GAN for Paraphrase Generation

Peiyu Liao
Email: pyliao@stanford.edu

***Abstract*—**

Paraphrase generation is a crucial task in Natural Language Processing (NLP), useful in applications such as dialogue generation and dataset generation for other NLP tasks. In this project, a conditional SeqGAN model is proposed to approach this task. Since Generative Adversarial Network (GAN) is an effective generator model in the Computer Vision (CV) domain, yet its application in the NLP area is still limited, this project serves to explore its effectiveness in this complex NLP task. A BLEU-2 score of 14.79 and METEOR score of 15.85 are achieved on the Quora question pair dataset. Sample paraphrase results are presented, which demonstrates that the core paraphrasing techniques are captured by the model. The training process is discussed, and suggestions are made for enhancing the training efficiency and stability of a GAN model.

***Index Terms*—GAN, SeqGAN, Paraphrase Generation**

1. Introduction

Paraphrasing is a task of restating a sentence "in other words" while preserving its meaning [1]. For example, "Academic Sinica is far from my home" is a paraphrase of "there is a long distance between Academic Sinica and where I live". Paraphrase is used in everyday life for simplifying a complex sentence, avoiding plagiarism in academic works, etc.

Paraphrase generation, which is the task of generating multiple paraphrases given a sentence, finds useful applications in Natural Language Processing (NLP) such as dataset generation for QA systems, dialogue generation, and so on. Some of the main challenges of this task include accurate preservation of the original meaning, diversity of the generated results, and prevention of direct word copying.

Despite the importance of this task, relatively few previous works have approached it effectively due to its challenging nature. Traditionally there are more feature-engineered methods such as rule-based [2], grammar-based [3], and Statistical Machine Translation (SMT)-based [4] methods, while more recent works tend to integrate deep learning techniques, including the Variational Autoencoder (VAE) model by Gupta et al. [5] and the deep reinforcement learning-based approach by Li et al. [6].

Speaking of deep generative models, Generative Adversarial Network (GAN) [7] introduced by Goodfellow et al. in 2014 has been widely-favored for generation tasks in recent years. GAN has achieved extraordinary results in various Computer Vision (CV) tasks, including image generation and interpolation [8], image-to-image translation [9], [10], and style transfer [11]. A basic GAN consists of a generator and a discriminator, with the generator trying to fool the discriminator as producing real data, while the discriminator trying to discriminate between real data and the generated fake data. The two components are essentially playing a mini-max game to alternately push the limits of each other.

The conventional GAN was found to have intrinsic difficulties in its application on token generation due to its discrete sampling process [12]. In spite of this, several techniques have been incorporated to overcome the problem, such as SeqGAN with Reinforcement Learning (RL) [13] and Wasserstein GAN (WGAN) using Wasserstein distance for measuring distribution divergence [14]. In light of the achievements of these works, GAN is considered a potentially good candidate in paraphrase generation and detection, as the adversarial training involved can be valuable in

enhancing the performance of both the generator and the discriminator. As a first step, this project focuses solely on paraphrase generation, as the results can be more conveniently evaluated by human. Paraphrase detection is left for the next stage.

In this project, a model based on SeqGAN is developed for paraphrase generation. For a pair of paraphrases, one of them serves as the condition to the model, while the other serves as the real sample. The Quora dataset released in 2017 [15] is the training and testing dataset for the model. The objectives of this project include:

1) Constructing a SeqGAN-based model for paraphrase generation.
2) Assessing the applicability of SeqGAN on a more complex NLP task with conditions included.
3) Experimenting with the training stability of a GAN model and the effects of the added training techniques.

In this report, related works on deep learning-based paraphrase generation are briefly reviewed in Section 2. Section 3 presents the detailed design of the model. The dataset and evaluation criteria are described in Section 4, and the results are presented and discussed in Section 5. An overall discussion follows in Section 6. Section 7 concludes the project and presents some of the possible future works.

The full code of the project is on github: https://github.com/pyliaorachel/SeqGAN-paraphrase-generation.

