# DivGAN: Towards Diverse Paraphrase Generation via Diversified Generative Adversarial Network

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

## Abstract

Paraphrases refer to texts that convey the same meaning with different expression forms. Traditional seq2seq-based models on paraphrase generation mainly focus on the fidelity while ignoring the diversity of outputs. In this paper, we propose a deep generative model to generate diverse paraphrases. We build our model based on the conditional generative adversarial network, and propose to incorporate a simple yet effective diversity loss term into the model in order to improve the diversity of outputs. The proposed diversity loss maximizes the ratio of pairwise distance between the generated texts and their corresponding latent codes, forcing the generator to focus more on the latent codes and produce diverse samples. Experimental results on benchmarks of paraphrase generation show that our proposed model can generate more diverse paraphrases compared with baselines.

## 1 Introduction

The task of paraphrase generation refers to rewriting a given sentence to a new paraphrase sentence, which requires that the generated sentence and input sentence are different in expression form, but have the same expressed meaning. Paraphrase generation is a fundamental task of natural language processing (NLP). The technique of paraphrase generation has been widely used in many downstream applications, such as information retrieval, question answering, machine translation, and so on.

Early works on paraphrase generation mainly focus on rule-based (McKeown, 1983; Meteer and Shaked, 1988), grammar-based (Narayan et al., 2016), lexicon-based (Bolshakov and Gelbukh, 2004; Kauchak and Barzilay, 2006), and statistical machine translation (SMT)-based methods (Kauchak and Barzilay, 2006; Zhao et al., 2009). Recently, with the release of large-scale paraphrase datasets, sequence-to-sequence (seq2seq) models (Prakash et al., 2016; Li et al., 2019; Kajiwara, 2019; Li et al., 2018; Gupta et al., 2018; Shakeri and Sethy, 2019; Yang et al., 2019) have become the dominant technique in the field of paraphrase generation.

Paraphrases should be diversified in nature, i.e., an input sentence can correspond to multiple plausible paraphrases. Traditional seq2seq-based methods tend to generate highly similar outputs since the maximum likelihood estimation (MLE)-based objective function mostly cares about the validity rather than the diversity of outputs. Some works introduce control mechanisms over seq2seq models to produce diverse outputs (Iyyer et al., 2018; Park et al., 2019; Chen et al., 2019). However, the templates or exemplars in control mechanism cannot cover all the possibility of paraphrase, and the introduction of control mechanism is inflexible. Xu et al. (2018b) propose to use a shared decoder with different decoder embeddings to generate different outputs, but the decoder embeddings are not explicitly encouraged and learned to produce different outputs.

Generative models, such as Variational Autoencoder (VAE) (Kingma and Welling, 2014) and Generative Adversarial Network (GAN) (Goodfellow et al., 2014), which learn distributions over the latent space, can generate diverse outputs. In this paper, we build a new framework on top of the conditional GAN (Mirza and Osindero, 2014) to generate diverse paraphrases. To get multiple outputs, the generative models often take an additional random vector (latent code) as inputs, where the noise vector is responsible for producing variations in the outputs. However, compared with the traditional GAN, the conditional GAN takes external conditional contexts as additional inputs. The conditional contexts are highly structured and complex



*Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2411–2421
November 16 - 20, 2020. 

compared to the latent vector, making the latent code easily ignored and inoperative. Besides, the GAN-based methods usually fall into the mode collapse (Salimans et al., 2016) problem, that only a few modes in the latent space can work.

We address the above problems by encouraging the generator to be sensitive to latent codes and explore more modes in the latent space. For this purpose, we incorporate the conditional GAN with a simple yet efficient diversity loss term. During training, the diversity loss maximizes the ratio of the pairwise distance between the generated texts and their corresponding latent codes. As a result, the generator is forced to pay attention to latent codes and has the chance to generate different outputs.

We conduct experiments on Quora and MSCOCO datasets. Experimental results show that our proposed model can generate more diverse paraphrases compared with baselines while retaining the same semantics.

In summary, the primary contributions of this paper are as follows:

- We propose a conditional GAN-based framework to generate diverse paraphrases.
- To make the latent code valid and to alleviate the mode collapse problem, we propose a diversity loss term, which makes the generator sensitive to the change of latent codes.
- The experimental results show that our model can successfully generate more diverse paraphrases.

## 2 Related Work

### 2.1 Paraphrase Generation

Seq2seq-based methods have been widely used in the task of paraphrase generation (Prakash et al., 2016; Li et al., 2019; Kajiwara, 2019). Li et al. (2018) further adopt reinforcement learning with policy gradient technique to generate semantically consistent paraphrases. Gupta et al. (2018) propose a conditional VAE-based framework to generate paraphrases from the latent space. Shakeri and Sethy (2019) improve the VAE framework by conditioning the generator on a label which specifies whether the paraphrases are semantically consistent or not. Yang et al. (2019) further introduce the CVAE-GAN framework for paraphrase generation.

