Diversity-Promoting GAN: A Cross-Entropy Based Generative Adversarial Network for Diversified Text Generation

Jingjing Xu, Xuancheng Ren, Junyang Lin, Xu Sun
MOE Key Lab of Computational Linguistics, School of EECS, Peking University
{jingjingxu,renxc,linjunyang,xusun}@pku.edu.cn

Abstract

Existing text generation methods tend to produce repeated and "boring" expressions. To tackle this problem, we propose a new text generation model, called Diversity-Promoting Generative Adversarial Network (DP-GAN). The proposed model assigns low reward for repeatedly generated text and high reward for "novel" and fluent text, encouraging the generator to produce diverse and informative text. Moreover, we propose a novel language-model based discriminator, which can better distinguish novel text from repeated text without the saturation problem compared with existing classifier-based discriminators. The experimental results on review generation and dialogue generation tasks demonstrate that our model can generate 相当大的 substantially more diverse and informative text than existing baselines.[1]

1 Introduction

Text generation is an important task in Natural Language Processing (NLP) as it lays the foundation for many applications, such as dialogue generation, machine translation (Ma et al., 2018b; Xu et al., 2018a), text summarization (Ma et al., 2018a), and table summarization (Liu et al., 2017). In these tasks, most of the systems are built upon the sequence-to-sequence paradigm (Sutskever et al., 2014), which is an end-to-end model that encodes a source sentence to a dense vector and then decodes the vector to a target sentence. The standard training method is based on Maximum Likelihood Estimation (MLE).

Although being widely applied, the conventional MLE training causes systems to repeatedly generate "boring" sentences, which usually are expressions with high frequency (e.g., "I am sorry" in dialogue generation (Li et al., 2016)). The major reason is that MLE encourages the model to overproduce high-frequency words.[2] The overestimation of high-frequency words discourages the model from generating low-frequency but meaningful words in real data, which makes generated text tend to be repeated and "boring".

To tackle this problem, we propose a new model for diversified text generation, called DP-GAN. The key idea is to build a discriminator that is responsible for giving reward to the generator based on the novelty of generated text. We consider the text that is frequently generated by the generator as the low-novelty text and the text that is uncommon in the generated data as the high-novelty text. Considering most of the real-world sentences are novel and fluent, we treat the real-world text as the positive example and the generated text as the negative example to train the discriminator. Such training mechanism encourages the discriminator to give higher reward for the text that looks like real-world data. The reward is fed back to the generator, which promotes the generator to generate diverse and fluent text via policy gradient. In this framework, a good discriminator that can assign reasonable reward for the generator is a critical component.

However, directly applying a classifier as the discriminator like most existing GAN models (e.g., SeqGAN (Yu et al., 2017)) cannot achieve satisfactory performance. The main problem is that the reward given by the classifier cannot reflect the novelty of text accurately. First, most existing classifier-based discriminators take the probability of a sequence being true as the reward. When a sentence fits the distribution of real-world

[1] The code is available at https://github.com/lancopku/DPGAN

[2] For example, the frequency ratios of "the", "and", "was" are 4.2%, 3.2%, 1.5% in real data, and they go up to 7.1%, 4.6%, 5.3% in the MLE generated data on our review generation task.

text and is far from the generated data, the reward saturates and scarcely distinguishes the difference between these novel sentences. For example, for a sentence A with mildly high novelty and a sentence B with extremely high novelty, the classifier cannot tell the difference and gives them saturated reward: 0.997 and 0.998. Second, in our tasks, we find that a simple classifier can reach very high accuracy (almost 99%), which makes most generated text receive reward around zero because the discriminator can identify them with high confidence. It shows that the classifier also cannot distinguish the difference between low-novelty text. The reason for this problem is that the training objective of the classifier-based GAN is in fact minimizing the Jensen-Shannon Divergence (JSD) between the distributions of the real data and the generated data (Nowozin et al., 2016). If the accuracy of classifier is too high, JSD fails to measure the distance between the two distributions, and cannot give reasonable reward to the model for generating real and diverse text (Arjovsky et al., 2017).

Instead of using a classifier, we propose a novel language-model based discriminator and use the output of the language model, cross-entropy, as the reward. The main advantage of our model lies in that the cross-entropy based reward for novel text is high and does not saturate, while the reward for text with low novelty is small but discriminative. The analysis of the experimental results shows that our discriminator can better distinguish novel text compared with traditional classifier-based discriminators.

