Penalty-based Sequence Generative Adversarial Networks with Enhanced Transformer for Text Generation

Mingjun Duan, Yubai Li*
School of Information and Communication Engineering
University of Electronic Science and Technology of China, UESTC
Chengdu, China
307265102@qq.com

***Abstract*—In this paper, we propose a new model to solve the problem of text generation, which is based on the concept of seqGAN, combines self-attention with modeling localness, and introduces the penalty-based objective function. This model has much better performance than the original model. In the original model, the generator's ability of the text feature extraction is insufficient. We introduce self-attention with modeling localness, which greatly enhances its ability to capture long distance and shortrange dependencies. In addition, we use the penalty-based objective function instead of the loss function of the original model to solve the problem of mode collapse. Experimental results demonstrate that our model consistently outperforms several state-of-the-art text generation methods in the quality of generated texts.**

Keywords-generative adversarial networks; natural language processing; text generation

I. INTRODUCTION

Text generation is an essential task in natural language processing. It has a wealth of practical applications, such as image subtitles, intelligent question answering, and machine translation. The application of generative adversarial networks (GAN) [Goodfellow et al. 2014] in text generation attracts much attention. Sequence generative adversarial nets (seqGAN) [Yu et al. 2017], which introduces the reinforcement learning algorithm, is a successful attempt.

Although seqGAN overcomes the limitation of GAN on discrete data adversarial training, using LSTM as a generator still has the problem of insufficient ability to extract text features. At the same time, solving the mode collapse caused by seqGAN's loss function is also a very challenging task.

In RNNs [Gehring et al. 2017], the encoder attempts to store sentence information of any arbitrary length into a fixed-size hidden vector, which may not be fully relevant to the target task, especially when the original sequence is long or very information-rich. To solve the problem above, the attention mechanism is proposed, which can capture useful context information and provide it to the decoder. We introduce self-attention in seqGAN to further achieve better text feature extraction.

In addition, one of the major drawbacks of GAN is the problem of "mode collapse", and it has been empirically proven that GAN prefers to generate samples around only a few modes whilst ignoring other modes [Theis et al. 2016]. So, there is a lack of diversity in generated texts. We replace seqGAN's loss function with a penalty-based objective function [Wang and Wan 2019], which aims to minimize the overall penalty rather than maximize the reward. Experiments show that this more reasonable measure forces the generator to generate more diverse text.

In this paper, we propose a novel Penalty-based Enhanced Sequence Generative Adversarial Network (PB-ESGAN) architecture. We use the encoder and decoder with self-attention to replace the generator of seqGAN, so that the decoder can make more effective use of context information, so as to improve the ability of text feature extraction. In addition, we optimize the loss function so that it minimizes the penalty as a training target to improve the mode collapse problem and diversify the text generation content. Experiments are performed on two datasets, and the results demonstrate the efficacy and superiority of our proposed model.

The major contributions of this paper are summarized as follows:

- We introduce a self-attention model with modeling localness and propose a novel framework PB-ESGAN, which can effectively generate generic and high-quality text content.
- We add a new penalty-based objective to solve the problem of mode collapse in seqGAN and make the generated text more diverse.
- Extensive experiments are performed on datasets, and the results prove that our model is better than other optimal models.

The rest of our paper is structured as follows: Section II discusses some background knowledge and related works, Section III gives a detailed description of our model. Section IV analyzed experiment results, Section V summarizes this work and the future direction.

*Corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

II. Background and Related Work

The original text generation uses the standard recurrent neural network language model [Mikolov et al. 2011]. It predicts each word of a sentence conditioned on the previous word and an evolving hidden state. However, when used for text generation, RNN based models trained by the maximum likelihood approach always suffer from exposure bias [Bengio et al. 2015]. Also, sentence level performance evaluation is not applicable to word level loss functions. There are some researches which use generative adversarial network (GAN) to solve the problems.

A. Generative Adversarial Networks

Generative Adversarial Nets are a recent novel class of deep generative models that have received widespread attention in recent years. Its great success in computer vision [Denton et al. 2015; Isola et al. 2016; Salimans et al. 2016] also proves that it is a promising model.

