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

Anonymous ACL submission

Abstract

This paper presents MuCGEC, a multi-reference multi-source evaluation dataset for Chinese Grammatical Error Correction (CGEC), consisting of 7,063 sentences from three different Chinese-as-a-Second-Language (CSL) learner sources. Each sentence has been corrected by three annotators, and their corrections are meticulously reviewed by an expert, resulting in 2.3 references on average per sentence. We conduct experiments with two mainstream CGEC models, i.e., the sequence-to-sequence (Seq2Seq) model and the sequence-to-edit (Seq2Edit) model, both enhanced with large pretrained language models, achieving competitive benchmark performance on previous and our datasets.We also discuss the CGEC evaluation methodologies, including the effect of multiple references and using a char-based metric. We will release our annotation guidelines, data, and code.

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, there has been increasing attention to GEC for its vital value in various downstream scenarios (Grundkiewicz et al., 2020; Wang et al., 2021).

To support the GEC study, 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). In contrast, such datasets for CGEC are relatively scarce. The two publicly available datasets are NLPCC18 and CGED, contributed by the NLPCC-2018 (Zhao et al., 2018) and CGED-2018&2020 shared tasks (Rao et al., 2018, 2020), respectively.

Most influential 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). Nevertheless, sentences in existing CGEC evaluation datasets always have only one reference (i.e., 87% of the sentences in NLPCC18 and all in CGED). This is possibly due to their adopted annotation workflow, where each sentence is assigned to only one annotator and multi-reference submission is not allowed. As strongly suggested by Bryant and Ng (2015), enforcing multi-reference annotation is crucial for both GEC model evaluation and data annotation. Because, obviously, there are usually multiple acceptable references with close meanings for an incorrect sentence, as illustrated by the example in Table 1. On the one hand, if the evaluation data gives only one reference and a GEC model outputs another valid alternative, then the model will be unfairly underestimated. To mitigate this phenomenon, a routine 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 taxing for the senior annotator to solely select the best one as the final golden answer.

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.

Besides the lack of multiple references, all existing CGEC datasets collect sentences from a single text source, which may be insufficient for robust model evaluation (Mita et al., 2019). Another flaw of them is the absence of strict quality control strate-

gies, e.g., annotation guidelines and review mechanisms. The above-mentioned problems may cause the unreliability of evaluation for CGEC models and hinder the development of this area.

To fill these gaps, this paper aims to build the first multi-reference multi-source evaluation dataset for CGEC. We first collect data for annotation from three divergent sources that cover both formal/informal texts. After investigating previous work on constructing GEC datasets, we compile comprehensive annotation guidelines for detailed illustration. Based on a specially constructed online annotation system, 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) We construct the first multi-reference multi-source evaluation dataset for CGEC, named MuCGEC, consisting of 7,063 sentences from three representative sources of CSL texts. Each sentence obtains 2.3 references on average. Further, we conduct detailed analyses on our new dataset to gain more insights.

(2) We employ two mainstream and competitive CGEC models based on large pretrained language models (PLMs), i.e., the Seq2Edit and Seq2Seq models, with an extremely effective ensemble strategy, to conduct strong benchmark experiments on our dataset. We also investigate the effect of multiple references and propose to use a char-based evaluation metric, which is simpler and more suitable than previous word-based ones for CGEC.

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

(3) Lang8[1] is a language learning platform, where native speakers voluntarily correct jottings uploaded by second-language learners. The NLPCC-2018 organizers collect about 717K Chinese sentence-correction pairs from Lang8 and employ them as the training data. We randomly select 2,000 potentially erroneous sentences with 30 to 60 characters for annotation.

Finally, we have obtained 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 suffers from 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 requires annotators to directly rewrite the whole sentence, as long as the resulting sentence does not change

[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.

the original meaning and is grammatically correct and fluent. Edits are extracted automatically from parallel sentences by additional tools (Bryant et al., 2017). This annotation paradigm has been proved to be efficient and cheap (Sakaguchi et al., 2016), and adopted by many datasets in other languages (Napoles et al., 2017, 2019; Syvokon and Nahorna, 2021). In this work, we adopt the *direct rewriting* paradigm. Besides the above-mentioned advantages, we believe it is beneficial for encouraging the variety of references since annotators can correct more freely.