Our contributions are listed as follows:

- We propose a new model, called DP-GAN, for diversified text generation, which assigns low reward for repeated text and high reward for novel and fluent text.
- We propose a novel language-model based discriminator that can better distinguish novel text from repeated text without the saturation problem.
- The experimental results on review generation and dialogue generation tasks show that our method can generate substantially more diverse and informative text than existing methods.

2 Related Work

A great deal of attention has been paid to developing data-driven methods for natural language dialogue generation. Conventional statistical approaches tend to rely extensively on hand-crafted rules and templates, require interaction with humans or simulated users to optimize parameters, or produce conversation responses in an information retrieval fashion. Such properties prevent training on the large corpora that are becoming increasingly available, or fail to produce novel natural language responses.

Currently, a popular model for text generation is the sequence-to-sequence model (Sutskever et al., 2014; Cho et al., 2014). However, the sequence-to-sequence model tends to generate short, repetitive (Lin et al., 2018), and dull text (Luo et al., 2018). Recent researches have focused on developing methods to generate informative (Xu et al., 2018b) and diverse text (Li et al., 2017, 2016; Guu et al., 2017; Shao et al., 2017). Reinforcement learning is incorporated into the model of conversation generation to generate more human-like speeches (Li et al., 2017). Moreover, there are also other methods to improve the diversity of the generated text by using mutual-information, prototype editing, and self attention (Li et al., 2016; Guu et al., 2017; Shao et al., 2017).

In this paper, to handle this problem, we propose to use adversarial training (Goodfellow et al., 2014; Denton et al., 2015; Li et al., 2017), which has achieved success in image generation (Radford et al., 2015; Chen et al., 2016; Gulrajani et al., 2017; Berthelot et al., 2017). However, training GAN is a non-trivial task and there are some previous researches that investigate methods to improve training performance, such as Wasserstein GAN (WGAN) (Arjovsky et al., 2017) and Energy-based GAN (EGAN) (Salimans et al., 2016; Gulrajani et al., 2017; Zhao et al., 2017; Berthelot et al., 2017). GAN in text generation has not shown significant improvement as it has in computer vision. This is partially because text generation is a process of sampling in discrete space where the normal gradient descent solution is not available, which makes it difficult to train. There are some researches that focus on tackling this problem. SeqGAN (Yu et al., 2017) incorporates the policy gradient into the model by treating the procedure of generation as a stochastic policy in reinforcement learning. Ranzato et al. (2016)

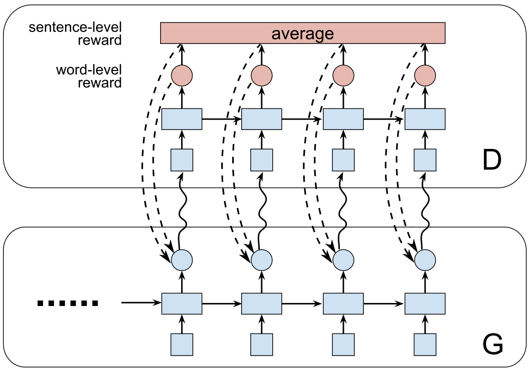

Figure 1: Illustration of DP-GAN. Lower: The generator is trained by policy gradient where the reward is provided by the discriminator. Upper: The discriminator is based on the language model trained over the real text and the generated text.

trains the sequence-to-sequence model with policy gradient for neural machine translation. Bahdanau et al. (2017) applies the actor-critic model on the same task.

3 Diversity-Promoting GAN

The basic structure of our DP-GAN contains a generator that is responsible for generating text and a discriminator that discriminates between the generated text and the real text. The sketch of DP-GAN is shown in Figure 1.

3.1 Overview

The generator G_θ is based on a sequence-to-sequence structure. Given a sentence as input, the generator is capable of generating long text, which contains multiple sentences of various lengths. To put it formally, given the input sentence $x_{1:m} = (x_1, x_2, x_3, ..., x_m)$ of m words from Γ, the vocabulary of words, the model generates the text of T sentences $Y_{1:T} = (y_1, ..., y_t, ..., y_T)$, where y_t from Λ, the set of candidate sentence. The term $y_t = (y_{t,1}, ..., y_{t,K})$ is the t^{th} sentence, where $y_{t,K}$ is the K^{th} word.

The discriminator D_ϕ is a language model. The output of the language model, cross entropy, is defined as the reward to train the generator. Our reward consists of two parts, the reward at the sentence level and that at the word level. With the discriminator and the reward function, we train the generator by reinforcement learning. A sketch of training DP-GAN is shown in Algorithm 1. The details are described as follows.

