A Semantically Consistent and Syntactically Variational Encoder-Decoder Framework for Paraphrase Generation

Wenqing Chen, Jidong Tian, Liqiang Xiao, Hao He*, Yaohui Jin*
MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
State Key Lab of Advanced Optical Communication System and Network,
Shanghai Jiao Tong University
{wenqingchen, hehao, jinyh}@sjtu.edu.cn

Abstract

Paraphrase generation aims to generate semantically consistent sentences with different syntactic realizations. Most of the recent studies rely on the typical encoder-decoder framework where the generation process is deterministic. However, in practice, the ability to generate multiple syntactically different paraphrases is important. Recent work proposed to cooperate variational inference on a target-related latent variable to introduce the diversity. But the latent variable may be contaminated by the semantic information of other unrelated sentences, and in turn, change the conveyed meaning of generated paraphrases. In this paper, we propose a semantically consistent and syntactically variational encoder-decoder framework, which uses adversarial learning to ensure the syntactic latent variable be semantic-free. Moreover, we adopt another discriminator to improve the word-level and sentence-level semantic consistency. So the proposed framework can generate multiple semantically consistent and syntactically different paraphrases. The experiments show that our model outperforms the baseline models on the metrics based on both n-gram matching and semantic similarity, and our model can generate multiple different paraphrases by assembling different syntactic variables.

1 Introduction

Paraphrase generation is a longstanding problem in Natural Language Processing (NLP) (McKeown, 1983), which aims to generate semantically consistent sentences for a given sentence with different syntactic realizations. The task is not only an important building block for many text generation systems such as question answering (Buck et al., 2018; Dong et al., 2017), machine translation (Cho et al., 2014), but also beneficial to some NLP tasks such as semantic parsing (Su and Yan, 2017), sentence-level representation learning (Patro et al., 2018), data augmentation (Kumar et al., 2019).

Neural network-based methods (Prakash et al., 2016; Gupta et al., 2018; Li et al., 2018; Fu et al., 2019) have shown great progress on paraphrase generation. The models mainly rely on the sequence-to-sequence (seq2seq) learning framework (Sutskever et al., 2014) with typical encoder-decoders, which are relatively deterministic during the testing stage. Generally, the models will select the best result through the beam search but are not able to produce multiple paraphrases in a principled way (Gupta et al., 2018). Due to the nature of beam-search, the quality of k-th variant will be worse than the first variant.

In practice, the ability to generate multiple high-quality and diverse paraphrases is an important characteristic of text generation systems. A *target-oriented seq2seq model* is applaudable to achieve this goal. For example, Gupta et al. (2018) applied variational inference (Kingma and Welling, 2014) on a target-related latent variable $\boldsymbol{z}$. During testing, the model can sample multiple latent variables $\boldsymbol{z}$ from a prior distribution to generate multiple different paraphrases. But the remained problem is that $\boldsymbol{z}$ may be contaminated by the semantic information of other unrelated sentences in the training set, leading to an unexpected semantic change of the generated sentences.

In this paper, we propose to constrain the target-related latent variable $\boldsymbol{z}$ to contain merely the syntactic information. To achieve this goal, we introduce a syntactic encoder to extract $\boldsymbol{z}_{\text{syn}}$ from the target $\boldsymbol{y}$, and

* Corresponding authors

Proceedings of the 28th International Conference on Computational Linguistics, pages 1186–1198
Barcelona, Spain (Online), December 8-13, 2020

develop a discriminator with adversarial learning to ensure z_{syn} is semantic-free. The idea is inspired by (Bao et al., 2019), which disentangled the latent space of variational autoencoder (VAE) into semantic and syntactic spaces. But they considered the bag of words (BOWs) as the semantic information for adversarial training. This is not optimal because human-generated paraphrases can use quite different words but still express the same meaning. Instead, our model is data-driven. We do not constrain the semantic variables to be syntax-free, as the syntactic information entangled in the semantic variables will be overwritten by the target-oriented syntactic variables.

Types	Sentences	Word Distance	Semantic Distance
Gold Reference	$\mathcal{S}_r$: It is an **excellent film**!	-	-
More Penalized	$\mathcal{S}_a$: It is an **easy way**!	2	$\mathcal{D}$
Less Penalized	$\mathcal{S}_b$: It is an **awesome movie**!	2	$<\mathcal{D}$

Table 1: Illustration of the problem of MLE. The sentence S_r is the reference, and the rest sentences are two generated samples. $\mathcal{S}_a$ and $\mathcal{S}_b$ have the same word distance to $\mathcal{S}_r$, but $\mathcal{S}_b$ is semantically similar to $\mathcal{S}_r$. MLE will equally penalize the phrases "easy way" and "awesome movie" because they are non-target.

When considering semantic consistency, there exists another problem in many text generation models that maximum likelihood estimation (MLE) which is implemented by the cross-entropy function will penalize all the non-target words. An example is shown in Table 1. The cross-entropy function will equally penalize the two generated sentences $\mathcal{S}_a$ and $\mathcal{S}_b$ because both of them have two words not match the gold ones. But the semantics of them are quite different. It means that MLE captures the word distance well but does not precisely reflect the semantic distance. Our proposition is that sentences with larger semantic distance should be more penalized. We develop another discriminator, which determines whether the generated sentences are semantically consistent with the references. Unlike the discriminator for the latent variable z_{syn}, this discriminator needs to have access to the sampled tokens, which will cause the non-differentiable problem. We adopt Gumbel-softmax (Jang et al., 2017; Maddison et al., 2017) to make the model end-to-end differentiable. And we introduce two losses to measure both word-level and sentence-level semantic consistency.

The experiments on two datasets show that our model yields competitive results over other baseline models, and can generate multiple syntactically different and semantically consistent paraphrases. The main contributions of this work are as follows:

- We propose a target-oriented seq2seq framework that involves different syntactic variables to generate multiple different paraphrases.
- Our method not only increases the syntactic diversity with variational inference but also improves the word-level and sentence-level semantic consistency for the generated paraphrases.
- The experiments use metrics based on both n-gram matching and semantic similarity, and demonstrate the effectiveness of our model.

