Improving Autoregressive Grammatical Error Correction with Non-autoregressive Models

Hang Cao[1], **Zhiquan Cao**[1], **Chi Hu**[1], **Baoyu Hou**[1],
Tong Xiao[1,2]*, **Jingbo Zhu**[1,2]
[1]NLP Lab, School of Computer Science and Engineering
Northeastern University, Shenyang, China
[2]NiuTrans Research, Shenyang, China
caohang_neu@outlook.com
{xiaotong,zhujingbo}@mail.neu.edu.cn

Abstract

Grammatical Error Correction (GEC) aims to correct grammatical errors in sentences. We find that autoregressive models tend to assign low probabilities to tokens that need corrections. Here we introduce additional signals to the training of GEC models so that these systems can learn to better predict at ambiguous positions. To do this, we use a non-autoregressive model as an auxiliary model, and develop a new regularization term of training by considering the difference in predictions between the autoregressive and non-autoregressive models. We experiment with this method on both English and Chinese GEC tasks. Experimental results show that our GEC system outperforms the baselines on all the data sets significantly.

1 Introduction

Grammatical Error Correction (GEC) has attracted much attention in recent years, which aims to correct grammatical errors in a given text automatically. It is widely applied to natural language processing scenarios such as Automatic Speech Recognition (ASR) (Kubis et al., 2020; Wang et al., 2020), writing assistant and language learning platforms, etc. The GEC task is characterized by a significant overlap between input and output sentences with only a few errors requiring modification.

Since the transformer-based autoregressive (Vaswani et al., 2017) (AR) model with sequence-to-sequence (seq2seq) architecture has been successful in many generation tasks, a few works (Chollampatt and Ng, 2018) have applied it to the GEC task by taking the incorrect text as the source language and the text without errors as the target language, which has become a mainstream paradigm. However, in the GEC task, the overlap of source and target sentences makes the AR model simply copy most of the tokens over from the input to the output. We further find that the AR has high confidence for the tokens that are unchanged between the source and target sentence, while it usually has low confidence for correcting operations such as insertion, deletion, and substitution. Figure 1 is an example to illustrate this phenomenon. Intuitively, we believe that the reasonable cause of this phenomenon is the class imbalance issue (Li and Shi, 2021). With the influence of this problem, the AR model cannot confidently predict these incorrect tokens according to only the local context. Therefore, a natural idea is to improve the model performance by exploiting the global information, which can be captured by the non-autoregressive (NAR) (Gu et al., 2018; Lee et al., 2018) model. Although prior works have explored combining the two approaches through joint training, a combination for the GEC task is still missing. Besides, due to the inconsistency between AR and NAR output, a simple combination of them will lead to poor performance.

	Incorrect	Correct
Sub	X: I have a apple. an	Y: I have an apple. 0.92 0.85 0.42 0.74
Del	X: I have ~~the~~ an apple.	Y: I have an apple. 0.90 0.89 0.35 0.65
Ins	X: I have apple. an	Y: I have an apple. 0.86 0.91 0.25 0.81

Figure 1: Illustration for the confidence in different types of errors, where Sub denotes substitution, Del means deletion and Ins is insertion.

In this paper, we propose a simple yet novel approach to focus on incorrect tokens and integrate global information with the non-autoregressive model. Specifically, by masking the tokens in the golden target sentence corresponding to the low confidence positions of the AR output, we construct the input for NAR method. We combine the AR and NAR generation mechanisms to effectively

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 12014–12027
July 9-14, 2023 ©2023 Association for Computational Linguistics

utilize global information by constraining the consistency of their output distribution.

We conduct experiments on standard English GEC datasets and evaluate the system against strong baselines. Experimental results show that our approach can consistently achieve better results without relying on any resources other than the training data. Furthermore, we compare with a combination method of AR and NAR to verify whether the proposed model is more favorable for the GEC task. Here we use the Chinese GEC dataset as a benchmark to validate the generalization ability of the model. Meanwhile, we also conduct comparative ablation studies to illustrate the effectiveness of our proposed method.

2 Related Work

Seq2seq for GEC In recent years, a number of Transformer-based AR methods have been developed for GEC tasks. Junczys-Dowmunt et al. (2018) adapt several methods from low-resource machine translation to GEC by regarding GEC as low-resource machine translation. Zhao et al. (2019) aim to copy the words that overlap between the source and target sentence. They propose a copy-augmented architecture for GEC task which is pre-trained with unlabeled data. A series of work focus on data augmentation (Grundkiewicz et al., 2019; Ge et al., 2018; Lichtarge et al., 2019), Xie et al. (2018) propose to synthesize "realistic" parallel corpus with grammatical errors by back-translation. Zhao and Wang (2020) add a dynamic masking method to the original source sentence during training, which enhances the model performance without requiring additional data. With the help of large pre-trained language models (Kaneko et al., 2020), the performance of Transformer based AR models can be improved effectively. Meanwhile, the NAR approach emerges as a competitive alternative, which can correct the errors by modeling the whole sentence information. Li and Shi (2021) apply a Conditional Random Fields (CRF) layer to conduct non-autoregressive sequence prediction by modeling the dependencies among neighbor tokens.