Some translation-based methods have also been proposed to generate paraphrases (Mallinson et al., 2017; Wieting et al., 2017; Guo et al., 2019). The main philosophy of these methods is to translate a text into another language (often referred to as "pivot language"), and translate it back to the original language. Then the original text and back-translated text are considered as a pair of paraphrases.

There are also some works trying to generate paraphrase in an unsupervised way. For example, Roy and Grangier (2019) adopt the vector-quantized VAE framework to discrete the latent space to generate paraphrases. Bao et al. (2019) decompose the latent space into syntactic and semantic space, and sample in the syntactic space while keeping semantics unchanged when generating paraphrases.

### 2.2 Generative Adversarial Nets

Generative Adversarial Nets was proposed by Goodfellow et al. (2014). The main idea of GAN is to train the generator and discriminator via minimax optimization, where the generator tries to generate realistic samples that match the real distribution, and the discriminator tries to distinguish between generated and real samples. GAN was first applied in the computer vision area. Some recent work have applied GAN-based framework in text generation (Yu et al., 2017; Kusner and Hernández-Lobato, 2016; Fedus et al., 2018; Guo et al., 2018; Wang and Wan, 2018). Applying GAN to text generation is nontrivial because generating discrete tokens is non-differentiable, making it difficult to optimize via back-propagation. The policy gradient technique (Sutton et al., 1999) is usually used to address this problem.

## 3 Methods

Given an input sentence $x = \{x_1, x_2, \cdots, x_n\}$, we seek to generate a set of $k$ paraphrase sentences $Y = \{y^{(1)}, y^{(2)}, \cdots, y^{(k)}\}$, that all $y \in Y$ have the same meaning with $x$, but are different in expression form.

### 3.1 Base Model

We build our model on top of the conditional GAN. The model consists of a generator $G$ and a discriminator $D$.

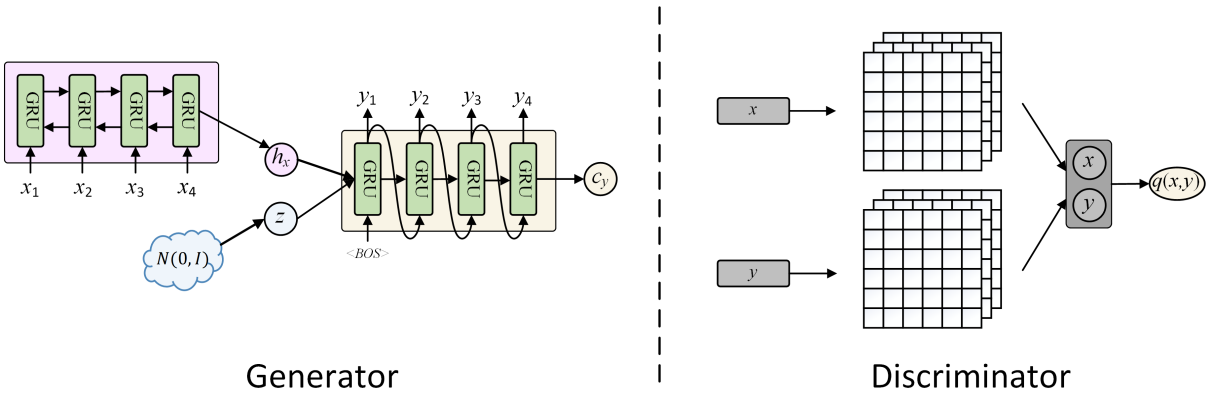


Figure 1: The overall framework of our proposed model. Our model consists of a generator and a discriminator. The generator is a GRU-based seq2seq network and the discriminator is a CNN network.

#### 3.1.1 Generator

The generator $G$ is a GRU-based seq2seq network which consists of a GRU encoder $G_{enc}$ and a GRU decoder $G_{dec}$. Given a text $x$, the encoder takes $x$ as input and encodes it into latent vector $h_x$. The decoder takes two inputs: the latent vector $h_x$ and a random vector $z$ sampled from the standard normal distribution, and generates the paraphrase $y$ corresponding to $x$. This process can be formalized as:

$$h_x = G_{enc}(x),\ y = G_{dec}(h_x, z) \tag{1}$$

and we abbreviate it as $y = G(x, z)$. It is worth noting that the generator is architecture-free and it can adopt many other seq2seq frameworks such as Transformer (Vaswani et al., 2017). Our work is orthogonal to those works that focus on designing sophisticated encoder and decoder architectures.

#### 3.1.2 Discriminator

The discriminator $D$ adopts a CNN network since CNN has recently been shown of great effectiveness in short text classification. Given a text $x$ and the paraphrase $y$, the CNN network encodes them into $C(x)$ and $C(y)$ of the same dimension respectively. Then the quality of the paraphrase is measured by a one-layer feed-forward network with sigmoid activation:

$$q(x, y) = \sigma(w[C(x); C(y)] + b) \tag{2}$$

where $w$ and $b$ are weight parameters, $\sigma$ refers to the sigmoid activation, and $q(x, y) \in [0, 1]$ is the quality of the paraphrase $y$ given the sentence $x$.

