Exploring Bilingual Parallel Corpora for Syntactically Controllable Paraphrase Generation

Mingtong Liu[1*], **Erguang Yang**[1], **Deyi Xiong**[2], **Yujie Zhang**[1†], **Chen Sheng**[3], **Changjian Hu**[3], **Jinan Xu**[1] and **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, {shengchen1, hucj1}@lenovo.com

Abstract

Paraphrase generation is of great importance to many downstream tasks in natural language processing. Recent efforts have focused on generating paraphrases in specific syntactic forms, which, generally, heavily relies on manually annotated paraphrase data that is not easily available for many languages and domains. In this paper, we propose a novel end-to-end framework to leverage existing large-scale bilingual parallel corpora to generate paraphrases under the control of syntactic exemplars. In order to train one model over the two languages of parallel corpora, we embed sentences of them into the same content and style spaces with shared content and style encoders using cross-lingual word embeddings. We propose an adversarial discriminator to disentangle the content and style space, and employ a latent variable to model the syntactic style of a given exemplar in order to guide the two decoders for generation. Additionally, we introduce cycle and masking learning schemes to efficiently train the model. Experiments and analyses demonstrate that the proposed model trained only on bilingual parallel data is capable of generating diverse paraphrases with desirable syntactic styles. Fine-tuning the trained model on a small paraphrase corpus makes it substantially outperform state-of-the-art paraphrase generation models trained on a larger paraphrase dataset.

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.*, 2018], and text summarization [Zhao *et al.*, 2018].

x: we need to further strengthen the agency's capacities.
z: the damage in this area seems to be quite minimal.
y: the capacity of this office needs to be reinforced even further.

x: his teammates' eyes got an ugly, hostile expression.
z: the smell of flowers was thick and sweet.
y: the eyes of his teammates had turned ugly and hostile.

Figure 1: Illustration of generating syntactically-controllable paraphrases with sentential exemplars, which transforms a given input x to a new sentence y that is semantically similar to x but in a different syntactic style like z.

Most recent state-of-the-art approaches to PG employ neural architectures [Prakash *et al.*, 2016; Hasan *et al.*, 2016; Gupta *et al.*, 2018], which normally depend on a large amount of manually annotated paraphrase corpus for training. As constructing a large paraphrase corpus is inevitably not cheap and time-consuming, quickly developing a high-quality PG system on a small corpus mount a formidable practical challenge in many languages and domains. An effective solution to this problem is transferring knowledge from a high-resource task with abundant annotated data to a low-resource task with limited or even no annotated data.

Recent progress has also witnessed that learning controllable paraphrase generation (CPG) with desirable styles is emerging as an area of intense focus in the literature, e.g., satisfying particular sentiment, template or syntactic structure [Ficler and Goldberg, 2017; John *et al.*, 2018; Iyyer *et al.*, 2018; Chen *et al.*, 2019]. This technique has benefited several NLP tasks, such as generating diverse and adversarial samples to improve model generalization capability and robustness [Iyyer *et al.*, 2018]. However, CPG aggravates the requirement of data annotation as guidance signals such as syntactic templates are needed [Chen *et al.*, 2019].

In this work, following recent efforts, we focus on using a sentential exemplar to control the syntactic realization of generated sentences [Wang *et al.*, 2019; Chen *et al.*, 2019], as

[*]Contribution during internship at Lenovo Research AI Lab.

[†]Corresponding author.

shown in Figure 1. But unlike existing CPG models that rely on large annotated paraphrase corpora, we are interested in developing a high-quality CPG model even without annotated paraphrase data.

As there are plenty of large-scale bilingual parallel corpora which can be regarded as paraphrases written in different languages, we propose to explore large-scale off-the-shelf bilingual corpora for CPG. We find a way to transfer knowledge in bilingual paraphrases into monolingual paraphrase generation and use sentences from bilingual corpora as syntactical exemplars, avoiding annotation work in manually creating paraphrases and guiding exemplars. Our method is therefore able to reduce the reliance of CPG on large paraphrase data.

