Toward Controlled Generation of Text

Zhiting Hu [1 2] **Zichao Yang** [1] **Xiaodan Liang** [1 2] **Ruslan Salakhutdinov** [1] **Eric P. Xing** [1 2]

Abstract

Generic generation and manipulation of text is challenging and has limited success compared to recent deep generative modeling in visual domain. This paper aims at generating plausible text sentences, whose attributes are controlled by learning disentangled latent representations with designated semantics. We propose a new neural generative model which combines variational auto-encoders (VAEs) and holistic attribute discriminators for effective imposition of semantic structures. The model can alternatively be seen as enhancing VAEs with the wake-sleep algorithm for leveraging fake samples as extra training data. With differentiable approximation to discrete text samples, explicit constraints on independent attribute controls, and efficient collaborative learning of generator and discriminators, our model learns interpretable representations from even only word annotations, and produces short sentences with desired attributes of sentiment and tenses. Quantitative experiments using trained classifiers as evaluators validate the accuracy of sentence and attribute generation.

1. Introduction

There is a surge of research interest in deep generative models (Hu et al., 2017), such as Variational Autoencoders (VAEs) (Kingma & Welling, 2013), Generative Adversarial Nets (GANs) (Goodfellow et al., 2014), and autoregressive models (van den Oord et al., 2016). Despite their impressive advances in visual domain, such as image generation (Radford et al., 2015), learning interpretable image representations (Chen et al., 2016), and image editing (Zhu et al., 2016), applications to natural language generation have been relatively less studied. Even generating realistic sentences is challenging as the generative models are required to capture complex semantic structures underlying sentences. Previous work have been mostly limited to task-specific applications in supervised settings, including machine translation (Bahdanau et al., 2014) and image captioning (Vinyals et al., 2015). However, autoencoder frameworks (Sutskever et al., 2014) and recurrent neural network language models (Mikolov et al., 2010) do not apply to generic text generation from arbitrary hidden representations due to the unsmoothness of effective hidden codes (Bowman et al., 2015). Very few recent attempts of using VAEs (Bowman et al., 2015; Tang et al., 2016) and GANs (Yu et al., 2017; Zhang et al., 2016) have been made to investigate generic text generation, while their generated text is largely randomized and uncontrollable.

In this paper we tackle the problem of *controlled* generation of text. That is, we focus on generating realistic sentences, whose attributes can be controlled by learning disentangled latent representations. To enable the manipulation of generated sentences, a few challenges need to be addressed.

A first challenge comes from the discrete nature of text samples. The resulting non-differentiability hinders the use of global discriminators that assess generated samples and back-propagate gradients to guide the optimization of generators in a holistic manner, as shown to be highly effective in continuous image generation and representation modeling (Chen et al., 2016; Larsen et al., 2016; Dosovitskiy & Brox, 2016). A number of recent approaches attempt to address the non-differentiability through policy learning (Yu et al., 2017) which tends to suffer from high variance during training, or continuous approximations (Zhang et al., 2016; Kusner & Hernndez-Lobato, 2016) where only preliminary qualitative results are presented. As an alternative to the discriminator based learning, semi-supervised VAEs (Kingma et al., 2014) minimize element-wise reconstruction error on observed examples and are applicable to discrete visibles. This, however, loses the holistic view of full sentences and can be inferior especially for modeling global abstract attributes (e.g., sentiment).

Another challenge for controllable generation relates to learning disentangled latent representations. Interpretability expects each part of the latent representation to govern and *only* focus on one aspect of the samples. Prior methods (Chen et al., 2016; Odena et al., 2016) on structured representation learning lack explicit enforcement of the in-

[1] Carnegie Mellon University [2] Petuum, Inc.. Correspondence to: Zhiting Hu <zhitingh@cs.cmu.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

dependence property on the full latent representation, and varying individual code may result in unexpected variation of other unspecified attributes besides the desired one.

In this paper, we propose a new text generative model that addresses the above issues, permitting highly disentangled representations with designated semantic structure, and generating sentences with dynamically specified attributes. We base our generator on VAEs in combination with holistic discriminators of attributes for effective imposition of structures on the latent code. End-to-end optimization is enabled with differentiable softmax approximation which anneals smoothly to discrete case and helps fast convergence. The probabilistic encoder of VAE also functions as an additional discriminator to capture variations of implicitly modeled aspects, and guide the generator to avoid entanglement during attribute code manipulation.

Our model can be interpreted as enhancing VAEs with an extended wake-sleep procedure (Hinton et al., 1995), where the sleep phase enables incorporation of generated samples for learning both the generator and discriminators in an alternating manner. The generator and the discriminators effectively provide feedback signals to each other, resulting in an efficient mutual bootstrapping framework. We show a little supervision (e.g., 100s of annotated sentences) is sufficient to learn structured representations.

Quantitative experiments demonstrate the efficacy of our method. We apply our model to generate sentences with controlled sentiment and tenses. Our method improves over previous generative models on the accuracy of generating specified attributes as well as performing classification using generated samples. We show our method learns highly disentangled representations from only word-level labels, and produces plausible short sentences.

