IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON INTELLIGENT INFORMATION SERVICES

Received December 30, 2019, accepted January 8, 2020, date of publication January 13, 2020, date of current version January 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966291

Text Generation Service Model Based on Truth-Guided SeqGAN

YUXI WU AND JUNLI WANG
College of Electronics and Information Engineering, Tongji University, Shanghai 200092, China
Corresponding author: Junli Wang (junliwang@tongji.edu.cn)

This work was supported by in part by the National Key Research and Development Program of China under Grant 2018YFC0831403, in part by the National Natural Science Foundation of China under Grant 61672381, and in part by the Fundamental Research Funds for the Central Universities.

ABSTRACT The Generative Adversarial Networks (GAN) has been successfully applied to the generation of text content such as poetry and speech, and it is a hot topic in the field of text generation. However, GAN has been facing the problem of training and convergence. For the generation model, this paper redefines on the loss function. The truth-guided method has been added to make the generated text closer to the real data. For the discriminant model, this paper designs a more suitable network structure. The self-attention mechanism has been added to the discrimination network to obtain richer semantic information. Finally, some experiments under different model structures and different parameters indicates the model with truth-guided and self-attention mechanism gets better results.

INDEX TERMS Text generation, generative adversarial networks, self-attention mechanism, truth-guided.

I. INTRODUCTION

With the rapid development of information network, service depth and service content are developing to a higher level. How to communication with service objects determines the accuracy and quality of information services. Text is one of the main carrier of information service. Text-based information service technologies include word segmentation, text classification [1], feature extraction and text generation. Text generation can provide information services in a certain area of interest and improve the enthusiasm and personality of the service object.

Text generation is an important research direction in the field of natural language processing, and text generation technology has great application prospects. For example, text generation technology can be applied to intelligent question and answer, machine translation and other systems to achieve more intelligent and natural human-computer interaction [2]. Text generation system can also implement automatic writing and publishing of news[3]. The current text generation technology has been used to generate scenes such as Jin Yong's novels, Shakespeare's works, and US President Trump's speech.

The associate editor coordinating the review of this manuscript and approving it for publication was Shouguang Wang.

Besides, generated texts could be used in different areas nowadays, including the network text analysis [4] that can be better express the relationship and meaning between words. Amancio and Raphael [5] proposed another great application of text generation technology, which is a methodology to distinguish real manuscripts from those generated models.

Text generation refers to using Machine Learning and Natural Language Processing technologies to enable artificial intelligence to have human language expression capabilities, including machine translation, sentence generation, and dialogue generation. Usually the strategy for text generation is using language model. This probabilistic model can predict the next most likely word based on the input data. Moreover, as a sequence data, there is a contextual relationship between two words. Therefore, deep learning-based text generation usually uses Recurrent Neural Networks and their variants (LSTM, and GRU) for semantic modeling.

Character based Recurrent Neural Network is a classic text generation technology based on deep learning [6]. Char-RNN's input and output are all in character units. The learning target of the network is to make the next character consistent with the target output of the training sample. Based on the classical sequence-to-sequence model, Gehrmann *et al.* [7] introduce multiple decoders, and use the hidden variable factor to specify which decoder generates the final text.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

In addition to using basic Recurrent Neural Network, Variational Auto Encoder is often used in text generation tasks [8]. VAE adds constraints to the encoder so that the encoder produces potential variables that obey the Gaussian distribution. The loss function of the network consists of two parts. One is the reconstruction error, which can be measured by the mean square error; the other is the difference between the distribution of potential variables and the Gaussian distribution, which can be measured by KL divergence. Ankush Gupta [9] proposed a new simplified framework based on VAE and LSTM, which can generate paraphrase for a given sentence.

Similar to VAE, Generative Adversarial Networks [10] contains two sub-models: the generation model G and the discriminant model D. The role of the generation model is to simulate the distribution of real data. The role of the discriminant model is to determine whether a sample is a real sample or a fake sample. The training goal of the network is to make the generation model perfectly fit the distribution of real data, make the discriminant model indistinguishable. GAN is very suitable for continuous data such as images, but it has difficulties in the face of discrete data such as text. The most fundamental reason is that during the sampling process, the probability output is extracted into discrete output. Some research scholars have directly improved the network by micro tuning the internal calculation methods of GAN. From the point of view, through replacing KL divergence, the return value of the Wasserstein GAN [11] is smoother. Some scholars have improved the Softmax function, such as Gumbel-Softmax [12].