#### 3.1.3 Training Objective

Considering that a good paraphrase should not only be natural, but also have the same meaning with the input sentence. Similar to Reed et al. (2016), We extend the discriminator $D$ to identify three types of paraphrases for each input sentence $x$: (1) $S_x$: the set of paraphrases produced by human corresponding to $x$, (2) $S_G$ the set of paraphrases produced by the generator $G$ corresponding to $x$, and (3) $S_{\backslash x}$ the set of paraphrases produced by human, but are randomly sampled from all paraphrases which may be irrelevant to the given sentence $x$. Then the training objective is given below:

$$\begin{aligned}\mathcal{L}_{gan}(x, y) = &\ \mathbb{E}_{y \in \mathcal{S}_x} \log q(x, y) \\ &+ \alpha \cdot \mathbb{E}_{y \in \mathcal{S}_G} \log\left(1 - q(x, y)\right) \\ &+ \beta \cdot \mathbb{E}_{y \in \mathcal{S}_{\backslash x}} \log\left(1 - q(x, y)\right)\end{aligned} \tag{3}$$

Notice that the irrelevant sentences given to the discriminator is a common practice of training CGANs. Without this term, theoretically any topic sentences given to the discriminator will be considered correct.

The goal of the generator is to generate paraphrases that are semantically consistent and natural (i.e., indistinguishable for the discriminator). Therefore it should minimize Eq. 3. The goal of the discriminator is to distinguish artificial paraphrases (i.e., those generated from the generator), the golden paraphrases (i.e., those produced by humans corresponding to the input), and irrelevant paraphrases (i.e., those produced by humans but irrelevant to the input). Therefore it should maximize Eq. 3. This can be formalized as the following minimax problem:

$$\min_G \max_D\ \mathcal{L}_{gan}(x, y) \tag{4}$$

We adopt the adversarial training technique to optimize problem 4. To address the problem that

the gradient cannot pass back to the generator, we formalize the generation of discrete tokens as a sequential decision-making process and adopt the policy gradient and early feedback techniques described in Yu et al. (2017). We recommend readers refer to Yu et al. (2017) for more details.

### 3.2 Diversity Loss Term

#### 3.2.1 Motivation

We find in experiments that directly applying the conditional GAN model described above does not satisfactorily generate diverse paraphrases. Specifically, even if we sample multiple different $z$, the generated paraphrases are the same in many cases. This means that the latent code does not work or has minor impacts. We think this is because the conditional texts are highly structured and provide strong prior knowledge to guide the generation process, making the latent code negligible. Besides, from the perspective of optimization, this can be interpreted as the mode collapse problem (Salimans et al., 2016), where only a few modes get learned and the generator only generates samples from a few modes.

To solve this problem and produce diverse paraphrases, we propose to encourage the generator to explore more modes in the latent space and make the generator sensitive to latent codes. Inspired by Odena et al. (2018), we incorporate the conditional GAN with a **diversity loss term**.

#### 3.2.2 Formulation

Given an input sentence $x$, we sample a set of $k$ latent codes $\{z^{(i)}\}_{i=1}^{k}$ from the Gaussian distribution and generate corresponding paraphrases $\{y^{(i)}|y^{(i)} = G(x, z^{(i)})\}_{i=1}^{k}$. For the convenience of narration, we denote $\tilde{y}^{(i)}$ as the vector representation of $y^{(i)}$, where $\tilde{y}^{(i)}$ is obtained by taking the hidden state of the last time step of $y^{(i)}$. We use the L2 distance $\left\|\tilde{y}^{(i)} - \tilde{y}^{(j)}\right\|_2$ to measure the difference between $\tilde{y}^{(i)}$ and $\tilde{y}^{(j)}$, and use $\left\|z^{(i)} - z^{(j)}\right\|_2$ to measure the difference between $z^{(i)}$ and $z^{(j)}$, and denote $u^{(i,j)}$ as the ratio of $\left\|\tilde{y}^{(i)} - \tilde{y}^{(j)}\right\|_2$ and $\left\|z^{(i)} - z^{(j)}\right\|_2$:

$$u^{(i,j)} = \left\|\tilde{y}^{(i)} - \tilde{y}^{(j)}\right\|_2 / \left\|z^{(i)} - z^{(j)}\right\|_2 \quad (5)$$

Then diversity loss is calculated as:

$$\mathcal{L}_{div} = \frac{1}{k \cdot (k-1)} \sum_{i=1}^{k} \sum_{j \neq i}^{k} \max\left(\lambda - u^{(i,j)}, 0\right) \quad (6)$$

where $\lambda$ is a slack factor.

