolds" under such circumstances, which we think is inaccurate and may cause trouble for GEC model evaluation, because there are many other acceptable candidates. To handle this problem, we instead insert a special tag named **context-dependent missing components (MC)**. We find about 1% of sentences in MuCGEC contain "[MC]" tags. Current GEC models cannot handle "[MC]", since "[MC]" is not included in existing training data and vocabulary. We leave this issue as future work.

2.4 Annotation Workflow and Tool

In order to encourage more diverse and high-quality references, we assign each sentence to three random annotators for independent annotation. Their submissions are then aggregated and sent to a random senior annotator (reviewer) for review. An annotator may submit multiple references for one sentence if he/she thinks they are all correct according to the guidelines. The job of the senior annotator includes: 1) modifying incorrect references into correct ones (sometimes just rejecting them); 2) adding new correct references according to the guidelines. After review, the accepted references are defined as **final golden references**.

For the sake of self-improvement, we employ a self-study mechanism that allows annotators to learn from their mistakes if they submit an incorrect reference. Concretely, if an annotator submits a reference that is not included in the final golden references, he/she has to modify his/her submission into a correct one. Moreover, the annotator can also make complaints if he/she insists that his/her submission is correct. We find that the self-study and making-complaints mechanisms can trigger very helpful discussions.

To improve annotation efficiency, we have developed a browser-based online annotation tool to support the above workflow and mechanisms. Due

Dataset	#sent	#err. sent (perc.)	chars/sent	edits/ref	refs/sent
Original NLPCC18	2000	1983 (99.2%)	29.7	2.0	1.1
MuCGEC (NLPCC18)	1996 (4)	1904 (95.4%)	29.7	2.5	2.5
MuCGEC (CGED)	3125 (12)	2988 (95.6%)	44.8	4.0	2.3
MuCGEC (Lang8)	1942 (58)	1652 (85.1%)	37.5	2.8	2.1
MuCGEC	7063 (74)	6544 (92.7%)	38.5	3.2	2.3

Table 4: Data statistics, including the number of sentences, the number (proportion) of erroneous sentences, the average number of characters per sentence, the average number of edits per reference, and the average number of references per sentence. Some sentences in our source data were discarded since annotators could not understand their meaning and thus were unable to correct them. Numbers in parentheses in the "#sent" row refer to such sentences.

to the space limitation, we show the visual interfaces for annotation and review in Appendix B.

2.5 Annotation Process

We employed 21 undergraduate students who are native speakers of Chinese and familiar with Chinese grammar as part-time annotators. Annotators received intensive training before real annotation. In the beginning, two authors of this paper, who were also in charge of compiling the guidelines, served as senior annotators for review. After one month, when the annotators were familiar with the job, we selected 5 outstanding annotators as senior annotators to join the review.

All participants were asked to annotate for at least 1 hour every day. The whole annotation process lasted for about 3 months.

2.6 Ethical Issues

All annotators and reviewers were paid for their work. The salary was determined by both submission numbers and annotation quality. The average salary of annotators and reviewers is 24 and 35 RMB per hour respectively.

All the data of the three sources are publicly available. Meanwhile, we have obtained permission from organizers of the NLPCC-2018 and CGED shared tasks to release our newly annotated references in a proper way. We are deeply grateful to them for their kind support.

3 Analysis of Our Annotated Data

Overall statistics of MuCGEC are shown in Table 4. We also include the original NLPCC18 dataset (Zhao et al., 2018) for comparison[2].

First, from the proportion of erroneous sentences, we can see that most of the sentences are considered to contain grammatical errors in the original annotation, but a considerable part of them are not corrected in our annotation. We attribute this to our strict control of the over-correction phenomenon.

Second, regarding sentence lengths, NLPCC18 has the shortest sentences, whereas CGED sentences are much longer. This may be because candidates on the HSK examination, an official Chinese proficiency test, tend to use long sentences to show their ability in Chinese.

Third, each sentence in the re-annotated NLPCC18 receives 2.5 references on average, which is more than twice that in the original NLPCC18 data. Overall, each sentence obtains 2.3 references. We believe the multi-reference characteristic makes our dataset more reliable for evaluation, which is further discussed in Section 6.3.

Finally, we compare the number of char-based edits per reference in different datasets. We describe how to derive such edits in detail in Section 6.2. We can see that the number of edit is tightly correlated with sentence length. The difference in the average sentence length and number of edits indicates that the three data sources may have a systematic discrepancy in quality and difficulty, which we believe is helpful for evaluating the generalization ability of models. Moreover, compared with NLPCC18-orig, we annotate 25% more edits (2.0 vs. 2.5) in each reference. We believe the major reason is that the original NLPCC18 data are annotated under the *minimal edit distance* principle (Nagata and Sakaguchi, 2016), which requires annotators to select a reference with fewer edits when correcting errors.

