Preference-Controlled Multi-Objective Reinforcement Learning for Conditional Text Generation

Wenqing Chen[1*], **Jidong Tian**[2, 3*], **Caoyun Fan**[2, 3], **Yitian Li**[2, 3], **Hao He**[2, 3†], **Yaohui Jin**[2, 3†]

[1] School of Software Engineering, Sun Yat-sen University
[2] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[3] State Key Lab of Advanced Optical Communication System and Network, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University
chenwq95@mail.sysu.edu.cn, {frank92, fcy3649, yitian_li, hehao, jinyh}@sjtu.edu.cn

Abstract

Conditional text generation is to generate text sequences conditioning on linguistic or non-linguistic data. The main line of existing work proposed deterministic models to improve the fidelity of the generated text but often ignored the diversity. Another line relied on conditional variational auto-encoders (CVAEs), which increased the diversity over their deterministic backbones. However, CVAEs regard diversity as an implicit objective and may not be optimal. In this paper, we raise two questions: i) Can diversity be further improved with an explicit objective? ii) Since fidelity and diversity are two conflicting objectives, how can we obtain different multi-objective optimal solutions according to user preferences? To answer question i), we propose a multi-objective reinforcement learning (MORL) method which explicitly takes CIDEr and Self-CIDEr scores as the fidelity-oriented and diversity-oriented rewards respectively. To answer question ii), we propose a preference-controlled MORL method, which can obtain infinite multi-objective optimal solutions by tuning the preference variable. We conduct extensive experiments on paraphrasing and image captioning tasks, which show that in the fidelity-diversity trade-off space, our model outperforms both deterministic and CVAE-based baselines.

Introduction

Conditional text generation covers a wide range of tasks such as machine translation (Bahdanau, Cho, and Bengio 2015), abstractive summarization (Nallapati et al. 2016), conversation (Vinyals and Le 2015), and image captioning (Xu et al. 2015). In practice, most of these tasks can be formulated as sequence-to-sequence (seq2seq) generation problems where models are required to generate proper text conditioning on linguistic or non-linguistic input data.

While recent work mainly focused on improving the semantic fidelity of generated text with respect to the input, diversity was often ignored. Given the input $\boldsymbol{x}$, models that capture the diverse nature of human language are expected to generate multiple faithful and diverse sentences $(\widetilde{\boldsymbol{y}}^1, \widetilde{\boldsymbol{y}}^2, ..., \widetilde{\boldsymbol{y}}^k)$. However, most of the existing models are relatively deterministic partially due to the greedy decoding or the beam-search decoding process. Specifically, greedy decoding only yields one text sequence. Beam search can yield different candidate sentences, but the k-th ($k > 1$) candidate would have lower quality than the top-1 (Gupta et al. 2018). It means that beam search does not produce multiple high-quality sentences in a principled way (Gupta et al. 2018; Chen et al. 2020).

Another line of work improves the diversity based on the conditional variational auto-encoder (CVAE), that has been applied on some tasks such as dialog response generation (Du et al. 2018), paraphrasing (Gupta et al. 2018; Chen et al. 2020), and image captioning (Wang, Schwing, and Lazebnik 2017). CVAE can be denoted by a model $p_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})$ where a latent variable $\boldsymbol{z}$ is sampled from a prior distribution $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ during testing (Sohn, Lee, and Yan 2015). Although the greedy decoding or the beam search is relatively deterministic, the sampling process $\boldsymbol{z} \sim p_\theta(\boldsymbol{z}|\boldsymbol{x})$ contributes to the diversity.

The remaining problem is that CVAE regards diversity as an implicit objective and may not be optimal. Specifically, CVAE assumes the sampling process $\boldsymbol{z} \sim p_\theta(\boldsymbol{z}|\boldsymbol{x})$ does not significantly hurt the quality of the generated text, and the diversity highly depends on the characteristics of $p_\theta(\boldsymbol{z}|\boldsymbol{x})$. We raise two questions for CVAE: i) **Can the diversity be further improved with an explicit objective?** ii) Since fidelity and diversity are two conflicting objectives, **how can we obtain different multi-objective optimal solutions according to user preferences?** To respond to the two questions, we propose a preference-controlled multi-objective reinforcement learning (PCMORL) method.

Firstly, for question i), we introduce a multi-objective reinforcement learning (MORL) strategy to improve both fidelity and diversity. Recent RL-based work on text generation mainly considered the fidelity-oriented rewards such as BLEU (Papineni et al. 2002) score for machine translation (Wu et al. 2018), ROUGE (Lin 2004) score for text summarization (Buciumas 2019), and CIDEr (Vedantam, Zitnick, and Parikh 2015) score for image captioning (Rennie et al. 2017). We choose the CIDEr score as the fidelity-oriented reward because it not only encourages the n-gram consistency but also penalizes less informative n-grams. Then we choose the Self-CIDEr (Wang and Chan 2019) score as the diversity-oriented reward, which is an explicit

*These authors contributed equally.
†Corresponding authors.

objective for diversity. Notably, though our method is based on RL, pre-training with maximum likelihood estimation (MLE) is also required in the field of text generation (Yu et al. 2017).

Secondly, for question ii), we propose a more challenging setting, that is, learning an efficient model which can switch on the multi-objective optimal curve in the inference stage by training once. To achieve this goal, we introduce an additional variable r representing the user preference of fidelity or diversity, which is sampled from a uniform distribution. Our model prefers fidelity when $r \to 0$ and prefers diversity when $r \to 1$. The controllability is achieved by adjusting the variance of the Gaussian distributions in PCMORL with a monotonic neural network (Wehenkel and Louppe 2019). Besides making the model preference-aware, we also use r to balance the CIDEr and the Self-CIDEr rewards during the MORL optimization period. In the inference stage, we can tune r to obtain different multi-objective optimal solutions.

We conduct experiments on two tasks including text-to-text and image-to-text generation. For the text-to-text scenario, we choose the paraphrasing task because paraphrasing is an important building block to improve the diversity of many text generation systems such as conversation (Vinyals and Le 2015), machine translation (Cho et al. 2014), and abstractive summarization (Nallapati et al. 2016). For the image-to-text scenario, the task image captioning is a natural choice (Anderson et al. 2018; Shi et al. 2018a). Extensive experiments show that our model consistently outperforms other baseline models in the fidelity-diversity space, and can obtain infinite multi-objective solutions by tuning the preference r.

The main contributions of this work can be summarized as follows:

- We consider diversity as an explicit optimization objective where the Self-CIDEr score is taken as the diversity-oriented reward in the MORL framework for conditional text classification.
- We propose a preference-controlled MORL strategy that allows our model to obtain infinite multi-objective solutions to balance fidelity and diversity.
- The experiments demonstrate that our model outperforms deterministic and CVAE-based baselines when jointly considering fidelity and diversity, and is more flexible in the inference stage.

Related Work

In this section, we briefly review recent progress in conditional text generation tasks.

