Pushing Paraphrase Away from Original Sentence: A Multi-Round Paraphrase Generation Approach

Zhe Lin and **Xiaojun Wan**
Wangxuan Institute of Computer Technology, Peking University
Center for Data Science, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{linzhe,wanxiaojun}@pku.edu.cn

Abstract

In recent years, neural paraphrase generation based on Seq2Seq has achieved superior performance, however, the generated paraphrase still has the problem of lack of diversity. In this paper, we focus on improving the diversity between the generated paraphrase and the original sentence, i.e., making generated paraphrase different from the original sentence as much as possible. We propose **BTmPG** (**B**ack-**T**ranslation guided **m**ulti-round **P**araphrase **G**eneration), which leverages multi-round paraphrase generation to improve diversity and employs back-translation to preserve semantic information. We evaluate BTmPG on two benchmark datasets. Both automatic and human evaluation show BTmPG can improve the diversity of paraphrase while preserving the semantics of the original sentence.

1 Introduction

Paraphrase generation or sentence paraphrasing is an important task in natural language processing, and it requires rewriting a sentence while preserving its semantics. Paraphrase generation has been widely used in many downstream tasks such as QA systems, semantic parsing, dialogue systems and so on.

In recent years, deep learning techniques like sequence-to-sequence(Seq2Seq) have achieved superior performance on natural language generation tasks (Zhao et al., 2010; Wubben et al., 2010). Many paraphrase models based on Seq2Seq have achieved inspiring results. For example, Prakash et al. (2016) leveraged stacked residual LSTM networks to generate paraphrase, and Gupta et al. (2018) proposed a deep generative framework based on variational auto-encoder for paraphrase generation.

Though paraphrase generation models based on Seq2Seq have demonstrated advanced ability, the generated paraphrase still has the problem of lack of diversity, i.e., the output paraphrase only makes trivial changes to the original sentence. A good paraphrase of a sentence is one that is semantically similar to that sentence while being (very) syntactically and/or lexically different from it (Bhagat and Hovy, 2013). Paraphrase which is too similar to the original sentence is much less useful in many real applications.

In this paper, we focus on improving the diversity of generated paraphrase, i.e., making generated paraphrase different from the original sentence as much as possible. An intuitive but uninvestigated idea is to adopt multi-round paraphrase generation. Concretely, we first send the original sentence into a paraphrase generation model to generate a paraphrase, and then we use the generated paraphrase as the input of the model to generate a new paraphrase. As long as we leverage a paraphrase generation model with strong diversity like variational auto-encoder (VAE)(Kingma and Welling, 2013), we can get the paraphrase as different as possible from the original sentence after multi-round generation.

However, existing paraphrase models can not ensure that the major semantics of the original sentence can be preserved after multi-round paraphrase generation, especially the model with strong diversity. With the increase of paraphrasing round, the generated sentence will be more and more different from the original sentence, and the semantics will be gradually different from the original sentence as well. To tackle this problem, we introduce back-translation to maintain the semantics of paraphrase. Back-translation, which translates the generated sentence into the original sentence, has been widely used in semi-supervised natural language generation (Zhao et al., 2020) and data augmentation(Li et al., 2020), and it can improve the robustness of machine-translation system (Li

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1548–1557
August 1–6, 2021. ©2021 Association for Computational Linguistics

and Specia, 2019). We assume that paraphrase with similar semantics can be translated back to the original sentence. So, we can leverage back-translation to provide guidance for multi-round paraphrase generation.

Particularly, we propose **B**ack-**T**ranslation guided **m**ulti-round **P**araphrase **G**eneration (**BTmPG**), by combining neural paraphrase model with back-translation to generate paraphrases in a multi-round process. The contributions of our work are summarized as below:

1) We propose a new multi-round paraphrase generation method to generate diverse paraphrase that is much different from the original sentence and leverage back-translation to preserve the major semantics during the multi-round paraphrase generation. Our code is publicly available at https://github.com/L-Zhe/BTmPG.

2) Automatic and human evaluation results demonstrate that our method can substantially improve the diversity of generated paraphrase, while preserving the semantics during multi-round paraphrase generation.

