Adversarial Text Generation via Sequence Contrast Discrimination

Ke Wang, Xiaojun Wan
Wangxuan Institute of Computer Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{wangke17, wanxiaojun}@pku.edu.cn

Abstract

In this paper, we propose a sequence contrast loss driven text generation framework, which learns the difference between real texts and generated texts and uses that difference. Specifically, our discriminator contains a discriminative sequence generator instead of a binary classifier, and measures the 'relative realism' of generated texts against real texts by making use of them simultaneously. Moreover, our generator uses discriminative sequences to directly improve itself, which not only replaces the gradient propagation process from the discriminator to the generator, but also avoids the time-consuming sampling process of estimating rewards in some previous methods. We conduct extensive experiments with various metrics, substantiating that our framework brings improvements in terms of training stability and the quality of generated texts.

1 Introduction

Generating human-like texts has always been a fundamental problem in the natural language processing field, which is essential to many applications such as machine translation (Bahdanau et al., 2015), image captioning (Fang et al., 2015), and dialogue systems (Reschke et al., 2013). Currently, the dominant approaches are auto-regressive models, such as Recurrent Neural Network (Mikolov et al., 2011), Transformer (Vaswani et al., 2017), and Convolutional Seq2Seq (Gehring et al., 2017), which have achieved impressive performances for the task of language generation using the Maximum Likelihood Estimation (MLE) method. Nevertheless, some studies reveal that such settings may have three main drawbacks: First, the MLE method makes the generative model extremely sensitive to rare samples, which results in the learned distribution being too conservative (Feng and McCulloch, 1992; Ahmad and Ahmad, 2019). Second, auto-regressive generation models suffer from exposure bias (Bengio et al., 2015) due to the dependence on the previous sampled output during the inferring phase. Third, they only consider the word-level objective and may fail to guarantee some sentence-level goals, such as realism, preserving semantic consistencies, long-range semantic structure, and so on (Ranzato et al., 2016).

Recently, lots of recent studies (Yu et al., 2017; Che et al., 2017; Lin et al., 2017; Zhang et al., 2017; Chen et al., 2018; Wang and Wan, 2018; Ke et al., 2019; Nie et al., 2019; Wang and Wan, 2019; Wang et al., 2019) try to apply generative adversarial networks (GAN) (Goodfellow et al., 2014) in text generation, which uses discriminator networks as loss functions to ensure these higher-level objectives. However, the discreteness of texts makes it difficult for the gradient to pass from the discriminator to the generator. The current solution is mainly based on reinforcement learning (Yu et al., 2017) or differentiable sampling functions (Jang et al., 2017). In addition, considering the complexity of the language, the generator is easily much weaker than the discriminator in practice, making it difficult to obtain a clear optimization direction from the discriminator and learn from scratch.

In this paper, by borrowing techniques from contrastive learning (Hadsell et al., 2006; Hénaff et al., 2019; He et al., 2019; Chen et al., 2020), we propose a sequence contrast loss driven adversarial learning framework for text generation, SLGAN. In our framework, the discriminator D is not just a simple binary classifier, but a Siamese network composed of a sequence generator G_d, which can provide sequences with discriminative information. In other words, our discriminator D measures the gap between the generated texts and the real texts, rather than simply predicting the probability of the generated data (by generator G) being real. Specifically, these discriminative sequences with well-formed textual structure information can be used to

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 47–53
November 16 - 20, 2020.

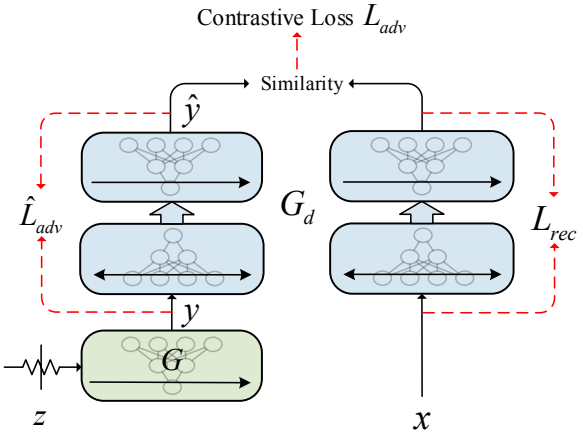

Figure 1: Illustration of SLGAN. $\boldsymbol{x}$ is the real text sampled from $\mathcal{D}$. $\boldsymbol{y}$ is the text generated by G, and $\hat{\boldsymbol{y}}$ is the discriminative text generated by G_d.