模型的训练这一块 我还是觉得放在后面比较好，一般放在methods的后面部分

Algorithm 1 The adversarial reinforcement learning algorithm for training the generator G_θ and the discriminator D_ϕ.

```
Initialize $G_\theta$, $D_\phi$ with random weights $\theta$, $\phi$
Pre-train $G_\theta$ using MLE on a sequence dataset $\mathcal{D} = (X, Y)$
Generate samples using $G_\theta$ for training $D_\phi$
Pre-train $D_\phi$ by Eq. (1)
$N$ = number of training iterations
$M$ = number of training generator
$K$ = number of training discriminator
for each $i = 1, 2, ..., N$ do
    for each $j = 1, 2, ..., M$ do
        Generate a sequence $Y_{1:T} \sim G_\theta$
        Compute rewards by Eq. (2) and Eq. (3)
        Update generator via policy gradient Eq. (5)
        Sample a sequence $Y_{1:T} \sim \mathcal{D}$
        Compute rewards by Eq. (2) and Eq. (3)
        Update generator parameters via Eq. (5)
    end for
    for each $j = 1, 2, ..., K$ do
        Generate samples using $G_\theta$
        Train discriminator $D_\phi$ by Eq. (1)
    end for
end for
```

这个算法没有看太懂，不是交替训练吗？为什么D和G的训练次数还不一样？

每次生成一个样本？？

3.2 Generator

For the concern of real-world applications, this paper assumes that the output of the model can be long text made up of multiple sentences. In order to generate multiple sentences, we build a standard hierarchical LSTM decoder (Li et al., 2015). The two layers of the LSTM are structured hierarchically. The bottom layer decodes the sentence representation and the top layer decodes each word based on the output of the bottom layer. The attention mechanism is used for word decoding (Bahdanau et al., 2014; Luong et al., 2015).

3.3 Discriminator

Most existing GAN models use a binary classifier as the discriminator. The probability of being true is regarded as the reward (Li et al., 2016; Yu et al., 2017). Different from that, we propose a language-model based discriminator D_ϕ that builds on a unidirectional LSTM. We use the output of the language model, cross-entropy, as the reward. Specifically, given a sentence y_t, the cross-entropy based reward for the k^{th} word is calculated as

$$\boxed{R(y_{t,k}) = -\log D_\phi(y_{t,k} | y_{t,<k})}$$

R(yt,k)=D(yt,k|yt,<k)
我可能应该使用这个reward

We maximize the reward of real-world text and minimize the reward of generated text to train the discriminator. The reason of minimizing the reward of generated text is that, we expect the text that is repeatedly generated by the generator can be identified by the discriminator and get lower reward. The motivation of maximizing the reward of real-world data lies in that, we expect not only the uncommon text in the generated data can get high reward, but also low-quality text can be punished to some extend. Considering the real-world text is diverse and fluent, we maximize the reward of real-world text to encourage the discriminator to give high reward for the text that looks like the real-world data. Therefore, such training mechanism avoids the problem of novel but low-quality text getting high reward. The loss function of the discriminator is formulated as follows:

$$\boxed{\begin{aligned} &J(\phi) = \\ &-(E_{Y\sim p_{data}}[R(Y)] - E_{Y\sim G_\theta}[R(Y)]) \end{aligned}} \tag{1}$$

where $R(Y)$ stands for the averaged reward of Y.

3.4 Reward

Our reward function consists of two parts, the sentence-level reward and the word-level reward, which are illustrated as follows.

3.4.1 Sentence-Level Reward

For a sentence y_t of K words, the reward at the sentence level is the averaged reward of each word:

$$\boxed{R(y_t) = -\frac{1}{K}\sum_{k=1}^{K}\log D_\phi(y_{t,k}|y_{t,<k})} \tag{2}$$

In contrast, the reward of the existing classifier-based discriminators (Li et al., 2016; Yu et al., 2017) is calculated as follows:

对于我的任务，我可能要使用这个
作为reward，但是我也是word级别的

$$\boxed{R(y_t) = D_\phi(true|y_t)}$$

where D_ϕ is a binary classifier judging how likely y_t is from the real-world data.

The major problem of the classifier-based discriminator is that the reward cannot reflect the novelty of text accurately. First, the reward for high-novelty text is easy to saturate, which scarcely distinguishes the difference between novel text. Second, we find that the discriminator can easily achieve very high accuracy on identifying the generated text, which makes most of them get reward around zero. It shows that the classifier still cannot tell the difference between the text with low novelty.

On the contrary, the analysis of experimental result shows that our proposed discriminator can better distinguish high-novelty text from low-novelty text without the saturation problem. The reward for high-novelty text is high and does not saturate while the reward for low-novelty text is small but discriminative.