2 Related Work

Paraphrase generation or sentence paraphrasing can been seen as a monolingual translation task. Prakash et al. (2016) leveraged stacked residual LSTM networks to generate paraphrase. Gupta et al. (2018) found deep generative model such as variational auto-encoder can be able to achieve better performance in paraphrase generation. Li et al. (2019) proposed DNPG to decompose a sentence into sentence-level pattern and phrase-level pattern to make neural paraphrase generation more interpretable and controllable, and they found DNPG can be adopted into unsupervised domain adaptation method for paraphrase generation. Fu et al. (2019) proposed a new paraphrase model with latent bag of words. Wang et al. (2019) found that adding semantics information into paraphrase model can significantly boost performance. Siddique et al. (2020) proposed an unsupervised paraphrase model with deep reinforcement learning framework. Liu et al. (2020) regarded paraphrase generation as an optimization problem and proposed a sophisticated objective function. All methods above focus on the generic quality of paraphrase and do not care about the diversity of paraphrase.

There are also some methods focusing on improving the diversity of paraphrase. Gupta et al. (2018) leveraged VAE to generate several different paraphrases by sampling the latent space. (Kumar et al., 2019) provided a novel formulation of the problem in terms of monotone sub-modular function maximization to generate diverse paraphrase. Goyal and Durrett (2020) used syntactic transformations to softly "reorder" the source sentence and guide paraphrase model. Thompson and Post (2020) introduced a simple paraphrase generation algorithm which discourages the production of n-grams that are present in the input to prevent trivial copies or near copies. Note that the purpose of the work (Gupta et al., 2018; An and Liu, 2019) is different from ours, while Thompson and Post (2020) has the same purpose with our work, i.e., pushing the generated paraphrase away from the original sentence.

3 Model

In this section, we introduce the components of our model in detail. First, we define the paraphrase generation task and give an overview of our model. Next, we describe the paraphrase model and the back-translation model. Then, we show how to use the gumble-softmax to connect the paraphrase model with the back-translation model. Finally, we describe the loss function and training process of our model in detail. Figure1 shows an overview of our model.

3.1 Notations and Overview

Our model regards paraphrase generation as a monolingual translation task. Given a paraphrase pair (S_0, P), which S_0 is the original/source sentence and P is the target paraphrase given in the dataset.

As is shown in Figure 1, we introduce a multi-round paraphrasing method. In the **first round** generation, we send S_0 into a paraphrase model to generate a paraphrase S_1. In the **second round** generation, we use the S_1 as the input of the model to generate a new paraphrase S_2. And so forth, in the ***i*-th round** generation, we send S_{i-1} into the paraphrase model to generate S_i.

Although multi-round generation can increase the paraphrase diversity, the semantics of paraphrase may change during generation. We thus introduce back-translation to tackle this problem based on the assumption that paraphrase can be translated back to the original sentence while the

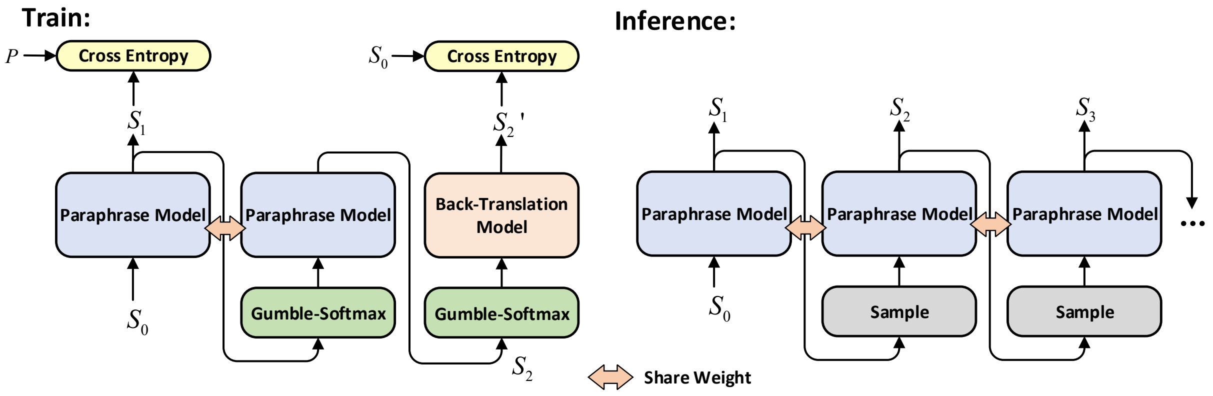

Figure 1: An overview of BTmPG, which leverages back-translation to guide paraphrase model during training and generates paraphrases in a multi-round process.

semantic information has not been changed. In the first round, we calculate the loss between S_1 and P to train our paraphrase model. In the i-th round, we send its generated paraphrase S_i into a back-translation model to generate $S_i^{'}$, and we optimize the cross-entropy loss between $S_i^{'}$ and S_0. The back-translation model which translates the paraphrase in i-th round back to the original sentence can guide the paraphrase to preserve semantics during multi-round generation.

