A Learning-Exploring Method to Generate Diverse Paraphrases with Multi-Objective Deep Reinforcement Learning

Mingtong Liu[1] *, **Erguang Yang**[1], **Deyi Xiong**[2], **Yujie Zhang**[1] †,
Yao Meng[3], **Changjian Hu**[3], **Jinan Xu**[1], **Yufeng Chen**[1]
[1] School of Computer Science and Information Technology && Beijing Key Lab
of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing, China
[2] College of Intelligence and Computing, Tianjin University, Tianjin, China
[3] Lenovo Research AI Lab, Beijing, China
{mingtongliu, 19112037, yjzhang, jaxu, chenyf}@bjtu.edu.cn,
dyxiong@tju.edu.cn, {mengyao1, hucj1}@lenovo.com

Abstract

Paraphrase generation (PG) is of great importance to many downstream tasks in natural language processing. Diversity is an essential nature to PG for enhancing generalization capability and robustness of downstream applications. Recently, neural sequence-to-sequence (Seq2Seq) models have shown promising results in PG. However, traditional model training for PG focuses on optimizing model prediction against single reference and employs cross-entropy loss, which objective is unable to encourage model to generate diverse paraphrases. In this work, we present a novel approach with multi-objective learning to PG. We propose a learning-exploring method to generate sentences as learning objectives from the learned data distribution, and employ reinforcement learning to combine these new learning objectives for model training. We first design a sample-based algorithm to explore diverse sentences. Then we introduce several reward functions to evaluate the sampled sentences as learning signals in terms of expressive diversity and semantic fidelity, aiming to generate diverse and high-quality paraphrases. To effectively optimize model performance satisfying different evaluating aspects, we use a GradNorm-based algorithm that automatically balances these training objectives. Experiments and analyses on Quora and Twitter datasets demonstrate that our proposed method not only gains a significant increase in diversity but also improves generation quality over several state-of-the-art baselines.

1 Introduction

Paraphrase generation (PG) creates different expressions that share the same meaning (e.g., “*how far is Earth from Sun*” and “*what is the distance between Sun and Earth*”). It is a crucial technology in many downstream natural language processing (NLP) applications such as question answering (Dong et al., 2017), machine translation (Zhou et al., 2019), and text summarization (Zhao et al., 2018).

Diversity is an essential characteristic of human language, as the meaning of a text can often have multiple different expressions. A good paraphrase generation system is often required to conform to two desired properties (Xu et al., 2018b). The first is *diversity*, capturing a wide range of linguistic variations. The second is *fidelity*, preserving semantic meanings while paraphrasing. Therefore, we hope to generate diverse paraphrases while ensuring same meaning, which is important for enhancing generalization capability and robustness of downstream applications (Iyyer et al., 2018). As shown in Table 1, we give some examples. These examples express the same meaning but with different diversities.

Most recent state-of-the-art approaches to PG (Prakash et al., 2016; Hasan et al., 2016; Gupta et al., 2018) employ neural sequence-to-sequence (Seq2Seq) models, which mainly uses one given reference for model learning, while the nature of paraphrasing indicates that we can paraphrase one sentence into several different sentences. Meanwhile these methods usually adopt the cross-entropy loss which requires a strict pairwise matching at the word level between the predicted sentence and the ground truth sentence. It lacks flexibility and may penalize the generation model for a diverse paraphrase even if the sentence retains the meaning. For example, considering one model prediction “I watched a *movie* last

* Contribution during internship at Lenovo Research AI Lab.

† Corresponding author.

Proceedings of the 28th International Conference on Computational Linguistics, pages 2310–2321
Barcelona, Spain (Online), December 8-13, 2020

Original Sentence	how can i improve my English language?
Paraphrase-A	how can i improve my English pronunciation?
Paraphrase-B	how can i increase my knowledge in English language?
Paraphrase-C	there is a possible way to improve my English language?
Paraphrase-D	what is the best way to increase my English knowledge?

Table 1: Paraphrases of an original sentence with increasing diversity.

night." and the reference "I saw a *film* last night.", the cross-entropy loss lacks the ability to properly optimize model to generate a diverse paraphrase even with only one changed token at word level.

In recent years, there are also growing interests in generating lexically and syntactically diverse paraphrases (Gupta et al., 2018; Xu et al., 2018b; Xu et al., 2018a; Park et al., 2019; Qian et al., 2019; Kajiwara, 2019). For Seq2Seq models, the techniques of generating diverse paraphrases mainly include two categories: i) applying decoding methods such as using beam search or multiple decoders; ii) introducing random noise as model input. Park et al. (2019) use multi-time decoding to diverse generation by considering those generated sentences previously. Qian et al. (2019) use multiple generators to generate a variety of different paraphrases. Gupta et al. (2018) employ a variational auto-encoder framework to produce multiple paraphrases according to different noise inputs. Although these methods can improve paraphrase generation with different decodings or noise inputs, their model training still lacks the ability of directly exploring diverse paraphrases as learning objectives.