[2]Here we do not compare with the original CGED and Lang8 datasets since: 1) the CGED-orig mainly focuses on error detection annotation and does not provide corrections for word-order errors; 2) the Lang8-orig is automatically collected from the internet, and its correction is quite noisy.

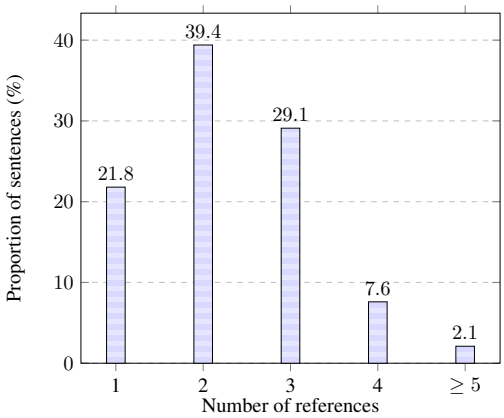

Figure 1: The proportion of sentences with different number of references in MuCGEC.

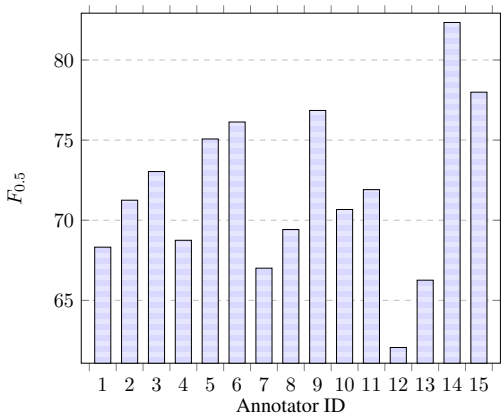

Figure 2: The human performance of the 15 annotators who annotated the most sentences.

Sentence distribution with respect to numbers of references is shown in Figure 1. Here, we only consider erroneous sentences. Identical references from different annotators are counted as one reference. Overall, most sentences have 2 references, accounting for 39.4%; 29.1% of sentences have 3 references; 21.8% of sentences have only 1 reference, most of which are short and easy to correct.

We believe that the average number of references could be further increased if more annotators were assigned to each sentence. It is also worth noticing that annotators tend to submit a single reference, despite the fact that our annotation tool allows annotators to submit multiple ones. We suspect the reason may be that it is more economical for annotators to do so. One the one hand, it may be easy to come up with the most suitable correction, whereas thinking of alternatives is more time-consuming. On the other hand, we did not give enough consideration to this issue when designing the salary computation formula. In the future, we plan to optimize (or simplify) our annotation workflow so that each annotator is required to give only one reference which he/she thinks is the best, and assign each sentence to more annotators if we need more references.

Human annotation performance. In order to assess the annotation ability of our annotators and human performance for CGEC task, we calculate char-based $F_{0.5}$ scores by evaluating all annotation submissions against the final golden references. We describe how to compute char-based metrics in detail in Section 6.2. Each reference submitted by an annotator is considered as a sample. Overall, the average $F_{0.5}$ is 72.12, which we believe could be higher if we discarded data that were annotated at the early stage of our project when annotators were less experienced and less familiar with our guidelines.

Figure 2 shows $F_{0.5}$ scores of 15 annotators who annotated the most sentences, in descending order of the number of annotated sentences. We can see that human performance varies across different annotators. The best annotator achieves an 82.34 $F_{0.5}$ score, while the annotator who completes the most tasks only gets a score of 68.32. This indicates that we should pay more attention to annotation quality when calculating salaries and prevent annotators from focusing too much on annotation speed.

Common mistakes made by annotators. We randomly select 300 invalid references rejected by reviewers and try to understand what mistakes are more frequently made by annotators. We manually classify all selected references into three mistake categories. The most frequent mistakes are caused by *incomplete correction* and account for 56.7% of the invalid references. We found that it was sometimes difficult for annotators to correct all errors without omissions, possibly due to the complexity or flexibility of Chinese grammar. The second frequent type of mistakes is *erroneous correction*, which means that the correction of old errors incurs new errors, with a proportion of 32.3%. Besides, 11.0% of submissions are rejected due to *meaning change*, which means that the correction changes the intended meaning of the original sentence.

4 Benchmark Models

To understand how well cutting-edge GEC models perform on our data, we adopt two mainstream GEC approaches, i.e., Seq2Edit and Seq2Seq. Both models are enhanced with PLMs. We also at-

tempt to combine them after observing their complementary power in dealing with different error types. This section briefly describes these benchmark models. Due to the space limitation, please kindly refer to Appendix C for more model details.

