CatGAN: Category-Aware Generative Adversarial Networks with Hierarchical Evolutionary Learning for Category Text Generation

Zhiyue Liu, Jiahai Wang,* Zhiwei Liang
School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
{liuzhy93, liangzhw25}@mail2.sysu.edu.cn, wangjiah@mail.sysu.edu.cn

Abstract

Generating multiple categories of texts is a challenging task and draws more and more attention. Since generative adversarial nets (GANs) have shown competitive results on general text generation, they are extended for category text generation in some previous works. However, the complicated model structures and learning strategies limit their performance and exacerbate the training instability. This paper proposes a category-aware GAN (CatGAN) which consists of an efficient category-aware model for category text generation and a hierarchical evolutionary learning algorithm for training our model. The category-aware model directly measures the gap between real samples and generated samples on each category, then reducing this gap will guide the model to generate high-quality category samples. The Gumbel-Softmax relaxation further frees our model from complicated learning strategies for updating CatGAN on discrete data. Moreover, only focusing on the sample quality normally leads the mode collapse problem, thus a hierarchical evolutionary learning algorithm is introduced to stabilize the training procedure and obtain the trade-off between quality and diversity while training CatGAN. Experimental results demonstrate that CatGAN outperforms most of the existing state-of-the-art methods.

Introduction

Nowadays, category text generation has received more and more attention. Generating coherent and meaningful text with different categories will bring great benefits to many natural language processing applications, such as sentiment analysis (Li et al. 2018) and dialogue generation (Li et al. 2017). Recently, generative adversarial net (GAN) (Goodfellow et al. 2014), which adopts the discriminator to guide the generator, is combined with the reinforcement learning (RL) algorithms (Williams 1992) to generate discrete text data for general text generation, and some competitive results have been reported in the previous works (Yu et al. 2017; Guo et al. 2018; Caccia et al. 2018). Compared with general text generation which only focuses on obtaining high-quality text, category text generation aims at automatically generating a variety of controllable category text to fit the task-specific applications. However, the category information of sentences can not be easily controlled, and it is also difficult to design an appropriate training objective for different categories. Thus, category text generation is a more challenging task. There are a few works (Wang and Wan 2018; Li et al. 2018) which try to extend the general text generation models for category text generation. They mostly employ a long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) as the generator and combine the auxiliary components (e.g., classifiers) with the RL algorithms on GANs to generate category text. The auxiliary components can help the model to focus on the category information.

The existing category text generation models have shown some positive results, but RL algorithms and auxiliary components complicate the learning strategy and the model, respectively, which may exacerbate some fundamental problems of GANs, including training instability and mode collapse. Firstly, most of the existing models (Wang and Wan 2018; Li et al. 2018) heavily rely on RL algorithms, and some strategies, such as Monte Carlo search, are adopted to guide the discriminator for providing reward signals. These complicated strategies further increase the training difficulty of GANs. The auxiliary components may carry more burden to the adversarial training, which also makes the training procedure more unstable. Secondly, the mode collapse problem is serious in the existing models. Because the LSTM based generator (Li et al. 2018) may lack enough expressive power, and category text generation, as a sequential decision process, also easily leads the generator to focus on some limited samples in the target distribution. For generating diversified samples, the temperature variable (Caccia et al. 2018; Guo et al. 2018) is employed to make GANs focus on either the quality or the diversity, but an improvement of diversity always leads to significant degradation of quality.

In this paper, a new category text generation framework, category-aware GAN (CatGAN), is proposed to deal with the above problems. CatGAN provides a category-aware model for category text generation and a hierarchical evolutionary learning algorithm for training the model and obtaining the balance between the sample quality and diversity. Firstly, a novel category-aware model is proposed, which includes the category-wise relativistic objective to estimate

*Corresponding author

the gap between the specific category generated samples and the corresponding real samples. The generator wants to make the generated samples as realistic as the real samples, while the discriminator is eager to enlarging this gap. The relativistic relation can guide our model to update more easily than strict ground-truth labels. A relational memory core (RMC) (Santoro et al. 2018) based generator, which promises a larger memory capacity and a better ability for catching the long-term dependencies, is adopted to replace LSTM. Further, instead of RL algorithms, CatGAN employs the Gumbel-Softmax relaxation (Jang, Gu, and Poole 2017; Maddison, Mnih, and Teh 2017) to generate the continuous approximation of the discrete generated samples. The continuous data allow the generator and the discriminator to be optimized directly during the adversarial procedure. Without any auxiliary components, the architecture of CatGAN is as concise as the classical GAN framework and only consists of one generator and one discriminator. Secondly, a hierarchical evolutionary learning algorithm is developed to train the category-aware model. The adversarial training can be seen as an evolutionary problem, and the discriminator provides the environment for a population of generators to evolve. For adapting the category text generation task, the evolution procedure is designed with two stages. In the first temperature-oriented stage, the temperature is subtly controlled to maintain the category text quality during the improvement of diversity. In the second objective-oriented stage, various training objectives are adopted to narrow the distances between the generated data and the real data from different perspectives on each category. Only the well-performing generator is preserved, and the generated samples will retain diversified and high-quality. Finally, although the evaluation metric of quality has been designed well (Guo et al. 2018), the evaluation metric of diversity is not explored well. This paper proposes a new evaluation metric of diversity based on the repeatability of the generated samples.

In summary, our contributions are as follows:

- A category-aware model is proposed for generating category text, which accurately takes the gap between real samples and generated samples on each category as an efficient learning signal.
- A hierarchical evolutionary learning algorithm is designed to train the category-aware model, and it specializes in text generation for making the generated samples more diversified and high-quality.
- An effective metric is presented to evaluate the sample diversity. Experimental results on synthetic and real data demonstrate that our model achieves a new state-of-the-art performance on both category text generation and general text generation.