2.3 Annotation Guidelines

After several months' survey on previous GEC data construction work, we have compiled 30-page comprehensive guidelines for CGEC annotation. During the annotation, our guidelines are gradually improved according to the feedback from our annotators.

To facilitate illustration, our guidelines adopt a two-tier hierarchical error taxonomy, including 5 major error types and 14 minor types, as shown in Table 2. The 5 major error types are decided by both referring to previous work and considering frequencies of error occurrences. Our guidelines describe in detail how to handle each minor error type and provide abundant typical examples. We will release our guidelines along with the data, which we hope can benefit future research.

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 for review. During annotation, an annotator may submit multiple references for one sentence if he/she thinks they are correct according to the guidelines. During the review, the job of the senior annotator includes 1) modifying incorrect references into correct ones (sometimes just rejecting them); 2) adding other correct references according to the guidelines. After review, the accepted references are defined as **Final Golden References**, which are ultimately used to evaluate CGEC models.

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, the annotator has to modify her/his submission by referring to the final golden references. Moreover, annotators can make complaints if they disapprove of the final golden references, which can trigger helpful discussions.

To improve annotation efficiency, we have developed a browser-based online annotation tool to support the above workflow and mechanisms. Due to the space limitation, we show the visual interfaces for annotation and review in Appendix A.

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 about our guidelines before the real annotation. In the beginning, two co-authors who were 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 is 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.

Dataset	#sent	#err. sent (perc.)	chars/sent	edits/ref	ref/sent
NLPCC18 (orig)	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 3: Data statistics, including sentence numbers, numbers (proportion) of erroneous sentences, averaged character numbers per sentence, averaged edit numbers per reference, and averaged reference numbers per sentence. Some sentences in our source data are thrown away since annotators cannot understand their meaning and thus are unable to correct them. Numbers in the parenthesis of the "#sent" row refer to such sentences.

3 Analysis of Our Annotated Data

This section presents a detailed analysis of the proposed MuCGEC dataset.

Overall statistics of our new dataset are shown in Table 3. We also include the original NLPCC18 dataset (Zhao et al., 2018) for comparison.[2]

First, regarding the proportion of erroneous sentences, most of the sentences are considered to contain grammatical errors in the previous 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 is the shortest, whereas CGED is much longer. This is possibly because, since HSK is an official Chinese proficiency test, candidates tend to use long sentences to show their ability in Chinese use.

Third, each sentence in re-annotated NLPCC18 receives 2.5 references on average, which is more than twice of 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 model evaluation, which is further discussed in Section 6.2.

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 edit number is tightly correlated with the sentence length. The difference in averaged sentence length and numbers 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.

Distribution regarding numbers of references. In Figure 1, we analyze the distribution of sentences regarding the numbers of references. Here, we only consider erroneous sentences. Same references from different annotators are calculated as one reference. Overall, most sentences have 2 references, and sentences having 3 references take a slightly lower proportion. There are 21.8% of sentences with only one reference. Most of them are short and easy to correct.

We believe that the average reference number should be further increased if more annotators are assigned for each sentence. Despite the fact that our annotation tool allows annotators to submit multiple answers, we find that most annotators tend to submit a single correction. Since it is usually easy to come up with the most suitable correction, but is more time-consuming to provide alternatives.

Human annotation performance. In order to know the annotation ability of our annotators and human performance for the CGEC task, we calculate char-based $F_{0.5}$ scores by evaluating the annotation submissions against the final golden references picked by senior annotators after review. 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 can be further improved if our annotators are more experienced and more familiar with our guidelines.

Figure 2 shows $F_{0.5}$ scores of 15 annotators who annotated the most sentences, in the descending

[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 collected from the internet, and its correction is quite noisy.

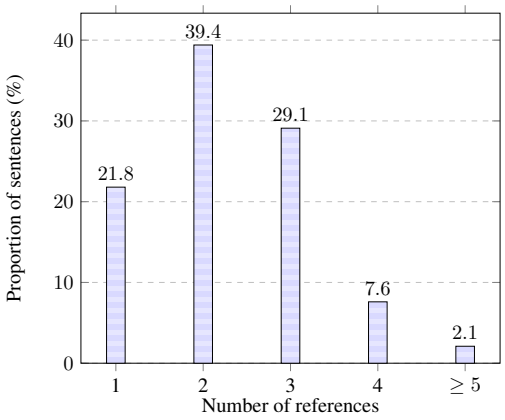

Figure 1: The proportion of sentences with the different number of references in MuCGEC.