2 Related Work

Recently, many neural network-based models are proposed for paraphrase generation and can be categorized into three groups: reconstruction-based learning, typical seq2seq learning, and target-oriented seq2seq learning.

Reconstruction-based Learning. The first category of studies mainly deals with paraphrase generation in an unsupervised manner, which adds constraints on language models (LMs) including RNN-LM (Mikolov et al., 2010) or VAE (Bowman et al., 2016). Kovaleva et al. (2018) introduced a similarity-based reconstruction loss to VAE which considered similarities between words in the embedding space. Miao et al. (2019) introduces three kinds of constraints on RNN-LM including keywords matching, word embedding similarity, and skip-thoughts similarity. However, the similarity-based losses could not guarantee the semantic consistency between two words. For example, the words "good", "great" and "bad" are all close in the embedding space because they appear in similar contexts. Recently, an intuitive approach was proposed which disentangled the latent space of VAE into syntactic and semantic spaces (Bao et al., 2019). In their model, the constituency parse tree was used to supervise the syntactic latent variable, and the BOWs were used to supervise the semantic latent variable. Although the proposal of disentanglement is promising, supervision with BOWs is not optimal because paraphrases are possible to use quite different words and still convey the same meaning.

Typical Seq2seq Learning. The second category of studies considered paraphrase generation as a typical seq2seq task with parallel data. Prakash et al. (2016) proposed to use a seq2seq model for paraphrase generation with residual stack LSTM, and still performs as a strong baseline (Fu et al., 2019). Recent studies improved seq2seq models by involving some efficient mechanisms such as copy and constrained decoding (Cao et al., 2017), inverse reinforcement learning (Li et al., 2018), decomposition of phrase-level and sentence-level patterns (Li et al., 2019), and content planning with latent bag of words (Fu et al., 2019). When a sentence has multiple paraphrases in training data, these models will convert them into multiple pairwise sentences. From the perspective of probability modeling, these studies maximize the log conditional probability $\sum_{i=1}^{k} \log p(\boldsymbol{y}^i|\boldsymbol{x})$ where $\boldsymbol{x}$ denotes the original sentence and $\boldsymbol{y}^i$ is the i-th sentence among k paraphrases.

Target-oriented Seq2seq Learning. Compared with the second category of studies, the third included the target information which substantially maximized the log probability $\sum_{i=1}^{k} \log p(\boldsymbol{y}^i|\boldsymbol{x}, \boldsymbol{z}^{\boldsymbol{y}^i})$ where $\boldsymbol{z}^{\boldsymbol{y}^i}$ conveyed the information of target $\boldsymbol{y}^i$. Apparently, there was a train-test discrepancy because $\boldsymbol{z}^{\boldsymbol{y}^i}$ was not available during testing. Gupta et al. (2018) tackled the issue by a combination of the seq2seq architecture with VAE which allowed $\boldsymbol{z}^{\boldsymbol{y}^i}$ to sample from a prior distribution. The remained problem is that $\boldsymbol{z}^{\boldsymbol{y}^i}$ may contain semantic information of other unrelated sentences, which is possible to mislead the model. Ideally, for paraphrase generation, $\boldsymbol{z}^{\boldsymbol{y}^i}$ is expected to only convey the syntactic information. Kumar et al. (2020) implicitly tackled this problem by focusing on a slightly different task, the syntactic-guided controlled paraphrase generation, which inputted an exemplar to tell the syntactic information. As a result, the train-test discrepancy does not exist in the controlled task. However, for the traditional paraphrase generation task, constraining $\boldsymbol{z}^{\boldsymbol{y}^i}$ is still a problem.

3 Background

3.1 Variational Autoencoder

Before introducing our models, we briefly review the architecture of VAE (Kingma and Welling, 2014), a generative model which allows to generate high-dimensional samples from a continuous space. In the probability model framework, the probability of data $\boldsymbol{x}$ can be computed by:

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z})\mathrm{d}\boldsymbol{z} = \int p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})\mathrm{d}\boldsymbol{z} \tag{1}$$

Since this integral is unavailable in closed form or requires exponential time to compute (Blei et al., 2016), it is approximated by maximizing the evidence lower bound (ELBO):

$$\log p_\theta(\boldsymbol{x}) \geq \mathrm{ELBO} = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})) \tag{2}$$

where $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ denotes the generator with parameters θ and $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is obtained by an encoder with parameters ϕ, and $p(\boldsymbol{z})$ is a prior distribution, for example, a Gaussian distribution. And $\mathrm{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler (KL) Divergence between the two distributions. Moreover, a previous work proposed β-VAE (Higgins et al., 2017) to use a weight β for the KL divergence. This approach was considered as a baseline for paraphrase generation (Fu et al., 2019).

3.2 Continuous Approximation

When a text generation model involves the process of sampling words and expecting a reward from a discriminator or an evaluator, it will suffer from the non-differentiable problem due to the discrete nature of texts. Recently, many studies use reinforcement learning (RL) (Yu et al., 2017; Lin et al., 2017; Guo et al., 2018; Li et al., 2018) or Gumbel-softmax (Jang et al., 2017; Maddison et al., 2017; Yang et al., 2018; Nie et al., 2019) to overcome the problem. In our model, we use Gumbel-softmax because it makes models end-to-end differentiable, improving the stability and speed of training over RL (Chen et al., 2018).