Reinforce Learning is currently one of the most advanced learning methods. SeqGAN [13] for the first time combines the reinforce learning and Generative Adversarial Networks and receives an outstanding representative of GAN's application to text generation. The paper [14] proposes that discriminator D is used as the source of reward value in reinforce learning. The generator looks for the word with the highest expectation of reward word by word. The discriminator gives a reward for the generated sequence in units of the whole sentence. During the training process, a discriminator training follows the multiple rounds of generator training.

The proposal of SeqGAN has made the application of reinforce learning and GAN to text generation a focus. However, GAN is difficult to train, and that limits its potential in the field of natural language processing. Since the output of generator G is discrete, discriminator D is difficult to update the gradient of generator G, so it is necessary to learn from the idea of reinforce learning. SeqGAN regards the generated word as the current state and the next word to be generated as an action. Since the discriminator D can only score a complete sequence, the possibility of subsequent occurrence of each action is supplemented by the Monte Carlo tree search method. The discriminator D judges these complete sequences and produces rewards. The parameters are updated back to the generator G, and the network can be trained to produce the next optimal action by reinforce learning.

Some people continue to improve based on SeqGAN. JiWei *et al.* [15] applied it to the dialogue generation task, and used the classic seq2seq model as the generator. They also used the result of discriminator D as the reward in reinforce learning, but it was calculated differently. The paper randomly selected one of the sequences of positive and negative sequences to train discriminator D. LeakGAN [16] mainly starts from the perspective of the input text length, and through the discriminator extracting the characteristics in the intermediate time step, the generator is instructed to generate a more complete sequence. Cooperative Training [17] proposed a new network training method, mainly to solve the problem that SeqGAN needs to perform pre-training using maximum likelihood. The paper introduced the harmonic M and used the estimate of the harmonic to replace the real value to train the generator G.

In SeqGAN, the Monte Carlo method is used to sample text sequences. Since there is no real data boot, it may repeatedly generate useless text, making the training unable to converge for a long time. In order to make SeqGAN more design in training, this paper proposes a new way to improve the generation model and discriminant model. The main works are:

1) A formal description of the generation model of SeqGAN based on truth guidance is established, including the initial state, transfer model, and cost function of the text generation problem. In the definition of cost function, the distance difference between generated text sequence and real text data is introduced as the total reward value to guide the network to converge to the optimal value faster.

2) A discriminant network model based on self-attention mechanism is designed to enable the model to synthesize the semantic expression of the entire text sequence for generation, called Self_combined model.

3) Experiments have verified the generation performance of the truth-guided generation model and the Self_combined model under the conditions of different proportional coefficients β, different hidden layers, and window size.

II. MODEL FRAMEWORK

A. FORMAL DESCRIPTION OF THE GENERATION MODEL

In this paper, the model under the truth guided based on the reinforcement learning and is called TG-SeqGAN. Each text generation process is regarded as a decision, and the discriminator D is used to judge the reward value of the current decision. The purpose of generator G is to make the next decision obtain a higher reward value. The product of the two is the target function that the network wants to maximize at that time. The sketch map of the whole model is shown in Figure 1.

As you can see, compared to the classic SeqGAN, TG-SeqGAN redefines the calculation of the reward value returned to the generator G. The total reward value is divided into two parts. Network-Reward is the discriminant score of the discriminator D for the sampled complete sequence.

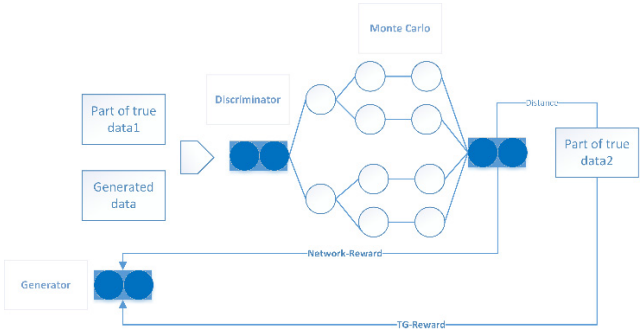


FIGURE 1. SeqGAN sketch map based on truth guidance.