In order to address these problems, in this work, we propose a novel learning-exploring method to generate paraphrases with multi-objective deep reinforcement learning. Our method makes it possible for every objective to focus on different aspects of generated paraphrases, breaking the restriction of learning with only one target sentence given by data in supervised learning. Concretely, we first train a paraphrase generation model with cross-entropy loss. Then we design a sample-based exploring algorithm to generate multiple candidate paraphrases from the learned data distribution, and utilize the explored sentences to train the generation model with deep reinforcement learning. Therefore, the model can be effectively trained in a learning-exploring fashion, to find more diverse paraphrases.

In particular, we use the variations between the generated paraphrase and the original input as learning signal of diversity. To ensure expressive diversity and semantic fidelity simultaneously, we design two rewards and one for each aspect. One reward is to distinguish whether the explored sentence has diverse expression, the other one is to examine whether the explored sentence conveys the same meaning with original input. The higher the confidence of one sentence is to be judged as a good paraphrase explored by the sample algorithm, that will be used for better model training. We also use a reward evaluated with reference to train model. Finally, we combine these rewards evaluated from different aspects as guiding signals to train the generation model via reinforcement learning. Furthermore, in order to effectively combine these rewards that optimize different aspects of a generated paraphrase, we additionally use a GradNorm-based algorithm (Chen et al., 2017) that automatically balances these training objectives.

In summary, our contributions are as follows:

- We propose a novel learning-exploring framework to learn to generate diverse paraphrases with multi-objective deep reinforcement learning.

- In order to enable the model to learn to generate diverse paraphrases, we propose to equip the model with several vital components: (1) sample-based exploring algorithm to generate diverse candidate paraphrases; (2) multiple reward functions for evaluating sampled sentences to ensure expressive diversity and semantic fidelity simultaneously; (3) GradNorm-based algorithm that automatically balances training objectives for effective learning.

- We conduct experiments on two standard benchmark datasets Quora and Twitter. Empirically, experimental results show that our new learning methods with reinforcement learning perform significantly better than several state-of-the-art baselines in term of diversity and quality.

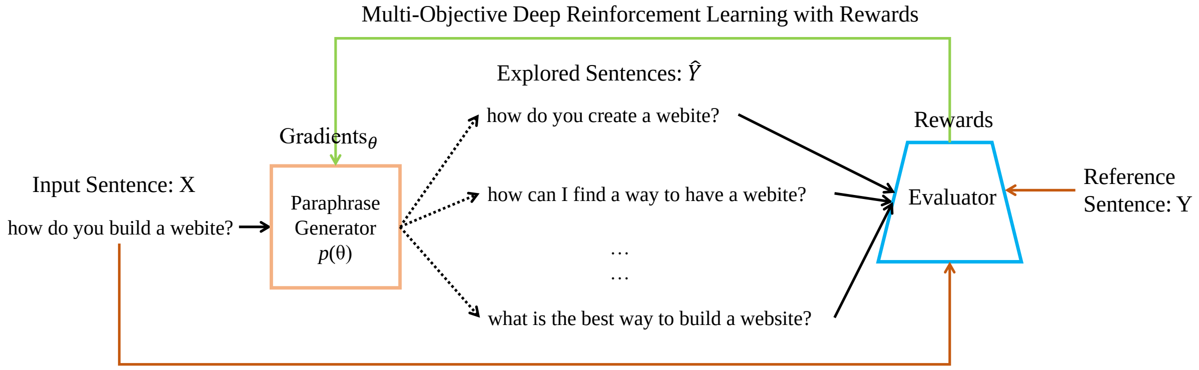
Multi-Objective Deep Reinforcement Learning with Rewards
Explored Sentences: Ŷ
Gradients$_\theta$
Input Sentence: X
how do you build a webite?
Paraphrase Generator p(θ)
how do you create a webite?
how can I find a way to have a webite?
…
…
what is the best way to build a website?
Rewards
Evaluator
Reference Sentence: Y

Figure 1: The proposed learning-exploring paraphrase generation framework with multi-objective deep reinforcement learning.

2 The Proposed Model

In this section, we elaborate our proposed model, including its essential components and their working mechanisms.

2.1 Problem and Framework

Given an input sentence $X = [x_1, x_2, \cdots, x_S]$ with length S, we aim to generate an output paraphrase Y = $[y_1, y_2, \cdots, y_T]$ with length T that has the same meaning but a different expression with X. We denote the pair of sentences in paraphrasing as $(X; Y)$. We use $Y_{1:t}$ to denote the subsequence of Y ranging from 1 to t and use $\hat{Y}$ to denote the sequence generated by the model.