The GAN framework establishes two distinct players, a generator G and discriminator D, and poses the two in an adversarial game. The discriminator model $D(x)$ computes the probability that a point x in data space is a sample from the data distribution (positive samples) that we are trying to model, rather than a sample from our generative model (negative samples). Concurrently, the generator uses a function $G(z)$ that maps samples z from the prior $p(z)$ to the data space. $G(z)$ is trained to maximally confuse the discriminator into believing that samples it generates come from the data distribution. The generator is trained by leveraging the gradient of $D(x)$, and using that to modify its parameters. The solution to this game can be expressed as following:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \quad (1)$$

where G and D are generative model and discriminative model respectively, $p_{data}(x)$ is the distribution of data x and p_z is the randomly assigned noise distribution.

The generator G and the discriminator D can be trained in two stages: (a) Train the discriminator to distinguish the true samples from the fake samples generated by the generator. (b) Train the generator so as to fool the discriminator with its generated samples. From a game theory point of view, the convergence of a GAN is reached when the generator and the discriminator reach a Nash equilibrium.

Although significant results have been achieved in many fields, GAN is considered to be difficult to apply on NLP, since the update of GAN based on continues space, but languages are discrete. Some works attempt to solve this problem, including Gumbel-softmax distribution [Kusner and Hern'andez-Lobato 2016], Professor Forcing [Lamb et al. 2016] and so on. However, it is more common to tackle this problem with a strategic gradient of reinforcement learning. One of the classic models is seqGAN.

The generator of seqGAN adopts the LSTM model, which limits the parallelism of the algorithm. Transformer [Vaswani et al. 2017] based on self-attention can improve this.

B. Self-Attention Model

Attention is a mechanism of attention, which can be described as normalizing both the query and key in vector, and calculating the output as a weighted synthesis of values. Recent evidence suggests that it performs well on a variety of tasks, such as machine translation and text categorization.

The Transformer model is based on self-attention. The encoder of Transformer is composed of a stack of six identical layers. Each layer consists of a multi-head self-attention and a simple position-wise fully connected feed-forward network. The decoder is also composed of a stack of six identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack.

Formally, given an input sequence $x = \{x_1, \cdots, x_I\}$, each hidden state in the l-th layer is constructed by attending to the states in the $(l-1)$-th layer. Specifically, the $(l-1)$-th layer $H^{l-1} \in R^{I\times d}$ is first transformed into the queries ($Q \in R^{I\times d}$), the keys ($K \in R^{I\times d}$), and the values ($V \in R^{I\times d}$) with three separate weight matrices.

The l-th layer is calculated as:

$$H^l = ATT(Q,K)V \quad (2)$$

where ATT($\cdot$) is a dot-product attention model, defined as:

$$ATT(Q,K) = softmax(energy) \quad (3)$$

$$energy = \frac{QK^T}{\sqrt{d}} \quad (4)$$

where $\sqrt{d}$ is the scaling factor with d being the dimensionality of layer states.

The Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers since it allows for significantly more parallelization. The output of the Transformer uses a weighted averaging operation, which fully considers all signals, but also disperses the distribution of attention, thereby overlooking the relation of neighboring signals.

In our model, we pushed GAN on NLP further based on previous works. The concept of seqGAN is used in our model to overcome the problem of parameter updating caused by discrete vectors. At the same time, through the localized modeling of the text, the Transformer is optimized to enhance the ability of capturing short-range dependencies [Yang et al. 2019]. For the task of text generation, we develop a novel PB-ESGAN. It can improve the feature extraction and text diversification, and obtain better results in two datasets.

III. THE APPROACH

In this section, we describe the architecture of the proposed penalty-based enhanced sequence generative adversarial net (referred to as PB-ESGAN) in detail.

A. *Model Overview*

Fig. 1 describes the complete architecture of the model. The whole framework can be divided into two adversarial learning objectives: generator learning and discriminator learning. The generator is based on the real dataset sentences and aims to generate sentences that are indistinguishable from the original dataset. Specifically, its purpose is to minimize the overall penalty. The discriminator D, conditioned on the real dataset sentences, tries to distinguish the machine-generated sentences from real sentences. D can be viewed as a dynamic objective since it is updated synchronously with G.