Deterministic Text Generation The majority of recent work focused on improving fidelity by developing deterministic models. Generally, a basic neural network for conditional text generation relies on the basic encoder-decoder architecture (Sutskever, Vinyals, and Le 2014; Cho et al. 2014). The following work proposed the attention mechanism for the encoder-decoder architecture which obtained dynamic representations of the input at different decoding steps (Bahdanau, Cho, and Bengio 2015; Xu et al. 2015). Self-attention, sometimes called intra-attention, is a special attention mechanism that is applied to a single sequence and can learn a strong representation of text (Parikh et al. 2016; Lin et al. 2017). Based on self-attention, Transformer was proposed and had shown its superiority on the seq2seq tasks with higher quality and efficiency (Vaswani et al. 2017). Recent studies proposed Transformer variants and pre-trained them on the large-scale corpus, yielding powerful pre-trained models like BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019), Transformer-XL (Dai et al. 2019), XLNet (Yang et al. 2019), BART (Lewis et al. 2020), and T5 (Raffel et al. 2020).

Reinforcement Learning The traditional optimization objective of the seq2seq generation is MLE, which suffers from the exposure bias problem (Ranzato et al. 2016), a type of train-test discrepancy. During training, the model generates each token conditioning on its preceding ground-truth tokens while conditioning on previously generated ones during testing. To deal with the train-test discrepancy, recent work proposed to use RL for text generation, which let a model sample each token conditioning on previously sampled tokens instead of ground-truths ones, and then provided token-level or sentence-level rewards (Rennie et al. 2017; Keneshloo et al. 2020). In the field of text generation, commonly used sentence-level rewards can be n-gram consistency scores, such as BLEU (Papineni et al. 2002) for machine translation (Wu et al. 2018), ROUGE (Lin 2004) for text summarization (Buciumas 2019), and CIDEr (Vedantam, Zitnick, and Parikh 2015) for image captioning (Rennie et al. 2017).

Diverse Text Generation Due to the diversity of natural language, a text generation system that can produce multiple diverse sentences is more human-like and less boring to users. Models for diverse text generation were typically built on CVAEs (Hu et al. 2017; Wang, Schwing, and Lazebnik 2017; Ruan, Ling, and Zhu 2020; Shao et al. 2021) or CGANs (Dai et al. 2017), either of which improved the diversity by the sampling of a latent variable $\boldsymbol{z}$ with the conditional generation process $p_\theta(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})$. Compared with CGANs, CVAEs seem to be more preferable for text generation due to two challenges of GAN: reward sparsity and mode collapse (Shi et al. 2018b). Reward sparsity is because the discriminator of GAN provides rewards at the end of the text sequence rather than stepwise rewards. Mode collapse means that GANs tend to learn limited patterns. Recent theoretical work indicated that VAE tends to cover all modes of data distribution while GAN prefers high-density regions but covers limited modes of data distribution (Hu et al. 2018). The empirical study showed that GANs did not outperform MLE-based models in the quality-diversity space (Caccia et al. 2020).

Our approach for diverse text generation is a two-stage optimization method. The first stage is based on a CVAE variant while the second stage is formulated as a MORL problem that involves both fidelity-oriented and diversity-oriented rewards.

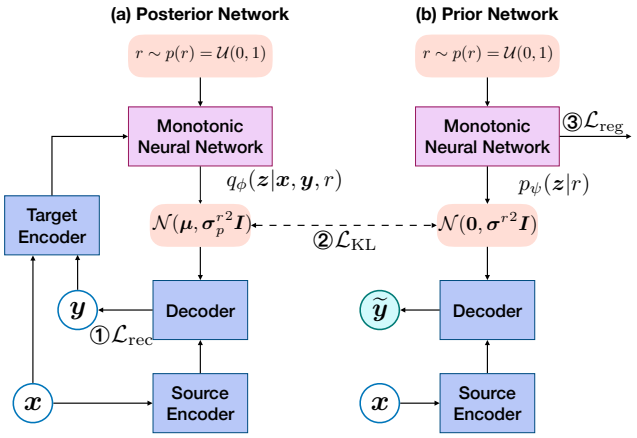


Figure 1: The architecture of PACVAE which uses a monotonic neural network to process the preference variable r and output the variance of the Gaussian distributions.

Preliminary

We briefly review the framework of CVAE (Sohn, Lee, and Yan 2015), which is an important baseline for diverse text generation. Assume that the input data and the output sequences are denoted by $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively, vanilla CVAE is to maximize the evidence lower bound (ELBO):

$$\begin{aligned}\log p(\boldsymbol{y}|\boldsymbol{x}) &= \log \int_{\boldsymbol{z}} p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})p(\boldsymbol{z}|\boldsymbol{x})\mathrm{d}\boldsymbol{z} \geq \text{ELBO} \\ &\geq \mathop{\mathbb{E}}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\log p_\theta(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})] \\ &- \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})\|p_\theta(\boldsymbol{z}|\boldsymbol{x}))\end{aligned} \tag{1}$$

where θ and ϕ denote the parameters of the prior and the posterior networks, respectively.

In some CVAE formulations, such as the one adopted in a previous work (Jain, Zhang, and Schwing 2017), the prior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ is relaxed to $p(\boldsymbol{z})$, which is a zero-mean unit-variance Gaussian $\mathcal{N}(0,\mathrm{I})$. We denote this type of CVAE as simplified CVAE (SCVAE).

Methodology

Task Definition

Most text generation tasks can be formulated as the sequence-to-sequence generation problem, and we investigate the **text-to-text** and **image-to-text** generation tasks in this paper. For the text-to-text task, the input is a text sequence represented by $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$ where the i-th element x_i represents a token. A word embedding layer will map $\boldsymbol{x}$ into a sequence of word embedding representations $\boldsymbol{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n\}$. For the image-to-text task, $\boldsymbol{x}$ is an image and we use a pre-trained object detection network (Anderson et al. 2018) to convert $\boldsymbol{x}$ into a sequence of object-based features $\boldsymbol{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n\}$, which is the same with previous work (Anderson et al. 2018; Wang et al. 2020; Pan et al. 2020). The element $\boldsymbol{e}_i$ represents the feature vector of the i-th detected object.

In this way, the text-to-text and image-to-text tasks can be unified with a general seq2seq model, the aim of which is to generate a group of text sequences $\widetilde{G} = (\widetilde{\boldsymbol{y}}^1, \widetilde{\boldsymbol{y}}^2, ..., \widetilde{\boldsymbol{y}}^k)$, and each sentence should be faithful to the input $\boldsymbol{x}$ and the group $\widetilde{G}$ should be diverse enough.

Since fidelity and diversity are two conflicting objectives, we further propose a challenging setting that introduces a preference variable $r \in [0,1]$ to switch on the fidelity-diversity optimal curve by training only once. In the inference stage, we can obtain infinite multi-objective optimal solutions by tuning r.

