Improving Chinese Grammatical Error Detection via Data augmentation by Conditional Error Generation

Tianchi Yue, Shulin Liu, Huihui Cai, Tao Yang, Shengkang Song, Tinghao Yu
Tencent AI Platform Department, China
{tianchiyue, forestliu, helenhhcai}@tencent.com
{rigorosyang, shengksong, maxwellyu}@tencent.com

Abstract

Chinese Grammatical Error Detection(CGED) aims at detecting grammatical errors in Chinese texts. One of the main challenges for CGED is the lack of annotated data. To alleviate this problem, previous studies proposed various methods to automatically generate more training samples, which can be roughly categorized into rule-based methods and model-based methods. The rule-based methods construct erroneous sentences by directly introducing noises into original sentences. However, the introduced noises are usually context-independent, which are quite different from those made by humans. The model-based methods utilize generative models to imitate human errors. The generative model may bring too many changes to the original sentences and generate semantically ambiguous sentences, so it is difficult to detect grammatical errors in these generated sentences. In addition, generated sentences may be error-free and thus become noisy data. To handle these problems, we propose CNEG, a novel **C**onditional **N**on-Autoregressive **E**rror **G**eneration model for generating Chinese grammatical errors. Specifically, in order to generate a context-dependent error, we first mask a span in a correct text, then predict an erroneous span conditioned on both the masked text and the correct span. Furthermore, we filter out error-free spans by measuring their perplexities in the original sentences. Experimental results show that our proposed method achieves better performance than all compared data augmentation methods on the CGED-2018 and CGED-2020 benchmarks.

1 Introduction

The goal of Grammatical Error Detection is to detect grammatical errors in texts (Rao et al., 2018). It is useful for many NLP applications such as writing assistant (Napoles et al., 2017), search engine (Gao et al., 2010), and speech recognition systems (Wang et al., 2020a), etc. Grammatical errors may appear in all languages (Dale et al., 2012; Bryant et al., 2019). In this paper, we only investigate the problem of Chinese Grammatical Error Detection (CGED).

我看过有一个因很长时间吸烟而终于死亡的人。
I have seen a person who finally died after smoking for a long time.
我看见一个因很长时间吸烟而最终死亡的人。
I saw a man who finally died of smoking for a long time.

Figure 1: An error-correction pair from CGED datasets. The first line is an erroneous sentence, tokens marked in blue color are selection errors, tokens marked in green color are redundant words. The second line is the corrected sentence.

Grammatical Error Detection is usually formulated as a sequence tagging task, where each erroneous token is assigned with an error type, e.g., selection errors and redundant words, as shown in Figure 1. Since annotating grammatical errors requires rich linguistic knowledge, it is expensive and time-consuming to annotate a large-scale corpus. Therefore, the scarcity of labeled data is one of the main challenges for this task. To handle this problem, previous works proposed various data augmentation methods to automatically generate more training samples (Kiyono et al., 2019; Wang et al., 2019; Lichtarge et al., 2019; Kasewa et al., 2018). The methods of generating erroneous sentences can be roughly categorized into the following two types: (1) **Rule-based methods**. These methods construct erroneous sentences by introducing noises into original texts, e.g., inserting, deleting, or replacing some words (Zhao et al., 2019; Wang et al., 2019). As the erroneous sentence shown in Figure 1, human grammatical errors are usually context dependent. On the contrary, the randomly introduced errors are context-independent (case I in Figure 2), therefore these noise-corrupted sentences are quite different from the erroneous sentences made by humans. (2) **Model-based methods**. In order to imitate human errors, many studies utilize neu-

Findings of the Association for Computational Linguistics: ACL 2022, pages 2966 - 2975

Original Sentence		我看见一个因很长时间吸烟而最终死亡的人。	(I saw a man who finally died of smoking for a long time.)
I	Context-independent	我看见一个因很长时间美丽吸烟而最终死亡的人。	(I saw a man who finally died of beauty smoking for a long time.)
II	Semantic-ambiguity	我看到一个男人死于吸烟，已经很久。	(I saw a man who died of smoking, it's been a long time.)
III	Error Free	我看到了一个长期吸烟而死的人 。	(I saw a man who died from smoking for a long time.)