Our proposed model mainly contains three components: paraphrase generator, sample-based exploring algorithm and reinforcement learning with explored paraphrase. Figure 1 gives an overview of our framework. Basically the generator can generate paraphrases of a given sentence, and the evaluator measures the quality of explored paraphrases in term of expressive diversity and semantic fidelity.

2.2 Paraphrase Generator

We frame paraphrase generation as a sequence-to-sequence (Seq2Seq) problem. We adopt the encoder-decoder framework (Bahdanau et al., 2014; See et al., 2017), both of which are implemented as recurrent neural networks (RNN). All RNNs use LSTM cells (Hochreiter and Schmidhuber, 1997). Given an input sentence X, the goal is to learn a model $p(\theta)$ that can generate a sentence $\hat{Y} = p_\theta(X)$ as its paraphrase. Traditionally the parameters θ are learned by maximizing the likelihood of the predicted sentence. Finally, model estimates the conditional probability $p(Y|X)$ via directly mapping the input sentence X to its target paraphrase Y. The learning objective is to minimize the cross-entropy loss:

$$L_{ce}(\theta) = -\sum_{t=1}^{T} \log p_\theta(y_t|Y_{1:t-1}, X) \tag{1}$$

We choose the pointer-generator (See et al., 2017) as our paraphrase generator. In pointer-generator, the decoder allows either generating words from a vocabulary or copying words from the input sentence, which alleviates the out-of-vocabulary problem (such as named entities) and improves the performance. The probability of copying words can be represented as:

$$p_{copy}(y_t|Y_{1:t-1}, X) = a_{ti}, \qquad a_{ti} = \frac{exp(f(s_{t-1}; h_i))}{\Sigma_i^S exp(f(s_{t-1}; h_i))} \tag{2}$$

where s_{t-1} is the decoder state at time step t - 1, h_i is the encoder hidden state at time step i. a_{ti} represents the attention weight at time step t and p_{copy} is actually a_{ti}. f is the attention function, which is a feed-forward neural network.

The probability of generating the next word from vocabulary is given by:

$$p_{vocab}(y_t|Y_{1:t-1}, X) = softmax(o_t), \qquad o_t = g(s_{t-1}, y_{t-1}, c_t), \qquad c_t = \Sigma_{i=1}^{S} a_{ti} h_i \tag{3}$$

where y_{t-1} is the word generated last step, c_t means the context vector computed by attention mechanism at time step t, g is a linear function. The overall probability of the next word is:

$$p(y_t|Y_{1:t-1}, X) = p_{gen} * p_{vocab} + (1 - p_{gen}) * p_{copy}, \qquad p_{gen} = m(s_t, y_{t-1}, c_t) \tag{4}$$

where $m(s_t, y_{t-1}, c_t)$ is a binary neural classifier with sigmoid activation output. p_{gen} acts as a gate to control whether the next word is generated from vocabulary or is copied from the input sentence.

Specifically, model predicts the next word at each decoding time step by sampling from probability distribution $\hat{y}_t \sim p_(y_t|Y_{1:t-1}, X)$. We will use the sampled sentences to train our model.

2.3 Sample-Based Exploring Algorithm

Our learning-exploring method uses explored sentences for model training. It is critical to set up an appropriate sample algorithm to acquire diverse paraphrases. To better sample more diverse paraphrases, we adopt Gumbel-Softmax technique (Jang et al., 2016), which injects noise to adjust the word distribution, enabling us to sample diverse sentences from the model's approximation of the data distribution.

We modify the probability distribution in Equation 3 by shaping the distribution through Gumbel noise. The Gumbel noise, treated as a form of regularization, is added to o_t in Equation 3, then softmax function is performed. The word distribution of y_t is approximated by:

$$p_{vocab}(y_t|Y_{1:t-1}, X) = softmax(\tilde{o}_t), \qquad \tilde{o}_t = (o_t + \eta)/\tau, \qquad \eta = -\log(-\log u) \tag{5}$$

where η is the Gumbel noise calculated from a uniform random variable $u \sim U(0, 1)$, τ is temperature. When $\tau \rightarrow 0$, the sample generated from the vocabulary is similar to the argmax operation, and when $\tau \rightarrow \infty$, the sample is gradually closing uniform distribution. Increasing the temperature increases the use of infrequent words (Holtzman et al., 2019), which has the implicit effect of weakening the tail distribution, making the model to explore more diverse generation.

Finally, according to $p_{vocab}(y_t|Y_{1:t-1}, X)$, we apply multinomial sampling (Chatterjee and Cancedda, 2010) to generate sentence $\hat{Y}$ for computing rewards, which produces each word one by one through multinomial sampling over the model's output distribution. The sampling terminate the expansion of a candidate sentence when an end of sentence (<EOS>) token is met.

