TRGAN: Text to Image Generation Through Optimizing Initial Image

Liang Zhao(✉), Xinwei Li, Pingda Huang, Zhikui Chen, Yanqi Dai, and Tianyu Li

School of Software Technology, Dalian University of Technology, Dalian, China
liangzhao@dlut.edu.cn

Abstract. Generative Adversarial Networks (GANs) have shown success in text-to-image generation tasks. Most of the current methods use multi-stages to generate images, but the quality of the final images is largely dependent on the quality of the initial generated images, thus it is difficult to generate high-quality images in the end if the initial images in the first stage are of low quality, low resolution, irregular shape, strange color, and unrealistic entity relations. Therefore, in this paper, we propose to design a multi-stage generation model, and we address this problem by developing a novel generation model called Text-representation Generative Adversarial Network (TRGAN). TRGAN contains two modules: Joint attention stacked generation module (JASGM) and Text generation in the opposite direction and correction module (TGOCM). In the JASGM module, the detailed feature is extracted from word-level information and the images are generated based on the global sentence attention. In the TGOCM module, the text descriptions are generated reversely, which can improve the quality of the initial images by matching the word-level feature vector. Experimental results present that our proposed model TRGAN outperforms the compared state-of-the-art text-to-image generation methods on CUB and COCO datasets.

Keywords: Text-to-image synthesis · Generative Adversarial Network · Text-image semantic understanding · Text generation reversely

1 Introduction

Nowadays, text-to-image synthesis [1–3] is one of the important applications of GANs, which is one of the most active research areas in recent years. Most early proposed methods of text-to-image using one-step to directly generate final results. However, with the development of text-to-image synthesis methods, the more recent approaches explore multi-stages to generate images from text descriptions, such as AttnGAN [4], StackGAN [5] and MirrorGAN [6]. Some researchers [4–7] take the entire sentence encoding as the basis, and then change the corresponding attribute for each word vector [8]. However, if the initial

T. Mantoro et al. (Eds.): ICONIP 2021, CCIS 1516, pp. 651–658, 2021.
https://doi.org/10.1007/978-3-030-92307-5_76

images are not real (that is, lacks substance, loses form, and is far from the real image), the quality of the image in the next stage will not improve promisingly. Therefore, the text-to-image generation not only needs multi-stage generation, but also needs to achieve different functions in different stages to generate more realistic images.

To tackle it, in this paper a text-to-image model is proposed for synthesizing images from text descriptions by multi-stages, called Text-representation Generative Adversarial Network (TRGAN). And its main contributions are as follows: Firstly, each stage performs different generation tasks for different functions in TRGAN. Secondly, in order to improve the quality of low-quality generated images in the initial stage, a layer of processing is designed in the second stage of generation, in which the generated image is encoded into the image vector as the condition for text vector generation. After that the method utilizes a discriminator to distinguish between the ground truth text vector and generated text vector (see Fig. 1).

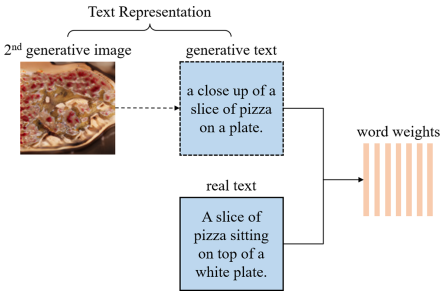

Fig. 1. A discriminator to distinguish between the ground truth text vector and generated text vector.

2 Text-Representation Generative Adversarial Network

In order to favorably generate images from the text description, we proposed the Text-representation Generative Adversarial Network (TRGAN) model. TRGAN is a complex structure with three stages. As shown in Fig. 2, the proposed TRGAN also contains two modules: JASGM and TROGM. The first two stages belong to module JASGM and the last stage belongs to module TROGM. In the JASGM module, the detailed feature information is captured from word-level information and the images are generated based on global sentence attention. In the TGOCM module, the text description is generated reversely from generated images to improve the quality of the initial images by matching the word-level feature vector. Details of the model will be introduced in the subsections.