2. Related Works

Gupta et al. [5] approached paraphrase generation with a combination of deep generative model VAE and Long Short-Term Memory (LSTM) model. Both the encoder and decoder of VAE take the original sentence as condition so that the generated sentence is matched with the condition. Remarkable results were reported, and the generated paraphrases achieved acceptable relevance and readability from human evaluations. Their work also set up the baseline performance on the Quora question pair dataset for paraphrase generation.

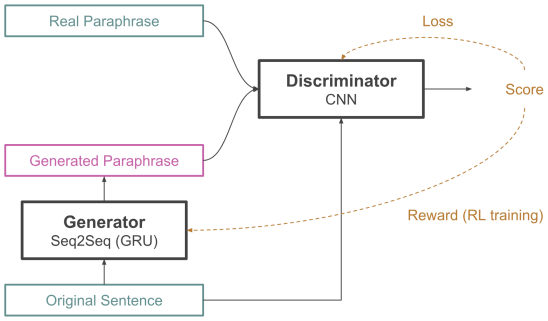
A later work evaluated on the Quora dataset was based on deep reinforcement learning [6]. The framework proposed consists of a Seq2Seq model as generator and a deep matching model as evaluator. The evaluator was trained through Inverse Reinforcement Learning (IRL) to achieve the optimal reward function for the input sequences, and the generator was trained through policy gradient with the rewards fed back from the evaluator.

While the methodology behind the IRL training method resembles GAN, the evaluator in this framework simply focuses on matching between the paraphrase sequence and the original sequence without considering the realness of the paraphrase sequences. This might run the risk of generating results that are similar to the original sentence under the specific evaluation measurement used in training, but are grammatically unreasonable. In addition, a matching criteria is necessary in the objective function of the matching model, which may be a restrictive requirement for the model.

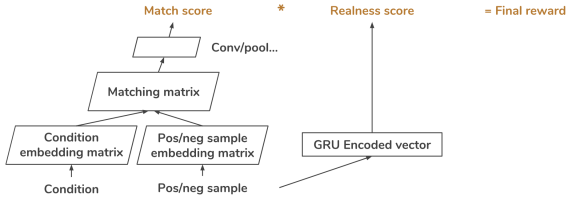
In our GAN model, the generator is also a Seq2Seq model, while the discriminator can be internally designed more flexibly than in previous works. It is also able to learn a more general reward function for scoring the paraphrase sequences.

3. Model

The model is comprised of a Seq2Seq generator and a CNN discriminator. It has the same training methodology as SeqGAN [13], in which the parameters in the generator is updated through Reinforcement Learning (RL) with the rewards received from the output of the discriminator, instead of through the backpropagated loss from the input to the discriminator. However, different from the standard SeqGAN, a condition is needed as an additional input to the discriminator. As a



(a) The model consists of a Seq2Seq generator and a CNN discriminator. The discriminator is trained by cross-entropy loss, and the generator is trained as a policy network via RL.



(b) The internal structure of the discriminator. The final output combines a match score between the original and paraphrase sentences with a realness score of the paraphrase sentence.

Figure 1: Conditional SeqGAN Model.

result, the model is referred to as a Conditional SeqGAN in this paper.

The overall model is shown in Figure 1a, and the details are delineated in the following subsections.

3.1. Conditional SeqGAN Model

A standard GAN [7] is composed of a generator G and a discriminator D. The generator tries to learn the true data distribution over some data by mapping a prior input noise z to data space $G(z)$. The discriminator learns to differentiate between the generated data distribution p_g and the true data distribution p_{data}, that is to output the probability $D(y)$ that the input y comes from the true data distribution. The generator is trained to maximize probability $D(G(z))$ so as to approximate the true data distribution, while the discriminator is trained to maximize the probability of correctly classifying the true and fake data. The objective of the discriminator with parameter set ϕ is to maximize the following equation:

$$J(\phi) = \mathbb{E}_{x\sim p_{data}}[logD(x)] + \mathbb{E}_{x\sim G}[log(1 - D(x))] \tag{1}$$

In SeqGAN, the generator is trained via policy gradient [16], with the rewards being the output from the discriminator. For a complete sequence Y of T timesteps, a state s is a subsequence $Y_{1:t-1}$, and an action a upon s is the next token y_t to be generated after the subsequence. The objective of the generator with parameter set θ is to maximize the expected end reward:

$$J(\theta) = \mathbb{E}[R_T|s_0, \theta] = \sum_{y\in V} G_\theta(y|s_0) \cdot Q^{G_\theta}_{D_\phi}(s_0, y) \tag{2}$$

where R_T is the accumulated reward at timestep T of Y, V is the vocabulary, s_0 is the start state, and $Q(s, a)$ is the Q-value function representing the expected end reward of taking action a at state s.