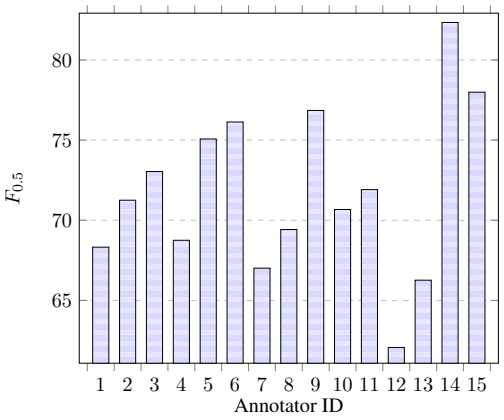

Figure 2: The human performance of the 15 annotators who annotate the most sentences.

order of annotated sentence numbers. 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. It indicates that we should pay more attention to annotation quality when calculating salaries and prevent annotators from focusing too much on annotation speed.

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 attempt 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 B 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 4).

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

The Seq2Seq model straightforwardly treats GEC as a monolingual translation task (Yuan and Briscoe, 2016). Recent work proposes 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 work and utilize the recently proposed Chinese BART from Shao et al. (2021) to initialize our Seq2Seq model.

The ensemble model. Several previous works have proved 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 6), and thus attempt to combine them. We adopt a simple edit-wise vote mechanism: aggregating edits from the results of each model, and only preserving edits that appear more than $N/2$ times, where N is the number of models. We experiment with two ensemble settings: 1) one Seq2Edit and one Seq2Seq, denoted as "1×Seq2Edit+1×Seq2Seq" , and 2) three Seq2Edit and three Seq2Seq, denoted as "3×Seq2Edit+3×Seq2Seq". The three Seq2Edit models are obtained by replacing the random seed, the same goes for the Seq2Seq.

5 Experiments on Original NLPCC18

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

	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 4: 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*.

Training data. For the sake of easy replicability, we limit our training data strictly to public resources, i.e., the Lang8 (Zhao et al., 2018) [4] data and the HSK (Xun, 2018) [5] data. We filter duplicate sentences that appear in our dataset, and only use the erroneous part for training. The final Lang8 and HSK data contains 1,092,285 and 95,320 sentences, respectively. The HSK data is cleaner and of higher quality than Lang8, but is much 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 work. Table 4 shows the results. For a fair comparison, we follow the official setting of the shared task, including using the word-based MaxMatch scorer (Dahlmeier and Ng, 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 using only Lang8 for training, our single Seq2Seq model is already quite competitive. It only underperforms MaskGEC (Zhao and Wang, 2020) by 1 $F_{0.5}$ score, which additionally uses data augmentation. After adding the HSK data, all our models achieve further performance-boosting by about 4 points. Both of our single benchmark models achieve SOTA performance under this 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.

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 [6] website. All PLMs are under the large configuration.

6 Experiments on MuCGEC

6.1 Data Splits

For hyperparameter tuning or model selection, previous work 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 problematic for replicability and complicated for comparison.

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 the word-based metric. As discussed in Section 5, previous CGEC datasets are annotated upon word sequences and thus adopt word-based metrics for evaluation. Therefore, before annotation and evaluation, each sentence should be segmented into words using a Chinese word segmentation (CWS) model. This introduces unnecessary uncertainty in the evaluation procedure. Sometimes, a correct edit may be judged as wrong due to word boundary mismatch. Different from English whose words are separated naturally, Chinese sentences are written without any word delimiters, so WS models perform much worse for Chinese than for English (Fu et al., 2020).

In view of these, we believe it is more suitable to abandon such word-based metrics in CGEC.

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

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

[6] 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 5: 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 results.

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

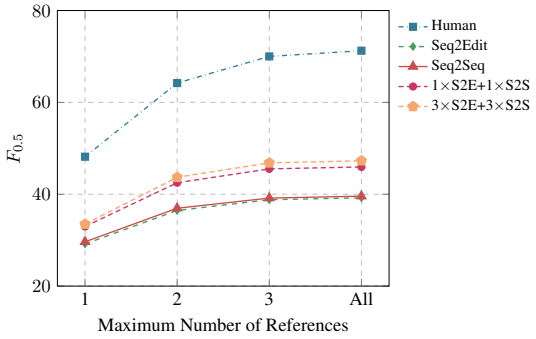

Figure 3: Effect of reference number on $F_{0.5}$.