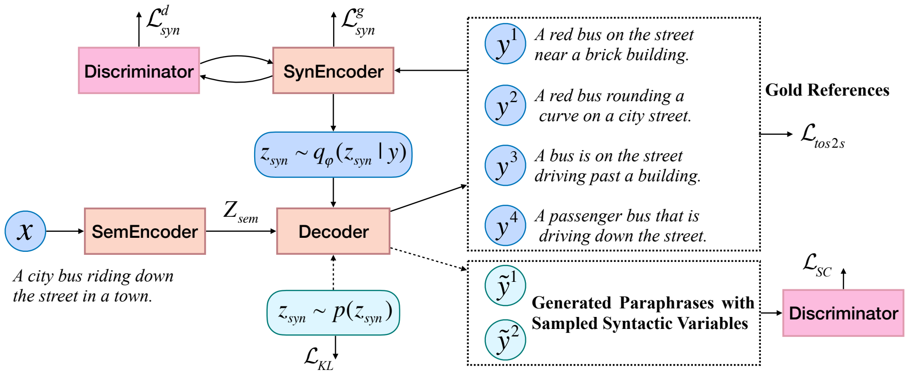
Assuming that the model outputs a logit value $\boldsymbol{o}_t$ when generating a sentence at tth timestep. A softmax function is used to produce probability $\boldsymbol{p}_t$ over the vocabulary set:

$$\boldsymbol{p}_t = \text{softmax}(\boldsymbol{o}_t) \tag{3}$$

Traditionally, a token w_t will be sampled from $\boldsymbol{p}_t$ with multinomial function or the argmax operation, both of which are non-differentiable. Gumbel-softmax uses a re-parameter trick by:

$$\widetilde{\boldsymbol{p}}_t = \text{softmax}((\boldsymbol{o}_t + \boldsymbol{g})/\tau) = \text{Gumbel-softmax}(\boldsymbol{p}_t; \tau) \tag{4}$$

where $\boldsymbol{g}$ samples from $\text{Gumbel}(0, 1)$ and τ is the temperature. When $\tau \rightarrow 0$, $\widetilde{\boldsymbol{p}}_t$ is approximated to the one-hot representation of the sampled token w_t. This process is a continuous approximation to the multinomial sampling, and we denote it by Gumbel-softmax$(\cdot)$ in the following sections.



A city bus riding down the street in a town.

A red bus on the street near a brick building.

A red bus rounding a curve on a city street.

A bus is on the street driving past a building.

A passenger bus that is driving down the street.

Figure 1: The architecture of the proposed model. The key idea is to generate multiple different paraphrases by involving different syntactic variables to the decoder.

4 Methodology

4.1 Semantically Consistent and Syntactically Variational Encoder-Decoder

Our method belongs to the category of *target-oriented seq2seq learning*, and aims to generate diverse paraphrases by involving the target-oriented syntactic information. We assume that each paraphrase should convey the same semantic with the original sentence, and multiple paraphrases have different syntaxes from each other. The architecture of our model is shown in Figure 1. The model contains a semantic encoder, a syntactic encoder, and a decoder with parameters ϕ, φ, and θ respectively. Given the sentence $\boldsymbol{x}$ and one of its paraphrases $\boldsymbol{y}$, the generation process can be defined as:

$$\boldsymbol{Z}_{\text{sem}} = \text{SemEncoder}(\boldsymbol{x}; \phi) \tag{5}$$

$$\boldsymbol{z}_{\text{syn}} = \text{SynEncoder}(\boldsymbol{y}; \varphi) \tag{6}$$

$$\boldsymbol{y} = \text{Decoder}(\boldsymbol{Z}_{\text{sem}}, \boldsymbol{z}_{\text{syn}}; \theta) \tag{7}$$

where $\boldsymbol{Z}_{\text{sem}}$ and $\boldsymbol{z}_{\text{syn}}$ denote the semantic and syntactic latent variables respectively. The variables $\boldsymbol{Z}_{\text{sem}}$ are a sequence of hidden states and $\boldsymbol{z}_{\text{syn}}$ is a vector representation. And our model can cooperate with the attention mechanism (Bahdanau et al., 2015). At each timestep, the decoder will produce a variable by the weighted sum of hidden states in $\boldsymbol{Z}_{\text{sem}}$ and then concatenate it with $\boldsymbol{z}_{\text{syn}}$ to decode each token. This process is modeling the probability $p\left(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_{\text{syn}}\right)$ instead of $p\left(\boldsymbol{y}|\boldsymbol{x}\right)$.

The key problem is how to constrain the syntactic variable $\boldsymbol{z}_{\text{syn}}$, as $\boldsymbol{y}$ is not available during testing. Similar to VAE, we apply variational inference on the variable $\boldsymbol{z}_{\text{syn}}$, which can be shown from the modeling of the likelihood $p(\boldsymbol{y}, \boldsymbol{x})$ and $p(\boldsymbol{y}|\boldsymbol{x})$:

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{x}) &= \int p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}_{\text{syn}}) \mathrm{d}\boldsymbol{z}_{\text{syn}} = \int p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_{\text{syn}}) p(\boldsymbol{x}|\boldsymbol{z}_{\text{syn}}) p(\boldsymbol{z}_{\text{syn}}) \mathrm{d}\boldsymbol{z}_{\text{syn}} \\ &= \int p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_{\text{syn}}) p(\boldsymbol{x}) p(\boldsymbol{z}_{\text{syn}}) \mathrm{d}\boldsymbol{z}_{\text{syn}}, \quad (\text{if } \ \boldsymbol{z}_{\text{syn}} \perp \boldsymbol{x}) \end{aligned} \tag{8}$$

where $\boldsymbol{z}_{\text{syn}} \perp \boldsymbol{x}$ means that $\boldsymbol{z}_{\text{syn}}$ is independent from $\boldsymbol{x}$. Since $p(\boldsymbol{x})$ can be moved outside of the integral, we divide both sides of Equation 8 by $p(\boldsymbol{x})$ to obtain the conditional probability:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \int p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_{\text{syn}}) p(\boldsymbol{z}_{\text{syn}}) \mathrm{d}\boldsymbol{z}_{\text{syn}} \tag{9}$$

