Adversarial Grammatical Error Correction

Vipul Raheja **Dimitris Alikaniotis**
Grammarly
firstname.lastname@grammarly.com

Abstract

Recent works in Grammatical Error Correction (GEC) have leveraged the progress in Neural Machine Translation (NMT), to learn rewrites from parallel corpora of grammatically incorrect and corrected sentences, achieving state-of-the-art results. At the same time, Generative Adversarial Networks (GANs) have been successful in generating realistic texts across many different tasks by learning to directly minimize the difference between human-generated and synthetic text. In this work, we present an adversarial learning approach to GEC, using the generator-discriminator framework. The generator is a Transformer model, trained to produce grammatically correct sentences given grammatically incorrect ones. The discriminator is a sentence-pair classification model, trained to judge a given pair of grammatically incorrect-correct sentences on the quality of grammatical correction. We pre-train both the discriminator and the generator on parallel texts and then fine-tune them further using a policy gradient method that assigns high rewards to sentences which could be true corrections of the grammatically incorrect text. Experimental results on FCE, CoNLL-14, and BEA-19 datasets show that Adversarial-GEC can achieve competitive GEC quality compared to NMT-based baselines.

1 Introduction

Grammatical Error Correction (GEC) has grown into a popular NLP task that deals with building systems for automatically correcting errors in written text (Ng et al., 2013, 2014). Evolving from the approaches of building error-specific machine learning classifiers (Tetreault and Chodorow, 2008; De Felice and Pulman, 2008; Tetreault et al., 2010; Dahlmeier and Ng, 2011; Rozovskaya and Roth, 2014), it has gained popularity as a monolingual Machine Translation (MT) problem, where the system learns to "translate" a given erroneous text to its corrected form (Brockett et al., 2006; Felice et al., 2014; Susanto et al., 2014). Initially, Statistical phrase-based Machine Translation (SMT) techniques were successfully applied to the task (Yuan and Felice, 2013; Junczys-Dowmunt and Grundkiewicz, 2016; Yuan et al., 2016) as as a way to handle all error types concurrently. More recently, several Neural Machine Translation (NMT) systems have been developed with promising results (Sutskever et al., 2014; Bahdanau et al., 2015; Cho et al., 2014), and their successful application to GEC, either in combination with SMT models (Chollampatt et al., 2016; Yuan and Briscoe, 2016; Yannakoudakis et al., 2017; Grundkiewicz and Junczys-Dowmunt, 2018), or strictly as neural models, has emerged as the new state-of-the-art (Xie et al., 2016; Schmaltz et al., 2017; Sakaguchi et al., 2017; Ji et al., 2017; Ge et al., 2018; Junczys-Dowmunt et al., 2018; Chollampatt and Ng, 2018a,b; Zhao et al., 2019).

Despite the successes of NMT-based models for GEC, a major challenge still lies in the definition of the evaluation metrics. Ideally, the metric should be able to quantify the (a) lexical overlap, (b) semantic similarity, and (c) grammaticality of a generated sentence, given a grammatically incorrect input sentence. In a straightforward application of NMT-based models to the GEC task, one would minimize a surrogate loss (e.g., cross-entropy), which is an upper bound on the true loss, and hence a loose approximation of these complex criteria. Moreover, NMT-based GEC models try to maximize n-gram or edit-based metrics, such as M^2 (Dahlmeier and Ng, 2012), I-Measure (Felice and Briscoe, 2015), or GLEU (Napoles et al., 2015) pushing the NMT-based models to generate sentences with n-gram precisions as high as possible, which may not necessarily lead to high-quality generation for the GEC task. In order to avoid these issues, we take a different approach, inspired by Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), which

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3075–3087
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

provide a framework that can be leveraged to directly model the task based on the differences in the input-output distributions and the complex criteria mentioned above. Moreover, GANs have shown remarkable ability to generate coherent and semantically meaningful text in many natural language processing tasks such as machine translation (Wu et al., 2018; Yang et al., 2018), dialogue generation (Li et al., 2017), and abstractive summarization (Liu et al., 2018; Wang and Lee, 2018) among others.

We propose a GAN-based generator-discriminator framework for grammatical error correction. The generator is a Sequence-to-Sequence (Seq2Seq) model, which is trained to "translate" a grammatically incorrect sentence to its grammatically correct rewrite. The discriminator, a deep neural sentence-pair classification model is trained to evaluate the probability of the generated sentence being a lexically-similar, meaning-preserving, and grammatically correct rewrite of the incorrect input sentence. Adversarial training between the two models is set up as optimizing a *min-max* objective, where the discriminator learns to distinguish whether a given input is sampled from the ground-truth (*human*-generated) or generator (*artificially*-generated) distributions, *maximizing* the difference between them. The generator, on the other hand, learns to trick the discriminator by producing high-quality correction candidates, thus, *minimizing* the difference between its output and a ground-truth corrected sentence. Further, the discriminator is used to fine-tune the generator using a policy gradient (Williams, 1992; Yu et al., 2017; Wu et al., 2018), rewarding high quality generated text when conditioned on the source, improving, thus, the generation results. By minimizing the difference between the human- and the artificially-generated distribution, we aim at directly optimizing the task based on the criteria mentioned above.

