

A Self-Refinement Strategy for Noise Reduction in Grammatical Error Correction

Masato Mita^{1,2} Shun Kiyono^{1,2} Masahiro Kaneko^{3,1} Jun Suzuki^{2,1} Kentaro Inui^{2,1}

¹RIKEN Center for Advanced Intelligence Project

²Tohoku University

³Tokyo Metropolitan University

{masato.mita, shun.kiyono}@riken.jp

kaneko-masahiro@ed.tmu.ac.jp

{jun.suzuki, inui}@ecei.tohoku.ac.jp

Abstract

Existing approaches for grammatical error correction (GEC) largely rely on supervised learning with manually created GEC datasets. However, there has been little focus on verifying and ensuring the quality of the datasets, and on how lower-quality data might affect GEC performance. We indeed found that there is a non-negligible amount of “noise” where errors were inappropriately edited or left uncorrected. To address this, we designed a self-refinement method where the key idea is to denoise these datasets by leveraging the prediction consistency of existing models, and outperformed strong denoising baseline methods. We further applied task-specific techniques and achieved state-of-the-art performance on the CoNLL-2014, JFLEG, and BEA-2019 benchmarks. We then analyzed the effect of the proposed denoising method, and found that our approach leads to improved coverage of corrections and facilitated fluency edits which are reflected in higher recall and overall performance.

1 Introduction

Grammatical error correction (GEC) is often considered a variant of machine translation (MT) (Brockett et al., 2006; Junczys-Dowmunt et al., 2018) due to their structural similarity—“translating” from source ungrammatical text to target grammatical text. At present, several neural encoder-decoder (EncDec) approaches have been introduced for this task and have achieved remarkable results (Chollampatt and Ng, 2018; Zhao et al., 2019; Kiyono et al., 2019). EncDec models tend to further improve in performance with increasing data size (Koehn and Knowles, 2017; Sennrich and Zhang, 2019), however, this is not necessarily true in GEC. For example, Lo et al. (2018) reported that an EncDec-based GEC model trained on EF-

1 : Errors are inappropriately edited

Source: *I want to **discuss about** the education.*

Target: *I want to **discuss of** the education.*

2 : Errors are left uncorrected

Source: *We **discuss about** our sales target.*

Target: *We **discuss about** our sales target.*

Table 1: Example of an inappropriately corrected error and an unchanged error in EFCamDat. We consider these types of errors to be dataset noise that might hinder GEC model performance.

CamDat (Geertzen et al., 2013)¹, the largest publicly available learner corpus as of today (two million sentence pairs), was outperformed by a model trained on a smaller dataset (e.g., 720K pairs). They hypothesized that this may be due to the noisiness of EFCamDat, i.e., the presence of sentence pairs whose correction still contained grammatical errors due to inappropriate edits or being left uncorrected. For example, in Table 1, “*discuss about*” should most likely have been corrected to “*discuss*”, and “*are discussing*”, respectively. We confirmed that there is a non-negligible amount of noise in commonly used GEC datasets (Section 3).

We recognise data noise as a generally overlooked issue in GEC, and consider the question of *whether a better GEC model can be built by reducing noise in GEC corpora*. To this end, we designed a *self-refining* approach—an effective denoising method where residual errors left by careless or unskilled annotators are corrected by an existing GEC model. This approach relies on the consistence of the GEC model’s predictions (Section 4).

We evaluated the effectiveness of our method over several GEC datasets, and found that it considerably outperformed baseline methods, includ-

¹https://corpus.mml.cam.ac.uk/efcamdat2/public_html/

ing three strong denoising baselines based on a filtering approach, which is a common approach in MT (Bei et al., 2018; Junczys-Dowmunt, 2018; Rossenbach et al., 2018). We further improved the performance by applying task-specific techniques and achieved state-of-the-art performance on the CoNLL-2014, JFLEG, and BEA-2019 benchmarks. Finally, through our analysis, we found unexpected benefits to our approach: (i) the approach benefits from the advantage of self-training in neural sequence generation due to its structural similarity (Section 6.3), (ii) resulted in significant increase in recall while maintaining equal precision, indicating improved coverage of correction (Section 6.4), and (iii) there seems to be a tendency for more fluent edits, possibly leading to more native-sounding corrections (Section 6.5). The last is reflected in performance on the JFLEG benchmark, which focuses on fluency edits.

In summary, we present a data denoising method which improves GEC performance, verify its effectiveness by comparing to both strong baselines and current best-performing models, and analyze how the method affects both GEC performance and the data itself.

2 Related Work

In GEC, previous studies have generally focused on typical errors, such as the use of articles (Han et al., 2006), prepositions (Felice and Pulman, 2008), and noun numbers (Nagata et al., 2006). More recently, many studies have addressed GEC as a MT problem where ungrammatical text is expected to be *translated* into grammatical text. This approach allows the adoption of sophisticated sequence-to-sequence architectures (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) that have achieved strong performance but require a large amount of data (Chollampatt and Ng, 2018; Junczys-Dowmunt et al., 2018; Kiyono et al., 2019). In GEC, the data are usually manually built by experts, which lead to an underlying assumption that the data is noise-free. Therefore, to the best of our knowledge, noise in existing common datasets remains largely under-explored and no previous research has investigated the effectiveness of denoising GEC datasets. Recently, Lichtarge et al. (2020) proposed a method for filtering large and noisy synthetic pretrained data in GEC by deriving example-level scores on their pretrained data.²

²Lichtarge et al. (2020) has appeared after our submission.

However, what they regard as noise consists of instances in source sentences (i.e., not target sentences) of the synthetic data that are outside the genuine learner error distribution, where they perform data selection based on the small and higher-quality genuine data (namely, the learner corpora we attempt to denoise in this study). Therefore, our methods are not comparable, and it is expected to further improve the performance by combining both methods, which we plan to investigate in our future work.