The Seq2Edit model treats GEC as a sequence labeling task and performs error corrections via a sequence of token-level edits, including insertion, deletion, and substitution (Malmi et al., 2019). A token corresponds to a word or a subword in English, and to a character in Chinese. With minor modifications to accommodate Chinese, we adopt GECToR (Omelianchuk et al., 2020), which achieves the SOTA performance on EGEC datasets.

Following recent Seq2Edit work like Awasthi et al. (2019) and Omelianchuk et al. (2020), we enhance GECToR by using PLMs as its encoder. After comparing several popular PLMs, we choose StructBERT (Wang et al., 2019)[3] due to its superior performance after fine-tuning (see Table 5).

The Seq2Seq model straightforwardly treats GEC as a monolingual translation task (Yuan and Briscoe, 2016). Recent works propose to enhance Transformer-based (Vaswani et al., 2017) Seq2Seq EGEC models with PLMs like T5 (Rothe et al., 2021) or BART (Katsumata and Komachi, 2020). Unlike BERT (Devlin et al., 2019), T5 and BART are specifically designed for text generation. Therefore, it is straightforward to continue training them on GEC data. We follow these works and utilize the recently proposed Chinese BART from Shao et al. (2021) to initialize our Seq2Seq model.

The ensemble model. Several previous works have proven the effectiveness of model ensemble for CGEC (Liang et al., 2020; Hinson et al., 2020). In this work, we clearly observe the complementary power of the above two models in fixing different error types (see Table 7), and thus attempt to combine them.

We adopt a simple edit-wise voting mechanism. The edits are at the char-based span level, and correspond to four error types. Please refer to Section 6.2 for detailed explanation of our char-based evaluation metric. More specifically, we aggregate edits from the results of each model, and only preserve edits that appear more than $N/2$ times, where N is the number of models. In other words, an edit is kept in the final result only if it is produced by a majority of models.[4]

We experiment with two ensemble settings: 1) one Seq2Edit and one Seq2Seq, denoted by "1×Seq2Edit+1×Seq2Seq", and 2) three Seq2Edit and three Seq2Seq, denoted by "3×Seq2Edit+3×Seq2Seq".The three Seq2Edit models are obtained using different random seeds for initialization, and the same goes for the Seq2Seq.

Other settings. To obtain the single-model performance of both kinds of models, we run them three times separately with different random seeds for initialization and calculate average metrics. For "1×Seq2Edit+1×Seq2Seq", we random select a pair of single models. For "3×Seq2Edit+3×Seq2Seq", we aggregate the results of all six single models.

5 Experiments on NLPCC18-Orig Data

In order to show that our benchmark models are competitive among existing CGEC models, we conduct experiments on the original NLPCC18 test set, on which most previous CGEC systems are tested.

Training data. For the sake of easy replicability, we limit our training data strictly to public resources, i.e., the Lang8[5] data (Zhao et al., 2018) and the HSK[6] data (Zhang, 2009). We filter duplicate sentences that appear in our dataset, and discard correct sentences. The final Lang8 and HSK data contains 1,092,285 and 95,320 sentence pairs, respectively. The HSK data is cleaner and of higher quality than Lang8, but is smaller. Following the re-weighting procedure of Junczys-Dowmunt et al. (2018), we duplicate the HSK data five times, and merge them with Lang8 data.

Comparison with previous works. Table 5 shows the results. For a fair comparison, we follow the official setting of the shared task, including the word-based MaxMatch scorer (Dahlmeier and Ng,

[3] https://github.com/alibaba/AliceMind/tree/main/StructBERT

[4] In fact, our voting strategy is a little more complex due to the adaption of two pieces of modification, which consistently improve performance in our preliminary experiments. First, for word-order errors, we set the preserving threshold to $N/2-1$ considering the Seq2Seq model is much superior in handling word-order errors than the Seq2Edit model. Imagine the scenario when all Seq2Seq models agree on correcting a word-order error, whereas all Seq2Edit models disagree. Using $N/2-1$ means that a word-order edit is kept in the final result even when it is produced by exactly half of models. Second, we use a set of simple heuristic rules to recognize spelling errors, a sub-type of substitution errors, and also use $N/2-1$ as the preserving threshold for them. This is also reasonable since both GEC models can obtain high precision scores on such errors.