During training, the diversity loss $\mathcal{L}_{div}$ are appended to the original objective function:

$$L = L_{gan} + \gamma L_{div} \quad (7)$$

where $\gamma$ is the weight parameter. Combining the diversity loss term, the optimization problem becomes

$$\min_G \max_D \ \mathcal{L}(x, y) \quad (8)$$

We use the same techniques described in Section 3.1.3 to solve this problem.

#### 3.2.3 Why does it work

In Eq. 6, $\mathcal{L}_{div} > 0 \Leftrightarrow u^{(i,j)} < \lambda \Leftrightarrow \|\tilde{y}^{(i)} - \tilde{y}^{(j)}\|_2 < \lambda \cdot \|z^{(i)} - z^{(j)}\|_2$, this means that the generator will be punished if it does not produce different paraphrases given different latent codes. Therefore, the generator are forced to focus more on the latent codes and generate different paraphrases.

From the perspective of mode collapse, minimizing Eq. 6 can prevent the generator from producing samples only from a few modes, and enhance the chances of producing samples from some minor modes. Minimizing Eq. 6 can be seen as maximizing $\left\|\tilde{y}^{(i)} - \tilde{y}^{(j)}\right\|_2 / \left\|z^{(i)} - z^{(j)}\right\|_2$, where $\left\|\tilde{y}^{(i)} - \tilde{y}^{(j)}\right\|_2 / \left\|z^{(i)} - z^{(j)}\right\|_2$ corresponds to a lower-bound of the gradient of the generator:

$$\begin{aligned}
&\frac{\left\|\tilde{y}^{(i)} - \tilde{y}^{(j)}\right\|}{\left\|z^{(i)} - z^{(j)}\right\|} \\
&= \frac{\left\| \int_\Gamma \nabla_z G(x, z)\, \mathrm{d}z \right\|}{\left\|z^{(i)} - z^{(j)}\right\|} \\
&= \frac{\left\| \int_0^1 \nabla_z G(x, \Gamma(t)) \cdot (z^{(i)} - z^{(j)})\, \mathrm{d}t \right\|}{\left\|z^{(i)} - z^{(j)}\right\|} \\
&\leq \frac{\int_0^1 \left\|\nabla_z G(x, \Gamma(t))\right\| \left\|z^{(i)} - z^{(j)}\right\| \mathrm{d}t}{\left\|z^{(i)} - z^{(j)}\right\|} \\
&= \int_0^1 \left\|\nabla_z G(x, \Gamma(t))\right\| \mathrm{d}t
\end{aligned} \quad (9)$$

where $\Gamma(t) = tz^{(i)} + (1-t)z^{(j)}$ is a line segment with $z^{(i)}$ and $z^{(j)}$ as the end points.

Eq. 9 reveals that for any two modes $z^{(i)}$ and $z^{(j)}$, maximizing Eq. 5 will increase the gradient of the generator between $z^{(i)}$ and $z^{(j)}$. Therefore, by increasing the gradient of the generator, more modes can be learned, and thus the generator has the chance to generate samples from minor modes.

## 4 Experiments

### 4.1 Dataset

There are many datasets for paraphrase generation. We choose the two most widely used datasets, Quora[1] and MSCOCO (Lin et al., 2014) for experiments.

**Quora** Quora dataset consists of over 400K candidate question paraphrase pairs, and each pair has a manually annotated label. The two questions are paraphrasing each other only when the question pair is annotated as 1. This dataset contains 155K paraphrase question pairs in total.

**MSCOCO** MSCOCO is a benchmark for the task of image captioning. This dataset contains over 82K training and 42K validation images, and each image contains at most five human-labeled captions. Similar to previous work on paraphrase generation, we consider different captions of the same image as paraphrases. Following previous work, we reduce the sentences to the size of 15 words.

### 4.2 Evaluation Metrics

**BLEU4** : BLEU4 is the most widely used evaluation metric in paraphrase generation. We report the average BLEU4 score of the $k$ outputs. Notice that some works also calculate the **ROUGE** or **TER** scores, but we think the role of these two metrics overlaps with the BLEU metric, as they all calculate the degree of overlap between outputs and references. Therefore we only calculate the BLEU score to evaluate the closeness of outputs to the references.

**Self-BLEU** : To evaluate the degree to which the generated paraphrases are different from the original sentence, we propose to calculate the BLEU4 score between the generated paraphrases and input sentence. We name it "self-BLEU". The lower the self-bleu score, the more significant the change in the generated paraphrase. We report the average Self-BLEU score of the $k$ outputs.

**Pairwise-BLEU** : We propose to calculate the "pairwise-BLEU" score to evaluate the difference between the $k$ different paraphrases generated from the same given sentence. Concretely, for $k$ outputs $\{y^1, y^2, \cdots, y^k\}$, we compute the BLEU4 score between all $y^i$ and $y^j$ $(i \neq j)$, and average the $k(k-1)/2$ scores. A low pairwise-BLEU score means a high diversity between outputs, and vice versa. We abbreviate the Pairwise-BLEU as "P-BLEU".