Figure 2: Illustration for some examples generated by various data augmentation methods. Tokens marked in red are the modifications against the original sentence. Example I is constructed by the rule-based method, and the introduced error is meaningless to the original sentence. It is too easy for the detection model to detect such error. Example II is constructed by the model-based method, which is quite different from the original sentence. It is difficult to judge which tokens are grammatical errors when comparing the generated sentence with the original sentence. Example III is also different from the original sentence, but it does not contain any grammatical errors.

ral generative models to generate grammatical errors, such as Seq2Seq models (Kasewa et al., 2018; Wan et al., 2020), and translation models which obtain erroneous sentences via round-trip translation through a bridge language (Zhou et al., 2020; Lichtarge et al., 2019). However, considering that the outputs of generative models are not usually faithful to the inputs (Weng et al., 2020), semantic and syntactic ambiguities may arise when the generative models bring too many changes to the original sentences (case II in Figure 2). Even human can not infer the correct sentences from these generated sentences, so it is also difficult for the detection model to automatically detect grammatical errors. Meanwhile, generated sentences may be error-free and become noisy data (case III in Figure 2). Therefore, these constructed samples have little contributions to improving the performance of detection models.

To handle the aforementioned problems, we propose CNEG, a novel **C**onditional **N**on-Autoregressive **E**rror **G**eneration (CNEG) model for generating Chinese grammatical errors. Figure 3 illustrates the architecture of the model. Specifically, to predict a context-dependent error, we first mask a span of a correct text, and utilize BERT (Devlin et al., 2019) to conduct non-autoregressive span prediction. In order to ensure that the generated sentence will be faithful to the original sentence, we force the model to generate span conditioned on the original span. Considering that the correct information is integrated into the model, we further introduce a penalty to encourage the model not to directly reconstruct the correct span. Our CNEG model is based on BERT, which is pre-trained on a large scale of Chinese corpus. Therefore, the model can generate errors that do not appear in the training dataset. Finally, in order to filter out the error-free spans, we also utilize a pre-trained BERT to measure the perplexities of generated spans.

The main contributions of this paper can be summarized as follows:

- We propose a new data augmentation method (CNEG) to tackle the data scarcity of CGED. We utilize BERT encoder with a non-autoregressive decoding layer as the backbone of our generative model to generate context-dependent errors.
- We incorporate the original span into our generative model, which enables the model to predict the erroneous span conditioned on the original span. And we introduce a filtering strategy to filter out error-free spans.
- Experimental results on the CGED datasets show that our method outperforms all previous methods, which demonstrates the effectiveness of our method. We release the source code for further use by the community[1].

2 Related Work

Chinese Grammatical Error Detection (CGED) aims at detecting grammatical errors in Chinese sentences (Rao et al., 2018). Most studies regard it as a sequence tagging task, where each token will be given a correct label or an error-type. Sequence labeling methods are widely used for CGED, such as feature-based statistical models (Chang et al., 2012), and neural models (Fu et al., 2018). Due to the effectiveness of BERT(Devlin et al., 2019) in many other NLP applications, recent studies adopt BERT as the basic architecture of CGED models (Fang et al., 2020; Wang et al., 2020b; Li and Shi, 2021). Wang et al. (2020b) propose a model that

[1] https://github.com/tc-yue/DA_CGED

combines ResNet and BERT to achieve state-of-the-art results on the CGED-2020 task. Li and Shi (2021) apply a CRF layer on BERT to introduce the dependency of tokens.

However, neural models usually require a large amount of training data, and manually annotating a large corpus is expensive and time-consuming. Therefore, many studies focus on data augmentation methods to automatically generate large-scale training samples to boost the performance of grammatical error detection models (Kiyono et al., 2019; Wang et al., 2019; Lichtarge et al., 2019; Kasewa et al., 2018). Kiyono et al. (2019) investigated different strategies of the incorporation of pseudo data, including the method of generating the pseudo data, the seed corpus for augmentation, and training strategies with these augmented samples. Wang et al. (2019) proposed a rule-based editing method that constructs the noise-corrupted text. Instead of directly adding noise into the sentence, Wan et al. (2020) introduce noise to the representation of a sentence and apply the Seq2Seq model to generate sentences with various error types. Lichtarge et al. (2019) use an intermediate language as a bridge to generate grammatical error samples. Zhou et al. (2020) consider that Neural Machine Translation (NMT) model is significantly better than the Statistical Machine Translation (SMT) model, then utilize NMT model and SMT model to generate correct and erroneous sentences respectively. Moreover, Wang and Zheng (2020) firstly identify the most vulnerable tokens by a seq2seq model, then replace these tokens with the grammatical errors which are collected from the training dataset.