2. Related Work

Remarkable progress has been made in deep generative modeling. Hu et al. (2017) provide a unified view of a diverse set of deep generative methods. Variational Autoencoders (VAEs) (Kingma & Welling, 2013) consist of encoder and generator networks which encode a data example to a latent representation and generate samples from the latent space, respectively. The model is trained by maximizing a variational lower bound on the data log-likelihood under the generative model. A KL divergence loss is minimized to match the posterior of the latent code with a prior, which enables every latent code from the prior to decode into a plausible sentence. Without the KL regularization, VAEs degenerate to autoencoders and become inapplicable for the generic generation. The vanilla VAEs are incompatible with discrete latents as they hinder differentiable parameterization for learning the encoder. Wake-sleep algorithm (Hinton et al., 1995) introduced for learning deep directed graphical models shares similarity with VAEs by also combining an inference network with the generator. The wake phase updates the generator with samples generated from the inference network on training data, while the sleep phase updates the inference network based on samples from the generator. Our method combines VAEs with an extended wake-sleep in which the sleep procedure updates both the generator and inference network (discriminators), enabling collaborative semi-supervised learning.

Besides reconstruction in raw data space, discriminator-based metric provides a different way for generator learning, i.e., the discriminator assesses generated samples and feedbacks learning signals. For instance, GANs (Goodfellow et al., 2014) use a discriminator to feedback the probability of a sample being recognized as a real example. Larsen et al. (2016) combine VAEs with GANs for enhanced image generation. Dosovitskiy & Brox (2016); Taigman et al. (2017) use discriminators to measure high-level perceptual similarity. Applying discriminators to text generation is hard due to the non-differentiability of discrete samples (Yu et al., 2017; Zhang et al., 2016; Kusner & Hernndez-Lobato, 2016). Bowman et al. (2015); Tang et al. (2016); Yang et al. (2017) instead use VAEs without discriminators. All these text generation methods do not learn disentangled latent representations, resulting in randomized and uncontrollable samples. In contrast, disentangled generation in visual domain has made impressive progress. E.g., InfoGAN (Chen et al., 2016), which resembles the extended sleep procedure of our joint VAE/wake-sleep algorithm, disentangles latent representation in an unsupervised manner. The semantic of each dimension is observed after training rather than designated by users in a controlled way. Siddharth et al. (2017); Kingma et al. (2014) base on VAEs and obtain disentangled image representations with semi-supervised learning. Zhou & Neubig (2017) extend semi-supervised VAEs for text transduction. In contrast, our model combines VAEs with discriminators which provide a better, holistic metric compared to element-wise reconstruction. Moreover, most of these approaches have only focused on the disentanglement of the structured part of latent representations, while ignoring potential dependence of the structured code with attributes not explicitly encoded. We address this by introducing an independency constraint, and show its effectiveness for improved interpretability.

3. Controlled Generation of Text

Our model aims to generate plausible sentences conditioned on representation vectors which are endowed with designated semantic structures. For instance, to control sentence sentiment, our model allocates one dimension of the latent representation to encode "positive" and "negative" semantics, and generates samples with desired sentiment by simply specifying a particular code. Benefiting from the disentangled structure, each such code is able to capture a salient attribute and is independent with other features. Our deep text generative model possesses several merits compared to prior work, as it 1) facilitates effective imposition of latent code semantics by enabling global discriminators to guide the discrete text generator learning; 2) improves model interpretability by explicitly enforcing the constraints on independent attribute controls; 3) permits efficient semi-supervised learning and bootstrapping by synthesizing variational auto-encoders with a tailored wake-sleep approach. We first present the overview of our framework (§3.1), then describe the model in detail (§3.2).

3.1. Model Overview

We build our framework starting from variational auto-encoders (§2) which have been used for text generation (Bowman et al., 2015), where sentence $\hat{\boldsymbol{x}}$ is generated conditioned on latent code $\boldsymbol{z}$. The vanilla VAE employs an unstructured vector $\boldsymbol{z}$ in which the dimensions are entangled. To model and control the attributes of interest in an interpretable way, we augment the unstructured variables $\boldsymbol{z}$ with a set of structured variables $\boldsymbol{c}$ each of which targets a salient and independent semantic feature of sentences.

We want our sentence generator to condition on the combined vector $(\boldsymbol{z}, \boldsymbol{c})$, and generate samples that fulfill the attributes as specified in the structured code $\boldsymbol{c}$. Conditional generation in the context of VAEs (e.g., semi-supervised VAEs (Kingma et al., 2014)) is often learned by reconstructing observed examples given their feature code. However, as demonstrated in visual domain, compared to computing element-wise distances in the data space, computing distances in the feature space allows invariance to distracting transformations and provides a better, holistic metric. Thus, for each attribute code in $\boldsymbol{c}$, we set up an individual discriminator to measure how well the generated samples match the desired attributes, and drive the generator to produce improved results. The difficulty of applying discriminators in our context is that text samples are discrete and non-differentiable, which breaks down gradient propagation from the discriminators to the generator. We use a continuous approximation based on softmax with a decreasing temperature, which anneals to the discrete case as training proceeds. This simple yet effective approach enjoys low variance and fast convergence.