measure the 'relative realism' (sequence contrast loss) of the generated texts against the real texts, and further improve the generator G. Intuitively, the discriminator can not only tell if the text generated by the generator is good, but also teach the generator in which direction to generate better text. Our motivations are two-fold: 1) Our discriminator can provide better discriminative information to the generator because it observes both 'fake' and 'real' data simultaneously. 2) Compared to other gradient propagation strategies based on reinforcement learning or differentiable sampling functions, the contrastive loss between generated sequences and discriminative sequences can improve the generator more time-efficiently and steadily.

We conduct experiments on both synthetic and real datasets, and use various metrics (i.e., fluency, novelty, generalization, diversity, human evaluation, and learning curve) to show that our approach not only produce more realistic samples but also greatly stabilize the adversarial training process.

2 Method

The architecture of our proposed model is depicted in Figure 1. The whole framework can be divided into two adversarial learning objectives: generator learning and discriminator learning. The goal of the discriminator D is to learn the difference ('relative realism') between fake texts ($\boldsymbol{y}$, texts generated by generators) and real texts ($\boldsymbol{x}$). While the goal of the generator G is to use this difference (discriminative sequences) to generate more realistic texts, which contains a word-level item ($\mathcal{L}_{mle}$) and a sentence-level item ($\hat{\mathcal{L}}_{adv}$).

To achieve the above goals, we start with two things. One is that discriminator D observes and uses both 'fake' and 'real' data at the same time, rather than considering them in an alternating fashion. The other is that the inside of the discriminator is not a binary classifier, but a sequence generator G_d. G_d aims to generate discriminative sequence $\hat{\boldsymbol{y}}$, which can be considered as a sequence representation to be used for better measurement of 'relative realism'. To some extent, G_d can be seen as an 'amplifier', and the closer the input text is to real texts, the less it changes. Further, $\hat{\boldsymbol{y}}$ not only can be used to measure the 'relative realism' of generated texts against real texts, but also can be used to directly affect G through sequence contrast loss. Therefore, by calculating the contrastive loss, the gradient back-propagation process from the discriminator to the generator is avoided, which is of significant importance in adversarial learning.

Discriminator Learning: The contrastive loss of our discriminator takes the output of the discriminative sequence generator G_d for a positive example (real texts $\boldsymbol{x}$), and calculates its similarity to an example of the same class ($\boldsymbol{x}$) and contrasts that with the distance to negative examples ($\boldsymbol{y}$, texts generated by generators):

$$\mathcal{L}_{discriminator} = \lambda_i Sim_s - Sim_d, \quad (1)$$

where Sim_d and Sim_s are the similarity measure of a pair of dissimilar points and a pair of similar points, respectively. $\lambda_i = \max\{\lambda, 1 - \alpha i\}$ is the coefficient to balance two terms at i-th epoch. It is worth noting that Eq 1 degenerates into the vanilla GAN's adversarial loss when $\lambda_i = 0$.

We use the KL-divergence to measure how similar two word distributions of generated sequences are to each other, and the inter-class loss Sim_d is:

$$Sim_d = \mathcal{L}_{adv} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}, z\sim\mathcal{P}} [||G_d(\boldsymbol{x};\theta_d) - G_d(G(z;\theta_g);\theta_d)||_{kl}], \quad (2)$$

where z is sampled from a noise distribution $\mathcal{P}$. The outputs of G_d is not a probability between 0 and 1, but a representation with more discriminative information. That is, the generator G_d in our discriminator takes input of the real data $\boldsymbol{x}$ or the fake data $G(z;\theta_g)$, and then generates word sequence $\hat{\boldsymbol{y}}$ for each input.

In addition, we consider making $\hat{\boldsymbol{y}}$ meaningful, with the purpose that it can be used not only to discriminate but also to represent 'realism' features.

We hence rewrite the intra-class loss Sim_s with a similar idea as:

$$Sim_s = \mathcal{L}_{rec} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[||G_d(\boldsymbol{x};\theta_d) - \boldsymbol{x}||_{kl}]. \quad (3)$$

In practice, we add noise to $\boldsymbol{x}$ by randomly replacing input word with the noise word ($< unk >$).

