Complementary Auxiliary Classifiers for Label-Conditional Text Generation

Yuan Li,[1*] **Chunyuan Li,**[2†] **Yizhe Zhang,**[2] **Xiujun Li,**[2] **Guoqing Zheng,**[2]
Lawrence Carin,[1] **Jianfeng Gao**[2]
[1]Duke University, [2]Microsoft Research, Redmond
[1]{yl558, lcarin}@duke.edu, [2]{chunyl, yizzhang, xiul, zheng, jfgao}@microsoft.com

Abstract

Learning to generate text with a given label is a challenging task because natural language sentences are highly variable and ambiguous. It renders difficulties in trade-off between sentence quality and label fidelity. In this paper, we present CARA to alleviate the issue, where two auxiliary classifiers work simultaneously to ensure that (1) the encoder learns disentangled features and (2) the generator produces label-related sentences. Two practical techniques are further proposed to improve the performance, including annealing the learning signal from the auxiliary classifier, and enhancing the encoder with pre-trained language models. To establish a comprehensive benchmark fostering future research, we consider a suite of four datasets, and systematically reproduce three representative methods. CARA shows consistent improvement over the previous methods on the task of label-conditional text generation, and achieves state-of-the-art on the task of attribute transfer.

Introduction

Text generation is an important challenge in natural language processing (NLP). Most previous research in this area has focused on unsupervised text generation (Bengio et al. 2003; Mikolov et al. 2010), and success has been achieved recently, such as pre-trained generative training (Radford et al. 2019; Yang et al. 2019). However, these setups measure the ability of models to generate the coherent content of a sentence, but do not address more natural human communication in a given context. For example, generating more engaging conversions requires conditioning on personality in image captioning and dialogue systems. Synthesis of coherent sentences requires conditioning on a given topic/sentiment. These setups can be formulated as label-conditional text generation.

Several attempts have been made to solve this problem. Most are based on deep latent variable models (LVMs), such as variational autoencoders (VAE) (Kingma and Welling 2013) and their conditional variants (Hu et al. 2017; Yang et al. 2017). One attractive property of these models is that they map sentences to global language features in a latent space, and allow manipulation of generated sentences with a specific tense, sentiment or topic.

One prominent challenge for LVMs is to learn smooth and disentangled latent representations, such that generation from this space results in realistic sentences and can be effectively controlled in the category it belongs to during conditional decoding. Hu *et al*. (Hu et al. 2017) proposed Ctrl-Gen, which uses a VAE to represent sentences as smooth Gaussian distributions, regularized towards a standard normal prior distribution in the latent space. However, VAE-based methods (Kingma and Welling 2013) are difficult to train due to the notorious posterior collapse and KL vanishing (Bowman et al. 2015). To solve this problem, ARAE (Zhao et al. 2017) uses adversarial learning to construct a flexible prior distribution. It further proposes an auxiliary classifier in the latent space to disentangle the learned latent feature from the conditional labels.

In this paper we re-examine the relationship between disentangled feature learning and label-conditional generation, and show that the former does not necessarily lead to the latter. We term this issue as a *non-identifiability* issue. This is manifested when the generator is able to resemble training samples perfectly but degenerates to ignoring conditional label and solely relying on the latent code, even though the encoder learns label-agnostic representations.

To solve this issue, we propose Complementary Auxiliary classifier Regularized Auto-encoder (CARA), where there are two classifiers to predict labels: a classifier in the latent space is used for disentangling feature learning as in ARAE, and a complementary classifier in the observation space is proposed to encourage the generator to contain the conditional label. We prove that the proposed complementary auxiliary classifiers introduce a cycle-consistency loss for conditional labels, leading to maximizing the mutual information between generated sentences and their corresponding labels. Further, we explore the trade-off between generation quality and label-conditional accuracy, and propose two practical techniques for improving overall performance: (i) an annealing training schedule for the proposed complementary auxiliary classifier to gradually incorporate

[*]Work performed during an internship at Microsoft Research
[†]Corresponding author

label-conditional supervision signals; (*ii*) BERT (Devlin et al. 2018) is used as an encoder to provide more universal and generalizable latent representations. We show it provides empirically improved LVM-based conditional text generation. We build a suite of four datasets for comprehensive study of label-conditional text generation. Quantitative and qualitative experimental results demonstrate that the proposed techniques consistently shows improved performance. We further apply CARA to the style transfer task, where CARA achieves state-of-the-art performance.

Preliminaries

Consider a training set $\mathcal{S} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ of pairwise data, where $\boldsymbol{x}_n = [x_{n,1}, x_{n,2}, ..., x_{n,T_n}]$ is a text sequence of length T_n, and $y_n \in \mathcal{Y}$ is its corresponding label/attribute/class. The goal of conditional text generation is to generate a new $\boldsymbol{x}$ for a given y. Due to the diversity of natural language, a latent code $\boldsymbol{z}$ is introduced to characterize diverse information associated with y, required to generate faithful and variable $\boldsymbol{x}$. Typically, $\boldsymbol{z}$ is drawn from an easily-sampled prior distribution $p(\boldsymbol{z})$, such as Gaussian $\mathcal{N}(0, \mathbf{I})$.