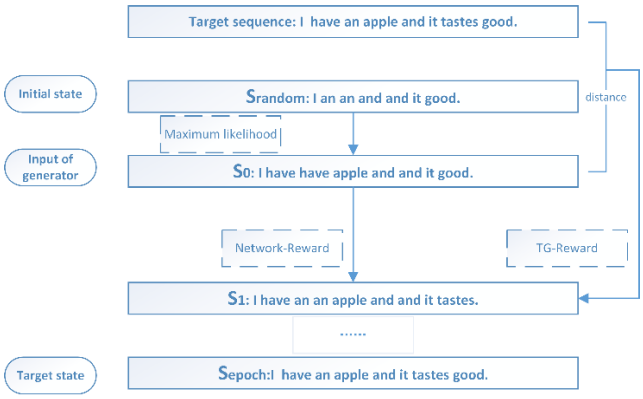


FIGURE 2. A example of the state transfer model.

The closer it is to a real text sequence, the closer the score is to 1. TG-Reward represents the semantic distance between the current sequence and the real text data in a batch. The smaller the distance, the closer the text sequence is to the expression of a real text. How the two rewards are calculated will be introduced in the following.

1) NETWORK-REWARD

Network-Reward is the normal reward obtained by the discriminator D. To introduce more clearly, in this paper, a formal description of the generation model is given, including: initial state, transfer model, cost function, etc.

Definition 1: The initial state S_{Random} refers to a randomly generated discrete vector whose length is the maximum length of the text sequence.

Definition 2: After S_{Random} passes through the pre-training network, a text vector S_0 with a certain semantic expression ability is obtained, and this state becomes the initial input state of the generation model.

As shown in the Figure2, the target sequence is "I have an apple and it tastes good.". The initial state is a random sequence vector, denoted as S_{random}, which represents a text whose semantic information is not clear. Due to the difficulty of training the GAN, the maximum likelihood method is used to do the pre-training. After the pre-training, the initial sequence S_0 entered into TG-SeqGAN is obtained. For each sequence state S_i, the total reward value determines which state S_{i+1} is to be transferred next. After N epochs, it finally reaches the target state S_{epoh}.

Definition 3: The Network-Reward value $Q_D^{G(\theta \mid S_i)}$ for a single time step refers to the reward value corresponding to the new word generated for each time step, which is accumulated to calculate the total Network-Reward value of the sentence generated by the entire network. The Network-Reward calculation for a single time step is shown in formula1:

$$Q_D^{G(\theta|S_i)}(S = Y_{1:T-1}) = D(Y_{1:T}) \tag{1}$$

Among them, $G^{(\theta|S_i)}$ represents a generated text vector under the parameter θ and state S_i. D represents the discriminant score of discriminator D; $Y_i$represents the sequence obtained at the time i. The state S can be taken from time 1 to time T-1, and the final output is the cumulative reward result from time 1 to time T.

Since discriminator D can only score complete sequences, if the word generated at the current moment is not the last word of the sentence, all of the possible sequences need to be filled through the Monte Carlo search. Then the reward values for each possible sequence are calculated. Take the average value as the reward value for the current text sequence.

$$Q_D^{G(\theta|S_i)}(S = Y_{1:T-1}) \begin{cases} \frac{1}{N}\sum_1^N D\left(Y_{1:T}^n\right), & \\ Y_{1:T}^n \in MC(Y_{1:t}; N) & if\ t < T \\ D\left(Y_{1:t}\right) & if\ t = T \end{cases} \tag{2}$$

TG-SeqGAN calculates the Network-Reward value only through the feedback of discriminator D, and then adjusts the network that have the same problem as training RNN. At the beginning of the training, due to the randomness of the initial value S0, many texts that are clearly "fake data" after judging by discriminator D. In the absence of truth data booting, such repeated generation can cause waste. In this paper, we propose to add the truth-guided to the calculation process of this reward and optimize the network by calculating the distance between the current text sequence and the training text data.

The calculation of the reward value depends on the results of the current state Si and Monte Carlo sampling. For each time step T, assuming that the current T word of the generator can be generated correctly, the rest of the words are required to complete this complete sequence. Finally, the sum of the Network-Reward of all time steps is calculated to obtain the total reward of the target state transfer.