The char-based evaluation metric is adopted in this work instead. First, given an input sentence and a correction, we obtain an optimal sequence of char-based edits that corresponds to the minimal edit distance. There are three types of char-based edits, including deleting a char for a *redundant error*, inserting a char for a *missing error*, or substituting a char with another one for a *substitution error*. Secondly, following the standard practice in both EGEC and CGEC, we merge consecutive edits of the same type as one span-level edit (Felice et al., 2016; Hinson et al., 2020). The above two steps are applied to both the system output sequence and all gold-standard references, transforming them into sets of span-level edits. Finally, we utilize the evaluation script from ERRANT (Bryant et al., 2017) to calculate the P/R/F values which does not need error-coded annotation.

6.3 Results and Analysis

Main results. Table 5 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. Please kindly note that we cannot present results of previous work in Table 4, since most of them did not release their code.[7]

The overall trend of performance is basically consistent with those on the original NLPCC18 dataset in Table 4. 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. Secondly, the model ensemble approach improves performance by a very large margin, as expected.

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 the CGEC research still has a long way to go.

Performance on four error types. Table 6 shows more fine-grained evaluation results on four error types. The word-order errors can be identified by heuristic rules following Hinson et al. (2020).

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

[7] BLCU (Ren et al., 2018) did release code, but its performance is much lower than the SOTA.

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 utilizing 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 reference number in our dataset. For example, when the maximum number of references is limited to 2, we remove all extra references if a sentence has more than 2 gold-standard references. The results on MuCGEC 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 slight 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 Work

EGEC resources. There is 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 for each original sentence additionally provides the language proficiency level of the writer. All four datasets are composed of essays from non-native English speakers and provide multiple references.

Recently, researchers start 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 thoroughly 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., Lang8 corpus (Zhao et al., 2018) and HSK corpus (Xun, 2018).

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; Fang et al., 2020). 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 also been proved very useful for CGEC.

8 Conclusions

This paper presents our 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 different text sources; 3) adopting more 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, which are more suitable than word-based ones. In summary, we believe this work will promote future research in CGEC.

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.

Meiyuan Fang, Kai Fu, Jiping Wang, Yang Liu, Jin Huang, and Yitao Duan. 2020. A hybrid system for nlptea-2020 cged shared task. In *Proceedings of NLPTEA@AACL*, pages 67–77.

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.

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. Uagec: 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)*, 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*, 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*.

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.

Endong Xun. 2018. Hsk dynamic composition corpus. http://hsk.blcu.edu.cn/.

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.

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.

A Interface

The annotation interface is shown in Figure 4. Annotators can use it to 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 text input box, and the annotator can directly modify it. Considering the existence of multiple acceptable corrections, we also provide a button to allow annotators to add additional text input boxes. For exceptional cases, some special tags can be used, such as *ERROR FREE* and *NOT ANNOTATABLE*.

The review interface is shown in Figure 5. It is used by expert annotators to judge whether a correction is acceptable. All corrections of a sentence will be shown on the screen, and reviewers can click a check box to mark each of them as correct or false. The text input box in the annotation interface is also available here, thus allowing reviewers to supplement extra valid corrections.

B Hyperparameters

Table 7 shows the detailed hyperparameters for training our two benchmark models. The results of all single models are averaged over 3 runs, and the results of the ensemble models are just calculated from a single run.