$$\begin{aligned} \log p(\boldsymbol{y}|\boldsymbol{x}) \geq \text{ELBO} &= \mathop{\mathbb{E}}_{\boldsymbol{z}_{\text{syn}} \sim q_{\varphi}(\boldsymbol{z}_{\text{syn}}|\boldsymbol{y})} \left[\log p_{\theta,\phi} \left(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_{\text{syn}}\right)\right] - \text{KL}\left(q_{\varphi}(\boldsymbol{z}_{\text{syn}}|\boldsymbol{y}) \| p(\boldsymbol{z}_{\text{syn}})\right) \\ &= -\mathcal{L}_{\text{tos2s}}(\phi; \theta) - \mathcal{L}_{\text{KL}}(\varphi) \end{aligned} \tag{10}$$

where maximizing the log likelihood $\log p(\boldsymbol{y}|\boldsymbol{x})$ is approximated to maximize the ELBO. And $p_{\theta,\phi}\left(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}_{\text{syn}}\right)$ can be modeled by Equation 5 and 7, and the posterior $q_{\varphi}(\boldsymbol{z}_{\text{syn}}|\boldsymbol{y})$ is modeled by Equation 6. Then the first term of Equation 10 is a considered as the *target-oriented seq2seq* loss denoted by $\mathcal{L}_{\text{tos2s}}(\phi; \theta)$. The second term is the KL loss denoted by $\mathcal{L}_{\text{KL}}(\varphi)$.

4.2 Adversarial Learning for Syntactic Variables

There are two important assumptions to make Equation 8 $-$ 10 true: 1) $\boldsymbol{z}_{\text{syn}}$ is independent from $\boldsymbol{x}$; 2) $\boldsymbol{z}_{\text{syn}}$ contains merely the syntactic information of $\boldsymbol{y}$. Since $\boldsymbol{z}_{\text{syn}}$ is extracted from $\boldsymbol{y}$ by Equation 6, the first assumption is met if $\boldsymbol{z}_{\text{syn}}$ does not contain the information shared by $\boldsymbol{x}$ and $\boldsymbol{y}$, which is typically the semantic information. The second assumption also requires that $\boldsymbol{z}_{\text{syn}}$ does not contain the semantic information. Therefore, we use adversarial learning to derive semantic-free information for $\boldsymbol{z}_{\text{syn}}$.

Given $\boldsymbol{z}^x_{\text{syn}} \sim q_{\varphi}(\boldsymbol{z}^x_{\text{syn}}|\boldsymbol{x})$ and $\boldsymbol{z}^y_{\text{syn}} \sim q_{\varphi}(\boldsymbol{z}^y_{\text{syn}}|\boldsymbol{y})$ corresponding to the syntactic variables of the original sentence $\boldsymbol{x}$ and the paraphrase $\boldsymbol{y}$ respectively, we employ a discriminator with trainable weight $W_{\text{syn}} \in \mathbb{R}^{4d_{\text{syn}} \times c}$ where d_{syn} denotes the dimension of the syntactic variables and $c = 2$ means that it is a binary classification process. The probability of whether $\boldsymbol{z}^x_{\text{syn}}$ and $\boldsymbol{z}^y_{\text{syn}}$ contain same semantic information can be computed by:

$$\widetilde{p}_{\boldsymbol{x},\boldsymbol{y}} = \text{softmax}\left(W_{\text{syn}}\left[\boldsymbol{z}^x_{\text{syn}}, \boldsymbol{z}^y_{\text{syn}}, |\boldsymbol{z}^x_{\text{syn}} - \boldsymbol{z}^y_{\text{syn}}|, \boldsymbol{z}^x_{\text{syn}} \odot \boldsymbol{z}^y_{\text{syn}}\ \right]\right) \tag{11}$$

where $|\cdot|$ means taking the absolute value, $\odot$ denotes the element-wise multiplication, and $[,]$ denotes the concatenation operation. Moreover, we construct negative samples by randomly sampling a sentence $\overline{\boldsymbol{x}} \neq \boldsymbol{x}$ in the dataset. The predicted probability is denoted by $\widetilde{p}_{\overline{\boldsymbol{x}},\boldsymbol{y}}$. Then the loss of the discriminator is computed by:

$$\mathcal{L}^{\text{d}}_{\text{syn}}(\chi) = -p_{\text{pos}} \log \widetilde{p}_{\boldsymbol{x},\boldsymbol{y}} - p_{\text{neg}} \log \widetilde{p}_{\overline{\boldsymbol{x}},\boldsymbol{y}} \tag{12}$$

where $p_{\text{pos}} = [1, 0]$ and $p_{\text{neg}} = [0, 1]$ representing the labels for the positive pair $(\boldsymbol{x}, \boldsymbol{y})$ and the negative pair $(\overline{\boldsymbol{x}}, \boldsymbol{y})$ respectively. And χ denotes the parameters (W_{syn}) of the discriminator. Equation 12 means that the discriminator is trying to recognize the semantic information shared between $\boldsymbol{x}$ and $\boldsymbol{y}$. Then the syntactic encoder is considered as the generator to fool the discriminator by minimizing the loss:

$$\mathcal{L}^{\text{g}}_{\text{syn}}(\varphi) = -p_{\text{neg}} \log \widetilde{p}_{\boldsymbol{x},\boldsymbol{y}} \tag{13}$$

And the generator and discriminator play an adversarial game by minimizing $\mathcal{L}^{\text{g}}_{\text{syn}}(\varphi)$ and $\mathcal{L}^{\text{d}}_{\text{syn}}(\chi)$ alternatively. When combining the other losses, the objective of our model is:

$$\min_{\theta,\phi,\varphi}[\mathcal{L}_{\text{tos2s}}(\phi; \theta) + \mathcal{L}_{\text{KL}}(\varphi) + \mathcal{L}^{\text{g}}_{\text{syn}}(\varphi)] + \min_{\chi}[\mathcal{L}^{\text{d}}_{\text{syn}}(\chi)] \tag{14}$$

where the first term is the total loss for the generator and the second term is the loss for the discriminator.