2.4 Reinforcement Learning with Explored Paraphrase

We adopt reinforcement learning (RL) (Sutton and Barto, 1998) to train our paraphrase generator by using the sampled sentences. Our paraphrase generator can be viewed as an "agent" that interacts with an external "environment" (original input or reference). The parameters of the agent define a policy, i.e., a conditional probability $p(y_t|Y_{1:t-1}, X)$. The agent will pick an action, i.e., the prediction of the next candidate word, according to the policy. When generating the EOS token, the agent observes a terminal reward for evaluating the generated sentence $\hat{Y}$. The reward is denoted as $R(\hat{Y}, Y^i)$, where Y^i represents a comparable sentence. The goal of the RL training is to minimize the negative expected reward:

$$L_{rl}(\theta) = -\mathbb{E}_{\hat{Y}^s \sim p_\theta}[R(\hat{Y}^s, Y^i)] \tag{6}$$

where $\hat{Y}^s = (\hat{y}_1^s, \hat{y}_2^s, \cdots, \hat{y}_t^s)$ is a sampled sentence and $\hat{y}_t^s$ is the word sampled from the model at the time step t. $\hat{Y}$ is the space of all candidate paraphrase sentences, which is exponentially large due to the large vocabulary size, making it impossible to exactly optimize L_{rl}. In practice, we adopt REINFORCE-based policy gradient approach (Williams, 1998) with a single sample from p_θ.

$$L_{rl}(\theta) \approx -R(\hat{Y}^s, Y^i), \qquad \hat{Y}^s \sim p_\theta \tag{7}$$

In order to reduce the variance of policy gradient method, a typical technique is subtracting baseline values from the original rewards. We use the self-critical algorithm (Rennie et al., 2017), in which the baseline is the reward of sentences generated in inference. Finally, the expected gradient based on REINFORCE of a non-differentiable reward function can be computed as follows:

$$\nabla_\theta L_{rl}(\theta) \approx -(R(\hat{Y}^s, Y^i) - b)\nabla_\theta \log p_\theta(\hat{Y}^s), \qquad b = R(\hat{Y}^b, Y^i) \tag{8}$$

where $p_\theta(\hat{Y}^s) = \prod_{t=1}^{T} p(\hat{y}_t^s|\hat{Y}_{1:t-1}^s, X)$ is the probability for generating sentence $\hat{Y}^s$ given X. $\hat{Y}^s$ is sampled from the output probability distribution, $\hat{y}_t^s \sim p(y_t|\hat{Y}_{1:t-1}^s, X)$. $\hat{Y}^b$ is the baseline output using the test time inference algorithm greedy search. We can see that minimizing $L_{rl}(\theta)$ is equivalent to maximizing the conditional likelihood of the sampled sentence $\hat{Y}^s$ if the generated sentence $\hat{Y}^s$ outperforms $\hat{Y}^b$, thus giving positive signals. As a result, reward will be increased as the generator increases the generation probability of better sentences while decreasing the chance of worse sentence generation.

3 Multi-Objective Learning

This section explains how to learn the generator using multi-objective deep reinforcement learning. Our training algorithm allows model to explore the space of possible paraphrases, making our model to generate more diverse paraphrases and also enhance model performance. As a result, multi-objective reinforcement learning helps paraphrase generation in two ways: (a) it directly optimizes the evaluation metric instead of maximizing the likelihood of the ground-truth reference and (b) it makes our model has the ability to explore unseen paraphrases beyond one single reference.

3.1 Rewards for Multi-Objective Learning

ROUGE Reward with Reference The first basic reward is based on the primary evaluation metric of ROUGE package (Lin, 2004). We compare a sampled sentence $\hat{Y}^{s_1}$ with ground-truth reference Y^{ref} with ROUGE score (namely ROUGE-ref), and then takes the score as a reward. The loss function is given by:

$$\nabla_\theta L_{rl_1}(\theta) \approx -(\text{ROUGE–ref}(\hat{Y}^{s_1}, Y^{ref}) - b)\nabla_\theta \log p_\theta(\hat{Y}^{s_1}), \qquad b = \text{ROUGE–ref}(\hat{Y}^b, Y^{ref}) \tag{9}$$

Similar to previous work (Li et al., 2018), we find that ROUGE-ref score as a reward works better compared to only using cross-entropy loss. This reward can be taken as sentence-level learning signal, which overcomes the full token-level matching issue of cross-entropy loss at training stage.

On the other hand, as pointed in Kajiwara (2019), paraphrase generation rewrites only a limited portion of an original input and the reference often includes some words occurred in the original input, thus a sentence with higher ROUGE-ref score may have low diversity (Miao et al., 2019). Therefore, the reward based on reference do not focus on the variations between the sampled sentence and the original input. Addressing these issues, we next introduce two new reward functions.