Definition 4: The sentence credibility α' is the proportion that words that have been generated in the current time step of the total number of words in the sentence. When the ratio is greater than the set threshold, it means that the current sentence is credible and the calculation results are added to the final loss function. Otherwise, the randomness of the sentence is too large to be added to the calculation of the loss function.

$$\alpha' = \frac{n}{N} \tag{3}$$

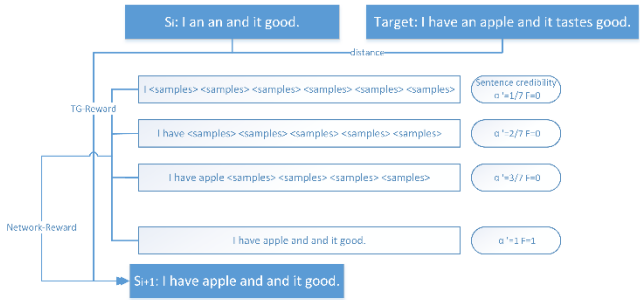


FIGURE 3. A example of total reward calculation.

$$\begin{cases} \mathrm{F} = 1 \ \alpha' \geq \alpha \\ \mathrm{F} = 0 \ \alpha' < \alpha \end{cases} \tag{4}$$

where n represents the number of words that have been generated by the generator, N represents the total number of words in the sentence, and F represents whether the text sequence is added to the final cost function.

2) TG-REWARD

Definition 5: Lsim measures the similarity between the current sequence and the real text data that measures it by means of European distances, cosine distances, etc., and optimizes it as part of the final objective function. The calculation formula is as follows:

$$L_{sim}\left(G^{(\theta|S_i)}, gt\right) = \gamma_1 Dis_{\cos\theta} + \gamma_2 Dis_o \tag{5}$$

where gt represents the vector representation of real text; $Dis_{\cos\theta}$ represents the cosine distance between two vectors, and Dis_o represents the European distance.

It is introduced in Definition 3 that when the state S is between 1 and T-1, since the discriminator D can only calculate a complete sequence. SeqGAN uses the Monte Carlo search method to sample the sentence for completion. Therefore, the proportion of non-sampling parts in a complete sentence also changes with the time step. The larger the proportion of non-sampling parts, the greater the ability of the sentence to express the generator G's ability to generate, it is more credible.

Total reward means the sum of the TG-Reward and the Network-Reward. As can be seen from Figure 3, the two reward values are obtained at the same time, the sentence reliability is mainly in the Network-Reward, TG-Reward is calculated by vector distance alone, and can be obtained synchronously with the training. In this Figure, the threshold is 0.5 when calculates F. In addition, as you can see, α'depends on the number of word that has already been generate.

3) TRAINING OF THE TG-SeqGAN

For GAN-based models, the overall goal of optimization is shown in formula.

$$\max_D \max_G V(D, G) = E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{6}$$

That is, the generator G hopes to minimize this cost function, so that the discriminator D cannot distinguish between the text it generates and the real text data. The discriminator D is just the opposite, trying to maximize this difference to distinguish whether the current text data is generated data.

Among them, for the generator G part, the reward value corresponding to each time step is calculated by means of reinforce learning as part of the generator G cost function. The reward value is calculated as follows:

$$J_{(\theta)} = E[R_T|S_i, \theta] = \sum\nolimits_{y_i\in Y} G(\theta|S_i) * Q_G^{D(\theta|S_i)} \quad \mathrm{i,j} = 1, 2 \ldots T \tag{7}$$

Under the conditions determined by the initial state S0 and the model parameter θ, the calculation results of definition 2 and definition 3 can be obtained. In the truth-guided model proposed in this paper, the calculation of $J_{(\theta)}$is different. The formula is as follows:

$$J_{(\theta)} = E[R_T|S_i, \theta] = \beta * \sum\nolimits_{y_i\in Y} G(\theta|S_i) * Q_G^{D(\theta|S_i)} + \alpha * L_{sim}(G^{(\theta|S_i)}, gt) \quad \mathrm{i,j} = 1, 2 \ldots T \tag{8}$$

The loss item Lsim is the difference between the generated result of generator G and the real text data gt. The weight of this loss is determined by the credibility α and the ratio coefficient β. The credibility α is different for different time steps. The ratio coefficient β is a definite value for the same network and the optimal solution can be determined by multiple experiments.