Pre-training with MLE

Although our final model is based on RL, we need to pre-train the model with MLE to stabilize the RL optimization process (Yu et al. 2017; Guo et al. 2018; Paulus, Xiong, and Socher 2018). We propose a CVAE variant as our base model, preference-aware CVAE (PACVAE). As shown in Figure 1, PACVAE consists of a source encoder $SEnc$, a target encoder $TEnc$, a decoder Dec, and a monotonic neural network MNN.

If excludes MNN, PACVAE degrades to SCVAE which has the prior distribution of $p(\boldsymbol{z})$. SCVAE is chosen as the base model of PACVAE because we empirically found that when fine-tuned with RL, vanilla CVAE performs close to a deterministic model while SCVAE can maintain diversity (shown in Section Experiments).

Based on SCVAE, PACVAE takes the user preference $r \in [0,1]$ into account. When $r \to 0$, PACVAE prefers fidelity and vice versa. Note that we could denote the user preference by a vector if more than two objectives existed, but in this paper, the scalar can cover our requirements. During the MLE period, the user preference r is sampled from a uniform distribution $p(r) = \mathcal{U}(0,1)$, and the log-likelihood becomes:

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = \mathop{\mathbb{E}}_{r\sim p(r)}[\log p(\boldsymbol{y}|\boldsymbol{x},r)] \tag{2}$$

which is approximated to the ELBO:

$$\begin{aligned}\log p(\boldsymbol{y}|\boldsymbol{x}) &\geq \mathop{\mathbb{E}}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y},r)}[\log p_\theta(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z},r)] \\ &- \mathop{\mathbb{E}}_{r\sim p(r)}[\mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y},r)\|p_\psi(\boldsymbol{z}|r))] \\ &= -\mathcal{L}_{\text{rec}} - \mathcal{L}_{\text{KL}}\end{aligned} \tag{3}$$

where $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{KL}}$ represent the reconstruction loss and KL loss respectively. The implementations of the encoders and the decoder are relatively common, and hence we introduce them in the appendix.

Then we mainly introduce the module MNN. We let $p_\psi(\boldsymbol{z}|r)$ and $q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y},r)$ be Gaussian distributions, of which the variance is controlled by r as shown in Figure 2. Intuitively, a higher-variance Gaussian distribution leads to higher diversity because of the less deterministic sampling process. For any two Gaussian distributions with the standard variance $\boldsymbol{\sigma}^{r_1}, \boldsymbol{\sigma}^{r_2} \in \mathbb{R}_+^{d\times d}$, we let:

$$\boldsymbol{\sigma}^{r_1}\mathrm{I} \succeq \boldsymbol{\sigma}^{r_2}\mathrm{I}, \forall r_1 \geq r_2 \tag{4}$$

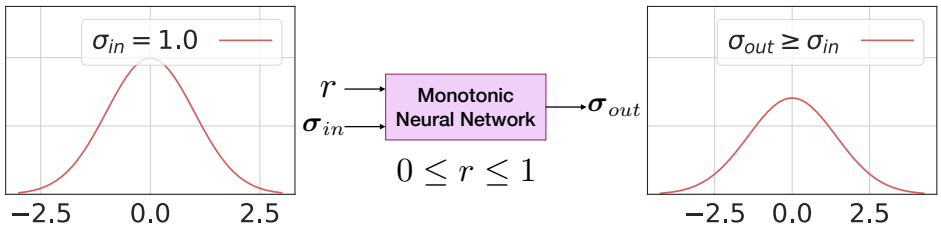


Figure 2: The key idea of the preference-aware monotonic neural network. As illustrated, the variance of the prior Gaussian distribution is controlled by the preference r. When $r \to 1$, the Gaussian becomes flatter and wider.

where $\succeq$ denotes the element-wise no-less-than. It suggests the monotonic property of MNN, and we follow the idea of making the derivative of a neural network strictly positive (Wehenkel and Louppe 2019). Specifically, we represent a diagonal element in $\boldsymbol{\sigma}^r$ by $F_\psi(r)$ and its derivative function by $f_\psi(r)$, then we have:

$$F_\psi(r) = \int_0^r f_\psi(t)\mathrm{d}t + F_\psi(0) \tag{5}$$

where $f_\psi(\cdot)$ consists of a multi-layer perception and an ELU activation unit to ensure $f_\psi(\cdot)$ be positive. $F_\psi(0)$ is an element in $\boldsymbol{\sigma}^0$ with the preference variable $r = 0$, and we let $\boldsymbol{\sigma}^0 = \mathrm{I}$.

The next thing is to optimize PACVAE. When considering the MLE objective with Equation 3, we found that the model tends to minimize the loss by regarding r as noise which makes $F_\psi(r) \approx F_\psi(0), \forall r$. To deal with this problem, we add a regularization loss term:

$$\mathcal{L}_{\text{reg}} = \frac{1}{d}\sum_{i=1}^{d}\left(\max\left(1 + \lambda_\sigma r - \frac{\sigma_i^r}{\sigma_i^0}\right), 0\right) \tag{6}$$

where d is the dimension of $\boldsymbol{\sigma}^r$, and σ_i^r is the i-th diagonal element in $\boldsymbol{\sigma}^r$. And $\lambda_\sigma \in \mathbb{R}^+$ denotes a hyperparameter with a positive value. Then the loss of PACVAE becomes:

$$\mathcal{L}_{\text{MLE}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{reg}} \tag{7}$$

Preference-Controlled MORL

The regularization term in Equation 7 is to make PACVAE aware of the preference variable r, but is still optimized with MLE which regards diversity as an implicit objective. Inspired by recent RL-based work, we propose a MORL method to fine-tune PACVAE, which introduces explicit objectives for both fidelity and diversity. Among the commonly used RL strategies, self-critical reinforcement learning (SCRL) is an appealing method to provide low-variant gradients and stabilize the training process (Rennie et al. 2017; Anderson et al. 2018; Huang et al. 2019; Wang et al. 2020). Based on the SCRL strategy, we propose a preference-controlled multi-objective reinforcement learning (PCMORL) model.

As shown in Figure 3, PCMORL reuses the prior network of PACVAE, which is finetuned with RL. Given an input $\boldsymbol{x}$ and a preference variable r, we sample a group of latent variables $Z = \left(\boldsymbol{z}^1, \boldsymbol{z}^2, ..., \boldsymbol{z}^K\right)$ from the prior distribution $p_\psi(\boldsymbol{z}|r)$ where K is the size of the group. Then we generate the corresponding group of candidates $\widetilde{G} = \left(\widetilde{\boldsymbol{y}}^1, \widetilde{\boldsymbol{y}}^2, ..., \widetilde{\boldsymbol{y}}^K\right)$ with the Monte-Carlo sampling:

$$\widetilde{\boldsymbol{y}}^k \sim \prod_k p_\theta\left(\widetilde{y}_t^k \mid \widetilde{\boldsymbol{y}}_{<t}^k, \boldsymbol{x}, \boldsymbol{z}^k\right), k \in [1, K] \tag{8}$$

where $\widetilde{y}_t^k$ denotes the t-th token of kth generated text sequence, and $\widetilde{\boldsymbol{y}}_{<t}^k$ denotes the generated sub-sequence preceding $\widetilde{y}_t^k$. Since we use the SCRL strategy to stabilize the learning process, we generate the baseline group of candidates $\overline{G} = \left(\overline{\boldsymbol{y}}^1, \overline{\boldsymbol{y}}^2, ..., \overline{\boldsymbol{y}}^K\right)$ by greedy decoding with the same group of latent variables Z.