3 Methodology

3.1 Problem Formulation

Our goal is to generate high-quality grammatical errors to improve the performance of CGED models. Given a sample $S = (E, C, Y)$ from CGED training dataset, where $E = [e_1, e_2, ..., e_m]$ is an erroneous sentence of m tokens. Each token e_i is assigned with a label $y_i \in \{0, ..., d\}$, where d is the number of error types and 0 represents non-error. $C = [c_1, c_2, ..., c_n]$ is the corresponding corrected text of n tokens. The goal of data augmentation method is to generate erroneous sentences based on the correct sentence C and the erroneous sentence E. And the goal of grammatical error detection model is to predict the label y_i of each token e_i.

In the following subsections, we first present the architecture of our generative model, as described in §3.2, then introduce the method of constructing erroneous sentences with the trained model, as described in §3.3.

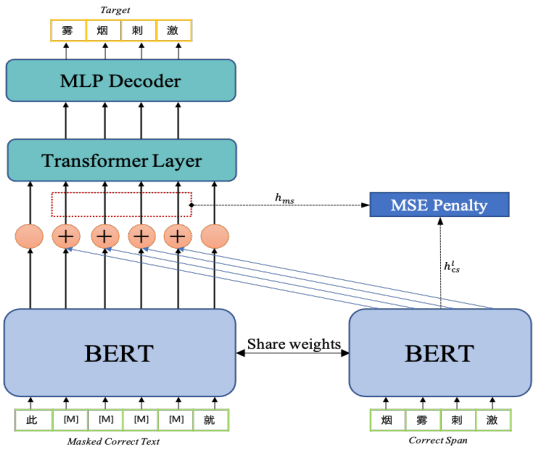

Figure 3: The architecture of our CNEG model.

3.2 CNEG Model

Figure 3 illustrates the architecture of the proposed CNEG model. To imitate human errors, our model first masks a span in a correct text, then predicts the erroneous span conditioned on the masked context and the correct span. In this subsection, we first describe the training samples for the generative model, then present the architecture of the model, finally introduce the learning objectives.

Training Samples Construction Given an erroneous sentence and its corresponding correct sentence, we collect the erroneous spans and their corresponding correct spans. Then we sample an erroneous span E_{span} of n_e tokens, and its corresponding correct span C_{span} of n_c tokens. As shown in Figure 3, the target of the model is the erroneous span, and the inputs of the model are the correct span and the masked correct text. To get the masked correct text C_{masked}, we replace the correct span C_{span} in the correct text C with a masked span M_{span} consisting of n_m `[MASK]` tokens, where $n_m \geq n_e$ and $n_m \geq n_c$. Since the erroneous span E_{span} and the correct span C_{span} may be not aligned in token level (e.g. E_{span} = "而于", C_{span} = "而终于"), the model can hardly learn the token-level mappings of those span. To handle this problem, we propose a strategy to align them. Assuming $n_m = 4$:

1. When $n_e = n_c$ (e.g. E_{span} = "死去的", C_{span} = "死亡的"), we pad one special token

[U] to the tail of C_{span} and the tail of E_{span} separately:

$$E_{span} = [死, 去, 的, [U]]$$

$$C_{span} = [死, 亡, 的, [U]]$$

2. When $n_e > n_c$ (e.g. E_{span} = "终于了", C_{span} = "终于"), which means that some tokens can be added to the tail of correct span, we pad two [U] tokens to the tail of C_{span} and one [U] token to the tail of E_{span}:

$$E_{span} = [终, 于, 了, [U]]$$

$$C_{span} = [终, 于, [U], [U]]$$

3. When $n_e < n_c$ (e.g. E_{span} = "而于", C_{span} = "而终于"), which means that some tokens can be deleted from C_{span}. Then, we insert one [U] into the missing position of E_{span}, and pad one [U] to each span:

$$E_{span} = [而, [U], 于, [U]]$$

$$C_{span} = [而, 终, 于, [U]]$$

where [U] is a placeholder which means no character in the position.

Conditional Context Representation Our architecture adopts BERT (Devlin et al., 2019) as the basic encoding model, which is initialized with a pre-trained Chinese BERT (Cui et al., 2019) to make full use of linguistic information from large-scale Chinese texts. BERT is constructed with a stacked layer structure, which has deep bidirectional representations by learning information from left to right and from right to left.

To predict the erroneous span conditioned on the original context, we use BERT to encode the masked correct text C_{masked} to obtain contextual representations of the masked span M_{span}, denoted as $\mathbf{h^l_{ms}}$, where l is the number of BERT layers. Previous masked language model applies an MLP decoder on this vector to conduct non-autoregressive prediction. However, the predicted sequence may be quite different from the original span.