3.4.2 Word-Level Reward

Considering that the reward for different words in a sentence y_t should be different, we further propose to use the reward at the word level as follows:

$$\boxed{R(y_{t,k}|y_{t,<k}) = -\log D_\phi(y_{t,k}|y_{t,<k}) \quad (3)}$$

It can be found that the classifier-based discriminator only provides reward for the finished sequence. Thus, for a sequence of length T, to evaluate the action-value for a word at the time step t, Monte Carlo Search (MCS) with a roll-out policy G_θ is usually applied to sample the unknown last $T-t$ tokens (Yu et al., 2017). However, this could be computationally expensive because the time complexity is $O(T^2)$. On the contrary, our discriminator can calculate the reward of all words with the time complexity of $O(T)$, which is more computationally efficient.

3.5 Policy Gradient Training

The loss function of the generator (policy) is to maximize the reward from the start state s_0 to the end state (Sutton et al., 1999):

$$\boxed{\begin{aligned} J(\theta) &= \sum_{t=1}^{T} E[R_{t,K}|s_{t-1},\theta] \\ &= \sum_{t=1}^{T}\sum_{y_{t,1}} G_\theta(y_{t,1}|s_{t-1}) Q_{D_\phi}^{G_\theta}(s_{t-1},y_{t,1}) \\ &\qquad\qquad\qquad\qquad\qquad\qquad (4) \end{aligned}}$$

这个公式没看太
懂，Q那么一堆是啥
玩意？

where $R_{t,K} = \sum_{k=1}^{K}\gamma^{k-1}R(y_t)R(y_{t,k})$ is the total reward for a complete sentence, including both the sentence-level and the word-level rewards. The term $Q_{D_\phi}^{G_\theta}(s_{t-1},y_{t,1})$ is estimated by $R_{t,1}$. The term γ is the discount rate and s_t is the initial state.

In this paper, we use the policy gradient method (Williams, 1992). The gradient of Eq. (4) is approximated as follows:

$$\boxed{\begin{aligned} &\nabla_\theta J(\theta) \simeq \\ &\sum_{t=1}^{T}\sum_{k=1}^{K}\gamma^{k-1}R_{t,k}\nabla_\theta \log G_\theta(y_{t,k}|y_{t,<k}) \end{aligned}} \tag{5}$$

where $R_{t,k} = \sum_{i=k}^{K} \gamma^{i-1} R(y_t) R(y_{t,i})$ is the total reward starting from step k.

Following previous work (Li et al., 2017), we also use teacher forcing (Bengio et al., 2015) to train the generator. In teacher forcing, the decoder receives the real-world text as input at each time step. The loss function of teacher forcing is the same with that of policy gradient training. The only difference is that the text is generated from G_θ in policy gradient training but from the real data in teacher forcing.

4 Experiment

We evaluate DP-GAN on two real-world natural language generation tasks, review generation and dialogue generation. We first introduce the dataset, the training details, the baselines, and the evaluation metrics. Then, we compare our model with the state-of-the-art models. Finally, we show the experimental results and provide the detailed analysis.

4.1 Datasets

Yelp Review Generation Dataset (Yelp): This dataset is provided by Yelp Dataset Challenge.[3] In our version of review generation, the model should generate a paragraph based on a given sentence. We build a new dataset for this task by splitting the data into two parts. In each review, we take the first sentence as the input text, and the following sentences as the target text. The processed Yelp dataset contains 1,400K, 400K, and 12K pairs for training, validation, and testing, respectively.

Amazon Review Generation Dataset (Amazon): This dataset is provided by McAuley and Leskovec (2013). It consists of review information of fine foods from Amazon. Like Yelp, we process this dataset by extracting the first sentence as the source text and the rest as the target text. The processed Amazon dataset contains 400K, 100K, and 12K pairs for training, validation, and testing, respectively.

OpenSubtitles Dialogue Dataset (Dialogue): This dataset[4] is used for dialogue generation. Following previous work, we treat each turn in the dataset as the target text and the two previous sentences as the source text. We remove the pairs whose response is shorter than 5 words. We randomly sample 1,800K, 500K, and 12K turns for training, validation, and testing, respectively.