Specifically, we propose to extend the widely-used encoder-decoder model [Bahdanau *et al.*, 2014] to include a content encoder for meaning modeling, a style encoder for style extraction from syntactical exemplars and two syntax-guided variational decoders. By projecting input sentences into the same space via cross-lingual embeddings, we share both of the content and style encoder across different languages. This sharing mechanism enables the model to be trained on bilingual parallel data and monolingual paraphrase data simultaneously, allowing knowledge transfer from bilingual sentence pairs to PG. To disentangle the content from style space, we employ an adversarial loss over the learned content and style representations. To provide global syntactic guiding signal for the two decoders, we introduce a variational latent variable to model style representation. Additionally, we introduce a masking learning scheme to reduce the dependence of the model on token representations of the syntactic exemplar, and therefore to encourage the style encoder to extract more syntax-related information.

In order to enable our model to generate paraphrases by only learning from bilingual parallel data without using any annotated monolingual paraphrases, we further propose a cycle learning scheme that uses the learned model to generate pseudo paraphrase and translation data, and updates the learned model by training it on the generated data. The strategy allows paraphrase generation and machine translation to benefit each other in a cycle, which forms the basis for our proposed multi-task learning framework for the two tasks.

In summary, our contributions are threefold as follows:

- We propose a new controllable paraphrase generation framework with syntactical exemplars, which benefits PG from bilingual parallel corpora.
- In order to enable the model to explore bilingual corpora for paraphrase generation, we propose to equip the model with several vital components and learning strategies in a new way: (1) two shared encoders for disentangled content and style representations, enhanced by cross-lingual embeddings to create the same space for different languages and an adversarial discriminator to distinguish content and syntactic style; (2) syntax-guided variational decoders that generate a sentence from both the encoded content representation and a latent syntactic variable learned by variational autoencoder; and (3) a translation-generation cycle learning scheme that enhances PG from translation and vice versa.

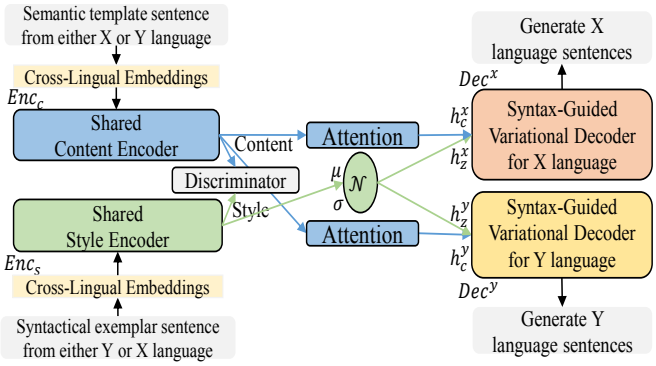

Figure 2: The proposed neural architecture for CPG.

- Experiment results show that our model can generate diverse paraphrases with desirable semantic content and syntactic style. When the model is fine-tuned on a small annotated paraphrase data, it substantially outperforms previous state-of-the-art approaches.

2 Related Work

Neural paraphrase generation (PG) 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. Gupta *et al.* [2018] use a variational autoencoder framework to generate diverse paraphrases. Despite their success, the performance of their models suffers from small training data. Different from their work, we substantially benefit PG from large-scale bilingual corpora, and therefore alleviate the heavy reliance of PG on annotated paraphrase data.

Our approach also relates to recent works on style transfer and controllable text generation. Many previous methods attempt to control the attributes of generated texts such as sentiment and formality [Ficler and Goldberg, 2017; John *et al.*, 2018; Lample *et al.*, 2018], and also to control the structural aspects of generated sentences [Wiseman *et al.*, 2018; Iyyer *et al.*, 2018]. More recent works use sentences as exemplars to graft their syntax patterns to other sentences [Wang *et al.*, 2019; Chen *et al.*, 2019]. Similar to them, we also use a sentential exemplar to provide a syntax form for the generation. However, significantly different from them, we extend the model to use bilingual corpora so that our model can generate paraphrases with the desirable syntactic styles even without using any annotated paraphrase training data.

3 The Proposed Model

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

3.1 Overall Architecture

Our model is illustrated in Figure 2. Its backbone is built on the encoder-decoder architecture [Bahdanau *et al.*, 2014].