In contrast, data noise is becoming an increasingly important topic in MT, where it is common to use automatically acquired parallel data via web crawling in addition to high-quality curated data. As a result, the MT field faces various data quality issues such as misalignment and incorrect translations, which may significantly impact translation quality (Khayrallah and Koehn, 2018). A straightforward solution is to apply a filtering approach, where noisy data are filtered out and a smaller subset of high-quality sentence pairs is retained (Bei et al., 2018; Junczys-Dowmunt, 2018; Rossenbach et al., 2018). Nevertheless, it is unclear whether such a filtering approach can be successfully applied to GEC, where commonly available datasets tend to be far smaller than those used in recent neural MT research. Hence, in this study, we investigate its effectiveness by conducting a comparative experiment using the proposed denoising approach.

3 Noise in GEC Datasets

In this study, we define *noise* as two types of residual grammatical errors in target sentences: inappropriate edits and those left uncorrected (Table 1). Most learner corpora, such as EFCamDat and Lang-8 (Mizumoto et al., 2011; Tajiri et al., 2012), are constructed based on correction logs in which the source texts are provided by human language learners and the corresponding corrected target texts are provided by editor (annotators). Unless each annotator has 100% accuracy, all corpora inevitably contain noise.

The presence of noise in GEC data was uncovered by previous work such as Lo et al. (2018), but the exact nature of it was unexplored. To confirm this, we manually assessed how much noise was contained in the following three commonly used training datasets: the BEA official training dataset (henceforth, *BEA-train*) provided in the

(1)	BEA-train
\mathbf{X} :	<i>I will make a poet to kill this pain.</i>
\mathbf{Y} :	<i>I will make a poem to kill this pain.</i>
\mathbf{Y}' :	<i>I will write a poem to get rid of this pain.</i>
(2)	EFCamDat
\mathbf{X} :	<i>The restaurant in front of movie teather.</i>
\mathbf{Y} :	<i>The restaurant in front of movie theater.</i>
\mathbf{Y}' :	<i>The restaurant is located opposite the movie theater.</i>
(3)	Lang-8
\mathbf{X} :	<i>Coordinate with product support team for potential customer show site visit ;</i>
\mathbf{Y} :	<i>Coordinate with product support team for potential customer show site visits;</i>
\mathbf{Y}' :	<i>Please coordinate with the product support team to escort potential customers to site visits.</i>

Table 2: Examples of original sources sentences (\mathbf{X}), original target sentences (\mathbf{Y}) and target sentences reviewed by the expert (\mathbf{Y}') in the most commonly used training data for GEC.

Dataset	WER (%noise)
BEA-train	37.1
EFCamDat	42.1
Lang-8	34.6

Table 3: Amount of noise in GEC training data estimated by WER.

BEA-2019 workshop (Bryant et al., 2019)³, EFCamDat, and the non-public Lang-8 corpus (henceforth, *Lang-8*)⁴. For 300 target sentences \mathbf{Y} from each dataset, one expert reviewed them and we obtained denoised ones \mathbf{Y}' (Table 2). We then calculated the word edit rate (WER) between the original target sentences \mathbf{Y} and the denoised target sentences \mathbf{Y}' . WER is defined as follows:

$$\text{WER} = \frac{\sum_{i=1}^N d(\mathbf{Y}_i, \mathbf{Y}'_i)}{\sum_{i=1}^N |\mathbf{Y}_i|} \quad (1)$$

where, $|\mathbf{Y}_i|$ is the total number of words in each original target sentences \mathbf{Y}_i and $d(\cdot)$ is the word-based Levenshtein distance. Table 3 shows the amount of noise in the datasets estimated by WER. Here, the WER values are slightly higher than expected, but this is most likely caused by fluency edits by the editor, making the sentence more native-like. Thus, we found that (i) there is a non-negligible amount of “noise” in the most commonly used training data for GEC, and (ii) EFCamDat is much noisier than the other two training datasets.

³See Appendix A for dataset details.

⁴A corpus consisting of correction logs from 2012 to 2019 in Lang-8. We recognize that BEA-train contains a subset of this corpus, but we use it without distinction in this study.

4 Proposed Denoising Method

The supervised learning problem for GEC is formally defined as follows. Let θ be all trainable parameters of a GEC model, and \mathcal{D} be training data consisting of pairs of an ungrammatical source sentence \mathbf{X} and a grammatical target sentence \mathbf{Y} , i.e., $\mathcal{D} = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^n$. Then, the objective is to find the optimal parameters $\hat{\theta}$ that minimize the following loss function $\mathcal{L}(\mathcal{D}, \theta)$ on training data \mathcal{D} :

$$\mathcal{L}(\mathcal{D}, \theta) = -\frac{1}{|\mathcal{D}|} \sum_{(\mathbf{X}, \mathbf{Y}) \in \mathcal{D}} \log(p(\mathbf{Y}|\mathbf{X}, \theta)). \quad (2)$$

Conventionally, training data \mathcal{D} is assumed to be “clean” parallel data. However, as argued in Section 3, this assumption typically does not hold in GEC. Here, we assume that training data \mathcal{D} is “noisy”, and, for clarity, we use the notation $\hat{\mathcal{D}}$ to represent “clean” parallel data, where “clean” means “denoised” in this context. The goal is, first, to obtain a new set $\hat{\mathcal{D}}$ by denoising \mathcal{D} , and then, to obtain a GEC model $\hat{\theta}$ on the new training data $\hat{\mathcal{D}}$.

To deal with data noise, a straightforward solution is to apply a filtering approach, where noisy data are filtered out and a smaller subset of high-quality sentence pairs is retained, as employed in MT. However, applying a filtering approach may not be the best choice in GEC for two reasons: (i) GEC is a low-resource task compared to MT, thus further reducing data size by filtering may be critically ineffective; (ii) Even noisy instances may still be useful for training since they might contain

Algorithm 1: Denoising GEC parallel data with self-refinement

Data: Noisy Parallel Data \mathcal{D} **Result:** Denoised Parallel Data $\hat{\mathcal{D}}$

```
1  $\hat{\mathcal{D}} = \{\}$  // create empty set
2 Train a base model and acquire  $\theta$  from  $\mathcal{D}$ 
3 for  $(X, Y) \in \mathcal{D}$  do
4    $Y' = \text{Beam\_Search\_Decoding}(Y; \theta)$ 
5   Compute perplexity  $\text{PPL}(Y)$  and  $\text{PPL}(Y')$ 
6   if  $\text{PPL}(Y) - \text{PPL}(Y') \geq 0$  then
7      $\hat{Y} = Y'$ 
8   else
9      $\hat{Y} = Y$ 
10   $\hat{\mathcal{D}} = \hat{\mathcal{D}} \cup \{(X, \hat{Y})\}$ 
11 Train a denoised new model  $\hat{\theta}$  from  $\hat{\mathcal{D}}$ 
```

some correct edits as well (Note that these correct edits would have also been lost to filtering, further decreasing the amount of informative cues in training).