In addition, we introduce gumble-softmax embedding to tackle the problem that the model with sampling operation between different rounds' generation can not be optimized by SGD optimizer.

3.2 Paraphrase Model

We require sufficient diversity of paraphrase model so that it is able to introduce enough changes in the paraphrase of each round. The VAE (Kingma and Welling, 2013; Rezende et al., 2014) is a deep generative model that allows learning rich, nonlinear representations for high dimensional inputs. It can improve the diversity by sampling from latent space. Bowman et al. (2016) proposed a new model to apply VAE to natural language generation for the first time. Our paraphrase model is based on conditional VAE with LSTM. Transformer (Vaswani et al., 2017) has achieved excellent performance in many tasks. But our experiments show that it may cause KL divergence to become 0, called posterior collapse, which means a decrease of diversity. So we do not employ Transformer as encoder and decoder.

We define the embedding matrices of S_i and P as $E_s^i = \{e_s^1, e_s^2, \cdots, e_s^{L_i}\}$ and $E_p = \{e_p^1, e_p^2, \cdots, e_p^M\}$ respectively, where $e_s^i, e_p^j \in \mathbb{R}^{d_e}$ are the embedding vector of the word in S_i and P, and d_e is the embedding dimension.

3.2.1 Encoder

Conditional VAE contains two encoders that share parameters: an original sentence encoder and a paraphrase encoder. We first send E_s^i into original sentence encoder to get its encoding matrix $O_s^i \in \mathbb{R}^{d_h \times L_i}$ and vector representation $h_s^i \in \mathbb{R}^{d_h}$ of S_i, where d_h is the hidden dimension of LSTM. Then we send E_p and h_s^i into paraphrase encoder to get its vector representation h_z. h_z is passed through two different feed-forward neural networks with parameter Φ to produce the mean μ and the variance σ^2 of the distribution of latent space. We can get the latent code $\boldsymbol{z} \in \mathbb{R}^{d_z}$ by sampling from latent space and reparameterization, where d_z is the dimension of latent code.

3.2.2 Decoder

We define the embedding matrix which be sent into decoder as $E_d = \{e_d^1, e_d^2, \cdots, e_d^N\} \in \mathbb{R}^{d_e \times N}$. Then, we concatenate z with the embedding vector e_d^i as the input of decoder. The decoder also takes h_s^i as input. The output of decoder is defined as $O_d^i \in \mathbb{R}^{d_h \times N}$. Then, an attention (Luong et al., 2015) and copy mechanism (See et al., 2017) are leveraged as follow. First, we get the attention weight p_a and attention vector V_a as follow.

$$\begin{aligned} p_a &= \mathrm{softmax}(O_d^i {O_s^i}^\top) \\ V_a &= p_a O_s^i \end{aligned} \tag{1}$$

Then, we leverage them to calculate the decoder probability p_d and copy probability η.

$$\begin{aligned} p_d &= \mathrm{softmax}(\mathbf{W}_o[O_d^i||V_a] + \mathbf{b}_o) \\ \eta &= \sigma(\mathbf{W}_h[O_d^i||V_a] + \mathbf{W}_s\{e_d^i||\boldsymbol{z}\}_{i=1}^N + \mathbf{b}_\eta) \end{aligned} \tag{2}$$

where $\mathbf{W}_o, \mathbf{W}_h, \mathbf{W}_b, \mathbf{b}_o, \mathbf{b}_\eta$ are learnable parameters. $||$ is the concatenation operation. σ is the sigmoid activation function. The final output probability of decoder is as follow.

$$p = \eta p_d + (1 - \eta) p_a \tag{3}$$

3.2.3 Loss Function of Paraphrase Model

The VAE with parameter Θ is trained by minimizing the following objective:

$$\begin{aligned} \mathcal{L}_{Para} = & - \mathrm{KL}(q_\Phi(\boldsymbol{z}|S_i, P)||p(\boldsymbol{z})) \\ & + \mathbb{E}_{q_\Phi(\boldsymbol{z}|P)}[\log p_\Theta(P|\boldsymbol{z}, S_i)] \end{aligned} \tag{4}$$

where KL stands for the KL divergence. Eq. 4 is called evidence lower bound, which provides a lower bound of $\log p(P|S_i; \Theta)$.