Generator Learning: The loss function of our generator includes two terms: one term ($\mathcal{L}_{mle}$) is to concern word-level fitness and another term ($\hat{\mathcal{L}}_{adv}$) is to ensure a higher level of 'realism' resembling qualities:

$$\mathcal{L}_{generator} = \mathcal{L}_{mle} + \hat{\lambda}_i \hat{\mathcal{L}}_{adv}, \quad (4)$$

where $\hat{\lambda}_i = \hat{\lambda}(i/k)$ is the balance coefficient and k is the number of all epochs.

Given a training sentence $\boldsymbol{x} = \{x_0, \dots x_t, \dots\}$ with length $|\boldsymbol{x}|$, the word-level objective $\mathcal{L}_{mle}$ is to minimize the negative log-likelihood loss as follows:

$$\mathcal{L}_{mle} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[-\sum_{t=1}^{|\boldsymbol{x}|-1} \log G(x_t|\boldsymbol{x}_{0:t-1})], \quad (5)$$

where $G(x_t|\boldsymbol{x}_{0:t-1})$ denotes the probability that the output of G is x_t under the condition of the former given sequence $\boldsymbol{x}_{0:t-1} = \{x_0, x_1, \dots x_{t-1}\}$ at time step t. While in the inference phase, generator G will take the previous sampled output $\boldsymbol{y}_{0:t-1}$ as the input at time step t. Here G is an auto-regressive generation model (e.g., RNN and its variants (Mikolov et al., 2011; Hochreiter and Schmidhuber, 1997; Chung et al., 2014), Transformer (Vaswani et al., 2017) and Convolutional Seq2Seq (Gehring et al., 2017)).

Furthermore, the other goal of generator G is to minimize Sim_d in Eq 2, with the intuition that using a discriminator network to learn the loss function of sentence-level properties (e.g., long-range semantic structure, preserving semantic consistencies, etc.) over time, rather than explicitly formulating these properties. According to the discriminator's loss (Eq 1), the closer $G(z;\theta_g)$ is to $\boldsymbol{x}$, the closer $G_d(G(z;\theta_g);\theta_d)$ is to $G(z;\theta_g)$. As such, we resort to an approximation approach to define the generator's adversarial loss as:

$$\hat{\mathcal{L}}_{adv} = \mathbb{E}_{z\sim\mathcal{P}}[||G_d(G(z;\theta_g);\theta_d) - G(z;\theta_g)||_{kl}]. \quad (6)$$

In this way, we can directly guide the generation of G by measuring the sequence contrast loss of the output between G and G_d, which not only avoids the gradient back-propagation process from the discriminator to the generator, but also makes the generator use the discriminator's discriminative information more effectively.

3 Experiments

3.1 Setup

In this study, we use Texygen (Zhu et al., 2018), a benchmarking platform that implements a majority of GAN-based text generation models and covers a set of metrics, to standardize comparisons with other GAN models. We compare SLGAN with several typical and state-of-the-art unsupervised generic text generation models, including MLE (Mikolov et al., 2011), SeqGAN (Yu et al., 2017), MaliGAN (Che et al., 2017), RankGAN(Lin et al., 2017), GSGAN (Kusner and Hernández-Lobato, 2016), TextGAN (Zhang et al., 2017), LeakGAN (Guo et al., 2018). Without loss of generality, we evaluate our model on two benchmark datasets: a synthetic dataset and a real text dataset (COCO image caption (Lin et al., 2014)).

3.1.1 Implementation Details

In our model, the default initial parameters of all generators follow a Gaussian distribution $\mathcal{N}(0,1)$. The total number of adversarial training epochs is 200 and the sampling temperature is set to 1.0. We set $\lambda = 1.0$ and $\alpha = 0.1$, and G_d is a seq2seq model based on single-layer RNN-GRU and Luong attention. $\hat{\lambda}$ is set to 1.0, and the number of all epochs $k = 200$, based on performance. G is a single-layer RNN-GRU network and can be easily extended to other types of generators as well. We implement our model based on Pytorch and use a TITAN X graphic card for learning.