Let $r_f(\cdot)$ be the fidelity-oriented reward function, calculating CIDEr scores (Vedantam, Zitnick, and Parikh 2015) which would not only encourage the fidelity of generated text but also penalize less informative phrases. Then the gradient for fidelity can be approximated by:

$$\nabla_\theta \mathcal{L}_f(\theta) \approx -\sum_{k=1}^{K}\left(r_f(\widetilde{\boldsymbol{y}}^k) - r_f(\overline{\boldsymbol{y}}^k)\right)\nabla_\theta \log p_\theta(\widetilde{\boldsymbol{y}}^k|\boldsymbol{x}, \boldsymbol{z}^k) \tag{9}$$

In terms of the diversity-oriented reward function $r_d(\cdot)$, we calculate the Self-CIDEr score (Wang and Chan 2019) for the candidate text groups. Self-CIDEr has shown a finer distinction ability (Wang and Chan 2019) on diversity than another commonly used metric Self-BLEU (Zhu et al. 2018).

Moreover, to get fine-grained rewards for each text in the group, we replace each text $\overline{\boldsymbol{y}}^k$ in $\overline{G}$ with $\widetilde{\boldsymbol{y}}^k$ in $\widetilde{G}$, resulting in other K reconstructed groups $\left(\widehat{G}^1, ..., \widehat{G}^K\right)$. For example, the i-th group $\widehat{G}^i = \left(\overline{\boldsymbol{y}}^1, ..., \widetilde{\boldsymbol{y}}^i, ..., \overline{\boldsymbol{y}}^K\right)$ is obtained by replacing $\overline{\boldsymbol{y}}^i$ in $\overline{G}$ with $\widetilde{\boldsymbol{y}}^i$. Then we measure the Self-CIDEr score for each group $\widehat{G}^i$ as the diversity-oriented reward, and the gradient is computed as follows:

$$\nabla_\theta \mathcal{L}_d(\theta) \approx -\sum_{k=1}^{K}\left(r_d(\widehat{G}^i) - r_d(\overline{G})\right)\nabla_\theta \log p_\theta(\widetilde{\boldsymbol{y}}^k|\boldsymbol{x}, \boldsymbol{z}^k) \tag{10}$$

We also use the preference variable with the MORL strategy, with the gradient computed as follows:

$$\nabla_\theta \mathcal{L}(\theta, r) = (1 - r)\alpha\nabla_\theta \mathcal{L}_f(\theta) + r\beta\nabla_\theta \mathcal{L}_d(\theta) \tag{11}$$

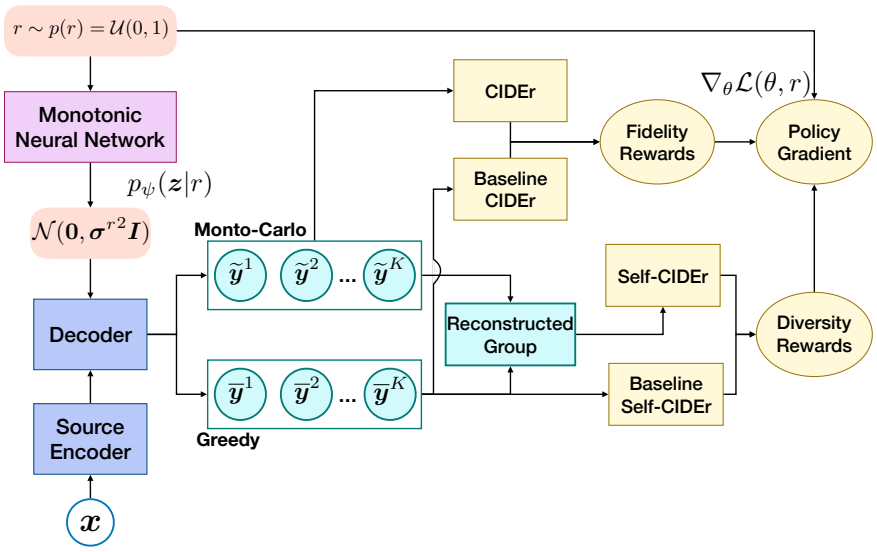


Figure 3: The architecture of PCMORL, which reuses the prior network of PACVAE and is fine-tuned with MORL.

where α and β are two hyperparameters. The user preference r is used to balance the two gradients.

The Inference Strategies

In the inference stage, we use two strategies to generate texts based on beam search. The first is a commonly used strategy in previous work (Fu, Feng, and Cunningham 2019; Huang et al. 2019; Ruan, Ling, and Zhu 2020), which keeps the top-1 text among all ranked texts through beam search. In this situation, beam search will not contribute to the diversity but the sampling process $\boldsymbol{z}^k \sim p_\psi(\boldsymbol{z}|r)$ will.

The second inference strategy lets beam search and the sampling process $\boldsymbol{z}^k \sim p_\psi(\boldsymbol{z}|r)$ both contribute to the diversity. Specifically, our model PCMORL first produces a text group $\widetilde{G}_l = \{\widetilde{\boldsymbol{y}}_m^k\}$ of size $M \times K$ where $\widetilde{\boldsymbol{y}}_m^k$ denotes the m-th top-ranking text via beam search given k-th latent variable $\boldsymbol{z}^k$. Then we reselect texts from $\widetilde{G}_l$ to construct a small group $\widetilde{G}_s$ of size K by removing duplicated texts and meanwhile minimizing the total rank number of beam search. And if beam size $M \geq K$, texts in $\widetilde{G}_l$ are guaranteed to be unique.

Task	Dataset	Train	Valid	Test	N_{ref}
Paraphrasing	Quora	116,263	3,000	30,000	1
Paraphrasing	MSCOCO	78,733	4,050	40,504	4
Captioning	MSCOCO	113,287	5,000	5,000	5

Table 1: Statistics of the datasets. N_{ref} represents the number of references for each data sample.