The derivation of the gradient of eq 2 is detailed in the original SeqGAN paper [13] and summarized below:

$$\nabla_\theta J(\theta) \simeq \sum_{t=1}^{T} \mathbb{E}_{y_t\sim G_\theta(y_t|Y_{1:t-1})}[\nabla_\theta \log G_\theta(y_t|Y_{1:t-1}) \cdot Q^{G_\theta}_{D_\phi}(Y_{1:t-1}, y_t)] \tag{3}$$

Training the model. The training procedure is outlined below:

1) Generator Maximum-Likelihood Estimation (MLE) pretraining with scheduled sampling [17]. Only positive (true) samples are involved in the training.
2) Discriminator pretraining. Both negative (generated) and positive (true) samples are involved.
3) Adversarial training. Generator trained by policy gradient for $g\text{–}step$ steps. Discriminator updated via cross-entropy loss for $d\text{–}step$ steps, each step trained for k epochs with the

same set of positive and negative examples.

The pretraining steps are added due to its demonstrated effectiveness in speeding up the training procedure [13].

3.2. Seq2Seq Generator

Seq2Seq is a Recurrent Neural Network (RNN) model that learns a mapping between variable-length input and output sequences [18], which is suitable for the generation task at hand. The input to the Seq2Seq model is the original sentence, and the output is the generated paraphrase sentence.

Gated Recurrent Unit (GRU) [19] is a variation of the vanilla RNN that was invented to capture long-term dependencies. Another variation Long Short-Term Memory (LSTM) [20] is of similar purpose but with an additional memory gate and thus more parameters to control over. As GRU is found to achieve comparable performance to LSTM under various experiments [21], GRU is chosen for our model for its relative simplicity.

To enhance the performance and stability of the training procedure, several training techniques are incorporated in the generator model:

Monte Carlo Search (MC search). In Seq2Seq, the discriminator only takes in a complete generated sequence, and the reward is an assessment on the entire sequence. However, intermediate tokens may have different contributions to the reward. For example, *I are a student* may get low rewards from the discriminator, but only the token *are* should be penalized heavily.

MC search [22] is one way to estimate rewards for individual tokens within a sequence. For each timestep t in a sequence of T tokens, the generator repeatedly generates the tokens from timestep $t+1$ to T under a rollout policy β for N times, where N is the rollout number. The N rewards are averaged as the estimated end reward for the action at timestep t.

Attention. In paraphrase generation, each part of the generated output corresponds to some part of the input condition. As a result, the attention mechanism [23] is added to our model to improve performance by aligning and attending to the input of the generation.

Reward Scaling. According to experiments, when the discriminator is much capable than the generator, the rewards assigned may be too small for the generator to update its parameters, that is the vanishing gradient problem. One solution to this is to rescale the rewards within a mini-batch so that the expectation and variance are constant across mini-batches. A ranking-based rescaling method [24] is utilized in our model:

$$R_i^t = \sigma(\delta \cdot (0.5 - \frac{rank(i)}{B})) \tag{4}$$

where R_i^t is the reward at the t-th timestep of the i-th item in a batch, $rank(i)$ is the ranking of this reward within the batch of the same timestep, and B is the batch size. δ controls the smoothness of the rescaling function, and σ re-projects the scores to an effective range. Taking reference from [24], $\delta = 12.0$ and $\sigma = sigmoid$ is chosen in our model.

Interleaved Training. After the MLE pretraining, there is no further opportunity for the generator to be trained by the positive samples directly. This runs the risk of the generator being trained out of track and cannot be easily tuned back to the correct training direction.