As an alternative to filtering, we propose a self-refinement (SR) approach for denoising GEC training data (Algorithm 1). The main idea is to train a GEC model (henceforth, *base model*) on noisy parallel data \mathcal{D} and to use it for refining target sentences in \mathcal{D} . Noisy annotations are potentially caused by carelessness or insufficient skills of annotators. This causes inconsistent corrections in similar context. In contrast, machine learning-based GEC models, such as EncDec, tend to be reliably consistent given similar contexts. Given noisy parallel data $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^n$, we generate new target sentences \hat{Y}_i from the original target sentences Y_i and pair them with their original source sentences X_i (line 4 in Algorithm 1). The consistency of the base model predictions ensures that the resulting parallel data $\hat{\mathcal{D}} = \{(X_i, \hat{Y}_i)\}_{i=1}^n$ contain noise to a less extent. It is worth noting that SR can be regarded as a variant of self-training due to its structural similarity, except that it takes the target sentences rather than the source sentences as input to the model. The algorithm itself is the key difference from existing methods based on self-training (Wang, 2019; Nie et al., 2019; Xie et al., 2020).

One challenge of this approach is that the base model may consistently make inaccurate corrections. We thus incorporate a fail-safe mechanism

as a sub-component to restore the original target sentence if the GEC model makes incorrect corrections (lines 5-9). For example, in cases such as in Table 1, the base model may predict every instance as “discuss about”. In this step, to determine whether to accept the output Y' of the base model as a new target sentence, we compare the perplexity of the model output $\text{PPL}(Y')$ with that of the original target sentence $\text{PPL}(Y)$. Language models are trained on native-written corpora, meaning they can reasonably be assumed to contain information needed to estimate grammaticality. We believe that a measure of perplexity is a straightforward approach to exploit this information.

5 Experiments

We evaluate the proposed method in two ways. First, we exclusively focus on investigating the effectiveness of the proposed denoising method (Section 5.3). Then, we compare our strongest model trained with denoised data (henceforth, *denoised model*), with current best-performing ones to investigate whether the proposed method has a complementary effect on existing task-specific techniques (Section 5.4).

5.1 Configurations

Dataset For the training dataset, we used the same datasets as mentioned in Section 3: BEA-train, EFCamDat, and Lang-8. In addition, we used the BEA official validation set (henceforth, *BEA-valid*) provided in the BEA-2019 workshop as validation data. The characteristics of the datasets are summarized in Table 4. For preprocessing, we tokenized the training data using the spaCy tokenizer⁵. Then, we removed sentence pairs where both sentences were identical or both longer than 80 tokens. Finally, we acquired subwords from the target sentence via the byte-pair-encoding (BPE) (Sennrich et al., 2016b) algorithm. We used the subword-nmt implementation⁶ and then applied BPE to splitting both source and target texts. The number of merge operations was set to 8,000.

Evaluation To investigate the effectiveness of the proposed method, we followed the work by Mita et al. (2019) and evaluated the performance of the GEC models across various GEC datasets in terms of the same evaluation metrics. We report

⁵<https://spacy.io/>

⁶<https://github.com/rsennrich/subword-nmt>

Dataset	#sent (pairs)	Split	Scorer
BEA-train	561,100	train	-
EFCamDAT	2,269,595	train	-
Lang-8	5,689,213	train	-
BEA-valid	2,377	valid	-
CoNLL-2014	1,312	test	M ² scorer & GLEU
JFLEG	747	test	M ² scorer & GLEU
BEA-test	4,477	test	ERRANT

Table 4: Summary of datasets used in our experiments.

the results measured by both M² scorer (Dahlmeier and Ng, 2012)⁷ and GLEU metric (Napoles et al., 2015, 2016)⁸ on both the CoNLL-2014 test set and the JFLEG test set (Napoles et al., 2017). All reported results (except those corresponding to the ensemble models) are the average of three distinct trials using three different random seeds. Let us emphasize that our focus is on denoising the *training* data, and denoising the test data is out of the scope of this study. The commonly used test data, such as CoNLL-2014 and JFLEG, have multiple references which can lower the noise factor. In addition to having multiple references, both JFLEG and CoNLL-2014 have been specifically constructed for GEC evaluation, while the training data (Lang-8 and EFCamDat) are more of an organic collection of learner and editor interactions. Naturally, we believe it is reasonable to assume that the test data are considerably cleaner.

Model We employed the “Transformer (big)” settings Vaswani et al. (2017) using the implementation in the fairseq toolkit (Ott et al., 2019). Details on the hyper-parameters are listed in Appendix B. As a language model for the fail-safe mechanism, we used the PyTorch implementation of GPT-2 (Radford et al., 2019)⁹. Note that to avoid a preference for shorter phrases, we normalized the perplexity by sentence length.

5.2 Baselines

As argued in Section 4, we hypothesized that the filtering-based denoising approaches are not well-suited for GEC. To verify this hypothesis, we employed the following three filtering-based denoising baseline methods in addition to a base model trained in noisy parallel data \mathcal{D} (henceforth, *no denoising*).