Experiments

Datasets

We experiment with two tasks including paraphrasing and image captioning. For paraphrasing, we follow recent work (Li et al. 2018; Fu, Feng, and Cunningham 2019; Chen et al. 2020; Zhou and Bhat 2021) and experiment on two commonly used datasets: Quora[1] and MSCOCO[2]. The Quora dataset was originally developed for duplicated question detection which contained about 140k pairs of paraphrase and 260k pairs of non-paraphrase sentences. We only use the paraphrase sentences and hold out 3k and 30k validation and test sets respectively. The MSCOCO dataset (Lin et al. 2014) was originally developed for image captioning. Each image has 5 corresponding captions, and we randomly choose 1 of the 5 captions as the source and consider the rest as the targets for paraphrasing. For image captioning, we also experiment on the MSCOCO dataset. The difference is that we use the "Karpathy" data split (Karpathy and Li 2015) with 5k images for validation, 5k images for testing, and the rest for training, which is consistent with most of previous work (Anderson et al. 2018; Huang et al. 2019; Wang et al. 2020; Pan et al. 2020; Shi et al. 2018a).

We keep the words that occur no less than 5 times in each dataset. The detailed statistics of the used datasets can be found in Table 1.

Evaluation

The generated sentences are evaluated with two types of metrics:

- **Fidelity**. This type of metrics includes traditional metrics measuring n-gram fidelity: CIDEr (Vedantam, Zitnick, and Parikh 2015), BLEU (Papineni et al. 2002), and

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

ROUGE (Lin 2004), which are computed with the publicly released code[3], and also include a metric measuring semantic fidelity: BERTScore (Zhang et al. 2020), which computes the similarities of BERT embeddings[4].

- **Diversity**. In terms of diversity, we use Self-CIDEr (Wang and Chan 2019) which has shown better distinction ability than Self-BLEU (Zhu et al. 2018). For example, the two text groups, G_1 ={"zebras grazing grass", "grazing grass", "zebras grazing"} and G_2 ={"zebras grazing", "zebras grazing", "zebras grazing"}, have the same Self-BLEU score of 1.0 while the Self-CIDEr score of G_1 will be lower than G_2.

Baseline Models

Two types of baseline models are compared including deterministic and variational models. The deterministic baselines include LSTM-Att which represent the encoder-decoder based on LSTM and the attention mechanism (Bahdanau, Cho, and Bengio 2015), Transformer (Vaswani et al. 2017), and XLAN which is a Transformer variant with higher-order attention and reaches state-of-the-art (SOTA) performance on image captioning (Pan et al. 2020). Moreover, for paraphrasing, we extend XLAN by stacking the XLAN encoder on a bidirectional LSTM encoder to make it position-aware, which is denoted by LSTM-XLAN. It is worth noting that a position embedding layer can also make the model position-aware, but we empirically find that using the LSTM encoder makes the model converge faster and perform better.

The variational models include SCVAE (Jain, Zhang, and Schwing 2017) and CVAE (Wang and Wan 2019), the backbone of which are XLAN and LSTM-XLAN for image captioning and paraphrasing, respectively.

Hyperparameters and Settings

The hidden sizes of embedding layers, bidirectional LSTM encoders, LSTM decoders, and attention layers are all set to 1,024. When dealing with the paraphrasing task, we use a randomly initialized word embedding layer. When dealing with the image captioning task, we use a pre-trained Faster R-CNN model (Anderson et al. 2018) to represent each image as an adaptive sequence of object-based feature vectors, $\boldsymbol{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_n\}$, where $n \in [10, 100]$ varies with input images and confidence thresholds following most of the recent SOTA models (Huang et al. 2019; Wang et al. 2020; Shi et al. 2018a; Zhou et al. 2020). The original dimension of each feature vector $\boldsymbol{e}_i$ is 2,048 and we project it to 1,024. All models are optimized with MLE for 70 epochs and RL for another 35 epochs. The learning rate is initialized as 0.0005 with a Noam schedule including 10,000 warmup steps in the MLE period and set to 0.00001 in the RL period. The batch size is set to 40. The default beam size is 3 and the generated text group size is 5. We set the hyperparameter $\lambda_\sigma = 2.0$ for paraphrasing and $\lambda_\sigma = 1.0$ for captioning. We set $\alpha = 0.5$ and $\beta = 1.0$ for paraphrasing on Quora, $\alpha = 1.0$ and $\beta = 1.0$ for other datasets.

[3]https://github.com/tylin/coco-caption

[4]The used BERT version is *roberta-large_L17_no-idf_version=0.3.8(hug_trans=4.5.0)*

Types	Models	Metrics				
		B4	RG	CD	BS	SC
MLE	LSTM-Att	24.5	53.1	222.2	50.3	0.0
	Transformer	25.3	52.8	241.6	53.3	0.0
	LSTM-XLAN	26.8	54.4	247.5	53.2	0.0
	SCVAE	26.3	54.3	246.0	53.3	26.0
	CVAE	26.2	53.5	240.5	52.7	12.0
RL	LSTM-XLAN	27.2	55.7	254.3	53.9	0.0
	SCVAE	27.1	55.6	254.0	53.9	20.8
	CVAE	26.9	55.4	252.5	53.8	3.8
	PCMORL$_{(0.0)}$	**28.5**	**56.7**	**267.8**	**55.5**	12.3
	PCMORL$_{(0.9)}$	25.0	53.3	235.9	51.3	**51.0**

(a) Quora for Paraphrasing

Types	Models	Metrics				
		B4	RG	CD	BS	SC
MLE	LSTM-Att	29.8	53.4	107.6	57.1	0.0
	Transformer	23.5	48.4	87.1	51.5	0.0
	LSTM-XLAN	29.9	53.3	109.7	58.2	0.0
	SCVAE	29.8	53.2	109.2	57.8	16.0
	CVAE	29.2	52.9	107.6	57.5	14.1
RL	LSTM-XLAN	**31.8**	54.9	119.0	**59.2**	0.0
	SCVAE	31.3	54.8	118.7	59.0	15.4
	CVAE	31.6	**55.0**	**119.1**	59.1	1.5
	PCMORL$_{(0.0)}$	31.4	54.9	**119.1**	58.9	15.6
	PCMORL$_{(0.9)}$	26.9	51.9	105.3	56.8	**62.7**