Yelp	Token	Dist-1	Dist-2	Dist-3	Dist-S
MLE	151.2K	1.2K	3.9K	6.6K	3.9K
PG-BLEU	131.1K	1.1K	3.3K	5.5K	3.1K
SeqGAN	140.5K	1.1K	3.5K	6.1K	3.6K
DP-GAN(S)	**438.6K**	1.7K	7.5K	15.7K	10.6K
DP-GAN(W)	271.9K	2.8K	14.8K	29.0K	12.6K
DP-GAN(SW)	406.8K	**3.4K**	**22.3K**	**49.6K**	**17.3K**
Amazon	Token	Dist-1	Dist-2	Dist-3	Dist-S
MLE	176.1K	0.6K	2.1K	3.5K	2.6K
PG-BLEU	124.5K	0.6K	1.9K	3.5K	2.3K
SeqGAN	217.3K	0.7K	2.6K	4.6K	3.2K
DP-GAN(S)	**467.6K**	0.8K	3.6K	7.6K	7.0K
DP-GAN(W)	279.4K	1.6K	8.9K	18.4K	9.6K
DP-GAN(SW)	383.6K	**1.9K**	**11.7K**	**26.3K**	**13.6K**
Dialogue	Token	Dist-1	Dist-2	Dist-3	Dist-S
MLE	81.1K	1.4K	4.4K	6.3K	4.1K
PG-BLEU	97.9K	1.2K	3.9K	5.5K	3.3K
SeqGAN	83.4K	1.4K	4.5K	6.5K	4.5K
DP-GAN(S)	**112.2K**	1.5K	5.2K	8.5K	5.6K
DP-GAN(W)	79.4K	1.9K	7.7K	11.4K	6.0K
DP-GAN(SW)	97.3K	**2.1K**	**10.8K**	**19.1K**	**8.0K**

Table 1: Performance of the DP-GAN and three baselines on review generation and dialogue generation tasks. Higher is better. DP-GAN(S), DP-GAN(W), and DP-GAN(SW) represent DP-GAN with only sentence-level reward, only word-level reward, and combined reward, respectively. *Token* represents the number of generated words. Dist-1, Dist-2, Dist-3, and Dist-S are respectively the number of distinct unigrams, bigrams, trigrms, and sentences in the generated text. For example, 1.2K in Dist-1 means 1200 distinct unigrams.

4.2 Baselines

We compare the proposed DP-GAN with the following baseline models:

MLE: The generator is a sequence-to-sequence model. The generator is trained with traditional MLE.

PG-BLEU: The generator is a sequence-to-sequence model. It is trained by policy gradient with the BLEU score of the generated text as the reward (Bahdanau et al., 2017). The advantage is that this model can directly optimize the task-specific score: BLEU.

SeqGAN: Sequence GAN (Yu et al., 2017) uses a binary classifier as the discriminator. Since it is originally for unconditional generation, for a fair comparison, we expand it to the version of conditional generation. We re-implement the generator by replacing a language model with a sequence-to-sequence model.

[3] https://www.yelp.com/dataset/challenge

[4] http://opus.lingfil.uu.se/OpenSubtitles.php

Yelp	Relevance	Diversity	Fluency	All
MLE	1.49	1.73	1.78	1.89
PG-BLEU	1.47	2.59	**1.38**	2.22
SeqGAN	1.48	2.40	1.54	2.12
DP-GAN	**1.32**	**1.23**	1.66	**1.51**
Amazon	Relevance	Diversity	Fluency	All
MLE	1.52	1.81	1.72	1.93
PG-BLEU	1.62	2.48	1.63	2.24
SeqGAN	1.56	2.37	**1.40**	1.97
DP-GAN	**1.31**	**1.25**	1.52	**1.50**
Dialogue	Relevance	Diversity	Fluency	All
MLE	1.19	1.84	1.37	1.87
PG-BLEU	**1.13**	1.85	1.21	1.75
SeqGAN	**1.13**	1.71	**1.20**	1.64
DP-GAN	**1.13**	**1.50**	1.30	**1.55**

Table 2: Results of human evaluation on the three datasets. The score represents the averaged ranking of each model and lower is better. *All* represents the ranking given by annotators based on a comprehensive consideration. It can be seen that DP-GAN results in the largest improvement in terms of diversity and relevance while slightly reducing fluency.

4.3 Training Details

For review generation, we set the number of generated sentences to 6 with the maximum length of 40 words for each generated sentence. Based on the performance on the validation set, we set the hidden size to 256, embedding size to 128, vocabulary size to 50K, and batch size to 64 for the proposed model and the baselines. We use the Adagrad (Duchi et al., 2011) optimizer with the initial learning rate 0.1. In adversarial training, the step for training the generator is 1K, the step for training the discriminator is 5K. Both the generator and the discriminator are pre-trained for 10 epochs before adversarial learning. In particular, for PG-BLEU and SeqGAN, before reinforcement learning or adversarial learning, we pre-train the sequence-to-sequence model for 10 epochs like DP-GAN. For dialogue generation, the settings are the same with review generation, except that we set the number of generated sentences to 1 with the maximum length of 40 words because there is only one sentence in the response.