Given $\boldsymbol{z}$ and y, the sampling procedure for the conditional $p_G(\boldsymbol{x}|\boldsymbol{z}, y) = \prod_{t=1}^T p_G(x_t|\boldsymbol{x}_{<t}, \boldsymbol{z}, y)$ is performed in a sequential manner, and an auto-regressive generator model G is used to generate x_t at the every time step:

$$x_t = G(\boldsymbol{x}_{<t}, \boldsymbol{z}, y), \tag{1}$$

where $\boldsymbol{x}_{<t}$ indicates all tokens before t. The synthesis of a given sentence continues until the end-of-sentence symbol is manifested.

Auto-encoder for Conditional Text Generation

The generator G is typically learned by maximizing the marginal log likelihood $\log p_G(\boldsymbol{x}) = \log \int p_G(\boldsymbol{x}|\boldsymbol{z}, y)p(\boldsymbol{z})p(y)\mathrm{d}\boldsymbol{z}\mathrm{d}y$, where $p(y)$ is the label distribution. However, the marginalizing integral wrt $\boldsymbol{z}$ is intractable to compute for many generator choices. Thus, variational inference is considered, and the true posterior $p_G(\boldsymbol{z}|\boldsymbol{x}) \propto p_G(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$ is approximated via the variational distribution $q_E(\boldsymbol{z}|\boldsymbol{x})$, implemented via an encoder:

$$\boldsymbol{z} = E(\boldsymbol{x}, \varsigma), \ \ \varsigma \sim q_0(\varsigma) \tag{2}$$

where $q_0(\varsigma)$ is a noise distribution. The observed sentence $\boldsymbol{x}$ can be represented using $\tilde{\boldsymbol{z}} \sim q_E(\boldsymbol{z}|\boldsymbol{x})$, and reconstructed as $\hat{\boldsymbol{x}} \sim p_G(\boldsymbol{x}|\tilde{\boldsymbol{z}}, y)$, where y is the corresponding label of $\boldsymbol{x}$. A reconstruction loss is applied to the generated sentence and observation:

$$\min_{E,G} \mathcal{L}_{\text{rec}} = -\mathbb{E}_{\tilde{\boldsymbol{z}} \sim q_E(\boldsymbol{z}|\boldsymbol{x}), y \sim p(y)}[\log p_G(\boldsymbol{x}|\tilde{\boldsymbol{z}}, y)]. \tag{3}$$

There are challenges in optimizing the auto-encoder objective in (3) for conditional text generation.

- **Smoothness**: The learned latent code $\boldsymbol{z}$ should be a smooth distribution such that samples or interposition of samples from it can lead to plausible sentence generation. This problem is inherited from using an auto-encoder for generic sentence generation.
- **Disentanglement**: For better control of attributes, $\boldsymbol{z}$ and y are desired to be disentangled, where the features encoded in each variable are exclusive of each other. Independence constraints are applied on these two variables in prior work, such as reconstruction of $\boldsymbol{z}$ through an encoder (Hu et al. 2017) and using a latent space discriminator to enforce $\boldsymbol{z}$ to be independent from y (Zhao et al. 2017).

Adversarially Regularized Auto-Encoders

Various techniques have been proposed to learn smooth and disentangled latent representations (Higgins et al. 2017; Alemi et al. 2017; Fu et al. 2019). In the context of conditional text generation, adversarially regularized autoencoders (ARAEs) (Zhao, Zhao, and Eskenazi 2017) have been introduced. Learning ARAEs can be viewed as proceeding in two alternating steps, including *disentangled feature learning* and *adversarial feature generation*.

In the first step, in addition to the reconstruction objective in (3), the inferred $\boldsymbol{z}$ is trained to only contain information exclusive of $\boldsymbol{y}$. This disentanglement objective is achieved using an auxiliary classifier $C(\boldsymbol{z})$:

$$\min_E \max_C \mathcal{L}_{\text{disentangle}} = \mathbb{E}_{\boldsymbol{z} \sim q_E(\boldsymbol{z}|\boldsymbol{x})}[\log p_C(y|\boldsymbol{z})], \tag{4}$$

where the classifier C is learned to categorize $\boldsymbol{z}$ into its corresponding labels, while E is forced to infer $\boldsymbol{z}$ to fake the classifier C. Ideally, at convergence, one cannot predict $\boldsymbol{y}$ using $\boldsymbol{z}$, and thus $\{\boldsymbol{y}, \boldsymbol{z}\}$ are disentangled.

In the second step, we learn a prior using a neural sampler $\boldsymbol{z} = S(\epsilon)$ to replace the fixed prior, where $\epsilon \sim p_0(\epsilon)$ is an auxiliary distribution that one may easily draw samples from. The neural sampler is trained to generate features, to match the distribution $q(\boldsymbol{z})$ learned in the first step.