Combination of AR and NAR The combination of AR and NAR modeling mechanisms has been discussed in other tasks. Wei et al. (2019) use a pre-trained AR model to supervise the decoding state of NAR, which can alleviate the problem of large search space. Li et al. (2019) propose that learning the hidden representation and attention distribution of AR by hints from the hidden representation can effectively improve the performance of NAR. Several approaches (Guo et al., 2020; Liu et al., 2020) are proposed to gradually guide the model transition from AR to NAR by designing the decoder input and semi-autoregressive tasks as courses. Some other works (Sun and Yang, 2020; Hao et al., 2021; Wang et al., 2022) attempt to utilize a unified framework to train AR and NAR jointly so that the NAR can be enhanced. Besides, Zhou et al. (2020) have also explored using the output of NAR to improve the AR performance. Unlike them, we focus on the GEC task and introduce the NAR model to utilize the global information to help the model understand the context around incorrect tokens.

3 Methodology

In this section, we elaborate on our proposed framework for GEC. As shown in figure 2, we introduce the CMLM-based NAR model to integrate more context information into our single model.

3.1 Overview

Given the training dataset (X, Y), the definition of the GEC task is to correct the original erroneous source X and generate a sentence Y without grammatical error, where $X = (x_1, x_2, ..., x_K)$ and $Y = (y_1, y_2, ..., y_N)$. Specifically, the transformer encoder takes the source sentence X as input. Different from previous seq2seq works, our decoder consists of two components: AR decoder and NAR decoder. We keep the AR decoder as a traditional seq2seq decoder without any change. For the NAR decoder, we mask the tokens in the input that corresponds to the positions of the low confidence tokens from the output distribution of the AR decoder. Then, we regenerate tokens in masked positions more accurately by bidirectional semantic modeling of the NAR decoder. Finally, we try to decrease the output distribution distance which is in masked positions between two manners to further improve the model performance during the training stage.

3.2 Mask low Confidence

Here, it is important that the output probability represents whether the model is confident in the prediction. As for the GEC task, there are only a

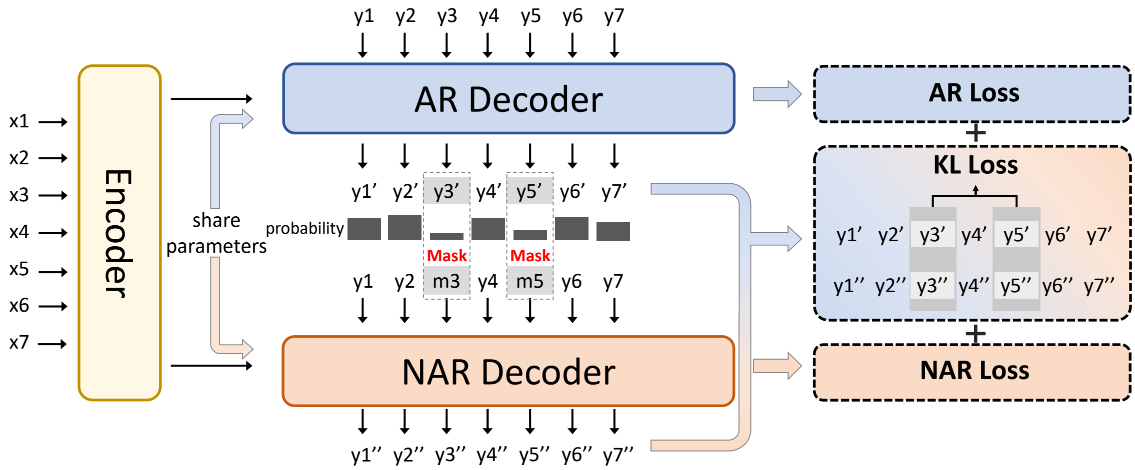

Figure 2: An illustration of our proposed model. The results $\{y_1', y_2', y_3', y_4', y_5', y_6', y_7'\}$ are predicted by AR Decoder with the golden target. The mask set $\{m_3, m_5\}$ indicates the position in the golden target that needs to be masked with a special token <mask>. After the mask operation, the input of the NAR decoder is $\{y_1, y_2, m_3, y_4, m_5, y_6, y_7\}$, and the output $\{y_3'', y_5''\}$ are re-predicted by CMLM based NAR decoder, and then the output are used to calibrate the corresponding AR decoder output.