4.4 Experimental Results

We conduct two kinds of evaluations in this work, automatic evaluation and human evaluation. The details of evaluation results are shown as follows.

4.4.1 Automatic Evaluation

We evaluate the proposed model in terms of several metrics that can reflect the diversity. The results are shown in Table 1. *Token* represents the total number of generated words. Dist-1, Dist-2, Dist-3, and Dist-S are respectively the number of distinct unigrams, bigrams, trigrms, and sentences. DP-GAN(S), DP-GAN(W), and DP-GAN(SW) represent DP-GAN with only sentence-level reward, only word-level reward, and combined reward, respectively. From the results, it is obvious that the proposed model substantially outperforms the existing models. PG-BLEU achieves slightly weaker results compared with MLE. The reason is that PG-BLEU uses BLEU score as the reward for reinforcement learning. However, the BLEU score is low for most of the generated text. The low reward makes it hard to learn from the real data. SeqGAN does not achieve better results, which suggests that the classifier-based discriminator fails to encourage the generator to produce diverse text.

In terms of the total number of generated words, DP-GAN(S) achieves better results than DP-GAN(W). Since the sentence-level reward reflects the novelty of the whole sentence, it gives repeated and short text low reward while novel and longer text high reward. Thus, the generator is encouraged to generate novel text. In terms of the number of distinct n-grams, DP-GAN(W) achieves better results than DP-GAN(S). It is because the word-level reward gives each word more precise score and novel n-grams could be better encouraged. As we can see, DP-GAN(SW), which combines the advantages of sentence-level and word-level rewards, generates not only more diverse n-grams than DP-GAN(S) but also longer text than DP-GAN(W). Since combining the word-level and sentence-level rewards achieves better results than using just one of them, we focus more on the combined reward in the following parts.

In review generation and dialogue generation tasks, it is a widely debated question how well the BLEU score against a single reference can reflect the quality of the generated text (Liu et al., 2016). Thus, although the proposed model achieves better BLEU scores compared with baselines, we omit the detailed comparisons in terms of BLEU for space.

4.4.2 Human Evaluation

We conduct a human evaluation on the test set. For all tasks, we randomly extract 200 samples from the test sets. Each item contains the input text and the text generated by the different systems. The items are distributed to three anno-

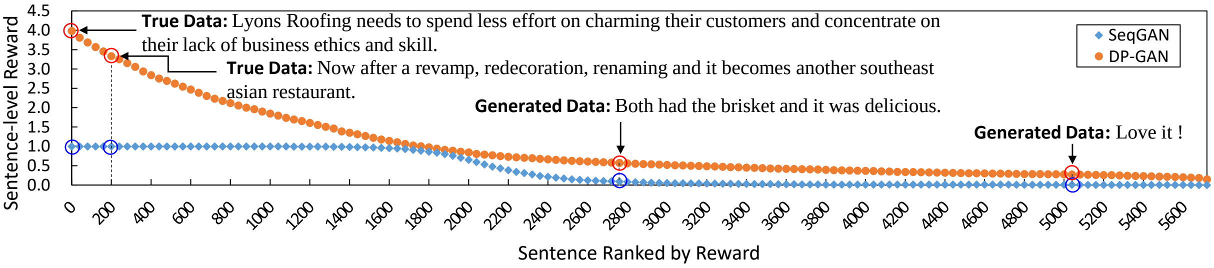

Figure 2: Distribution of rewards between SeqGAN and DP-GAN. The upper two sentences are sampled from the real-world data and the lower two sentences are sampled from the generated data. It is important to note that the sentence-level reward of DP-GAN is averaged word-level reward and a long sentence does not indicate a high score. As we can see, the reward distribution of SeqGAN saturates and cannot distinguish the novelty of the text accurately. In contrast, DP-GAN has a strong ability of resisting reward saturation and can give more precise reward for text in terms of novelty.

