Making Use of Latent Space in Language GANs for Generating Diverse Text without Pre-training

Takeshi Kojima, Yusuke Iwasawa, Yutaka Matsuo
Department of Technology Management for Innovation
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
t.kojima,iwasawa,matsuo@weblab.t.u-tokyo.ac.jp

Abstract

Generating diverse texts is an important factor for unsupervised text generation. One approach is to produce the diversity of texts conditioned by the sampled latent code. Although several generative adversarial networks (GANs) have been proposed thus far, these models still suffer from mode-collapsing if the models are not pre-trained. In this paper, we propose a GAN model that aims to improve the approach to generating diverse texts conditioned by the latent space. The generator of our model uses Gumbel-Softmax distribution for the word sampling process. To ensure that the text is generated conditioned upon the sampled latent code, reconstruction loss is introduced in our objective function. The discriminator of our model iteratively inspects incomplete partial texts and learns to distinguish whether they are real or fake by using the standard GAN objective function. Experimental results using the COCO Image Captions dataset show that, although our model is not pre-trained, the performance of our model is quite competitive with the existing baseline models, which requires pre-training.

1 Introduction

Generative adversarial networks (GANs) (Goodfellow et al., 2014) have recently received significant attention in the field of unsupervised text generation, which aims to generate realistic texts by unsupervised learning approach.

For language GANs, the diversity of the generated texts is an important evaluation metric. There are mainly two approaches to produce the diversity of texts by the generative models. One approach, which includes SeqGAN (Yu et al., 2017) and LeakGAN (Guo et al., 2018), is to generate the diverse texts by sampling words during the text-generation process. The generators of these models set the initial state as zero and randomly sample every word depending on the word distribution, and thus we cannot control the generated text depending on any conditions. The other approach, which includes TextGAN (Zhang et al., 2017) and FM-GAN (Chen et al., 2018), produces the diversity of texts depending on the randomly sampled latent code from the prior distribution. These models set the latent code information at the initial state or the input of every time step for the generator.

In this paper, we propose a GAN model that aims to improve the approach to generating diverse texts from the latent space. As for TextGAN and FM-GAN, the generator almost decisively selects each word using soft-argmax approximation to generate a sentence depending on the latent space information. To avoid mode-collapsing, instead of using standard GAN objective function, the discriminator of each model respectively measures the Maximum Mean Discrepancy (MMD) or the Feature Mover's Distance (FMD) between the true text representations and fake ones. These models succeed in generating diverse texts if the generators are pre-trained by a Variational Autoencoder. However, it is verified that these methods still fall into mode-collapsing if the generator is not pre-trained (Section 4.3.1). One of the possible reasons for the mode-collapsing is the deterministic word sampling process through a soft-argmax approximation from the beginning of the training. Deterministic word sampling process hinders the generator from exploring a variety of text generation, which may lead the generator to fall into sub-optimal point. The second possible reason is that the discriminator tries to discriminate the completed sentences. Generating a good-completed sentence from the beginning of the training is too difficult for the generator because the possible number of combinations of words increases exponentially if the number of words sampled becomes large. Therefore, there is a possibility that, without pre-training, the

Proceedings of the 16th Conference of the European Chapter of the Associationfor Computational Linguistics: Student Research Workshop, pages 175–182
April 19 - 23, 2021.

discriminator does not serve useful signals to the generator if the discriminator looks at only completed sentences. Based on these assumptions, our model adopts the following approach: The generator randomly samples words depending on the word probability distribution using the Gumbel-Softmax distribution (Jang et al., 2016). To ensure that the texts generated are conditioned upon the latent code, a reconstructor is introduced, which is fed the generated sentence and outputs the reconstructed latent code. The generator and the reconstructor cooperatively minimize the reconstruction loss. The discriminator of our model iteratively inspects incomplete partial texts and learns to distinguish whether they are real or fake using the standard GAN objective.

We trained our model using the COCO Image Captions dataset (Lin et al., 2014) for the experiment. The results show that although our model is not pre-trained, its performance is quite competitive with the existing baseline models. We also found that, by controlling the weight of the reconstruction loss coefficient, our model can obtain a higher diversity of generated texts even when texts are generated by greedy decoding.

