A DIRECTNOISE Algorithm

The DIRECTNOISE algorithm is described in Algorithm 1 Here, $\boldsymbol{X}$ consists of sequence of I tokens, namely, $\boldsymbol{X} = (x_1, \ldots, x_I)$ where x_i denotes i-th token of $\boldsymbol{X}$. Similarly, $\boldsymbol{Y}$ consists of sequence of J tokens, namely, $\boldsymbol{Y} = (y_1, \ldots, y_J)$ where y_j denotes j-th token of $\boldsymbol{Y}$.

Algorithm 1: DIRECTNOISE Algorithm

Data: Grammatical sentence $\boldsymbol{Y} \in \mathcal{T}$
Result: Pseudo Corpus $\mathcal{D}_p$

$\mathcal{D}_p = \{\}$ `// create empty set`
$\boldsymbol{\mu} = \{\mu_{\text{mask}}, \mu_{\text{deletion}}, \mu_{\text{insertion}}, \mu_{\text{keep}}\}$ s.t. $\Sigma\boldsymbol{\mu}=1$
for $\boldsymbol{Y} \in \mathcal{T}$ **do**
  $\boldsymbol{X} = (\,)$
  for $j \in (1, \ldots, J)$ **do**
    $action \sim Cat(action|\boldsymbol{\mu})$
    if *action is keep* **then**
      append y_j to $\boldsymbol{X}$
    else if *action is mask* **then**
      append $\langle mask \rangle$ to $\boldsymbol{X}$
    else if *action is deletion* **then**
      continue
    else if *action is insertion* **then**
      append y_j to $\boldsymbol{X}$
      $w = sample\_from\_unigram\_distribution(\mathcal{D}_g)$
      append w to $\boldsymbol{X}$
  $\mathcal{D}_p = \mathcal{D}_p \cup \{(\boldsymbol{X}, \boldsymbol{Y})\}$

B BEA-2019 Workshop Official Dataset

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

C Data Preparation Process

The training data (BEA-train) is tokenized using spaCy tokenizer[14]. We used en_core_web_sm-2.1.0 model[15]. We remove sentence pairs that have identical source and target sentences from the training set, following (Chollampatt and Ng, 2018). Then we acquire subwords from target sentence through byte-pair-encoding (BPE) (Sennrich et al., 2016c) algorithm. We used subword-nmt implementation[16]. We apply BPE splitting to both source and target text. The number of merge operation is set to 8,000.

[14] https://spacy.io/
[15] https://github.com/explosion/spacy-models/releases/tag/en_core_web_sm-2.1.0
[16] https://github.com/rsennrich/subword-nmt

D Hyper-parameter Settings

Configurations	Values
Model Architecture	Transformer (Vaswani et al., 2017) ("big" setting)
Optimizer	Adam (Kingma and Ba, 2015) ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Same as described in Section 5.3 of Vaswani et al. (2017)
Number of Epochs	40
Dropout	0.3
Stopping Criterion	Train model for 40 epochs. During the training, save model parameter for every 500 updates. Then take average of last 20 checkpoints.
Gradient Clipping	1.0
Loss Function	Label smoothed cross entropy (smoothing value: $\epsilon_{ls} = 0.1$) (Szegedy et al., 2016)
Beam Search	Beam size 5 with length-normalization

Table 6: Hyper-parameter for JOINT optimization

Configurations	Values
Pretraining	
Model Architecture	Transformer (Vaswani et al., 2017) ("big" setting)
Optimizer	Adam (Kingma and Ba, 2015) ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Same as described in Section 5.3 of Vaswani et al. (2017)
Number of Epochs	10
Dropout	0.3
Gradient Clipping	1.0
Loss Function	Label smoothed cross entropy (smoothing value: $\epsilon_{ls} = 0.1$) (Szegedy et al., 2016)
Fine-tuning	
Model Architecture	Transformer (Vaswani et al., 2017) ("big" setting)
Optimizer	Adafactor (Shazeer and Stern, 2018)
Learning Rate Schedule	Constant learning rate of 3×10^{-5}
Number of Epochs	30
Dropout	0.3
Stopping Criterion	Use the model with the best validation perplexity on BEA-valid
Gradient Clipping	1.0
Loss Function	Label smoothed cross entropy (smoothing value: $\epsilon_{ls} = 0.1$) (Szegedy et al., 2016)
Beam Search	Beam size 5 with length-normalization

Table 7: Hyper-parameter for PRETRAIN optimization

E Mask Probability of DIRECTNOISE

In this paper, we exclusively focused on the effectiveness of μ_{mask}, and therefore we deliberately fixed $\mu_{\text{keep}} = 0.2$, and used $\mu_{\text{insertion}} = \mu_{\text{deletion}} = (1 - \mu_{\text{keep}} - \mu_{\text{mask}})/2$

We investigated the effectiveness of changing mask probability μ_{mask} of BACKTRANS (NOISY) by evaluating the model performance on BEA-valid. We used entire SimpleWiki as the seed corpus $\mathcal{T}$. The result is summarized in Figure 2. Here, increasing μ_{mask} within the range of $0.1 < \mu_{\text{mask}} < 0.5$ slightly improved the performance. Thus, used $\mu_{\text{mask}} = 0.5$ in the experiment (Section 4).