Intuitively, having an interpretable representation would imply that each structured code in $\boldsymbol{c}$ can independently control its target feature, without entangling with other attributes, especially those not explicitly modeled. We encourage the independency by enforcing those irrelevant attributes to be completely captured in the unstructured code $\boldsymbol{z}$ and thus be separated from $\boldsymbol{c}$ that we will manipulate. To this end, we reuse the VAE encoder as an additional discriminator for recognizing the attributes modeled in $\boldsymbol{z}$, and train the generator so that these unstructured attributes can be recovered from the generated samples. As a result, varying different attribute codes will keep the unstructured attributes invariant as long as $\boldsymbol{z}$ is unchanged.

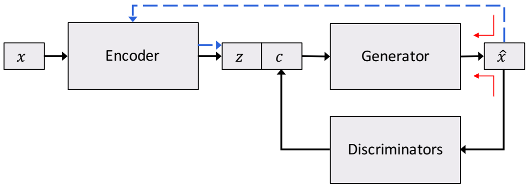

Figure 1. The generative model, where $\boldsymbol{z}$ is unstructured latent code and $\boldsymbol{c}$ is structured code targeting sentence attributes to control. Blue dashed arrows denote the proposed independency constraint (section 3.2 for details), and red arrows denote gradient propagation enabled by the differentiable approximation.

Figure 1 shows the overall model structure. Our complete model incorporates VAEs and attribute discriminators, in which the VAE component trains the generator to reconstruct real sentences for generating plausible text, while the discriminators enforce the generator to produce attributes coherent with the conditioned code. The attribute discriminators are learned to fit labeled examples to entail designated semantics, as well as trained to explain samples from the generator. That is, the generator and the discriminators form a pair of collaborative learners and provide feedback signals to each other. The collaborative optimization resembles wake-sleep algorithm. We show the combined VAE/wake-sleep learning enables a highly efficient semi-supervised framework, which requires only a little supervision to obtain interpretable representation and generation.

3.2. Model Structure

We now describe our model in detail, by presenting the learning of generator and discriminators, respectively.

Generator Learning

The generator G is an LSTM-RNN for generating token sequence $\hat{\boldsymbol{x}} = \{\hat{x}_1, \ldots, \hat{x}_T\}$ conditioned on the latent code $(\boldsymbol{z}, \boldsymbol{c})$, which depicts a generative distribution:

$$\begin{aligned} \hat{\boldsymbol{x}} \sim G(\boldsymbol{z}, \boldsymbol{c}) &= p_G(\hat{\boldsymbol{x}}|\boldsymbol{z}, \boldsymbol{c}) \\ &= \prod\nolimits_t p(\hat{x}_t|\hat{\boldsymbol{x}}^{<t}, \boldsymbol{z}, \boldsymbol{c}), \end{aligned} \tag{1}$$

where $\hat{\boldsymbol{x}}^{<t}$ indicates the tokens preceding $\hat{x}_t$. The generation thus involves a sequence of discrete decision making which samples a token from a multinomial distribution parametrized using softmax function at each time step t:

$$\hat{x}_t \sim \text{softmax}(\boldsymbol{o}_t/\tau), \tag{2}$$

where $\boldsymbol{o}_t$ is the logit vector as the inputs to the softmax function, and $\tau > 0$ is the temperature normally set to 1.

The unstructured part $\boldsymbol{z}$ of the representation is modeled as continuous variables with standard Gaussian prior $p(\boldsymbol{z})$, while the structured code $\boldsymbol{c}$ can contain both continuous and discrete variables to encode different attributes (e.g., sentiment categories, formality) with appropriate prior $p(\boldsymbol{c})$. Given observation $\boldsymbol{x}$, the base VAE includes a conditional probabilistic encoder E to infer the latents $\boldsymbol{z}$:

$$\boldsymbol{z} \sim E(\boldsymbol{x}) = q_E(\boldsymbol{z}|\boldsymbol{x}). \tag{3}$$

Let $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_E$ denote the parameters of the generator G and the encoder E, respectively. The VAE is then optimized to minimize the reconstruction error of observed real sentences, and at the same time regularize the encoder to be close to the prior $p(\boldsymbol{z})$:

$$\begin{aligned}\mathcal{L}_{\text{VAE}}(\boldsymbol{\theta}_G, \boldsymbol{\theta}_E; \boldsymbol{x}) = & -\text{KL}(q_E(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z})) \\ & + \mathbb{E}_{q_E(\boldsymbol{z}|\boldsymbol{x}) q_D(\boldsymbol{c}|\boldsymbol{x})} \left[\log p_G(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})\right],\end{aligned} \tag{4}$$

where $\text{KL}(\cdot\|\cdot)$ is the KL-divergence; and $q_D(\boldsymbol{c}|\boldsymbol{x})$ is the conditional distribution defined by the discriminator D for each structured variable in $\boldsymbol{c}$:

$$D(\boldsymbol{x}) = q_D(\boldsymbol{c}|\boldsymbol{x}). \tag{5}$$

Here, for notational simplicity, we assume only one structured variable and thus one discriminator, though our model specification can straightforwardly be applied to many attributes. The distribution over $(\boldsymbol{z}, \boldsymbol{c})$ factors into q_E and q_D as we are learning disentangled representations. Note that here the discriminator D and code $\boldsymbol{c}$ are not learned with the VAE loss, but instead optimized with the objectives described shortly. Besides the reconstruction loss which drives the generator to produce realistic sentences, the discriminator provides extra learning signals which enforce the generator to produce coherent attribute that matches the structured code in $\boldsymbol{c}$. However, as it is impossible to propagate gradients from the discriminator through the discrete samples, we resort to a deterministic continuous approximation. The approximation replaces the sampled token $\hat{x}_t$ (represented as a one-hot vector) at each step with the probability vector in Eq.(2) which is differentiable w.r.t the generator's parameters. The probability vector is used as the output at the current step and the input to the next step along the sequence of decision making. The resulting "soft" generated sentence, denoted as $\widetilde{G}_\tau(\boldsymbol{z}, \boldsymbol{c})$, is fed into the discriminator[1] to measure the fitness to the target attribute, leading to the following loss for improving G:

$$\mathcal{L}_{\text{Attr},c}(\boldsymbol{\theta}_G) = \mathbb{E}_{p(\boldsymbol{z})p(\boldsymbol{c})} \left[\log q_D(\boldsymbol{c}|\widetilde{G}_\tau(\boldsymbol{z}, \boldsymbol{c}))\right]. \tag{6}$$

The temperature τ (Eq.2) is set to $\tau \to 0$ as training proceeds, yielding increasingly peaked distributions that finally emulate discrete case. The simple deterministic approximation effectively leads to reduced variance and fast convergence during training, which enables efficient learning of the conditional generator. The diversity of generation results is guaranteed since we use the approximation only for attribute modeling and the base sentence generation is learned through VAEs.

With the objective in Eq.(6), each structured attribute of generated sentences is controlled through the corresponding code in $\boldsymbol{c}$ and is independent with other variables in the latent representation. However, it is still possible that other attributes not explicitly modeled may also entangle with the code in $\boldsymbol{c}$, and thus varying a dimension of $\boldsymbol{c}$ can yield unexpected variation of these attributes we are not interested in. To address this, we introduce the independency constraint which separates these attributes with $\boldsymbol{c}$ by enforcing them to be fully captured by the unstructured part $\boldsymbol{z}$. Therefore, besides the attributes explicitly encoded in $\boldsymbol{c}$, we also train the generator so that other non-explicit attributes can be correctly recognized from the generated samples and match the unstructured code $\boldsymbol{z}$. Instead of building a new discriminator, we reuse the variational encoder E which serves precisely to infer the latents $\boldsymbol{z}$ in the base VAE. The loss is in the same form as with Eq.(6) except replacing the discriminator conditional q_D with the encoder conditional q_E:

$$\mathcal{L}_{\text{Attr},z}(\boldsymbol{\theta}_G) = \mathbb{E}_{p(\boldsymbol{z})p(\boldsymbol{c})} \left[\log q_E(\boldsymbol{z}|\widetilde{G}_\tau(\boldsymbol{z}, \boldsymbol{c}))\right]. \tag{7}$$

Note that, as the discriminator in Eq.(6), the encoder now performs inference over generated samples from the prior, as opposed to observed examples as in VAEs.

Combining Eqs.(4)-(7) we obtain the generator objective:

$$\min_{\boldsymbol{\theta}_G} \mathcal{L}_G = \mathcal{L}_{\text{VAE}} + \lambda_c \mathcal{L}_{\text{Attr},c} + \lambda_z \mathcal{L}_{\text{Attr},z}, \tag{8}$$

where λ_c and λ_z are balancing parameters. The variational encoder is trained by minimizing the VAE loss, i.e., $\min_{\boldsymbol{\theta}_E} \mathcal{L}_{\text{VAE}}$.

Discriminator Learning

The discriminator D is trained to accurately infer the sentence attribute and evaluate the error of recovering the desired feature as specified in the latent code. For instance, for categorical attribute, the discriminator can be formulated as a sentence classifier; while for continuous target a probabilistic regressor can be used. The discriminator is learned in a different way compared to the VAE encoder, since the target attributes can be discrete which are not supported in the VAE framework. Moreover, in contrast to the unstructured code $\boldsymbol{z}$ which is learned in an unsupervised manner, the structured variable $\boldsymbol{c}$ uses labeled examples to

[1] The probability vector thus functions to average over the word embedding matrix to obtain a "soft" word embedding at each step.

Algorithm 1 Controlled Generation of Text

```
Input: A large corpus of unlabeled sentences $\mathcal{X} = \{\boldsymbol{x}\}$
       A few sentence attribute labels $\mathcal{X}_L = \{(\boldsymbol{x}_L, \boldsymbol{c}_L)\}$
       Parameters: $\lambda_c, \lambda_z, \lambda_u, \beta$ – balancing parameters
1: Initialize the base VAE by minimizing Eq.(4) on $\mathcal{X}$ with $\boldsymbol{c}$ sampled from prior $p(\boldsymbol{c})$
2: repeat
3:     Train the discriminator $D$ by Eq.(11)
4:     Train the generator $G$ and the encoder $E$ by Eq.(8) and minimizing Eq.(4), respectively.
5: until convergence
Output: Sentence generator $G$ conditioned on disentangled representation $(\boldsymbol{z}, \boldsymbol{c})$
```

entail designated semantics. We derive an efficient semi-supervised learning method for the discriminator.