**BERTScore** : To evaluate the semantic changes of the generated paraphrase compared with the input sentence, we calculate the BERTScore (Zhang et al., 2020) between the generated paraphrase and input sentence. We report the average BLEU4 score of the $k$ outputs.

**Human Evaluation** : In addition to the above automatic evaluation metrics, we also conduct human evaluation. We randomly sample 50 examples from the test set of Quora and MSCOCO datasets respectively. We ask five volunteers to evaluate the quality of the generated paraphrases from the following three aspects: (1) *Fidelity*: how semantically consistent are the generated paraphrases compared to the input sentence? (2) *Fluency*: how fluent are the generated paraphrases? (3) *Diversity*: how diverse are the generated paraphrases? (4) *Variability*: How much change do the generated paraphrases have in the form of expression compared with the input sentences? These scores are all between 1-5, with 5 being the best.

### 4.3 Competitive Models

We compare our model with the following baselines:

**LSTM** The stacked residual-LSTM proposed by Prakash et al. (2016). We reimplemented this baseline ourselves.

**Transformer** The standard Transformer model proposed by Vaswani et al. (2017). To improve the diversity of outputs, we test three variants: (1) **Transformer + beam**: using beam search to generate $k$ different outputs, (2) **Transformer + divbeam**: using the diverse beam search proposed by Vijayakumar et al. (2016) to generate $k$ different outputs, and (3) **Transformer + sampling**: using the sampling strategy to generate each token in the decoding stage.

**VAE-SVG** The variational auto-encoder model described in Gupta et al. (2018). We implement this model ourselves to participate in the experiments.

**D-PAGE** The **D**iverse **Pa**raphrase **Ge**neration model proposed by Xu et al. (2018b). They use a

[1] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

| Method | Quora | | | | MSCOCO | | | |
|---|---|---|---|---|---|---|---|---|
| | BLEU ↑ | Self-BLEU ↓ | P-BLEU ↓ | BERTScore ↑ | BLEU ↑ | Self-BLEU ↓ | P-BLEU ↓ | BERTScore ↑ |
| Source | 32.05 | 100.00 | - | 100 | 10.60 | 100.00 | - | 100 |
| Reference | 100.00 | 32.05 | - | 79.24 | 100.00 | 10.60 | - | 63.74 |
| Residual-LSTM | 24.57* | 42.79* | 46.01* | 77.88* | 18.36* | 10.52* | 47.90* | 62.22* |
| Transformer + beam | 30.59 | 42.30* | 49.69* | 80.69 | 22.06 | 9.44* | 49.26* | 66.86 |
| Transformer + divbeam | 30.07 | 36.48* | 35.73* | 81.02 | 20.28 | 11.11* | 38.79* | 63.06* |
| Transformer+sampling | 21.69* | **20.31** | **26.07** | 63.40* | 7.49* | **3.04** | 18.83* | 50.73* |
| VAE-SVG | 32.00 | 37.53* | 44.42* | 79.44 | 23.90 | 9.28 | 35.10* | 61.74* |
| D-PAGE | 29.29 | 36.68* | 40.10* | 80.65 | 22.00 | 9.13 | 39.49* | 66.13 |
| DPGAN | 23.78* | 23.64 | 34.94* | 76.19* | 12.54* | 6.99 | 19.49* | 57.13* |
| CGAN | 29.83 | 38.73* | 53.06* | 80.19 | 22.03 | 11.26* | 44.55* | **66.97** |
| DivGAN (average) | 28.49 | 33.90 | 32.64 | 80.31 | 20.63 | 8.51 | **15.45** | 66.31 |
| DivGAN (best) | 31.56 | 34.31 | - | **81.08** | 24.06 | 10.51 | - | 66.70 |

Table 1: Experimental results of paraphrase generation on Quora and MSCOCO datasets. Statistically significant improvements ($p < 0.01$) over DivGAN (average) are marked with *.

| Method | Quora | MSCOCO |
|---|---|---|
| Source | 32.05 | 10.60 |
| DNPG | 25.03 | **29.16** |
| RbM-SL | **35.81** | - |
| MC-WGAN | 27.54 | 22.22 |
| MC-WGAN (best) | 32.33 | 27.83 |
| UPSA | 18.18 | 14.16 |
| DivGAN (average) | 28.49 | 20.63 |
| DivGAN (best) | 31.56 | 24.06 |

Table 2: BLEU4 score results on Quora and MSCOCO dataset.

shared decoder with different decoder embeddings to generate different outputs.

**DPGAN** The **D**iversity-**P**romoting **GAN** proposed by Xu et al. (2018a). They assign low reward for repeated text and high reward for novel text to prompt diverse outputs.[2]

**CGAN** The conditional GAN with the same architecture as our model, but without the diversity loss term.

**Other baselines** We also report the results of DNPG (Li et al., 2019), RbM-SL (Li et al., 2018), MC-WGAN (An and Liu, 2019), and UPSA (Liu et al., 2019). Notice that they focus on generating high-quality single paraphrase, and do not test to generate multiple paraphrases in their experiments. Thus we can only list their BLEU4 scores for reference.[3]

### 4.4 Implementation Details

For the generator, the encoder is set as a one-layer bidirectional GRU network with inner self-attention, and the decoder is set as a two-layer unidirectional GRU network. The dimension of the input and hidden size is set to 512. The latent code dimension is set to 512, and the latent code is concatenated to each input token. For the discriminator, the CNN network is the same as Kim (2014), where the size of filter windows are set as 3, 4, 5 with 100 feature maps each.