The model has two encoders: a content encoder Enc_c for encoding the content of a sentence and a style encoder Enc_s for modeling the syntactic style of the given exemplar. The two encoders are shared across the two different languages in the bilingual training corpora. We use cross-lingual word embeddings to bridge the two different languages, and introduce a discriminator with adversarial loss on the two encoders to enforce the separation of the style and content spaces. We also have two syntax-guided variational decoders Dec^x and Dec^y, each of which is to generate sentences in one language with the guidance of syntactic exemplars. In this work, we use a two-layer bidirectional RNN in each encoder, and two-layer unidirectional RNN in each decoder. All RNNs use LSTM cells [Hochreiter and Schmidhuber, 1997].

We formulate the problem of syntactic-controlled paraphrase generation as follows. Given two sentences x and z as input, we want the generated sentence y from our CPG model to inherit the meaning of sentence x and be in the syntactic form of sentence z. We refer to x and z as the semantic template and syntactic exemplar, respectively.

We can use bilingual parallel sentence pairs (x, y) to train our model. z can be any valid sentence from the same language of y, but we set z = y at training procedure to avoid constructing additional syntactical exemplars. In this way, we want our model to learn for y the meaning of x from the content encoder and the syntactic style of y itself from the disentangled style encoder.

Concretely, for the two languages X and Y in bilingual training data, the model is trained in two directions X→Y and Y→X. We use (x, z; y) to train X→Y, where x and z are input to the content and style encoder and y is the ground truth for the Y language decoder. Similarly, we use (y, z; x) to train Y→X. To save space, we use the X→Y direction to introduce our model thereafter and all computations in the other direction can be derived accordingly.

3.2 Content and Style Encoders

As we use the two encoders Enc_c and Enc_s for both X and Y language, we project sentences from the two languages into the same content and style spaces. For this, we use the pre-trained cross-lingual embeddings that are kept fixed during training as the input of the two encoders. Concretely, we use word2vec to train word embeddings for each language on monolingual corpus. After that, we employ the unsupervised self-learning method [Artetxe *et al.*, 2017] to obtain cross-lingual embeddings in the same space.

3.3 Adversarial Discriminator

Inspired by previous work [John *et al.*, 2018], we design a discriminator to enforce the separation of the content from the style space. In particular, the discriminator predicts a $content$ label when the input from the content space, while a $style$ label when the input from the style space.

Formally, for each sentence, let s be the vector representation learned by either the content or style encoder, we first use the mean pooling operation to average across each position, which is followed by a feed-forward neural network (FFN). After that, a two-way softmax layer is applied to s, given by:

$$y_s = \mathrm{softmax}(W_{dis}\mathrm{FFN}(s) + b_{dis}) \quad (1)$$

where $\theta_{dis} = [W_{dis}, b_{dis}]$ are parameters, and y_s indicates the probability that the input is from the content or style space.

The discriminator is trained with a cross-entropy loss against the ground-truth distributions $t_s(\cdot)$:

$$loss_{adv}(\theta_{Enc_c}, \theta_{Enc_s}) = -\sum_{l \in labels} t_s(l) \log y_s(l) \quad (2)$$

where θ_{Enc_c} and θ_{Enc_s} are parameters of the two encoders, $l \in \{content, style\}$ and $y_s(l)$ is the predicted distribution.

3.4 Syntax-Guided Variational Decoders

We use two non-shared decoders (Dec^x) and (Dec^y), one decoder per language. Each decoder generates sentences in the corresponding language based on the semantic and syntactic representations learned by the encoders. To bridge the decoders with the content encoder, we use the attention mechanism with bilinear product [Bahdanau *et al.*, 2014] to compute semantic representation h_c. To incorporate information from the style encoder into the decoders, we introduce a latent variable z based on variational autoencoder (VAE) [Kingma *et al.*, 2014] to model the underlying syntactic style as a global signal for generation.

In particular, because of the nature of bilinguality in the style encoder shared by the two languages, we make the latent variable z have the same distribution across the two different languages in shared space. Following the setting of VAE, we use KL-divergence to encourage the posterior distribution $q(z|y)$ to be close to the prior $p(z|x, y)$, where the prior $p(z|x, y)$ is modeled from two languages.