ROUGE Reward with Input To obtain more diverse paraphrases, we introduce the second reward function. We compare a sampled sentence $\hat{Y}^{s_2}$ with the original input Y^{ori} by computing ROUGE score (namely ROUGE-ori), which can focus on the word variations between the sampled sentence and the original input. We use the negative ROUGE score as reward, in which the lower word overlap, the better variation. The score reflects model ability to produce diverse paraphrases. The loss function is given by:

$$\nabla_\theta L_{rl_2}(\theta) \approx -(b - \text{ROUGE–ori}(\hat{Y}^{s_2}, Y^{ori}))\nabla_\theta \log p_\theta(\hat{Y}^{s_2}), \quad b = \text{ROUGE–ori}(\hat{Y}^b, Y^{ori}) \tag{10}$$

Semantic Similarity Reward with Input The explored paraphrases may have distant semantic similarity with original input, which may hurts fidelity, preserving semantic meanings while paraphrasing. We further introduce a semantic similarity reward to ensure the semantic accuracy of explored sentences. We adopt embedding-based method (Sharma et al., 2017) for computing semantic similarity score, which do not increase extra learnable parameters. We use Greed Matching (GM) to compute the semantic similarity score between a sampled sentence $\hat{Y}^{s_3}$ and the original input Y^{ori}. The computation of semantic similarity score is as follow:

$$G(C,r) = \frac{\sum w \in C \max_{\hat{w}\in r} cos\_sim(e_w, e_{\hat{w}})}{Length(C)}, GM(\hat{Y}^{s_3}, Y^{ori}) = \frac{G(\hat{Y}^{s_3}, Y^{ori}) + G(Y^{ori}, \hat{Y}^{s_3})}{2} \tag{11}$$

Each word in the candidate sentence C is greedily matched to a word in the reference sentence r based on their embeddings' cosine similarity. The score is an average of these similarities over the number of words in the candidate sentence.

Then we take the semantic similarity score compared with original input (namely SEM-ori) as a reward, and the loss function is given by:

$$\nabla_\theta L_{rl_3}(\theta) \approx -(\text{SEM–ori}(\hat{Y}^{s_3}, Y^{ori}) - b)\nabla_\theta \log p_\theta(\hat{Y}^{s_3}), \quad b = \text{SEM–ori}(\hat{Y}^{b}, Y^{ori}) \tag{12}$$

3.2 Multi-Objective Optimization

Our objective function combines the maximum-likelihood cross-entropy loss (L_{ce}) with rewards from policy gradient reinforcement learning to jointly optimize our model. Finally, the over learning objective is to minimize the following combined losses:

$$L_{all}(\theta) = \alpha_0 * L_{ce} + \alpha_1 * L_{rl_1} + \alpha_2 * L_{rl_2} + \alpha_3 * L_{rl_3} \tag{13}$$

where α is the the weights to combine these losses.

Optimizing multiple objectives at the same time is important for final performance, in which one objective can easily dominate the learning of a shared model, leading the other objectives are ineffective. Previous work (Wu et al., 2018) choose fixed weights α from manual experience for RL learning. Different from them, we use an adaptive method GradNorm (Chen et al., 2017), and the α_i is vary at each training step t: $\alpha_i = \alpha_i(t)$. The GradNorm algorithm controls gradient magnitudes through tuning of the multi-objective loss function. To optimize the weights $\alpha_i(t)$ for gradient balancing, following Chen et al. (2017), we penalize the network when back-propagated gradients from any task are too large or too small. If objective i is training relatively quickly, then its weight $\alpha_i(t)$ should decrease relative to other objective weights to allow other objectives more influence on training.

4 Experiment

In this section, we described the datasets, experimental setup, evaluation metrics and the results of our experiments.

4.1 Datasets

We conducted experiments on two standard datasets Quora and Twitter to evaluate the proposed model.

Quora Dataset This dataset is a paired paraphrase dataset in question domain. It consists of 150K paraphrase pairs. Following previous work (Li et al., 2018; Qian et al., 2019), we used 30K pairs and 4K pairs as test set and validation set, and 100K pairs for training, respectively.

Twitter Dataset This dataset is Twitter URL paraphrasing corpus (Lan et al., 2017) that contains two subsets, one is labelled by human annotators while the other is labelled automatically by algorithm. Following previous work (Li et al., 2018; Qian et al., 2019), we sampled 5K pairs as the test set and 1K pairs as validation set from the labeled subset, while using the remaining 110K pairs as training set.

4.2 Model Configuration

We used the following experimental setting for our model. Following Li et al. (2018), we maintained a fixed-size vocabulary of 5K shared by the words in input and output, and truncate all the sentences longer than 20 words. For paraphrase generators, we used two-layer LSTM for both encoder and decoder. The hidden dimension of encoder and decoder was set to 256, and the word embedding dimension was 128. We adopted ROUGE-1 score for computing rewards.