We evaluate the effectiveness of our approach on three standard datasets on the task, observing that the discriminator can provide reasonably consistent guidance to the generator and further help improve its performance. Experimental results indicate that our model can achieve significantly better performance than strong NMT-based baselines.

In summary, we make the following contributions:

- This work is, to the best of our knowledge, the first to apply generative adversarial training to the GEC task.

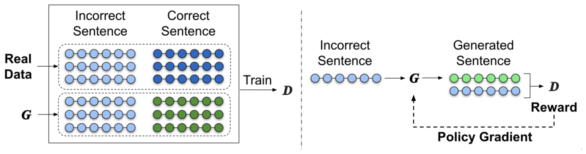

Figure 1: Adversarial-GEC training. **Left**: D is trained over the real and the generated data by a pre-trained G. **Right**: G is further trained by policy gradient where the final reward is provided by D and is passed back to the generator.

- We propose a sentence-pair classification-based discriminator, that can better distinguish grammatical text from ungrammatical text by learning to directly optimize the task rather than constructing or relying on n-gram or edit-based metrics. We analyze different formulations of the discriminator, and provide insights into how its setup, pre-training and integration into the framework can be leveraged for stable training and better performance.

- We conduct extensive experiments on standard GEC datasets and evaluate the system against strong baselines, showing that the proposed model consistently achieves better results in a self-contained single model setting, without relying on any resources other than just the training data.

2 Adversarial GEC

Fig. 1 outlines our approach which consists of two components the (a) Generator (G) and (b) Discriminator (D).

2.1 Generator

Following recent NMT-based state-of-the-art GEC systems, we treat a grammatically incorrect sentence as the source and its grammatically corrected counterpart as the target. Formally, given a sequence $x = [x_1, x_2, \ldots, x_S]$, we aim to generate another sequence $y = [y_1, y_2, ..., y_T]$ which is the grammatically corrected form of x. We denote a pair of incorrect-correct sentences as (x, y). Given a sequence x, the generator learns to produce another sequence $y' \approx y$.

While the generator can be any Seq2Seq model, we use two common Encoder-Decoder architectures for GEC; an attention-based RNN (Luong et al., 2015) and a Transformer (Vaswani et al., 2017).

2.2 Discriminator

In this framework, a critical component is a discriminator that is responsible for providing the appropriate reward to the generator based on the quality of the generated text. Most GAN architectures typically use a single-sentence *real*-vs-*fake* classifier[1] as the discriminator (Yu et al., 2017). However, we argue that such a formulation does not accurately express the GEC task objective. A conventional GAN discriminator would provide the probability of a sentence being grammatically correct as the reward. However, it would be especially harder for such a classifier to differentiate between a ground-truth correction and a generated sentence that fits the distribution of real-world text and is far from the generated data, but does not make the intended corrections or changes the semantics of the source. Moreover, it would also be unable to provide a proportionate reward to a partially corrected sentence. Due to the lack of contextual knowledge about *what* has been corrected, such a classifier would struggle to differentiate between low-quality or unsuitably corrected sequences. Consequently, it will end up giving them rewards comparable to sentences which are truly the corrected forms of given incorrect source sentences.

In the GEC task, we ultimately want the generator to generate corrected sentences that fit the constraints mentioned in Section 1. Hence, we formulate the objective of the discriminator as being two-fold: first, to be able to evaluate the quality of the generated text in terms of its validity compared to the ground-truth distribution, and second, to measure its quality as the appropriate rewrite for a given input sentence. In summary, the discriminator needs to be able to measure the degree of "grammatical correctness" of an output sentence, given its corresponding input sentence, instead of only distinguishing between *real*-vs-*fake* Therefore, instead of training a single-sentence classifier, we train on incorrect-correct sentence pairs. We consider ground-truth data (x, y) as high-quality corrections (positive examples), while data sampled from the generator (x, y') as low-quality (negative examples). We experiment with two discriminator models for both the single-sentence and sentence-pair formulations: CNN- and RNN-based due to their simplicity, widespread use in sentence-pair modeling tasks, and ease of implementation.

2.3 Adversarial Training

Adversarial training between G and D (parameterized by θ and ϕ, respectively) is set up as optimizing a *min-max* objective, formulated as the following objective function $V(G_\theta, D_\phi)$:

$$\begin{aligned}&\min_{\theta}\max_{\phi} V(G_\theta, D_\phi) \\ &= \mathbb{E}_{(x,y)\sim P_{data}}[\log D_\phi(x,y)] + \\ &\quad \mathbb{E}_{x\sim P_{data},\, y'\sim P_{G_\theta(\cdot|x)}}[\log(1 - D_\phi(x,y'))]\end{aligned} \tag{1}$$

where P_{data} is the underlying training data distribution and $P_{G_\theta(\cdot|x)}$ is the distribution of the generator output.