Related Work

Traditional recurrent neural network (RNN) based text generation models (Graves 2013) always suffer from the exposure bias problem (Huszár 2015; Bengio et al. 2015). Different from these RNN based models which are trained by maximum likelihood estimation (MLE), GAN introduces a minimax game between the generator and the discriminator. However, GAN is designed to output differentiable data, which has a conflict with the discrete text generation.

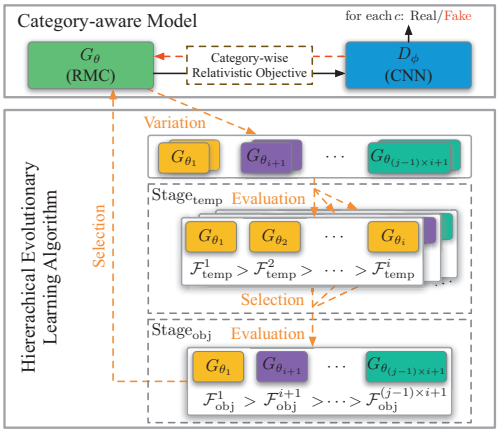
The same RL algorithm is adopted in SeqGAN (Yu et al. 2017) and LeakGAN (Guo et al. 2018) to solve the above problem, and the discriminator can guide the generator by the reward signal. However, LeakGAN shows that the reward signal is not sufficiently informative. MaskGAN (Fedus, Goodfellow, and Dai 2018) adopts the actor-critic algorithm for filling in missing text conditioned on the surrounding context. RankGAN (Lin et al. 2017) replaces the original binary classifier with a ranking model as the discriminator. Approximating methods are another way to handle the non-differentiable problem of discrete data. TextGAN (Zhang et al. 2017) and FM-GAN (Chen et al. 2018) apply an annealed softmax to approximate the argmax operation. Gu et al. (2018) and RelGAN (Nie, Narodytska, and Patel 2019) adopt the Gumbel-Softmax relaxation to approximate the categorical distribution, and this relaxation method helps to train GANs and improve the generation quality.

The above methods focus on general text generation, and category text generation is drawing more attention. CSGAN (Li et al. 2018) proposes a descriptor which consists of a discriminator and an auxiliary classifier, where the classifier distinguishes the sentence category to guide the generator. The adversarial procedure of CSGAN is similar to SeqGAN, and the less-informative reward signal limits the model performance. SentiGAN (Wang and Wan 2018) contains multiple generators, and each generator aims at generating the samples of a specific sentiment label. However, as the category number grows up, the multiple generators will significantly raise the number of trainable parameters, which may reduce the efficiency and amplify the training instability. Experiments will show that the proposed CatGAN is more effective than the previous methods using auxiliary components.

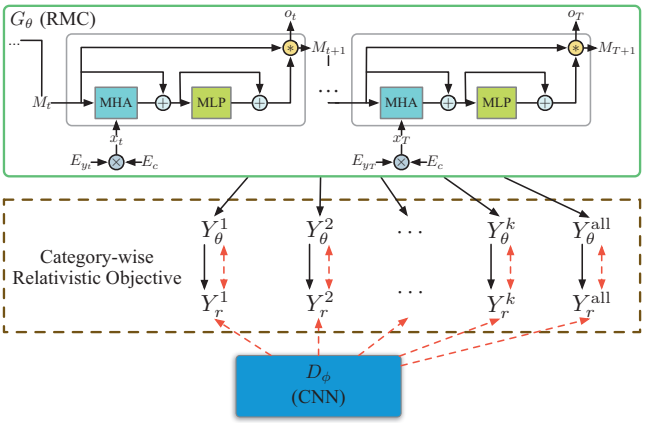
Recently, the evolutionary learning algorithm is firstly introduced to optimize the adversarial model for image generation (Wang et al. 2019). For generating better category text, both the quality and diversity should be focused. CatGAN makes the first attempt to solve category text generation with the evolutionary learning algorithm. Our hierarchical evolutionary learning algorithm is designed with two stages, the temperature-oriented stage and the objective-oriented stage, to explore the possible solutions of the generator for improving the performance on the sample quality and diversity.

Methodology

The category text generation task is denoted as follows. Given a dataset with k categories, supposing we want to generate a sentence with the specific category c, then a θ-parameterized generator G_θ is trained to generate the sentence $Y_\theta^c = (y_1, \ldots, y_t, \ldots, y_T)$, $y_t \in \mathcal{V}$, where $\mathcal{V}$ is the vocabulary of candidate tokens. In order to guide the generator G_θ effectively, a ϕ-parameterized discriminator D_ϕ also need to be trained to provide a learning signal $D(Y_\theta^c)$ for G_θ to update when the whole sentence Y_θ^c has been generated.



(a) CatGAN



(b) Category-aware Model

Figure 1: (a) The overall framework of CatGAN. A population of generators $\{G_\theta\}$ evolves in a dynamic environment denoted by the discriminator D_ϕ. In each round of evolution, the individuals $\{G_{\theta_1}, G_{\theta_2}, \cdots\}$ after the variation process undergo two stages of evaluation and selection to hierarchically form a new population, where $i = |\mathbb{T}|$ and $j = |\mathbb{O}|$. The individuals, mutated according to the same training objective under different temperatures, are denoted with the same color. (b) Category-aware model. The red dotted line represents the training process of the discriminator, while the black full line represents the training process of the generator. The MHA denotes the multi-head dot product attention.

Overall Framework

The overall framework of CatGAN is shown in Fig. 1 (a). CatGAN consists of two core parts, the category-aware model and the hierarchical evolutionary learning algorithm.