3.1.2 Dataset Statistics

A summary of statistics for each dataset is provided in Table 1. To be fair, on the synthetic and real datasets, we train all models using the same-size (size = 10,000) training set and use the models to generate the same-size (size = 10,000) set of sentences for evaluation.

3.2 Synthetic Data Experiment

Here we use the synthetic dataset used by Texygen (Zhu et al., 2018), which consists of a set of sequential tokens which can be seen as the simulated

Datasets	#Train	#Test	#Vocab	Max-Length
Synthetic	10,000	10,000	5,000	20
Real	10,000	10,000	4,684	38

Table 1: Statistics for the synthetic and real dataset we use.

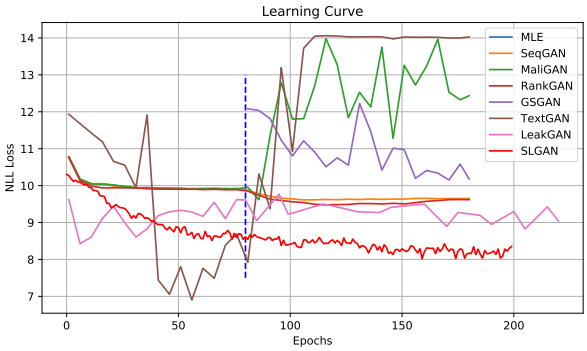

Figure 2: The illustration of learning curves. Dotted line is the end of pre-training for baseline models except GSGAN and TextGAN.

data comparing to the real-word language data. We compare our model with various models on this dataset, as shown in Figure 2. We observe that our model outperforms all other competitors with a large margin and the NLL loss declines rapidly and steadily from the beginning, demonstrating that our model is more stable and time-efficient.

3.3 Real Data Experiment

We also conduct experiments on a real-world dataset (i.e., COCO image caption), and present a variety of evaluation methods for a comprehensive comparison.

Fluency: We show the perplexity of generated sentences in Figure 3, which shows that our model is good at keeping the fluency of sentences.

Novelty: We use the novelty measure in (Wang and Wan, 2018) to investigate how different the generated sentences and the training corpus are. From

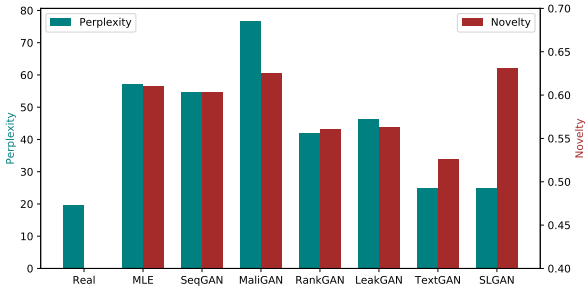

Figure 3: Comparison of fluency (lower perplexity means better fluency) and novelty of generated sentences.

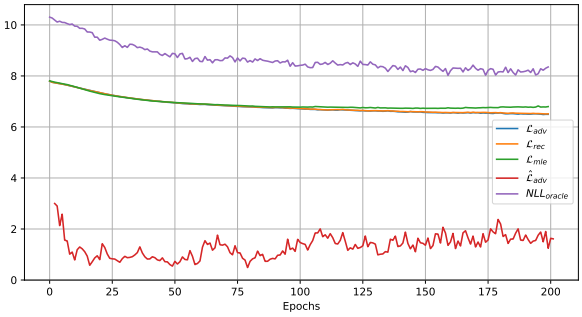

Figure 4: Different loss curves during adversarial training process.

the results in Figure 3, we observe that our model has a better ability to generate new sequences.

Generalization: Same as Texygen, we also evaluate BLEU (Papineni et al., 2002) between generated sentences and the test set to see the generalization capacity of different models. The BLEU scores are shown in Table 2, which show that our model has a rather good generalization capacity. Moreover, as the order (i.e., n) of n-gram rises, the corresponding BLEU performance of our model does not drop as fast as other models.

Diversity: We use Self-BLEU to evaluate how one sentence resembles the rest in a generated collection. From Table 2, we see that the sentences generated by our model have the lowest Self-BLEU score, indicating the highest diversity.