B. DISCRIMINANT MODEL

In the original SeqGAN, the generator G uses a traditional RNN structure, which can select an optimized LSTM or GRU structure. The discriminator D uses a one-dimensional convolution neural network, and at the same time, the classification effect is better. TG-SeqGAN also introduced highway-network [18].

For discriminators, the network structure is simple and fast, but it can be further optimized in terms of accuracy and convergence speed.

The self-attention mechanism is first proposed in the paper [19] and has a very strong ability to solve the problem of long-distance dependence sequences. In the original SeqGAN, the convolution neural network is used to encode the output vector sequence. This is actually a "local code" that can only establish short-distance dependence. The attention mechanism can "dynamically" generate different connection weights and can handle longer information sequences.

Text classification models that combine Convolutional Neural Network (CNN) and Attention Mechanisms has already been widely used, and there are many ways to combine the two structures. These two structures capture different text features. Therefore, the best way they work together is in parallel. If you design self-attention after CNN, it will result in semantic loss, and the attention mechanism will make mistakes.

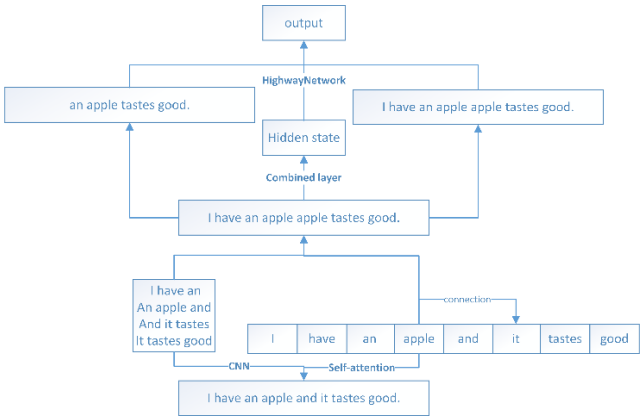


FIGURE 4. A example to discribe the structure of Self_combined model.

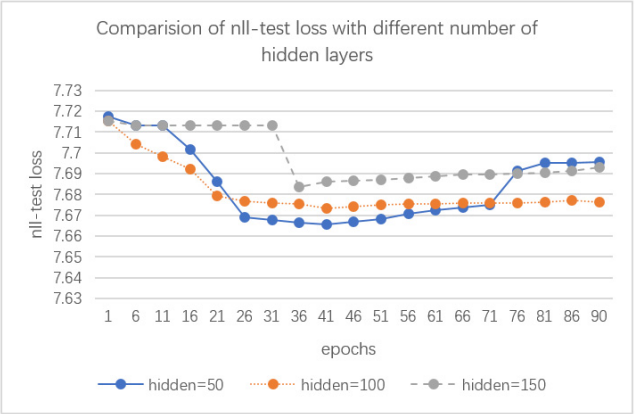


FIGURE 5. Comparision of nll-test loss with different number of hidden layers.

In this paper, the Self_combined model is composed of two parallel modules. On the left is the convolution neural network CNN. It is used to extract the strong semantic information in the sequence. On the right is the self-attention layer. Both of them use the generation vector as input.

In the self-attention layer, attention focus and attention source are the generating vector matrices of the input generator. The formula for the attention weight is:

$$f(x_i) = \sum_j V^T \cdot \tanh(U \cdot x_i) \tag{9}$$

Considering that the characteristics obtained by convolution neural network or recurrent neural network model are also very helpful for classification, the discriminant network model combines the results of the attention layer with the output results of the parallel neural network layer. The neural network layer formula combining two parts is as follows:

$$h = \tanh(W_1 a + W_2 r + b) \tag{10}$$

Among them, a is the output of the attention layer, which is the result of the maximum pooling of the convolution neural network layer, and b represents bias.

The improvement of the network structure can be seen in the Figure 4. The input sequence is "I have an apple and it tastes good.". Using CNN will obtain some local semantic information, such as "I have an","an apple and","it tastes good". Meanwhile, using self-attention will connect "apple" and "it" together. After combined there semantic features, the discriminant model of the TG-SeqGAN using Highway Network to keep important information of the input sentence and then obtain the output result.