Formally, let $\boldsymbol{\theta}_D$ denote the parameters of the discriminator. To learn specified semantic meaning, we use a set of labeled examples $\mathcal{X}_L = \{(\boldsymbol{x}_L, \boldsymbol{c}_L)\}$ to train the discriminator D with the following objective:

$$\mathcal{L}_s(\boldsymbol{\theta}_D) = \mathbb{E}_{\mathcal{X}_L}\left[\log q_D(\boldsymbol{c}_L|\boldsymbol{x}_L)\right]. \tag{9}$$

Besides, the conditional generator G is also capable of synthesizing (noisy) sentence-attribute pairs $(\hat{\boldsymbol{x}}, \boldsymbol{c})$ which can be used to augment training data for semi-supervised learning. To alleviate the issue of noisy data and ensure robustness of model optimization, we incorporate a minimum entropy regularization term (Grandvalet et al., 2004; Reed et al., 2014). The resulting objective is thus:

$$\mathcal{L}_u(\boldsymbol{\theta}_D) = \mathbb{E}_{p_G(\hat{\boldsymbol{x}}|\boldsymbol{z},\boldsymbol{c})p(\boldsymbol{z})p(\boldsymbol{c})}\big[\log q_D(\boldsymbol{c}|\hat{\boldsymbol{x}}) + \beta\mathcal{H}(q_D(\boldsymbol{c}'|\hat{\boldsymbol{x}}))\big], \tag{10}$$

where $\mathcal{H}(q_D(\boldsymbol{c}'|\hat{\boldsymbol{x}}))$ is the empirical Shannon entropy of distribution q_D evaluated on the generated sentence $\hat{\boldsymbol{x}}$; and β is the balancing parameter. Intuitively, the minimum entropy regularization encourages the model to have high confidence in predicting labels.

The joint training objective of the discriminator using both labeled examples and synthesized samples is then given as:

$$\min_{\boldsymbol{\theta}_D} \mathcal{L}_D = \mathcal{L}_s + \lambda_u\mathcal{L}_u, \tag{11}$$

where λ_u is the balancing parameter.

Summarization and Discussion

We have derived our model and its learning procedure. The generator is first initialized by training the base VAE on a large corpus of unlabeled sentences, through the objective of minimizing Eq.(4) with the latent code $\boldsymbol{c}$ at this time sampled from the prior distribution $p(\boldsymbol{c})$. The full model is then trained by alternating the optimization of the generator and the discriminator, as summarized in Algorithm 1.

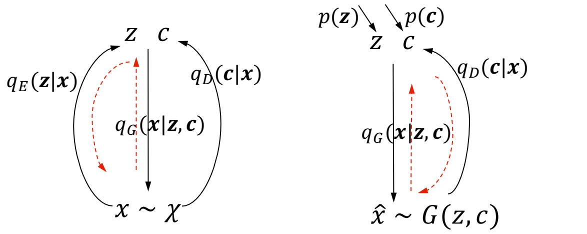
z
c
$q_E(\boldsymbol{z}|\boldsymbol{x})$
$q_D(\boldsymbol{c}|\boldsymbol{x})$
$q_G(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{c})$
$x \sim \mathcal{X}$
$p(\boldsymbol{z})$
$p(\boldsymbol{c})$
z
c
$q_D(\boldsymbol{c}|\boldsymbol{x})$
$q_G(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{c})$
$\hat{x} \sim G(z,c)$

Figure 2. **Left:** The VAE and wake procedure, corresponding to Eq.(4). **Right:** The sleep procedure, corresponding to Eqs.(6)-(7) and (10). Black arrows denote inference and generation; red dashed arrows denote gradient propagation. The two steps in the sleep procedure, i.e., optimizing the discriminator and the generator, respectively, are performed in an alternating manner.

Our model can be viewed as combining the VAE framework with an extended wake-sleep method, as illustrated in Figure 2. Specifically, in Eq.(10), samples are produced by the generator and used as targets for maximum likelihood training of the discriminator. This resembles the sleep phase of wake-sleep. Eqs.(6)-(7) further leverage the generated samples to improve the generator. We can see the above together as an extended sleep procedure based on "dream" samples obtained by ancestral sampling from the generative network. On the other hand, Eq.(4) samples $\boldsymbol{c}$ from the discriminator distribution $q_D(\boldsymbol{c}|\boldsymbol{x})$ on observation $\boldsymbol{x}$, to form a target for training the generator, which corresponds to the wake phase. The effective combination enables discrete latent code, holistic discriminator metrics, and efficient mutual bootstrapping.