To alleviate this problem, we propose a conditional component to incorporate the correct span. Specifically, we apply the same BERT to encode the correct span C_{span} and get corresponding hidden vectors, denoted as $\mathbf{h^l_{cs}}$. Then we add this vector to the representation of the masked span:

$$h_{ms} = h^l_{ms} + h^l_{cs} \quad (1)$$

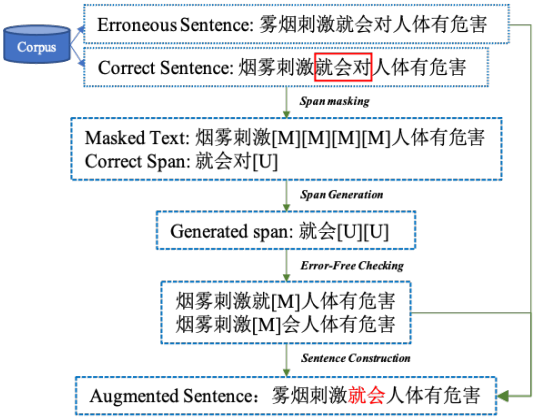

Figure 4: Data flow of the erroneous sentence construction. Token [U] in correct span is padding tokens to make the length of the correct span equal to that of the masked span.

As show in the left part of Figure 3, we further apply a transformer layer on the new representation, therefore the masked span representation $\mathbf{h^{l+1}_{ms}}$ is conditioned on both the context C_{masked} and the correct span C_{span}. Finally, we apply a MLP layer and a softmax layer to transform the vector $\mathbf{h^{l+1}_{ms}}$ to the generative probability p, it is defined as:

$$p = softmax(Wh^{l+1}_{ms} + b) \quad (2)$$

We adopt cross entropy loss as the objective function:

$$L_{MLM} = -\sum_{i=1}^{n_m}\sum_{j=1}^{c} logy^i_j logp^i_j \quad (3)$$

where c is the size of vocabulary and n_m is the length of masked span.

MSE Penalty As we integrate the original span into the model by Eq.1, the model will tend to directly reconstruct the correct span when $\mathbf{h_{ms}}$ and $\mathbf{h^l_{cs}}$ are too similar. To lead the model not to pay all attention to the correct span, we add a penalty to force $\mathbf{h_{ms}}$ to be different from $\mathbf{h^l_{cs}}$ by maximizing the distance between two vectors:

$$L_{MSE} = -MSE(h_{ms}, h^l_{cs}) \quad (4)$$

where MSE means the mean squared error loss function. Then the final loss of the model is:

$$Loss = \lambda \cdot L_{MSE} + L_{MLM} \quad (5)$$

where λ is a hyperparameter.

Algorithm 1 Erroneous sentence construction

Input:
f: CNEG model
C: a correct text of n tokens
E: an erroneous text
T: a threshold to filter error-free span
M_{span}: a span of `[MASK]` tokens
Output:
A: augmented dataset

1: Set length of masked correct span as n_c;
2: Initialize an empty mapping *M*={ }
3: **for** $i \in [0, n - n_q]$ **do**
4: Get a correct span $C_{span} = C[i : i + n_c]$
5: Form a masked text $C_{masked} = C[: i] + M_{span} + C[i + n_c :]$
6: Predict $G_{span} = f(C_{masked}, C_{span})$
7: Add (C_{span}, G_{span}) into mapping *M*
8: **for** $C_{span}, G_{span} \in M$ **do**
9: **if** $PPL(G_{span}) < T$ or $PPL(G_{span}) < PPL(C_{span})$ **then**
10: continue
11: **if** $C_{span} \in E$ **then**
12: Replace C_{span} in E with G_{span} and form a synthetic sentence S
13: Get the label sequence Y of S
14: Add (S, Y) to A
15: **return** A

3.3 Erroneous Sentence Construction

In this subsection, we describe our method of constructing erroneous sentences. As the example shown in Figure 4, we first mask a span in the correct text and generate a span with the trained model, then check if the span contain grammatical errors, finally we use the erroneous span to construct the erroneous sentence.

Erroneous Span Generation In this step, we utilize the trained model to generate grammatically erroneous spans for a correct text. Specifically, given an erroneous text and its corrected text, we first initial an empty correction-to-error mapping M, and mask a span within the correct text, then feed the correct span C_{span} and the masked correct text C_{masked} to the CNEG model to generate a span G_{span}, finally add the C_{span} and G_{span} pair to the mapping. Since the span masking can be conducted like a sliding window, we will get a correction-to-error mapping for each correct text (lines 3-7 in Algorithm 1).