A discriminator $D(\boldsymbol{z})$ is introduced to distinguish the domain of $\boldsymbol{z}$, with $d = 1$ indicating $\boldsymbol{z}$ comes from $E(\boldsymbol{x})$ and $d = 0$ indicating $\boldsymbol{z}$ comes from $S(\epsilon)$. The min-max objective in the latent space can be written as:

$$\min_{E,S} \max_D \ \ \mathcal{L}_{\text{adversarial}} = \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})}[\log p_D(d = 1|E(\boldsymbol{x}))] + \mathbb{E}_{\epsilon \sim p_0(\epsilon)}[\log p_D(d = 0|S(\epsilon))]. \tag{5}$$

where $p(\boldsymbol{x})$ is data distribution. When the optimum is achieved, the generated feature distribution $\pi(\boldsymbol{z})$ induced by $\boldsymbol{z} = S(\epsilon)$ can match the disentangled distribution $q_E(\boldsymbol{z}) = \int q_E(\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$.

The two steps are updated iteratively. Since (4) in the first step ensures the disentangled representation of $q(\boldsymbol{z})$ wrt to $p(y)$, and (5) in the second step ensures the marginal distribution matches $\pi(\boldsymbol{z}) = q(\boldsymbol{z})$, the two steps together can guarantee that the generated features $\pi(\boldsymbol{z})$ from the neural sampler S can characterize the disentangled representation wrt to $p(y)$ at its optimum.

The full objective of ARAEs for the encoder E and generator G is:

$$\min_{E,G} \mathcal{L}_{\text{ARAE}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{disentangle}} + \mathcal{L}_{\text{adversarial}} \tag{6}$$

We see that ARAE can be viewed as regularizing the basic auto-encoder objective in (3) with a disentanglement term and an adversarial term.

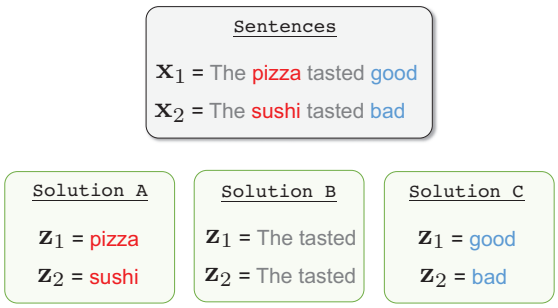

Figure 1: Illustration of possible solutions to the disentangled feature learning objective in (7). For the example sentences shown in the top row, three possible solutions are provided in the bottom row. Solution A and B are valid to (7) and Solution C is false.

Advantages of ARAEs ARAE has the following characteristics: (i) it provides a smooth latent space for discrete sequences, with a flexible learned prior. Compared to variational auto-encoder (VAE)-based methods (Kingma and Welling 2013), ARAE does not suffer from KL vanishing or posterior collapse (Bowman et al. 2015; Kingma and Welling 2013; Chen et al. 2016) caused by using KL-divergence with a fixed prior as regularization in the latent space. (ii) the classifier C applied in the latent space explicitly constrains the encoder E to only record attribute-independent features, encouraging disentanglement of latent features and attributes during encoding, and thus improving control and manipulation of attributes during conditional generation performed by generator G.

Complementary Auxiliary Classifiers

Pitfalls of ARAEs In ARAE the generator G learns to generate text $\boldsymbol{x}$ from a joint distribution constructed by the latent code $\boldsymbol{z}$ and the label condition y. During the disentangled feature learning stage of ARAE, the encoder E is trained to satisfy two objective:

$$\min_{E} \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{disentangle}} \tag{7}$$

This means that (i) the reconstruction $\mathcal{L}_{\text{rec}}$ in (3) encourages $\boldsymbol{z}$ to be unique in the latent space, so that the original $\boldsymbol{x}$ can be perfectly reconstructed; and (ii) the disentanglement $\mathcal{L}_{\text{disentangle}}$ in (4) encourages $\boldsymbol{z}$ to be disentangled from y, thus y becomes the only source to control the label of the generated sentences. However, we argue that it is challenging for the encoder E to produce $\boldsymbol{z}$ that simultaneously satisfies both objectives, and generator G to only rely on $\boldsymbol{z}$ (ignore the dependence of y) during generation.

Issues on Disentanglement vs Controllable Generation The above implies that disentanglement does not necessarily lead to controllable generation. We call this phenomenon the *non-identifiability* issue in disentangled feature learning. This problem occurs only under the assumption that training samples are non-parallel, where there is no sentence pair that has the same content but the opposite labels. This is common in many cases, such as text style transfer and conditional generation with non-parallel data.

We construct an example in Figure 1 to explain this further, where the dataset only contains two sentence samples $\mathcal{S}_1 = \{(\boldsymbol{x}_1, y=1), (\boldsymbol{x}_2, y=0)\}$. Here the conditioning information y is the sentiment, and the auxiliary classifier C learns to categorize "`good`" into $y=1$ and "`bad`" into $y=0$. We show three possible solution candidates. First, the disentanglement objective encourages the encoder to learn to summarize label-agnostic information, such as "`The pizza tasted`" and "`The sushi tasted`" into $\boldsymbol{z}$, which is independent from the label-related information such as "`good`" and "`bad`". Therefore, Solutions A and B are valid, and Solution C is invalid to the disentanglement objective. Further, it can be shown that both Solutions A and B satisfy the reconstruction objective, as the the joint $\{\boldsymbol{z}, y\}$ is unique enough to identify its corresponding $\boldsymbol{x}$. If Solution A is chosen, the model can perfectly generate $\mathcal{S}_1$ using $p_G(\boldsymbol{x}|\boldsymbol{z})$, rather than $p_G(\boldsymbol{x}|\boldsymbol{z}, y)$.