Formally, the joint training objective for a training instance (x, y) is defined as follows:

$$\mathcal{J}(\theta) = -KL(q(z|y)||p(z|x, y)) + \prod_{t=1}^{T} p(y_t|y_{1:t-1}, z, x) \quad (3)$$

where KL stands for KL divergence between the posterior $q(z|y)$ and the prior $p(z|x, y)$. The $p(y_t|y_{1:t-1}, z, x)$ is the decoder with the guidance from z, where y_t is the t-th word of y and $p(y_t|y_{1:t-1}, z, x)$ is given by a softmax over a vocabulary V. The details of the syntax-guided variational decoder are given below.

Latent Syntactic Variable with VAE. We model the posterior and prior with the multivariate Gaussian distribution, which allows the latent syntactic variable to be in a continuous space [Bowman *et al.*, 2015]. We use two sentences x and y to compute the posterior $q(z|y)$ and prior $p(z|x, y)$:

$$q(z|y) = \mathcal{N}(z; \mu(y), \sigma(y)^2 I) \quad (4)$$

$$p(z|x, y) = \mathcal{N}(z; \mu(x, y), \sigma(x, y)^2 I) \quad (5)$$

where the mean μ and s.d. σ of the approximate posterior and prior are the outputs of the style encoder.

Concretely, for the output of the style encoder, we use the mean pooling operation to average across each position to get h_s^x and h_s^y. Then we use two FFNs to project the h_s^x and h_s^y into a latent space for computing the posterior and prior.

$$z_y = \mathrm{FFN}(\mathrm{h_s^y}) \quad (6)$$

$$z_{xy} = \mathrm{FFN}([\mathrm{h_s^x}; \mathrm{h_s^y}]) \quad (7)$$

where $[\cdot;\cdot]$ refers to the concatenation operation.

In this latent syntactic space, we obtain the prior Gaussian parameters $\mu(x, y)$ and $\sigma(x, y)$ through a linear mapping:

$$\mu(x, y) = W_\mu z_{xy} + b_\mu \quad (8)$$

$$\log \sigma(x, y)^2 = W_\sigma z_{xy} + b_\sigma \quad (9)$$

where θ_{vae}=[W_μ,b_μ,W_σ,b_σ] are parameters. We use the same method as Eqs (8) and (9) to compute the posterior Gaussian parameters $\mu(y)$ and $\sigma(y)$, except that the input is z_y.

Then we employ the reparameteried technique [Kingma *et al.*, 2014] for VAE training, setting $h_z = \mu(x, y)+\sigma(x, y)\odot\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$. This facilitates its optimization since we can apply the standard backpropagation to compute the gradient in an end-to-end manner.

Decoding Stage. At each decoding step *t*, we use the latent syntactic variable h_z as a global signal to control the decoder prediction, which enforces the generated sentence to satisfy the given syntactic style. Specifically, we concatenate the latent syntactic variable h_z, the attentional semantic representation h_c and the previous word's embedding as the input to the decoder, for predicting the word at next time step.

4 Cycle Learning

In order to train our model in a true paraphrase generation setting, we exploit a "cycle" learning scheme to further improve the model, by creating pseudo paraphrase training data.

Our model is also designed to be able to perform paraphrase generation if we use the decoder that generates sentences in the same language as that of the input of the content encoder. Therefore, we can adopt the following two methods to generate pseudo training data. We use the learned model in an inference way to perform generation via greedy decoding.

Given a bilingual parallel sentence pair (x, y), we use the learned model to translate sentence x from one language to the other language. The generated translation y′ should be very close to the original sentence y in terms of meanings.

$$\mathrm{y}' = Dec^{y}(Enc_c(\mathrm{x}), Enc_s(\mathrm{z})) \quad (10)$$

where we set z = x. Specially, in order to generate diverse sentences, we inject random noise into the latent variable in the continuous syntactic space, where $h_z = \mu(z) + \epsilon$, $\epsilon \sim \mathcal{N}(0, I)$. Thus we obtain pseudo paraphrase pairs (y′, y).