With the help of the category-wise relativistic objective, the proposed category-aware model employs a RMC based generator to generate texts with a specific category c to fool the discriminator, while the discriminator is trained to discriminate between the real samples and the generated samples for each category. In our model, the Gumbel-Softmax relaxation enables the gradients to pass back to the generator from the discriminator directly. For training the model and boosting the performance, this paper proposes the hierarchical evolutionary learning algorithm, which evolves a population of generators $\{G_\theta\}$ via combining various mutation strategies in a given environment D_ϕ. At the end of the evolutionary learning algorithm, the best-performing generator is preserved to generate the realistic sentences with the given category.

Category-aware Model

The category-aware model is shown in Fig. 1 (b), which is guided by a novel category-wise relativistic objective to generate category samples. It includes a generator G_θ and a basic CNN based discriminator D_ϕ (Kim 2014).

Category-wise Relativistic Objective. In the standard GAN (Goodfellow et al. 2014), the discriminator is trained on the ground-truth labels to predict the probability that the input data are real. By this training method, the discriminator cannot provide an informative signal to update the generator (Jolicoeur-Martineau 2018). Thus, this paper proposes a novel training objective based on the relativistic relation between real data and generated data on each category.

Formally, Y_r^c and Y_θ^c denote the real data sampled from the real data distribution P_r^c and the generated data sampled from the generated data distribution P_θ^c on the category c, respectively. The category-wise relativistic objective contains the summed category loss and the enhanced real-fake loss. For the discriminator objective, it is defined as follows:

$$l_{D_\phi}^{\text{CatRa}} = \sum_{c=1}^{k} \mathcal{L}^{\text{Ra}}(Y_r^c, Y_\theta^c) + \mathcal{L}^{\text{Ra}}(Y_r^{\text{all}}, Y_\theta^{\text{all}}), \quad (1)$$

where Y_r^{all} and Y_θ^{all} are sampled from the real data distribution P_r^{all} and the generated data distribution P_θ^{all} on all categories, respectively. On the right-hand side, the first term measures the distance between the real data and the generated data on each category, while the second term measures the overall distance on all categories. Similar to the form of RaGAN (Jolicoeur-Martineau 2018), $\mathcal{L}^{\text{Ra}}(\cdot)$ is defined by:

$$\mathcal{L}^{\text{Ra}}(Y_r, Y_\theta) = - \mathbb{E}_{Y_r \sim P_r}\left[\log\left(\bar{D}_\phi(Y_r)\right)\right] - \mathbb{E}_{Y_\theta \sim P_\theta}\left[\log\left(1 - \bar{D}_\phi(Y_\theta)\right)\right], \quad (2)$$

where the relativistic relation is measured by:

$$\bar{D}_\phi(Y) = \begin{cases} \text{sigmoid}(D_\phi(Y) - \mathbb{E}_{Y_\theta \sim P_\theta}[D_\phi(Y_\theta)]) & \text{if } Y \text{ is real} \\ \text{sigmoid}(D_\phi(Y) - \mathbb{E}_{Y_r \sim P_r}[D_\phi(Y_r)]) & \text{otherwise.} \end{cases} \quad (3)$$

Intuitively, the relativistic relation shows the gap between the probabilities of being real on the real samples and that on the generated samples. For each category, the generator wants to reduce this gap for making generated samples as realistic as real samples, while the discriminator wants to increase the probability that real samples are more realistic than generated samples. Generally, the generator objective can be set to $l_{G_\theta}^{\text{CatRa}} = -l_{D_\phi}^{\text{CatRa}}$. Compared with the standard GAN objective, the category-wise relativistic objective can efficiently train our model for category text generation.

Relational Memory Core based Generator. Since the LSTM based generator may lack enough expressive power for text generation, relational memory core (RMC) is employed as the generator G_θ. The basic concept of RMC is to consider a fixed set of memory slots (e.g., memory matrix) and allow self-attention mechanism (Vaswani et al. 2017) to interact in these memories. The increased capacity of memory boosts the expressive power and the ability to capture the category information. Given a new vocabulary observation y_t at time t, it is represented by the embedded token E_{y_t}, and the embedded category E_c is built to control the category information. Then, the input vector x_t of the generator is obtained by a linear transformation W_x on the concatenation of E_{y_t} and E_c:

$$x_t = [E_{y_t}; E_c]W_x, \tag{4}$$

where $[;]$ denotes the row-wise concatenation.

Considering a memory matrix M_t, Fig. 1 (b) shows how M_{t+1} is updated from M_t by incorporating x_t at time t. As implied by the name of the multi-head dot product attention (MHA), a H-heads RMC contains H groups of linear transformation weights for query M_tW_q, key $[M_t; x_t]W_k$ and value $[M_t; x_t]W_v$. Then, $\tilde{M}_{t+1}$ can be interpreted as a proposed update to M_t as follows:

$$\tilde{M}_{t+1} = \sigma\left(\frac{M_tW_q([M_t; x_t]W_k)^T}{\sqrt{d_k}}\right)[M_t; x_t]W_v, \tag{5}$$

where $\sigma(\cdot)$ denotes the row-wise softmax function, and d_k is the column dimension of $[M_t; x_t]W_k$. Thus, the next memory M_{t+1} and the generator output o_t are obtained by:

$$M_{t+1} = \psi_1(\tilde{M}_{t+1}, M_t), o_t = \psi_2(\tilde{M}_{t+1}, M_t), \tag{6}$$

respectively, where the two parameterized functions $\psi_1(\cdot)$ and $\psi_2(\cdot)$ both represent the interactions between $\tilde{M}_{t+1}$ and M_t by leveraging residual connections, multi-layer perception (MLP) and gated operations.