Ideally, the generator is expected to synthesize sentences successfully conditioned on any combination of label-agnostic and label-related information, by training with the objectives in (6). However, as the dataset is non-parallel, the model only sees a limited combination of label-agnostic and label-related information. For example, in the illustrative dataset, the model is only able to see pairs of "`pizza, good`" and "`sushi, bad`", instead of a full set of pairs (*e.g.,* "`pizza, good`", "`pizza, bad`", "`sushi, good`" and "`sushi, bad`"). Thus, by only relying on one source such as "`pizza/ sushi`" or "`good/ bad`", the generator is able to perfectly reconstruct all training sentences. In this case, the generator may learn a degenerated distribution $p_G(\boldsymbol{x}|\boldsymbol{z})$ or $p_G(\boldsymbol{x}|y)$ instead of $p_G(\boldsymbol{x}|\boldsymbol{z}, y)$, depending on which source is dominant in the dataset.

Proposed Method: CARA

To solve this issue, we propose **C***omplementary* **A***uxiliary classifier* **R***egularized* **A***uto-encoder* (CARA). It enhances ARAE with an additional auxiliary classifier for the generated sentences. Specifically, during training, the generator synthesizes a sentence $\hat{\boldsymbol{x}}$ conditioned on a latent code $\boldsymbol{z}$ and a sampled conditional label y, which is then fed to a classifier. The generator is optimized to minimize the classification loss of the classifier corresponding to the conditioned label. Let C^{MI} denote the classifier in the output space. The formulation can be written as:

$$\max_{E,G,C^{\text{MI}}} \mathcal{L}_{\text{MI}} = \mathbb{E}_{\boldsymbol{z}\sim q_E(\boldsymbol{z}|\boldsymbol{x}), y\sim p(y), \boldsymbol{x}\sim p_G(\boldsymbol{z},y)} \log p_{C^{\text{MI}}}(y|\boldsymbol{x}) \tag{8}$$

The full model architecture of CARA is illustrated in Figure 2. Since the output of the generator is discrete, this complicates loss back-propagation from the classifier in the output space. We therefore follow (Hu et al. 2017), adopting the Gumbel Softmax distribution for continuous approximation of discrete output. Note that the idea of introducing the additional auxiliary classifier has been studied in TAC-GAN (Gong et al. 2019). The motivations are different: the

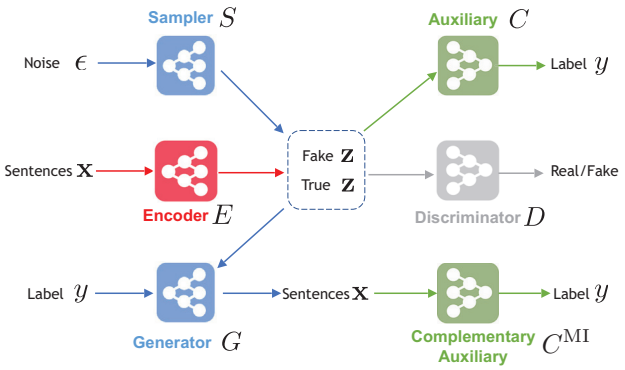

Figure 2: Illustration of the proposed CARA method. A complementary auxiliary classifier C^{MI} is proposed to guide the generated sentences to contain the label information. In the training stage, E learns to maximize the classifier loss of C for disentanglement, and G learns to minimize the classifier loss of C^{MI} for conditional generation. In the evaluation stage, only the networks in blue (*i.e.,* the neural sampler S and generator G) are used for label-conditional text generation. Further, we consider BERT as the encoder E, and add C^{MI} with an annealing schedule.

complementary classifier in CARA aims to encourage label-related information, while twin classifier in TAC-GAN aims to match the joint distributions.

Connection to Mutual Information Note that the generator G and the proposed complementary auxiliary classifier C^{MI} constitute a path for the reconstruction of y, as illustrated in the third row of Figure 2:

$$y \xrightarrow{G} \boldsymbol{x} \xrightarrow{C^{\text{MI}}} y$$

By marginalizing the dependence on $\boldsymbol{z}$ in (8), we have the reconstruction loss for y, written in its log likelihood form as:

$$\mathcal{F}_E = \mathbb{E}_{p\sim p(y),\boldsymbol{x}\sim p(\boldsymbol{x}|y)}[\log p_{C^{\text{MI}}}(y|\boldsymbol{x})]. \tag{9}$$

Following (Li et al. 2017), we show that this cycle-consistency term for y leads to maximizing the mutual information between the generated $\boldsymbol{x}$ and its label y.