Bowman et al. (2016) figured out that variational inference for text generation often yields models that ignore their latent variables, a phenomenon called posterior collapse. This may cause the low diversity of generated sentences. To tackle this problem, we propose a diversity loss. We find that the diversity of the generated sentence is affected by its first word. For example, the first word can determine the form of a question sentence. Unfortunately, compared with the questions beginning with “Is, May, Would”, we are more likely to collect questions beginning with “What, When, How”. This can lead to serious category imbalances when generating the first word. So we set the penalty coefficient of the first-word loss as follow.

$$\mathcal{L}_{w_1} = \ln\left(\frac{N_b}{n_{w_1}} e\right) \log p(w_1|\Theta) \tag{5}$$

where N_b is the batch size during the training process, n_{w_1} is the number of sentences beginning with w_1 in this batch. e is the Euler’s number that can make sure the penalty coefficient always no less than 1.

3.3 Back-Translation Model

Back-translation model aims to make sure the semantics of the generated paraphrase are the same with the original sentence during multi-round generation. It translates S_i back to S_0. Different from paraphrase model which needs diversity, back-translation model is more focused on semantics maintaining. We employ Transformer (Vaswani et al., 2017) with copy mechanism as back-translation model because of its excellent performance in many tasks.

The loss function of back-translation model is as follow: [1]:

$$\begin{aligned} \mathcal{L}_s^i &= \mathrm{CrossEntropy}(\mathrm{BTModel}(S_i), S_0) \\ \mathcal{L}_p &= \mathrm{CrossEntropy}(\mathrm{BTModel}(P), S_0) \\ \mathcal{L}_{BT} &= \mathcal{L}_p + \lambda \sum_i \mathcal{L}_s^i \end{aligned} \tag{6}$$

where λ is a hyper-parameter. $\mathrm{BTModel}$ indicates the back-translation model.

There are two parts in the loss of back-translation model: $\mathcal{L}_s^i$ and $\mathcal{L}_p$. We assume the i-th round paraphrase can be translated back to the original sentence S_0 if its semantics are preserved and thus we optimize $\mathcal{L}_s^i$. Similarly, the paraphrase P can be translated back to the original sentence S_0 as well, so we also leverage $\mathcal{L}_p$ to train back-translation model. This can improve the generalization ability of the back-translation model, because back-translation model tends to guide paraphrase model to copy original sentence without changes if we do not employ true paraphrase data to train it.

3.4 Gumble-Softmax Embedding

We employ gumble-softmax embedding to connect each module of our model. We first define an embedding operation as follow:

$$\mathrm{Embed}(X) = \mathbf{W}_e X \tag{7}$$

For the probability p generated by paraphrase model, we leverage gumble-softmax(Jang et al., 2017) to get its one-hot matrix without sampling from multinomial distribution. Then we can get the embedding matrix E as follow:

$$\begin{aligned} \mathrm{GS}(\boldsymbol{\pi}) &= \mathrm{softmax}((\log(\pi)_i + g_i)/\tau) \\ E &= \mathrm{Embed}\left(\mathrm{GS}(p)\right) \end{aligned} \tag{8}$$

where $\boldsymbol{\pi}$ is a multinominal distribution wih k dimension, $g_1, g_2, \cdots, g_i$ are i.i.d samples drawn from $\mathrm{Gumbel}(0, 1)$. τ is a hyper-parameter.

There are three places in our model needing to leverage gumble-softmax embedding. First, we leverage it to embed the output probability of the paraphrase model as the input of the next-round paraphrase model. Next, gumble-softmax embedding is also used to connect the back-translation model with the paraphrase model. Figure 1 shows

[1] Note that we also use S_0, S_i and P to denote the one-hot matrix of corresponding sentences.

these two cases. Finally, it is used in the multi-round paraphrase generation process to replace the teacher forcing. Generally, Seq2Seq model employs teacher forcing for model training, with using ground truth to guide the generation process. However, there is no ground truth in multi-round paraphrase generation, it can only generate sentence with a autoregressive method. We employ gumble-softmax to replace sampling in each step of the autoregressive process. Figure 2 shows this process.

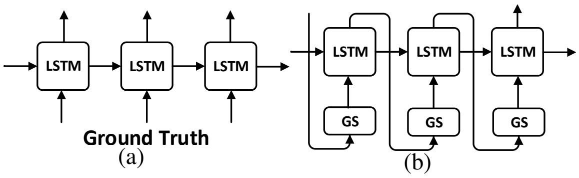

Figure 2: Figure (a) shows the decoder with teacher forcing in the first round generation. Figure (b) shows the decoder with autoregression in other-round generation.