Directly sampling y_{t+1} from the multinomial distribution $\sigma(o_t)$ will cause the non-differentiability problem (Yu et al. 2017), thus the Gumbel-Softmax relaxation is employed with the generator to approximate the samples. The Gumbel-Max trick (Maddison, Mnih, and Teh 2017) and the softmax function $\sigma(\cdot)$ are used to sample discrete sentences and approximate the argmax function, respectively. The Gumbel-Max trick samples the discrete token y_{t+1} at time t by:

$$y_{t+1} = \underset{1\le d\le|\mathcal{V}|}{\text{argmax}}(o_t^{(d)} + g_t^{(d)}), \tag{7}$$

where $o_t^{(d)}$ is the value of the d-th dimension of o_t, and $g_t^{(d)}$ is sampled from the Gumbel distribution, where $g_t^{(d)} = -\log(-\log U_t^{(d)})$ with $U_t^{(d)} \sim \text{Uniform}(0, 1)$. The differentiable approximation of argmax is obtained by:

$$\hat{y}_{t+1} = \sigma(\tau(o_t + g_t)), \tag{8}$$

where τ is the temperature variable. Since the softmax-like token $\hat{y}_{t+1}$ is differentiable with respect to o_t, it is used as the input of the discriminator instead of the discrete token y_{t+1}. τ can adjust bias and variance while approximating y_{t+1}. Larger τ brings lower bias but higher variance (Tucker et al. 2017), allowing the generator to obtain higher diversity but poorer quality samples (Caccia et al. 2018).

Hierarchical Evolutionary Learning Algorithm

Unlike previous text generation methods, which adopt the fixed temperature strategy and one adversarial objective to train a generator and a discriminator, our hierarchical evolutionary learning algorithm evolves a population of generators, with various temperatures and objectives, to play the adversarial game with the discriminator.

Variation. During the variation procedure, the individuals $\{G_{\theta_1}, G_{\theta_2}, \cdots\}$ are mutated from the parents $\{G_\theta\}$ via asexual reproduction based on the combination of two kinds of strategies, the temperature mutation strategy (TMS) and the objective mutation strategy (OMS). TMS is to maintain the high sample quality when the diversity improves (i.e., τ increases). To explore the possible solutions of the generator in the parameter space, OMS further stabilizes the model training process via leveraging various training objectives.

Previous works adopt the monotone increasing function $f_{\tau_{\text{tar}}}(n)$ to boost τ over training iterations, where τ_{tar} is the target temperature, and $n \in [1, N]$ denotes the current iteration of the maximum iterations N. τ obtains a subtle increment after each iteration. Although the monotone increasing τ brings diversity, it leads to quality degradation. In one iteration, the subtle change of τ only affects the tightness of the relaxation for one batch of training samples which cannot fully represent all training samples. Thus, various subtle changes have the potential to improve the sample quality. With overall increasing τ, TMS aims at finding the optimal temperature change direction according to the quality in each iteration. The comparison between the evolutionary temperature and the monotone increasing temperature is given in the supplementary material[1]. Formally, $\mathbb{T}$ denotes the TMS set that includes the temperatures with various change directions:

$$\mathbb{T} = \{f_{\tau_{\text{tar}}}(n-1), f_{\tau_{\text{tar}}}(n), f_{\tau_{\text{tar}}}(n+1)\}. \tag{9}$$

Besides, $\mathbb{O}$ denotes the OMS set which contains several relativistic training objectives for the generator as follows:

$$\mathbb{O} = \{l_{G_\theta}^{\text{CatRS}}, l_{G_\theta}^{\text{CatRa}}\}, \tag{10}$$

where $l_{G_\theta}^{\text{CatRS}}$ is another way to measure the the relativistic relation for all categories, similar to the form of $l_{G_\theta}^{\text{CatRa}}$:

$$\begin{aligned} l_{G_\theta}^{\text{CatRS}} = &-\sum_{c=1}^{k} \mathbb{E}_{\substack{Y_r^c\sim P_r^c \\ Y_\theta^c\sim P_\theta^c}}[\log(\text{sigmoid}(D_\phi(Y_\theta^c) - D_\phi(Y_r^c)))] \\ &- \mathbb{E}_{\substack{Y_r^{\text{all}}\sim P_r^{\text{all}} \\ Y_\theta^{\text{all}}\sim P_\theta^{\text{all}}}}[\log(\text{sigmoid}(D_\phi(Y_\theta^{\text{all}}) - D_\phi(Y_r^{\text{all}})))]. \end{aligned} \tag{11}$$

It is worth noting that adding more temperatures and objectives into $\mathbb{T}$ and $\mathbb{O}$ is feasible. The Cartesian product of $\mathbb{T}$ and $\mathbb{O}$ constitutes all mutation directions $\mathbb{M} = \mathbb{T} \times \mathbb{O}$ on each round of evolution. Each individual is mutated by one mutation direction. That is, the generator is updated by a specific training objective under a certain temperature.

[1] https://arxiv.org/abs/1911.06641

Hierarchical Evaluation. Since the goals of TMS and OMS are different, two stages, including the temperature-oriented stage $\text{Stage}_{\text{temp}}$ and the objective-oriented stage $\text{Stage}_{\text{obj}}$, are designed. Both the evaluation procedure and the selection procedure are divided into the above two stages, where $\text{Stage}_{\text{temp}}$ can preliminarily filter the individuals for $\text{Stage}_{\text{obj}}$. In $\text{Stage}_{\text{temp}}$, the individuals with different temperatures in $\mathbb{T}$ can only be compared under the same objective. For each objective in $\mathbb{O}$, the individual with the optimal temperature is preserved for the further selection. Then, $\text{Stage}_{\text{obj}}$ selects the best individual considering overall performance. Thus, the proposed learning algorithm including two stages, $\text{Stage}_{\text{temp}}$ and $\text{Stage}_{\text{obj}}$, is considered as the hierarchical evolutionary learning algorithm. Two properties, the sample diversity and quality, are mainly considered to measure the performance of each individual in the whole hierarchical evolutionary learning algorithm.