Corollary 1 *For random variables $\boldsymbol{x}$ and y, the mutual information between $\boldsymbol{x}$ and y can be written as*

$$I(\boldsymbol{x}, y) \geq H(y) + \mathcal{F}_E \tag{10}$$

The proof is provided in the Appendix. Since the label distribution $p(y)$ is known, $H(y)$ is fixed, and $\mathcal{F}_E$ becomes a lower bound for $I(\boldsymbol{x}, y)$. In practice, CARA maximizes $\mathcal{F}_E$ in (9), which effectively maximizes the mutual information, enforcing the generated $\boldsymbol{x}$ to contain the label information.

The full objective for CARA is:

$$\min_{E,G} \mathcal{L}_{\text{CARA}} = \mathcal{L}_{\text{ARAE}} + \lambda \mathcal{L}_{\text{MI}} \tag{11}$$

where λ is the weighting hyper-parameter for $\mathcal{L}_{\text{MI}}$.

Note that the proposed CARA model has two classifiers, C and C^{MI}, to predict labels. Classifier C operates in the latent space, and the encoder is trained to maximize the classification loss so that disentangled features can be learned. The other classifier C^{MI} is in the observation space, and the generator is trained to minimize the classification loss so that the generated sentences can be controlled. As such, the two auxiliary classifiers are complementary: C^{MI} helps reduce the non-identifiability issue that C causes, and C disentangles $\boldsymbol{z}$ and y so that C^{MI} can effectively constitute a cycle-consistency term to maximize the mutual information.

Trade-off Between Generation Quality and Conditional Accuracy

Though CARA can generate more controllable sentences, due to the proposed auxiliary classifier C^{MI}, it introduces an additional issue on how to tune λ to balance sentence quality (*e.g.,* coherence and faithful linguistic meaning) and label-conditional accuracy (*e.g.,* reflecting the category it belongs), as discussed by (Pang and Gimpel 2018; Li et al. 2019). When λ is small, the label-agnostic information may dominate, as $p_G(\boldsymbol{x}|\boldsymbol{z}, y)$ tends to degenerate to $p_G(\boldsymbol{x}|\boldsymbol{z})$. Example scenarios include text style transfer where only few sentiment-connected words (*e.g.,* "like", "hate", "good" and "bad") in a sentence can indicate sentiment, or topic transfer where the number of topical words (*e.g.,* "philosophy", "finance" and "sports") in a sentence (*e.g.,* "I just watched a video about how to play tennis this morning.") can be few. The result is that the model can perfectly resemble the training sentences, but fail to change the conditional category when given different conditional labels. When λ is large, the label information can dominate, and $p_G(\boldsymbol{x}|\boldsymbol{z}, y)$ tends to degenerate to $p_G(\boldsymbol{x}|y)$. The model learns to output label-related sentences but sacrifices sentence quality (*e.g.,* "I like like like" and "It is bad, bad, bad"). We propose two practical techniques to alleviate this issue.

Annealing Training Schedule We first consider an annealing training schedule on λ, to gradually incorporate $\mathcal{L}_{\text{MI}}$ into training without greatly changing the learning from $\mathcal{L}_{\text{rec}}$. Formally, λ has the form:

$$\lambda_t = \begin{cases} f(\frac{t}{T}), & \text{if } \frac{t}{T} \leq R \\ \lambda_{max}, & \text{otherwise} \end{cases} \tag{12}$$

where t is the iteration step, T is the total number of iterations, and f is a monotonically increasing function. There are two hyper-parameters:

- R: the proportion of total iterations used for increasing λ to a plateau.
- λ_{max}: the maximum value λ can reach.

f monotonically increases λ from 0 to λ_{max} using R portion of T iterations.

BERT as a Strong Encoder We consider to improve the encoder for stronger sentence representation $\boldsymbol{z}$. Recently, models that are pre-trained with large-scale text data have been shown to provide superior generalization capability when fine-tuned for various language-understanding

tasks (Devlin et al. 2018; Radford et al. 2018; 2019). It has been shown in (Subramanian et al. 2018) that general purpose encoders help text generation. In this paper, we consider BERT (Devlin et al. 2018) as our encoder to provide more generalizable sentence representations, so as to balance model learning from different losses and make training more stable.

Related Work

Difference with Attribute Transfer

This paper focuses on label-conditional text generation, where $\boldsymbol{z}$ is drawn from a latent distribution, allowing generation of diverse sentences. This task is different from attribute transfer, where $\boldsymbol{z}$ is extracted from a given sentence and fixed during sentence generation. Therefore, label-conditional text generation requires learning a smooth latent space such that sampling from this space leads to faithful linguistic sentences. Many methods for attribute transfer treat the problem as sequence-to-sequence translation, and source sentences must be provided in order to transfer labels in target sentences (Li et al. 2018; Xu et al. 2018; Zhang, Ding, and Soricut 2018). Such methods are not able to conduct label-conditional generation due to the lack of a smooth space to draw samples from. In contrast, the proposed CARA can be used for both label-conditional generation and style transfer tasks.