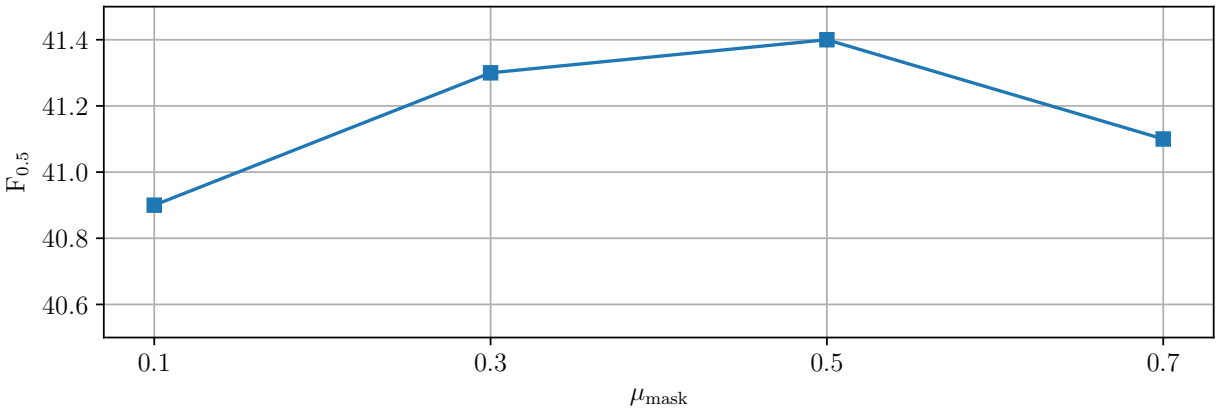

Figure 2: Performance of the model on BEA-valid as parameter of DIRECTNOISE (μ_{mask}) is varied.

F Noise Strength of BACKTRANS (NOISY)

We investigated the effectiveness of varying β_{random} hyper-parameter of BACKTRANS (NOISY) by evaluating its performance on BEA-valid (Figure 3). We used entire SimpleWiki as the seed corpus $\mathcal{T}$. The figure shows that the performance of backtranslation without noise ($\beta_{\text{random}} = 0$) is worse than the baseline. We believe that when there is no noise, reverse-model becomes too conservative to generate grammatical error, as discussed by Xie et al. (2018). Thus, the generated pseudo data cannot provide useful teaching signal for the model.

In terms of the scale of the noise, $\beta_{\text{random}} = 6$ is the best value for BACKTRANS (NOISY). Thus, we used this value in the experiment (Section 4).

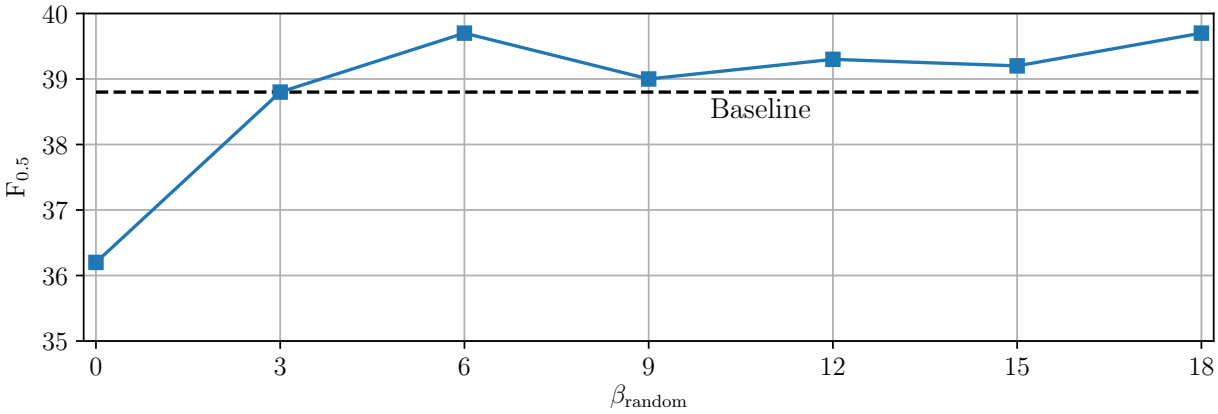

Figure 3: Performance of the model on BEA-valid as parameter of BACKTRANS (NOISY) (β_{random}) is varied.

G Examples of Noisy Sentences

Figure 4 shows examples of noisy sentences that are generated by BACKTRANS (NOISY) and DIRECTNOISE.