Algorithm 1 Mask strategy

Input: The AR decoder output y_{AR}, the golden target y_{tgt}, the mask ratio δ

Output: The NAR input x_{NAR}

while not converged **do**
  Select the max probability for each token;
  $max\_logit \leftarrow \text{get\_max\_logit}(y_{AR})$;
  $L_t \leftarrow \text{get\_target\_length}(y_{tgt})$;
  $L_m = L_t \times \delta$;
  Get the L_m index of the low confidence positions in maxim logit;
  $Index \leftarrow \text{select\_index}(max\_logit, L_m)$;
  Replace with <mask> in the y_{tgt} corresponding position;
  $x_{NAR} \leftarrow \text{mask\_target}(Index, y_{tgt})$;
end while

few tokens that need to be modified (About 10%). Therefore, the model tends to focus on high confidence correct tokens that need to be kept, but not so much on low confidence tokens that need to be modified. As mentioned above, we choose the low confidence positions in the AR output distribution and substitute them with special symbols <mask> as the input of the NAR decoder. In this way, the NAR decoder is forced to learn the knowledge of low confidence tokens from the bidirectional context in hidden layers, which helps to boost the performance.

To construct the input effectively, we design a special mask strategy. Details are described in Algorithm 1. Specifically, we select the maximum probability of each token from the AR decoder output distribution. Then, we reorder each token in the output sentence from low confidence to high confidence to get a specific number of positions for the low confidence tokens. We introduce a special token <mask> to mask the token at the corresponding position in the golden target, which serves as a placeholder to represent the position where the target token needs to be regenerated. The golden target after the masking operation is used as input to the NAR decoder to introduce bidirectional contextual information.

3.3 Restrict Output Consistency

The objective of our model is to overcome the limitations of AR models by introducing the NAR generation mechanism, and then correct sentences with grammatical errors. A common way is to implicitly pass the information learned by the NAR branch to the AR branch using the parameter-sharing method. Specifically, we share the parameters of the Transformer layer in both manners. However, there is a huge difference between the AR manner and the NAR manner in the training process, as shown in Equation 1 and Equation 2, where the AR generation process is more concerned with local dependencies, while the NAR generation process is more concerned with global dependencies.

$$P(Y|X) = \prod_{i=1}^{N} P_{AR}(y_i|X, Y_{<i}), \quad (1)$$

$$P(Y|X) = \prod_{i=1}^{N} P_{NAR}(y_i|X). \quad (2)$$

The inconsistency between the two generation methods can lead to direct parameter sharing between the two branches without enabling the AR manner to obtain the exact information provided by the NAR manner. This sharing method only implicitly considers the correlation of model parameters and ignores the inconsistency between the two generation methods, which seriously hinders performance.

In contrast, in our work, to make the AR manner learn the information from NAR in a way that is more adapted to AR generation, we take an explicit approach to constrain the two manners. This approach can avoid the inconsistency caused by the different ways of AR and NAR generation and break the performance bottleneck. In practice, we accomplish explicit information modeling by using bidirectional Kullback-Leibler (KL) divergence to force the AR and NAR output distributions at the mask positions to be consistent with each other. Fortunately, Liang et al. (2022) also use KL divergence to combine the advantages of AR and NAR, which gives us much inspiration.

3.4 Training and Inference

Multi-Task Framework We learn the GEC model under the multi-task learning framework, including an AR primary task and a NAR auxiliary task. It should be noted that AR and NAR manners are regarded as two different tasks. For the AR task, we employ the negative log-likelihood (NLL) as the loss function which is akin to the traditional seq2seq. Therefore, the optimization objective is:

$$\mathcal{L}_{AR} = -\sum_{i=1}^{N} \log P_{AR}(y_i|X, Y_{<i}), \quad (3)$$

where N is the target length, and $Y_{<i}$ represents the tokens before the i-th time step. $P_{AR}(y_i|X, Y_{<i})$ represents the output probability of the AR decoder, which will be used in the later process.

For the NAR task, we obtain the positions of the specified number of low confidence tokens based on the mask ratio δ, and replace the tokens with the special symbols <mask> at the corresponding positions of the golden target. The loss function $\mathcal{L}_{NAR}$ for NAR task is to minimize the sum of negative log-likelihood in masked positions:

$$\mathcal{L}_{NAR} = -\sum_{i=1}^{M} \log P_{NAR}(y_i|X, Y_{mask}), \quad (4)$$

where M is the number of the masked tokens, and Y_{mask} is the set of the tokens in masked. In this way, the NAR decoder regenerates the masked tokens with more context information to help the AR task. Then the loss function of the multi-task framework is:

$$\mathcal{L}_m = \lambda_t \mathcal{L}_{NAR} + (1 - \lambda_t)\mathcal{L}_{AR}, \quad (5)$$

where λ_t is the important factor to balance the weight of AR and NAR tasks during training. We will present the design details in the following paragraphs.