⁷<https://github.com/nusnlp/m2scorer/releases>

⁸<https://github.com/cnap/gec-ranking>

⁹<https://github.com/huggingface/transformers>

Cross-entropy filtering (CE filtering) The dual conditional cross-entropy filtering method was proposed by Junczys-Dowmunt (2018) and achieved the highest performance on the noisy parallel corpus filtering task at WMT2018 (Koehn et al., 2018). In this study, we prepared forward and reverse pre-train models using the BEA-train dataset to adapt the filtering method to GEC. We obtained the filtered data by removing 20% of the sentence pairs¹⁰ with higher scores from the training data and used them for training.

Sentence-level error detection filtering (SED filtering) Asano et al. (2019) demonstrated the effectiveness of the sentence-level error detection (SED) model as a filtering tool to preprocess GEC input. Considering these findings, we adopted SED as a filtering-based denoising method for training data. More specifically, we discarded the source-target sentence pairs in the noisy parallel data \mathcal{D} if the SED model predicted the target sentence as an incorrect one. Following Asano et al. (2019), we obtained binary-labeled data using the BEA-train dataset to prepare a training set for the SED model, and then fine-tuned BERT (Devlin et al., 2019) on the prepared data.

Language model filtering (LM filtering) Language model-based filtering is a method based on the hypothesis that if the perplexity of a target sentence is larger than that of the source sentence, the target sentence is more likely to contain noise. LM filtering has the same motivation as the one underlying the fail-safe mechanism. We used GPT-2 as the pre-trained language model.

5.3 Results

Table 5 shows the results of the main experiment. The experimental results show that SR significantly outperformed the others, including the three strong denoising baseline models on the multiple datasets. Applying SR to EFCamDat, for instance, yielded a larger performance improvement than without denoising (e.g. no denoising $\mathbf{EF} = 40.3$ vs SR $\mathbf{EF} = 48.4$ in CoNLL-2014 when using $F_{0.5}$). Notably, we observed a similar trend when using both BEA-train and Lang-8 datasets as the training data, which indicated that SR was potentially effective for any corpora, not being limited to EFCamDat.

Furthermore, we compared the effectiveness of SR to other denoising methods. The filtering-based

¹⁰The value is derived from a preliminary experiment (Appendix C).

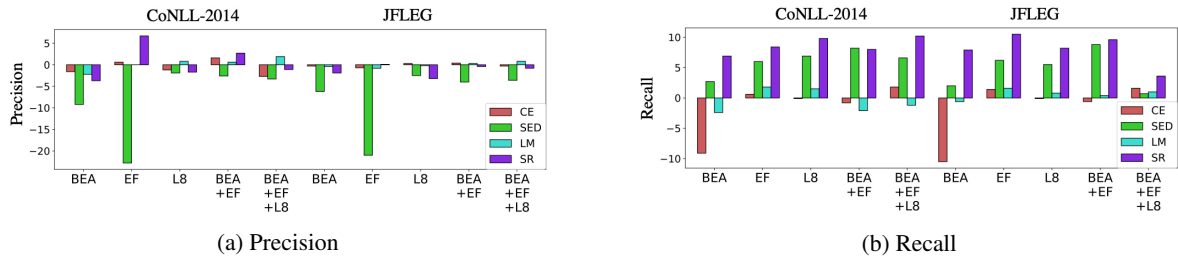


Figure 1: Increases and decreases in precision and recall for denoising methods when no denoising is set to 0.

Model	CoNLL-2014		JFLEG	
	F _{0.5}	GLEU	F _{0.5}	GLEU
BEA-train (BEA):				
No denoising BEA	49.6	63.3	58.7	52.3
CE filtering BEA	42.9	61.0	52.7	49.0
SED filtering BEA	45.1	62.7	55.6	52.2
LM filtering BEA	47.1	63.0	58.3	52.6
SR BEA (Ours)	50.3	64.2	60.5	54.8
EFCamDAT (EF):				
No denoising EF	40.3	61.3	59.5	53.7
CE filtering EF	40.9	61.5	59.8	54.2
SED filtering EF	26.5	54.0	47.5	49.7
LM filtering EF	41.2	61.7	59.7	54.2
SR EF (Ours)	48.4	63.5	63.9	57.1
Lang-8 (L8):				
No denoising L8	54.9	65.9	68.4	58.1
CE filtering L8	54.1	65.3	68.6	58.2
SED filtering L8	55.7	67.1	68.5	60.7
LM filtering L8	55.9	66.3	68.6	59.1
SR L8 (Ours)	56.5	67.7	68.6	61.0
BEA+EF:				
No denoising BEA+EF	49.1	63.4	62.0	53.9
CE filtering BEA+EF	49.6	63.3	61.9	54.5
SED filtering BEA+EF	51.2	64.9	62.8	56.7
LM filtering BEA+EF	48.3	63.3	62.3	54.7
SR BEA+EF (Ours)	54.5	65.2	65.5	58.0
BEA+EF+L8:				
No denoising BEA+EF+L8	56.1	65.7	67.0	56.9
CE filtering BEA+EF+L8	55.0	66.0	68.6	58.2
SED filtering BEA+EF+L8	56.1	67.3	67.7	60.3
LM filtering BEA+EF+L8	56.7	65.9	68.0	57.8
SR BEA+EF+L8 (Ours)	58.8	68.0	70.6	61.4

Table 5: Result of denoising experiments with cross-corpora evaluation: a **bold** value indicates the best result in each training data.

methods, such as SED and LM filtering, generally achieved better results compared to the baseline models; however, they resulted in lower performance in smaller datasets such as BEA. This could be caused by the fact that these filtering methods have filtered out the training instances containing not only noise but also many correct corrections that may still be partially useful for training. As shown in Table 6, we analyzed the size of each training dataset after filtering.

Figure 1 shows the increases and decreases in precision and recall when the performance without denoising is set as 0. The experimental results show that there was a certain pattern underlying the denoising effect. More specifically, reducing the noise by SR has little impact on the precision, but it has significantly improved the recall, indicating improved coverage of correction. We provide the detailed analysis on this question in Section 6.4.

5.4 Comparison with Existing Models

In the second experiment, we compared our best denoised model with the current best performing models to investigate whether SR works well with existing task-specific techniques. We incorporated task-specific techniques that have been widely used in shared tasks such as BEA-2019 and WMT-2019¹¹ into the proposed denoised model to further improve the performance. Concerning the task-specific techniques, we followed the work reported by Kiyono et al. (2019), as detailed below.