4.3 Ensuring Semantic Consistency

There remains a train-test discrepancy where $\boldsymbol{z}_{\text{syn}} \sim q_{\varphi}(\boldsymbol{z}_{\text{syn}}|\boldsymbol{y})$ during training while $\boldsymbol{z}_{\text{syn}} \sim p(\boldsymbol{z}_{\text{syn}})$ during testing. Minimizing the KL divergence between $q_{\varphi}(\boldsymbol{z}_{\text{syn}}|\boldsymbol{y})$ and $p(\boldsymbol{z}_{\text{syn}})$ can help reduce the discrepancy, but does not provide end-to-end guarantee for the semantic consistency. Therefore, we further employ another discriminator D_{ψ} with parameters ψ, which consists of a sentence encoder, and a fully-connected neural network followed with the softmax function. For arbitrary two sentences represented by the sequences of one-hot vectors $\boldsymbol{u} \in \mathbb{R}^{T\times V}$ and $\boldsymbol{v} \in \mathbb{R}^{T\times V}$, the discriminator predicts the probability $\widetilde{p}_{\boldsymbol{u},\boldsymbol{v}} \in \mathbb{R}^{2}$ of whether two sentences are semantically consistent:

$$\widetilde{p}_{\boldsymbol{u},\boldsymbol{v}} = D_{\psi}(\boldsymbol{u},\boldsymbol{v}) \tag{15}$$

where T and V represent the maximum length of the sentences and the vocabulary size respectively. Traditionally, when $\boldsymbol{z}_{\text{syn}} \sim q_{\varphi}(\boldsymbol{z}_{\text{syn}}|\boldsymbol{y})$, the model minimizes $\mathcal{L}_{\text{tos2s}}(\phi;\theta)$ with MLE:

$$\max_{\boldsymbol{z}_{\text{syn}} \sim q_{\varphi}(\boldsymbol{z}_{\text{syn}}|\boldsymbol{y})} \sum_{t=1}^{T} \log p_{\theta,\phi}\left(\widetilde{y}_t = y_t|\boldsymbol{y}_{<t},\boldsymbol{x},\boldsymbol{z}_{\text{syn}}\right) \tag{16}$$

where $\widetilde{y}_t$ and y_t denote the predicted and referenced tokens respectively at t-th timestep, and $\boldsymbol{y}_{<t}$ denotes the sequence of tokens preceding y_t. However, when $\boldsymbol{z}_{\text{syn}} \sim p(\boldsymbol{z}_{\text{syn}})$, the syntactic information is different from that of $\boldsymbol{z}_{\text{syn}} \sim q_{\varphi}(\boldsymbol{z}_{\text{syn}}|\boldsymbol{y})$, and the predicted tokens is therefore not required to match all the tokens of $\boldsymbol{y}$. Instead, we assume that there is a set of semantically consistent words $W_c(y_t)$ with respect to y_t, using which will not change the conveyed meaning.

$$\max_{\boldsymbol{z}_{\text{syn}} \sim p(\boldsymbol{z}_{\text{syn}})} \sum_{t=1}^{T} \log p_{\theta,\phi}\left(\widetilde{y}_t \in W_c(y_t)|\boldsymbol{y}_{<t},\boldsymbol{x},\boldsymbol{z}_{\text{syn}}\right) \tag{17}$$

where the objective is to ensure the **word-level semantic consistency** (WSC). We construct a sequence of tokens represented by one-hot vectors $\hat{\boldsymbol{y}} = (\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, ..., \hat{\boldsymbol{y}}_T)$. The sentence is obtained by replacing a certain ratio (η) of tokens in $\boldsymbol{y}$ with predicted tokens $\widetilde{y}_t$ sampled from the predicted probability distribution $\widetilde{\boldsymbol{p}}_t \in \mathbb{R}^{V}$. The process can be described by:

$$\widetilde{\boldsymbol{p}}_t = p_{\theta,\phi}\left(\widetilde{y}_t|\boldsymbol{y}_{<t},\boldsymbol{x},\boldsymbol{z}_{\text{syn}}\right),\ \boldsymbol{z}_{\text{syn}} \sim p(\boldsymbol{z}_{\text{syn}}) \tag{18}$$

$$\hat{\boldsymbol{y}}_t = \begin{cases} \text{Gumbel-softmax}(\widetilde{\boldsymbol{p}}_t;\tau), & rand() < \eta \\ \text{one-hot}(y_t), & \text{otherwise} \end{cases} \tag{19}$$

where $rand()$ is a random function to sample numbers between 0 and 1 following the uniform distribution. Then the loss for word-level semantically consistency is computed by:

$$\mathcal{L}_{\text{wsc}} = -p_{\text{pos}} \log D_{\psi}(\hat{\boldsymbol{y}},\boldsymbol{x}) \tag{20}$$

Moreover, we further reduce the train-test discrepancy by reducing the exposure bias problem (Ranzato et al., 2016). We let each token in the sentence $\widetilde{\boldsymbol{s}}$ be generated conditioning on previously generated tokens instead of gold ones, and get a sentence-level feedback from the discriminator:

$$\max_{\boldsymbol{z}_{\text{syn}} \sim p(\boldsymbol{z}_{\text{syn}})} \log p_{\theta,\phi}\left(\widetilde{\boldsymbol{y}} \in S_c(\boldsymbol{y})|\boldsymbol{x},\boldsymbol{z}_{\text{syn}}\right) \tag{21}$$

$$\widetilde{\boldsymbol{y}}_t = \text{Gumbel-softmax}\left(p(\widetilde{y}_t|\widetilde{\boldsymbol{y}}_{<t},\boldsymbol{x},\boldsymbol{z}_{\text{syn}});\tau\right) \tag{22}$$

$$\mathcal{L}_{\text{ssc}} = -p_{\text{pos}} \log D_{\psi}(\widetilde{\boldsymbol{y}},\boldsymbol{x}) \tag{23}$$