B. *Generator Learning*

The sentence generation process is viewed as a sequence of actions that are taken according to a policy regulated by the generator. The generator defines the policy and generates the target sentences based on the source sentences. In the PB-ESGAN model, we use the encoder and decoder with self-attention as a generator and introduce localness modeling as a gaussian bias.

The original self-attention model has the advantage of capturing long-distance dependencies by directly paying attention to all input elements. But this process is completed by weighted average operation, which will cause it to ignore the relationship of neighboring signals. The relationship of neighboring signals usually corresponds to the concept of phrases, which plays an important role in text generation. The ability to learn local contexts can be enhanced by modeling localness for self-attention.

The modeling localness is designed as a learnable gaussian bias G, which can be expressed as follows, where P_i represent predicted central position and D_i represent the window size:

$$G_{i,j} = -\frac{(j-P_i)^2}{2\sigma_i^2}, \quad \sigma_i = \frac{D_i}{2} \tag{5}$$

$$\begin{bmatrix} P_i \\ D_i \end{bmatrix} = I \bullet sigmoid(\begin{bmatrix} P_i \\ z_i \end{bmatrix}) \tag{6}$$

The prediction of each central position depends on its corresponding query vector:

$$P_i = U_P^T \tanh(W_P Q_i) \tag{7}$$

We choose a flexible manner called Query-Specific Window to calculate the window size:

$$z_i = U_d^T \tanh(W_P Q_i) \tag{8}$$

Finally, the gaussian bias is applied to modify the original attention distribution, so that the locally strengthened weight distribution is:

$$ATT(Q,K) = softmax(energy + G) \tag{9}$$

The enhanced self-attention model, while maintaining the strength of the original model to capture long-distance dependencies, enhances its ability to capture short-range dependencies.

C. *Discriminator Learning*

Deep discriminative models such as convolutional neural network (CNN) [Kim 2014] and recurrent convolutional neural network (RCNN) [Lai et al. 2015] have shown high performance in complicated sequence classification tasks. In this paper, we use the concept of seqGAN and choose CNN as our discriminator to classify the entire sequence rather than the unfinished one.

Given the original dataset text sequence $x_1, x_2, \cdots, x_T$, and the target generating text sequence $y_1, y_2, \cdots, y_T$, we build the source matrix and target matrix respectively as:

$$X_{1:T} = x_1; x_2; \cdots; x_T \tag{10}$$

and:

$$Y_{1:T} = y_1; y_2; \cdots; y_T \tag{11}$$

where $x_t, y_t \in R^k$ is the k-dimensional word embedding and the semicolon is the concatenation operator. For the source matrix $X_{1:T}$, a kernel $\omega_j \in R^{l \times k}$ applies a convolutional operation to a window size of l words to produce a series of feature maps:

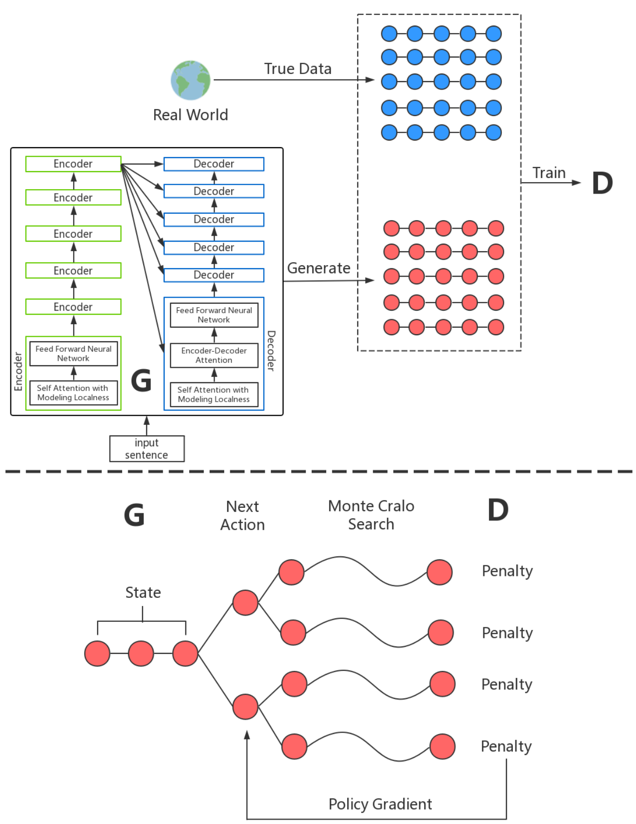

Figure 1. The whole architecture of our proposed penalty-based sequence generative adversarial networks with enhanced Transformer.