Pre-training with pseudo data (PRET) Kiyono et al. (2019) investigated the applicability of incorporating pseudo data into the model and confirmed the reliability of their proposed settings by showing acceptable performance on several datasets. We trained the proposed model using their pre-trained model “PRETLARGE+SSE” settings¹².

Right-to-left re-ranking (R2L) R2L is a common approach used to improve model performance by re-ranking using right-to-left models trained in the reverse direction (Sennrich et al., 2016a, 2017) in MT. More recently, previous studies confirmed the effectiveness of this approach when applied to GEC (Ge et al., 2018; Grundkiewicz et al., 2019). We adapted R2L to the proposed model. Specifically, we generated n -best hypotheses using an

¹¹<http://www.statmt.org/wmt19/>

¹²<https://github.com/butsugiri/gec-pseudodata>

Filtering method	BEA-train	EFCamDat	Lang-8
CE filtering	448,880 (∇ 20.0%)	1,815,676 (∇ 20.0%)	4,551,370 (∇ 20.0%)
SED filtering	317,957 (∇ 43.3%)	1,250,744 (∇ 44.9%)	3,314,440 (∇ 41.7%)
LM filtering	456,347 (∇ 18.7%)	1,936,238 (∇ 14.7%)	4,651,085 (∇ 18.2%)

Table 6: The size of the filtered data. The numbers in parentheses indicate each reduction rates.

Model	CoNLL-2014		JFLEG		BEA
	F _{0.5}	GLEU	F _{0.5}	GLEU	F _{0.5}
Single model:					
Junczys-Dowmunt et al. (2018)	53.0	-	-	57.9	-
Lichtarge et al. (2019)	56.8	-	-	61.6	-
Awasthi et al. (2019)	59.7	-	-	60.3	-
Kiyono et al. (2019)	61.3	68.6	71.3	59.7	64.2
SR +PRET+SED	61.4	69.3	72.5	63.3	65.5
Ensemble model:					
Junczys-Dowmunt et al. (2018)	55.8	-	-	59.9	-
Lichtarge et al. (2019)	60.4	-	-	63.3	-
Grundkiewicz et al. (2019)	64.2	-	-	61.2	69.5
Kiyono et al. (2019)	65.0	68.8	72.9	61.4	70.2
SR +PRET+R2L+SED	63.1	69.8	73.9	63.7	67.8

Table 7: Comparison with existing top models: a **bold** value denotes the best result within the column. Both SR and BEA indicate SR_{BEA+EF+L8} and BEA-test, respectively.

ensemble of four left-to-right (L2R) models and then re-scored the hypotheses using these models. We then re-ranked the n -best hypotheses based on the sum of the both two scores.

Sentence-level error detection (SED) SED is used to identify whether a given sentence contains any grammatical errors. Following the work presented by Asano et al. (2019), we employed a strategy based on reducing the number of false positives by only considering sentences that contained grammatical errors in the GEC model, using an SED model. We implemented the same model employed for SED filtering.

We evaluated the performance of the proposed best denoised model incorporated with the task-specific techniques on the three existing benchmarks: CoNLL-2014, JFLEG, and BEA-test, and then compared the scores with existing best-performing models. Table 7 shows the results for both the single and the ensemble models after applying PRET, SED¹³, and R2L to SR¹⁴. Since the reference of BEA-test is publicly unavailable,

¹³See Appendix F for an ablation study of SED.

¹⁴Improved results on the CoNLL-2014 and BEA-2019 have been appeared in arXiv less than 3 months before our submission (Kaneko et al., 2020; Omelianchuk et al., 2020) that are considered contemporaneous to our submission. More detailed experimental results, including a comparison with them, are presented in Appendix E for reference.

we evaluated the models on CodaLab¹⁵ under the rules of BEA-2019 workshop. We confirmed that our best denoised model works complementarily with existing task-specific techniques, as compared with the performance presented in Table 5. As a result, our best denoised model achieved state-of-the-art performance on the CoNLL-2014, JFLEG, and BEA-2019 benchmarks. Noteworthy is that the proposed model achieved state-of-the-art results on the JFLEG benchmark in terms of both single (GLEU = 63.3) and ensemble results (GLEU = 63.7). We provide a detailed analysis on this question in Section 6.5.

6 Analysis

6.1 Noise Reduction

To evaluate the quality of the dataset after denoising, a researcher with a high level of English proficiency (not involved with this work) manually evaluated 500 triples of source sentences \mathbf{X} , original target sentences \mathbf{Y} , and generated target sentences \mathbf{Y}' obtained by applying SR to EFCamDat satisfying $\mathbf{X} \neq \mathbf{Y} \neq \hat{\mathbf{Y}}$ (Table 8). We can see that 73.6% of the replaced samples were determined to be appropriate corrections, including cases where both were correct. For reference, we provide examples of a confusion set before and after denoising in the

¹⁵<https://competitions.codalab.org>

1: Improved by denoising (66.4%)
<i>X</i> : how about to going to movie . <i>Y</i> : How about to going to movie . <i>Y'</i> : How about going to a movie .
2: Both are correct (7.2 %)
<i>X</i> : I'm twenty-nine old. <i>Y</i> : I'm twenty-nine years old. <i>Y'</i> : I'm 29 years old.
3: Meaning is not preserved (10.4 %)
<i>X</i> : you need keep calm. <i>Y</i> : You need to keep calm. <i>Y'</i> : You need to be calm.
4: Added Unnecessary information (8.8 %)
<i>X</i> : The are a few of chair and desk. <i>Y</i> : There are a few chairs and desks. <i>Y'</i> : There are a few chairs and desks too.
5: Contains errors (3.8 %)
<i>X</i> : There are very positive news for us. <i>Y</i> : There is very positive news for us . <i>Y'</i> : There is a very positive news for us .
6: Lack of fluency (3.4 %)
<i>X</i> : I go in my work on the bike. <i>Y</i> : I go to work by bike. <i>Y'</i> : I go to work on my bike.