Following previous work on GAN-based text generation, we pre-train the generator using standard MLE loss for 25 epochs, and pre-train the discriminator using the objective in Eq. 3 for 5 epochs. After pre-training, the generator and discriminator are trained alternatively, where each iteration consists of a G-step followed by a D-step.

We use the NLTK[4] tool to process the English texts. The vocabulary sizes are set as 50,000 and 80,000 for Quora and MSCOCO datasets, respectively. We set $\alpha = 0.8$ and $\beta = 0.8$ in Eq. 3, and $\gamma = 10$ in Eq. 7 according to the performance on the validation set.

### 4.5 Experiments Setup

For generative models, we sample $z^{(1)}, z^{(2)}, \cdots, z^{(k)}$ from the Gaussian distribution to generate $k$ outputs. For Transformer models, we use the beam search to generate $k$ outputs. We set $k = 3$ for all models in experiments.

## 5 Results and Analysis

### 5.1 Results of Automatical Evaluation Metrics

The comparison results of our model and main baseline models on Quora and MSCOCO datasets are

[2] https://github.com/lancopku/DPGAN

[3] They do not release their codes, so we cannot get their results of generating multiple paraphrases.

[4] https://github.com/nltk/nltk

| Method | Fid. ↑ | Flu. ↑ | Div. ↑ | Var. ↑ |
|---|---|---|---|---|
| Reference | 3.66 | 4.04 | - | 3.20 |
| Transformer+beam | **3.67** | 3.92 | 3.47 | 3.20 |
| Transformer+divbeam | 3.61 | 3.84 | 3.70 | 3.37 |
| Transformer+sampling | 2.42 | 2.69 | 3.86 | 3.74 |
| VAE-SVG | 3.36 | 3.92 | 3.62 | 3.23 |
| D-PAGE | 3.56 | 3.81 | 3.43 | 3.31 |
| DPGAN | 3.28 | 3.93 | 3.71 | **3.75** |
| CGAN | 3.47 | **4.10** | 3.50 | 3.36 |
| DivGAN | 3.58 | 4.03 | **3.87** | 3.34 |

Table 3: Results of the human evaluation on the Quora dataset.

| Method | Fid. ↑ | Flu. ↑ | Div. ↑ | Var. ↑ |
|---|---|---|---|---|
| Reference | 3.07 | **3.90** | - | 3.51 |
| Transformer+beam | **3.15** | 3.77 | 3.34 | 3.46 |
| Transformer+divbeam | 3.00 | 3.75 | 3.73 | 3.52 |
| Transformer+sampling | 2.18 | 2.07 | 3.57 | **3.75** |
| VAE-SVG | 2.95 | 3.80 | 3.82 | 3.47 |
| D-PAGE | 3.02 | 3.71 | 3.48 | 3.37 |
| DPGAN | 2.95 | 3.66 | 3.76 | 3.71 |
| CGAN | 3.11 | 3.83 | 3.79 | 3.43 |
| DivGAN | 3.12 | 3.80 | **3.95** | 3.52 |

Table 4: Results of the human evaluation on the MSCOCO dataset.

shown in Table 1. For the other baselines, we also show the BLEU4 scores in Table 2 for reference.

In terms of BLEU4 score, our **DivGAN (average)** performs worse than **RbM-SL**, **MC-WGAN**, **VAE-SVG**, **D-PAGE** and those transformer-based methods. However, we strongly argue that **this does not mean that the quality of our generated paraphrases is worse than those generated by these models**. Previous works have shown that BLEU is not a good measure for evaluating several text generation tasks, including dialogue generation (Liu et al., 2016), sentence simplification (Sulem et al., 2018) and paraphrase generation (Liu et al., 2010; An and Liu, 2019). First, we also think that the BLEU itself is not is a perfectly reasonable metric for the paraphrase generation task. The paraphrases are highly diversified in nature, but there is only one reference in these paraphrase datasets. Taking the sentences *"what can i do to overcome anxiety"* with the human reference *"what do i do to reduce my anxiety"* for example, our model generates sentences like *"how do i overcome anxiety"* or *"what's the best way to overcome anxiety"* which are low in BLEU score, but are good paraphrases from human's point of view. Therefore, we think that a high BLEU score only indicates a high degree of overlap between the generated paraphrase and reference, but does not indicate high quality. Second, the BERTScore and the human evaluation results show that the paraphrases we generate are no worse than these models in terms of relevance and fluency, and even better than these models. It is worth mentioning that in terms of BERTScore and human evaluation, the **DivGAN** model even outperforms the human reference. Third, we also find that the more diverse the paraphrases generated, the lower the average BLEU score is. This is because once we generate a paraphrase which is very similar to the reference, the diverse loss will encourage the rest paraphrases to be different from this paraphrase, which causes the BLEU score of the rest $k-1$ paraphrases to be lower, thereby lowering the average BLEU score. We calculate the highest BLEU score among the $k$ results, and find that it is $3 \sim 4$ points higher than the average score (see **DivGAN (best)**).