To ensure the generator does not misbehave seriously and to stabilize the training procedure, the policy gradient training in each adversarial training iteration is followed by a few number of epochs of MLE training with positive samples.

3.3. CNN Discriminator

The discriminator takes in an original sentence as the condition and the corresponding paraphrase sentence as the input to judge. The discriminator is essentially an evaluator assessing two criteria:

1) the match score, measuring the semantic closeness between the original and paraphrase sentences;

2) the realness score, measuring how natural the paraphrase is.

The condition and paraphrase are matched via a Convolutional Neural Network (CNN) model, which is based on the text matching model proposed by Pang et al. for image recognition [25]. Every token in a sequence of T timesteps is mapped with its word embedding vector of dimension E, which forms an embedding matrix of size $T \times E$. The two matrices, one from the condition and one from the paraphrase, are dotted with each other to form a matching matrix. This matrix is then forwarded through a CNN network to output the probability that the two sequences semantically match with each other.

On the other hand, the paraphrase sequence itself is encoded through a GRU network and forwarded through a fully-connected layer to output the probability of its realness.

The match score and the realness score are multiplied together to produce the final reward for the paraphrase sequence. The entire structure of the discriminator is visualized in Figure 1b.

4. Experiments

In this section, the dataset used in training and evaluating the model and the evaluation criteria of the experiments are described.

4.1. Dataset

The Quora question pair dataset [15], released in 2017, consists of question pairs that are or are not marked to be duplicates by users on the Quora website. Among all questions pairs, around 150K of them are marked to be duplicates, and they are taken as the paraphrase dataset for training and assessing our model.

One limitation of this dataset is that duplicate questions do not always equal paraphrases, since the duplicate questions may be too similar in their wordings, or they may have different meanings but similar scope such that the users do not think they should be answered repeatedly. This may limit our model to produce truly acceptable paraphrases given a sentence. However, since no other qualified paraphrase dataset was found to be available, and the Quora dataset was also used in some recent works on paraphrase generation, it was still adopted in our experiments.

Due to time and resource restrictions, and to match with the available experimental settings from previous works for comparison purpose, the dataset is split into a training set of 53K and a testing set of 3K pairs of questions. 10-fold cross-validation is applied during training.

4.2. Evaluation Criteria

To the best of our knowledge, no effective automatic evaluation metric exists for assessing paraphrase sentence pairs. In our experiment, two common alternatives are selected, the BLEU score [26] and the METEOR score [27]. Since they are both based on n-gram matching, exact same sentences are scored highly, which is not ideal for paraphrases. However, the scores still serve the purpose of verifying the reproduction of the same results in our experiments.

A more reasonable evaluation metric is human evaluation. This was not conducted in this project due to time limitation, but it should be the main evaluation criteria in the future development of this project.

5. Results

The conditional SeqGAN model was trained with the following settings: MC search rollout number = 3, batch size = 16, embedding dimension = 50, hidden dim = 64; during pretraining, generator epochs = 50, discriminator steps = 10, discriminator epochs = 5; during adversarial training, total iteration = 30, generator training steps = 10, discriminator steps = 5, discriminator epochs = 3. The word embeddings are pre-populated with the GloVe pretrained word embeddings [28].

5.1. Sample Results

Table 1 presents some sample generated results for the given sentences.

From the table, several signs of paraphrasing can be observed in each sample:

1) Structural difference.
2) Synonym, higher-level meaning.
3) Synonym, same meaning in other words.
4) Synonym.

One may also observe that the generated sentences are too similar in their wordings. As mentioned previously, this is due to the fact that the Quora dataset is not an ideal dataset for paraphrases, as the sentences of a pair may be almost identical.

Nevertheless, the sample results has demonstrated the learning ability of the conditional SeqGAN model, as some of the paraphrasing techniques are captured.

5.2. BLEU and METEOR Score Evaluation

The BLEU and METEOR scores are not the ideal metrics for paraphrases, as explained in Section 4.2. It is nonetheless presented for result reproduction. The scores from previous works are also presented for reference, as shown in Table 2. More specifically, BLEU-2 scores are measured, the same as in previous works.