Table 8: Result of manual evaluation. Samples of input sentences (*X*), original target sentences (*Y*) and generated target sentences by our methods (*Y'*).

Appendix D.

6.2 Effect of the Fail-safe Mechanism

Next, we quantitatively and qualitatively analyzed the effectiveness of the fail-safe mechanism integrated into SR.

Quantitatively, Table 9 provides the results of the ablation study of the fail-safe mechanism on CoNLL-2014. Our main proposal was to include a self-refining step to clean up training data, but we found that the added fail-safe mechanism serves as a sub-component to further improve performance.

Qualitatively, we directly observed the decisions of the fail-safe mechanism and how it affected denoising. Table 10 provides examples for cases when SR activates and deactivates the fail-safe mechanism in EFCamDat. In the upper example (Table 10-1), **discuss of* in the source sentence should have been corrected to *discuss*; however, it was inaccurately edited to **discuss about* in the target sentence. In this case, SR succeeded in selecting the correct model output with a lower perplexity without activating the fail-safe mechanism. On the other hand, in the lower example, the model made

an incorrect “correction” (**in* → *at*). However, SR successfully activated the fail-safe mechanism and thus retained the correct original target sentence.

6.3 Benefits from Self-training

SR performed surprisingly well considering its simplicity. One reason might be that SR benefited from the advantages of self-training, as it could be regarded as a variant of self-training (Section 4). He et al. (2020) investigated the effect of self-training in neural sequence generation and found that the dropout in the pseudo-training step (namely, the training step of the denoised model in this study) played an important role in providing a smoothing effect, meaning that semantically similar inputs were mapped to the same or similar targets. As GEC also ideally holds the assumption of generating consistent targets for a similar context, this smoothing effect could contribute to avoiding overfitting and improving fitting the target distribution in the pseudo-training step. In fact, we confirmed that performance deteriorated when dropout was not applied in the training step of the denoised model, as shown in Table 11. In the case of relatively noisy data such as EFCamDat and Lang-8, the performance was better than without denoising, even without dropout. This could be explained by the presence of the denoising effect that was the objective of this study.

6.4 On the Increase of Recall

A pattern emerged when denoising with SR—recall significantly increased, while precision was mostly maintained (Figure 1). To clarify this observation, we manually assessed the amount of noise before and after denoising. Specifically, in the same way as in Section 3, we asked the expert to review 500 samples of the target sentence before denoising *Y* and the target sentences after denoising \hat{Y} . We then calculated the amount of noise using WER (Eq. 1). As a result, we observed a decrease in WER from 43.2% to 31.3% before and after denoising, respectively. This can be interpreted as (i) a large part of the noise was due to uncorrected errors, and (ii) the effect on model training was to correct the bias towards leaving errors unedited, resulting in higher recall.

6.5 Facilitating Fluency Edits

The results presented in Table 7 indicate that the proposed denoised model tends to (i) perform better

Model	EF			BEA+EF			BEA+EF+Lang-8		
	Prec.	Rec.	F _{0.5}	Prec.	Rec.	F _{0.5}	Prec.	Rec.	F _{0.5}
No denoising	48.5	24.0	40.3	58.5	30.0	49.1	62.8	39.2	56.1
SR w/o fail-safe	49.8	32.3	44.3	57.4	41.0	53.1	59.5	45.8	56.1
SR	55.2	32.4	48.4	61.2	38.0	54.5	61.7	49.4	58.8

Table 9: Ablation study of the fail-safe mechanism.

1 : Fail-safe deactivates:	ppl.
X : By the way, I have to *discuss of the education.	94.65
Y : By the way, I have to *discuss about education.	79.64
Y' : By the way, I have to discuss education.	73.37
2 : Fail-safe activates:	ppl.
X : Then I was treated in the hospital for one month.	34.34
Y : I was treated in the hospital for one month.	32.42
Y' : I was treated *at the hospital for one month.	33.59

Table 10: Examples of input sentences (X), original target sentences (Y) and generated target sentences by our methods (Y') when our method activates and deactivates the fail-safe in EFCamDat. ppl. indicates perplexity.

Model	F _{0.5}	GLEU
No denoising BEA	49.6	63.3
SR BEA w/ dropout	50.3	64.2
SR BEA w/o dropout	49.5	63.9
No denoising EF	40.3	61.3
SR EF w/ dropout	48.4	63.5
SR EF w/o dropout	47.3	63.0
No denoising L8	54.9	65.9
SR L8 w/ dropout	56.5	67.7
SR L8 w/o dropout	55.6	67.5

Table 11: Ablation study on the influence of dropout.

on JFLEG and (ii) be specifically highly rated in GLEU compared to other best-performing models. JFLEG was proposed by Napoles et al. (2017) for the development and evaluation of GEC models in terms of fluency and grammaticality, i.e., making a sentence more native sounding. Moreover, they showed that GLEU was correlated more strongly with humans than M^2 in JFLEG. The fact that SR is rated higher on JFLEG using GLEU than other best-performing models can be interpreted as achieving more fluent editing. One reason might be that SR performs a perplexity check on both the original target sentences and the new ones obtained after denoising, which always results in $PPL(Y) \geq PPL(\hat{Y})$ between \mathcal{D} and $\hat{\mathcal{D}}$. Therefore, SR can be expected to refine not only grammaticality but also

fluency of the target sentences, and as a result, the proposed denoised model is capable of performing more native-sounding corrections.

7 Conclusion and Future Work

In this study, we focused on the quality of GEC datasets. The motivation behind our study was based on the hypothesis that the carelessness or insufficient skill of the annotators involved in data annotation could often lead to producing noisy datasets. To address this problem, we presented a self-refinement approach as a simple but effective denoising method which improved GEC performance, and verified its effectiveness by comparing to both strong baselines based on filtering approach and current best-performing models. Furthermore, we analyzed how SR affects both GEC performance and the data itself.