[5] http://tcci.ccf.org.cn/conference/2018/taskdata.php

[6] http://hsk.blcu.edu.cn

	P	R	$F_{0.5}$
Trained on Lang8			
YouDao (Fu et al., 2018)♢	35.24	18.64	29.91
AliGM (Zhou et al., 2018)♢	41.00	13.75	29.36
BLCU (Ren et al., 2018)♢	47.63	12.56	30.57
HRG (Hinson et al., 2020)♢	36.79	**27.82**	34.56
MaskGEC (Zhao and Wang, 2020)♡	**44.36**	22.18	**36.97**
Our Seq2Edit	**39.83**	23.01	34.75
Our Seq2Seq	37.67	**29.88**	**35.80**
1×Seq2Edit+1×Seq2Seq♢	**58.15**	18.35	40.55
3×Seq2Edit+3×Seq2Seq♢	55.58	**19.78**	**40.81**
Trained on Lang8+HSK			
TEA (Wang et al., 2020)♡	39.43	22.80	34.41
WCDA (Tang et al., 2021)♡	**47.41**	**23.72**	**39.51**
Our Seq2Edit (BERT)	39.61	28.53	36.76
Our Seq2Edit (RoBERTa)	39.74	30.44	37.54
Our Seq2Edit (MacBERT)	40.46	30.73	38.05
Our Seq2Edit (StructBERT)	**42.88**	30.19	**39.55**
Our Seq2Seq	41.44	**32.89**	39.39
1×Seq2Edit+1×Seq2Seq♢	**60.72**	22.48	45.31
3×Seq2Edit+3×Seq2Seq♢	59.38	**24.18**	**45.99**

Table 5: Performance comparison on the original NLPCC18 dataset (Zhao et al., 2018) using the official **word-based** evaluation script. The first group lists models that use only Lang8 for training, whereas the second group shows those using both Lang8 and HSK data. Models marked by ♢ use *model ensemble*, and those marked by ♡ use *data augmentation*.

2012) for calculating the P/R/F values. We segment model outputs by adopting the PKUNLP word segmentation (WS) tool provided by the shared task organizers (Zhao et al., 2018).

When only using Lang8 for training, our single Seq2Seq model is already quite competitive. Its performance is only lower than MaskGEC (Zhao and Wang, 2020) by 1 point in $F_{0.5}$. Please notice that MaskGEC extra uses data augmentation. For now, our benchmark models do not use any synthetic data for simplicity, but we believe data augmentation could further boost the performance of our models.

Adding the HSK training data improves performance of all our models by about 4 points. Our two benchmark models already achieve SOTA performance under the single-model setting.

The model ensemble technique leads to obvious performance gains (more than 5 points) over single models. However, the gains from increasing the number of component models seem rather small. We try to explain this issue in Section 6.3.

For Seq2Edit, we additionally present results with other PLMs besides StructBERT, including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and MacBERT (Cui et al., 2020) from the Hugging Face[7] website. We use the "large" variants of all PLMs.

6 Experiments on MuCGEC

6.1 Data Splits

For hyperparameter tuning or model selection, previous works on other CGEC datasets often randomly sample some sentence pairs from training data as the dev set (Wang et al., 2020; Zhao and Wang, 2020; Hinson et al., 2020), which is inconvenient for reproducing or comparing.

In this work, we propose to provide a fixed dev set for our newly annotated dataset, by randomly selecting 1,125 sentences from the CGED source, denoted as CGED-dev. The remaining 5,938 sentences are used as the test set, in which each data source has a roughly equal amount of sentences, i.e., 1,996 sentences for NLPCC18-test, 2,000 for CGED-test, and 1,942 for Lang8-test.

6.2 Evaluation Metrics

Problems with word-based metrics. As discussed in Section 5, previous CGEC datasets are annotated upon word sequences and adopt word-based metrics for evaluation. Before annotation and evaluation, a sentence needs to be segmented into words using a Chinese word segmentation (CWS) model. We believe this will introduce unnecessary uncertainty in CGEC evaluation procedure. First, CWS models inevitably produce word segmentation errors (Fu et al., 2020). Second, there are multiple heterogeneous CWS standards. Finally, we found that a correct edit may be judged as wrong due to the word boundary mismatch.

Char-based span-level evaluation metrics are adopted in this work instead. First, given an input sentence and a correction, we obtain an optimal sequence of *char-level edits* with the minimal edit distance. We consider three types of *char-level edits*, corresponding to three error types:

- Deleting a char for a *redundant error*;
- Inserting a char for a *missing error*;
- Substituting a char with another one for a *substitution error*;

Then, we convert all *char-level edits* into *span-level* by merging consecutive edits of the same type, following previous practice in EGEC and CGEC (Felice et al., 2016; Hinson et al., 2020)

[7] https://huggingface.co/

	NLPCC18-test			CGED-test			Lang8-test			All-test		
	P	**R**	$F_{0.5}$	**P**	**R**	$F_{0.5}$	**P**	**R**	$F_{0.5}$	**P**	**R**	$F_{0.5}$
Seq2Edit	**50.09**	32.09	**45.04**	42.87	**27.69**	38.64	**39.65**	21.62	**33.98**	**44.11**	27.18	39.22
Seq2Seq	47.99	**35.12**	44.71	**46.04**	26.97	**40.34**	36.10	**25.01**	33.16	43.81	**28.56**	**39.58**
1×Seq2Edit+1×Seq2Seq	**74.13**	24.11	52.39	**68.59**	20.35	46.53	**62.25**	14.23	37.17	**68.92**	19.68	45.94
3×Seq2Edit+3×Seq2Seq	72.82	**26.38**	**53.81**	67.95	**21.58**	**47.52**	60.65	**16.39**	**39.38**	67.76	**21.42**	**47.29**
Human	75.77	66.15	73.63	74.14	64.84	72.00	72.31	62.26	70.05	73.47	63.75	71.25

Table 6: Performance of models and our annotators on MuCGEC, using the **char-based** metric. For calculating the human performance, each submitted result is considered as a sample if an annotator submits multiple references.