There are two things to notice. The first one is that the scores are raised after the model has undertaken adversarial training, which demonstrated that the adversarial training process is guiding the model in the right direction. Secondly, the overall model is not tuned due to time limitation for the project, and the performance may be enhanced given more time and resources.

5.3. Training Process

There are three stages of improvements for our model in this project:

1) Discriminator based on Bidirectional GRU (Bi-GRU). Generator without attention, reward scaling, and interleaved training.
2) Discriminator based on CNN. Generator without attention, reward scaling, and interleaved training.
3) Final model.

The accuracy and loss trends of the three stages are presented in Figure 2 to demonstrate the enhancement achieved at each stage.

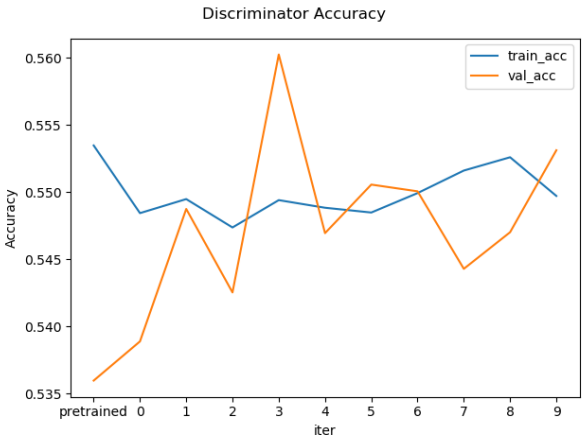
At stage 1, it is easily seen that the training was not successful, as the discriminator can only achieve an accuracy a little better than random guessing. Moreover, the loss trends are unstable. Without a strong discriminator, the rewards for the generator would be meaningless. The results indicates that Bi-GRU solely may not be effective for text matching.

After changing the discriminator model to CNN at stage 2, the accuracy of the discriminator increases significantly, and the loss trends are more stable. However, the discriminator is now overwhelmingly stronger than the generator, thus the vanishing gradient problem occurs for the generator, as discussed in Section 3.2. The reward scaling technique was added in stage 3 to alleviate the problem, along with other training techniques.

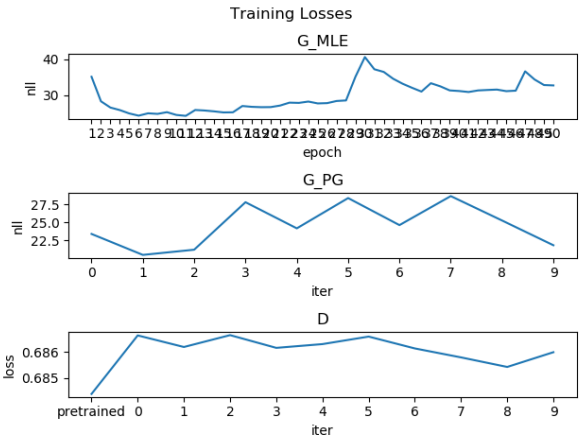
The final model at stage 3 behaves more reasonably. The discriminator is strong but also competing with the generator, as its accuracy is constantly brought down by the generator. The loss trends are less stable than in stage 2, indicating that a healthy competition is taking place between the discriminator and the generator.

6. Discussion

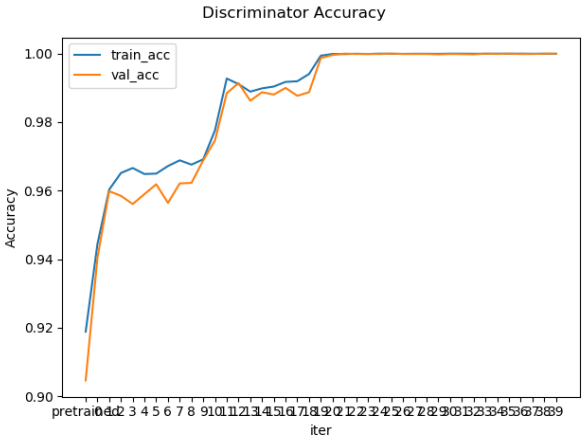
Observations. Several observations are made in this project. Firstly, the discriminator should be focussed on first when training a GAN model so that it can provide useful feedback to the generator. Secondly, CNN is demonstrated to be a suitable model for text matching problems. Lastly, the conditional SeqGAN model has the added