In training, we used Adam optimizer (Kingma and Ba, 2014) with β_1 = 0.9 and β_2 = 0.98 for optimization. We pre-trained model 10 epoch with cross-entropy loss with a learning rate of 1e-3 and then started RL training. In reinforcement learning, the learning rate decreased to 1e-4. The mini-batch size was fixed at 64. We set τ = 0.3 for Gumbel-softmax sampling. For GradNorm-based (Chen et al., 2017) multi-objective optimization, the initial wight was 1 for all losses, and we used the parameters in the linear layer before softmax to automatically adjust weight gradients. At test time, we used beam search of width 5 on all our models to generate our final predictions.

Model	**B-2↑**	**R-1↑**	**R-2↑**	**MET↑**	**Emb(A/E/G)↑**	**B-ori-1↓**
Previous Work						
Residual LSTM	37.38	59.21	32.43	28.17	-	-
VAE-SVG-eq	-	-	-	24.70	-	-
EDD-LG (shared)	32.40	44.90	-	23.10	-	-
TRANS	37.46	-	-	36.04	80.61/-/64.81	-
TRANSEQ	38.75	-	-	35.84	81.50/-/65.52	-
RbM-SL	43.54	64.39	38.11	32.84	-	-
RbM-IRL	43.09	64.02	37.72	31.97	-	-
DEPD	36.84	-	-	29.28	-	-
DPNG	-	63.73	37.75	-	-	-
This Work						
Seq2SeqAttn	40.20	59.27	32.00	30.49	92.58/71.94/82.63	60.57
PNet	42.89	61.81	35.27	34.21	93.35/75.22/92.11	67.91
PNet + LE-ROUGE-ref	44.99	63.84	36.97	34.78	93.73/76.10/92.12	70.54
PNet + LE-ROUGE-ori	42.82	59.96	33.93	33.32	93.10/74.13/90.61	66.63
PNet + LE-SEM-ori	45.01	63.08	36.61	34.83	93.81/76.34/92.12	71.06
PNet + all	**44.63**	**64.66**	**38.22**	**35.02**	**93.74/76.33/92.31**	**67.60**

Table 2: Performance on Quora dataset.

4.3 Automatic Evaluation Metrics

Following previous work (Prakash et al., 2016; Hasan et al., 2016) on paraphrase generation, we adopted well-known automatic evaluation metrics BLEU (B) (Papineni et al., 2002), ROUGE (R) (Lin, 2004) and METEOR (MET) (Lavie and Agarwal, 2007) to compute *lexical similarity* with reference. Pervious studies have shown that these metrics perform well in evaluating generated paraphrases.

These n-gram-based matching may obtain low score for predictions with highly lexical and syntactical variation, but these predictions are not necessarily poor quality (Chen and Dolan, 2011; Wang et al., 2019). We further used *Embedding Similarity* (Sharma et al., 2017) to evaluate generated paraphrases. This metric measures the semantic similarity between the reference and prediction based on the cosine similarity of their embeddings on word and sentence level. Following previous work (Park et al., 2019; Egonmwan and Chali, 2019), we used average, extreme, and greedy (A/E/G) embedding similarities.

Besides, we hope to generate more diverse paraphrases when preserving meaning. However, previous work (Miao et al., 2019) has shown that it is insufficient when only comparing with reference because simply copying the input sentence itself yields the highest BLEU-ref score. To evaluate the variation of generated paraphrases, following Miao et al. (2019), we used BLEU-ori (B-ori) metric that against the original input sentence, in which the lower n-gram overlaps, the better variation and diversity.

4.4 Baselines

We compared our model with several state-of-the-art models in the paraphrase generation field.

- Residual LSTM (Prakash et al., 2016): This implements stacked residual LSTM networks.
- VAE-SVG-eq (Gupta et al., 2018): This employs a variational autoencoder as its main component.
- EDD-LG (Patro et al., 2018): This introduces semantic discriminator to learn encoder and decoder.
- TRANS and TRANSSEQ (Egonmwan and Chali, 2019): This integrates Transformer model (Vaswani et al., 2017) and Recurrent Neural Network GRU (Cho et al., 2014) as encoder.
- RbM-SL and RbM-IRL (Li et al., 2018): This is a generator-evaluator framework with the matching-based semantic evaluator trained by reinforcement learning.
- DEPD (Qian et al., 2019): This uses multiple generators trained by reinforcement learning to generate a variety of different paraphrases.