III. MODEL PERFORMANCE ANALYSIS

In this paper, the model based on truth guidance and the discriminant model based on self-attention mechanism is analyzed and compared experimentally. In addition, the variable parameters (such as reward weight β) are optimized.

MS COCO Dataset:

The MS COCO dataset originated from Microsoft COCO dataset, which is funded by Microsoft in 2014. It is a large, rich object detection, segmentation and subtitle dataset. Includes 80 categories object instance segmentation, 91 categories item segmentation, key point detection of character instances, and 5 image titles for each image.

A. EVALUATION INDICATORS

1) NLL-TEST LOSS

The new metric NLL-test loss is defined, which is the average negative logarithmic possibility of real test data from the generator. The calculation formula is as follows:

$$NLL_{test} = -E_{Y_{1:T} \sim G_{oracle}} [\sum_{t=1}^{T} \log(G_\theta(y_t | Y_{1:t-1}))] \tag{11}$$

Among them, G_{oracle} [20] represents the result of a True Model that has been trained before, and the True Model refers to a randomly Initialized LSTM. $G_\theta(y_t|Y_{1:t-1})$ represents the result of the generator G from time step 1 to time step t $-$ 1 under parameter θ.

2) EMBEDDING SIMILARITY

Inspired by BLEU, compared to the traditional method of similarity calculation words one by one, the literature proposes a word vector based method to judge the similarity of the two documents. First, text in a real data set is trained to express word vectors through a skip-gram model. For each word vector, calculate the cosine distance of the word for other words and get the matrix W, where each Wi, j $=$ $\cos\left(e'_i, e_j\right)$. This matrix is defined as a similar matrix of a real data set. By the same method, the word vector is calculated for generating data and a similar matrix, W 'is obtained.

Through these two similar matrices, we can calculate Embedding Similarity. The calculation formula is as follows:

$$EmbSim = \log(\frac{\sum_{i=1}^{N} \cos\left(W'_i, W_i\right)}{N}) \tag{12}$$

Among them, W_i represents the line i in the similarity matrix W, and N represents the total number of words.

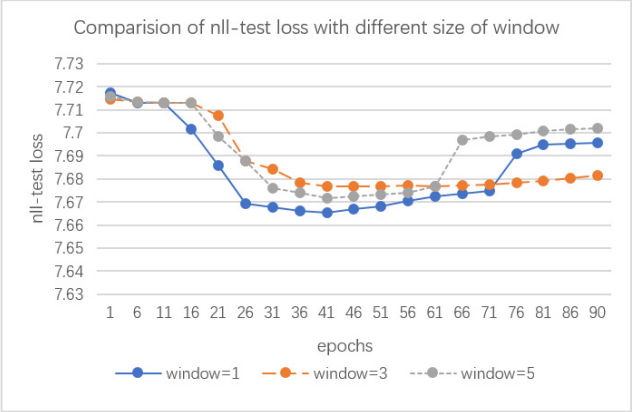


FIGURE 6. Comparision of nll-test loss with different size of window.

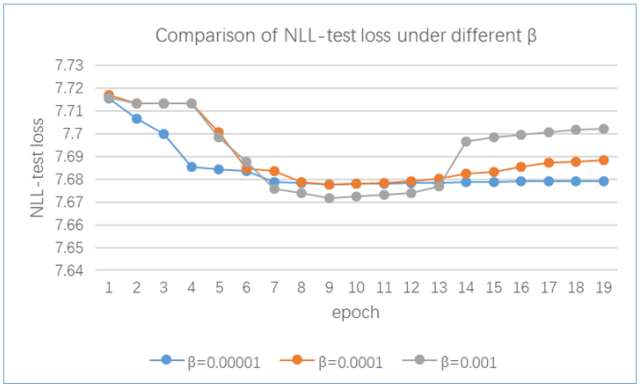


FIGURE 7. Comparison of NLL-test loss under different β.

TABLE 1. Embedding Similarity comparison under different β values.

	β =0.001	β=0.0001	β=0.00001
Embedding Similarity	-0.01570866	-0.01834203	-0.0157715

B. PARAMETER ANALYSIS AND SETTINGS

1) In the truth guidance generation model, in addition to the original reward value in SeqGAN, this article adds a new reward value to the generation model: the semantic distance between the text sequence generated by the current generator G and a batch of true text data. The optimal value of the proportional coefficient β can be obtained through experiments.

