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

- Task: Text Generation.
- Problem: Existing text generation methods tend to produce repeated and "boring" expressions.

Proposed:
- We propose a new model, called Diversity-Promoting Generative Adversarial Network (DP-GAN) for diversified text generation.
- The proposed model assigns low reward for repeatedly generated text and high reward for "novel" and fluent text.
- Moreover, we propose a novel language model-based discriminator without the saturation problem.

Approach

Our model contains a generator and a discriminator. The sketch is shown in Figure 1.

- The generator $G_θ$ is responsible for generating text, which is based on a sequence-to-sequence structure.
- The discriminator $D_φ$ is a language model.
- We define cross entropy as the reward to train the generator.

Rewards

Our reward function consists of two parts:

- **Sentence-Level Reward**
  
  For a sentence $y_t$ of $K$ words, the reward at the sentence level is the averaged reward of each word:
  
  $R(y_t) = -\frac{1}{K} \sum_{k=1}^{K} \log D_φ(y_t, y_{-k})$  \hspace{1cm} (1)

- **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:
  
  $R(y_t, y_{-k}) = -\log D_φ(y_t, y_{-k})$  \hspace{1cm} (2)

Policy Gradient Training

Adversarial reinforcement training:

1. Initialize $G_θ, D_φ$ with random weights $\theta, \phi$.
2. Pre-train $G_θ$ using MLE on a dataset $\mathcal{D} = \{(X, Y)\}$.
3. Generate samples using $G_θ$ for training $D_φ$.
4. Pre-train $D_φ$.
5. $N =$ number of training iterations
6. $M =$ number of training generator
7. $K =$ number of training discriminator
8. for each $i = 1, 2, ..., N$ do
9.  for each $j = 1, 2, ..., M$ do
10.  Generate a sequence $Y_{j,i}$ using $G_θ$
11.  Update generator via policy gradient
12.  Sample a sequence $Y_{j,i}$ using $D_φ$
13.  Update generator parameters
14.  end for
15.  end for
16. Generate samples using $G_θ$
17. Train discriminator $D_φ$
18. end for
19. end for

Experiment Dataset

Task:
- Review generation: Yelp & Amazon.
- Dialogue generation: OpenSub.

Results

Automatic evaluation results:

<table>
<thead>
<tr>
<th>Method</th>
<th>Token</th>
<th>Dist-1</th>
<th>Dist-2</th>
<th>Dist-3</th>
<th>Dist-S</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>1.56</td>
<td>1.94</td>
<td>1.79</td>
<td>1.98</td>
<td></td>
</tr>
<tr>
<td>PG-BLEU</td>
<td>1.14</td>
<td>1.14</td>
<td>1.14</td>
<td>1.14</td>
<td>1.14</td>
</tr>
<tr>
<td>SeqGAN</td>
<td>3.83</td>
<td>3.83</td>
<td>3.83</td>
<td>3.83</td>
<td>3.83</td>
</tr>
<tr>
<td>DP-GAN</td>
<td>6.06K</td>
<td>2.45K</td>
<td>2.22K</td>
<td>1.99K</td>
<td>1.73K</td>
</tr>
</tbody>
</table>

Dialogue: Reference Diversity Fluency All

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
<th>Diversity</th>
<th>Fluency</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>1.76</td>
<td>1.76</td>
<td>1.76</td>
<td>1.76</td>
</tr>
<tr>
<td>PG-BLEU</td>
<td>1.22</td>
<td>1.22</td>
<td>1.22</td>
<td>1.22</td>
</tr>
<tr>
<td>SeqGAN</td>
<td>1.32</td>
<td>1.32</td>
<td>1.32</td>
<td>1.32</td>
</tr>
<tr>
<td>DP-GAN</td>
<td>1.32</td>
<td>1.32</td>
<td>1.32</td>
<td>1.32</td>
</tr>
</tbody>
</table>

Human evaluation results:

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
<th>Diversity</th>
<th>Fluency</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>1.19</td>
<td>1.19</td>
<td>1.19</td>
<td>1.19</td>
</tr>
<tr>
<td>PG-BLEU</td>
<td>1.05</td>
<td>1.05</td>
<td>1.05</td>
<td>1.05</td>
</tr>
<tr>
<td>SeqGAN</td>
<td>1.13</td>
<td>1.13</td>
<td>1.13</td>
<td>1.13</td>
</tr>
<tr>
<td>DP-GAN</td>
<td>1.32</td>
<td>1.32</td>
<td>1.32</td>
<td>1.32</td>
</tr>
</tbody>
</table>

Case Study

Examples generated by our method and the state-of-art models.

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: 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.

Analysis

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.

Contribution

- 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.