	Seq2Edit	*Seq2Seq*	*Ensemble*	**Human**
Missing (**29.2%**)	41.09	40.93	42.25	69.72
Redundant (**16.1%**)	43.11	37.65	54.18	72.78
Substitution (**48.9%**)	35.99	39.98	47.37	71.69
Word-order (**5.8%**)	28.28	40.33	42.44	72.58

Table 7: $F_{0.5}$ scores for each error type on All-test. The bold numbers in parentheses show the proportion of each error type. "*Ensemble*" refers to "*3×Seq2Edit+3×Seq2Seq*".

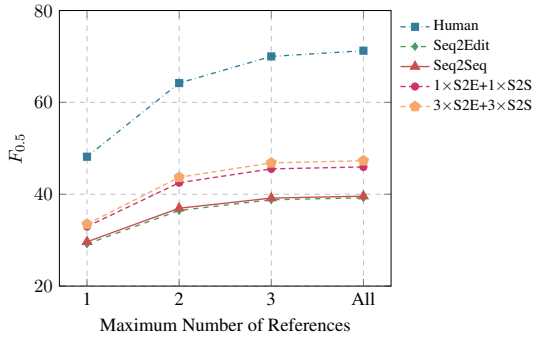

Figure 3: Effect of the number of references on $F_{0.5}$.

The above two steps are applied to both the system output sequence and golden reference, transforming them into sets of span-level edits. Finally, we can calculate the P/R/F value by comparing them. If there are multiple golden references, we will choose the one with the highest F-score.

Span-level word-order errors. When calculating overall metrics, we only consider above three types of errors. When analyzing, we distinguish the fourth error type — word-order. A span-level word-order error is usually composed of a redundant and a missing error, where the deleted span is the same as the inserted one. We use simple heuristic rules to identify such errors (Hinson et al., 2020).

Please kindly notice that we release our evaluation script as well.

6.3 Results and Analysis

Main results. Table 6 shows the char-based performance of the benchmark models and our annotators on MuCGEC. All models are trained on Lang8+HSK, as described in Section 4.

The overall trend of performance change is basically consistent with those on the original NLPCC18 dataset in Table 5. First, the Seq2Seq and Seq2Edit models perform quite closely on $F_{0.5}$, but clearly exhibit divergent strength in precision and recall, giving a strong motivation for combining them. Second, the model ensemble approach improves performance by a very large margin.

One interesting observation is that on MuCGEC, "3×Seq2Edit+3×Seq2Seq" substantially outperforms "1×Seq2Edit+1×Seq2Seq" on All-test and all three subsets. In contrast, the improvement is only modest on the original NLPCC18 test data. We suspect this may indicate that a multi-reference dataset can more accurately evaluate model performance. However, it may require further human investigation for more insights.

Finally, there is still a huge performance gap between models and humans, indicating that CGEC research still has a long way to go.

Performance on four error types. Table 7 shows more fine-grained evaluation results on four error types.

It is clear that the Seq2Edit model is better at handling redundant errors, whereas the Seq2Seq model is superior in dealing with substitution and word-order errors. For missing errors, the two perform similarly well.

These phenomena are quite interesting and can be understood after considering the underlying model architectures. On the one hand, to correct redundant errors, the Seq2Edit model only needs to perform a fixed deletion operation, which is a much more implicit choice for the Seq2Seq model, since its goal is to rewrite the whole sentence. On the other hand, the Seq2Seq is suitable to substitute or reorder words due to its natural capability of uti-

lizing language model information, especially with the enhancement of BART (Lewis et al., 2020).

Again, the model ensemble approach substantially improves performance on all error types. The ensemble model is closest to the human on redundant errors, probably because they are the easiest to correct. The largest gap occurs in word-order errors, which require global structure knowledge to correct and are extremely challenging.

Influence of the number of references. To understand the impact of the number of references on performance evaluation, we deliberately reduce the available number of reference in our dataset. For example, when the maximum number of references is limited to 2, we only kept the first two references in the dataset if a sentence has more than 2 golden references. The results are shown in Figure 3.