With this objective function, the discriminator learns to predict whether a given sentence pair has been sampled from the ground-truth data (x, y) or from G_θ: (x, y'). G_θ tries to confuse D_ϕ by generating high-quality corrected samples $y' \approx y$, given a ground-truth input sentence x. Formally, the objective function of D_ϕ is defined as the standard binary cross entropy (BCE) loss:

$$\begin{aligned}\mathcal{L}_d = &\ \mathbb{E}_{(x,y)\sim P_{data}} \log D_\phi(x,y) \\ &+ \mathbb{E}_{x\sim P_{data}, y'\sim P_{G_\theta(\cdot|x)}} \log(1 - D_\phi(x,y'))\end{aligned} \tag{2}$$

The objective of the generator can be formulated as optimizing the following loss:

$$\mathcal{L}_g = \mathbb{E}_{x\sim P_{data}, y'\sim P_{G_\theta(\cdot|x)}} \log(1 - D_\phi(x,y')) \tag{3}$$

However, since the generator performs discrete sampling to obtain y', we cannot directly use the gradient-based approach to backpropagate the gradients, making $V(G_\theta, D_\phi)$ non-differentiable with respect to θ. To address this issue, borrowing from Cai and Wang (2018) and Wu et al. (2018), we use single-sample based REINFORCE (Williams, 1992), a Monte-Carlo policy gradient method to optimize G_θ. In Reinforcement Learning (RL) terms, the generator G_θ acts as the agent under the policy $G_\theta(\cdot|x)$, and the generated grammatically corrected sentence y' is the action. The environment is characterized via the input sequence x and the discriminator D_ϕ, which provides the reward $-\log(1 - D_\phi(x, y'))$ based on the discriminative loss of $D_\phi(x, y')$. The generator improves itself by maximizing the reward returned from the environment. The gradients $\nabla_\phi \mathcal{L}_d$ and $\nabla_\theta \mathcal{L}_g$ can thus be estimated by sampling a correction from the generator $y' \sim G(\cdot|x)$ as follows:

$$\nabla_\phi \mathcal{L}_d = \nabla_\phi \log D_\phi(x,y) + \nabla_\phi \log(1 - D_\phi(x,y')) \tag{4}$$

$$\nabla_\theta \mathcal{L}_g = \nabla_\theta \log G_\theta(y'|x) \log(1 - D_\phi(x,y')) \tag{5}$$

[1] In this context, *fake* would be a (grammatically) incorrect and *real* a (grammatically) correct sentence.

where ϕ and θ can be updated as per the REINFORCE algorithm.

2.4 Training Strategies

While REINFORCE provides a framework where the reward function does not have to be differentiable, the discrete reward space due to the use of a single sampled y' to perform the Monte Carlo estimation leads to the problem of high variance, resulting in unstable training - a widely acknowledged limitation of RL methods. In practice, we find that adversarially training the generator solely with Eq. 3 is unstable, even when it is pre-trained. This is due to the sparsity of the rewards provided to the generator, which happens only once it has fully generated a sentence. This is also compounded by the fact that we do not generate multiple samples for computational efficiency. Hence, the generator training becomes brittle and finds it extremely difficult to get out of bad local minima or mode collapse. To alleviate this issue, we leverage the following measures: baseline reward, and teacher forcing/interleaved training to train the generator.

Baseline Reward A popular technique to alleviate the variance issue is the subtraction of baseline values from the original rewards. The baseline reward could be computed using various approaches. Yang et al. (2018) use a constant value, Rennie et al. (2017) use the reward of the sequence obtained by the current model with a greedy sampling strategy, Ranzato et al. (2016), Bahdanau et al. (2017), and Liu et al. (2017) use an MLP to estimate the baseline reward. However, these methods rely on approximating the terminal reward using intermediate states, or incorporating word-level rewards via rollout strategies for better credit assignment. Moreover, such approaches have been found to be extremely time-consuming, given the large decoding space. Based on prior works on RL for modeling dialog systems, which also have discrete action-reward spaces (Sankar and Ravi, 2019; Su et al., 2015), we use a moving average of the historical reward values as the baseline, which stabilizes the training process and is computationally tractable.

Interleaved Training Following Guo et al. (2018) and Wu et al. (2018), we interleave MLE and Policy Gradient training. This combination of an adversarial objective with MLE is an important factor in successfully training G. By some probability λ (more details in Section 5.3), randomly chosen mini-batches are trained with the Policy Gradient (discriminator reward), while other mini-batches are trained using MLE. This alternation improves training stability, as MLE acts as a regularizer to ensure a smoother model update, alleviating the negative effects brought by high gradient estimation variance of the one-step Monte Carlo sample in REINFORCE. After this generator update, it is used to generate more realistic corrections, which are then used to train the discriminator. This approach is equivalent to the teacher forcing step in Li et al. (2017) and Yang et al. (2018), where, after every iteration of policy gradient training update, they update the generator using teacher forcing by making the discriminator automatically assign a reward of 1 to the ground-truth data, which is used by the generator to further update itself.

Split	Dataset	Sentences	Tokens
Train	FCE-train	27k	454k
	BEA19-train	34k	628k
	CoNLL14-train	57k	1.1M
	Lang-8	1M	13M
Dev	CoNLL13	1.3k	28k
	FCE-dev	1.9k	28k
	BEA19-dev	4.3k	87k
Test	CoNLL14-test	1.3k	30k
	FCE-test	2.4k	36k
	BEA19-test	4.4k	85k

Table 1: Dataset splits and sizes.