In the process of network convergence, as the number of hidden layer nerve elements increases, the value of NLL-test loss also increases, indicating that too many neurons will produce over fitting and reduce the performance of the network, so the optimal number of hidden layers is 50. By the time the number of hidden layers is 150, the network has experienced dramatic fluctuations and is very unstable.

2) There are two hidden layer parameters from the attention layer, U and V. Where U represents the number of hidden layer units from the attention layer, which can be compared with values 50, 100, and 150, respectively. V can be seen as each input has a corresponding score of V. These scores can be added and then normalized, and values 1, 3, and 5 can be compared, respectively.

The window here refers to the entered fractions, which can be added to obtain more balanced semantic information. The window here refers to the entered fractions, which can be added to obtain more balanced semantic information. It can be seen that when the window is 1, the effect is better, so the window value is fixed to 1 in the later experiment.

C. MODEL PERFORMANCE ANALYSIS

The performance analysis of the model is compared from two aspects: adjusting the network structure and adjusting the proportional coefficient of the reward value. From the point of view of network structure, there are several different ways of attention mechanism combined with convolution neural network: the attention layer can be used instead of the pool layer in convolution neural network and try to accumulate the results of multiple attention fusion, called Cov_attention model. It is also possible to add self-attention mechanism to the input layer to make the semantic information of the input layer more abundant. In order not to affect the extraction of strong semantic information in the sentence by the original convolution neural network, this paper designs a combination of two encoding results.

From the point of view of adjusting the proportional coefficient β of the reward value, the β can measure the importance of the discriminator D Network-Reward and the true value part TG-Reward. Under different β, the comparison of the model's NLL-test loss and Embedding Similarity is shown in the Figure 7.

As can be seen from Figure 7, when the ratio coefficient is 0.001, the experimental results obtained are better, and as the ratio decreases, the effect of the reward value of the truth-guided part decreases, and the quality of the model generation decreases. It shows that the truth guidance part does play a positive role in guiding the generation model to continuously generate text that is close to real data.

Table 1 shows the difference in Embedding Similarity under different proportional coefficients. As the proportion of truth guidance decreases, the semantic similarity between word vectors gradually increases. By adjusting β, the optimal ratio of Network-Reward to TG-Reward can be obtained. Help the model get better word vector similarity.

2) Under different network structures, the comparison of the model's NLL-test loss and Embedding Similarity is shown in the Figure 8:

As can be seen from Figure 8, the structure of the Self_combined model is very helpful for extracting semantic information in the original text. By extracting semantic features from the text sequence generated by generator G from self-attention and convolution, the convergence of the network can be accelerated. The smaller generated text of NLL-test Loss is obtained, and the overall effect is increased by 0.1 %. The method of using attention mechanism to

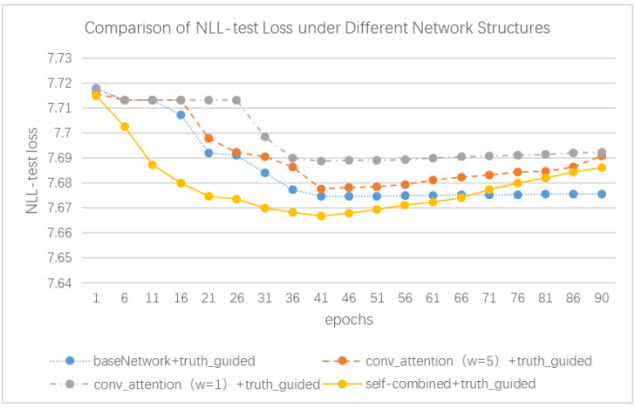


FIGURE 8. Comparison of NLL-test loss under different network structures.

TABLE 2. Comparison of embedding similarity under different network structures.

	baseNetwork+truth_guided	conv_attention(w=1)+truth_guided	selfcombined+truth_guided
Embedding Similarity	-0.01391775	-0.01520435	-0.01597811

replace convolution pool layer has reduced the performance of the model because it loses the expression of the strongest semantic information.

Table 2 compares the differences in Embedding Similarity under different network structures. After changing the network structure, the generated words are more diverse and learn semantic rather than Glyph information, so the semantic similarity will decrease, but the overall deviation is not large.