Error-free Span Filtering Although our CNEG model takes the erroneous spans as the predicting targets, we cannot ensure that each generated span will contain at least one grammatical error. If we assign error-types to error-free spans, they will become noises for the detection model later. Therefore, it is necessary to filter out the error-free spans. Mita et al. (2020) compare the perplexities of generated sentences and correct sentences to determine whether the generated sentences are grammatically correct. However, since the sentence-level perplexity is affected by too many tokens, the sentence with larger perplexity may also be grammatically correct. To address this issue, we introduce a method that uses span-level perplexity to identify whether the generated span is erroneous (lines 9-10 in Algorithm 1). To calculate $PPL(G_{span})$, we replace the masked span M_{span} in the masked correct text C_{masked} with the generated span G_{span}, and mask the word w_i of the generated span one by one, then utilize pre-trained BERT to predict the probability $P(w_i)$ of the masked word w_i :

$$P(w_i) = P(w_i|w_1, ..., w_{i-1}, w_{i+1}, ..., w_N) \quad (6)$$

We calculate the perplexity of the generated span $PPL(G_{span})$ by the following equation:

$$PPL(G_{span}) = exp\{-\frac{1}{N}\sum_{i=1}^{N} P(w_i)\} \quad (7)$$

Where N is the length of the generated span. We use the same method to calculate the perplexity of the correct span $PPL(C_{span})$. Then we can filter out the generated span whose perplexity is smaller than corresponding $PPL(C_{span})$ and smaller than a threshold T, where T is a hyper-parameter. Finally, we will obtain a high-quality correction-to-error mapping for a correct text.

Synthetic Sentence Construction After obtaining the erroneous span, we can construct a training sample for CGED (lines 11-14 in Algorithm 1). Specifically, given an erroneous sentence E from training dataset, we select a generated span G_{span} and a corresponding correct span C_{span} from the mapping. If the erroneous sentence E contains the correct span C_{span}, we will replace the correct span C_{span} with the generated span G_{span} to form a synthetic sentence S. Then we use a rule-based method to automatically annotate the synthetic sentence S to obtain a label sequence Y. Finally, we add the sample (S, Y) to the augmented dataset.

Dataset	S	C	E	E_{span}
Train	21582	41	21541	53940
Validation	3154	1174	1980	4871
Test-2018	3546	1562	1984	5040
Test-2020	1457	307	1150	3660

Table 1: Distribution of datasets. S, C, E and E_{span} denote the amount of sentences, the amount of correct sentences, the amount of erroneous sentences and the amount of erroneous spans, respectively. Test-2018 and Test-2020 denote the test dataset of CGED-2018 and the test dataset of CGED-2020, respectively.

4 Experimental setup

4.1 Datasets

We conduct experiments on public datasets from CGED tasks (Lee et al., 2016; Rao et al., 2017, 2018, 2020), which contain thousands of Chinese text written by foreign language learners. Following the work of (Wang et al., 2020b), we select 2016, 2017, 2018 and 2020 training dataset as our training dataset.

CNEG Model We use error-correction sentence-pairs from the training dataset to train the generative model. Then we use the trained model to construct erroneous sentences by the same dataset.

CGED Model Each data augmentation method will generate some samples, we combine them with the training dataset to form a new dataset, which can be used for training the detection model later. For evaluating the performance of CGED model, we use the test dataset from CGED-2017 for validation, use the test dataset from CGED-2018 and the test dataset from CGED-2020 for testing separately. The statistics of datasets are given in Table 1.

4.2 Evaluation Metrics

We adopt the same evaluation method as used in (Rao et al., 2018). It includes three levels:

- **Detection level**. This level is to detect whether a sentence contains error, and can be considered as a binary-classification of a sentence.
- **Identification level**. This level is to identify all error-types of a sentence, and can be considered as a multi-label classification of a sentence.
- **Position level**. This level is to locate the erroneous words and identify their corresponding error types. However, there is no explicit word boundary in Chinese text, we measure this score on Chinese character-level in our experiment.

We use F1-score to measure each level.

4.3 Implementation Details

CNEG Model: The BERT encoder of our generative model is initialized with a Chinese BERT (Cui et al., 2019), which is also used for measuring the perplexities of generated spans later. We use the Adam optimizer with an initial learning rate of $5e^{-5}$ and train the generative model for 10 epochs. The λ in Eq. 5 is set to 0.5, and the threshold T in Algorithm 1 is set to 2.
CGED Model: We evaluate various data augmentation methods by training the BERT-based sequence labeling models on the augmented datasets. To predict the label of each token, we apply a fully-connected layer to perform token classification based on the representation of the last transformer layer, and the hidden size of the classification layer is 768. For all experiments, we use the Adam optimizer with an initial learning rate of $7e^{-5}$. All experiments are conducted for 5 runs and the averaged score is reported.