3.5 Loss Function

We train paraphrase model together with back-translation model. The total loss of our model is as follow:

$$\mathcal{L} = \mathcal{L}_{para} + \mathcal{L}_{BT} \tag{9}$$

Although we define a multi-round paraphrase model, we only train the first two rounds. Because we find that training too many rounds requires large computing resources, but can not improve the model performance significantly. During inference, we can generate paraphrase more than two rounds.

4 Experiment

4.1 Datasets

We evaluate our BTmPG model on two benchmark datasets:

MSCOCO[2] (Lin et al., 2014) dataset contains human annotated captions of over 120k images. Each image contains five captions from five different annotators. This dataset has been widely used in previous works (Prakash et al., 2016; Gupta et al., 2018; Cao and Wan, 2020). We sample the MSCOCO according to Prakash et al. (2016).

Quora[3] dataset is a question paraphrase dataset. It contains over 400k question pairs. Each pair marked with a binary value indicates whether the questions in the pair are truly a duplicate of each other. So we select all such question pairs with binary value 1 as paraphrase dataset. There are about 150k question pairs in total. We randomly divide the training, validation and the test set.

Table 1 provides statistics of these two benchmark datasets.

Dataset	Train Set	Valid Set	Test Set
MSCOCO	206,852	3,000	3,000
Quora	129,263	3,000	3,000

Table 1: Statistic for datasets: the sizes of training, validation and test set.

4.2 Evaluation Metrics

We use five widely-used metrics to evaluate paraphrases: BLEU4, self-BLEU, self-TER, BERTScore and p-BLEU.

BLEU4 is widely used in generation tasks. It can measure how well the sentences generated by our model can match the references. Notice that some works also calculate the ROUGE(Lin, 2004) or METEOR, but we think the role of these two metrics overlaps with BLEU4, as they all calculate the overlap degree between outputs and references. Therefore we only calculate BLEU4 to evaluate the match degree between outputs and references.

We evaluate the difference between the output sentence and the original sentence with two metrics. One of them is **self-BLEU** which is the BLEU4 score between the output sentence and the original sentence. The lower the value of self-BLEU, the more difference between output sentences and original sentences. Another is **self-TER**[4]. **TER**(Zaidan and Callison-Burch, 2010) is used to evaluate the edit distance between two sentences. Self-TER is calculated as the TER between the output sentence and the original sentence.

BERTScore [5] is proposed by Zhang et al. (2020) to evaluate the semantic similarity between the output sentence and the original sentence. BERTScore

[2] https://cocodataset.org/

[3] https://www.kaggle.com/c/quora-question-pairs/data?select=train.csv.zip

[4] We use the tool at https://github.com/jhclark/multeval.

[5] The tool of BERTScore is available at https://github.com/Tiiiger/bert_score

has been widely leveraged to measure semantic preserving in the paraphrase generation task (Cao and Wan, 2020). However, there may be some problems for BERTScore on our task due to the low score for reference. This is because BERTScore is not perfect in measuring semantic relevance. But as far as we know, there is no better score to evaluate semantic preserving, so we report BERTScore as a reference for semantic preserving. More evaluation about semantic relevance is shown in human evaluation.

We leverage **p-BLEU** (Cao and Wan, 2020) to evaluate the difference between outputs in different rounds. Concretely, for outputs in k rounds $\{y_1, y_2, \cdots, y_k\}$, the p-BLEU can be calculated as follow.

$$\text{p-BLEU} = \frac{\sum_i \sum_{j \neq i} \text{BLEU4}(y_i, y_j)}{k \times (k-1)} \tag{10}$$

The lower p-BLEU means higher diversity between outputs in different rounds.

Notice that, BLEU4 may not suitable for our task , because we focus on the diversity of paraphrase. BLEU4 can only measure the match degree between outputs and references. However, a sentence usually has many more reference paraphrases, while the target given in the dataset is only one reference. So we also perform human evaluation to evaluate the semantic relevance, readability and diversity of generated paraphrases.

4.3 Baseline

As our model focuses on the diversity of paraphrase, we mainly compare our model with VAE-SVG-eq (Gupta et al., 2018), DiPS(Kumar et al., 2019)[6], SOW-REAP(Goyal and Durrett, 2020)[7] and the decoding method proposed by Thompson and Post (2020)[8]. The last method penalizes the n-gram appearing in the original sentence to make the paraphrase different from the original sentence and enhance diversity. We mark this method as N-gram Penalty. We employ two different hyper-parameters provided by the authors: one of them is low penalty for N-gram, and another is high penalty. In addition, we also compare our model with Transformer and Transformer copy.