$$c_{ij} = \rho(\omega_j \otimes X_{i:i+l-1} + b) \quad (12)$$

where $\otimes$ operator is the summation of elementwise production and b is a bias term. ρ is a nonlinear activation function which is implemented as ReLu in this paper. To get the final feature with respect to kernel ω_j, a max-over-time pooling operation is leveraged over the feature maps:

$$\tilde{c}_j = \max\left\{c_{j1}, \cdots, c_{jT-l+1}\right\} \quad (13)$$

We use various numbers of kernels with different window sizes to extract different features, which are then concatenated to form the original dataset sentence representation c_x. Identically, the generated sentence representation c_y can be extracted from the target matrix $Y_{1:T}$. Finally, given the original dataset sentence, the probability that the generated sentence is being real can be computed as:

$$p = \sigma(V\left[c_x; c_y\right]) \quad (14)$$

where V is the transform matrix which transforms the concatenation of c_x and c_y into a 2-dimension embedding and σ is the logistic function.

D. The Penalty-Based Objective

We force the generator to generate diversified examples by changing the loss function, rather than just generate repetitive and "safe" samples, so as to avoid the problem of mode collapse and help improve the diversity and quality of generated texts. We use the concept of the objective function in SentiGAN and compare the generator's objective function of GAN, SeqGAN and SentiGAN as follows:

$$J_G(X) = \begin{cases} E_{X \sim P_g}[-\log(D(X;\theta_d))] \\ \quad GAN \\ E_{X \sim P_g}\left[-\log(G(X|S;\theta_g)D(X;\theta_d))\right] \\ \quad SeqGAN \\ E_{X \sim P_g}\left[G(X|S;\theta_g)V(X)\right] \\ \quad Our\ Model \end{cases} \quad (15)$$

There are two main improvements in the objective function. First, our penalty-based objective function can be viewed as a measure of the Wasserstein distance [Arjovsky et al. 2017] and can provide a meaningful gradient, while the other two loss functions cannot. Second, we use loss rather than reward. Our penalty-based loss function equation $G(X|S;\theta_g)V(X)$ can be viewed as adding $G(X \mid S;\theta_g)$ to the reward-based loss function $(-G(X|S;\theta_g)D(X;\theta_d))$, so it can generate more diversified examples rather than repeating "good" samples.

IV. Experiments

In this section, we firstly describe the experiment data from two datasets (product descriptions and movie reviews). We evaluated the text generation performance of PB-ESGAN and compared it with the results of several state-of-the-art text generation methods.

A. Datasets and Preprocessing

For text generation, we chose two different datasets: the product description dataset in the Alibaba Cloud and the movie review dataset from Douban. The corpus of the product description dataset is collected from Alibaba, China's largest e-commerce website, and includes the product title and product description. The other corpus used is reviews collected form the largest Chinese movie review website www.douban.com. The preprocessing steps of the two datasets are the same. The Chinese sentences were segmented by the word segmentation toolkit Jieba. To speed up the training procedure, when conducting experiments on the model, we randomly selected 100000 records in each corpus as datasets.

B. Model Parameters and Evaluation

For the generator, we set the dimension of word embedding as 512, dropout rate as 0.1 and the head number as 8, following the Transformer. The encoder and decoder both have a stack of 6 layers. We use beam search with a beam size of 4 and length penalty =0.6. All models are implemented in Pytorch.

We use BLEU score as an evaluation metric to measure the similarity degree between the generated texts and the human-created texts. BLEU is originally designed to automatically judge the machine translation quality [Papineni et al. 2002]. The key point is to compare the similarity between the results created by machine and the references provided by human.