2.1 JASGM: Joint Attention Stacked Generation Module

In this section, we mainly focus on the properties of detail and embed the given text description into local word-level features. Specifically, we need to process

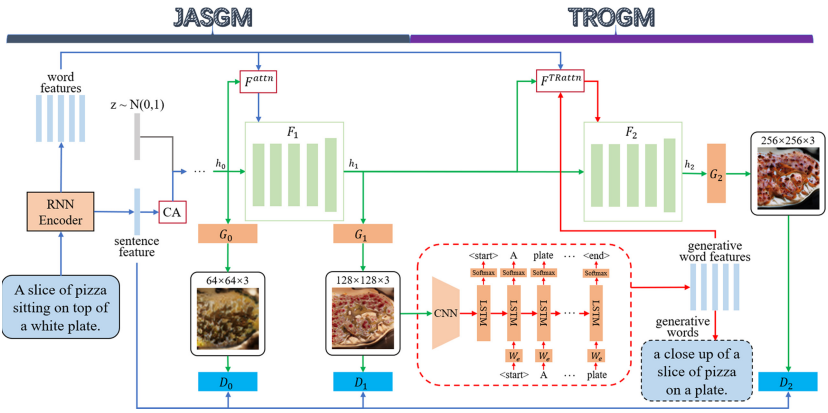

Fig. 2. The architecture of the proposed TRGAN, including JASGM and TROGM modules, which realize different functions, respectively.

word by word in the sentence, so we choose RNN, the recurrent neural network (RNN) [9], to extract word embedding (w_0, w_1, w_{t-1}) from the given text description T.

$$\begin{aligned} f_t &= g_1\left(v_{w_t} + w_{s_{t-1}}\right); \\ W_t &= g_2\left(v_{h_t}\right), \end{aligned} \tag{1}$$

where $w = \left\{w^l \mid l = 0, \ldots, L-1\right\}$, f represents the output of hidden layer.

In our module, an attention word-level feature context matrix Att^w_{i-1} is generated. After that, the word-level weight matrix Att^w_{i-1} and visual feature f_i are as inputs to the perceptron, and then the perceptual layer transforms word-level features into the common semantic space of visual features. Finally, the visual feature f_i of the next stage is further generated through the computation of word-level weight matrix Att^w_{i-1} and visual feature f_{i-1}.

As shown in Fig. 2, the proposed TRGAN has three generators (G0, G1, G2), which take the hidden states (h0, h1, h2) as input, and three discriminators (D1, D2, D3). The images (X1, X2, X3) are generated from low-resolution to high-resolution by generators. First, the feature is extracted from a global sentence vector using a random noise vector, and then the visual feature vector extracted from the perceptron is combined to generate the image of the initial stage.

$$\begin{aligned} f_0 &= F_0\left(z, F^{ca}(s)\right); \\ f_i &= F_i\left(f_{i-1}, F_{att_i}\left(f_{i-1}, w\right)\right), i \in \{1, 2\}; \\ \hat{I}_i &= G_i\left(f_i\right). \end{aligned} \tag{2}$$

Herein, z is a noise vector usually sampled from a standard normal distribution, $f_i \in \mathbb{R}^{M_i \times N_i}$, $I_i \in \mathbb{R}^{q_i \times q_i}$, and $z \sim N(0,1)$. F_{att_i} is the proposed word level attention model. Then, each word vector is computed for each region of the

image based on its hidden features h (query). Each part of the initial image is plotted according to the weight of each word for each region.

2.2 TGOCM: Text Generation in the Opposite Direction and Correction Module

The TGOCM is divided into four parts, which are generating text in the opposite direction, matching word-level attention, jointing attention mechanism and correcting image. The following is a detailed description of each part. We employ the widely used encoder-decoder architecture, which needs to be implemented using CNN [10] and RNN [11] models, respectively. The structure of the model mainly includes three parts: a) Feature Extractor, the size of the extracted image features is 2048, with dense layers, and we reduce the size to 256 nodes. b) Sequence Processor, the embedding layer handles the text input, followed by the LSTM layer [12]. c) Decoder, combining the outputs of the above two layers, we process them as dense layers to make the final prediction.

$$\begin{aligned} x_2 &= CNN\left(I_2\right); \\ x_t &= W_e T_t, t \in \{0, \ldots L-1\}; \\ p_{t+1} &= LSTM\left(x_t\right), t \in \{0, \ldots L-1\}, \end{aligned} \tag{3}$$

in which $x_2 \in \mathbb{R}^{M_{m-1}}$ is the visual feature used as the input to inform the LSTM for the image content. $W_e \in \mathbb{R}^{M_{m-1} \times D}$ represents a word embedding matrix, which maps word features to the visual feature space. p_{t+1} is a predicted probability distribution over the words.

Here, we can compare the real semantics with the generated semantics. By calculating the similarity [13] between the two semantics, and according to the similarity of the word, it gives the corresponding weight to each word.

$$\cos(\theta) = \frac{\sum_{i=1}^{n}\left(x_i \times y_i\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i\right)^2} \times \sqrt{\sum_{i=1}^{D}\left(y_i\right)^2}} \tag{4}$$

where x_i represents the actual text, y_i represents the generated text, if the cosine is closer to 1, it means that the angle between them is closer to 0°, which means that the two vectors are more similar, and the angle between them is equal to 0, which means that the two vectors are equal.