4.4 Training Details

For both datasets, we truncate all the sentences longer than 20 words and maintain a vocabulary size of 25k. During testing, we replace UNK with the original word with the highest copy probability.

For paraphrase model, we leverage 2-layer LSTM. We set the embedding dimension d_e to 300, hidden size d_h of LSTM to 512. We set the latent code dimension d_z to 128. For back-translation model, we leverage Transformer-copy with 3-layer encoder and decoder. We set the model size to 450, and the head number of multi-head attention to 9. We set λ to 1, which will be discussed in our ablation study. For the hyper-parameter τ in gumble-softmax, we refer (Nie et al., 2019) to increase the τ over iterations via an exponential policy: $\tau = \tau_{\max}^{-n_e/N_e}$, where n_e is the current epoch and N_e is the total number of epoch. We set $\tau_{\max}$ to 5. We train our model for 30 epochs. We set batch size to 50, and we select the model of the final epoch to generate paraphrase in test set.

5 Result

5.1 Automatic Evaluation

Table 2 shows the results of automatic evaluation. Our model substantially improves the BERTScore in the first round of paraphrase generation and generally gets the state-of-the-art performance. The value of self-BLEU can be significantly reduced with the increase of the round number of paraphrase generation while maintaining semantics.

For both datasets, the first round paraphrase generation of our model achieves the highest BERTScore than any other models. This is because back-translation model can provide sufficient semantic guidance for paraphrase model. As the increase of the round number, the values of self-BLEU and self-TER are reduced significantly, which means the paraphrase sentences our model generated are more and more different from original sentences. While BERTScore can still maintain a relatively high value. (A slight reduction of BERTScore is acceptable as BERTScore is not perfect in measuring semantic relevance.) We find that the paraphrase generated in the fifth round is good with balancing the diversity and the relevancy.

DiPS gets the BERTScore similar to round 5 generation, while its outputs lack of diversity com-

[6] The code is available at https://github.com/malllabiisc/DiPS.

[7] The code is available at https://github.com/tagoyal/sow-reap-paraphrasing.

[8] DNPG (Li et al., 2019), which controls semantics through encoding different levels of granularity respectively, can also enhance diversity. But the code and outputs are not provided, so we are not able to use it as baseline.

Model		MSCOCO				Quora			
		BLEU4	self-BLEU ↓	self-TER ↑	BERTScore ↑	BLEU4	self-BLEU ↓	self-TER ↑	BERTScore ↑
Reference		-	8.12	78.40	46.20	-	31.46	54.92	67.37
VAE-SVG-eq		25.07	13.77	66.92	51.72	22.52	36.05	50.25	67.06
Transformer		25.81	14.92	65.74	54.47	26.22	37.99	46.42	67.53
Transformer copy		26.80	17.94	62.49	56.49	28.97	43.69	42.52	71.05
DiPS		23.52	12.23	67.31	51.40	23.38	29.24	54.41	63.09
SOW-REAP		15.31	44.22	39.42	**63.68**	15.36	47.62	38.98	62.21
N-gram Penalty-low		25.37	12.16	66.66	53.47	26.00	36.65	47.31	66.93
N-gram Penalty-high		23.68	0.00*	69.24	52.08	17.53	0.00*	**59.30**	59.20
BTmPG (Ours)	R1	25.54	18.50	61.78	59.34	28.02	58.47	33.99	**77.21**
	R5	23.65	12.58	68.27	54.07	23.15	37.89	48.62	65.90
	R10	22.42	**10.98**	**70.10**	52.37	22.17	**34.15**	53.34	62.91

Table 2: Automatic evaluation results on MSCOCO and Quora test sets. In the table, R1, R5 and R10 mean the first round, the fifth round and the tenth round of paraphrase generation.

Model	p-BLEU ↓	
	MSCOCO	Quora
VAE-SVG-eq	75.52	81.50
BTmPG(Ours)	**62.83**	**67.60**

Table 3: The p-BLEU score for VAE-SVG-eq and BTmPG

pared with our model. SOW-REAP gets the highest BERTScore for MSCOCO, but it does not perform well on self-BLEU. Because SOW-REAP tends to generate paraphrase without change, the semantics of the paraphrase may be similar with the original sentence but the paraphrase lacks of diversity. N-gram Penalty with high penalty can lead self-BLEU to 0, as it strictly does not allow to generate those 4-grams appearing in the original sentence. Although the N-gram Penalty method can generate outputs totally different from original sentences, it fails to preserve the major semantics. However, our BTmPG model can increase diversity as much as possible while preserving major semantics.