In addition to BLEU-3 and BLEU-4, we also use self-BLEU, a metric to evaluate the diversity of the generated data. Since BLEU aims to assess how similar two sentences are, it can also be used to evaluate how one sentence resembles the rest in a generated collection. Regarding one sentence as hypothesis and the others as reference, we can calculate BLEU score for every generated sentence, and define the average BLEU score to be the self-BLEU of the document. We use self-BLEU-3 as the self-BLEU value.

A higher self-BLEU score implies less diversity of the document, and more serious mode collapse of the GAN model.

C. Experimental Results

In this section, we will give the experimental results of our proposed model and previous state-of-the-art methods on product description dataset and movie review dataset.

We compared our proposed PB-ESGAN model with several current models, including the main comparison with the original seqGAN model. Our model is not only better than the seqGAN model, but also shows better text generation performance compared to some seqGAN optimization models. We use BLEU-3, BLEU-4 and self-BLEU to measure the similarity between the machine-generated text and the artificially generated text from various aspects.

Table Ⅰ gives the results of the four models on the product description dataset and the results on the movie review dataset

TABLE I. EXPERIMENTAL RESULTS ON PRODUCT DESCRIPTION DATASET

Models	*BLEU-3*	*BLEU-4*	*self-BLEU*
seqGAN	0.632	0.489	0.685
+Transformer	0.649	0.492	0.678
+Transformer with Modeling Localness	0.663	0.509	0.656
Our model	**0.681**	**0.514**	**0.644**

TABLE II. EXPERIMENTAL RESULTS ON MOVIE REVIEW DATASET

Models	*BLEU-3*	*BLEU-4*	*self-BLEU*
seqGAN	0.527	0.419	0.585
+Transformer	0.544	0.422	0.578
+Transformer with Modeling Localness	0.571	0.450	0.569
Our model	**0.593**	**0.468**	**0.556**

are shown in Table Ⅱ. These results indicate that the proposed PB-ESGAN consistently outperforms the baselines and it shows better text generation performance than the naive seqGAN and the model guided only by the penalty-based objective

D. Results Analysis

With the full use of text information and the introduction of the self-attention with localness modeling and the penalty-based objective function, our model shows good performance on various datasets. As shown in Fig. 2, compared with the original seqGAN model, our model is significantly improved in product description dataset, and achieves improvement up to +4.9 BLEU-3 points. Compared with enhanced seqGAN, which introduces Transformer with modeling localness, our method still achieves +1.8 BLEU-3 points improvement. In movie review dataset, the PB-ESGAN leads to more significant improvement, +6.6 BLEU-3 points compared to the original model, which is shown in Fig. 3.

Besides, the self-BLEU score has decreased significantly, this means that the diversity of the documents generated by our model is more abundant, and the problem of mode collapse has also been solved. This shows that our proposed model can be applied to text generation with better performance.

Experiments on different datasets show that the Transformer significantly improves the ability to extract text features, and the self-attention with localness modeling is better than Transformer. Compared to the naive seqGAN, which is trained with the original loss function, PB-ESGAN utilizes the penalty-based objective function to guide the generator to generate sentences with higher BLEU points. Actually, the PB-ESGAN incorporates the advantages of the above two approaches to generate higher-quality text.

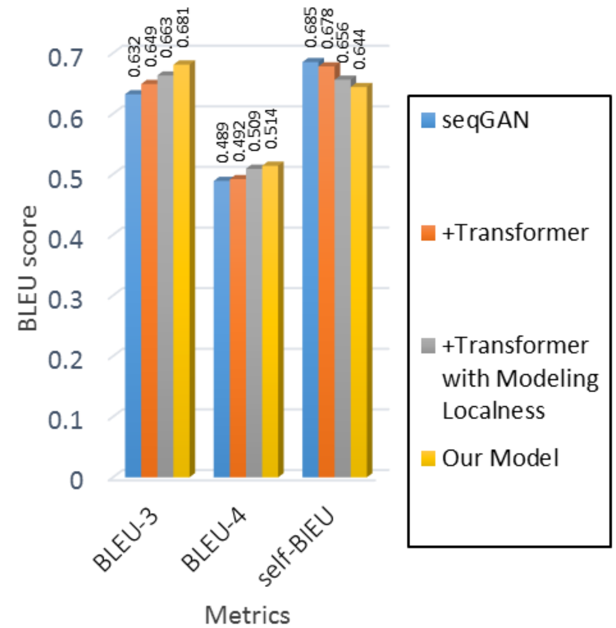

Figure 2. BLEU-3, BLEU-4 and self-BLEU of four models on the product description dataset.