Differences from Prior Work

The most related work with ours is Ctrl-Gen (Hu et al. 2017), ARAE (Zhao et al. 2017) and NN-Outlines (Subramanian et al. 2018). In Ctrl-Gen, a label classifier is also incorporated in the observation space. However, our model learns a smoother latent space, via adversarial learning, and a more effective disentanglement constraint is enforced by the auxiliary classifier in the latent space. It is noted that ARAE adopts separate generators for each conditional label, and train them only with samples within the corresponding category. This aims to implicitly alleviate the *non-identifiability* issue, but sacrifices performance as each generator is trained with less samples. It is also inherently impractical when the number of labels increases. On the contrary, CARA does not suffer from either performance degrading due to less samples or excessive parameters. NN-Outlines used a pre-trained general purpose sentence encoder for providing black-box high-level "outlines", and an generative adversarial network in the latent space to match the distribution of the latent representations induced by the encoder. Compared to CARA, NN-Outlines does not have any disentanglement constraints, or additional label control supervision signals.

Experiments

Code and experiment setup is available at Github [1].

Experimental Setting

Datasets Since label-conditional text generation is less comprehensively studied, we consider a suite of four datasets to study this problem, as summarized in Table 1.

Dataset	Attribute	Train	Valid	Test
Personality Captioning	Happy	864	30	39
	Angry	868	26	42
	Malicious	872	14	58
Style Captioning	Humorous	6000	300	300
	Romantic	6000	300	300
	Factual	6000	300	300
Yahoo Questions	Science & Math	126K	14K	6000
	Entertainment & Music	126K	14K	6000
	Politics & Government	126K	14K	6000
Yelp	Positive	270K	2000	500
	Negative	180K	2000	500

Table 1: Dataset statistics

- **Personality captioning** (Shuster et al. 2019) was proposed for engaging image captioning via personality. Each image contains captions with one designated personality. We choose captions from 3 distinctive personalities: `happy`, `angry` and `malicious`.
- **Style captioning** We adapt the style-based image captioning dataset in (Gan et al. 2017) to construct the style-based caption generation task. Three different styles are considered: `humorous`, `romantic` and `factual`.
- **Topic-based question generation** We choose three categories from the Yahoo dataset (Zhang, Zhao, and LeCun 2015): `Society & Culture`, `Business & Finance`, and `Family & Relationships`. We follow (Zhao et al. 2017) to only generate questions.
- **Sentiment manipulation** We use the Yelp dataset with binary sentiment labels. We follow the setup of (Shen et al. 2017) for data splitting.

Evaluation We consider three metrics: (1) *BLEU* for sentence quality, (2) *Accuracy* for conditional generation capability. The accuracy is assessed by an oracle classifier to correctly predict the attributes that generated sentences are conditioned on. (3) *G-score* is reported as the geometric mean of Accuracy and BLEU (Xu et al. 2018). This is the most important metric, as it evaluates the overall performance.

We consider two settings: (*i*) *Conditional generation*. $\boldsymbol{z}$ is generated by the neural sampler S and y is uniformly sampled, $(\boldsymbol{z}, y)$ is used for generation. BLEU of each generated sentence is computed by comparing with all sentences in the test set, as there are no source sentences. We further report Self-BLEU (Zhu et al. 2018) to evaluate the diversity of generated sentences. (*ii*) *Attribute transfer*. $\boldsymbol{z}$ of a sentence is extracted by encoder E. It is combined with a different label for transfer generation. In this setting, we follow common practice to incorporate the generator an attention mechanism to attend encoded features of source sentences. BLEU of each transferred sentence is computed by comparing with its source sentence.

Baselines We compare with three baselines: (1) *Ctrl-Gen* (Hu et al. 2017); (2) *ARAE* (Zhao et al. 2017), and a version without using separate generators (ARAE-); and (3) *NN-Outlines* (Subramanian et al. 2018) proposes the use

[1] https://github.com/s1155026040/CARA

Model	Conditional Generation				Attribute Transfer		
	ACC↑	BLEU↑	G-score↑	Self-BLEU↓	ACC↑	BLEU↑	G-score↑
Ctrl-Gen	65.49	11.14	27.01	96.94	67.61	22.87	39.32
ARAE	66.66	2.64	13.27	99.98	69.87	0.00	0.00
ARAE-	**71.83**	14.31	32.06	98.81	88.03	20.31	42.28
NN-Outlines	60.12	5.35	17.93	93.57	-	-	-
CARA	70.42	15.90	33.46	95.08	**91.55**	21.61	**44.48**
CARA_{A}	67.61	17.56	**34.46**	94.46	84.51	19.55	40.65
CARA_{AB}	61.13	**18.85**	33.95	**88.95**	66.20	**29.67**	44.32

Table 2: Personality captioning results.

Model	Conditional Generation				Attribute Transfer		
	ACC↑	BLEU↑	G-score↑	Self-BLEU↓	ACC↑	BLEU↑	G-score↑
Ctrl-Gen	41.53	28.53	34.42	76.22	42.60	16.73	26.70
ARAE	43.90	58.53	50.69	99.78	47.02	2.58	11.01
ARAE-	32.80	**61.94**	45.07	89.63	36.67	0.66	4.92
NN-Outlines	37.68	17.27	25.51	87.32	-	-	-
CARA	41.98	52.45	46.92	83.76	42.32	1.08	6.76
CARA_{A}	44.40	57.49	50.52	86.62	**45.47**	1.19	7.36
CARA_{AB}	**47.80**	55.38	**51.45**	**75.18**	40.20	**22.60**	**30.14**

Table 3: Style captioning results.