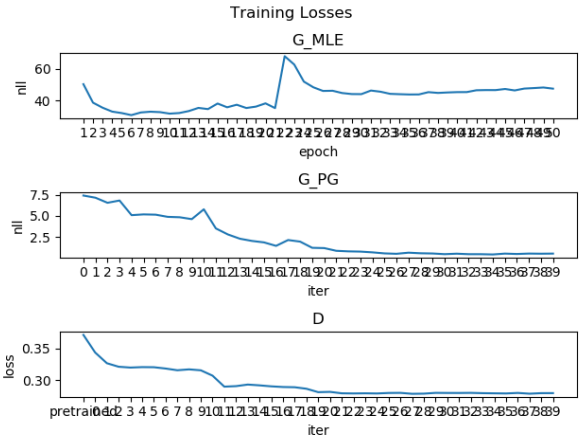
(a) Stage 1 accuracy trend.



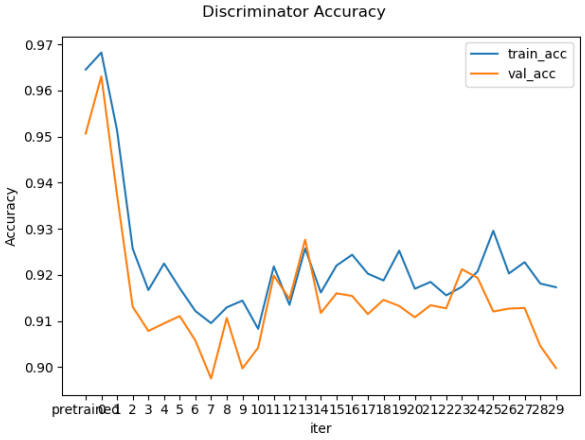
(b) Stage 1 loss trends.



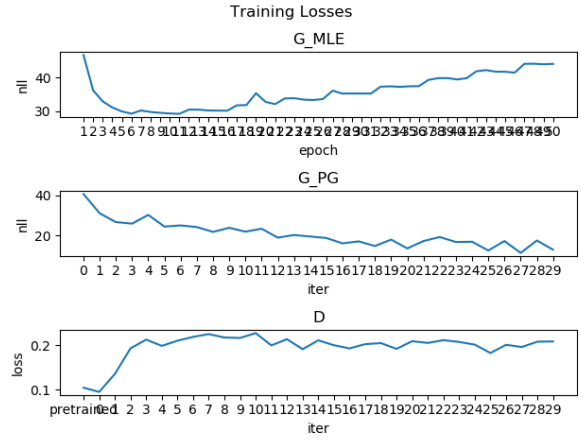
(c) Stage 2 accuracy trend.



(d) Stage 2 loss trends.



(e) Stage 3 accuracy trend.



(f) Stage 3 loss trends.

Figure 2: Accuracy and loss trends of the three stages of development, including discriminator accuracy, generator loss in pretraining (G_MLE), generator loss in adversarial training (G_PG), and discriminator loss in adversarial training (D).

Table 1: Sample results.

1)	Original	what are the safety precautions on handling shotguns proposed by the nra in north carolina ?
	Reference	what are the safety precautions on handling shotguns proposed by the nra in vermont ?
	Generated	**what are the safety precautions proposed by the nra precautions on handling shotguns proposed**
2)	Original	how do i improve professional email writing skills ?
	Reference	how do i improve my email writing skills ?
	Generated	**how can i improve my writing skills ?**
3)	Original	how do i get rid of my belly fat ?
	Reference	what should i do for belly fat ?
	Generated	**how can i i lose fat ? how do i get rid of weight ?**
4)	Original	what is quickbooks tech support number in arizona ?
	Reference	what is the quickbooks customer support phone number usa ?
	Generated	**what is the quickbooks softwares support phone number ?**

Table 2: BLEU and METEOR Scores.