Training of the discriminators need supervised data to impose designated semantics. Discriminators for different attributes can be trained independently on separate labeled sets. That is, the model does not require a sentence to be annotated with all attributes, but instead needs only independent labeled data for each individual attribute. Moreover, as the labeled data are used only for learning attribute semantics instead of direct sentence generation, we are allowed to extend the data scope beyond labeled sentences to, e.g., labeled words or phrases. As shown in the experiments (section 4), our method is able to effectively lift the word level knowledge to sentence level and generate convincing sentences. Finally, with the augmented unsupervised training in the sleep phrase, we show a little supervision is sufficient for learning structured representations.

4. Experiments

We apply our model to generate short sentences (length $\leq$ 15) with controlled sentiment and tense. Quantitative experiments using trained classifiers as evaluators show our model gives improved generation accuracy. Disentangled representation is learned with a few labels or only word annotations. We also validate the effect of the proposed independency constraint for interpretable generation.

Datasets

Sentence corpus. We use a large IMDB text corpus (Diao et al., 2014) for training the generative models. This is a collection of 350K movie reviews. We select sentences containing at most 15 words, and replace infrequent words with the token "<unk>". The resulting dataset contains around 1.4M sentences with the vocabulary size of 16K.

Sentiment. To control the sentiment ("positive" or "negative") of generated sentences, we test on the following labeled sentiment data: (1) Stanford Sentiment Treebank-2 (**SST-full**) (Socher et al., 2013) consists of 6920/872/1821 movie review sentences with binary sentiment annotations in the train/dev/test sets, respectively. We use the 2837 training examples with sentence length $\leq$ 15, and evaluate classification accuracy on the original test set. (2) **SST-small.** To study the size of labeled data required in the semi-supervised learning for accurate attribute control, we sample a small subset from SST-full, containing only 250 labeled sentences for training. (3) **Lexicon.** We also investigate the effectiveness of our model in terms of using word-level labels for sentence-level control. The lexicon from (Wilson et al., 2005) contains 2700 words with sentiment labels. We use the lexicon for training by treating the words as sentences, and evaluate on the SST-full test set. (4) **IMDB.** We collect a dataset from the IMDB corpus by randomly selecting positive and negative movie reviews. The dataset has 5K/1K/10K sentences in train/dev/test.

Tense. The second attribute is the tense of the main verb in a sentence. Though no corpus with sentence tense annotations is readily available, our method is able to learn from only labeled words and generate desired sentences. We compile from the TimeBank (timeml.org) dataset and obtain a lexicon of 5250 words and phrases labeled with one of {"past", "present", "future"}. The lexicon mainly consists of verbs in different tenses (e.g., "was", "will be") as well as time expressions (e.g., "in the future").

Parameter Setting

The generator and encoder are set as single-layer LSTM RNNs with input/hidden dimension of 300 and max sample length of 15. Discriminators are set as ConvNets. Detailed configurations are in the supplements. To avoid vanishingly small KL term in the VAE module (Eq.4) (Bowman et al., 2015), we use a KL term weight linearly annealing from 0 to 1 during training. Balancing parameters are set to $\lambda_c = \lambda_z = \lambda_u = 0.1$, and β is selected on the dev sets. At test time sentences are generated with Eq.(1).

4.1. Accuracy of Generated Attributes

We quantitatively measure sentence attribute control by evaluating the accuracy of generating designated sentiment, and the effect of using samples for training classifiers. We compare with semi-supervised VAE (S-VAE) (Kingma et al., 2014), one of the few existing deep models capable of conditional text generation. S-VAE learns to reconstruct observed sentences given attribute code, and no discriminators are used. See §2 and 3.1 for more discussions.

Model	Dataset		
	SST-full	SST-small	Lexicon
S-VAE	0.822	0.679	0.660
Ours	**0.851**	**0.707**	**0.701**

Table 1. Sentiment accuracy of generated sentences. S-VAE (Kingma et al., 2014) and our model are trained on the three sentiment datasets and generate 30K sentences, respectively.

We use a state-of-the-art sentiment classifier (Hu et al., 2016a) which achieves 90% accuracy on the SST test set, to automatically evaluate the sentiment generation accuracy. Specifically, we generate sentences given sentiment code $\boldsymbol{c}$, and use the pre-trained sentiment classifier to assign sentiment labels to the generated sentences. The accuracy is calculated as the percentage of the predictions that match the sentiment code $\boldsymbol{c}$. Table 1 shows the results on 30K sentences by the two models which are trained with SST-full, SST-small, and Lexicon, respectively. We see that our method consistently outperforms S-VAE on all datasets. In particular, trained with only 250 labeled examples in SST-small, our model achieves reasonable generation accuracy, demonstrating the ability of learning disentangled representations with very little supervision. More importantly, given only word-level annotations in Lexicon, our model successfully transfers the knowledge to sentence level and generates desired sentiments reasonably well. Compared to our method that drives learning by directly assessing generated sentences, S-VAE attempts to capture sentiment semantics only by reconstructing labeled words, which is less efficient and gives inferior performance.