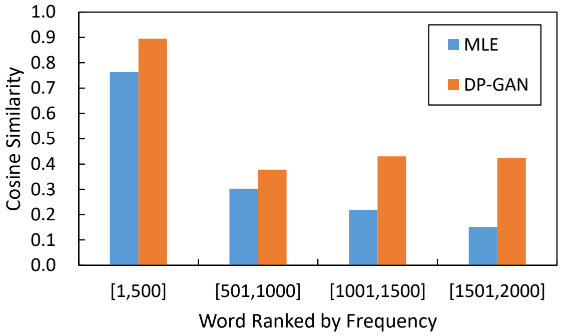

Figure 3: Cosine similarity between the real-world data distribution and the generated data distributions of various models. For example, the first column represents the cosine similarity on top 500 words with the highest frequencies in real-world data. As we can see, the generated data distribution of DP-GAN is closer to the real-world data distribution, especially considering the words of low frequency.

tators who have no knowledge about which system the text is from. Following the work of Li et al. (2017), we require them to rank the generated text considering relevance, diversity, and fluency. It is important to note that all the annotators have linguistic background. Relevance means that how likely the generated text is related to the input text. Diversity means that how much the generated text provides specific information, rather than "dull" and repeated information. Fluency means that how likely the generated text is produced by human. *All* represents the ranking given by annotators based on a comprehensive consideration of all human evaluation metrics. The results of human evaluation are shown in Table 2. It needs to be mentioned that in the special case that several pieces of generated text are exactly the same, they are given the same ranking. The inter-annotator agreement is satisfactory considering the difficulty in the human evaluation. The Pearson's correlation coefficient is 0.76 on diversity, 0.59 on fluency and 0.27 on relevance, with $p < 0.0001$. The p-value is all below 0.001, indicating high correlation and agreement. Table 2 shows that DP-GAN brings the largest improvement in terms of diversity and relevance while scarcely reducing fluency. First, DP-GAN significantly outperforms baselines in term of diversity. Second, DP-GAN achieves the best performance in terms of relevance. Diverse text generated by DP-GAN brings more specific expressions, increasing the performance of relevance. For example, given input sentence*"food is good"*, the relevance of *"love it, beef is the best"* is obviously better than that of *"love it"*. Third, the fluency of DP-GAN is slightly worse than that of baselines. It is mainly due to the fact that the text generated by DP-GAN is much longer. From Table 1, we can see that the number of words generated by DP-GAN is almost three times than that of baselines in the Yelp dataset.

4.4.3 Analysis: Why It Works

In this section, we provide detailed analysis to see why our proposed method works better.

In Figure 2, we demonstrate the reward distributions of our model and SeqGAN. It can be seen that the reward of SeqGAN cannot reflect the novelty of text accurately. First, when the novelty of text is relatively high, the reward given by SeqGAN saturates and cannot tell the difference between high-novelty text. Even with different degrees of novelty, the two high-novelty sentences sampled from the true data in the figure get nearly

Input: *One of my favorite places to eat.*
MLE: *Service is great.*
PG-BLEU: *Service is always good.*
SeqGAN: *Love the chicken and waffles. Service is always great.*
DP-GAN: *Love the fact that they have a large selection of food. Service is always great and the food is always fresh. I've been to this place a few times and have never been disappointed.*

Input: *Excellent service, clean, nice ambience.*
MLE: *Food is very good.*
PG-BLEU: *Food is very good.*
SeqGAN: *Food is very good.*
DP-GAN: *I had the chicken and beef and it was delicious. Food is good and the service is great. I would recommend this place to anyone who wants to eat here.*

Input: *I brought friends – seven of them – who'd never been to a Bar Louie before.*
MLE: *Food was mediocre, but the service was great. I will not be returning.*
PG-BLEU: *Food was terrible. Wouldn't recommend this place to anyone.*
SeqGAN: *Food was just okay. I wouldn't recommend this place.*
DP-GAN: *Ordered a chicken salad with a side of fries, and it was just ok. Food was good, but the service was slow. I had to ask for a refill on my drink, and the bartender was rude. I wouldn't recommend this place to anyone.*

Input: *I don't think it's too spicy, so i add a little hot curry spice to the sauce.*
MLE: *It's great for cooking. I have to say, I'm not sure if it is the same.*
PG-BLEU: *Love it!*
SeqGAN: *Love it! Love this stuff. I have used it for years and it is very good.*
DP-GAN:*I've tried many different brands of hot sauces but this one is the best. I've also used it in soups and stews. I also like that it is organic and has a great flavor.*

Table 3: Examples generated by DP-GAN and the state-of-the-art models on the Yelp dataset. It can be found that the text generated by baselines is more generic and repeated, while our model generates text with more specific details and higher diversity.

the same reward in SeqGAN. Second, most generated text receives reward around zero because of the high accuracy of classifier. It is hard for such reward to distinguish the difference between low-novelty text. For example, as shown in the figure, "*Both had the brisket and it was delicious*" is much more informative than "*Love it!*". The discriminator of SeqGAN gives them practically the same reward, while the proposed discriminator can better distinguish the two sentences in terms of novelty. In fact, the classifier in SeqGAN trained for 10 epochs can reach very high accuracy, that is, 98.35% and 99.63% for Yelp and Amazon, respectively. If the accuracy of classifier is too high, the classifier cannot give reasonable reward to the generator for generating real and diverse text (Arjovsky et al., 2017).