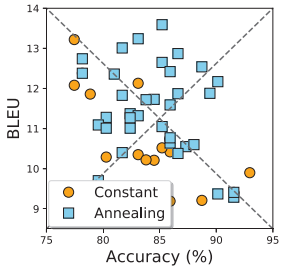

Figure 3: Schedule comparison.

of a general purpose encoder for text generation, and we implement it using BERT.

We consider three variants of CARA:

- *CARA*: the basic model;
- $CARA_A$: CARA with annealing training scheme;
- $CARA_{AB}$: CARA with annealing and BERT encoder.

We provide training details in the Appendix.

Results & Analysis

Results of the four datasets are shown in Tables 2, 3, 4 and 5, respectively. We aim to answer the following questions.

Effectiveness of C^{MI} schedule Figure 3 compares the constant and annealing schedules for λ. The two schedules with various configurations are performed, and each dot represent the result for each configuration. The constant schedule has difficulties balancing BLEU and Accuracy, while the annealing schedule can help balance the two metrics, as demonstrated by the a large number of dots positioned closer to the diagonal line.

Effectiveness of BERT The incorporation of BERT yields consistent improvement for both conditional generation and attribute transfer. Note that the intra-class diversity of Personality and Style captioning datasets are quite high. Most models without BERT show low diversity on these two challenging datasets By incorporating BERT as a strong encoder, the collapse is not observed in CARA_{AB}, demonstrating that a strong universal encoder provides more meaningful latent codes for the generator to rely on in decoding.

Attribute Transfer As CARA can be applied to the attribute transfer task, we further compare it with models specifically designed for this task. On the Yelp dataset, we compare with additional previous methods, including Cross-Align (Shen et al. 2017), MultiDecoder (Fu et al. 2018), DeleteAndRetrieve (Li et al. 2018), StyleTransformer (Dai et al. 2019), Back-Translation (Prabhumoye et al. 2018), and iVAE_{MI} (Le Fang 2019). Due to limited space, we provide a comparison with more methods in the Appendix.

Conditional Text Generation The proposed CARA_{AB} consistently achieves the best G-score for all datasets, except Personality captioning. It indicates that CARA is a strong competitor for controllable sentence generation. Meanwhile, the lower Self-BLEU scores of CARA shows that the generated sentence samples of CARA are diverse. On two large-scale datasets, Yahoo and Yelp, CARA provides consistent improvement. ARAE achieves similar overall performance as ARAE-, while CARA improves ARAE-. This verifies our

Model	Conditional Generation				Attribute Transfer		
	ACC↑	BLEU↑	G-score↑	Self-BLEU↓	ACC↑	BLEU↑	G-score↑
Ctrl-Gen	60.47	21.38	35.96	59.21	60.12	64.21	62.13
ARAE	50.05	22.94	33.88	84.24	37.73	17.06	25.37
ARAE-	46.79	23.74	33.33	51.93	49.38	60.23	54.54
NN-Outlines	38.81	19.90	27.79	51.53	-	-	-
CARA	63.67	25.60	40.31	52.45	68.26	34.10	48.25
CARA_{A}	67.55	25.88	41.81	**50.33**	69.17	68.96	69.06
CARA_{AB}	**69.67**	**30.62**	**46.19**	56.11	**75.61**	**69.97**	**72.74**

Table 4: Topic-based question generation results on Yahoo dataset.

Model	Conditional Generation				Attribute Transfer		
	ACC↑	BLEU↑	G-score↑	Self-BLEU↓	ACC↑	BLEU↑	G-score↑
Cross-Align	-	-	-	-	79.5	12.4	31.40
MultiDecoder	-	-	-	-	47.6	13.25	25.11
DeleteAndRetrieve	-	-	-	-	88.7	14.75	36.17
StyleTransformer	-	-	-	-	93.6	17.1	40.01
iVAE_{MI}	-	-	-	-	92.0	36.7	58.11
Ctrl-Gen	87.81	28.83	50.31	51.23	87.57	37.75	57.50
ARAE	**96.72**	20.18	44.18	35.81	85.43	22.97	44.30
ARAE-	60.07	27.63	40.74	**33.04**	61.29	28.62	41.88
NN-Outlines	55.31	19.84	33.12	54.57	-	-	-
CARA	91.45	30.12	52.48	52.08	91.69	43.64	63.26
CARA_{A}	91.49	32.46	54.50	49.74	92.42	46.28	65.40
CARA_{AB}	94.90	**37.23**	**59.44**	44.51	**95.45**	**53.25**	**71.29**

Table 5: Sentiment transfer results on Yelp dataset.