When the maximum number of references increases, the performance of both models and humans increases continuously, especially for humans. As only a few sentences have more than 3 references, the improvement is quite small when the maximum number of references increases from 3 to All. This trend suggests that compared with single-reference datasets, a multi-reference dataset reduces the risk of underestimating performance, and thus is more reliable for model evaluation.

7 Related Works

EGEC resources. There has been a lot of work on EGEC data construction. As the two earliest EGEC datasets, FCE (Yannakoudakis et al., 2011) and NUCLE (Dahlmeier et al., 2013) adopt the *error-coded* annotation paradigm. In contrast, JFLEG (Napoles et al., 2017) collects sentences from TOFEL exams and adopts the *direct rewriting* paradigm. W&I (Bryant et al., 2019) also chooses the *direct rewriting* paradigm, and extra annotates a score indicating the language proficiency level of the writer for each input sentence. All four datasets are composed of essays from non-native English speakers and provide multiple references.

Recently, researchers have started to annotate small-scale EGEC data for texts written by native English speakers, including AESW (Daudaravicius et al., 2016), LOCNESS (Bryant et al., 2019), GMEG (Napoles et al., 2019) and CWEB (Flachs et al., 2020). In the future, we plan to extend this work to texts written by native Chinese speakers.

CGEC resources. Compared with EGEC, progress in CGEC data construction largely lags behind. As discussed in Section 1, NLPCC18 (Zhao et al., 2018) and CGED (Rao et al., 2018, 2020) are the only two evaluation datasets for CGEC research. Besides them, there are also a few resources for training CGEC models, e.g., the Lang8 corpus (Zhao et al., 2018) and the HSK corpus (Zhang, 2009).

Concurrently with this work, Wang et al. (2022) present a multi-reference CGEC dataset, named as yet another Chinese learner corpus (YACLC), containing 32,124 sentences from Lang8. Each sentence is annotated by 10 annotators.

Recent progress in CGEC. In the NLPCC-2018 shared task (Zhao et al., 2018), many systems adopt Seq2Seq models, based on RNN/CNN. Recent work mainly utilizes Transformer (Wang et al., 2020; Zhao and Wang, 2020; Tang et al., 2021). Hinson et al. (2020) first employ a Seq2Edit model for CGEC, and achieve comparable performance with the Seq2Seq counterparts. Some systems in the CGED-2020 shared task (Rao et al., 2020) directly employ the open-source Seq2Edit model, i.e., GECToR (Liang et al., 2020). Most Seq2Edit models use PLMs like BERT (Devlin et al., 2019) to initialize their encoders. Besides the above two mainstream models, Li and Shi (2021) for the first time apply a non-autoregressive neural machine translation model to CGEC.

Besides modeling optimization, techniques like data augmentation (Zhao and Wang, 2020; Tang et al., 2021) and model ensemble (Hinson et al., 2020) have proven to be very useful for CGEC.

8 Conclusions

This paper presents MuCGEC, a newly annotated evaluation dataset for CGEC, consisting of 7,063 sentences written by CSL learners. Compared with existing CGEC datasets, ours can support more reliable evaluation due to three important features: 1) providing multiple references; 2) covering three text sources; 3) adopting strict quality control (i.e., annotation guidelines and workflow). After describing the data construction process, we perform detailed analyses of our data. Then, we adopt two mainstream and competitive CGEC models, i.e., Seq2Seq and Seq2Edit, and carry out benchmark experiments. We also propose to adopt char-based evaluation metrics to replace previously used word-based ones.

Acknowledgements

We want to thank the anonymous reviewers and Sebastian Schuster for their great help. We are also grateful to Zeyang Liu for building the annotation system, and Lei Zhang, Fukang Yan, Jiayu Shen, Houquan Zhou, Yu Zhang for helping us improve this paper, and all annotators for their great effort in data annotation. This work was partially supported by National Natural Science Foundation of China (Grant No.62176173 and No.61876116) and by Alibaba Group through Alibaba Innovative Research Program.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of EMNLP-IJCNLP*, pages 4260–4270.

Christopher Bryant, Mariano Felice, Øistein E Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of BEA@ACL*, pages 52–75.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of ACL*, pages 793–805.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proceedings of ACL*, pages 697–707.

Leshem Choshen and Omri Abend. 2018. Inherent biases in reference-based evaluation for grammatical error correction. In *Proceedings of ACL*, pages 632–642.

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

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of NAACL-HLT*, pages 568–572.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The nus corpus of learner English. In *Proceedings of BEA@NAACL-HLT*, pages 22–31.

Vidas Daudaravicius, Rafael E Banchs, Elena Volodina, and Courtney Napoles. 2016. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of BEA@NAACL-HLT*, pages 53–62.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In *Proceedings of COLING*, pages 825–835.

Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. Grammatical error correction in low error density domains: a new benchmark and analyses. In *Proceedings of EMNLP*, pages 8467–8478.