For evaluating the diversity, a new metric named NLL_{div} is proposed. NLL_{div} calculates the negative log-likelihood of generated samples on the generator by:

$$\text{NLL}_{\text{div}} = -\mathbb{E}_{Y_\theta \sim P_\theta}[\log P_\theta(y_1, \cdots, y_T)], \quad (12)$$

where P_θ is the generated sample distribution. NLL_{div} can captures the repeatability of the generated samples, which will better reflect the mode collapse issue. When the generator can only learn some limited patterns from the real data or assign all its probability mass to a small region, the value of NLL_{div} will become extremely low.

For evaluating the quality, $\bar{D}_\phi(Y_\theta)$ in Eq. 3 can accurately measure the gap between the generated samples and the real samples. The higher $\bar{D}_\phi(Y_\theta)$, the better quality sentences that the generator can generate. Therefore, the evaluation scores in $\text{Stage}_{\text{temp}}$ and $\text{Stage}_{\text{obj}}$ are respectively defined as:

$$\mathcal{F}_{\text{temp}} = \mathbb{E}_{Y_\theta \sim P_\theta}[\bar{D}_\phi(Y_\theta)], \quad (13)$$

$$\mathcal{F}_{\text{obj}} = \mathbb{E}_{Y_\theta \sim P_\theta}[\bar{D}_\phi(Y_\theta)] + \lambda \text{NLL}_{\text{div}}, \quad (14)$$

where λ can be tuned to balance the quality and diversity. $\mathcal{F}_{\text{temp}}$ aims at maintaining the quality when τ increases in $\text{Stage}_{\text{temp}}$, and $\mathcal{F}_{\text{obj}}$ wants to stabilize the training process and further balance the sample quality and diversity. Then, we expect to maximize $\mathcal{F}_{\text{temp}}$ and $\mathcal{F}_{\text{obj}}$ hierarchically.

Hierarchical Selection. The evaluation procedure of each stage corresponds to a selection process, which selects the individuals with larger evaluation scores. Firstly, according to each objective in $\mathbb{O}$, the individual which has the largest $\mathcal{F}_{\text{temp}}$ is preserved with a selected direction in $\mathbb{T}$. Secondly, the surviving individuals are further filtered based on $\mathcal{F}_{\text{obj}}$ to obtain the best-performing generators as the new parents, which will participate in future adversarial training. Following the principle of "survival of the fittest", the optimal temperature and training objective are selected for the generator, allowing the whole model is trained as expected.

Experimentation

Experimental Setting

Evaluation Metrics. Some evaluation metrics have been widely used to measure the performance of text generation models from various aspects (Semeniuta, Severyn, and Gelly 2018). Generally, the negative log-likelihood $\text{NLL}_{\text{oracle}}$ (Yu et al. 2017) is used to measure the quality on synthetic data. Two evaluation metrics, NLL_{div} and NLL_{gen}, are adopted to measure the diversity, where NLL_{div} is defined by Eq. 12, and NLL_{gen} (Zhu et al. 2018) is the reversed direction of $\text{NLL}_{\text{oracle}}$. $\text{NLL}_{\text{oracle}}$ and NLL_{gen} are defined as follows:

$$\text{NLL}_{\text{oracle}} = -\mathbb{E}_{Y_\theta \sim P_\theta}[\log P_r(y_1, \cdots, y_T)], \quad (15)$$

$$\text{NLL}_{\text{gen}} = -\mathbb{E}_{Y_r \sim P_r}[\log P_\theta(r_1, \cdots, r_T)], \quad (16)$$

where P_θ is the generated data distribution and P_r is the real data distribution. $\text{NLL}_{\text{oracle}}$ is sensitive to the quality, while NLL_{div} and NLL_{gen} are sensitive to the diversity.

Since $\text{NLL}_{\text{oracle}}$ cannot evaluate the quality of real data, the BLEU scores (Zhu et al. 2018) are adopted. For category text generation, the harmonic mean values of the metrics on each category are obtained to evaluate the performance. The repeatable experiment code is made publicly available for further research[2].

Table 1: The $\text{NLL}_{\text{oracle}}$ scores on category text generation. For the $\text{NLL}_{\text{oracle}}$ scores, the lower the better.

Length	SentiGAN	CSGAN	CatGAN
20	6.976	8.431	**6.649 ± 0.097**
40	6.821	7.621	**6.498 ± 0.186**

Table 2: The performance comparison on MR. ↑ means higher is better, and ↓ means lower is better.

Method	SentiGAN	CSGAN	CatGAN
BLEU-2 ↑	0.532	0.452	**0.589 ± 0.041**
BLEU-3 ↑	0.285	0.204	**0.335 ± 0.032**
BLEU-4 ↑	0.167	0.112	**0.194 ± 0.028**
BLEU-5 ↑	0.143	0.082	**0.144 ± 0.028**
NLL_{gen} ↓	2.436	2.912	**1.619 ± 0.169**
NLL_{div} ↑	0.484	0.254	**0.535 ± 0.045**

Table 3: The performance comparison on AR.