3 Experiments

3.1 Data

In line with previous works, we use the public NUCLE corpus (used in the CoNLL 2014 GEC Shared Task (Ng et al., 2014; Dahlmeier et al., 2013)), the FCE Corpus (Yannakoudakis et al., 2011), the Lang-8 Corpus of Learner English (Tajiri et al., 2012), and the Write & Improve and LOCNESS (W&I+L) dataset from the BEA 2019 Shared Task (Bryant et al., 2019; Granger, 1998), as our parallel training datasets. We use CoNLL-2013 (Ng et al., 2013), FCE-dev, and BEA19-dev as our development sets, and for our test splits,[2] we use the FCE-test, CoNLL-2014 (Ng et al., 2014) test, and the BEA19 test set (evaluated by ERRANT (Bryant et al., 2017)). We report $F_{0.5}$ scores evaluated by the M^2 scorer (Dahlmeier and Ng, 2012) for both of these test datasets.

[2] We could not use JFLEG (Napoles et al., 2015) corpus for evaluation due to licensing restrictions.

3.2 Baselines

We use the two generators introduced in Section 2.1 as baseline generators. Building on these baselines, we develop GAN frameworks, in combination with the following setups of discriminators - a) SS: CNN- and RNN-based Single Sentence classifier,[3] and b) SP: CNN- and RNN-based Sentence-Pair classifier (Section 2.2). We also experiment with using the GLEU score directly as the reward for an input-output sentence pair. This setting overlaps with the work of Sakaguchi et al. (2017).[4]

3.3 Implementation Details

3.3.1 Data

Following Junczys-Dowmunt et al. (2018), we use byte-pair encoding (BPE) sub-word units (Sennrich et al., 2016), which is also the way to address the issue of out-of-vocabulary words. The vocabulary is based on 35k most frequent BPE subword units, where both the source and target side use the same vocabulary.

3.3.2 Generators

We refer to Junczys-Dowmunt et al. (2018) for our training setup, who laid out extensive guidelines for adapting NMT-based models for the GEC task. For the RNN-based generator, following Luong et al. (2015), we use 4 layers of bi-directional GRUs in both the encoder and decoder. We set the word embedding size to 512, size of hidden units for both encoder and decoder as 1024. For the Transformer, following the BASE model in Vaswani et al. (2017), we set up the model architecture with the encoder and decoder both having a stack of six layers of self-attention/feed-forward sub-layers. The word embedding size is set to 512, and the number of attention heads to 8. The size of the inner layer in the position-wise feed-forward network is set to 2048. In order to discourage copying (Gal and Ghahramani, 2016; Junczys-Dowmunt et al., 2018; Grundkiewicz et al., 2019) we use strong dropout for regularization: layer dropout of 0.3 for both the RNN and Transformer models, attention dropout of 0.1, and source and target word dropout of 0.2 and 0.1 respectively. These hyperparameters were chosen as prescribed in the referred works, but also worked well in practice when tuned on the development sets.

Algorithm 1 Adversarial-GEC

```
Initialize $G_\theta$, $D_\phi$ with random weights $\theta$, $\phi$.
Pre-train $G_\theta$ on ground-truth dataset $\mathcal{D} = (X, Y)$ with MLE loss
Generate negative samples $\mathcal{D}' = (X, Y')$ using $G_\theta$ for training $D_\phi$
Pre-train $D_\phi$ on $\mathcal{D}$ and $\mathcal{D}'$ until initial accuracy $\varepsilon$ with BCE loss
while not converged do
    Sample $(X, Y) \sim P_{data}$
    Sample $Y' \sim G_\theta(\cdot|X)$
    Sample $\rho \sim [0, 1]$ to determine interleaving
    if $\rho \leq \lambda$ then
        Compute Rewards $R$ for $(X, Y')$ using $D_\phi$
        Update $G_\theta$ via Policy Gradient using $R$
    else
        Update $G_\theta$ via teacher-forcing using MLE
    Train $D_\phi$ using Eqn. 2, on $(X, Y)$ and $(X, Y')$
*Parameter update equations for $G_\theta$ and $D_\phi$ are as follows:
$\theta \leftarrow \theta - \alpha_g \nabla_{\theta_G}$
$\phi \leftarrow \phi - \alpha_d \nabla_{\phi_D}$
```

3.3.3 Sentence-Pair Discriminators

The RNN-based discriminator model is set up as a siamese network, sharing the same embeddings and weights, each processing one of the two sentences. The RNN-based model, for each sentence in the pair, consists of a word embedding layer of size 300, followed by two layers of bi-directional GRU, with hidden size of 128. There are residual connections at each time step between the layers. The bi-directional outputs of the last recurrent layer of both the sentences in the pair are concatenated, and used as input to a dense feed-forward layer with an output of size 128, followed by a sigmoid. We use dropout on the recurrent units and between layers (both with probability 0.2). For the CNN-based discriminator, we use the convolutional matching model used by Wu et al. (2018) since Hu et al. (2014) found it to have a superior performance to the siamese architecture.