Curriculum Learning Compared with the AR task, the NAR task is more complex, and unreasonable weight setting will make training difficult. For example, the excessive weight of the NAR task will disturb the parameter learning of the AR primary task at the beginning. Inspired by curriculum learning (Bengio et al., 2009), which is to imitate the human learning process, we propose the dynamic weight strategy. More concretely, we start with $\lambda_t = 0$ and gradually increase the NAR task weight λ_t to introduce learning signals. The dynamic weight scheme is:

$$\lambda_t = \frac{t}{T}, \quad (6)$$

where t and T are the current and total steps of training. We increase the weight linearly in all the experiments.

It is not enough to use only the hard parameter sharing method mentioned above, we regularize the two output distributions P_{AR} and P_{NAR} for unconfident words with the token-level bidirectional Kullback-Leibler divergence to further transfer the knowledge of NAR:

$$\begin{aligned}\mathcal{L}_{KL} = &\sum_{Y_{mask}} KL(P_{AR}||P_{NAR}) + \\ &\sum_{Y_{mask}} KL(P_{NAR}||P_{AR}).\end{aligned} \quad (7)$$

The final training objective for our GEC model is a combination of the three terms reviewed above as:

$$\mathcal{L} = \lambda_t \mathcal{L}_{NAR} + (1 - \lambda_t)\mathcal{L}_{AR} + \alpha \mathcal{L}_{KL}. \quad (8)$$

Model	Architecture	Precision	Recall	$F_{0.5}$
Transformer Big†	1024-1024-4096	65.26	27.19	50.98
LaserTagger⋆ (Malmi et al., 2019)	-	50.9	26.9	43.2
Adversarial-GEC (Raheja and Alikaniotis, 2020)	512-512-2048	64.68	22.57	47.10
ESD+ESC⋆ (Chen et al., 2020)	1024-1024-4096	66.0	24.7	49.5
SAD(9+3) (Sun et al., 2021)	1024-1024-4096	58.8	**33.1**	50.9
S2A (Li et al., 2022)	1024-1024-4096	65.9	28.9	52.5
CMLM† (Ghazvininejad et al., 2019)	1024-1024-4096	46.3	27.17	40.59
Levenshtein Transformer⋆ (Gu et al., 2019)	1024-1024-4096	39.9	24.4	35.4
JANUS† (Liang et al., 2022)	1024-1024-4096	66.22	27.76	51.85
Ours-base	512-512-2048	**66.63**	28.70	52.70
Ours	1024-1024-4096	65.10	32.29	**54.11**

Table 1: The results of systems on the CoNLL-2014 English GEC task. For the models with ⋆, their performance is from (Chen et al., 2020). † indicates the models are implemented by us with the released codes of the original papers. The Architecture column represents the embedding, hidden, and FFN size of the model. Here we **bold** the best results of the models.

Inference During the inference stage, we use the AR decoder to generate the correct sentences, and the inference efficiency is the same as the traditional seq2seq model since the NAR decoder is only used in training.

4 Experimental Setup

4.1 Datasets

To validate the effectiveness of our proposed GEC model, we conduct a set of experiments on both the restricted track of the BEA-2019 GEC shared task (Bryant et al., 2019) and NLPCC 2018 Task 2 (Zhao et al., 2018).

BEA-2019 GEC shared task This is a public dataset for the English GEC task, we follow the setting of (Chollampatt and Ng, 2018) and take the FCE training set (Yannakoudakis et al., 2011), Lang-8 Corpus of Learner English (Mizumoto et al., 2011), NUCLE (Dahlmeier et al., 2013) and W&I+LOCNESS (Granger, 2014; Bryant et al., 2019) as the training set. The development set is a subset of NUCLE, and our model is evaluated on the CoNLL-2014 (Ng et al., 2014), which is a well-known English GEC benchmark test set. Specifically, we use pre-processed script[1] in (Chollampatt and Ng, 2018) to obtain the parallel corpus.

NLPCC 2018 Task 2 It is the first and latest benchmark dataset for Chinese GEC. We combine the incorrect sentence with each corrected sentence to build the parallel sentence pairs as described in (Zhao and Wang, 2020) and get 1.2 million sentence pairs in all. Next, we randomly sample 5,000 training instances as the development set. The official test set extracted from PKU Chinese Learner Corpus contains 2,000 samples. We use the combination of two group annotations that mark the golden edits of grammatical errors in these sentences to evaluate our model. Following the setting of NLPCC 2018 Task (Zhao et al., 2018), the tokenization of training data is implemented with the PKUNLP tool[2].