In terms of the Pairwise-BLEU score, the **DivGAN** model significantly outperforms all baselines (except the **Transformer + sampling** model on Quora dataset), indicating that the proposed model can generate diverse sentences effectively. We notice that just by removing the diverse loss term from **DivGAN**, the Pairwise-BLEU of **CGAN** is greatly increased (from 32.64 to 53.06 on Quora, and from 15.45 to 44.55 on MSCOCO). By checking the outputs, we find that **CGAN** generates a lot of repeated sentences, thereby boosting the Pairwise-BLEU score. We find that our **DivGAN** occasionally produces repeated sentences either, but the number of repeated sentences generated by **DivGAN** is far less than that of **C-GAN**, **D-PAGE** and **VAE-SVG**. These results demonstrate the effectiveness of our proposed diverse loss.

The **Transformer + sampling** model seems to be able to generate diverse outputs according to the low scores of Self-BLEU and Pairwise-BLEU. However, by checking the outputs, we find that **Transformer + sampling** model produces large amounts of meaningless text, such as sentences in Table 5. These near-randomly generated tokens make **Transformer + sampling**'s Self-BLEU and Pairwise-BLEU scores lower, making the BLEU and BERTScore scores lower, either.

Although the **D-PAGE** tries to obtain different outputs from using different decoder embeddings, we find that the sentences generated by different decoders are the same, or of little changes in many cases. This is because the decoders are not explicitly encouraged to produce different results.

The **DPGAN** model can achieve a low pairwise-bleu score, but its BLEU4 and BERTScore are also low. By checking the outputs, we find that this is because **DPGAN** tends to produce long sentences. To generate "novel" sentences, **DPGAN** uses the cross-entropy loss as the reward and long sentences can have a high reward. Therefore, **DPGAN** achieves low BLEU4 score as the references are relatively short, and achieves low BERTScore as the long text will change the semantics to some extent.

In terms of BERTScore, it can be seen that although our model achieves lower BLEU scores in some cases, it can achieve similar or even higher BERTScore. To some extent, this shows that although the paraphrases generated by our model are more different from human references, the quality of these paraphrases is still good.

### 5.2 Results of Human Evaluations

Table 3 and Table 4 show the results of the human evaluation on the Quora and MSCOCO datasets, respectively.

It can be seen that in terms of the quality (fidelity, fluency, and variability), all models' scores are close to the human reference, except the **Transformer + sampling**. This shows that all models can generate human-like paraphrases. But in terms of the diversity score, our proposed model surpasses other competitive models, indicating that our model can generate more diverse paraphrases.

## 6 Case Study

Table 5 shows outputs of different models for an input sentence from the Quora dataset. We have the following observations.

First, using traditional beam search can produce different outputs, but the generated texts are of high similarity with minor modification (for example, replacing "can you" with "do you", or replacing "while awake" with "while you are awake"). Secondly, using the sampling strategy during decoding sometimes produces unnatural output, especially at the beginning or end of the sentence (see the second and third sentences in Table 5). Thirdly, **VAE-SVG** and **C-GAN** sometimes produce the same outputs (see the first and third sentences in **C-GAN** in Table 5), indicating that the latent codes sometimes do not work well. **Transformer + divbeam**, **DPGAN**, and our **DivGAN** model can produce high-quality and diverse outputs. By comparing more generated samples from the test set, we find that our **DivGAN** model can generate more diverse samples than the other two models.

| |
|---|
| **Source Text:** can you dream while awake ? |
| **Reference:** can people dream while they are awake ? |
| **Transformer + beam:**<br>1: can you dream while awake ?<br>2: can you dream while you are awake ?<br>3: do you dream while awake ? |
| **Transformer + divbeam:**<br>1: can you dream while you are awake ?<br>2: how can i dream while awake ?<br>3: what are some ways to dream while awake ? |
| **Transformer + sampling:**<br>1: can you dream while you're awake ?<br>2: importantly .5 . can you dream while you have awake ?<br>3: can you dream while you're awake ? fiance , so , i / anything . |
| **VAE-SVG:**<br>1: can you dream when you are awake ?<br>2: do you dream while awake ?<br>3: can you dream when you wake up ? |
| **D-PAGE:**<br>1: can you dream while awake ?<br>2: can you dream while you are awake ?<br>3: do you dream while awake ? |
| **DPGAN:**<br>1: how can you dream while you are wake up ?<br>2: are there some ways for you to dream awake ?<br>3: how do you dream while you had awake? |
| **C-GAN:**<br>1: can you dream while you are awake ?<br>2: how can i dream while awake ?<br>3: can you dream while you are awake ? |
| **DivGAN:**<br>1: how do you dream while you are awake ?<br>2: is it possible to dream while you have awake ?<br>3: do you dream while awake ? |

Table 5: An example of the case study from the Quora dataset.