Human Evaluation: We randomly extract 100 sentences from the generated sentences and then hire three workers on Amazon Mechanical Turk to rate each of them according to its 'grammaticality', 'topicality', and 'overall' aspects, where 'topicality' indicates the semantic consistency of the entire sentence. The rating score ranges from 1 to 5, and 5 is the best. As shown in Table 2, our model outperforms several baseline models, especially in the aspects of 'topicality' and overall quality.

Training Stability: We also show the different loss curves of our model during the adversarial training process in Figure 4. As can be seen in Figure 4, the adversarial process between G and G_d is quite stable. Firstly, the discriminator is not powerful enough to let loss $\mathcal{L}_{adv}$ fall to 0, because it does more things than a simple binary prediction. Secondly, the ability ($\hat{\mathcal{L}}_{adv}$) of the generator to attempt to deceive the discriminator has been fluctuating. As the discriminator has been getting better, we argue that the capabilities of the generator are constantly being enhanced, that is, more similar to real texts.

Models	Generalization ↑				Diversity ↓				Human Evaluation ↑		
	BLEU-2	BLEU-3	BLEU-4	BLEU-5	BLEU-2	BLEU-3	BLEU-4	BLEU-5	Grammaticality	Topicality	Overall
MLE	0.731	0.497	0.305	0.189	0.916	0.769	0.583	0.408	3.68	2.03	2.57
SeqGAN	0.745	0.498	0.294	0.180	0.950	0.840	0.670	0.498	3.73	3.29	3.36
MaliGAN	0.673	0.432	0.257	0.159	0.918	0.781	0.606	0.437	3.83	2.32	2.79
RankGAN	0.743	0.467	0.264	0.156	0.959	0.882	0.762	0.618	3.94	3.83	3.78
LeakGAN	0.746	**0.528**	**0.355**	0.230	0.966	0.913	0.848	0.780	4.08	4.04	3.96
TextGAN	0.593	0.463	0.277	0.207	0.942	0.931	0.804	0.746	**4.23**	3.46	3.99
SLGAN	**0.753**	0.502	0.348	**0.251**	**0.751**	**0.573**	**0.422**	**0.313**	3.93	**4.29**	**4.16**

Table 2: Results on real dataset. ↓ means the smaller the better, and ↑ is the opposite. The best scores are bold and our scores are underlined. The kappa coefficient of the three workers is 0.63.

4 Conclusion and Future Work

In this study, we propose a sequence contrast loss for adversarial text generation, where the discriminator outputs discriminative sequences rather than binary classification probabilities. Extensive experimental results demonstrate that our model brings improvements in training stability and the quality of generated texts.

In future work, we will expand our method to have specific targets, to benefit more conditional text generation tasks (e.g., sentimental text generation, dialogue response generation).

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772036), Beijing Academy of Artificial Intelligence (BAAI) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We appreciate the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

Kaisar Ahmad and Sheikh Parvaiz Ahmad. 2019. A comparative study of maximum likelihood estimation and bayesian estimation for erlang distribution and its applications. In *Statistical Methodologies*. IntechOpen.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *NeurIPS 2015*, pages 1171–1179.

Tong Che, Yanran Li, Ruixiang Zhang, R. Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. 2017. Maximum-likelihood augmented discrete generative adversarial networks. *CoRR*, abs/1702.07983.

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 *NeurIPS 2018*, pages 4671–4682.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Hao Fang, Saurabh Gupta, Forrest N. Iandola, Rupesh Kumar Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. 2015. From captions to visual concepts and back. In *CVPR 2015*, pages 1473–1482.

Ziding Feng and Charles E McCulloch. 1992. Statistical inference using maximum likelihood estimation and the generalized likelihood ratio when the true parameter is on the boundary of the parameter space. *Statistics & Probability Letters*, 13(4):325–332.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML 2017*, pages 1243–1252.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NeurIPS 2014*, 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 2018*, pages 5141–5148.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR 2006*, pages 1735–1742.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2019. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722.

Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aäron van den Oord. 2019. Data-efficient image recognition with contrastive predictive coding. *CoRR*, abs/1905.09272.

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

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *ICLR 2017*. OpenReview.net.

Pei Ke, Fei Huang, Minlie Huang, and Xiaoyan Zhu. 2019. ARAML: A stable adversarial training framework for text generation. In *EMNLP-IJCNLP 2019*, pages 4270–4280. Association for Computational Linguistics.