	`Business & Finance`
ARAE	Where was the most emst adie place you apply as? Do you need a flat right now? *What is the law was a parent's length at their child's pepmed?*
CARA_{AB}	What is the conversion of irish money to american money? Knowing what the effect of ads are on people , why do we allow ads showing beautiful people. Where is the best place to look for a grant for a nonprofit soccer club?
	`Family & Relationships`
ARAE	What is the meaning of compliment? *What would you do if you just got out of heavyyme and have no job no where to* When ur a level 2 do u get 20some the same day?
CARA_{AB}	Why does a cheating man act like he is not cheating if he isn't interested in his Why do people think that children involved in a gay/lesiban adoption will be rebound what does it mean when someone tells you they always think about you?

Table 6: Qualitative results of conditional generation in topic-based question generation. Sentences in *Italic* form indicate their demonstrated categories do not match with their conditioned labels.

assumption that ARAE's solution for alleviating the *non-identifility* is less effective than CARA, and sacrifices sentence quality due to less training samples for each generator.

Generated Samples In Table 6, we show the samples for conditional generation. The sentences generated by ARAE may lose its label information, while CARA shows strong dependence on the labels. Please see more results on label-conditional generation and attribute transfer in Appendix.

Conclusions

We have described CARA for label-conditional text generation. CARA utilizes one auxiliary classifier for disentangled feature learning in the latent space, and the other auxiliary classifier for more accurate label-conditioning on the generated sentences. An annealing training schedule and adopting BERT as a strong encoder further improve CARA's performance. Experiments on four datasets consistently show that CARA achieves both improved natural sentences generation and accurate label transfer.

References

Alemi, A. A.; Poole, B.; Fischer, I.; Dillon, J. V.; Saurous, R. A.; and Murphy, K. 2017. Fixing a broken ELBO. *arXiv preprint arXiv:1711.00464*.

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Jozefowicz, R.; and Bengio, S. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Chen, X.; Kingma, D. P.; Salimans, T.; Duan, Y.; Dhariwal, P.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*.

Dai, N.; Liang, J.; Qiu, X.; and Huang, X. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. *arXiv preprint arXiv:1905.05621*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fu, Z.; Tan, X.; Peng, N.; Zhao, D.; and Yan, R. 2018. Style transfer in text: Exploration and evaluation. In *AAAI*.

Fu, H.; Li, C.; Liu, X.; Gao, J.; Celikyilmaz, A.; and Carin, L. 2019. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *NAACL*.

Gan, C.; Gan, Z.; He, X.; Gao, J.; and Deng, L. 2017. Stylenet: Generating attractive visual captions with styles. In *CVPR*.

Gong, M.; Xu, Y.; Li, C.; Zhang, K.; and Batmanghelich, K. 2019. Twin auxilary classifiers GAN. In *Advances in Neural Information Processing Systems*.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*.

Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017. Toward controlled generation of text. *ICML*.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational Bayes. *ICLR*.

Le Fang, Chunyuan Li, J. G. W. D. C. C. 2019. Implicit deep latent variable models for text generation. *EMNLP*.

Li, C.; Liu, H.; Chen, C.; Pu, Y.; Chen, L.; Henao, R.; and Carin, L. 2017. Alice: Towards understanding adversarial learning for joint distribution matching. In *NIPS*.

Li, J.; Jia, R.; He, H.; and Liang, P. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*.

Li, D.; Zhang, Y.; Gan, Z.; Cheng, Y.; Brockett, C.; Sun, M.-T.; and Dolan, B. 2019. Domain adaptive text style transfer. In *EMNLP*.

Mikolov, T.; Karafiát, M.; Burget, L.; Černockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Pang, Y., and Gimpel, K. 2018. Learning criteria and evaluation metrics for textual transfer between non-parallel corpora. *arXiv preprint arXiv:1810.11878*.

Prabhumoye, S.; Tsvetkov, Y.; Salakhutdinov, R.; and Black, A. W. 2018. Style transfer through back-translation. *arXiv preprint arXiv:1804.09000*.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8).

Shen, T.; Lei, T.; Barzilay, R.; and Jaakkola, T. 2017. Style transfer from non-parallel text by cross-alignment. In *NIPS*.

Shuster, K.; Humeau, S.; Hu, H.; Bordes, A.; and Weston, J. 2019. Engaging image captioning via personality. In *CVPR*.

Subramanian, S.; Mudumba, S. R.; Sordoni, A.; Trischler, A.; Courville, A. C.; and Pal, C. 2018. Towards text generation with adversarially learned neural outlines. In *Advances in Neural Information Processing Systems*, 7551–7563.

Xu, J.; Sun, X.; Zeng, Q.; Ren, X.; Zhang, X.; Wang, H.; and Li, W. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. *arXiv preprint arXiv:1805.05181*.

Yang, Z.; Hu, Z.; Salakhutdinov, R.; and Berg-Kirkpatrick, T. 2017. Improved variational autoencoders for text modeling using dilated convolutions. *ICML*.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhang, Y.; Ding, N.; and Soricut, R. 2018. Shaped: Shared-private encoder-decoder for text style adaptation. *arXiv preprint arXiv:1804.04093*.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Zhao, J.; Kim, Y.; Zhang, K.; Rush, A. M.; and LeCun, Y. 2017. Adversarially regularized autoencoders. *arXiv preprint arXiv:1706.04223*.

Zhao, T.; Zhao, R.; and Eskenazi, M. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *ACL*.

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*.