Jinlan Fu, Pengfei Liu, Qi Zhang, and Xuan-Jing Huang. 2020. Is Chinese word segmentation a solved task? Rethinking neural Chinese word segmentation. In *Proceedings of EMNLP*, pages 5676–5686.

Kai Fu, Jin Huang, and Yitao Duan. 2018. Youdao's winning solution to the NLPCC-2018 task 2 challenge: a neural machine translation approach to Chinese grammatical error correction. In *CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC)*, pages 341–350.

Roman Grundkiewicz, Christopher Bryant, and Mariano Felice. 2020. A crash course in automatic grammatical error correction. In *Proceedings of COLING: Tutorial Abstracts*, pages 33–38.

Charles Hinson, Hen-Hsen Huang, and Hsin-Hsi Chen. 2020. Heterogeneous recycle generation for Chinese grammatical error correction. In *Proceedings of COLING*, pages 2191–2201.

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 NAACL-HLT*, pages 595–606.

Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In *Proceedings of AACL*, pages 827–832.

Diederik P Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

Piji Li and Shuming Shi. 2021. Tail-to-tail non-autoregressive sequence prediction for Chinese grammatical error correction. In *Proceedings of ACL*, pages 4973–4984.

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 NLPTEA@AACL*, pages 57–66.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: a robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: high-precision text editing. In *Proceedings of EMNLP-IJCNLP*, pages 5054–5065.

Masato Mita, Tomoya Mizumoto, Masahiro Kaneko, Ryo Nagata, and Kentaro Inui. 2019. Cross-corpora evaluation and analysis of grammatical error correction models—is single-corpus evaluation enough? In *Proceedings of NAACL-HLT (Short)*, pages 1309–1314.

Ryo Nagata and Keisuke Sakaguchi. 2016. Phrase structure annotation and parsing for learner English. In *Proceedings of ACL*, pages 1837–1847.

Courtney Napoles, Maria Nădejde, and Joel Tetreault. 2019. Enabling robust grammatical error correction in new domains: data sets, metrics, and analyses. *TACL*, 7:551–566.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: a fluency corpus and benchmark for grammatical error correction. In *Proceedings of EACL*, pages 229–234.

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: Shared Task*, pages 1–14.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of CoNLL: Shared Task*, pages 1–12.

Jakub Náplava, Milan Straka, Jana Straková, and Alexandr Rosen. 2022. Czech grammar error correction with a large and diverse corpus. *TACL*, 10:452–467.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: tag, not rewrite. In *Proceedings of BEA@ACL*, pages 163–170.

Gaoqi Rao, Qi Gong, Baolin Zhang, and Endong Xun. 2018. Overview of NLPTEA-2018 share task Chinese grammatical error diagnosis. In *Proceedings of NLPTEA@ACL*, pages 42–51.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020. Overview of NLPTEA-2020 shared task for Chinese grammatical error diagnosis. In *Proceedings of NLPTEA@AACL*, pages 25–35.

Hongkai Ren, Liner Yang, and Endong Xun. 2018. A sequence to sequence learning for Chinese grammatical error correction. In *CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC)*, pages 401–410.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of ACL-IJCNLP*, pages 702–707.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: fluency instead of grammaticality. *TACL*, 4:169–182.

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*.

Oleksiy Syvokon and Olena Nahorna. 2021. UA-GEC: grammatical error correction and fluency corpus for the Ukrainian language. *arXiv preprint arXiv:2103.16997*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of ICCV*, pages 2818–2826.

Zecheng Tang, Yixin Ji, Yibo Zhao, and Junhui Li. 2021. Chinese grammatical error correction enhanced by data augmentation from word and character levels. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics (CCL) (in Chinese)*, pages 813–824.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008.

Chencheng Wang, Liner Yang, Yingying Wang, Yongping Du, and Erhong Yang. 2020. Chinese grammatical error correction method based on transformer enhanced architecture. *Journal of Chinese Information Processing (in Chinese)*, 34(6):106–114.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2019. StructBERT: incorporating language structures into pre-training for deep language understanding. In *Proceedings of ICLR*.

Yingying Wang, Cunliang Kong, Liner Yang, Yijun Wang, Xiaorong Lu, Renfen Hu, Shan He, Zhenghao Liu, Yun Chen, Erhong Yang, and Maosong Sun. 2022. YACLC: a Chinese learner corpus with multidimensional annotation. *arXiv preprint arXiv:2104.05507*.

Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. A comprehensive survey of grammatical error correction. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(5):1–51.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of ACL*, pages 180–189.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of NAACL-HLT*, pages 380–386.

Baolin Zhang. 2009. Features and functions of the HSK dynamic composition corpus (in Chinese). *International Chinese Language Education*, 4:71–79.

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 (NLPCC)*, pages 439–445.