To explore the pairwise diversity of our model's outputs in different rounds, we also calculate the p-BLEU values for VAE-SVG-eq and our model (p-BLEU is not suitable for other models). For VAE-SG-Eq, we generate 10 outputs by random sampling the latent space. For our model, we select the first 10 rounds outputs. Table 3 shows the results of p-BLEU. The p-BLEU value of our model is much lower than VAE-SVG-eq, which means that our model has better ability to generate multiple diversified paraphrases than VAE-SVG-eq.

5.2 Ablation Study

In this section, we will explore the role of back-translation model in preserving semantics. We set the hyper-parameter λ from 0 to 5. A bigger λ means back-translation provides more semantic guidance to paraphrase model. $\lambda = 0$ means that we remove back-translation model totally. We generate paraphrases of 20 rounds and calculate the values of BERTScore. In order to explore the effect of leveraging other paraphrase model in the multi-round generation framework, we also adopt VAE-SVG-eq in a multi-round generation process to generate paraphrases of 20 rounds on Quora, and compute the values of BERTScore. Figure 3 shows the trend of BERTScore with the increase of the round number.

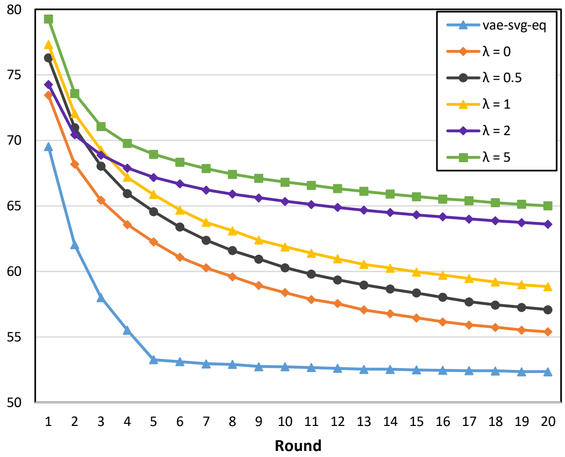

Figure 3: The BERTScore of paraphrases of 20 rounds on Quora.

Obviously, compared with VAE-SVG-eq, our improved VAE model can preserve semantics better. Back-translation can much improve the lower bound of BERTScore , which means back-translation can help to preserve the semantics during multi-round paraphrase generation.

We also calculate the p-BLEU for the paraphrases of the first 10 rounds for different λ. Table 4 shows the result. From the table we can know that, although back-translation can help to preserve semantics, a higher λ can lead to a lack of diversity of paraphrase. Therefore, it is wise to select an appropriate λ according to the actual requirement.

λ	0	0.5	1	2	5
p-BLEU	63.53	66.40	67.61	74.83	88.05

Table 4: The p-BLEU of paraphrases of the first 10 rounds for different λ.

5.3 Human Evaluation

We perform human evaluation on system outputs with respect to three aspects: relevancy, fluency and diversity. Relevancy indicates if the semantics of outputs and original are identical. Fluency indicates the readability of output sentences. Diversity indicates the lexical and syntactic differences between output sentences and original sentences and thus we use two indicators for lexical diversity and syntactic diversity respectively.

We randomly sample 100 sentences from each test set and get a total of 200 sentences for evaluation. We employ 6 graduate students to rate each instance. We ensure every instance is rated by at least three judges. Table 5 shows the result of human evaluation.

Model		**Relevancy**	**Fluency**	**Diversity**	
				Lexical	**Syntactic**
VAE-SVG-eq		3.24	3.44	3.93	4.01
Transformer		3.82	**3.96**	3.71	3.77
DiPS		3.62	3.50	3.64	3.70
SOW-REAP		3.59	3.34	2.79	3.88
N-gram Penalty		3.44	3.65	3.79	3.68
BTmPG (Ours)	R1	**4.12**	3.92	3.65	3.85
	R5	3.93	3.81	3.95	4.00
	R10	3.84	3.82	**4.20**	**4.15**

Table 5: Human evaluation results.

From the table, we can see that the paraphrase in the first round can preserve more semantics of original sentence but lack of diversity. With the increase of the round number, the relevancy score decreases slightly, but the diversity scores increase substantially. Fluency may be influenced by diversity, because human may feel a slight decrease of fluency with the increase of diversity. As compared with other models, our model can generate paraphrases with high diversity, while maintaining semantics and fluency well. Previous models like SOW-REAP and DiPS can not maintain the semantics, though they can produce paraphrases with relatively high diversity.