3.3.4 Training

A major challenge with GANs is that the joint training between the generator and the discriminator needs to be carefully coordinated, in order to stabilize the training (Yu et al., 2017; Li et al., 2017; Yang et al., 2018; Wu et al., 2018; Fedus et al., 2018; Wang and Lee, 2018). Therefore, we first pre-train the generator model G_θ using maximum likelihood estimation (MLE) on the ground-truth training dataset until convergence. This stage is

[3] A failed formulation using language models is described in Section 5.4.

[4] We are unable to provide comparison against Sakaguchi et al. (2017) because they report results on JFLEG Corpus. See Section 3.1 for details.

	FCE			CoNLL14			BEA19		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Baselines									
RNN	58.50	20.85	42.97	60.37	18.74	41.80	49.21	34.44	45.32
Transformer	60.87	25.03	47.30	63.98	21.52	45.88	50.38	35.43	46.45
Adversarial-GEC (Our System)									
RNN + CNN	**64.21**	22.46	46.81	59.31	21.01	43.46	**54.21**	34.37	48.6
Transformer + CNN	62.53	**27.82**	**50.04**	**64.68**	**22.57**	**47.10**	53.78	**36.52**	**49.13**
Recent GEC Systems									
Ji et al. (2017)[†]	-	-	-	-	-	41.53	-	-	-
Grundkiewicz and Junczys-Dowmunt (2018)[††]	-	-	-	**66.61**	17.58	42.76	-	-	-
Chollampatt and Ng (2018a)[‡,†]	-	-	-	59.68	23.15	45.36	-	-	-
Zhao et al. (2019)[¶]	-	-	-	55.96	30.73	48.07	-	-	-
Kaneko et al. (2020)	**61.7**	**46.4**	**57.9**	59.2	**31.2**	**50.2**	**51.5**	**43.2**	**49.6**

Table 2: Results of Adversarial-GEC against single-model NMT baselines of state-of-the-art GEC systems. [†]Trained on non-public CLC data, [††]Trained on NUCLE and Lang-8, [‡]MLConv - single model, [¶]Trained on One-Billion Word Benchmark

essential to enable the joint training to converge later, since the action space during generation is immense and applying Policy Gradient training from scratch would lead to slow and unstable training. The pre-trained model is then used to decode the training data x using beam search (size 4), and generate the output sentences y', essentially building the negative examples in the training data for the discriminator (x, y'). The discriminator is initially pre-trained on a combination of the ground-truth parallel data (x, y) and the machine-generated data (x, y'), where y' is sampled from the pre-trained generator model. The discriminator is trained until the classification accuracy reaches ε (further analysis in Section 5.2). Once the generator and the discriminator have been pre-trained, they are adversarially co-trained, where the generator is trained with a combination of MLE and Policy Gradient (and teacher forcing), until the performance of G_θ does not improve on the development set.[5]

4 Results

In contrast to related works on Neural GEC, we do not use a lot of the heuristics that most recent systems leverage in order to enhance their model performance pre- and post-training. These heuristics include using spellcheckers to correct spelling errors in the data, pre-trained language models trained on large quantities of external data, synthetic data generation, re-ranking systems to sort the outputs of the generator model, among others. We chose to keep our framework simple compared to most contemporary works in that we do not leverage anything beyond what the raw training data and the baseline architectures have to offer, which makes it simple and self-contained. This decision was in the interest of system complexity, training time, and clear evaluations. The goal of this work is not to build a state-of-the-art GEC system but to demonstrate the value of adversarial training. Hence, we report results in a single-model setting, without the use of any external data or resources beyond the training data.

The results of Adversarial-GEC compared to baseline models are presented in Table 2.[6] These results are based on the best performing (on the development set) parameters $\varepsilon = 0.7$, $\lambda = 0.4$ using the CNN sentence-pair discriminator. The results demonstrate a substantial improvement in $F_{0.5}$ for both adversarially trained models, across all evaluation datasets. Overall, the RNN model achieves greater gains on precision than the Transformer, which achieves greater gains on recall. We carry out statistical significance tests with bootstrap resampling, and correcting for multiple comparisons, obtain significant gains over the baselines ($p < 0.01$).

As mentioned in Sections 2.2 and 3.2, we experiment with three discriminator formulations (SS, SP, GLEU) in the Adversarial-GEC setting to provide the rewards to guide the generators. Table 3 describes the results of using the two kinds of discriminators in each formulation (CNN, RNN) of

[5] More details in Appendix A.

[6] While multiple recent works based on pre-trained Transformers such as Kiyono et al. (2019) and Omelianchuk et al. (2020) have pushed the state-of-the-art in GEC, they are not comparable to our work because of the use of pretrained LMs, ensembles and synthetic training data.

	Generator	FCE	CoNLL14	BEA19
SS: Single-Sentence Discriminator				
CNN	RNN	41.68	40.23	45.53
	Transformer	43.45	41.52	46.31
RNN	RNN	41.21	39.25	45.58
	Transformer	41.36	39.84	46.86
SP: Sentence-Pair Discriminator				
CNN	RNN	46.81	43.46	48.6
	Transformer	**50.04**	**47.10**	**49.13**
RNN	RNN	46.45	43.17	48.11
	Transformer	49.88	46.95	49.02
GLEU	RNN	43.35	42.1	46.68
	Transformer	45.65	45.9	47.84

Table 3: Impact of training different Discriminator task formulations and models on $F_{0.5}$ test splits.

the discriminative task, and doesn't show a significant difference in either formulation.