Method	SentiGAN	CSGAN	CatGAN
BLEU-2 ↑	0.870	0.879	**0.987 ± 0.002**
BLEU-3 ↑	0.801	0.674	**0.943 ± 0.006**
BLEU-4 ↑	0.691	0.442	**0.867 ± 0.016**
BLEU-5 ↑	0.554	0.256	**0.751 ± 0.029**
NLL_{gen} ↓	3.374	3.197	**3.104 ± 0.203**
NLL_{div} ↑	0.892	1.264	**1.539 ± 0.050**

Datasets. Both synthetic and real data are employed to test CatGAN, as in previous works (Guo et al. 2018). For category text generation, synthetic data include 20,000 samples, and each 10,000 samples are obtained from different oracle-LSTM (Yu et al. 2017), and real data include movie reviews (MR) (Socher et al. 2013) and amazon reviews (AR) (McAuley et al. 2015). MR has two sentiment classes (negative and positive), and AR includes two types of product reviews (book and application). For general text generation, synthetic data include 10,000 training samples generated by an oracle-LSTM, and real data contain EMNLP2017 WMT News (EN). All real data employ the same preprocessing as in LeakGAN (Guo et al. 2018). MR has 4,503

[2]https://github.com/williamSYSU/CatGAN

Table 4: The impact of λ on AR.

Method	0	0.001	0.01	0.1	1
BLEU-2 ↑	0.982	**0.987**	0.982	0.980	0.982
BLEU-3 ↑	0.942	**0.943**	0.936	0.927	0.917
BLEU-4 ↑	**0.868**	0.867	0.841	0.825	0.803
BLEU-5 ↑	**0.757**	0.751	0.712	0.682	0.654
NLL_{gen} ↓	3.577	3.104	2.902	2.552	**2.404**
NLL_{div} ↑	1.470	1.539	1.605	1.655	**1.689**

Table 5: The ablation study on AR.

Method	CatGAN w/o H	CatGAN w/o T	CatGAN w/o O	CatGAN
BLEU-2 ↑	0.977	0.986	0.979	**0.987**
BLEU-3 ↑	0.900	0.934	0.911	**0.943**
BLEU-4 ↑	0.772	0.836	0.792	**0.867**
BLEU-5 ↑	0.613	0.703	0.638	**0.751**
NLL_{gen} ↓	3.440	3.135	3.166	**3.104**
NLL_{div} ↑	1.524	1.555	**1.618**	1.539

samples, including 3,152 samples for training and 1,351 samples for testing. For AR, each category review includes 100,000 samples for training and 10,000 samples for testing, and each sample may have multiple sentences. EN contains 200,000 training samples and 10,000 test samples.

Compared Models. Several state-of-the-art methods are set as baselines in the experiments. For category text generation, SentiGAN (Wang and Wan 2018) and CSGAN (Li et al. 2018) are compared with CatGAN ($k = 2$). For general text generation, four models are compared with CatGAN ($k = 1$), including SeqGAN (Yu et al. 2017), RankGAN (Lin et al. 2017), LeakGAN (Guo et al. 2018), and RelGAN (Nie, Narodytska, and Patel 2019). The standard MLE training is used for all models before the adversarial training. For the models which need the temperature, the exponential function $f_{\tau_{\text{tar}}}(n) = \tau_{\text{tar}}^{n/N}$ is adopted to increase the temperature, and τ_{tar} is set to 1 on synthetic data and 100 on real data. Adam (Kingma and Ba 2014) is employed to optimize our model. CatGAN is run with 6 random seeds on all experiments, and the final scores are presented with means and standard deviations (see the supplementary material for more detailed settings).

Category Text Generation Experiments

Synthetic Data Experiments. The synthetic data experiments are set with sequence length 20 and 40. $\text{NLL}_{\text{oracle}}$ is used to measure the sample quality, and the ground-truth scores are 5.748 and 4.015 for different sequence length, respectively. In Table 1, multiple generators help SentiGAN to obtain competitive results, and CatGAN outperforms SentiGAN by 0.327 and 0.323 on $\text{NLL}_{\text{oracle}}$, which illustrates that our model can obtain better quality on all categories.

Real Data Experiments. The real data experiments are conducted on MR and AR. After the same preprocessing, MR consists of 6,216 unique words with the maximum sentence length 15, and AR contains 6,416 unique words with the maximum sentence length 40. The results over generated samples are shown in Table 2 and Table 3. On MR, since SentiGAN is designed to generate sentiment text, it shows better results than CSGAN on the BLEU scores. On AR,

Table 6: The $\text{NLL}_{\text{oracle}}$ scores on general text generation.

Length	SeqGAN	RankGAN	LeakGAN	RelGAN	CatGAN
20	8.736	8.247	7.038	6.680	**6.377 ± 0.116**
40	10.310	9.958	7.191	6.756	**6.235 ± 0.131**

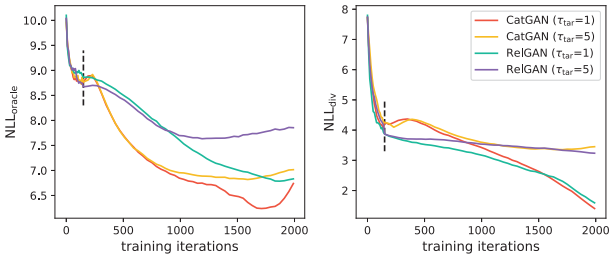


Figure 2: The training curves of CatGAN and RelGAN on the synthetic data of length 20 under different temperatures $\tau_{\text{tar}} \in \{1, 5\}$. The performances are evaluated with $\text{NLL}_{\text{oracle}}$ (left) and NLL_{div} (right). The black dotted line represents the end of pre-training.

Table 7: The performance comparison on EN.