Meanwhile, each column of h is a feature vector of a sub-region of the image. For the j^{th} sub-region, its word-context vector is a dynamic representation of word vectors relevant to h_j, which is calculated by

$$c_j = \sum_{i=0}^{T-1} \beta_{j,i} e'_i, \text{ where } \beta_{j,i} = \frac{\exp\left(s'_{j,i}\right)}{\sum_{k=0}^{T-1} \exp\left(s'_{j,k}\right)}, \tag{5}$$

where $s'_{j,i} = h_j^T e'_i$, and $\beta_{j,i}$ indicates the weight that the model attends to the i^{th} word when generating the j^{th} sub-region of the image.

Each word is given the corresponding weight from the matching and word-level attention module. In this way, we can not only locate the specific region, but also focus on the word vector with great loss. Based on the above work, we multiply two matrices. This points the way for the final phase of the generation. The final stage has the function of correcting and optimizing the generated image according to the attention mechanism. Such targeted optimization generation will make the generated image quality promisingly.

2.3 Objective Function

The whole model is divided into three generation stages, so we will describe the objective function in three stages. The generator losses can be defined as:

$$L_G = \sum_{i=0}^{2} \mathcal{L}_{G_i} + \alpha L_{G1} + \beta L_{cap} + \lambda L_{ws}, \tag{6}$$

Herein, L_{G1}, L_{cap} and L_{ws} represent three stages of the loss, respectively. The discriminator works against the generator to determine whether the generated image is true, the calculation method is a conventional algorithm.

The adversarial loss for D_i [4] is defined as:

$$\begin{aligned} \mathcal{L}_{D_i} = & -\frac{1}{2}\mathbb{E}_{x_i \sim p_{\text{data}\ i}} \left[\log D_i\left(x_i\right)\right] \\ & -\frac{1}{2}\mathbb{E}_{\hat{x}_i \sim p_{G_i}} \left[\log\left(1 - D_i\left(\hat{x}_i\right)\right)\right] \\ & -\frac{1}{2}\mathbb{E}_{x_i \sim p_{\text{data}_i}} \left[\log D_i\left(x_i, \bar{e}\right)\right] \\ & -\frac{1}{2}\mathbb{E}_{\hat{x}_i \sim p_{G_i}} \left[\log\left(1 - D_i\left(\hat{x}_i, \bar{e}\right)\right)\right]. \end{aligned} \tag{7}$$

3 Experiments

In this section, we first introduce the datasets, training details, and evaluation metrics used in our experiments. In addition, we carried out extensive experiments that evaluate the proposed model, which is compared with some state-of-the-art models (i.e., StackGAN [14], StackGAN++ [5], AttnGAN [4] and MirrorGAN [6]) by some basic evaluation indicators.

3.1 Datasets

Most of the studies on text-to-image are based on CUB and complex integrated COCO datasets. Each image has 10 text descriptions in the CUB dataset and each image has 5 text descriptions in the COCO dataset.

3.2 Training Details

Firstly, we pre-train the three models of text encoding, image encoding and text reproduction. To simplify the training process, we directly load the pre-trained model and parameters into our overall model. We preprocess the COCO dataset and randomly select a quarter of the original training sets and test sets for training and testing. The training process is performed for 300 epochs on the CUB birds dataset and 300 epochs on the COCO dataset.

Fig. 3. Examples of images generated by AttnGAN, MirrorGAN and TRGAN conditioned on text descriptions from CUB and COCO test datasets and the corresponding ground truth.

3.3 Results

Quantitative Results: The TRGAN we proposed is based on a multi-stage structure generated from low resolution to high resolution. GAN-INT-CLS [15], GAWWN [16], AttnGAN, StackGAN++ and MirrorGAN proposed in previous studies are also based on a multi-stage structure generated from low resolution to high resolution. So we compared the TRGAN with the previous models (AttnGAN, StackGAN++ and MirrorGAN). As shown in Table 1, compared with MirrorGAN which employs Siamese Network to ensure text-image semantic consistency on a simple dataset CUB, our TRGAN improves the IS from 4.56 to 4.66 and the R-Precision from 60.42 to 69.05. This is because our TRGAN can generate a better initial image and optimizing it in the subsequent generation process. It proved that our model has a higher resolution on images of a single entity and multiple entities.