2 Related Work

The language GANs, in which the generator and discriminator optimize their objective functions in an adversarial manner to generate realistic texts, have two main perspectives.

As the first perspective, a reinforcement learning approach is used for optimizing the generator. SeqGAN (Yu et al., 2017), LeakGAN (Guo et al., 2018), MaskGAN (Fedus et al., 2018), RankGAN (Lin et al., 2017), RL-GAN (Caccia et al., 2019), and ScratchGAN (de Masson d'Autume et al., 2019) are typical models. These models are non-differentiable from the discriminator to the generator. Therefore, the generator cannot be optimized using a standard GAN approach. Instead, the output of the discriminator is regarded as a reward for the sampling of words, and the expected rewards are maximized to optimize the generator. This reinforcement approach generally produces the diversity of texts during the word sampling process.

As the second perspective, the model is end-to-end differentiable from the discriminator to the generator. TextGAN (Zhang et al., 2017), RelGAN (Nie et al., 2018), and FM-GAN (Chen et al., 2018) are typical models. The sampling of words is approximated using a soft-argmax approximation or the Gumbel-Softmax distribution, which is used to create approximated one-hot vectors by lowering the temperature of the softmax function. Some models of this approach produce the diversity of texts by sampling latent code from the prior distribution. Note that GSGAN (Kusner and Hernández-Lobato, 2016) is also an end-to-end differentiable model for discretized data, but does not verify the effectiveness in the case of text generation.

Model	Generate text Conditioned by Latent space	Require Pretraining
SeqGAN, LeakGAN, etc	No	Yes (MLE)
ScratchGAN, COT, MLE	No	No
TextGAN, FM-GAN	Yes	Yes (VAE)
Ours[GAN], VAE-Based	Yes	No

Table 1: Summary of previous studies, where (·) indicates a pretraining approach. MLE, Maximum Likelihood Estimator; VAE, Variational Autoencoder.

Other text generation approaches beyond those described above also exist, such as a VAE-based model (Bowman et al., 2016; Bao et al., 2019) and COT(Lu et al., 2019) among others (Gagnon-Marchand et al. (2019), Li et al. (2019)).

To the best of our knowledge, our approach is the first GAN model that does not require variational Autoencoder pre-training and is able to generate texts conditioned by the latent code[1]. Previous studies are summarized in Table 1.

3 Model

Figure 1 shows a schematic illustration of the proposed method. We describe the details in the following paragraphs.

Our goal is to generate sentences conditioned by the latent space in a GAN framework. When training language GANs, if the discriminator only looks at the complete sentences, the generator obtains no learning signals early in the training be-

[1]For FM-GAN, no description regarding the necessity of pre-training is provided in this paper. However, their released code refers to the pre-training procedure and is available at https://github.com/vijini/FM-GAN. We also verified that a model without VAE pre-training cannot achieve the expected performance (Figure 3).