(b) MSCOCO for Paraphrasing

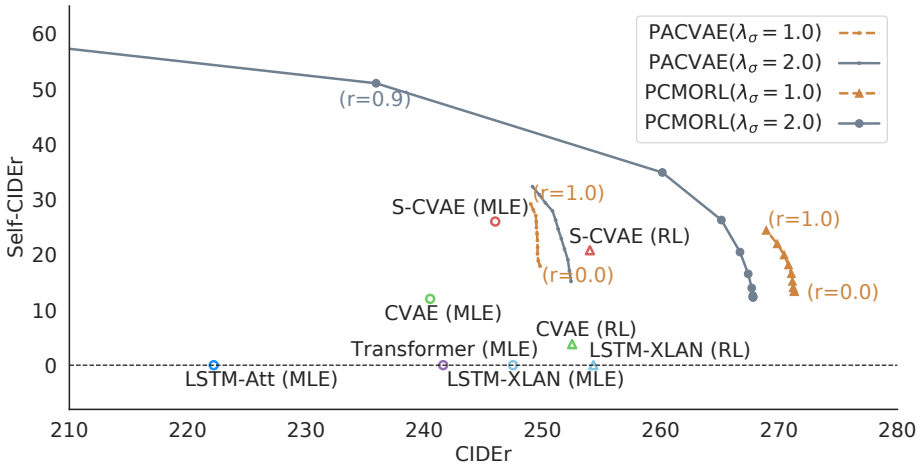
Types	Models	Metrics				
		B4	RG	CD	BS	SC
MLE	LSTM-Att	37.1	57.3	114.6	63.3	0.0
	Transformer	36.8	57.0	115.9	63.7	0.0
	XLAN	37.9	58.0	120.3	64.8	0.0
	SCVAE	37.1	57.5	119.1	64.5	18.5
	CVAE	37.2	57.6	119.7	64.7	11.0
RL	XLAN	**39.5**	59.0	**131.2**	**66.4**	0.0
	SCVAE	38.8	**59.1**	130.9	66.2	11.5
	CVAE	38.6	**59.1**	131.0	66.2	1.6
	PCMORL$_{(0.0)}$	38.9	59.0	131.0	66.3	16.2
	PCMORL$_{(0.9)}$	35.3	57.2	123.4	65.2	**54.9**

(c) MSCOCO for Image Captioning

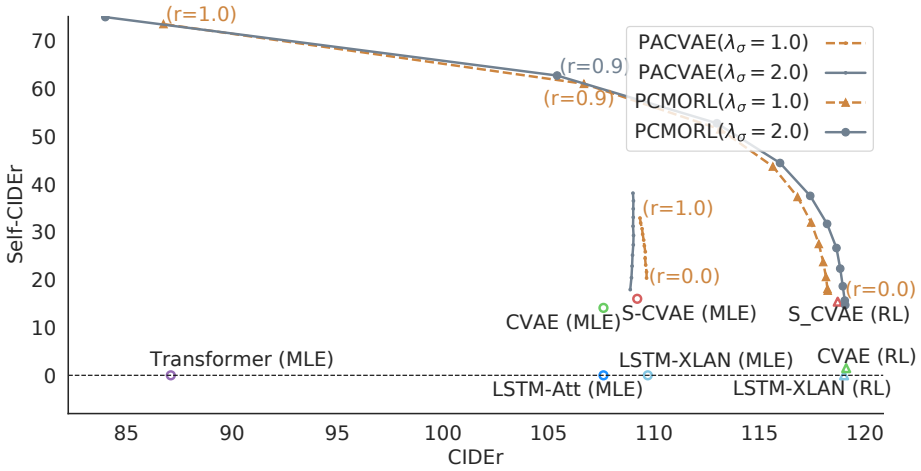
Table 2: The results of different models in MLE and RL training periods. B4, RG, CD, BS, and SC stand for BLEU-4, ROUGE, CIDEr, BERTScore, and Self-CIDEr respectively. For all metrics, the higher is the better. The bold represent the best scores. The numbers in the brackets after PCMORL represent the value of user preference r.

Main Results

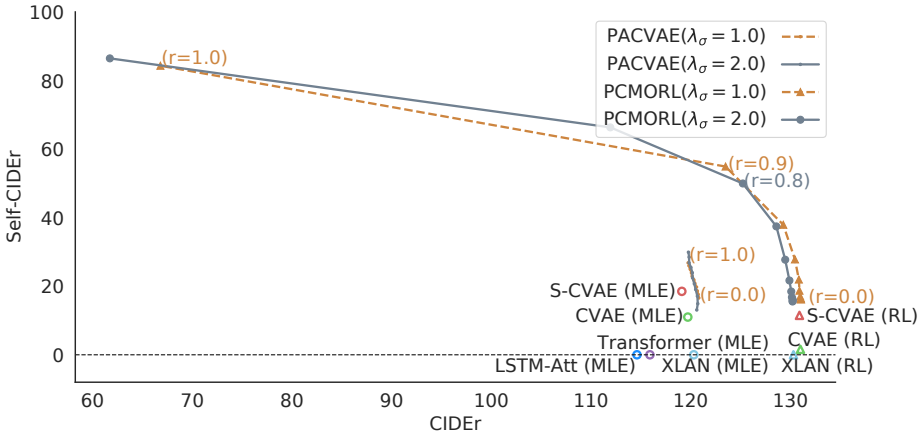
Table 2 presents the performance of our model *PCMORL* and the compared models on three text generation scenarios



(a) Quora for Paraphrasing



(b) MSCOCO for Paraphrasing



(c) MSCOCO for Image Captioning

Figure 4: Fidelity-diversity space for the three scenarios. The mainly compared metrics are CIDEr and Self-CIDEr scores, representing fidelity and diversity respectively. Higher values in both axes are better.

with the first inference strategy. As shown, when the preference $r = 0.0$, PCMORL achieves similar CIDEr scores to its

deterministic backbone, LSTM-XLAN or XLAN in the RL period. A surprising finding on the Quora dataset in Table 2a is that PCMORL achieves the CIDEr score of 267.8 points which is 13.5 points higher than its deterministic baseline LSTM-XLAN. The reason may be that jointly optimizing CIDEr and Self-CIDEr scores with MORL alleviates overfitting to some extent.

When the preference $r = 0.9$, PCMORL has the Self-CIDEr scores of 51.0, 62.7, and 54.9 points for the three scenarios, respectively, which are higher than SCVAE and CVAE in both MLE and RL periods. It means that considering diversity as an explicit objective can further improve diversity over using an implicit objective in CVAE-based models. Moreover, regarding diversity, CVAE-based models are not suitable to fine-tune with RL. Specifically, in the RL period, CVAE degrades into nearly a deterministic model, which gets the Self-CIDEr scores of 3.8, 1.5, and 1.6 in the three scenarios, respectively. SCVAE has higher Self-CIDEr scores than CVAE but gets lower Self-CIDEr scores than itself in the MLE period. PCMORL can further improve diversity over CVAE-based models.

In addition to automatic quantitative evaluation, many previous studies on text generation have also included human evaluation of semantic fidelity, which is a high-level property of the generated text. However, when it comes to evaluating diversity, automatic quantitative metrics are more appropriate as we expect syntactic diversity, which is a low-level property. Moreover, assessing the diversity of a group of texts is not a straightforward task for humans (Tevet and Berant 2021). Therefore, in this paper, we mainly rely on quantitative evaluation methods.

Fidelity-Diversity Trade-off

Since fidelity and diversity are two conflicting objectives, Table 2 is not sufficient to jointly compare the fidelity and diversity of all the models. Thus we show the results of different models in the fidelity-diversity space in Figure 4 where the mainly compared metrics are CIDEr and Self-CIDEr scores, representing fidelity and diversity, respectively. We explore the hyperparameter λ_σ with values 1.0 and 2.0, and obtain trade-off curves by tuning the preference variable r from 0.0 to 1.0 with an interval of 0.1 for PACVAE and PCMORL. As shown in Figure 4, the deterministic models have the Self-CIDEr score of 0.0. SCVAE and CVAE improve the Self-CIDEr scores over the deterministic backbone LSTM-XLAN or XLAN. The trade-off curve of PCMORL sits at the top right of other models, which performs the best when jointly considering fidelity and diversity. PCMORL with the hyperparameter $\lambda_\sigma = 2.0$ has a higher bound for the Self-CIDEr scores than that with $\lambda_\sigma = 1.0$. An interesting finding in Figure 4a is that PCMORL with $\lambda_\sigma = 1.0$ has a narrow range of Self-CIDEr scores due to the narrow range of Self-CIDEr scores of PACVAE. It indicates that the pre-training process of PACVAE matters.