where the objective is to ensure **sentence-level semantic consistency** (SSC). $S_c(\boldsymbol{y})$ denotes the set of semantically consistent sentences, and $\widetilde{\boldsymbol{y}} = (\widetilde{\boldsymbol{y}}_1, \widetilde{\boldsymbol{y}}_2, ..., \widetilde{\boldsymbol{y}}_T)$ denotes the sequence of generated tokens with one-hot representations. The discriminator will also include positive samples $(\boldsymbol{x}, \boldsymbol{y})$ and negative samples $(\overline{\boldsymbol{x}}, \boldsymbol{y})$ to learn to predict whether two sentences are semantically consistent:

$$\mathcal{L}_{\text{sc}}(\theta, \phi, \varphi, \psi) = \mathcal{L}_{\text{wsc}} + \mathcal{L}_{\text{ssc}} - p_{\text{pos}} \log D_\psi(\boldsymbol{x}, \boldsymbol{y}) - p_{\text{neg}} \log D_\psi(\overline{\boldsymbol{x}}, \boldsymbol{y}) \tag{24}$$

Then, the final objective can be computed by:

$$\min_{\theta,\phi,\varphi,\psi} [\mathcal{L}_{\text{tos2s}}(\phi; \theta) + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}(\varphi) + \lambda^{\text{g}}_{\text{syn}} \mathcal{L}^{\text{g}}_{\text{syn}}(\varphi) + \lambda_{\text{sc}} \mathcal{L}_{\text{sc}}(\theta, \phi, \varphi, \psi)] + \lambda^{\text{d}}_{\text{syn}} \min_{\chi} [\mathcal{L}^{\text{d}}_{\text{syn}}(\chi)] \tag{25}$$

where λ_{KL}, $\lambda^{\text{g}}_{\text{syn}}$, λ_{sc}, and $\lambda^{\text{d}}_{\text{syn}}$ are the hyperparameters to balance the losses in overall objective.

Dataset	Train	Valid	Test	N_{para}	L_{avg}	L_{95}	Vocab.
Quora	116,263	3,000	30,000	2	11.2	20	30,997
MSCOCO	78,733	4,050	40,504	5	11.3	16	27,801

Table 2: Statistics of two datasets. N_{para} represents the number of paraphrases in one sample. L_{avg} and L_{95} denote the average length of all sentences and the maximum length of 95% of sentences respectively.

5 Experiments

5.1 Datasets

Following previous work on paraphrase generation, we experiment on two datasets: Quora (Lin et al., 2014) [1] and MSCOCO[2]. The Quora dataset is originally developed for duplicated question detection which contains about 140k pairs of paraphrase and 260k pairs of non-paraphrase sentence pairs. We only use the paraphrase sentences and hold out 3k and 30k validation and test sets respectively. We set the maximum decoding length to be 20 which equals the maximum length of 95% of sentences. The MSCOCO dataset is originally developed for image captioning and each image has 5 captions. In our experiments, we randomly choose 1 of the 5 captions as the source and use the rest 4 captions as the targets. The original dataset contains about 80k and 40k samples in the train and test sets respectively. We randomly hold out about 4k samples from the train set as the validation set. The detailed statistics of the two datasets are shown in Table 2.

5.2 Evaluation and Settings

The evaluation of paraphrase generation remains an open issue. Most of previous studies (Prakash et al., 2016; Gupta et al., 2018; Li et al., 2018; Bao et al., 2019; Fu et al., 2019) adopt metrics based on n-gram matching, such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). To compare our model with them, we also report the n-gram metrics (1-4 grams in BLEU, 1-2 gram in ROUGE). However, we observe that they are not always sufficient to evaluate the semantic consistency because human-generated paraphrases have lower BLEU or ROUGE scores than machine-generated on the MSCOCO dataset (will be discussed in Section 5.3). Therefore, we further employ a metric BERTCS (Reimers and Gurevych, 2019) which computes the cosine similarity of sentence-level embeddings of fine-tuned BERT (Devlin et al., 2019). We choose the BERT-base model fine-tuned on SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018) datasets with mean-tokens pooling[3]. Moreover, since simply copying the source sentence is not an interesting model but definitely yields semantically consistent outputs, we evaluate the syntactic difference from the source sentence based on BLEU-ori (up to 4 grams) which were recently used to evaluate the reconstruction-based models (Miao et al., 2019; Bao et al., 2019).

Compared Models. We compare our models with three categories of existing methods introduced in Section 2. The reconstruction-based models for comparison include β-VAE (Higgins et al., 2017) with $\beta = 1e^{-3}$ and $\beta = 1e^{-4}$, and DSS-VAE (Bao et al., 2019). The typical seq2seq models include

[1] https://www.kaggle.com/c/quora-question-pairs/data

[2] http://cocodataset.org/

[3] https://github.com/UKPLab/sentence-transformers

vanilla seq2seq LSTM with (or without) the attention mechanism (Bahdanau et al., 2015), and LBOW-Topk which is the state-of-the-art (SOTA) model (Fu et al., 2019). The compared target-oriented seq2seq model is the variational encoder-decoder (VAE-SVG-eq) (Gupta et al., 2018). Since variational models can generate multiple paraphrases for a source sentence by sampling multiple latent variables, we can select the best one with highest BERTCS scores computed with the source sentence (not with the reference sentences because they not available in practice). This searching mechanism is also used in (Gupta et al., 2018) and is denoted by VarSearch in the following sections. We search 5 times for both VAE-SVG-eq and our model.