Zewei Zhao and Houfeng Wang. 2020. MaskGEC: improving neural grammatical error correction via dynamic masking. In *Proceedings of AAAI*, pages 1226–1233.

Junpei Zhou, Chen Li, Hengyou Liu, Zuyi Bao, Guangwei Xu, and Linlin Li. 2018. Chinese grammatical error correction using statistical and neural models. In *CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC)*, pages 117–128.

Appendices

A More Discussion on the Char-based Span-level Metric

We find that ERRANT_ZH (Hinson et al., 2020), a useful evaluation tool for CGEC, only merges edits for redundant errors and missing errors, and does not merge edits for substitution errors. However, as discussed in Section 6.2, we also merge the consecutive edits for substitution errors. We hope future research can adopt our simplified version unless there is a strong reason. Meanwhile, we should keep thinking about which evaluation metrics are more suitable for CGEC task.

B Interface

Figure 4 shows our design of annotation interface in our annotation tool, where annotators correct assigned sentences. Given an annotation task, this interface presents a potentially wrong sentence and a text input box. The original sentence is copied into the input box below, so that the annotator can directly modify it. To support multiple corrections, we also provide a button to add additional input boxes. Two special buttons are provided to deal with special cases. The *error free* button means that the sentence is correct; the *not annotatable* button means that the annotator can not understand the sentence.

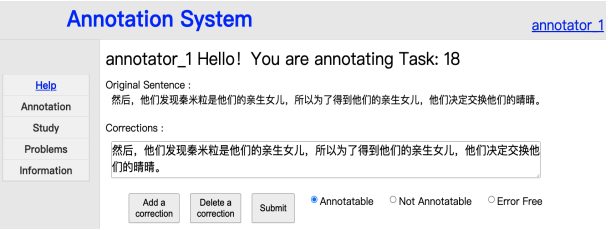

Figure 4: The screenshot of the annotation interface.

Figure 5 shows the review interface, where senior annotators judge whether the submitted corrections are correct. All corrections of a sentence from annotators are shown on the screen, and reviewers click a check box to mark each of them as correct or incorrect. The input box below allows reviewers to supplement extra valid corrections.

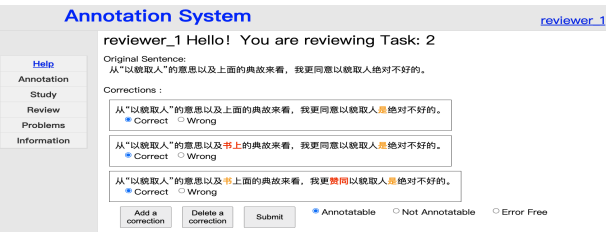

Figure 5: The screenshot of the review interface.

C Hyperparameters

Table 8 and Table 9 shows the detailed hyperparameters for training our two benchmark models. Due to the GPU memory limitation, we truncated sentences longer than 100 characters when training the Seq2Seq model. In other words, extra characters in the input sentences and the references are discarded.

A useful trick. We find that some sentences in MuCGEC are actually composed of multiple sentences. Therefore, we split one input sentence into multiple ones based on punctuation marks such as periods, exclamation marks, and so on. Then, we

perform corrections on the smaller sentences and concatenate the results. We find this trick can consistently improve performance in our preliminary experiments. For now, we decide not to break the sentences when releasing in order to be consistent with the sources where the data comes from.

Configurations	Values
Model architecture	BART (Lewis et al., 2020)
Pretrained model	Chinese-BART-Large (Shao et al., 2021)
Number of max epochs	10
Devices	8 Nvidia V100 GPU (32GB)
Batch size per GPU	32
Optimizer	Adam (Kingma and Ba, 2014) ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$)
Learning rate	3×10^{-6}
Learning rate scheduler	Polynomial
Gradient accumulation steps	4
Dropout	0.1
Gradient clipping	1.0
Loss function	Label smoothed cross entropy (label-smoothing=0.1) (Szegedy et al., 2016)
Total training time	About 20 hours
Stopping criteria	Loss value on the dev set
Patience	3

Table 8: Hyperparameter values of our Seq2Seq model.

Configurations	Values
Model architecture	GECToR (Omelianchuk et al., 2020)
Pretrained model	Chinese-Struct-Bert-Large (Wang et al., 2019)
Number of max epochs	20
Number of cold epochs	2
Devices	1 Nvidia V100 GPU (32GB)
Optimizer	Adam (Kingma and Ba, 2014) ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$)
Cold learning rate	1×10^{-3}
Learning rate	1×10^{-5}
Batch size	128
Loss function	Cross entropy
Total training time	About 10 hours
Stopping criteria	Label prediction accuracy on the dev set
Patience	3

Table 9: Hyperparameter values of our Seq2Edit model. “Cold” means that freeze the parameters of BERT.