IV. SUMMARY

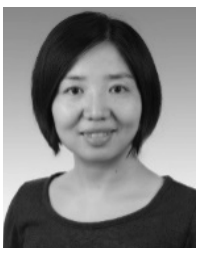
In this paper, the text generation task is studied and analyzed and SeqGAN is used as the base model. To accelerate the convergence of the network and guide the generation model to generate more similar to real text, TG-SeqGAN chose to use truth-guided. At the same time, the network structure of the discriminant model is improved, and the semantic information of the whole sentence is fused by the self-attention mechanism. SeqGAN based on true value guidance has improved the convergence speed of the text generation model, and the text quality generated by the network has also been improved after stabilization, both on NLL-test loss and Embedding Similarity.

REFERENCES

[1] H. F. De Arruda, L. D. F. Costa, and D. R. Amancio, "Using complex networks for text classification: Discriminating informative and imaginative documents," *Europhys. Lett.*, vol. 113, no. 2, Jan. 2016, Art. no. 28007.
[2] S. Lu, Y. Zhu, and W. Zhang, "Neural text generation: Past, present and beyond," *Comput. Lang.*, 2018.
[3] Y. Chen, L. Xueqiang, and J. Zhou, "Research on automatic writing of NBA sports news," *Acta Scientiarum Naturalium Universitatis Pekinensis*, vol. 53, no. 2, pp. 211–218, 2017.
[4] D. R. Amancio, F. N. Silva, and L. D. F. Costa, "Concentric network symmetry grasps authors' styles in word adjacency networks," *Europhys. Lett.*, vol. 110, no. 6, 2015, Art. no. 68001.
[5] D. R. Amancio, "Comparing the topological properties of real and artificially generated scientific manuscripts," *Scientometrics*, vol. 105, no. 3, pp. 1763–1779, Dec. 2015.
[6] Z. Xiang, J. Zhao, and Y. Lecun, "Character-level convolutional networks for text classification," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015.
[7] S. Gehrmann, F. Dai, H. Elder, and A. Rush, "End-to-end content and plan selection for data-to-text generation," in *Proc. 11th Int. Conf. Natural Lang. Gener.*, 2018.
[8] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: https://arxiv.org/abs/1312.6114
[9] A. Gupta, A. Agarwal, and P. Singh, "A deep generative framework for paraphrase generation," *Comput. Lang.*, 2017.
[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 3, 2014, pp. 2672–2680.
[11] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: https://arxiv.org/abs/1701.07875
[12] J. M. Kusner and J. M. Hernández-Lobato, "GANS for sequences of discrete elements with the Gumbel-softmax distribution," Tech. Rep., 2016.
[13] A. G. Barto, "Reinforcement learning," *A Bradford Book*, vol. 15, no. 7. 1998, pp. 665–685.
[14] L. Yu, W. Zhang, and J. Wang, "SeqGAN: Sequence generative adversarial nets with policy gradient," Tech. Rep., 2016.
[15] J. Li, W. Monroe, and A. Ritter, "Deep reinforcement learning for dialogue generation," Tech. Rep., 2016.
[16] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang, "Long text generation via adversarial training with leaked information," 2017, *arXiv:1709.08624*. [Online]. Available: https://arxiv.org/abs/1709.08624
[17] J. Xie, L. Yang, and R. Gao, "Cooperative training of descriptor and generator networks," 2016, p. 1.
[18] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *Comput. Sci.*, 2015.
[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*. [Online]. Available: https://arxiv.org/abs/1706.03762
[20] F. Huszár, "How (not) to train your generative model: Scheduled sampling, likelihood, adversary?" 2015, *arXiv:1511.05101*. [Online]. Available: https://arxiv.org/abs/1511.05101



YUXI WU received the B.S. degree in computer science from Northeastern University, Shenyang, China, in 2017. She is currently the Graduate Student with the College of Electronics and Information Engineering, Tongji University. Her research interests include text classification, text generation, and deep learning.



JUNLI WANG received the B.S. degree in computing mathematics from Jilin University, Changchun, China, in 2000, and the Ph.D. degree in computer science from Tongji University, Shanghai, China, in 2007. She is currently an Associate Researcher with the College of Electronics and Information Engineering, Tongji University. Her research interests include text data analysis, deep learning, and attention mechanism.