Recently, several methods that incorporate pre-trained masked language models such as BERT, XLNet (Yang et al., 2019), and RoBERTa (Liu et al., 2019) into EncDec based GEC have been proposed and achieved remarkable results (Kaneko et al., 2020; Omelianchuk et al., 2020). These approaches modify the model architecture and do not directly compete with the data-driven approaches discussed in this study. Thus, the combination of these methods can be expected to further improve the performance, which we plan to investigate in our future work.

Acknowledgments

We thank the Tohoku NLP laboratory members who provided us with their valuable advice. We are grateful to Tomoya Mizumoto and Ana Brassard for their insightful comments and suggestions.

References

Hiroki Asano, Masato Mita, Tomoya Mizumoto, and Jun Suzuki. 2019. The AIP-Tohoku System at the BEA-2019 Shared Task. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2019)*, pages 176–182.

- Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel Iterative Edit Models for Local Sequence Transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 4259–4269.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.
- Chao Bei, Hao Zong, Yiming Wang, Baoyong Fan, Shiqi Li, and Conghu Yuan. 2018. An Empirical Study of Machine Translation for the Shared Task of WMT18. In *Proceedings of the Third Conference on Machine Translation (WMT 2018): Shared Task Papers*, pages 340–344.
- Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL Errors Using Phrasal SMT Techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 249–256.
- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2019)*, pages 52–75.
- Shamil Chollampatt and Hwee Tou Ng. 2018. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 5755–5762.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. Better Evaluation for Grammatical Error Correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2012)*, 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 the 8th Workshop on Building Educational Applications Using NLP (BEA 2013)*, pages 22–31.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*, pages 4171–4186.
- Rachele De Felice and Stephen G. Pulman. 2008. A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 169–176.
- Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching Human-level Performance in Automatic Grammatical Error Correction: An Empirical Study. *arXiv preprint arXiv:1807.01270*.
- Jeroen Geertzen, Dora Alexopoulou, and Anna Korhonen. 2013. *Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge Open Language Database (EFCAMDAT)*.
- Sylviane Granger. 1998. The computer learner corpus: A versatile new source of data for SLA research. In Sylviane Granger, editor, *Learner English on Computer*, pages 3–18.
- Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2019)*, pages 252–263.
- Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting Errors in English Article Usage by Non-Native Speakers. *Natural Language Engineering*, 12(2):115–129.
- Junxian He, Jiatao Gu, Jiajun Shen, and Marc’Aurelio Ranzato. 2020. Revisiting Self-Training for Neural Sequence Generation. In *International Conference on Learning Representations (ICLR 2020)*.
- Marcin Junczys-Dowmunt. 2018. Dual Conditional Cross-Entropy Filtering of Noisy Parallel Corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers (WMT 2018)*, pages 888–895.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching Neural Grammatical Error Correction as a Low-Resource Machine Translation Task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2018)*, pages 595–606.
- Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-Decoder Models Can Benefit from Pre-trained Masked Language Models in Grammatical Error Correction. *arXiv preprint arXiv:2005.00987*.
- Huda Khayrallah and Philipp Koehn. 2018. On the Impact of Various Types of Noise on Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83.

- Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An Empirical Study of Incorporating Pseudo Data into Grammatical Error Correction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 1236–1242.
- Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 Shared Task on Parallel Corpus Filtering. In *Proceedings of the Third Conference on Machine Translation (WMT 2018): Shared Task Papers*, pages 726–739.
- Philipp Koehn and Rebecca Knowles. 2017. Six Challenges for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation (WMT 2018)*, pages 28–39.
- Jared Lichtarge, Chris Alberti, and Shankar Kumar. 2020. Data weighted training strategies for grammatical error correction. *arXiv preprint arXiv:2008.02976*.
- Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora Generation for Grammatical Error Correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*, pages 3291–3301.
- 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*.
- Yu-Chun Lo, Jhih-Jie Chen, Chingyu Yang, and Jason Chang. 2018. Cool English: a Grammatical Error Correction System Based on Large Learner Corpora. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018): System Demonstrations*, pages 82–85.
- 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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*, pages 1309–1314.
- Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, pages 147–155.
- Ryo Nagata, Atsuo Kawai, Koichiro Morihito, and Naoki Isu. 2006. A feedback-augmented method for detecting errors in the writing of learners of English. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 241–248.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground Truth for Grammatical Error Correction Metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL & IJCNLP 2015)*, pages 588–593.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2016. GLEU Without Tuning. *arXiv preprint arXiv:1605.02592*.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 229–234.
- Feng Nie, Jin-Ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. 2019. A simple recipe towards reducing hallucination in neural surface realisation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 2673–2679.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanyskiy. 2020. GECToR – Grammatical Error Correction: Tag, Not Rewrite. *arXiv preprint arXiv:2005.12592*.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*, pages 48–53.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Nick Rossenbach, Jan Rosendahl, Yunsu Kim, Miguel Graça, Aman Gokrani, and Hermann Ney. 2018. The RWTH Aachen University Filtering System for the WMT 2018 Parallel Corpus Filtering Task. In *Proceedings of the Third Conference on Machine Translation (WMT 2018): Shared Task Papers*, pages 946–954.
- Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of Edinburgh’s neural MT systems for WMT17. In *Proceedings of the Second Conference on Machine Translation (WMT 2017)*, pages 389–399.

- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers (WMT 2016)*, pages 371–376.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725.
- Rico Sennrich and Biao Zhang. 2019. Revisiting Low-Resource Neural Machine Translation: A Case Study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 211–221.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 28 (NIPS 2014)*, pages 3104–3112.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 2818–2826.
- Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and Aspect Error Correction for ESL Learners Using Global Context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 198–202.
- 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 *Advances in Neural Information Processing Systems 31 (NIPS 2017)*, pages 5998–6008.
- Hongmin Wang. 2019. Revisiting challenges in data-to-text generation with fact grounding. In *Proceedings of the 12th International Conference on Natural Language Generation (INLG 2019)*, pages 311–322.
- Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020)*.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Álchay-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32 (NIPS 2019)*, pages 5753–5763.
- Helen Yannakoudakis, Øistein E. Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. Developing an Automated Writing Placement system for ESL Learners. *Applied Measurement in Education*, 31(3):251–267.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 180–189.
- Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*, pages 156–165.