	BLEU-2	**METEOR**
VAE-SVG-eq [5]	17.3	22.2
RbM-IRL [6]	***34.79**	***26.67**
Ours (after pretraining)	9.14	13.21
Ours (after adversarial training)	14.79	15.85

complexity of condition matching and variable-length input and outputs, and the results may serve as a reference performance for SeqGAN on NLP tasks that are more challenging than the language modeling task in the original SeqGAN model.

Limitations. There are multiple limitations throughout the project that restricts the performance of the generated results and evaluation process. First of all, the Quora dataset is not ideal for paraphrase generation, as the sentences of a pair are often too similar to each other, hence the generated paraphrases will also be alike. In addition, the objective function of our model does not penalize sentence pairs that are too similar, which can be taken into consideration in future improvements. Finally, the hyper-parameters were carefully selected to fit the resources available; increasing the dimensions is likely to boost the performance of our model.

Possible future improvements. A set of possible improvement techniques that is not applied due to time restriction is listed below:

1) Train for more iterations.
2) Increase embedding and hidden dimensions, batch size, etc.
3) Add beam search.
4) Apply a more intricate objective function.

7. Conclusions and Future Works

In this paper, a conditional SeqGAN model is built. The performance of the model is slightly restricted due to several limitations, yet the overall performance has demonstrated its capability as a paraphrase generation model.

Several training strategies are suggested. The discriminator should be improved first to make sure it actually has the ability to differentiate. The choice of the discriminator model should be taken care of, since it may make a lot of difference.

In this project, most of the necessary testing tools were created, and the environments were properly set up, which make the future development easier to be focused on improving the model structure itself.

In the future, the possible improvement techniques provided in Section 6 can be applied to enhance the performance. Human evaluations should be conducted for better assessment of the model performance. Also, ablation study can be conducted on the various training techniques applied in the current model for understanding which one is most helpful in different domains.

Furthermore, when the generator performs reasonably well, the discriminator can be used for paraphrase detection. Finally, it may be useful to design a more suitable evaluation metric for paraphrases based on semantic matching and output variability.

References

[1] Merriam-Webster.com. (2011) Paraphrase. [Online]. Available: https://www.merriam-webster.com/dictionary/paraphrase

[2] K. R. McKeown, "Paraphrasing questions using given and new information," *Computational Linguistics*, vol. 9, no. 1, pp. 1–10, 1983.

[3] S. Narayan, S. Reddy, and S. B. Cohen, "Paraphrase generation from latent-variable pcfgs for semantic parsing," *arXiv preprint arXiv:1601.06068*, 2016.

[4] S. Zhao, C. Niu, M. Zhou, T. Liu, and S. Li, "Combining multiple resources to improve smt-based paraphrasing model," *Proceedings of ACL-08: HLT*, pp. 1021–1029, 2008.

[5] A. Gupta, A. Agarwal, P. Singh, and P. Rai, "A deep generative framework for paraphrase generation," *arXiv preprint arXiv:1709.05074*, 2017.

[6] Z. Li, X. Jiang, L. Shang, and H. Li, "Paraphrase generation with deep reinforcement learning," *arXiv preprint arXiv:1711.00279*, 2017.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[8] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[9] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 700–708.

[10] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint*, 2017.

[11] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," *CoRR, abs/1703.07511*, vol. 2, 2017.

[12] I. Goodfellow. (2016) Generative adversarial networks for text. [Online]. Available: https://www.reddit.com/r/MachineLearning/comments/40ldq6/generative_adversarial_networks_for_text/

[13] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient." in *AAAI*, 2017, pp. 2852–2858.

[14] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[15] Quora. (2017) First quora dataset release: Question pairs. [Online]. Available: https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

[16] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[17] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[18] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[19] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[22] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[23] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[24] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang, "Long text generation via adversarial training with leaked information," *arXiv preprint arXiv:1709.08624*, 2017.

[25] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition." in *AAAI*, 2016, pp. 2793–2799.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[27] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[28] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.