Original:	He died there , but the death date is not clear .
BACKTRANS (NOISY):	He died at there , but death date is not clear .
DIRECTNOISE:	⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ , 2 but ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ is not ⟨*mask*⟩ ⟨*mask*⟩
Original:	On seeing her his joy knew no bounds .
BACKTRANS (NOISY):	On seeing her joyful knew no bounds .
DIRECTNOISE:	⟨*mask*⟩ ⟨*mask*⟩ her crahis ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ bke ⟨*mask*⟩ .
Original:	Gre@@ en@@ space Information for G@@ rea@@ ter London .
BACKTRANS (NOISY):	The information for Gre@@ en@@ space information about G@@ rea@@ ter London .
DIRECTNOISE:	⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ for ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩
Original:	The cli@@ p is mixed with images of Toronto streets during power failure .
BACKTRANS (NOISY):	The cli@@ p is mix with images of Toronto streets during power failure .
DIRECTNOISE:	The ⟨*mask*⟩ is mixed ⟨*mask*⟩ images si@@ of The ⟨*mask*⟩ streets large ⟨*mask*⟩ power R@@ failure place ⟨*mask*⟩
Original:	At the in@@ stitute , she introduced tis@@ sue culture methods that she had learned in the U.@@ S.
BACKTRANS (NOISY):	At in@@ stitute , She introduced tis@@ sue culture method that she learned in U.@@ S.
DIRECTNOISE:	⟨*mask*⟩ the the ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ tis@@ culture R@@ methods , she P ⟨*mask*⟩ the s U.@@ ⟨*mask*⟩

Figure 4: Examples of sentences generated by BACKTRANS (NOISY) and DIRECTNOISE methods.

Figure 5 shows examples generated by DIRECTNOISE, when changing the mask probability (μ_{mask}).

μ_{mask}	Output Sentence
N/A	He threw the sand@@ wi@@ ch at his wife .
0.1	He ale threw , ch his ne@@ wife dar@@ ⟨*mask*⟩
0.3	⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ ch at ament his Research .
0.5	He o threw the sand@@ ch ⟨*mask*⟩ his ⟨*mask*⟩ .
0.7	⟨*mask*⟩ ⟨*mask*⟩ sand@@ ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ wife ⟨*mask*⟩

Figure 5: Examples generated when varying μ_{mask}. N/A denotes original text.

H Performance of the Model without Fine-tuning

PRETRAIN setting undergoes two optimization steps, namely, pretraining with pseudo data $\mathcal{D}_p$ and fine-tuning with genuine parallel data $\mathcal{D}_g$. We report the performance of models with pretraining only (Figure 6).

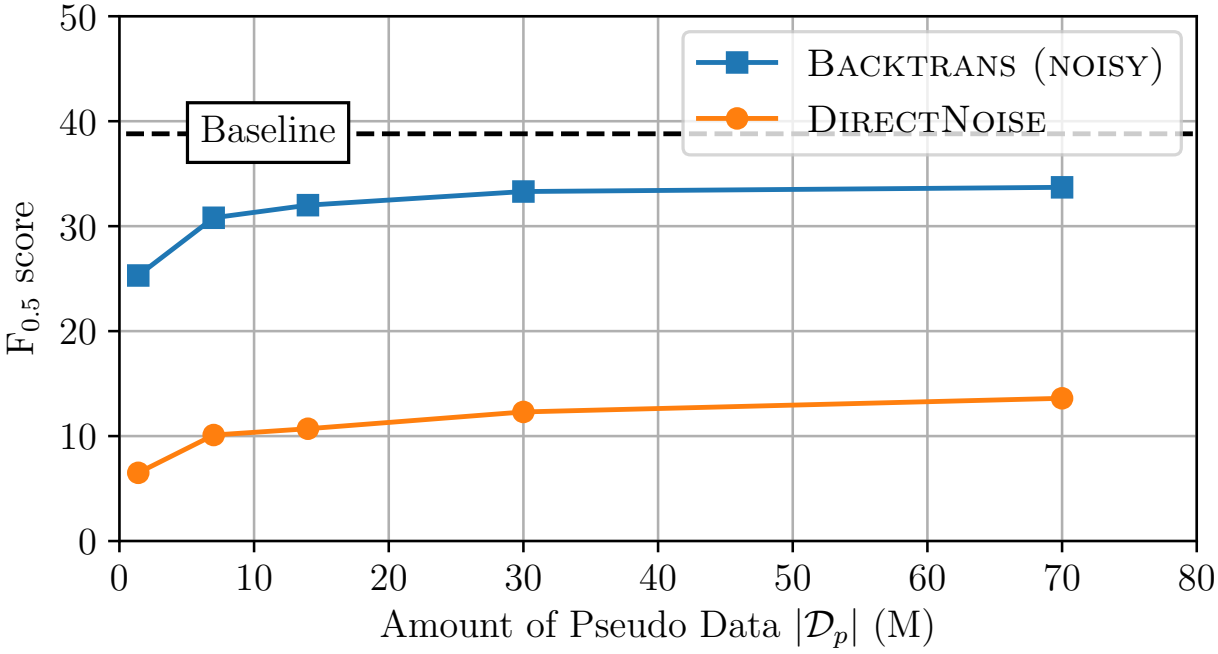

Figure 6: Performance on BEA-valid when varying the amount of pseudo data ($|\mathcal{D}_p|$)