4.4 Compared Methods

We compare our augmentation method with several baseline methods.
Raw is the original training dataset without any other augmented samples.
DirectNoise (Wang et al., 2019) is an editing based method that introduces noise into a text by inserting, deleting or replace some words.
Seq2seq (Kasewa et al., 2018) takes the corrected sentences as the inputs and the erroneous sentences as the predicting targets of the model.
BackTranslation (Lichtarge et al., 2019) first translates the original sentence into a bridge language, the translated sentence will be translated back into the source language. In this experiment, we select English as the bridge language .
ADV (Wang and Zheng, 2020) is an adversarial method that constructs adversarial examples by targeting the weak spots of the models and replacing these weak tokens by correction-to-error mapping.
CNEG is our proposed augmentation method that first generates context-dependent erroneous spans, and constructs erroneous sentences.
CNEG w/o Filter is a variation of our method that constructs erroneous sentence without error-free span filtering strategy, as proposed in §3.3.

Method	CGED-2018			CGED-2020		
	D-F	I-F	P-F	D-F	I-F	P-F
Raw	80.66	64.93	49.77	87.39	60.27	32.78
DirectNoise	79.20	63.06	48.02	88.91	59.11	31.37
Seq2Seq	79.81	63.26	49.49	86.76	58.29	31.40
BackTranslation	80.20	64.01	48.81	87.03	59.89	31.92
ADV	80.71	64.79	50.10	87.11	60.20	32.81
CNEG (ours)	**80.9**	**66.88**	**52.26**	**88.12**	**62.00**	**33.99**
CNEG w/o Filter (ours)	80.47	66.37	51.92	87.03	59.16	33.14

Table 2: Main results on the CGED datasets. The best results are **in bold**. CGED-2018 denotes the test dataset of CGED-2018. CGED-2020 denotes the test dataset of CGED-2020. D-F denotes the F-score of detection-level. I-F denotes the F-score of identification level. P-F denotes the F-score of Position-level.

5 Experimental Results

5.1 Main Results

The experimental results on the CGED datasets are shown in Table 2. Our observations are as follows:

Whole-sentence generation methods degrade the performance on both of the test datasets. *Seq2Seq* and *BackTranslation* get worse results than *Raw* dataset because they treat the erroneous sentence generation as a whole sentence generation task, which is not controllable. By comparing our *CNEG w/o Filter* with *Seq2Seq*, we observe that span-generation method improves about 2.3% on the position-level of CGED-2018, and 1.7% on the position-level of CGED-2020.

Context-dependent errors are beneficial. Although *DirectNoise* shows effectiveness in some previous studies, it has no effect on the CGED dataset because the randomly introduced errors are context-independent, which are too easy for the detection model to detect such errors. Among the compared methods, *ADV* performs the best because it constructs errors considering about the contextual information. Even without filtering strategy, *CNEG w/o Filter* outperforms *ADV* by a large margin because it can generate more diversified errors, improving position-level F-score by 2.2% and 1.1% on the two test datasets.

Error-free filtering is necessary. We observe that *CNEG* further improves *CNEG-filter* by 0.6% on the position-level of CGED-2020. Without filtering strategy, the performance on detection-level shows significant decline. The reason is that the noisy augmented data can hurt the model performance. This result demonstrates the effectiveness of filtering out error-free span.

Method	D-F	I-F	P-F
CNEG (ours)	80.9	**66.88**	**52.26**
CNEG w/o Con	79.59	66.03	51.31
CNEG w/o Pen	**81.32**	65.83	51.86

Table 3: Ablation results on the CGED-2018.

Method	Sentence
Correct	烟雾刺激就会对人体有危害。
CNEG (ours)	雾烟刺激就会人体有危害。
CNEG w/o Con	雾烟刺激真的是对人体有危害。
CNEG w/o Pen	雾烟刺激就会对人体有危害。

Table 4: Examples generated by the models. The masked correct span are marked in green. The generated spans are marked in red. Errors from the original erroneous sentence are marked in blue.

5.2 Effects of Components of Generative Model

For further analyzing the effectiveness of the components of our proposed model, we also conduct ablation experiments as follows:

CNEG w/o Con is a variation of our model that predicts error not conditioned on the original span, which is described in §3.2.

CNEG w/o Pen is a variation of our method that trains generation model without the MSE penalty , which is described in §3.2.