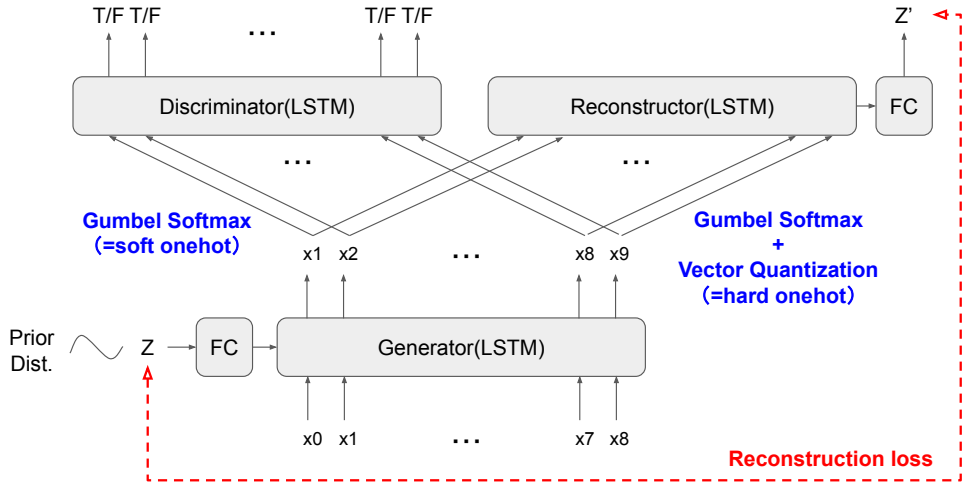

Figure 1: Overview of our model. The latent code z sampled from the prior distribution is fed into the fully connected layer and set as the initial state of the generator. The generator iteratively generates soft one-hot vectors of words using the Gumbel-Softmax distribution. The discriminator is fed the soft one-hot vectors and outputs the dense reward iteratively for the GAN loss. The reconstructor is fed the vector-quantized hard one-hot vectors and outputs the reconstructed latent code for the reconstruction loss. This network is end-to-end differentiable from the discriminator/reconstructor to the generator.

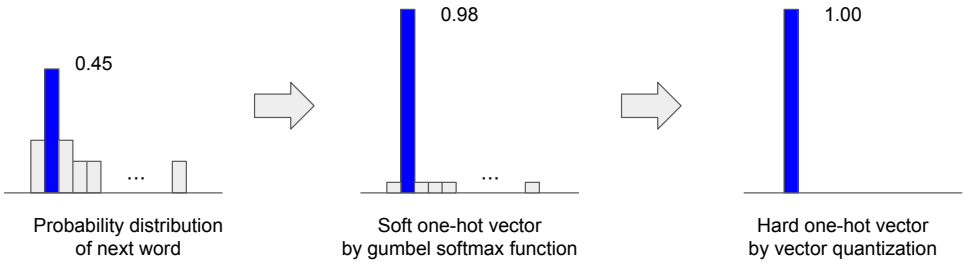

Figure 2: Example of vector quantization of words from soft one-hot vector to hard one-hot vector. This approach can back-propagate the loss from the reconstructor to the generator using only the sampled word parameters. Implementation details: hard one-hot = onehot(Argmax(soft one-hot)) - StopGradient[soft one-hot] + soft one-hot

cause the complete sentences generated are easily determined to be fake by the discriminator(de Masson d'Autume et al., 2019). To address this problem, following the idea of Fedus et al. (2018), Semeniuta et al. (2018), and de Masson d'Autume et al. (2019), our discriminator D_ϕ iteratively inspects incomplete partial texts and learns to distinguish whether they are real or fake. Therefore, the generator can obtain more informative signals from the recurrent discriminator during the iterative word sampling process. By using this recurrent discriminator, it is expected that our model does not require pre-training, as reported in ScratchGAN (de Masson d'Autume et al., 2019). The objective function of the discriminator D_ϕ is the same as that in ScratchGAN, except that the generator p_θ generates a sequence of tokens $\{x_1, ..., x_T\}$ depending on the sampled latent code z from the prior distribution $p(z)$.

$$\max_{\phi} \sum_{t=1}^{T} \mathbb{E}_{p^*(x_t|x_{<t})}[logD_\phi(x_t|x_{<t})] \quad (1)$$
$$+ \sum_{t=1}^{T} \mathbb{E}_{p(z)p_\theta(x_t|z,x_{<t})}[log(1 - D_\phi(x_t|x_{<t}))],$$

where $x_{<t} := \{x_1, .., x_{t-1}\}$ denotes a sequence of words before timestep t, and p^* is the real data distribution.