Another advantage of PCMORL over deterministic models and CVAE-based models is that we can obtain infinite solutions by tuning the preference variable r in the inference stage.

Models	Quora		MC-1		MC-2	
	CD	SC	CD	SC	CD	SC
(LSTM-)XLAN	234.7	52.8	114.1	54.6	123.9	54.3
PCMORL$_{r=0.0}$	**246.1**	52.9	**115.4**	55.1	**125.9**	54.4
PCMORL$_{r=0.8}$	241.1	56.8	110.7	61.2	125.3	56.6
PCMORL$_{r=0.9}$	220.8	**62.8**	103.9	**66.2**	120.5	**62.1**

Table 3: Results of models under the second inference strategy which lets beam search and sampling of the latent variable both contribute to the diversity. MC-1 and MC-2 denote the MSCOCO datasets for paraphrasing and image captioning, respectively.

Compared with PACVAE We further compare PCMORL with PACVAE in Figure 4. Generally, the Self-CIDEr score of PACVAE grows with the increase of the preference r while the CIDEr score does not change much. This is due to the MLE objective which lets the sampling of the latent variable $\boldsymbol{z}$ not significantly influence the reconstruction of the target text. After fine-tuning the prior network of PACVAE, PCMORL can further improve the trade-off curve.

The Second Inference Strategy

We apply the second inference strategy which lets beam search and the sampling of $\boldsymbol{z}$ both contribute to the diversity. In Table 3, we compare PCMORL with its deterministic backbone in the RL period where the beam size is set to 5. Generally, models with the second inference strategy have much higher Self-CIDEr scores than the first strategy. For example, LSTM-XLAN or XLAN achieves the Self-CIDEr scores of 52.8, 54.6, and 54.3 for the three scenarios, respectively, with the second inference strategy while it has the Self-CIDEr scores of 0.0 with the first inference strategy. However, the CIDEr scores decrease from 254.3, 119.0 and 131.2 points to 234.7, 114.1 and 123.9 points, respectively. As stated in previous work (Gupta et al. 2018), beam search itself does not yield multiple high-quality texts in a principled way.

When $r = 0.0$, PCMORL outperforms the baseline LSTM-XLAN or XLAN by 11.4, 1.3, and 2.0 points of CIDEr scores with similar Self-CIDEr scores. It means that mixing beam search with the sampling of $\boldsymbol{z}$ could improve the quality to some extent. When $r = 0.9$, PCMORL further improves the Self-CIDEr scores to 62.8, 66.2, and 62.1 points.

Case Study

To directly see the effect of tuning preference r, we show two cases of PCMORL with the first inference strategy in Figure 5. The two cases are from the Quora dataset for paraphrasing and the MSCOCO dataset for image captioning respectively. For each input data, we generate 5 sentences by sampling 5 different latent variable $\boldsymbol{z}$. In the first case, PCMORL generates 3 repeated sentences when $r = 0.0$ and 1 repeated sentences when $r = 0.8$. In the second case, PCMORL generates 5 unique sentences when $r = 0.8$. The

Inputs	PCMORL (r=0.0)	PCMORL (r=0.8)
Can height be increased after age 21?	how can i increase my height after 21 ?	can i increase the height after 21 ?
	how can i increase my height after 21 ? (repeated)	how can i increase height after 21 ?
	how can i increase my height after 21 years ?	how can i increase my height after 21 ?
	how can i increase my height after 21 ? (repeated)	is it possible to increase height after 21 ?
	how can i increase my height after 21 ? (repeated)	is it possible to increase height after 21 ? (repeated)
	two parrots sitting on top of a tree branch	two parrots sitting on a branch of a tree
	two colorful parrots perched on a tree branch	two colorful parrots are sitting on top of a branch
	two parrots sitting on top of a tree branch (repeated)	two colorful parrots sitting on a tree branch
	two parrots sitting on top of a tree branch (repeated)	two colorful parrots sitting on a branch in a table
	two parrots sitting on top of a tree branch (repeated)	two colorful parrots perched on a tree branch

Figure 5: The case study of PCMORL. Two samples are from the Quora and MSCOCO datasets for paraphrasing and image captioning, respectively.

diversity is improved by increasing the preference r. In practice, selecting a proper value of the preference r will make PCMORL more attractive to users than that setting $r = 0.0$.

Conclusion and Future Work

In this paper, we propose a preference-controlled MORL method for conditional text generation, which can tune the trade-off of fidelity and diversity by changing the user preference r in the inference stage. Firstly, we propose a CVAE variant, PACVAE, with a monotonic neural network to tune the variance of the Gaussian prior distribution. Then, we propose PCMORL which fine-tunes the prior network of PACVAE and introduces two explicit objectives for fidelity and diversity, respectively. The experiments show that PCMORL outperforms deterministic and CVAE-based baselines in the fidelity-diversity space, and can obtain infinite solutions by tuning the preference variable.

Our PCMORL strategy can be used to fine-tune pre-trained language models (PLMs), but we are more interested in lightweight fine-tuning, such as adapters or prompt-tuning. We leave the combination of PCMORL and lightweight fine-tuning PLMs in future work.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (2022YFC3340904), and the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *CVPR 2018*, 6077–6086.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In Bengio, Y.; and LeCun, Y., eds., *ICLR 2015*.

Buciumas, S. 2019. Reinforcement Learning Models for Abstractive Text Summarization. In Lo, D.; Kim, D.; and Gamess, E., eds., *Proceedings of the 2019 ACM Southeast Conference, ACM SE '19, Kennesaw, GA, USA, April 18-20, 2019*, 270–271. ACM.

Caccia, M.; Caccia, L.; Fedus, W.; Larochelle, H.; Pineau, J.; and Charlin, L. 2020. Language GANs Falling Short. In *ICLR 2020*.

Chen, W.; Tian, J.; Xiao, L.; He, H.; and Jin, Y. 2020. A Semantically Consistent and Syntactically Variational Encoder-Decoder Framework for Paraphrase Generation. In Scott, D.; Bel, N.; and Zong, C., eds., *COLING 2020*, 1186–1198.

Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *EMNLP 2014*, 1724–1734.

Dai, B.; Fidler, S.; Urtasun, R.; and Lin, D. 2017. Towards Diverse and Natural Image Descriptions via a Conditional GAN. In *ICCV 2017*, 2989–2998.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J. G.; Le, Q. V.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *ACL 2019*, 2978–2988.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for

Language Understanding. In *NAACL-HLT 2019*, 4171–4186.