We also enhance translation by using the learned paraphrase generation model. We use the learned model to paraphrase a sentence x to another x′ in the same language. The sentence x′ can be translated back to the original sentence y in the other language.

$$\mathrm{x}' = Dec^{x}(Enc_c(\mathrm{x}), Enc_s(\mathrm{z})) \quad (11)$$

where we set z = x. We also use noise in the syntactic variable. In this way, we generate pseudo bilingual pairs (x′, y).

5 Training and Inference

Training Procedure. Given two bilingually aligned sentences x in X language and y in Y language, we use (x, y) to train the parameters related to the direction from X to Y (i.e., the two encoders, discriminator and the Y decoder), and (y, x) for the other direction from Y to X. Meanwhile, we also use the generated pseudo parallel data (as described in Sect. 4) to train the model. We take (x′, y) as (x, y) to train the translation from X to Y, and (y′, x) as (y, x) to train the translation from Y to X. Specially, we take (x′, x) to train the paraphrase generation model for X language, and (y′, y) to train the paraphrase generation model for Y language. We alternately use the six types of sentence pairs to train the model.

Training Objectives. The training loss for a training instance (x, z; y) (where z = y) is a combination of the generation loss, the KL-divergence penalty (Eq. 3), and the adversarial loss (Eq. 2).

$$loss_{all}(\theta) = \lambda_1 loss(y|z, x) + \lambda_2 loss_{KL} + \lambda_3 loss_{adv} \quad (12)$$

where $loss(y|z, x) = -\sum_{t=1}^{T} \log p(y_t|y_{1:t-1}, z, x)$ is the cross-entropy loss for generating sentence y given x and z, and λ_* are hyperparameters that balance these losses.

Masking on Syntactic Exemplars. During training, in order to avoid additionally collecting syntactic exemplars, we set z = y as the input of the style encoder. In order to encourage the model to capture more syntax-related information through context and reduce the dependence on token representations, we use the [$\mathbb{MASK}$] token as introduced in Devlin *et al.* [2019] to mask the syntactic input of each training example at the token level. We replace a chosen token with the [$\mathbb{MASK}$] 30% of the time, a random token 10% of the time and do not change the chosen token 60% of the time.

Inference. Given a semantic template and syntactic exemplar from the same language, we input the semantic template to the content encoder and the syntactic exemplar to the style encoder, and then use the decoder corresponding to the language of the inputs to generate a syntactically controllable paraphrase.

6 Experiments

We conducted two groups of experiments, one for controllable paraphrase generation and the other for general paraphrase generation without syntactic guidance. We used the Chinese-English bilingual dataset CWMT (neu2017) to train the proposed model, and the ParaNMT dataset for evaluating controllable paraphrase generation, the Quora and MSCOCO datasets for evaluating general paraphrase generation.

6.1 Datasets

CWMT Chinese-English Dataset. This dataset contains 2 million bilingual parallel pairs from news domain. It has been originally released publicly for Chinese-English machine translation evaluation.[1]

ParaNMT Dataset. Following previous work [Chen *et al.*, 2019], we used this paraphrase dataset [Wieting and Gimpel, 2018] to evaluate model performance for controllable paraphrase generation. For fair comparison, we used the same training data (500K sentence pairs) as Chen *et al.* [2019]. The manually annotated 800 instances created by Chen *et al.* [2019] were used as our test set, and 500 for development set.

[1] http://nlp.nju.edu.cn/cwmt-wmt/

Model	BL↑	R-1↑	R-2↑	R-L↑	MET↑	ST↓
Baseline model						
Baseline (500K)	12.2	44.8	20.3	46.8	23.8	8.6
The proposed model trained on bilingual parallel data						
Ours (0.5M)	8.7	39.3	14.9	41.7	20.5	9.8
Ours (1.0M)	11.0	43.1	18.2	45.5	22.8	8.9
Ours (1.5M)	11.6	43.7	19.1	46.2	23.3	8.4
Ours (2.0M)	**12.0**	**44.3**	**19.3**	**46.6**	**23.6**	**8.2**
Fine-tuning on paraphrase data						
Ours (+50K)	12.9	45.8	19.6	46.8	23.9	8.1
Ours (+100K)	13.3	45.9	20.6	48.2	25.6	7.6
Ours (+200K)	13.6	46.3	21.2	48.9	25.9	7.3
Ours (+500K)	**14.3**	**47.3**	**22.9**	**49.3**	**26.2**	**6.6**
Previous work [Chen *et al.*, 2019]						
VGVAE	3.5	24.8	7.3	29.7	12.6	10.6
VGVAE + WPL	4.5	26.5	8.2	31.5	13.3	10.0
VGVAE + LC	3.3	24.0	7.2	29.4	12.5	9.1
VGVAE + WN	13.0	43.2	20.2	47.0	23.8	6.8
VGVAE + all	13.6	44.7	21.0	48.3	24.8	6.7