Model	B-2↑	R-1↑	R-2↑	MET↑	Emb(A/E/G)↑	B-ori-1↓
Previous Work						
Residual LSTM	33.90	32.50	16.86	13.65	-	-
RbM-SL	44.67	41.87	24.23	19.97	-	-
RbM-IRL	45.74	42.15	24.73	20.18	-	-
DEPD	34.23	-	-	24.29	-	-
This Work						
Seq2SeqAttn	30.86	40.65	28.35	20.68	82.91/53.71/87.36	48.31
PNet	31.10	43.21	28.94	23.91	84.04/54.66/90.16	52.87
PNet + LE-ROUGE-ref	33.45	45.89	31.20	25.60	84.92/56.86/92.28	57.16
PNet + LE-ROUGE-ori	28.48	39.67	26.30	21.93	81.52/51.43/88.42	46.39
PNet + LE-SEM-ori	28.64	40.36	26.43	21.87	82.68/52.37/90.81	49.19
PNet + all	**32.84**	**45.51**	**30.61**	**25.20**	**85.57/57.46/93.81**	**53.21**

Table 3: Performance on Twitter dataset.

- DPNG (Li et al., 2019): This is a Transformer-based model that can learn and generate paraphrases of a sentence at different levels of granularity (word or phrase) in a disentangled way.

4.5 Results

Baseline Cross-Entropy Model Results Our paraphrase generation model has attention mechanism (Seq2SeqAtt) and pointer-generator network (PNet). For better observing model behaviour, we first trained two baselines by applying cross-entropy optimization. As we can see, the model with pointer-generator network effectively improves performance in all metrics related to reference. And it is also natural for PNet to obtain higher scores than Seq2SeqAtt in BLEU-ori-1 metric, as copy mechanism directly copes words from input to resolve UNK and entity word generation problem.

Multi-Objective Model Results Results of automatic evaluation on Quora are shown in Table 2. First, we can see that using ROUGE-ref as reward in the proposed learning-exploring (LE) framework (LE-ROUGE-ref) improves the performance in metrics related to reference, as compared to the cross-entropy baseline. When using ROUGE-input as reward, we can see that the model obtains higher diversity score but lower BLEU and ROUGE score related to reference. A main reason is that every test case only has one reference sentence, which makes the word matching-based evaluation metric more difficult to measure the real quality of diverse paraphrases. But, the model obtains comparable METEOR and Embedding similarity score, since both of them consider synonym matching, and therefore they measure the generation quality more accurately. These results indicate that using ROUGE-input as reward in the learning-exploring framework can exactly improve diversity of paraphrase generation.

When using semantic similarity reward (LE-SEM-ori), we can see that the model obtains better performance than baseline PNet in valuation metrics related to reference, but low diversity score, since the learning objective does not consider variation in model training procedure. These results show that semantic similarity reward is effective to ensure the semantic fidelity, preserving meanings while paraphrasing.

Finally, when we combine all the learning objectives, we can obtain better performance on Quora than several state-of-the-art baselines in term of diversity and quality. Our model not only obtains better BLEU, ROUGE, METEOR score in traditional evaluation metrics, but also better performance on semantic similarity and diversity. These results demonstrate the effectiveness of our proposed learning-exploring method with multi-objective deep reinforcement learning for Quora paraphrase generation.

Table 3 shows scores on Twitter dataset. Finally, our model achieves the best ROUGE and METEOR score than all baselines. It also shows that the different learning objectives have the ability to focus on the different natures of generated paraphrases. These results on Twitter further demonstrate that our proposed multi-objective learning can improve paraphrase generation in a learning-exploring fashion.

Num	Original Question	Reference	Pointer-generator	Ours
1	how can i become top writer on quora, what should i care most in this process?	what should i do to become a top writer on quora in 2017?	how do i become a top writer on quora?	what should i do to become a top writer on quora?
2	if human population growth is gon na continue at high rates, is it better that the growth happens in & developing countries or in developed countries?	if human population growth is going to continue at high rates, is it better that the growth happens in developing countries or in developed countries?	if human population growth is slow in developing countries, is it better that the growth in developing countries or in developed countries?	if the whole population is going to take at high frequency, is it better that the increase occurs in developed countries or in develing countries?
3	how should i edit my question correctly if quora marks down my question for improvement?	why is it that every time i ask a question in quora it tells me that your question needs improvement?	what should i do when someone marks my question for improvement?	what happens to a question on quora if it is marked as needing further improvement?

Table 4: Examples of generating paraphrases from pointer-generator and our model on Quora dataset.

4.6 Case Study and Discussion

Table 4 displays several generated examples by pointer-generator and our model. We can see that the proposed model produces fairly good samples in terms of both closeness in meaning and diversity in expression, because the model is encouraged to output better paraphrases during the learning phase.

In these examples, although pointer-generator indeed generates different sentences, it only yields little diversity. Compared to sentences generated by pointer-generator, those produced by our model have obvious variations compared to original input. In the first example, our model has better meaning preservation. In the second example, pointer-generator generates similar sentence and yields obvious meaning changes from the input sentence. In the third example, the sentence generated by pointer-generator show minor variations, while this generated by our model presents richer expressions.