4.2 Settings

While in the training process, we use the base model configuration of the Transformer for the Chinese GEC task, with 6 layers, the number of self-attention heads is set to 8, the embedding dimension is 512 and the size of FFN layer is 2048, the dropout and weight decay is 0.3 and 0.01 respectively. In the English GEC task, we use the big Transformer setting, which contains 6 layers and 16 self-attention heads, the size of word vectors on the source side and the target side are 1024, the FFN layer size is 4096, the dropout is applied with a probability of 0.1 and the weight decay value is set to be 0.0001. We adopt Adam (Kingma and Ba, 2015) optimizer with initial learning rate 0.0005 and 0.0007 for Chinese and English GEC tasks respectively, and a beta value of (0.9, 0.98). We use

[1] https://github.com/nusnlp/mlconvgec2018/tree/master/data

[2] https://github.com/zhaoyyoo/NLPCC2018_GEC

Model	Model type	Precision	Recall	$F_{0.5}$
Transformer	Single	36.91	15.57	28.97
YouDao (Fu et al., 2018)	Ensemble	35.24	**18.64**	29.91
AliGM (Zhou et al., 2018)	Ensemble	41.00	13.75	29.36
BLCU (Ren et al., 2018)	Ensemble	**47.63**	12.56	30.57
ESD+ESC (Chen et al., 2020)	Single	37.3	14.5	28.4
SAD(9+3) (Sun et al., 2021)	Single	33.0	20.5	29.4
S2A (Li et al., 2022)	Single	36.57	18.25	30.46
Ours	Single	41.90	15.24	**31.04**

Table 2: The results of systems on the NLPCC-2018 Chinese GEC task. For a fair comparison, all the results are produced by training on the original NLPCC-2018 training data. We **bold** the best results.

learning rate schedule as in (Vaswani et al., 2017), 10,000 warmup steps for the Chinese GEC task and 4,000 for the English GEC task. Lable smoothing is added with an epsilon value of 0.1. We use 32K Byte Pair Encoding (BPE) (Sennrich et al., 2016) for tokenization on Chinese and English GEC tasks. We save the checkpoint for each epoch and select the best checkpoint based on the loss on the development set. The beam size is 5 during the inference stage. All experiments are based on fairseq (Ott et al., 2019).

4.3 Baselines

We compare the performance of the proposed model with several representative baseline methods on both English and Chinese GEC tasks. Specifically, for the English GEC task, **Transformer Big** is the typical AR model. **LaserTagger** proposes to predict tags with a smaller vocabulary (Malmi et al., 2019). **Adversarial-GEC** presents an adversarial learning approach to generate realistic texts in a generator-discriminator framework (Raheja and Alikaniotis, 2020). **ESD+ESC** is a pipeline model (Chen et al., 2020). **SAD** employs a new decoding method with a shallow decoder to conduct the prediction (Sun et al., 2021). **S2A** proposes to integrate action probabilities into token prediction probabilities to obtain the final results (Li et al., 2022). **Levenshtein Transformer** (Gu et al., 2019) and **CMLM** (Ghazvininejad et al., 2019) are NAR models, which achieve excellent performance with an iterative generation paradigm. In addition, we also compare with **JANUS** (Liang et al., 2022), which joints AR and NAR training for sequence generation.

For the Chinese GEC task, we compare our model to all previous systems in the NLPCC 2018 dataset. **YouDao** corrects the sentences independently by utilizing five different mixture models (Fu et al., 2018). **AliGM** combines three approaches, including NMT-based, SMT-based, and rule-based together (Zhou et al., 2018). **BLCU** is based on a multi-layer convolutional seq2seq model (Ren et al., 2018).

4.4 Evaluation Metrics

Following the typical previous works (Chen et al., 2020; Li et al., 2022), we use the official MaxMatch (M^2) (Dahlmeier and Ng, 2012) scorer for evaluation of our grammatical error correction system. M^2 scorer computes the sequence of phrase-level edits between a source sentence and a system hypothesis that achieves the maximal overlap with the gold standard annotation. Given the set of system edits and the set of gold edits for all sentences, the value of precision, recall, and $F_{0.5}$ are computed by m2scorer [3].