Table 1: Results on controllable paraphrase generation. BL: BLEU, R: ROUGE, MET: METEOR, ST: Syntactic tree edit distance.

Quora Dataset. This dataset is a paired paraphrase dataset in question domain. It consists of 150K paraphrase pairs. Following previous work [Hasan *et al.*, 2016; Gupta *et al.*, 2018], we used 100K and 4K pairs for training and testing, and the remaining pairs as development set, respectively.

MSCOCO Dataset. This dataset [Lin *et al.*, 2014] contains human-annotated captions of over 120K images. Each image contains five captions from five different annotators. Following Prakash *et al.* [2016], we obtained a collection of 330K instances for training and 20K instances for testing.

6.2 Model Configuration

We used Adam [Kingma and Ba, 2014] for optimization. We set the initial learning rate to 5e-4. We set the mini-batch size to 50 for each training corpus. The size of cross-lingual word embeddings and the hidden states of the encoders and decoders were set 512. The size of the latent syntactic variable were 256. We trained the cross-lingual embeddings on the CWMT dataset. For the VAE training, following Bowman *et al.* [2015], we set the weight λ_2 to zero at the start of training, and gradually increased this weight to 1 as training progressed. We set λ_1 to 1 and λ_3 to 1e-5.

6.3 Automatic Evaluation Metrics

Semantic Accuracy. Following previous work [Prakash *et al.*, 2016; Hasan *et al.*, 2016], we used well-known automatic evaluation metrics: BLEU (BL), ROUGE (R) and METEOR (MET). Pervious studies have shown that these metrics perform well in evaluating generated paraphrases.

Syntactic Similarity. To measure the syntactic similarity, we reported the syntactic tree (ST) edit distance. We computed the tree edit distance between constituency parse trees after removing word tokens following Chen *et al.* [2019].

6.4 Controllable Paraphrase Generation

Experiment Settings. We trained the proposed model in the setting that the input and output is the same language. The model trained in this way was used as the baseline model. We then trained our model on the bilingual corpus (CWMT Chinese-English Dataset) to compare against the baseline model. We also analyzed the effect of using different amount of training data. Additionally, we investigated the performance of fine-tuning our bilingually trained model on the condition that a small paraphrase dataset is available. Finally, we compared the proposed method with previous work.

Model	BL↑	R-1↑	R-2↑	R-L↑	MET↑	ST↓
MT	10.4	41.6	17.1	43.3	20.5	9.6
Ours	**11.0**	**43.1**	**18.2**	**45.5**	**22.8**	**8.9**

Table 2: Comparison to CPG models trained on MT-generated paraphrase data.

Model	BL↑	R-1↑	R-2↑	R-L↑	MET↑	ST↓
Ours	11.0	43.1	18.2	45.5	22.8	8.9
- KL loss	6.7	34.3	10.9	35.9	17.8	11.8
- Adv loss	9.4	40.4	14.5	42.9	21.7	9.8
- Cycle	6.4	36.6	10.3	37.7	18.5	11.4
- Masking	5.4	31.5	9.4	34.4	16.0	11.6

Table 3: Effect of each component.

Results. The results are shown in Table 1. It can be seen that our model achieves competitive performance on the CPG task, even without using any annotated paraphrase training data. For the baseline model, please notice that we used the same training data as the model VGVAE + WN [Chen *et al.*, 2019], but we did not use external POS tags to preprocess syntactic inputs, and only used masking scheme.

Compared to the baseline model trained on paraphrase data, the performance of our model trained on bilingual data is surprisingly promising, which is quite comparable to the baseline model in terms of the six evaluation metrics. The results with different amount of bilingual training data show that our method can obtain a good performance even if the training data is relatively small, and suggest that more training data can further improve the quality of generated paraphrases. This validates that our proposed model is able to learn to generate syntactically-controllable paraphrases from bilingual parallel data.