A The BEA-2019 official dataset

The BEA-2019 Shared Task provided participants with the following datasets as official datasets: Lang-8 (Mizumoto et al., 2011; Tajiri et al., 2012), the National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013), the First Certificate in English corpus (Yannakoudakis et al., 2011), and W&I+LOCNESS (Yannakoudakis et al., 2018; Granger, 1998). The official dataset is publicly available at <https://www.cl.cam.ac.uk/research/nl/bea2019st/>.

B Hyper-parameter settings

Configurations	Values
Model Architecture	Transformer (Vaswani et al., 2017)
Optimizer	Adam (Kingma and Ba, 2015)
Learning Rate Schedule	Same as described in Section 5.3 of Vaswani et al. (2017)
Number of Epochs	30
Dropout	0.3
Stopping Criterion	Train model for 30 epochs. During the training, save model parameter for every 500 updates.
Gradient Clipping	1.0
Loss Function	Label smoothed cross entropy (Szegedy et al., 2016)
Beam Search	Beam size 5 with length normalization

Table 12: Detailed hyper-parameters used for the base GEC model.

C Preliminary experiment of the cross-entropy filtering

We investigated the effectiveness of changing the threshold of CE filtering by evaluating the model performance on BEA-valid. In this study, we prepared a forward and reverse pre-train model using BEA-train and CoNLL-2013 for as a training and validation set, respectively.

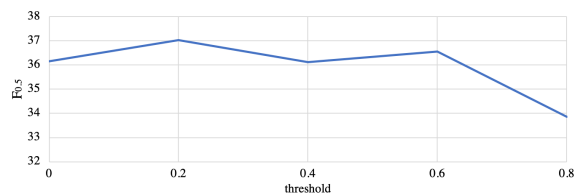


Figure 2: Performance of base GEC model on BEA-valid as threshold of CE filtering is varied.

D Examples of a confusion set before and after denoising

Table 13 provides examples of a confusion set before and after applying the denoising method to EFCamDat. We confirmed that we succeeded in reducing the noisy confusion set, including (*discuss about, *discuss about) or (*enter in, *enter in) in the target sentences using the proposed denoising.

Confusion set (X, Y)	Y (%)	\hat{Y} (%)
(*discuss about, *discuss about)	66.7	49.5
(*discuss about, discuss)	33.0	50.2
(*discuss about, *discuss in)	0.3	0.3
(*enter in, *enter in)	61.6	31.7
(*enter in, enter)	38.4	68.3

Table 13: Examples of confusion set before and after denoising in EFCamDat.

E Results of comparison with existing models

F Ablation study of SED

Model	CoNLL-2014				JFLEG				BEA-test		
	Prec.	Rec.	F _{0.5}	GLEU	Prec.	Rec.	F _{0.5}	GLEU	Prec.	Rec.	F _{0.5}
Single model:											
Junczys-Dowmunt et al. (2018)	-	-	53.0	-	-	-	-	57.9	-	-	-
Lichtarge et al. (2019)	65.5	37.1	56.8	-	-	-	-	61.6	-	-	-
Awasthi et al. (2019)	66.1	43.0	59.7	-	-	-	-	60.3	-	-	-
Kiyono et al. (2019)	67.9	44.1	61.3	68.6	76.6	55.8	71.3	59.7	65.5	59.4	64.2
Kaneko et al. (2020)	69.2	45.6	62.6	-	-	-	-	61.3	67.1	60.1	65.6
Omelianchuk et al. (2020)	77.5	40.1	65.3	-	-	-	-	-	79.2	53.9	72.4
SR _{BEA+EF+L8} +PRET	63.8	52.4	61.1	69.6	74.9	62.5	72.0	63.4	59.9	66.9	61.2
Ensemble model:											
Junczys-Dowmunt et al. (2018)	61.9	40.2	55.8	-	-	-	-	59.9	-	-	-
Lichtarge et al. (2019)	66.7	43.9	60.4	-	-	-	-	63.3	-	-	-
Grundkiewicz et al. (2019)	-	-	64.2	-	-	-	-	61.2	72.3	60.1	69.5
Kiyono et al. (2019)	72.4	46.1	65.0	68.8	79.5	54.6	72.9	61.4	74.7	56.7	70.2
Kaneko et al. (2020)	72.6	46.4	65.2	-	-	-	-	62.0	72.3	61.4	69.8
Omelianchuk et al. (2020)	78.2	41.5	66.5	-	-	-	-	-	78.9	58.2	73.6
SR _{BEA+EF+L8} +PRET + R2L	65.5	53.2	62.6	70.1	76.5	63.3	73.4	63.9	62.9	67.7	63.8

Table 14: Comparison with existing models: a **bold** value denotes the best result within the column. SR and BEA indicate SR_{BEA+EF+L8} and BEA-test, respectively. Kaneko et al. (2020) and Omelianchuk et al. (2020) have appeared on arXiv less than 3 months before our submission and are considered contemporaneous to our submission.

Model	CoNLL-2014				JFLEG				BEA-test		
	Prec.	Rec.	F _{0.5}	GLEU	Prec.	Rec.	F _{0.5}	GLEU	Prec.	Rec.	F _{0.5}
Single model:											
SR _{BEA+EF+L8} +PRET	63.8	52.4	61.1	69.6	74.9	62.5	72.0	63.4	59.9	66.9	61.2
SR _{BEA+EF+L8} +PRET+SED	65.2	49.9	61.4	69.3	76.3	60.6	72.5	63.3	66.7	61.3	65.5
Ensemble model:											
SR _{BEA+EF+L8} +PRET + R2L	65.5	53.2	62.6	70.1	76.5	63.3	73.4	63.9	62.9	67.7	63.8
SR _{BEA+EF+L8} +PRET+R2L+SED	67.1	50.8	63.1	69.8	77.8	61.5	73.9	63.7	69.4	62.1	67.8

Table 15: Ablation study of SED