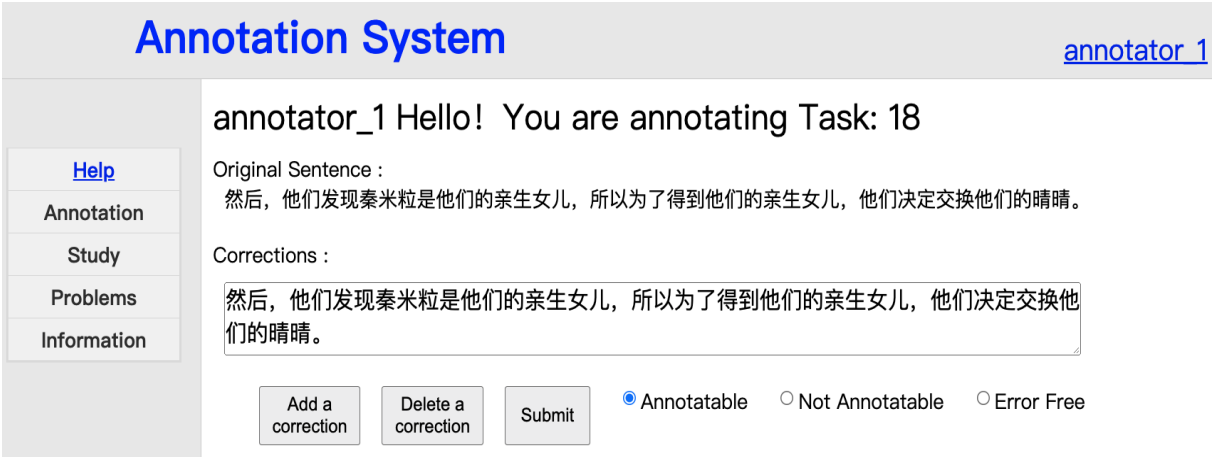

Figure 4: The screenshot of the annotation interface.

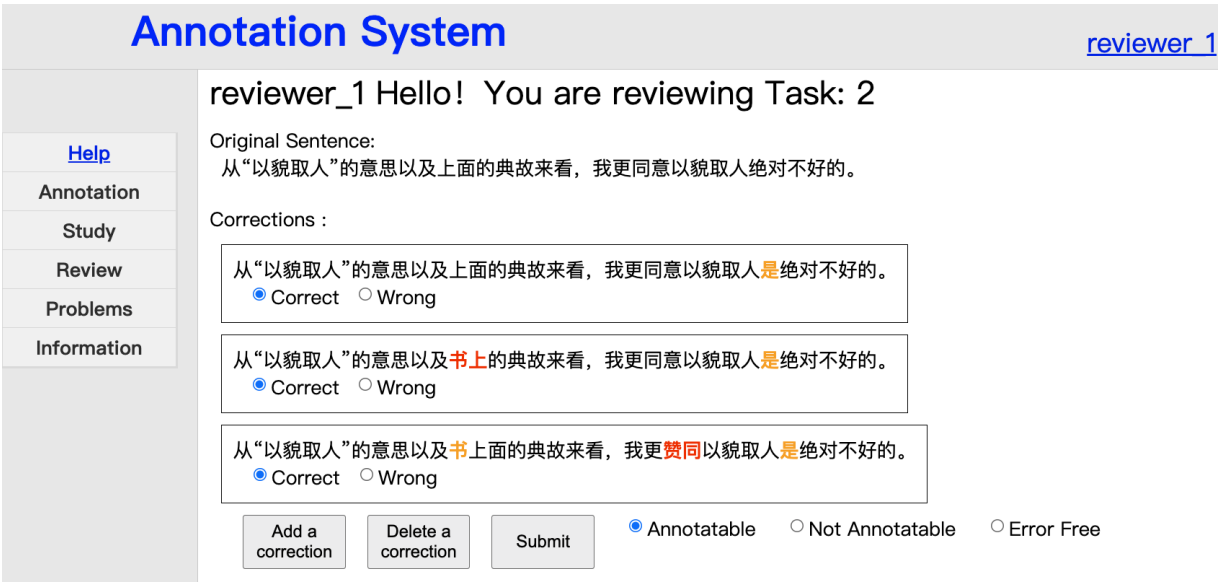

Figure 5: The screenshot of the review interface.

Configurations	Values
Seq2Seq	
Model architecture	BART (Lewis et al., 2020)
Pretrained model	Chinese-BART-Large (Shao et al., 2021)
Number of epochs	20
Devices	8 Nvidia V100 GPU (32GB)
Batch size per GPU	32
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$) (Kingma and Ba, 2014)
Learning rate	3×10^{-5}
Learning rate scheduler	Polynomial
Gradient accumulation steps	4
Warmup updates	1000
Warmup init learning rate	1×10^{-7}
Dropout	0.3
Gradient clipping	1.0
Loss function	Label smoothed cross entropy (label-smoothing=0.1) (Szegedy et al., 2016)
Beam size	12
GPU hours	About 20 hours
Seq2Edit	
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 ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$) (Kingma and Ba, 2014)
Cold learning rate	1×10^{-3}
Learning rate	1×10^{-5}
Batch size	128
Loss function	Cross entropy
GPU hours	About 10 hours

Table 7: Hyperparameter values of our benchmark models.