Method	SeqGAN	RankGAN	LeakGAN	RelGAN	CatGAN
BLEU-2 ↑	0.777	0.727	0.826	0.881	**0.954 ± 0.001**
BLEU-3 ↑	0.491	0.435	0.645	0.705	**0.804 ± 0.008**
BLEU-4 ↑	0.261	0.209	0.437	0.501	**0.603 ± 0.010**
BLEU-5 ↑	0.138	0.101	0.272	0.319	**0.402 ± 0.008**
NLL_{gen} ↓	2.773	3.345	2.356	2.482	**2.316 ± 0.138**
NLL_{div} ↑	1.695	1.178	1.291	1.117	**1.716 ± 0.143**

the sufficient training samples improve the performance of all methods. CSGAN heavily relies on the auxiliary classifier and the RL algorithm, and it shows a significant quality degradation than CatGAN on BLEU while generating long sentences on AR. Compared with the baselines, CatGAN is not limited by the category type of data and obtains the better BLEU scores on both MR and AR, which also shows that CatGAN can catch the dependencies in short and long sentences. Besides, on AR, CatGAN gets 3.104 on NLL_{gen} and 1.539 on NLL_{div}, which illustrates that our model can maintain good diversity while significantly improving quality. Actually, optimizing NLL_{div} based score function can lead to a better NLL_{div}, yet CatGAN still consistently bests NLL_{div} and an existing metric, NLL_{gen}.

The impact of λ is investigated on AR. In the right-hand side of Eq. 14, the first term used to measure the quality lies in $[0, 1]$, while the second term NLL_{div} usually lies in $[0, 10]$. To balance the sample quality and diversity, λ is set to increase from 0 to 1. Table 4 shows that the increase of λ triggers the increase of diversity but the degradation of quality, especially for BLEU-4 and BLEU-5. In practice, λ is set to 0.001 for CatGAN on all experiments, since it shows a good trade-off between quality and diversity.

For illustrating the effectiveness of TMS and OMS, the ablation study is conducted on AR. The whole hierarchical evolutionary learning algorithm is removed as CatGAN w/o H, TMS is removed from CatGAN as CatGAN w/o T, and OMS is replaced with $l_{G_\theta}^{\text{CatRa}}$ to form CatGAN w/o O. The results are shown in Table 5. Compared with CatGAN,

Table 8: Samples from different methods on MR and AR.

Dataset	SentiGAN	CSGAN	CatGAN
MR	**Negative:** a tired, talky a hole in the worst movie. **Positive:** a touching, and politically potent piece of work, a film.	**Negative:** goes interesting, and the comedy were interesting. **(Wrong category)** **Positive:** it 's a treat. **(Short)**	**Negative:** the premise is intriguing but quickly becomes distasteful and creepy. **Positive:** one of the greatest family-oriented, fantasy-adventure movies ever.
AR	**Book:** i have read as the series. i could recommend it to my other walker series. **(Short)** **Application:** i love it. i love it. i would recommend a great game. **(Short)**	**Book:** this book had an hard time for what's a nice fast read. good character is great reading. what works worth the money. **Application:** i got this game from amazon it is that i have even it said weather tries. **(Unreadable)**	**Book:** this is a really good book in a series. the characters are great and they are so easy to read. it is a good read, can't wait for the next book. **Application:** great game. i play it until i get to level 3. it 's a nice game for the whole family. my kids too, and it does a great job.

CatGAN w/o T shows the degradation on all BLEU scores and only increases NLL_{div} by 0.016, which means the worse quality and the similar diversity, respectively. Although CatGAN w/o O achieves competitive sample diversity over CatGAN, it shows a significant degradation on BLEU, which means OMS can effectively guide our model. CatGAN w/o H gets the worse sample quality than CatGAN w/o O, but it still outperforms SentiGAN. The ablation study illustrates that combining TMS and OMS facilitates generating diversified and high-quality samples on real data.

General Text Generation Experiments

The experiments on general text generation are further to show the contribution of the hierarchical evolutionary learning algorithm. General text generation can be considered as the special case of category text generation when $k = 1$.

Synthetic Data Experiments. The synthetic data experiments run with sequence length 20 and 40, and the ground-truth $\text{NLL}_{\text{oracle}}$ scores are 5.750 and 4.071, respectively. The results are presented in Table 6. Compared with all baselines, CatGAN outperforms the best of them by 0.303 and 0.521 on $\text{NLL}_{\text{oracle}}$ with different sequence length, respectively, which verifies the better sample quality. Specially, with sequence length 40, CatGAN significantly improves the metric by 0.956 and 3.723 over LeakGAN and RankGAN, respectively, and it illustrates that our model is more powerful to catch long-term dependencies. With the help of the hierarchical evolutionary learning algorithm, CatGAN outperforms RelGAN which also employs the temperature and the Gumbel-Softmax relaxation.

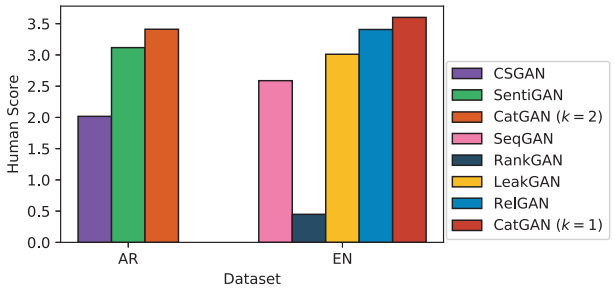


Figure 3: The performance comparison of the human score.

The trade-off between quality and diversity under different temperatures is shown in Fig. 2. It illustrates that the improvement of quality is always accompanied by the decline in diversity, and higher τ_{tar} brings more diversity. Compared with RelGAN, CatGAN shows better quality under the same diversity. With the help of TMS, CatGAN greatly improves the quality over RelGAN under the temperature $\tau_{\text{tar}} = 5$. The results validate that the hierarchical evolutionary learning algorithm can reduce the impact of mode collapse.