In a practical sense, the typical word sampling process makes the differentiability from the discriminator to the generator impossible because sampling a word from a word probability distribution is equivalent to creating a non-differentiable one-hot vector. As a workaround, we use the Gumbel-Softmax distribution (Jang et al., 2016), which enables our model to sample words from p_θ by creating an approximated one-hot vector while making

the differentiability from the discriminator to the generator possible. Here, we call this one-hot vector a "soft one-hot vector." The Gumbel-Softmax distribution used to create the soft one-hot vector $\tilde{x}$ is set as follows:

$$\tilde{x} = softmax((log(p_\theta) + g)/\tau_1) \quad \text{where } p_\theta = softmax(o/\tau_2) \tag{2}$$

Here, g is a randomly sampled value from the Gumbel distribution $Gumbel(0, 1)$, o is the output from the generator. Note that τ_1 is the Gumbel-Softmax temperature, and τ_2 is the word probability distribution temperature.

To ensure that the texts generated are conditioned by the sampled latent code, our model introduces a reconstructor R_ψ, which is fed the generated text and outputs a reconstructed latent code to minimize the reconstruction loss between the latent code and the reconstructed code. The generator p_θ and reconstructor R_ψ are optimized simultaneously. Therefore, the objective function of the reconstructor is added to that of the generator multiplied by a coefficient λ.

$$\min_{\theta,\psi} \sum_{t=1}^{T} \mathbb{E}_{p(z)p_\theta(x_t|z,x_{<t})}[log(1 - D_\phi(x_t|x_{<t}))] + \lambda \mathbb{E}_{p(z)p_\theta(x_1..,x_t)} \left[\frac{1}{N_z} ||R_\psi(x_1, .., x_t) - z||_2^2 \right] \tag{3}$$

where N_z denotes the dimension size of the latent code z. It should be noted that the joint distribution $p_\theta(x_1...x_t)$ is decomposed into the iterative conditional distribution $p_\theta(x_1...x_t) = p_\theta(x_1)p_\theta(x_2|x_1)...p_\theta(x_t|x_{<t})$ such that conditional sampling can be executed using the Gumbel-Softmax distribution described above.

We found several heuristic approaches for stabilizing the training. As the first, vector quantization (Van Den Oord et al., 2017) is applied to the soft one-hot vector to create a "hard one-hot vector" for the reconstructor input. Vector quantization can back-propagate the loss from the reconstructor to the generator using only the sampled word parameters. By using this approach, reconstruction loss training is stabilized. Figure 2 illustrates an example of the vector quantization process. As the second technique, if a blank token is chosen at any time step by the Gumebel softmax of the generator, the rest of the sentence is automatically padded with blank tokens. The pseudocode is given in Appendix C.

4 Experiment

First, we describe the data setting and evaluation metrics for the experiment. Second, we describe the experimental results to better evaluate the performance of our model.

4.1 Data Setting

We experimentally evaluated the quality and diversity of our generated models using a real sentence dataset, i.e., COCO Image Captions (Lin et al., 2014). We used the same sampled data as in Zhu et al. (2018), which consist of 10,000 training texts and 10,000 evaluation texts[2]. The maximum sentence length was 37 tokens, the average length of the sentence was 11.3 tokens, and the vocabulary size was 6577. For the experiment, the end of the text was padded with blank tokens.

4.2 Evaluation Metrics

The quality and diversity of the generated text were measured using the Negative BLEU score and the Self-BLEU score (Zhu et al., 2018), respectively. In tuition, the BLEU score measures the quality of the generated sentences through a comparison with real sentences from the viewpoint of how much the N-gram words overlap. The negative BLEU score is defined as the -1 $*$ BLEU score. The Self-BLEU scores measure the diversity of every generated sentence by comparing with the other generated sentences by inspecting how much the N-gram words overlap. Therefore, a lower value indicates a better performance for both metrics. We draw the temperature curve(Caccia et al., 2019) for each model, in which the texts are generated by gradually changing the temperature of the softmax function, and plotting the quality and diversity score for every temperature point on a two-dimensional quality-diversity canvas. Therefore, the closer the curve is to the origin, the better the performance of the model. We plot the results for temperatures at intervals of 0.0 to 1.0 with 0.1 increments. Note that at a temperature of 0.0, the model generates the text using a greedy approach, which can be interpreted as temperature $\tau \to 0$. In principle, as the temperature decreases, the quality of the generated texts increases, but the diversity decreases. Thus, the greedy decoding case is the upper leftmost point on each temperature curve. We generate 10,000 texts from the trained model at every temperature