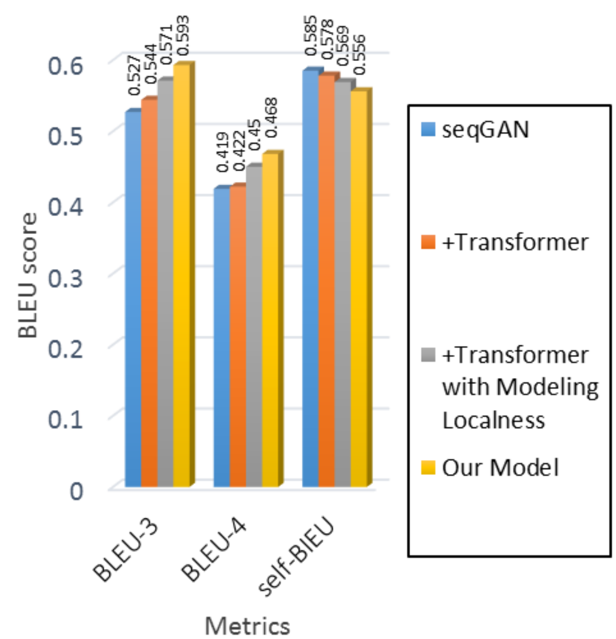

Figure 3. BLEU-3, BLEU-4 and self-BLEU of four models on movie review dataset.

At the same time, due to the self-attention model and the penalty-based objective function added to our model, the speed will be reduced during training, so the hyperparameter setting of the neural network and the optimization of the model will be the focus of our subsequent work.

V. CONCLUSION AND FUTURE WORK

In this work, to unlock the potential of GAN in NLP, we proposed a novel Penalty-based Enhanced Sequence

Generative Adversarial Network model. PB-ESGAN combines self-attention with localness modeling and modifies the objective function to be based on the minimum penalty. To verify the effectiveness of our approach, we compare four models on two datasets. These text generation tasks show that our approach consistently achieves significant improvements. We will further scale up the model to flexible types of text, and explore more interesting and useful applications.

REFERENCES

[1] Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Networks." (2014).

[2] Yu, Lantao, Zhang, Weinan, Wang, Jun, and Yu, Yong. "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient." (2016).

[3] Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. "Convolutional Sequence to Sequence Learning." (2017).

[4] Ledig, Theis, Huszar, Caballero, Cunningham, Acosta, Aitken, Tejani, Totz, Wang, and Shi. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network." (2016).

[5] Wang, K., and X. Wan. "Automatic Generation of Sentimental Texts via Mixture Adversarial Networks." Artificial Intelligence 275 (2019): 540-558.

[6] Mikolov, Tomas, Stefan Kombrink, Lukas Burget, Jan Honza Cernocky, and Sanjeev Khudanpur. "Extensions of Recurrent Neural Network Language Model." 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2011): 5528-531.

[7] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. "Scheduled sampling for sequence prediction with recurrent neural networks." (2011).

[8] Denton, Emily, Soumith Chintala, Arthur Szlam, and Rob Fergus. "Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks." (2015).

[9] Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. "Image-to-Image Translation with Conditional Adversarial Networks." (2016).

[10] Salimans, Tim, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved Techniques for Training GANs." (2016).

[11] Kusner, Matt J., and José Miguel Hernández-Lobato. "GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution." (2016).

[12] Yang, Baosong, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. "Modeling Localness for Self-Attention Networks." (2018).

[13] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need." (2017).

[14] Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." (2014).

[15] Lai S, Xu L, Liu K, and Zhao J. " Recurrent convolutional neural networks for text classification." (2015).

[16] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein GAN." (2017).

[17] Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. " Bleu: a method for automatic evaluation of machine translation." (2002).