## 7 Conclusions

In this paper, we propose a conditional generative adversarial network based model to tackle the task of diverse paraphrase generation. To solve the problem of the minor impacts of the latent codes and the mode collapse in the conditional GAN, we propose to add a diversity loss term to the objective. The diversity loss term encourages the generator to explore more in the latent space and generate samples from some minor modes. Experimental results demonstrate the effectiveness of the proposed diversity loss term. In the future, we will apply the diversity loss to more tasks and models.

## Acknowledgments

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

## References

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

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 *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.

Igor A. Bolshakov and Alexander F. Gelbukh. 2004. Synonymous paraphrasing using wordnet and internet. In *Natural Language Processing and Information Systems, 9th International Conference on Applications of Natural Languages to Information Systems, NLDB 2004, Salford, UK, June 23-25, 2004, Proceedings*, volume 3136 of *Lecture Notes in Computer Science*, pages 312–323. Springer.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *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 5972–5984. Association for Computational Linguistics.

William Fedus, Ian J. Goodfellow, and Andrew M. Dai. 2018. Maskgan: Better text generation via filling in the _____. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *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.

Yinpeng Guo, Yi Liao, Xin Jiang, Qing Zhang, Yibo Zhang, and Qun Liu. 2019. Zero-shot paraphrase generation with multilingual language models. *CoRR*, abs/1911.03597.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *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.

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

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *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 6047–6052. Association for Computational Linguistics.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA*. The Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *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 1746–1751. ACL.

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

Matt J. Kusner and José Miguel Hernández-Lobato. 2016. GANS for sequences of discrete elements with the gumbel-softmax distribution. *CoRR*, abs/1611.04051.

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

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2010. PEM: A paraphrase evaluation metric exploiting parallel texts. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 923–932. ACL.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2122–2132. The Association for Computational Linguistics.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2019. Unsupervised paraphrasing by simulated annealing. *CoRR*, abs/1909.03588.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 881–893. Association for Computational Linguistics.

Kathleen R. McKeown. 1983. Paraphrasing questions using given and new information. *Am. J. Comput. Linguistics*, 9(1):1–10.

Marie Meteer and Varda Shaked. 1988. Strategies for effective paraphrasing. In *Proceedings of the 12th International Conference on Computational Linguistics, COLING '88, Budapest, Hungary, August 22-27, 1988*, pages 431–436. John von Neumann Society for Computing Sciences, Budapest.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *CoRR*, abs/1411.1784.

Shashi Narayan, Siva Reddy, and Shay B. Cohen. 2016. Paraphrase generation from latent-variable pcfgs for semantic parsing. In *INLG 2016 - Proceedings of the Ninth International Natural Language Generation Conference, September 5-8, 2016, Edinburgh, UK*, pages 153–162. The Association for Computer Linguistics.

Augustus Odena, Jacob Buckman, Catherine Olsson, Tom B. Brown, Christopher Olah, Colin Raffel, and Ian J. Goodfellow. 2018. Is generator conditioning causally related to GAN performance? In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3846–3855. PMLR.

Sunghyun Park, Seung-won Hwang, Fuxiang Chen, Jaegul Choo, Jung-Woo Ha, Sunghun Kim, and Jinyeong Yim. 2019. Paraphrase diversification using counterfactual debiasing. 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 6883–6891. AAAI Press.

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 *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2923–2934. ACL.

Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1060–1069. JMLR.org.

Aurko Roy and David Grangier. 2019. Unsupervised paraphrasing without translation. In *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 6033–6039. Association for Computational Linguistics.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2226–2234.

Siamak Shakeri and Abhinav Sethy. 2019. Label dependent deep variational paraphrase generation. *CoRR*, abs/1911.11952.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 738–744. Association for Computational Linguistics.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1057–1063. The MIT Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424.

Ke Wang and Xiaojun Wan. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4446–4452. ijcai.org.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 274–285. Association for Computational Linguistics.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018a. Diversity-promoting GAN: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3940–3949. Association for Computational Linguistics.

Qiongkai Xu, Juyan Zhang, Lizhen Qu, Lexing Xie, and Richard Nock. 2018b. D-PAGE: diverse paraphrase generation. *CoRR*, abs/1808.04364.

Qian Yang, Zhouyuan Huo, Dinghan Shen, Yong Cheng, Wenlin Wang, Guoyin Wang, and Lawrence Carin. 2019. An end-to-end generative architecture for paraphrase generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3130–3140. Association for Computational Linguistics.

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

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

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 834–842. The Association for Computer Linguistics.