Qualitative Results: For qualitative evaluation, Fig. 3 shows text-to-image synthesis examples generated by our TRGAN and the state-of-the-art models. In general, our TRGAN approach generates images with more vivid details as well as more clear backgrounds in most cases, comparing to the AttnGAN, MirrorGAN and ground truth. In conclusion, the reason is that although StackGAN, AttnGAN, MirrorGAN used their stacked architecture or cross-modal spatial attention, it is not completely solved. However, our model aims at improving

Table 1. IS scores and R-Precision of the six models on the CUB dataset.

Dataset	Method	IS	R-Precision	Dataset	IS	R-Precision
CUB	GAN-INT-CLS [15]	2.88 ± 0.04	/	COCO		
CUB	GAWWN [16]	3.62 ± .07	/	COCO		
CUB	StackGAN++ [5]	4.04 ± 0.06	/	COCO	1.09 ± 0.12	/
CUB	AttnGAN [4]	4.36 ± 0.03	67.82 ± 4.43	COCO	1.69 ± 0.09	56.95 ± 0.45
CUB	MirrorGAN [6]	4.56 ± 0.05	60.42 ± 2.75	COCO	4.46 ± 0.20	60.78 ± 0.41
CUB	**TRGAN**	**4.66 ± 0.13**	**69.05 ± 2.25**	**COCO**	**4.52 ± 0.11**	**62.3 ± 0.33**

the quality of the initial image first, and targeted optimization aims at generating regions.

Ablation Study: We next conduct ablation studies on the proposed model and its variants. To validate the effectiveness of generating the text module in reverse, we conduct several comparative experiments by excluding/including these components in TRGAN. We compare the baseline model and reverse generated text module in the second and last stage, respectively. The IS score increases from 4.33 to 4.49 by adding a reverse-generated text module, then the IS score increases from 4.49 to 4.69 by adding the model on different stages (stage 2 and stage 3), as shown in Fig. 4. That's why we can change the quality of the initial image to ensure the quality of the result.

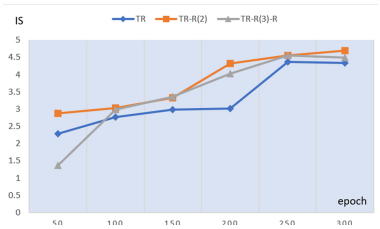

Fig. 4. The results of baseline model, adding reverse generated text model comparison.

4 Conclusions

In this paper, we have proposed a new framework called Text-representation Generative Adversarial Network (TRGAN). The whole framework consists of two modules, namely JASGM and TROGM. The first modules focus on the generation of fine-grained features. In the second module, the image of the previous stage is repaired and corrected based on the attention mechanism. Extensive experiment results show that our proposed TRGAN significantly outperforms state-of-the-art models on the CUB and COCO datasets.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (61906030), the Fundamental Research Funds for the Central Universities (DUT20RC(4)009), Natural Science Foundation of Liaoning Province (2020-BS-063), and the Equipment Advance Research Fund (80904010301).

References

1. Zhe, G., Gan, C., He, X., Pu, Y., Li, D.: Semantic compositional networks for visual captioning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
2. Nam, S., Kim, Y., Kim, S.J.: Text-adaptive generative adversarial networks: manipulating images with natural language (2018)
3. Li, B., Qi, X., Lukasiewicz, T., Torr, P.: Text-guided image manipulation, Manigan (2019)
4. Tao, X., Zhang, P., Huang, Q., Han, Z., He, X.: AttnGAN: fine-grained text to image generation with attentional generative adversarial networks (2017)
5. Han, Z., et al.: StackGAN++: realistic image synthesis with stacked generative adversarial networks. IEEE Trans. Pattern Anal. Mach. Intell. **99**, 1 (2017)
6. Qiao, T., Zhang, J., Xu, D., Tao, D.: MirrorGAN: learning text-to-image generation by redescription. IEEE (2019)
7. Banerjee, S., Das, S.: SD-GAN: structural and denoising GAN reveals facial parts under occlusion (2020)
8. Baraheem, S.S., Nguyen, T.V.: Text-to-image via mask anchor points. Pattern Recogn. Lett. **133**, 25–32 (2020)
9. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation, Computer Science (2014)
10. Wu, S., Zhong, S., Liu, Y.: Deep residual learning for image steganalysis. Multimedia Tools Appl. **77**(9), 10437–10453 (2017). https://doi.org/10.1007/s11042-017-4440-4
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling. In: Interspeech (2012)
13. Nguyen, H.V., Bai, L.: Cosine similarity metric learning for face verification. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010. LNCS, vol. 6493, pp. 709–720. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19309-5_55
14. Zhang, H., et al.: StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. IEEE (2017)
15. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. JMLR.org (2016)
16. Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)