5.4 Case Study

We perform case studies for better understanding the model performance. Table 6 shows an example of Quora, which include paraphrases of the first 15 rounds.

Cases from Quora	
Original	why did modi scrap rs 500 & rs 1000 notes ? and what 's the reason for the sudden introduction of the 2000 rupee note ?
Reference	why did goi demobilise 500 and 1000 rupee notes ?
Round1	why did the indian government ban the 500 and 1000 rupee notes and why is it bringing to ?
Round2	what do you think about the ban on 500 and 1000 denomination notes in india ?
Round4	how do you see the pm modi 's move of banning 500 and 1000 rupee currency notes ?
Round5	what do you think of the decision by the indian government to demonetize 500 and 1000 rupee notes ?
Round9	is modi 's decision on demonetization of 500 and 1000 notes by public modi ?
Round11	was the decision by the indian government to demonetize 500 and 1000 notes right or wrong ?
Round12	would banning notes of denominations 500 and 1000 help to curb the black money in india ?
Round13	what will be the effects of banning 500 and 1000 rupees on indian economy ?
Round14	what are the advantage of banning 500 and 1000 rupees in Indian ?
Round15	what are the pros and corns of banning 500 and 1000 rupees by indian government ?

Table 6: An example of Quora and the generated paraphrases in multiple rounds. The word in color means that it does not appear in the original sentence.

This case shows how does our model modify sentences during multi-round paraphrase generation process. With the increase of round number, the difference between the generated paraphrase and the original sentence becomes larger, while the paraphrase still preserves the major semantics of the original sentence.

6 Conclusion

In this paper, we focus on improving the diversity of generated paraphrase, i.e., making the generated paraphrase much more different from the original sentence. We propose a multi-round paraphrase generation method **BTmPG** with the guidance of back-translation. Both automatic and human evaluation results show that our method can generate diverse paraphrase while maintaining semantics. Ablation study proves back-translation is very helpful to preserve semantics. In the future, we will explore other methods such as GAN, to improve paraphrase diversity. We will also test our method on more languages other than English.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772036), Beijing Academy of Artificial Intelligence (BAAI) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We appreciate the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

Zhecheng An and Sicong Liu. 2019. Towards diverse paraphrase generation using multi-class wasserstein GAN. *CoRR*, abs/1909.13827.

Rahul Bhagat and Eduard Hovy. 2013. Squibs: What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Yue Cao and Xiaojun Wan. 2020. DivGAN: Towards diverse paraphrase generation via diversified generative adversarial network. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2411–2421, Online. Association for Computational Linguistics.

Yao Fu, Yansong Feng, and John P Cunningham. 2019. Paraphrase generation with latent bag of words. In *Advances in Neural Information Processing Systems*, pages 13645–13656.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 238–252, Online. Association for Computational Linguistics.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. *International conference on learning representations*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *International Conference on Learning Representations*.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3609–3619, Minneapolis, Minnesota. Association for Computational Linguistics.

Hongzheng Li, Jiu Sha, and Can Shi. 2020. Revisiting back-translation for low-resource machine translation between chinese and vietnamese. *IEEE Access*, 8:119931–119939.

Zhenhao Li and Lucia Specia. 2019. Improving neural machine translation robustness via data augmentation: Beyond back-translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 328–336, Hong Kong, China. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2020. Unsupervised paraphrasing by simulated annealing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 302–312, Online. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Weili Nie, Nina Narodytska, and Ankit Patel. 2019. Relgan: Relational generative adversarial networks for text generation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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, Osaka, Japan. The COLING 2016 Organizing Committee.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *international conference on machine learning.*, pages 1278–1286.

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, Vancouver, Canada. Association for Computational Linguistics.

A. B. Siddique, Samet Oymak, and Vagelis Hristidis. 2020. Unsupervised paraphrasing via deep reinforcement learning. In *Association for Computing Machinery*, KDD '20, page 1800–1809.

Brian Thompson and Matt Post. 2020. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online. Association for Computational Linguistics.

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.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2010. Paraphrase generation as monolingual translation: Data and evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*.

Omar F. Zaidan and Chris Callison-Burch. 2010. Predicting human-targeted translation edit rate via untrained human annotators. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 369–372, Los Angeles, California. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Shiqi Zhao, Haifeng Wang, Xiang Lan, and Ting Liu. 2010. Leveraging multiple MT engines for paraphrase generation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1326–1334, Beijing, China. Coling 2010 Organizing Committee.

Yanbin Zhao, Lu Chen, Zhi Chen, and Kai Yu. 2020. Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. In *AAAI*, pages 9668–9675.