5 Results

5.1 Main Results

The results of our proposed approach and recent models on English GEC task are shown in Table 1. We can see that our approach significantly outperforms the baselines mentioned above. Our model achieves an improvement above Transformer Big by nearly 3.1 in $F_{0.5}$ score, and performs better than the strong baseline S2A, by a large margin of 1.6 $F_{0.5}$. Moreover, the proposed model surpasses the recent JANUS model by $F_{0.5}$ score of 2.3, which shows excellent performance on multiple tasks by combining AR and NAR. This result implies that our designed joint training method is more suit-

[3] https://github.com/nusnlp/m2scorer

Mask Ratio	BEA-2019				NLPCC-2018			
	Precision	Recall	$F_{0.5}$	F_1	Precision	Recall	$F_{0.5}$	F_1
10%	61.70	31.31	51.67	41.61	38.99	14.57	29.20	21.22
15%	62.32	31.82	52.30	40.65	41.90	**15.24**	**31.04**	**22.35**
20%	**65.10**	**32.29**	**54.11**	**43.21**	**42.24**	14.95	30.94	22.09
25%	64.87	30.61	53.01	41.68	40.01	14.48	29.58	21.26
30%	63.35	29.68	51.63	40.56	41.18	13.68	29.36	20.54
35%	64.51	30.43	52.71	41.48	41.73	13.70	29.61	20.63

Table 3: Effect of Mask Ratio. The mask ratio represents the percentage of low-confidence tokens in a sentence that are masked. Best results of the Chinese GEC task and the English GEC task are **bold** separately.

Model	Sub	Del	Ins
AR	58.12%	77.50%	73.82%
Ours	52.30%	52.84%	61.70%

Table 4: The Correction Coincidence Rate of the AR model and our method on the English CoNLL-2014 test set.

able for the GEC task. It is noteworthy that our model with Transformer base settings still consistently exceeds the baselines with Transformer big settings. These results all support that our proposed approach can effectively improve the AR GEC by using a NAR model.

To validate the effectiveness of our approach, we conduct experiments on the Chinese GEC task and present the results in Table 2. These results demonstrate that the Chinese GEC task is more challenging than the English GEC. Despite this, the proposed model yields a higher $F_{0.5}$ than the listed methods. Moreover, we can observe that all the top three models are ensemble models, including YouDao, AliGM, and BLCU, but our single model still surpasses them. This result means that our model is generalizable.

5.2 Fix more Grammar Errors

We carefully investigate the number of different types of errors corrected in the two datasets, and find that most of the corrected grammar errors are the same between the proposed method and the AR model. To show the advantages of our model intuitively, we propose a Correction Coincidence Rate, which is the number of overlaps of correction errors to the total number of respective correction errors. The results are summarized in Table 4. For computational convenience, the errors are broadly categorized into insertion, deletion, and substitution. From Table 4, the overlap rate of our proposed method on all types of error modifications is much lower. For instance, the percentage of deletion decreases by 25%. This indicates that our model is able to correct more grammar errors while maintaining the ability of the AR model.

5.3 Ablation Analysis

Effect of Mask Ratio In this section, we present exhaustive investigations on the impact of mask ratio. Here we vary the mask ratios in $\{0.1, 0.15, 0.2, 0.3, 0.35\}$ and conduct experiments in BEA-2019 and NLPCC 2018. The corresponding results are provided in Table 3. It can be observed that all mask ratios outperform the Transformer baseline. A reasonable reason is that the masking operation makes the model focus more on incorrect tokens, and the model is forced to capture more context information, which facilitates error correction. On the other hand, a small mask ratio (e.g., 0.1) cannot perform as well as a large one (e.g., 0.15), which means that there is a fraction of incorrect tokens that are not focused on. However, too much masking ratio is also not good. It will result in many correct words being masked, which may prevent the correction of incorrect tokens. Note that the choice of mask ratio is distinct for different datasets, and the best balance choices for BEA-2019 and NLPCC-2018 are 0.2 and 0.15 respectively.

Effect of KL Loss Weight α We explore the effect of KL-divergence loss weight α in Equation 8. The result is illustrated in Table 6. By comparing the performance with KL loss and without KL loss, we can see that the performance of the former is consistently better than the performance of the latter, which suggests that KL loss can be further combined with information from AR and NAR to correct errors. In addition, the performance is lower

Type	Samples
SRC	I think the family will stay mentally healty as it is, without having emtional stress.
TGT	I think the family will stay mentally healthy as it is, without having emotional stress.
Transformer	I think the family will stay mentally healty as it is, without having emtional stress.
Ours	I think the family will stay mentally healthy as it is, without having emotional stress.
SRC	While we do know that we should not discriminate them based on their limitations...
TGT	While we do know that we should not discriminate against them based on their limitations...
Transformer	While we do know that we should not discriminate them based on their limitations...
Ours	While we do know that we should not discriminate against them based on their limitations...
SRC	First and foremost, I would like to share ~~on~~ the advantages of using such social media...
TGT	First and foremost, I would like to share the advantages of using such social media...
Transformer	First and foremost, I would like to share on the advantages of using such social media...
Ours	First and foremost, I would like to share the advantages of using such social media...