Du, J.; Li, W.; He, Y.; Xu, R.; Bing, L.; and Wang, X. 2018. Variational Autoregressive Decoder for Neural Response Generation. In *EMNLP 2018*, 3154–3163.

Fu, Y.; Feng, Y.; and Cunningham, J. P. 2019. Paraphrase Generation with Latent Bag of Words. In *NeurIPS 2019*, 13623–13634.

Guo, J.; Lu, S.; Cai, H.; Zhang, W.; Yu, Y.; and Wang, J. 2018. Long Text Generation via Adversarial Training with Leaked Information. In *AAAI 2018*, 5141–5148.

Gupta, A.; Agarwal, A.; Singh, P.; and Rai, P. 2018. A Deep Generative Framework for Paraphrase Generation. In *AAAI 2018*, 5149–5156.

Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017. Toward Controlled Generation of Text. In *ICML 2017*, volume 70, 1587–1596.

Hu, Z.; Yang, Z.; Salakhutdinov, R.; and Xing, E. P. 2018. On Unifying Deep Generative Models. In *ICLR 2018*.

Huang, L.; Wang, W.; Chen, J.; and Wei, X. 2019. Attention on Attention for Image Captioning. In *ICCV 2019*, 4633–4642.

Jain, U.; Zhang, Z.; and Schwing, A. G. 2017. Creativity: Generating Diverse Questions Using Variational Autoencoders. In *CVPR 2017*, 5415–5424.

Karpathy, A.; and Li, F. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR 2015*, 3128–3137.

Keneshloo, Y.; Shi, T.; Ramakrishnan, N.; and Reddy, C. K. 2020. Deep Reinforcement Learning for Sequence-to-Sequence Models. *IEEE Trans. Neural Networks Learn. Syst.*, 31(7): 2469–2489.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL 2020*, 7871–7880.

Li, Z.; Jiang, X.; Shang, L.; and Li, H. 2018. Paraphrase Generation with Deep Reinforcement Learning. In *EMNLP 2018*, 3865–3878.

Lin, C. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *ACL 2004*, 74–81.

Lin, T.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In Fleet, D. J.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, 740–755. Springer.

Lin, Z.; Feng, M.; dos Santos, C. N.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A Structured Self-Attentive Sentence Embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, abs/1907.11692.

Nallapati, R.; Zhou, B.; dos Santos, C. N.; Gülçehre, Ç.; and Xiang, B. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *CoNLL 2016*, 280–290.

Pan, Y.; Yao, T.; Li, Y.; and Mei, T. 2020. X-Linear Attention Networks for Image Captioning. In *CVPR 2020*, 10968–10977.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL 2002*, 311–318.

Parikh, A. P.; Täckström, O.; Das, D.; and Uszkoreit, J. 2016. A Decomposable Attention Model for Natural Language Inference. In Su, J.; Carreras, X.; and Duh, K., eds., *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2249–2255. The Association for Computational Linguistics.

Paulus, R.; Xiong, C.; and Socher, R. 2018. A Deep Reinforced Model for Abstractive Summarization. In *ICLR 2018*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67.

Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2016. Sequence Level Training with Recurrent Neural Networks. In *ICLR 2016*.

Rennie, S. J.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-Critical Sequence Training for Image Captioning. In *CVPR 2017*, 1179–1195.

Ruan, Y.; Ling, Z.; and Zhu, X. 2020. Condition-Transforming Variational Autoencoder for Generating Diverse Short Text Conversations. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 19(6): 79:1–79:13.

Shao, H.; Wang, J.; Lin, H.; Zhang, X.; Zhang, A.; Ji, H.; and Abdelzaher, T. F. 2021. Controllable and Diverse Text Generation in E-commerce. In *WWW 2021*, 2392–2401.

Shi, Z.; Chen, X.; Qiu, X.; and Huang, X. 2018a. Toward Diverse Text Generation with Inverse Reinforcement Learning. In *IJCAI 2018*, 4361–4367.

Shi, Z.; Chen, X.; Qiu, X.; and Huang, X. 2018b. Toward Diverse Text Generation with Inverse Reinforcement Learning. In *IJCAI 2018*, 4361–4367.

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In *NeurIPS, 2015*, 3483–3491.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *NeurIPS 2014*, 3104–3112.

Tevet, G.; and Berant, J. 2021. Evaluating the Evaluation of Diversity in Natural Language Generation. In *EACL 2021*, 326–346.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NeurIPS 2017*, 5998–6008.

Vedantam, R.; Zitnick, C. L.; and Parikh, D. 2015. CIDEr: Consensus-based image description evaluation. In *CVPR 2015*, 4566–4575.

Vinyals, O.; and Le, Q. V. 2015. A Neural Conversational Model. In *ICML Deep Learning Workshop*.

Wang, L.; Schwing, A. G.; and Lazebnik, S. 2017. Diverse and Accurate Image Description Using a Variational Auto-Encoder with an Additive Gaussian Encoding Space. In *NeurIPS 2017*, 5756–5766.

Wang, Q.; and Chan, A. B. 2019. Describing Like Humans: On Diversity in Image Captioning. In *CVPR 2019*, 4195–4203.

Wang, T.; Huang, J.; Zhang, H.; and Sun, Q. 2020. Visual Commonsense R-CNN. In *CVPR 2020*, 10757–10767.

Wang, T.; and Wan, X. 2019. T-CVAE: Transformer-Based Conditioned Variational Autoencoder for Story Completion. In *IJCAI 2019*, 5233–5239.

Wehenkel, A.; and Louppe, G. 2019. Unconstrained Monotonic Neural Networks. In *NeurIPS 2019*, 1543–1553.

Wu, L.; Tian, F.; Qin, T.; Lai, J.; and Liu, T. 2018. A Study of Reinforcement Learning for Neural Machine Translation. In *EMNLP 2018*, 3612–3621.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A. C.; Salakhutdinov, R.; Zemel, R. S.; and Bengio, Y. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, 2048–2057.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J. G.; Salakhutdinov, R.; and Le, Q. V. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NeurIPS 2019*, 5754–5764.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *AAAI 2017*, 2852–2858.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. In *ICLR 2020*.

Zhou, J.; and Bhat, S. 2021. Paraphrase generation: A survey of the state of the art. In *EMNLP 2021*, 5075–5086.

Zhou, L.; Palangi, H.; Zhang, L.; Hu, H.; Corso, J. J.; and Gao, J. 2020. Unified Vision-Language Pre-Training for Image Captioning and VQA. In *AAAI 2020*, 13041–13049.

Zhu, Y.; Lu, S.; Zheng, L.; Guo, J.; Zhang, W.; Wang, J.; and Yu, Y. 2018. Texygen: A Benchmarking Platform for Text Generation Models. In *SIGIR 2018*, 1097–1100.