5 Discussion

In this section, we describe experimental results on adversarial training strategies, based on validation data splits. There are three parts to making the training work (a) formulating the discriminator task to compute the reward, (b) reducing the variance in rewards for better gradient estimation, and (c) combining the MLE and Adversarial objectives for more stable training.

5.1 Discriminator Formulation

We observe in Table 3 that the single-sentence discriminator (SS) performs the worst against all discriminator formulations. Furthermore, SS performs even worse than the baseline generators, which points to the direction that it acts as a barrier in their ability to generalize.

We attribute this performance limitation to two factors. First, since the model does not consider the original sentence, it lacks the ability to learn the parts of the sentence which make it ungrammatical, rewarding similarly marginally correct and highly incorrect sentences. We investigate this idea by feeding the discriminator incorrect sentences sampled from P_{data} and observe that they get nearly the same reward from SS despite their varying degrees of incorrectness. This impedes generator improvement as any inaccuracies are penalized disproportionately. Secondly, producing grammatically correct sequences is not enough to solve the task. A generated sequence can be grammatically correct, albeit semantically or lexically different. A discriminator which lacks the contextual information provided by the original sentence can reward such sequences with a high reward propagating such false starts. Therefore, a generator that produces only *one* grammatical sentence would receive a high reward from the discriminator.

On the other hand, GLEU achieves better performance compared to SS but weaker when compared to SP. This corroborates the above argument as GLEU, essentially being a special case of the SP formulation, is able to provide higher quality reward since it tries to account for fluency *and* grammaticality in evaluation on references. SP, on the other hand, is able to go beyond the GLEU score's low-level n-gram matching criteria, learning latent characteristics of the GEC task and providing a more appropriate reward to the generator. Acting in this way provides a much smoother objective compared with GLEU since the latter is quite sensitive to slight translation differences at the word or phrase level. Second, the generator and discriminator co-evolve. The dynamics of the discriminator make the generator grow in an adaptive way rather than controlled by a fixed evaluation metric such as GLEU, achieving better distributional alignment, which is further verified by its superior performance.

5.2 Balancing Discriminator Pre-Training

Since GAN training is a *min-max* loss optimization with alternating updates to the generator and the discriminator, it is hard to reach a global optimum, which is a saddle point. To successfully reach the saddle point, balancing the generator and the discriminator co-training is essential. But the discriminator usually converges faster than the generator, so it is hard to achieve that balance. Failure to do so often leads to problems like mode collapse or inability to learn altogether. While the generator is pre-trained to reach the best development-set performance, we control the discriminator pre-training to balance the adversarial training. Hence, we evaluate the impact of the pre-trained discriminator's accuracy ε as a tunable hyperparameter. We pre-train seven RNN discriminators to reach accuracy in the range $[0.6, 0.9]$. With these discriminators, we train corresponding Adversarial-GEC models (using a Transformer generator, $\lambda = 0.4$) and evaluate their performance on the development set at regular intervals. Fig. 2 shows that the initial accuracy of the discriminator significantly impacts the final performance and needs to be set carefully. If it is either too high (0.85 and 0.9) or too low (0.6 and

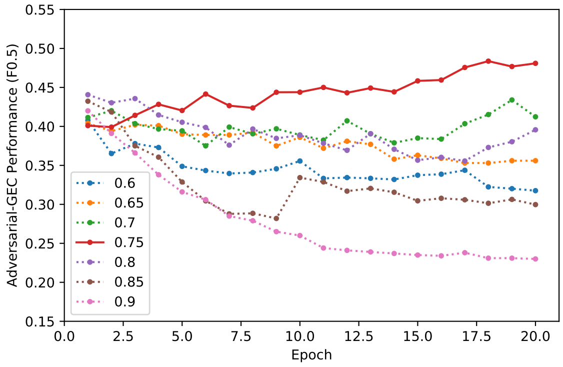

Figure 2: $F_{0.5}$ scores on the dev set using pre-trained Transformer, and CNN discriminators with varying initial accuracy ε.

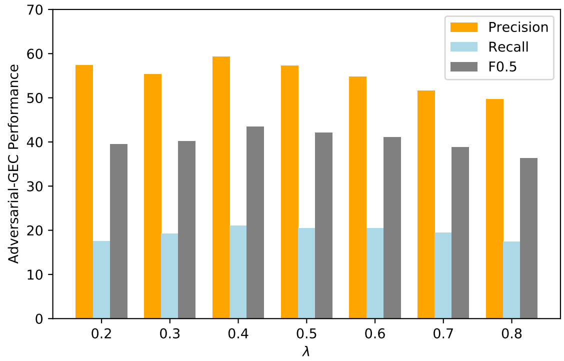

Figure 3: Adversarial-GEC performance on the dev set (Transformer + CNN), varying parameter λ to alternate between MLE and Policy Gradient training.