Results are shown in Table 3. Experimental results show that *CNEG* significantly performs better than *CNEG w/o Con* and *CNEG w/o Penalty*. We also present several generated sentences in Table 4. *CNEG w/o Con* generates a grammatical error, which is quite different from the original span and should be corrected to "真的对", and we should assign a *redundant* label to the token "是". However, when comparing the generated span with the

Method	D-F	I-F	P-F
PSE	80.2	64.98	50.21
GME	**81.2**	65.62	50.94
PME	80.9	**66.88**	**52.26**

Table 5: Results of different sentence construction methods on CGED-2018.

original span, *selection-error* labels are automatically assigned to the tokens in the generated span, which will confuse the detection model. *CNEG w/o Pen* directly reconstructs the original span, which is useless for data augmentation. Our methods generates a grammatical error by missing an important token in the original span, which is beneficial for the detection model. These results demonstrate the effectiveness of our proposed components.

5.3 Effects of Multi-Error Sentences

Our augmentation method masks a span in a correct sentence and then predicts an erroneous span. To construct an erroneous sentence, the direct method is to replace the masked span with the predicted span, then the synthetic sentence will contain an erroneous span, we call this method *PSE* (Plug-in Single-Error). However, each sentence in CGED dataset contains over two errors on average, as shown in the Table 1. To make the synthetic sentences be consistent with multi-error sentences, we develop two multi-error sentences construction strategies. First, as described in §3.3, we locate the correct span in the corresponding erroneous sentence and replace it with the erroneous span. The synthetic sentence will contain original errors and a generated error, we call this method *PME* (Plug-in Multi-Error). Second, we mask a correct span in an erroneous sentence, and utilize the model to predict an erroneous span. Then the new sentence will contain the original errors and a generated error, we call this method *GME*(Generated Multi-error). To figure it out which is the better choice, we conduct experiments on the datasets augmented by those methods. We report the results in Table 5. We observe that the *PSE* gets the worst performance. The reason is that single-error is too easy for the detecting model. *PME* outperforms *GME*, the reason may be that *GME* can not predict beneficial spans with the noisy context. Therefore, we can conclude that inserting the erroneous span into the original erroneous sentence is the most effective method.

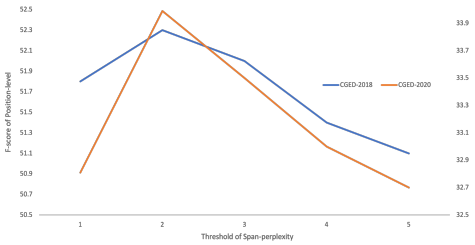

Figure 5: Performance of data augmentation with different filter threshold. The left axes is for CGED-2018, the right axes is for CGED-2020.

	Sentence
(a)	从小就是(形象)形影不离的一对。
(b)	从小就是(内容)形影不离的一对。
(c)	第二天(变)天气变得很好。
(d)	第二天(给)天气变得很好。

Table 6: Constructed examples. (a) and (c) are generated by our model. (b) and (d) are generated by direct noise method. Errors are marked in red.

5.4 Effects of Different Threshold For Filtering Strategy

Results on Table 2 show that with the help of filtering strategy, *CNEG* can further improve by 1% over *CNEG w/o filter*. In this subsection, to further evaluate the effectiveness of our filtering strategy, we set different filtering thresholds to construct several augmentation datasets, then train detection models with these datasets. The evaluation results are show in Figure 5.

We can observe that when threshold is around 2, the method achieves the best performance on both the CGED-2018 and CGED-2020. When the threshold is lower than 2, the performances of detection model decrease significantly. The reason is that there are many error-free spans whose perplexities are lower than 2, when these error-free spans are added into the training dataset, the detection model will be confused. When the threshold is higher than 4, the methods also achieve worse performance. The reason is that most generated errors are filtered out, the reserved erroneous spans are too limited for boosting the performance of detection models.

5.5 Case Study

As we demonstrated, our model can better imitate human grammatical errors. In Table 6, we list some augmented examples. The first two sentences are selection errors, sentence (a) replaces

"形影" with a near-synonym "形象", sentence (b) replaces "形影" with a random noun "内容". The last two sentences are redundant errors, sentence (c) inserts "变" in front of "天气" where "变天气" is a phrase but not correct for here, sentence (d) inserts a random verb "给" in front of "天气" to generate a obviously redundant error. Unlike human who usually makes context-dependent errors, the direct noise method always introduces random errors, while our model generates highly context-dependent errors. Hence, our method can generate high quality and diverse errors which could not constructed by direct noise method.