We next use the generated samples to augment the sentiment datasets and train sentiment classifiers. While not aiming to build best-performing classifiers on these datasets, the classification accuracy serves as an auxiliary measure of the sentence generation quality. That is, higher-quality sentences with more accurate sentiment attribute can predictably help yield stronger sentiment classifiers. Figure 3 shows the accuracy of classifiers trained on the four datasets with different augmentations. "Std" is a ConvNet trained on the standard original datasets, with the same network structure as with the sentiment discriminator in our model. "H-reg" additionally imposes the minimum entropy regularization on the generated sentences. "Ours" incorporates the minimum entropy regularization and the sentiment attribute code $\boldsymbol{c}$ of the generated sentences, as in Eq.(10). S-VAE uses the same protocol as our method to augment with the data generated by the S-VAE model. Comparison in Figure 3 shows that our method consistently gives the best performance on four datasets. For instance,

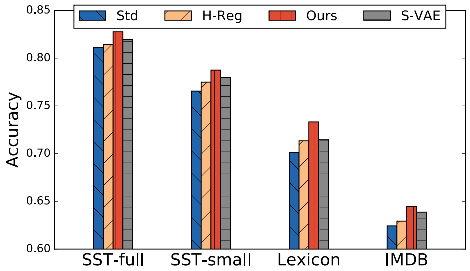

Figure 3. Test-set accuracy of classifiers trained on four sentiment datasets augmented with different methods (see text for details). The first three datasets use the SST-full test set for evaluation.

on Lexicon, our approach achieves 0.733 accuracy, compared to 0.701 of "Std". The improvement of "H-Reg" over "Std" shows positive effect of the minimum entropy regularization on generated sentences. Further incorporating the conditioned sentiment code of the generated samples, as in "Ours" and "S-VAE", provides additional performance gains, indicating the advantages of conditional generation for automatic creation of labeled data. Consistent with the above experiment, our model outperforms S-VAE.

4.2. Disentangled Representation

We study the interpretability of generation and the explicit independency constraint (Eq.7) for disentangled control.

Table 2 compares the samples generated by models with and without the constraint term, respectively. In the left column where the constraint applies, each pair of sentences, conditioned on different sentiment codes, are highly relevant in terms of, e.g., subject, tone, and wording which are not explicitly modeled in the structured code $\boldsymbol{c}$ while instead implicitly encoded in the unstructured code $\boldsymbol{z}$. Varying the sentiment code precisely changes the sentiment of the sentences (and paraphrases slightly to ensure fluency), while keeping other aspects unchanged. In contrast, the results in the right column, where the independency constraint is unactivated, show that varying the sentiment code not only changes the polarity of samples, but can also change other aspects unexpected to control, making the generation results less interpretable and predictable.

We demonstrate the power of learned disentangled representation by varying one attribute variable at a time. Table 3 shows the generation results. We see that each attribute variable in our model successfully controls its corresponding attribute, and is disentangled with other attribute code. The right column of the table shows meaningful variation of sentence tense as the tense code varies. Note that the semantic of tense is learned only from a lexicon without complete sentence examples. Our model successfully captures the key ingredients (e.g., verb "was" for past tense and "will be" for future tense) and combines with the knowledge of well-formed sentences to generate realistic samples with specified tense attributes. Table 4 further shows generated sentences with varying code $\boldsymbol{z}$ in different settings of structured attribute factors. We obtain samples that are diverse in content while consistent in sentiment and tense.

We also occasionally observed failure cases as in Table 5, such as implausible sentences, unexpected variations of irrelevant attributes, and inaccurate attribute generations. Improved modeling is expected such as using dilated convolutions as decoder, and decoding with beam search, etc. Better quantitative evaluations are also desired.

5. Discussions

We have proposed a deep generative model that learns interpretable latent representations and generates sentences with specified attributes. We obtained meaningful generation with restricted sentence length, and improved accuracy on sentiment and tense attributes. In the future we would like to improve the modeling and training as above, and extend to generate longer sentences/paragraphs and control more attributes with fine-grained structures.

Our approach combines VAEs with attribute discriminators and imposes explicit independency constraints on attribute controls, enabling disentangled latent code. Semi-supervised learning within the joint VAE/wake-sleep framework is effective with little or incomplete supervision. Hu et al. (2017) develop a unified view of a diverse set of deep generative paradigms, including GANs, VAEs, and wake-sleep algorithm. Our model can be alternatively motivated under the view as enhancing VAEs with the extended sleep phase and by leveraging generated samples.

Interpretability of the latent representations not only allows dynamic control of generated attributes, but also provides an interface that connects the end-to-end neural model with conventional structured methods. For instance, we can encode structured constraints (e.g., logic rules or probabilistic structured models) on the interpretable latent code, to incorporate prior knowledge or human intentions (Hu et al., 2016a;b); or plug the disentangled generation model into dialog systems to generate natural language responses from structured dialog states (Young et al., 2013).

Though we have focused on the generation capacity of our model, the proposed collaborative semi-supervised learning framework also helps improve the discriminators by generating labeled samples for data augmentation (e.g., see Figure 3). More generally, for any discriminative task, we can build a conditional generative model to synthesize additional labeled data. The accurate attribute generation of our approach can offer larger performance gains compared to previous generative methods.