Real Data Experiments. The real data experiments are conducted on the EN dataset. After the preprocessing, EN contains 5,255 unique words with the maximum sentence length 51. Table 7 shows that CatGAN consistently outperforms other methods on BLEU, which illustrates its power to generate high-quality long sentences. Under the premise of achieving better BLEU scores, our model improves NLL_{gen} and NLL_{div} by 0.04 and 0.021 than the best performance of the baselines, which shows CatGAN can maintain the higher sample quality and diversity simultaneously.

Human Evaluation

The human evaluation is conducted for further evaluating the sample quality of generated sentences on AR and EN. The sample quality is measured based on grammatical and semantic correctness, and the detailed protocol is provided in the supplementary material. For category text generation, each model randomly generates 100 samples for each category, then these samples with category information are rated by five graduate students with the score from 1 to 5, where 1 means the worst quality and 5 means the best. The harmonic mean values of the average score on each category are shown in Fig. 3. For general text generation, the average score over 100 generated sentences from each model is reported. The human evaluation results demonstrate that CatGAN can generate better quality samples than other baselines.

Case Study

With trained on MR and AR, the generated sentences from SentiGAN, CSGAN, and CatGAN are listed in Table 8. As

shown by the examples, CSGAN shows some problems, such as short length, wrong category and unreadable sentence. Especially on AR, CSGAN lacks the ability for catching the long-term dependencies and generates many unreadable sentences. SentiGAN is capable of obtaining sentiment category text, but it also cannot generate the high-quality long sentences on AR, which may due to the gap between the distributions of two products from AR is larger than the gap of various sentiments. To summarize, CatGAN produces the samples which are longer, more readable and accurate on different categories (see the supplementary material for more samples).

Conclusion

This paper proposes CatGAN for category text generation. In order to guide the category-aware model to obtain category samples accurately, the informative updating signal is provided by measuring the relativistic relation between generated samples and the corresponding real samples on each category. Besides, a hierarchical evolutionary learning algorithm is developed to train CatGAN and improve generation performance. It allows the model to preserve the well-performing offspring, where the generated category samples can retain diversified and high-quality after each training iteration. Experimental results on several datasets demonstrate that CatGAN achieves a better performance than most of the existing state-of-the-art methods on both category text generation and general text generation.

Acknowledgments

This work is supported by the National Key R&D Program of China (2018AAA0101203), and the National Natural Science Foundation of China (61673403, U1611262).

References

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*, 1171–1179.

Caccia, M.; Caccia, L.; Fedus, W.; Larochelle, H.; Pineau, J.; and Charlin, L. 2018. Language gans falling short. *arXiv preprint arXiv:1811.02549*.

Chen, L.; Dai, S.; Tao, C.; Zhang, H.; Gan, Z.; Shen, D.; Zhang, Y.; Wang, G.; Zhang, R.; and Carin, L. 2018. Adversarial text generation via feature-mover's distance. In *NIPS*, 4666–4677.

Fedus, W.; Goodfellow, I.; and Dai, A. M. 2018. MaskGAN: Better text generation via filling in the _. In *ICLR*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

Graves, A. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Gu, J.; Im, D. J.; and Li, V. O. 2018. Neural machine translation with gumbel-greedy decoding. In *AAAI*.

Guo, J.; Lu, S.; Cai, H.; Zhang, W.; Yu, Y.; and Wang, J. 2018. Long text generation via adversarial training with leaked information. In *AAAI*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Huszár, F. 2015. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical reparameterization with gumbel-softmax. In *ICLR*.

Jolicoeur-Martineau, A. 2018. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, J.; Monroe, W.; Shi, T.; Jean, S.; Ritter, A.; and Jurafsky, D. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Li, Y.; Pan, Q.; Wang, S.; Yang, T.; and Cambria, E. 2018. A generative model for category text generation. *Information Sciences* 450:301–315.

Lin, K.; Li, D.; He, X.; Zhang, Z.; and Sun, M.-T. 2017. Adversarial ranking for language generation. In *NIPS*, 3155–3165.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*.

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*, 43–52.

Nie, W.; Narodytska, N.; and Patel, A. 2019. RelGAN: Relational generative adversarial networks for text generation. In *ICLR*.

Santoro, A.; Faulkner, R.; Raposo, D.; Rae, J.; Chrzanowski, M.; Weber, T.; Wierstra, D.; Vinyals, O.; Pascanu, R.; and Lillicrap, T. 2018. Relational recurrent neural networks. In *NIPS*, 7299–7310.

Semeniuta, S.; Severyn, A.; and Gelly, S. 2018. On accurate evaluation of gans for language generation. *arXiv preprint arXiv:1806.04936*.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 1631–1642.

Tucker, G.; Mnih, A.; Maddison, C. J.; Lawson, J.; and Sohl-Dickstein, J. 2017. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *NIPS*, 2627–2636.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 5998–6008.

Wang, K., and Wan, X. 2018. SentiGAN: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, 4446–4452.

Wang, C.; Xu, C.; Yao, X.; and Tao, D. 2019. Evolutionary generative adversarial networks. *IEEE Transactions on Evolutionary Computation*.

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

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *AAAI*.

Zhang, Y.; Gan, Z.; Fan, K.; Chen, Z.; Henao, R.; Shen, D.; and Carin, L. 2017. Adversarial feature matching for text generation. In *ICML*, 4006–4015.

Zhu, Y.; Lu, S.; Zheng, L.; Guo, J.; Zhang, W.; Wang, J.; and Yu, Y. 2018. Texygen: A benchmarking platform for text generation models. In *SIGIR*, 1097–1100.