6 Conclusions

In this paper, considering that grammatical errors made by humans are usually context-dependent, we propose a conditional non-autoregressive error generation method (**CNEG**) for data augmentation of CGED. By introducing the correct span into the non-autoregressive model, the model will generate errors conditioned on both the context and the correct span. Observing that the model may generate correct spans, a filtering strategy is proposed to filter out error-free spans. Experimental results show that our method outperforms all compared data augmentation methods on the CGED datasets, which demonstrates the effectiveness of our method.

References

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*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Ru-Yng Chang, Chung-Hsien Wu, and Philips Kokoh Prasetyo. 2012. Error diagnosis of chinese sentences using inductive learning algorithm and decomposition-based testing mechanism. *ACM Trans. Asian Lang. Inf. Process.*, 11(1):3:1–3:24.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese BERT. *CoRR*, abs/1906.08101.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62, Montréal, Canada. Association for Computational Linguistics.

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: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Meiyuan Fang, Kai Fu, Jiping Wang, Yang Liu, Jin Huang, and Yitao Duan. 2020. A hybrid system for NLPTEA-2020 CGED shared task. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 67–77, Suzhou, China. Association for Computational Linguistics.

Ruiji Fu, Zhengqi Pei, Jiefu Gong, Wei Song, Dechuan Teng, Wanxiang Che, Shijin Wang, Guoping Hu, and Ting Liu. 2018. Chinese grammatical error diagnosis using statistical and prior knowledge driven features with probabilistic ensemble enhancement. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, NLP-TEA@ACL 2018, Melbourne, Australia, July 19, 2018*, pages 52–59. Association for Computational Linguistics.

Jianfeng Gao, Xiaolong Li, Daniel Micol, Chris Quirk, and Xu Sun. 2010. A large scale ranker-based system for search query spelling correction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 358–366, Beijing, China. Coling 2010 Organizing Committee.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a right: Generating better errors to improve grammatical error detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4977–4983. Association for Computational Linguistics.

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, Hong Kong, China, November 3-7, 2019*, pages 1236–1242. Association for Computational Linguistics.

Lung-Hao Lee, Gaoqi Rao, Liang-Chih Yu, Endong Xun, Baolin Zhang, and Li-Ping Chang. 2016. Overview of NLP-TEA 2016 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pages 40–48, Osaka, Japan. The COLING 2016 Organizing Committee.

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.

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.

Masato Mita, Shun Kiyono, Masahiro Kaneko, Jun Suzuki, and Kentaro Inui. 2020. A self-refinement strategy for noise reduction in 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 267–280. Association for Computational Linguistics.

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: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Gaoqi Rao, Qi Gong, Baolin Zhang, and Endong Xun. 2018. Overview of NLPTEA-2018 share task chinese grammatical error diagnosis. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, NLPTEA@ACL 2018, Melbourne, Australia, July 19, 2018*, pages 42–51. Association for Computational Linguistics.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020. Overview of NLPTEA-2020 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 25–35, Suzhou, China. Association for Computational Linguistics.

Gaoqi Rao, Baolin Zhang, Endong Xun, and Lung-Hao Lee. 2017. IJCNLP-2017 task 1: Chinese grammatical error diagnosis. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 1–8, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Zhaohong Wan, Xiaojun Wan, and Wenguang Wang. 2020. Improving grammatical error correction with data augmentation by editing latent representation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2202–2212. International Committee on Computational Linguistics.

Haoyu Wang, Shuyan Dong, Yue Liu, James Logan, Ashish Kumar Agrawal, and Yang Liu. 2020a. ASR Error Correction with Augmented Transformer for Entity Retrieval. In *Proc. Interspeech 2020*, pages 1550–1554.

Liang Wang, Wei Zhao, Ruoyu Jia, Sujian Li, and Jingming Liu. 2019. Denoising based sequence-to-sequence pre-training for text generation. 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 4001–4013. Association for Computational Linguistics.

Lihao Wang and Xiaoqing Zheng. 2020. Improving grammatical error correction models with purpose-built adversarial examples. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2858–2869. Association for Computational Linguistics.

Shaolei Wang, Baoxin Wang, Jiefu Gong, Zhongyuan Wang, Xiao Hu, Xingyi Duan, Zizhuo Shen, Gang Yue, Ruiji Fu, Dayong Wu, et al. 2020b. Combining resnet and transformer for chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 36–43.

Rongxiang Weng, Heng Yu, Xiangpeng Wei, and Weihua Luo. 2020. Towards enhancing faithfulness for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2675–2684, Online. Association for Computational 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.

Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. 2020. Improving grammatical error correction with machine translation pairs. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 318–328. Association for Computational Linguistics.