Table 5: Case studies of the original Transformer model and our proposed model on the English CoNLL-2014 test set. The tokens in red and wave line are errors, while tokens with underline and in green are the corrections made by the gold target or our model.

TrainSet	α	Precision	Recall	$F_{0.5}$
BEA-2019	0	59.47	31.82	50.66
	0.3	61.72	32.21	52.16
	0.4	60.68	31.27	51.07
	0.5	**65.10**	**32.29**	**54.11**
	0.6	60.51	31.88	51.30
	0.7	65.03	31.29	53.50
NLPCC-2018	0	37.58	14.34	28.38
	0.8	40.53	**15.54**	30.66
	0.9	39.46	14.24	29.14
	1.0	**41.90**	15.24	**31.04**
	1.1	39.34	14.23	29.07
	1.2	40.32	13.88	29.20

Table 6: The results with different weight. α equal 0 represents the model without KL loss. Weights are adjusted according to different datasets. Best results are **bold**.

than the baseline when the value of α is 0, i.e., the model is fused using only the simple method of parameter sharing. It indicates that simple fusion will lead to poor performance. We also find that the performance is not optimal when α is set too small or too large. We believe that the model does not learn enough information when α is set too small, while setting it too large leads to the introduction of too much noise.

5.4 Case Study

In order to qualitatively show the effectiveness of global context information, we conduct case studies with Transformer and our proposed model. We pick the cases from the CoNLL-2014 English GEC test set. The results are listed in Table 5. Generally, it is easy to see that both approaches can copy most of the correct tokens from the source to the target. Nevertheless, when correcting grammatical errors, our approach can predict more accurately by considering more context information. For example, as shown in the third sample in Table 5, the AR model generates the phase "share on" which tends to be consistent with the source language, while our model can delete the token "on" by utilizing more context information. This again confirms that our method can make use of the global information to correct errors.

6 Conclusion

In this work, we propose a joint AR and NAR learning objective for the GEC, using a multi-task learning framework. To better predict tokens at low-confidence positions, we introduce additional signals to the training of GEC models by using the NAR model as an auxiliary model. Meanwhile, we develop a new regularization term of training to constrain the inconsistency between the two manners. Through our experiments in the English and Chinese GEC task, the proposed approach can significantly improve the GEC model performance without additional inference costs.

In the future, we are also interested in introducing syntax and lexical knowledge to focus on incorrect tokens to further improve performance.

7 Limitations

In this work, we achieve a noticeable improvement in the GEC task by introducing additional context information with a NAR model. However, in order to focus on incorrect tokens, the input of the NAR is required to be constructed based on the AR output distribution. In this way, the AR and NAR model perform sequentially, which leads to much time consumption in the training stage. In the future, we will apply a layer dropout strategy to speed up model training. On the other hand, due to the limitation of computation resources, all experiments are conducted on two Nvidia TITAN V GPUs with 12GB VRAM. Therefore, we could not compare with the state-of-the-art models which are pre-trained with 100M synthetic parallel examples (Li et al., 2022). We left it as our future work.

8 Acknowledgments

This work was supported in part by the National Science Foundation of China (No.62276056), the National Key R&D Program of China, the China HTRD Center Project (No.2020AAA0107904), the Natural Science Foundation of Liaoning Province of China (2022-KF-16-01), the Yunnan Provincial Major Science and Technology Special Plan Projects (No.202103AA080015), the Fundamental Research Funds for the Central Universities (Nos.N2216016, N2216001, and N2216002), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009).

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

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@ACL 2019, Florence, Italy, August 2, 2019*, pages 52–75. Association for Computational Linguistics.

Mengyun Chen, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020. Improving the efficiency of grammatical error correction with erroneous span detection and correction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7162–7169. Association for Computational Linguistics.

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-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5755–5762. AAAI Press.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, pages 568–572. The Association for Computational Linguistics.

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 Eighth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@NAACL-HLT 2013, June 13, 2013, Atlanta, Georgia, USA*, pages 22–31. The Association for Computer Linguistics.

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 *Natural Language Processing and Chinese Computing - 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26-30, 2018, Proceedings, Part I*, volume 11108 of *Lecture Notes in Computer Science*, pages 341–350. Springer.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Fluency boost learning and inference for neural grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1055–1065. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. 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, Hong Kong, China, November 3-7, 2019*, pages 6111–6120. Association for Computational Linguistics.

Sylviane Granger. 2014. The computer learner corpus: a versatile new source of data for sla research. In *Learner English on computer*, pages 3–18. Routledge.