5 Related Work

Neural paraphrase generation is often formalized as a sequence-to-sequence (Seq2Seq) learning formalism. Prakash et al. (2016) employ a stacked residual LSTM network in the Seq2Seq model to enlarge the model capacity. Hasan et al. (2016) incorporate the attention mechanism (Bahdanau et al., 2014) to generate paraphrases. Egonmwan and Chali (2019) integrate Transformer model (Vaswani et al., 2017) and Recurrent Neural Network GRU (Cho et al., 2014) to learn long-range dependencies in the input sequence. Li et al. (2018) propose a generator-evaluator architecture to reinforce the paraphrase generator by a reward function. Li et al. (2019) suppose a sentence-level paraphrase can be decomposed to word/phrase-level paraphrase and learn to generate paraphrases at different levels of granularity.

More recent works also focus on generating diverse paraphrases, which is important for improving model generalization capability and robustness of downstream applications. Gupta et al. (2018) use a variational autoencoder framework to generate diverse paraphrases by introducing random noise as input. Iyyer et al. (2018) harness syntactic-tree template information for controllable paraphrase generation. Chen et al. (2019) use sentences as exemplars to graft their syntax style to generated paraphrases. Qian et al. (2019) uses multiple generators trained by reinforcement learning to generate diverse paraphrases.

Similar to these works, we also adopt Seq2Seq model for paraphrase generation. However, significantly different from them, our work extend the Seq2Seq model to use explored paraphrases for model training via deep reinforcement learning. We further introduce evaluation metrics in terms of expressive diversity and semantic similarity for model learning. Finally, our model can effectively generate paraphrases by exploring unseen paraphrases beyond one single reference in a learning-exploring fashion.

6 Conclusion

In this work, we have presented a novel method to paraphrase generation in a learning-exploring fashion via multi-objective reinforcement learning. We designed sample-based exploring algorithm to acquire diverse paraphrases for model training, and used reinforcement learning with expressive diversity and semantic similarity rewards. Experiments and analyses on both Quora and Twitter datasets show that the proposed method can effectively learn to generate high-quality paraphrases and achieves better performance over several strong baselines. These results disclose that we can improve paraphrase generation by using explored sentences, breaking the restriction of single reference in supervised learning.

Acknowledgement

The research work descried in this paper has been supported by the National Nature Science Foundation of China (No. 61876198, 61976015, 61976016, 61370130 and 61976016), and also supported by the Beijing Municipal Natural Science Foundation (No. 4172047), and the International Science and Technology Cooperation Program of China (No. K11F100010). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Samidh Chatterjee and Nicola Cancedda. 2010. Minimum error rate training by sampling the translation lattice. In *Conference on Empirical Methods in Natural Language Processing*.

David L Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2017. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *CoRR*, abs/1711.02257.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886.

Elozino Egonmwan and Yllias Chali. 2019. Transformer and seq2seq model for paraphrase generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 249–255, Hong Kong, November. Association for Computational Linguistics.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Sadid A Hasan, Bo Liu, Joey Liu, Ashequl Qadir, Kathy Lee, Vivek Datla, Aaditya Prakash, and Oladimeji Farri. 2016. Neural clinical paraphrase generation with attention. In *Proceedings of the Clinical Natural Language Processing Workshop*, pages 42–53.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the NAACL*, pages 1875–1885.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark, September. Association for Computational Linguistics.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. Association for Computational Linguistics.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019. Decomposable neural paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3403–3414, Florence, Italy, July. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6834–6842.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc Meeting of the Association for Computational Linguistics*.

Sunghyun Park, Seung-won Hwang, Fuxiang Chen, Jaegul Choo, Jung-Woo Ha, Sunghun Kim, and Jinyeong Yim. 2019. Paraphrase diversification using counterfactual debiasing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6883–6891.

Badri Narayana Patro, Vinod Kumar Kurmi, Sandeep Kumar, and Vinay Namboodiri. 2018. Learning semantic sentence embeddings using sequential pair-wise discriminator. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2715–2729.

Aaditya Prakash, Sadid A Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual lstm networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2923–2934.

Lihua Qian, Lin Qiu, Weinan Zhang, Xin Jiang, and Yong Yu. 2019. Exploring diverse expressions for paraphrase generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3164–3173.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *Computing Research Repository*.

Richard S Sutton and Andrew G Barto. 1998. Reinforcement learning: An introduction.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. 2019. A task in a suit and a tie: paraphrase generation with semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7176–7183.

Ronald J. Williams. 1998. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018a. Dp-gan: Diversity-promoting generative adversarial network for generating informative and diversified text. *arXiv*, pages arXiv–1802.

Qiongkai Xu, Juyan Zhang, Lizhen Qu, Lexing Xie, and Richard Nock. 2018b. D-page: Diverse paraphrase generation. *arXiv: Computation and Language*.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173.

Zhong Zhou, Matthias Sperber, and Alexander Waibel. 2019. Paraphrases as foreign languages in multilingual neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, Florence, Italy, July. Association for Computational Linguistics.