[2]The dataset is available at https://github.com/geek-ai/Texygen/tree/master/data

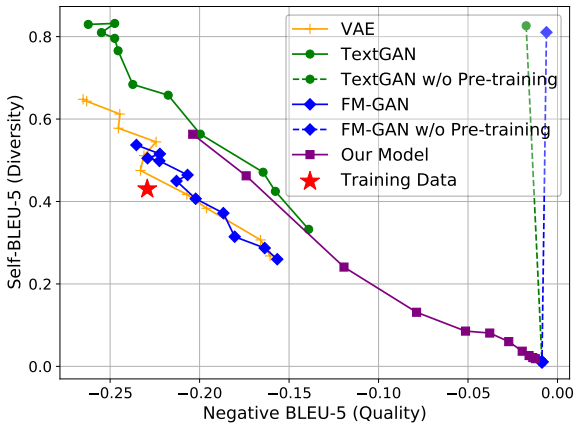

Figure 3: Performance comparison of our model with baseline models. The lower value indicates the better performance for diversity and quality metrics.

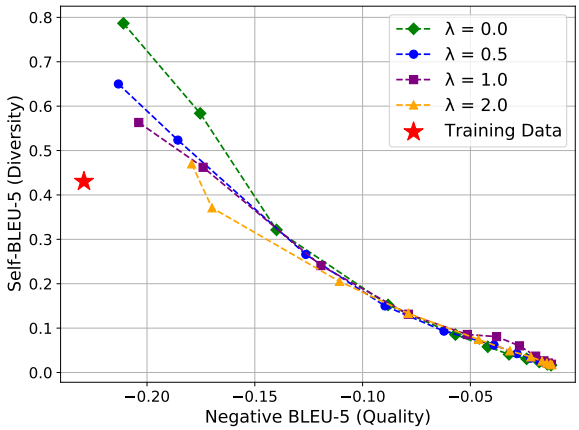

Figure 4: Effect of changing reconstruction loss coefficient λ.

for evaluation. The generated sample texts can be found at Table 2. Note that in the Gumbel-Softmax distribution, the changing target of the temperature is τ_2, and not τ_1 in equation (2).

4.3 Results

4.3.1 Comparison with Baseline Models

We compared the performance of our model with the baseline models: VAE, FM-GAN, and TextGAN. VAE used the CNN-LSTM autoencoder architecture as in Gan et al. (2017). FM-GAN and TextGAN require VAE pre-training. Our model settings and hyperparameter details can be found in Appendix A and B. Figure 3 illustrates the temperature curves for each model to compare the performance from the viewpoint of quality and diversity. This result indicates that the overall performance of our model is slightly inferior to that of the baseline models, but is quite competitive despite our model not being pre-trained. We also evaluated FM-GAN and TextGAN without pre-training case. Both of them achieved far worse performance than the pre-trained case. This result indicates that these models without pre-training fall into mode-collapsing.

4.3.2 Effect of Changing Value of λ

We observe the effect of changing the value of the reconstruction coefficient λ in equation (3) on the performance of our model. Figure 4 shows the temperature curves for different coefficients. The result indicates that, as the value of λ increases, the performance of our models improves. However, if the value of λ is too high such as $\lambda = 2.0$, the quality of the generated sentence significantly worsens. We also found that for the greedy decoding case, which is the upper-left point of each curve, as λ increases, the diversity of the generated sentences increases and the quality-diversity distribution becomes closer to that of the training data. Greedy decoding is the most extreme case for verifying if the generated sentences are conditioned by the latent space. Therefore, it can be assumed that the reconstruction loss has the ability to make the generated text more dependent on the latent space information.

5 Conclusion and Discussion

This paper proposed a GAN model that aims to improve the approach to generating diverse texts conditioned by a latent space. In a quantitative experiment using the COCO Image Captions dataset, it was shown that although our model is not pre-trained, its performance is quite competitive with the existing baseline models, which require pre-training. Future work will include further improvements to the performance of our model, and application of our model to other tasks that need to transform the data between domains through a latent space, such as improving the quality and diversity of machine translation or multi-modal learning related to text generation.