Matt J. Kusner and José Miguel Hernández-Lobato. 2016. GANS for sequences of discrete elements with the gumbel-softmax distribution. *CoRR*, abs/1611.04051.

Kevin Lin, Dianqi Li, Xiaodong He, Ming-Ting Sun, and Zhengyou Zhang. 2017. Adversarial ranking for language generation. In *NeurIPS 2017*, pages 3158–3168.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In *ECCV 2014*, pages 740–755.

Tomas Mikolov, Stefan Kombrink, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *ICASSP 2011*, pages 5528–5531.

Weili Nie, Nina Narodytska, and Ankit Patel. 2019. Relgan: Relational generative adversarial networks for text generation. In *ICLR 2019*. OpenReview.net.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL 2002*, pages 311–318.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *ICLR 2016*.

Kevin Reschke, Adam Vogel, and Dan Jurafsky. 2013. Generating recommendation dialogs by extracting information from user reviews. In *ACL 2013*, pages 499–504.

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 *NeurIPS 2017*, pages 6000–6010.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *NeurIPS 2019*, pages 11034–11044.

Ke Wang and Xiaojun Wan. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI 2018*, pages 4446–4452.

Ke Wang and Xiaojun Wan. 2019. Automatic generation of sentimental texts via mixture adversarial networks. *Artif. Intell.*, 275:540–558.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI 2017*, pages 2852–2858.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *ICML 2017*, pages 4006–4015.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *SIGIR 2018*, pages 1097–1100.

A Appendices

A.1 Implementation Details

In our model, the default initial parameters of all generators follow a Gaussian distribution $\mathcal{N}(0, 1)$. The total number of adversarial training epochs is 200 and the sampling temperature is set to 1.0. We set $\lambda = 1.0$ and $\alpha = 0.1$, and G_d is a seq2seq model based on single-layer RNN-GRU and Luong attention. $\hat{\lambda}$ is set to 1.0, and the number of all epochs $k = 200$, based on performance. G is a single-layer RNN-GRU network and can be easily extended to other types of generators as well. We implement our model based on Pytorch and use a TITAN X graphic card for learning.

A.2 Generated Cases

In Table 3, we show example sentences generated by different models trained on a real-world dataset. From the examples, we see that: 1) Although the sentence produced by the MLE method is longer, it may have unreadable and unreasonable problems. 2) The sentences generated by LeakGAN and TextGAN are more readable, but they are not diversified and relatively short. 3) In particular, compared with all benchmark methods, the sentences

MLE	a store is blue sink in a water bottle . ***(Unreasonable)***
	serious air force jet mid flight during a cobblestone day , where a flooded street
	a simple bathroom with some wood cupboards .
	a girafee is standng in the spot for a village in parking spot with four hinged cakes trees
	a jet jet flying away on the runway , in the sky .
	a fat orange motorcycle is low building .
	a bathroom with a sink , a sink , refrigerator and the walls . ***(Unreadable)***
	a living room with a blue roof and green traffic lights blue .
	person sitting in a commercial plane at night .
LeakGAN	a view of a parking desk with two plungers
	a desk with multiple large monitors . ***(Very short)***
	a woman wearing a glass is sitting on a cupboard .
	a kitchen with a shelf area .
	a man tinkers with his ear .
	a white stove top open from a wood oven .
	a group of men talking .
	a kitchen with a shelf area . ***(Repeated)***
	two people sitting on .
TextGAN	a man riding a motorcycle . ***(Very short)***
	is to a bathroom with a sink . ***(Unreadable)***
	a man is on a motorcycle .
	a white toilet a sink .
	with a sink and a table .
	a motorcycle in a blue sky .
	a bathroom with a sink .
	a man is sitting on a motorcycle . ***(Repeated)***
	a bathroom with a sink .
SLGAN	a group of people sat in front of the house together .
	several people stood in front of the bicycle .
	a person is holding a monitor range in the kitchen .
	a woman is riding a motorcycle on the street .
	three adults sat in his car with hats .
	two people in a public parking lot .
	white bathtub , toilet and basin under the bathroom wall .
	an old brick building with a wooden manufacturer next to it .
	a motor scooter parked in the street with a crowd waiting for a parade.

Table 3: Example sentences generated by different models.

produced by our model are more readable, diversified and of better quality.