Hyperparameters. Word embeddings are 300-dimensional and initialized with GloVe (Pennington et al., 2014). The dimension of the encoders and the decoder are based on two-layer LSTM with 500 state size. The latent space dimension is also set to 500. We set a fixed temperature of $\tau = 0.01$ for Gumbel-softmax during training. The weights for different losses are $\lambda_{\text{KL}} = 0.2$ (with the annealing trick), $\lambda_{\text{syn}}^{\text{g}} = 0.5$, $\lambda_{\text{sc}} = 0.5$, and $\lambda_{\text{syn}}^{\text{d}} = 0.5$ respectively. The replacement ratio η for word-level semantic consistency is set to 0.5. The learning rate of all models is set to 5×10^{-4}. The batch size is set to 32. All models are trained for 15 epochs. We report the averaged metrics after the training process is repeated 3 times.

	B-1	B-2	B-3	B-4	R-1	R-2	R-L	BertCS	BLEU-ori (↓)
Copying (Positive)	62.56	48.23	38.80	31.17	63.64	37.53	60.94	85.00	100.00
Sampling (Negative)	17.69	4.96	2.40	1.13	18.09	1.50	17.73	31.70	-
β-VAE ($\beta = 10^{-3}$)	44.93	30.38	22.32	16.94	47.27	21.47	46.78	66.18	49.97
β-VAE ($\beta = 10^{-4}$)	48.90	34.09	25.32	19.29	51.46	25.09	50.72	72.11	59.49
DSS-VAE †	-	-	-	20.54	-	-	-	-	52.77
Seq2Seq	50.02	36.81	28.67	22.94	55.96	30.30	55.69	77.07	26.93
Seq2Seq-Att	51.77	38.42	30.01	24.02	57.56	31.47	57.16	78.88	30.87
LBOW-Topk †	**55.79**	**42.03**	32.71	26.17	58.79	34.57	56.43	-	-
VAE-SVG-eq (+ VarSearch)	50.33	36.89	28.74	23.06	56.44	30.12	56.02	77.82	**26.76**
SCSVED (ours)	54.02	40.67	32.29	26.04	58.76	33.77	58.93	81.01	33.59
SCSVED (+ VarSearch) (ours)	54.26	41.56	**33.36**	**27.37**	**60.28**	**35.26**	**59.83**	**81.60**	33.97

Table 3: The results on *Quora*. B-i and R-j stand for BLEU and ROUGE scores respectively. Larger values are better except that BLEU-ori prefers lower values. The symbol † means the cited results.

	B-1	B-2	B-3	B-4	R-1	R-2	R-L	BertCS	BLEU-ori (↓)
Copying (Positive)	65.74	44.56	29.78	19.85	37.32	12.08	33.01	72.27	100.00
Sampling (Negative)	34.39	11.75	4.38	1.81	17.38	1.45	14.30	20.02	-
β-VAE ($\beta = 10^{-3}$)	65.09	44.02	29.35	19.52	36.92	11.89	32.69	71.45	90.80
β-VAE ($\beta = 10^{-4}$)	65.29	44.19	29.48	19.63	37.02	11.93	32.77	71.69	92.30
Seq2Seq	71.68	51.50	36.08	25.21	39.75	14.64	36.00	70.53	**15.00**
Seq2Seq-Att	71.84	51.51	36.17	25.32	39.83	14.65	36.06	70.75	15.01
LBOW-Topk †	72.60	51.14	35.66	25.27	**42.08**	**16.13**	**38.16**	-	-
VAE-SVG-eq (+ VarSearch)	72.89	52.42	36.93	25.99	40.10	15.18	36.13	70.98	15.23
SCSVED (ours)	73.75	53.66	38.32	27.33	40.65	15.39	37.03	71.80	16.27
SCSVED (+ VarSearch) (ours)	**74.11**	**54.35**	**39.19**	**28.24**	40.90	15.70	37.33	**71.94**	16.44

Table 4: The results on *MSCOCO*. B-i and R-j stand for BLEU and ROUGE scores respectively. Larger values are better except that BLEU-ori prefers lower values. The symbol † means the cited results.

5.3 Main Results

Table 3 and 4 show the overall performance of different models. To understand what is an applaudable score on each metric, we do a preliminary experiment by designing a copying and a randomly sampling model, which can be considered as the upper and lower bound for metrics. Higher B-i, R-j, and BertCS scores represent better consistency with reference sentences. Lower BLEU-ori scores represent better syntactic differences from source sentences. The interesting finding on the MSCOCO dataset is that the source sentences, which are human-generated paraphrases with regard to reference sentences, have lower B-i and R-j scores than the machine-generated. The possible reason may be that humans will use diverse n-grams and still express the same meaning while machines prefer to use high-frequency n-grams. And BertCS scores confirm the high semantic consistency of human-generated paraphrases.

Generally, our model with variational search achieves competitive B-i, R-j scores, and the best BertCS scores on the Quora and MSCOCO datasets. Compared with the previous SOTA model LBOW-Topk, our model improves B-4 by 1.20 and 2.97 points on Quora and MSCOCO respectively. Compared with Seq2Seq-Att, our model improves B-4 and BertCS by 3.35 and 2.72 points respectively on Quora, and 2.92 and 1.19 points respectively on MSCOCO. When compared with variational models including β-VAE and VAE-SVG-eq, our model also outperforms them with a large margin. The reason may be that the sampled variational latent variables in their models contain semantic information, and lead to a change of the conveyed meaning. DSS-VAE which disentangles the semantic and syntactic representations outperforms β-VAE with an increase of B-4 and a decrease of BLEU-ori scores on Quora but does not outperform seq2seq models. It means that the disentanglement of the latent spaces is not sufficient to guarantee the decoder of VAE to generate semantically consistent sentences.

	Quora					MSCOCO				
	B-2	B-4	R-2	R-L	BertCS	B-2	B-4	R-2	R-L	BertCS
SCSVED	**40.67**	**26.04**	**33.77**	**58.93**	**81.01**	**53.66**	**27.33**	**15.39**	**37.03**	**71.80**
SCSVED - SSC	40.12	25.49	33.20	58.47	80.57	52.99	26.19	15.09	36.72	71.43
SCSVED - SSC - WSC	39.51	25.02	32.90	58.30	80.19	52.54	25.99	15.03	36.43	71.15
SCSVED - SSC - WSC - SynAdv	36.31	22.28	29.79	55.78	77.52	51.18	24.76	13.87	35.23	69.86
Seq2Seq-Att	38.42	24.02	31.47	57.16	78.88	51.51	25.32	14.65	36.06	70.75

Table 5: Results of the ablation study.