0.65), the model performs poorly. This points to the need for a balanced relationship between the generator and the discriminator. If the discriminator is too strong, the generator is extremely penalized for its erroneous predictions, and the performance progressively gets worse. On the other hand, if the discriminator is too weak, it is unable to give the most appropriate guidance to the generator. Empirically, we pre-train the discriminator until its accuracy reaches the 0.7-0.75 range.

5.3 Combining MLE and Adversarial Objectives

As noted in Section 2.4, a key factor in successfully training G_θ is the combination of adversarial and MLE objectives where we define the hyperparameter λ to control the trade-off between MLE and adversarial training. That is, for any mini-batch, determined by a probability λ, G_θ is optimized by the MLE objective or adversarial objective to improve the stability in model training. We experiment with the range $[0.2, 0.8]$ for λ. The results in Fig. 3 show that combining the MLE objective with the adversarial objective is helpful to stabilize the training and improve the model performance, as we expected. This confirms prior findings that MLE acts as a regularizer to guarantee smooth model updates, alleviating the negative effects brought by high gradient estimation variance of the one-step Monte-Carlo sample in REINFORCE. However, further increasing λ does not bring more gain. The best trade-off between MLE and adversarial objective in our experiment is $\lambda = 0.4$, which is the value we use in our experiments.

5.4 Experiments with Language Models

In the SS setting, we also experimented with a locally-normalized language model as a discriminator. The intuition here was that using a language model with token-level locally normalized probabilities could offer a more direct training signal to the generator. If a generated sentence does not match the distribution of ground-truth data, it will have high perplexity when evaluated by a language model that was trained on ground-truth data. Not only can it provide an overall evaluation score for the whole sentence, but can also assign a probability to each token, thus providing more information on which word is to blame if the overall perplexity is very high. However, in spite of all the training strategies described in Section 2.4, training a language model was highly unstable, due to the use of a single sample to approximate the expected gradient, leading to high variance in gradient estimates. In future works, we aim to explore this idea using better generator models and better, larger-scale language models such as BERT (Devlin et al., 2018) and GPT-3 (Brown et al., 2020).

6 Related Work

While the choice of a sentence-pair discriminator is close to Yang et al. (2018) and Wu et al. (2018), our work differs from Yang et al. (2018) in that their learning objective is a combination of the discriminator reward (D) and a smoothed sentence-level BLEU (Papineni et al., 2002) as the static reward (Q). The use of a sentence-pair discriminator is related to our work, we do not combine rewards from D and Q. Incorporating Q in the objective stems from the motivation to directly optimize for the evaluation metric, we choose to not force the evaluation metric-based reward into the objective, since most GEC metrics are reference-based, and have shown to be limiting for the task (Choshen and Abend, 2018; Chollampatt and Ng, 2018c). Similarly, among existing works for GEC, our work is the closest to Sakaguchi et al. (2017), but they also directly maximize GLEU in training their GEC system, using a REINFORCE-based approach similar to ours. We instead let the model learn the latent nuances of the objective directly from the data, and provide the appropriate reward to the generator, preserving the learning objective as in Yu et al. (2017), albeit with a different discriminator framework. Our work is closest to Wu et al. (2018), who built an RNNSearch-based Generator (Bahdanau et al., 2015) and a CNN-based sentence-pair discriminator for NMT.

7 Conclusion

We propose a task-appropriate training objective for GEC, using an adversarial training framework consisting of a generator and a discriminator, based on the Adversarial-NMT framework of Wu et al. (2018). The generator is modeled as a Seq2Seq model, and the discriminator is modeled as a deep sentence-pair matching model, which provides rewards to the generator input-output. The framework supervises the generator to reflect the mapping within (source, target) sentence, and an efficient policy gradient algorithm to tackle the optimization difficulty brought by the discrete nature of generation. Experiments on standard GEC test datasets demonstrate the effectiveness of our framework for the task. Additionally, we provide insights into how the discriminator setup, pre-training and integration into the framework can be optimized for stable training and better performance. We show that the proposed framework consistently achieves better results in a self-contained single model setting, without relying on any external resources. In the future, we plan to improve the task-specific framework and training techniques based on recent state-of-the-art methods (Grundkiewicz et al., 2019; Choe et al., 2019), and improve issues with sparse rewards by exploring better credit assignment techniques.

Acknowledgments

We would like to thank our friends and colleagues: Vivek Kulkarni, Artem Chernodub, Kostiantyn Omelianchuk, Oleksandr Skurzhanskyi, Oleksiy Syvokon, and Chad Mills, for their insightful feedback, and the anonymous reviewers for their helpful comments.

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting esl errors using phrasal smt techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Sydney, Australia. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

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.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Liwei Cai and William Yang Wang. 2018. KBGAN: Adversarial learning for knowledge graph embeddings. In *Proceedings of the 2018 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1470–1480, New Orleans, Louisiana. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. 2019. A neural grammatical error correction system built on better pre-training and sequential transfer learning. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, Florence, Italy. Association for Computational Linguistics.

Shamil Chollampatt and Hwee Tou Ng. 2018a. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Shamil Chollampatt and Hwee Tou Ng. 2018b. Neural quality estimation of grammatical error correction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2528–2539, Brussels, Belgium. Association for Computational Linguistics.