Fine-tuning our model on a small amount of available paraphrase data, we can gain further improvements. When compared to the baseline trained on 500K paraphrase data, fine-tuning our model on only 100K (20%) paraphrase data can obtain the same performance. With more paraphrase data, the performance of our model can be further steadily improved on all evaluation metrics. Finally, when all 500k paraphrase pairs are used to fine tune our model, it reaches the performance level that is substantially better than the previous work [Chen *et al.*, 2019] on all evaluation metrics. For the challenging syntactically-controllable paraphrase generation task, our initial success of using off-the-shelf bilingual data shows a promising direction for paraphrase generation that does not need a large scale of annotated paraphrase data.

Comparison to Model Trained on MT-Generated Paraphrase Data. Following Wieting and Gimpel [2018], we generated paraphrase data via neural machine translation (NMT). We spilt the bilingual data into two parts, 1M for training a Transformer NMT system [Vaswani *et al.*, 2017], and 1M for using the trained NMT model to construct paraphrase data. We translated Chinese sentences into English, and paired the translations and original English sentences in-

Sem	extensive consultation has taken place over the past year.
Syn	there are our own troops like little ants.
Ref	there have been extensive consultations over the past year.
Gen	there are many extensive consultations in the past year.
Sem	it is hard for me to imagine where they could be hiding it underground.
Syn	they ca n't imagine when he 'll be able to walk.
Ref	i ca n't imagine where they could be hiding it underground.
Gen	i ca n't imagine where they could hide it from underground.

Table 4: Examples of generating controllable paraphrases from the model trained only on bilingual parallel data. Sem: semantic input, Syn: syntactic input, Ref: reference, Gen: generated paraphrase.

Model	BL-1↑	R-1↑	R-2↑	MET↑
Baseline model				
Baseline (100K)	51.2	57.0	31.6	26.7
The proposed model trained on bilingual parallel data				
Ours (0.5M)	37.7	47.8	18.3	18.8
Ours (1.0M)	38.6	50.1	20.9	20.0
Ours (1.5M)	41.2	51.3	21.4	20.9
Ours (2.0M)	**45.3**	**52.7**	**23.2**	**22.4**
Fine-tuning on paraphrase data				
Ours (+10K)	49.4	57.8	29.8	25.5
Ours (+30K)	50.8	59.8	32.3	26.7
Ours (+50K)	52.0	60.9	33.4	27.4
Ours (+100K)	**53.9**	**62.2**	**35.2**	**28.5**
Previous work				
VAE-SVG (50K)	17.1	-	-	21.3
EDD-LG (50K)	41.1	-	-	20.1
VAE-SVG (100K)	22.5	-	-	24.6
EDD-LG (100K)	45.7	-	-	23.1

Table 5: Results on the Quora dataset.

to paraphrases. The paraphrase data generated in this way were used to train our model. As shown in Table 2, the model trained on the bilingual corpus is better than the model trained on MT-generated paraphrase data.

Ablation Study. To better understand the impact of each component of our model on learning controllable paraphrase generation from bilingual data, we conducted ablation study. The results are shown in Table 3. In this experiment, we used 1M bilingual corpus as training data.

When the KL loss for building the syntactic space is removed, the performance is obviously degraded. We also observe that disentangling the content and style space via the proposed adversarial discriminator is important for the model to learn syntactically-controllable generation. Without the cycle learning, the performance also substantially drops. The masking technique has the largest impact on performance as it successfully prevents the model from just copying.

Generated Examples. To take a deep look into the proposed model only relying on bilingual data, we manually analyzed some examples, as shown in Table 4.

We can see that the proposed model produces fairly good samples in terms of both closeness in meaning and diversity in expressions. Meanwhile, our model trained only on bilingual data can effectively generate meaning-preserved paraphrases in the syntactic form similar to syntactic inputs.

6.5 General Paraphrase Generation

In order to study our model in learning PG from bilingual data, we conducted the second group of experiments without the controllable nature by omitting the part of the style encoder.