References

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio.

Real Data	a bicycle replica with a clock as the front wheel. a black honda motorcycle parked in front of a garage. a room with blue walls and a white sink and door. a car that seems to be parked illegally behind a legally parked car a large passenger airplane flying through the air. there is a gol plane taking off in a partly cloudy sky. blue and white color scheme in a small bathroom. this is a blue and white bathroom with a wall sink and a lifesaver on the wall. a blue boat themed bathroom with a life preserver on the wall
VAE	a bathroom sink with only the tub in the bathroom a large boy and a plane sitting on the landing . a clock tower with pots and windows a car at an open door leading to a bunch of foot . office space force force jet on display during day . an image of benches on a street and chairs being terminal . a large airplane flying in the open of a kitchen . a couple of an airplane flying through the clear blue oven . an airplane with some chairs on a table by the counter .
FM-GAN	a man wearing two sheep on a blue umbrella a group of birds standing around a table in a forest . a bathroom with a vanity , sink , and white and shower . a building with a clock on a clock tower a large white plane sits on a sidewalk in the kitchen . a row of cars are parked outside the street at an intersection . a woman looks plays in the kitchen an orange and woman walking around a park bench . a man standing in the kitchen at a tv .
TextGAN	a person with a football standing in front of a house an old airplane flying through a blue sky above a house . a man sitting on a bed with a dog and fries inside a car . a group of people riding bicycles down a city street . two motorcycles lined up with green seats in snow . a man wearing glasses wearing glasses and black bookbag riding a horse down a street . a bathroom with a toilet , shower , and toilet , trash can on the wall a cat drinking the back of a white toilet paper a man and motorcycle riders are riding on the road
Our Model	a small bird sit on a white bathroom with a mirror seat . two chefs counter standing in front of a toilet . a modern black and white checkered oven underneath area . looking off from you doors from doors . a white kitchen with chrome space at cabinets . a racing plane in a sky by land on a track . there is a yellow bathroom stands next to a toilet under a mirror . a kitchen with wooden appliances in flight a bathroom that has a mirror and a wall and basket . an image of men are crossing from the car .

Table 2: Randomly selected samples of COCO Image Captions from real data, VAE, FM-GAN, TextGAN, and our model. Text generations are based on greedy decoding for all models.

2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2019. Language gans falling short. In *International Conference on Learning Representations*.

Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. 2018. Adversarial text generation via feature-mover's distance. In *Advances in Neural Information Processing Systems*, pages 4666–4677.

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

Jules Gagnon-Marchand, Hamed Sadeghi, Md Akmal Haidar, and Mehdi Rezagholizadeh. 2019. Salsa-text: self attentive latent space based adversarial text generation. In *Canadian Conference on Artificial Intelligence*, pages 119–131. Springer.

Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin. 2017. Learning generic sentence representations using convolutional neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2390–2400.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

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*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*.

Matt J Kusner and José Miguel Hernández-Lobato. 2016. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*.

Zhongliang Li, Tian Xia, Xingyu Lou, Kaihe Xu, Shaojun Wang, and Jing Xiao. 2019. Adversarial discrete sequence generation without explicit neuralnetworks as discriminators. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3089–3098.

Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*, pages 3155–3165.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Sidi Lu, Lantao Yu, Siyuan Feng, Yaoming Zhu, and Weinan Zhang. 2019. Cot: Cooperative training for generative modeling of discrete data. In *International Conference on Machine Learning*, pages 4164–4172.

Cyprien de Masson d'Autume, Shakir Mohamed, Mihaela Rosca, and Jack Rae. 2019. Training language gans from scratch. In *Advances in Neural Information Processing Systems*, pages 4300–4311.

Weili Nie, Nina Narodytska, and Ankit Patel. 2018. Relgan: Relational generative adversarial networks for text generation. In *International conference on learning representations*.

Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. 2018. On accurate evaluation of gans for language generation. *arXiv preprint arXiv:1806.04936*.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-first AAAI conference on artificial intelligence*.

Yizhe Zhang, Zhe Gan, K. Fan, Z. Chen, Ricardo Henao, Dinghan Shen, and L. Carin. 2017. Adversarial feature matching for text generation. In *ICML*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. *SIGIR*.

A Model Settings

- For the generator, reconstructor, and discriminator, we used long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997).
- Two different fully connected layers are set to linearly transform z into the initial states C_0 and H_0 respectively for the LSTM network of the generator.
- Positional information of 8-dimensions in size (de Masson d'Autume et al., 2019) is concatenated with the word embeddings in each network.
- A dropout is applied to the word embeddings before the word embeddings are fed into each LSTM network of the discriminator.
- All trainable parameters are optimized using Adam (Kingma and Ba, 2015). A weight decay is applied to all trainable parameters of the discriminator using an L2 penalty.
- The prior distribution of Latent space is defined as Gaussian distribution $G(0, 1)$.

B Hyperparameters of Our Model for COCO Image Captions Experiment

- LSTM feature size of the discriminator: 64
- LSTM feature size of the generator: 128
- LSTM feature size of the reconstructor: 128
- Dimension size of latent code z: 8
- Learning rate for Adam: 0.0002
- β_1 for Adam: 0.5
- β_2 for Adam: 0.999
- Minibatch size: 256
- Dropout rate for word embedding: 0.1
- τ_1 in Equation (2): 0.1
- τ_2 in Equation (2): 0.1
- λ in Equation (3): 1.0
- Weight decay rate: 0.0001
- Iteration size: 50000

C Training Algorithm of Our Model

Algorithm 1
GS:=GumbelSoftmax,
VQ:=VectorQuantization,
SG:=StopGradient

Require: initial generator parameter θ, discriminator parameter ϕ, reconstructor parameter ψ.

while θ, ϕ, ψ has not converged **do**
  Sample $x \sim p^*(x)$, $z \sim p(z)$
  for $t = 1, ..., T$ **do**
    if $Argmax(\tilde{x_{t-1}}) = BlankID$ **then**
      $\tilde{x_t} \leftarrow Onehot(BlankID)$
    else
      $\tilde{x_t} \leftarrow GS(P_\theta(\tilde{x_t}|z, \ddot{x_{<t}}))$
    end if
    $\dot{x_t} \leftarrow VQ(\tilde{x_t})$
    $\ddot{x_t} \leftarrow SG(\dot{x_t})$
  end for
  $D_t \leftarrow -\frac{1}{T}\sum_{t=1}^{T} logD_\phi(x_t|x_{<t})$
  $D_f \leftarrow -\frac{1}{T}\sum_{t=1}^{T} log(1 - D_\phi(\tilde{x_t}|\tilde{x_{<t}}))$
  $\mathcal{L} \leftarrow D_t + D_f$
  $\phi \leftarrow Adam(\nabla_\phi \mathcal{L}, \phi)$

  Sample $z \sim p(z)$
  for $t = 1, ..., T$ **do**
    if $Argmax(\tilde{x_{t-1}}) = BlankID$ **then**
      $\tilde{x_t} \leftarrow Onehot(BlankID)$
    else
      $\tilde{x_t} \leftarrow GS(P_\theta(\tilde{x_t}|z, \ddot{x_{<t}}))$
    end if
    $\dot{x_t} \leftarrow VQ(\tilde{x_t})$
    $\ddot{x_t} \leftarrow SG(\dot{x_t})$
  end for
  $D_f \leftarrow \frac{1}{T}\sum_{t=1}^{T} log(1 - D_\phi(\tilde{x_t}|\tilde{x_{<t}}))$
  $\mathcal{L} \leftarrow D_f + \lambda R_\psi(\dot{x_1}, ..., \dot{x_T})$
  $\theta \leftarrow Adam(\nabla_\theta \mathcal{L}, \theta)$
  $\psi \leftarrow Adam(\nabla_\psi \mathcal{L}, \psi)$
end while