Shamil Chollampatt and Hwee Tou Ng. 2018c. A reassessment of reference-based grammatical error correction metrics. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2730–2741, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shamil Chollampatt, Kaveh Taghipour, and Hwee Tou Ng. 2016. Neural network translation models for grammatical error correction. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, New York, USA.

Leshem Choshen and Omri Abend. 2018. Inherent biases in reference-based evaluation for grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, Melbourne, Australia. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2011. Grammatical error correction with alternating structure optimization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 915–923, Portland, Oregon, USA. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. 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*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

Rachele De Felice and Stephen G. Pulman. 2008. A classifier-based approach to preposition and determiner error correction in l2 english. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 169–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

William Fedus, Ian Goodfellow, and Andrew M. Dai. 2018. MaskGAN: Better text generation via filling in the ______. In *International Conference on Learning Representations*.

Mariano Felice and Ted Briscoe. 2015. Towards a standard evaluation method for grammatical error detection and correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 578–587, Denver, Colorado. Association for Computational Linguistics.

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24, Baltimore, Maryland. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 1027–1035, Red Hook, NY, USA. Curran Associates Inc.

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 (Volume 1: Long Papers)*, pages 1055–1065, Melbourne, Australia. Association for Computational Linguistics.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680, Cambridge, MA, USA. MIT Press.

Sylviane Granger. 1998. The computerized learner corpus: a versatile new source of data for sla research.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near human-level performance in grammatical error correction with hybrid machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 284–290, New Orleans, Louisiana. Association for Computational Linguistics.

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*, pages 252–263, Florence, Italy. Association for Computational Linguistics.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *AAAI Conference on Artificial Intelligence*.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2042–2050, Cambridge, MA, USA. MIT Press.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A nested attention neural hybrid model for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–762, Vancouver, Canada. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556, Austin, Texas. 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, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. 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.

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)*, pages 1236–1242, Hong Kong, China. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169, Copenhagen, Denmark. Association for Computational Linguistics.

Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. 2018. Generative adversarial network for abstractive text summarization. In *AAAI Conference on Artificial Intelligence*.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. 2017. Improved image captioning via policy gradient optimization of spider. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 873–881.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 588–593.

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*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The conll-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA â†’ Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alla Rozovskaya and Dan Roth. 2014. Building a state-of-the-art grammatical error correction system. *Transactions of the Association for Computational Linguistics*, 2:419–434.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2017. Grammatical error correction with neural reinforcement learning. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 366–372, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chinnadhurai Sankar and Sujith Ravi. 2019. Deep reinforcement learning for modeling chit-chat dialog with discrete attributes. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 1–10, Stockholm, Sweden. Association for Computational Linguistics.

Allen Schmaltz, Yoon Kim, Alexander Rush, and Stuart Shieber. 2017. Adapting sequence models for sentence correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2807–2813, Copenhagen, Denmark. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Pei-Hao Su, David Vandyke, Milica Gašić, Nikola Mrkšić, Tsung-Hsien Wen, and Steve Young. 2015. Reward shaping with recurrent neural networks for speeding up on-line policy learning in spoken dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 417–421, Prague, Czech Republic. Association for Computational Linguistics.

Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng. 2014. System combination for grammatical error correction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 951–962, Doha, Qatar. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for ESL learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.

Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using parse features for preposition selection and error detection. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 353–358, Uppsala, Sweden. Association for Computational Linguistics.

Joel R. Tetreault and Martin Chodorow. 2008. The ups and downs of preposition error detection in esl writing. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 865–872, Stroudsburg, PA, USA. Association for Computational Linguistics.

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 *NIPS*, pages 6000–6010.

Yaushian Wang and Hung-yi Lee. 2018. Learning to encode text as human-readable summaries using generative adversarial networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4187–4195, Brussels, Belgium. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.

Lijun Wu, Yingce Xia, Fei Tian, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. Adversarial neural machine translation. In *Proceedings of The 10th Asian Conference on Machine Learning*, volume 95 of *Proceedings of Machine Learning Research*, pages 534–549. PMLR.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Daniel Jurafsky, and Andrew Y. Ng. 2016. Neural language correction with character-based attention. *CoRR*, abs/1603.09727.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Improving neural machine translation with conditional sequence generative adversarial nets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1346–1355, New Orleans, Louisiana. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *ACL*, pages 180–189. The Association for Computer Linguistics.

Helen Yannakoudakis, Marek Rei, Øistein E. Andersen, and Zheng Yuan. 2017. Neural sequence-labelling models for grammatical error correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2795–2806, Copenhagen, Denmark. Association for Computational Linguistics.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 2852–2858. AAAI Press.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.

Zheng Yuan, Ted Briscoe, and Mariano Felice. 2016. Candidate re-ranking for SMT-based grammatical error correction. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 256–266, San Diego, CA. Association for Computational Linguistics.

Zheng Yuan and Mariano Felice. 2013. Constrained grammatical error correction using statistical machine translation. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 52–61, Sofia, Bulgaria. 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. *CoRR*, abs/1903.00138.