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@ACL 2019, Florence, Italy, August 2, 2019*, pages 252–263. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11179–11189.

Junliang Guo, Xu Tan, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. 2020. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7839–7846. AAAI Press.

Yongchang Hao, Shilin He, Wenxiang Jiao, Zhaopeng Tu, Michael R. Lyu, and Xing Wang. 2021. Multi-task learning with shared encoder for non-autoregressive machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3989–3996. Association for Computational Linguistics.

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: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 595–606. Association for Computational Linguistics.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4248–4254. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Marek Kubis, Zygmunt Vetulani, Mikolaj Wypych, and Tomasz Zietkiewicz. 2020. Open challenge for correcting errors of speech recognition systems. *CoRR*, abs/2001.03041.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1173–1182. Association for Computational Linguistics.

Jiquan Li, Junliang Guo, Yongxin Zhu, Xin Sheng, Deqiang Jiang, Bo Ren, and Linli Xu. 2022. Sequence-to-action: Grammatical error correction with action guided sequence generation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10974–10982. AAAI Press.

Piji Li and Shuming Shi. 2021. Tail-to-tail non-autoregressive sequence prediction for chinese grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4973–4984. Association for Computational Linguistics.

Zhuohan Li, Zi Lin, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Hint-based training for non-autoregressive machine translation. 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, Hong Kong, China, November 3-7, 2019*, pages 5707–5712. Association for Computational Linguistics.

Xiaobo Liang, Lijun Wu, Juntao Li, and Min Zhang. 2022. Janus: Joint autoregressive and nonautoregressive training with auxiliary loss for sequence generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, pages 8050–8060. The Association for Computer Linguistics.

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: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7,* *2019, Volume 1 (Long and Short Papers)*, pages 3291–3301. Association for Computational Linguistics.

Jinglin Liu, Yi Ren, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. Task-level curriculum learning for non-autoregressive neural machine translation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3861–3867. ijcai.org.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. 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, Hong Kong, China, November 3-7, 2019*, pages 5053–5064. Association for Computational Linguistics.

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 *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 147–155. The Association for Computer Linguistics.

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 the Eighteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014*, pages 1–14. ACL.

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: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics.

Vipul Raheja and Dimitris Alikaniotis. 2020. Adversarial grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 3075–3087. Association for Computational Linguistics.

Hongkai Ren, Liner Yang, and Endong Xun. 2018. A sequence to sequence learning for chinese grammatical error correction. In *Natural Language Processing and Chinese Computing - 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26-30, 2018, Proceedings, Part II*, volume 11109 of *Lecture Notes in Computer Science*, pages 401–410. Springer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Xin Sun, Tao Ge, Furu Wei, and Houfeng Wang. 2021. Instantaneous grammatical error correction with shallow aggressive decoding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5937–5947. Association for Computational Linguistics.

Zhiqing Sun and Yiming Yang. 2020. An EM approach to non-autoregressive conditional sequence generation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9249–9258. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Haoyu Wang, Shuyan Dong, Yue Liu, James Logan, Ashish Kumar Agrawal, and Yang Liu. 2020. ASR error correction with augmented transformer for entity retrieval. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 1550–1554. ISCA.

Xinyou Wang, Zaixiang Zheng, and Shujian Huang. 2022. Helping the weak makes you strong: Simple multi-task learning improves non-autoregressive translators. *CoRR*, abs/2211.06075.

Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. 2019. Imitation learning for non-autoregressive neural machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1304–1312. Association for Computational Linguistics.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Y. Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 619–628. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 180–189. The Association for Computer Linguistics.

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: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 156–165. Association for Computational Linguistics.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the NLPCC 2018 shared task: Grammatical error correction. In *Natural Language Processing and Chinese Computing - 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26-30, 2018, Proceedings, Part II*, volume 11109 of *Lecture Notes in Computer Science*, pages 439–445. Springer.

Zewei Zhao and Houfeng Wang. 2020. Maskgec: Improving neural grammatical error correction via dynamic masking. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 1226–1233. AAAI Press.

Junpei Zhou, Chen Li, Hengyou Liu, Zuyi Bao, Guangwei Xu, and Linlin Li. 2018. Chinese grammatical error correction using statistical and neural models. In *Natural Language Processing and Chinese Computing - 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26-30, 2018, Proceedings, Part II*, volume 11109 of *Lecture Notes in Computer Science*, pages 117–128. Springer.

Long Zhou, Jiajun Zhang, and Chengqing Zong. 2020. Improving autoregressive NMT with non-autoregressive model. In *Proceedings of the First Workshop on Automatic Simultaneous Translation*, pages 24–29, Seattle, Washington. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☑ A2. Did you discuss any potential risks of your work?
7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
7

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
5.3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4.4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.