Model	BL-1↑	R-1↑	R-2↑	MET↑
Baseline model				
Baseline (300K)	35.6	38.9	14.2	15.8
The proposed model trained on bilingual parallel data				
Ours (0.5M)	6.0	18.8	5.0	6.1
Ours (1.0M)	8.5	20.9	5.4	6.3
Ours (1.5M)	10.2	21.1	5.6	6.8
Ours (2.0M)	**11.2**	**22.1**	**6.2**	**7.4**
Fine-tuning on paraphrase data				
Ours (+50K)	34.6	38.4	13.9	15.1
Ours (+100K)	36.0	38.7	14.1	15.6
Ours (+200K)	37.5	39.2	14.6	16.2
Ours (+300K)	**37.7**	**39.8**	**15.2**	**16.5**

Table 6: Results on the MSCOCO dataset.

Results. The experiment results on the Quora dataset are shown in Table 5. We compared our model with previous methods VAE [Gupta *et al.*, 2018] and EDD-LG [Patro *et al.*, 2018] that are built on large annotated paraphrase data (100K). As we can see, the proposed model obtains very strong results that are comparable to the previous state-of-the-art models trained on large paraphrase data. The fine-tuning results on smaller paraphrase data show that our model with only 30K (30%) paraphrase data outperforms the state-of-the-art models trained on 100K paraphrase data. With the increasing of the amount of paraphrase data, the performance is further improved.

The experiment results on the MSCOCO dataset are displayed in Table 6. The MSCOCO is an image caption dataset. Since different annotators may focus on different objects in the same image, the descriptions created by them may be quite different in meaning. Thus, the model trained on bilingual data cannot capture these differences. However, our model still obtains promising performance when it is fine-tuned with a small annotated data.

As both the MSCOCO and Quora dataset are smaller than the ParaNMT dataset, the results here further demonstrate the effectiveness of our method in low-resource scenarios.

7 Conclusion

In this work, we have presented a novel method to learn controllable paraphrase generation from bilingual data. We model semantic content and syntactic style via two shared encoders and one adversarial discriminator. Experiments and analyses disclose that the proposed method can effectively learn to generate high-quality meaning-preserved paraphrases in the syntactic forms controlled by given exemplars. This allows us to significantly reduce the amount of needed manually annotated paraphrase data, and makes it easier to develop a high-quality paraphrase generation system with large-scale available bilingual corpora.

Acknowledgments

The present research was supported by the National Nature Science Foundation of China (No. 61876198, 61976015, 61976016, 61370130 and 61473294), and also supported by the Beijing Municipal Natural Science Foundation (No. 4172047), and the International Science and Technology Cooperation Program of China (No. K11F100010).

References

[Artetxe *et al.*, 2017] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, 2017.

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

[Bowman *et al.*, 2015] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

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

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the NAACL*, pages 4171–4186, 2019.

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

[Ficler and Goldberg, 2017] Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, 2017.

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

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

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

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

[John *et al.*, 2018] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for text style transfer. *arXiv preprint arXiv:1808.04339*, 2018.

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

[Kingma *et al.*, 2014] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[Lample *et al.*, 2018] Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Y-Lan Boureau, et al. Multiple-attribute text rewriting. 2018.

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

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

[Prakash *et al.*, 2016] Aaditya Prakash, Sadid A Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Neural paraphrase generation with stacked residual lstm networks. *arXiv preprint arXiv:1610.03098*, 2016.

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

[Wang *et al.*, 2019] Wentao Wang, Zhiting Hu, Zichao Yang, Haoran Shi, Frank Xu, and Eric Xing. Toward unsupervised text content manipulation. *arXiv preprint arXiv:1901.09501*, 2019.

[Wieting and Gimpel, 2018] John Wieting and Kevin Gimpel. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, 2018.

[Wiseman *et al.*, 2018] Sam Wiseman, Stuart Shieber, and Alexander Rush. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187, 2018.

[Zhao *et al.*, 2018] Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on EMNLP*, pages 3164–3173, 2018.

[Zhou *et al.*, 2018] Zhong Zhou, Matthias Sperber, and Alex Waibel. Paraphrases as foreign languages in multilingual neural machine translation. *arXiv preprint arXiv:1808.08438*, 2018.