In contrast, the language-model based reward given by DP-GAN better reflect the novelty of the text. The novel text is given high reward that does not saturate. The generated data, which can be less novel, is given relatively low but nonzero reward that can encourage the generator to generate diverse expressions. The refined reward leads to more efficient training, thus resulting in better performance.

We also compare the cosine similarity between the real-world data distribution and the generated data distributions of various models. Figure 3 shows the results. We calculate the cosine distance between two vectors, where each element is the frequency of a word indexed by its rank in real-world data. For example, the first element in the vector means the frequency of the word that ranks first in real-world data. The word frequency vector is divided into 4 vectors to show the similarity of words of different frequencies. The distribution of words are more similar when they occur more frequently in real-world data. As DP-GAN promotes diversity, words of low frequency in real-world data are better learned and the similarity is much better than that of MLE. In all, the generated data distribution of DP-GAN is closer to the real-world data distribution in all intervals, especially considering the words of low frequency.

Table 3 presents the examples generated by different models on the Yelp dataset. It can be found that the text generated by MLE is more generic and repeated, while PG-BLEU and SeqGAN do not perform obviously better than MLE. Moreover, it can be clearly seen that our model generates text with more specific details and higher diversity.

5 Conclusions

In this paper, we propose a new model, called DP-GAN, to promote the diversity of the generated text. DP-GAN assigns low reward for repeated text and high reward for novel and fluent text, encouraging the generator to produce novel and diverse text. We evaluate DP-GAN on two tasks and the findings are concluded as follows: First, the proposed method substantially outperforms the baseline methods in automatic and human evaluations. It shows that DP-GAN is capable of producing more diverse and informative text. Second, the proposed discriminator can better distinguish novel text from repeated text with the saturation problem compared without traditional classifier-based discriminators. Third, with the improvement of diversity, the generated data distribution of DP-GAN is closer to the real-world data distribution compared with that of MLE.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No. 61673028). We thank Wei Li and Bingzhen Wei for providing the thoughtful suggestions. Xu Sun is the corresponding author of this paper.

References

Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *ICML 2017*, pages 214–223.

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 *ICLR 2017*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR 2014*.

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

David Berthelot, Tom Schumm, and Luke Metz. 2017. BEGAN: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS 2016*, pages 2172–2180.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014*, pages 1724–1734.

Emily L. Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS 2015*, pages 1486–1494.

John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

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 *NIPS 2014*, pages 2672–2680.

Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved training of wasserstein gans. In *NIPS 2017*, pages 5769–5779.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2017. Generating sentences by editing prototypes. *CoRR*, abs/1709.08878.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL 2016*, pages 110–119.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *ACL 2015*, pages 1106–1115.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *EMNLP 2017*, pages 2157–2169.

Junyang Lin, Xu Sun, Shuming Ma, and Qi Su. 2018. Global encoding for abstractive summarization. *CoRR*, abs/1805.03989.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP 2016*, pages 2122–2132.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2017. Table-to-text generation by structure-aware seq2seq learning. *CoRR*, abs/1711.09724.

Liangchen Luo, Jingjing Xu, Junyang Lin, Qi Zeng, and Xu Sun. 2018. An auto-encoder matching model for learning utterance-level semantic dependency in dialogue generation. In *EMNLP, 2018*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP 2015*, pages 1412–1421.

Shuming Ma, Xu Sun, Junyang Lin, and Houfeng Wang. 2018a. Autoencoder as assistant supervisor: Improving text representation for chinese social media text summarization. *CoRR*, abs/1805.04869.

Shuming Ma, Xu Sun, Yizhong Wang, and Junyang Lin. 2018b. Bag-of-words as target for neural machine translation. *CoRR*, abs/1805.04871.

Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *WWW 2013*, pages 897–908.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS 2016*, pages 271–279.

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434.

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

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *NIPS 2016*, pages 2226–2234.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *EMNLP 2017*, pages 2210–2219.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS 2014*, pages 3104–3112.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS 1999*, pages 1057–1063.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Jingjing Xu, Xu Sun, Qi Zeng, Xuancheng Ren, Xiaodong Zhang, Houfeng Wang, and Wenjie Li. 2018a. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *ACL, 2018*.

Jingjing Xu, Yi Zhang, Qi Zeng, Xuancheng Ren, Xiaoyan Cai, and Xu Sun. 2018b. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *EMNLP, 2018*.

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

Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun. 2017. Energy-based generative adversarial network. In *ICLR 2017*.