5.4 Ablation Study

To analyze which mechanisms are driving the improvements, we present an ablation study in Table 5. We eliminate sentence-level and word-level semantic consistency (SSC and WSC), syntactically adversarial learning (SynAdv) one by one, which results in three ablated models. Further eliminating the variational inference of syntactic variables yields the Seq2Seq-Att model. Generally, the three mechanisms are all influential. For example, eliminating the two semantic consistent losses leads to a total drop of BertCS by 0.82 and 0.65 points on Quora and MSCOCO respectively. When further eliminating SynAdv, the model has worse performance than Seq2Seq-Att. It demonstrates the importance of guaranteeing the syntactic variable to be semantic-free.

Models & Settings		Sentences	BertCS
Source		The male skateboarder **does a stunt** on the brown ramp.	-
References		A person skateboarding on a skate board ramp.	-
		A skateboarder is attempting to skate down a piece of metal.	-
Seq2Seq-Att	Top 1 of Beam-10	A man riding a skateboard up the side of a ramp.	77.12
	Top 2 of Beam-10	A man riding a skateboard on a ramp.	76.76
	Top 3 of Beam-10	A man riding a skateboard down a ramp.	76.32
SCSVED	$\boldsymbol{z}^i_{\text{syn}} \sim p(\boldsymbol{z}_{\text{syn}})$	A man on a skateboard **doing a trick** on a ramp.	82.77
	$\boldsymbol{z}^j_{\text{syn}} \sim p(\boldsymbol{z}_{\text{syn}})$	A man **doing a trick** on a skateboard.	78.69
	$\boldsymbol{z}^k_{\text{syn}} \sim p(\boldsymbol{z}_{\text{syn}})$	A man riding a skateboard on a ramp.	76.76

Table 6: An example of the generated sentences of the models on MSCOCO dataset.

5.5 Case Study

To help understand our model, we present a case study in Table 6. For the MSCOCO dataset, each image has multiple diverse captions. We show the source and two gold references for an image. After training, Seq2Seq-Att and our model both produce three paraphrases for the given source, and BertCS scores are presented to measure their semantic consistency with respect to the source sentence. Following traditional seq2seq models, we choose the top 3 results through the beam search for Seq2Seq-Att. The results show that the three generated sentences lack diversity. Different from Seq2Seq-Att, our model generates 3 paraphrases by sampling 3 different latent variables $z^i_{\text{syn}}, z^j_{\text{syn}}, z^k_{\text{syn}}$, which produces high-quality and diverse paraphrases. However, it is worth noting that the variable z_{syn} is data-driven, which means that the information in z_{syn} may not perfectly match human-defined syntaxes. Moreover, the references may contain additional information than the source, which is not statistically easy to learn. This phenomenon can explain why the BLEU and ROUGE scores of the references are lower than machine-generated sentences in Table 5. However, the key information is preserved.

6 Conclusion

In this paper, we propose a semantically consistent and syntactically variational encoder-decoder framework for paraphrase generation, which enables the model to generate different paraphrases according to different syntactic variables. We first introduce an adversarial learning method to ensure the variational syntactic variable not be contaminated by semantic information, and further develop word-level and sentence-level objectives to ensure the generated sentences be semantic consistent. The experiments show that our model yields competitive results and can generate high-quality and diverse paraphrases.

Acknowledgements

The authors sincerely thank the anonymous reviewers for their constructive comments on the improvement of this paper. This work was supported by National Key Research and Development Program of China under Grant 2018YFC0830400.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-Yu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6008–6019. Association for Computational Linguistics.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. 2016. Variational inference: A review for statisticians. *CoRR*, abs/1601.00670.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In Yoav Goldberg and Stefan Riezler, editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 10–21. ACL.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2018. Ask the right questions: Active question reformulation with reinforcement learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2017. Joint copying and restricted generation for paraphrase. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3152–3158. AAAI Press.

Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. 2018. Adversarial text generation via feature-mover's distance. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 4671–4682.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 875–886. Association for Computational Linguistics.

Yao Fu, Yansong Feng, and John P. Cunningham. 2019. Paraphrase generation with latent bag of words. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13623–13634.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5141–5148. AAAI Press.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5149–5156. AAAI Press.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Olga Kovaleva, Anna Rumshisky, and Alexey Romanov. 2018. Similarity-Based Reconstruction Loss for Meaning Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4875–4880, Brussels, Belgium, October. Association for Computational Linguistics.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha P. Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN,* *USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3609–3619. Association for Computational Linguistics.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha P. Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *CoRR*, abs/2005.08417.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3865–3878. Association for Computational Linguistics.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019. Decomposable neural paraphrase generation. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3403–3414. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.

Kevin Lin, Dianqi Li, Xiaodong He, Ming-Ting Sun, and Zhengyou Zhang. 2017. Adversarial ranking for language generation. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 3155–3165.

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

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Kathleen R. McKeown. 1983. Paraphrasing questions using given and new information. *American Journal of Computational Linguistics*, 9(1):1–10.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. CGMH: constrained sentence generation by metropolis-hastings sampling. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6834–6842. AAAI Press.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, editors, *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048. ISCA.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Badri Narayana Patro, Vinod Kumar Kurmi, Sandeep Kumar, and Vinay P. Namboodiri. 2018. Learning semantic sentence embeddings using sequential pair-wise discriminator. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2715–2729. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek V. Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual LSTM networks. In Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad, editors, *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2923–2934. ACL.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Yu Su and Xifeng Yan. 2017. Cross-domain semantic parsing via paraphrasing. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1235–1246. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 7298–7309.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 2852–2858. AAAI Press.