Acknowledgments This research is supported by NSF IIS1447676, ONR N000141410684, and ONR N000141712463.

w/ independency constraint	**w/o independency constraint**
the film is strictly routine !	the acting is bad .
the film is full of imagination .	the movie is so much fun .
after watching this movie , i felt that disappointed .	none of this is very original .
after seeing this film , i ’m a fan .	highly recommended viewing for its courage , and ideas .
the acting is uniformly bad either .	too bland
the performances are uniformly good .	highly watchable
this is just awful .	i can analyze this movie without more than three words .
this is pure genius .	i highly recommend this film to anyone who appreciates music .

Table 2. Samples from models with or without independency constraint on attribute control (i.e., Eq.7). Each pair of sentences are generated with sentiment code set to “negative” and “positive”, respectively, while fixing the unstructured code $\boldsymbol{z}$. The SST-full dataset is used for learning the sentiment representation.

Varying the code of tense	
i thought the movie was too bland and too much	this was one of the outstanding thrillers of the last decade
i guess the movie is too bland and too much	this is one of the outstanding thrillers of the all time
i guess the film will have been too bland	this will be one of the great thrillers of the all time

Table 3. Each triple of sentences is generated by varying the tense code while fixing the sentiment code and $\boldsymbol{z}$.

Varying the unstructured code $\boldsymbol{z}$	
(“negative”, “past”)	*(“positive”, “past”)*
the acting was also kind of hit or miss .	his acting was impeccable
i wish i ’d never seen it	this was spectacular , i saw it in theaters twice
by the end i was so lost i just did n’t care anymore	it was a lot of fun
(“negative”, “present”)	*(“positive”, “present”)*
the movie is very close to the show in plot and characters	this is one of the better dance films
the era seems impossibly distant	i ’ve always been a big fan of the smart dialogue .
i think by the end of the film , it has confused itself	i recommend you go see this, especially if you hurt
(“negative”, “future”)	*(“positive”, “future”)*
i wo n’t watch the movie	i hope he ’ll make more movies in the future
and that would be devastating !	i will definitely be buying this on dvd
i wo n’t get into the story because there really is n’t one	you will be thinking about it afterwards, i promise you

Table 4. Samples by varying the unstructured code $\boldsymbol{z}$ given sentiment (“positive”/“negative”) and tense (“past”/“present”/“future”) code.

Failure cases	
the plot is not so original	it does n’t get any better the other dance movies
the plot weaves us into <unk>	it does n’t reach them , but the stories look
he is a horrible actor ’s most part	i just think so
he ’s a better actor than a standup	i just think !

Table 5. Failure cases when varying sentiment code with other codes fixed.

References

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

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

Chen, Xi, Duan, Yan, Houthooft, Rein, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2172–2180, 2016.

Diao, Qiming, Qiu, Minghui, Wu, Chao-Yuan, Smola, Alexander J, Jiang, Jing, and Wang, Chong. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 193–202. ACM, 2014.

Dosovitskiy, Alexey and Brox, Thomas. Generating images with perceptual similarity metrics based on deep networks. *arXiv preprint arXiv:1602.02644*, 2016.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Grandvalet, Yves, Bengio, Yoshua, et al. Semi-supervised learning by entropy minimization. In *NIPS*, volume 17, pp. 529–536, 2004.

Hinton, Geoffrey E, Dayan, Peter, Frey, Brendan J, and Neal, Radford M. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158, 1995.

Hu, Zhiting, Ma, Xuezhe, Liu, Zhengzhong, Hovy, Eduard, and Xing, Eric. Harnessing deep neural networks with logic rules. In *ACL*, 2016a.

Hu, Zhiting, Yang, Zichao, Salakhutdinov, Ruslan, and Xing, Eric P. Deep neural networks with massive learned knowledge. In *EMNLP*, 2016b.

Hu, Zhiting, Yang, Zichao, Salakhutdinov, Ruslan, and Xing, Eric P. On unifying deep generative models. *arXiv preprint arXiv:1706.00550*, 2017.

Kingma, Diederik P and Welling, Max. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, Diederik P, Mohamed, Shakir, Rezende, Danilo Jimenez, and Welling, Max. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pp. 3581–3589, 2014.

Kusner, Matt and Hernndez-Lobato, Jos. GANs for sequences of discrete elements with the Gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*, 2016.

Larsen, Anders Boesen Lindbo, Sønderby, Søren Kaae, and Winther, Ole. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016.

Mikolov, Tomas, Karafiát, Martin, Burget, Lukas, Cernockỳ, Jan, and Khudanpur, Sanjeev. Recurrent neural network based language model. In *Interspeech*, volume 2, pp. 3, 2010.

Odena, Augustus, Olah, Christopher, and Shlens, Jonathon. Conditional image synthesis with auxiliary classifier GANs. *arXiv preprint arXiv:1610.09585*, 2016.

Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Reed, Scott, Lee, Honglak, Anguelov, Dragomir, Szegedy, Christian, Erhan, Dumitru, and Rabinovich, Andrew. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Siddharth, N., Paige, Brooks, Desmaison, Alban, Meent, Jan-Willem van de, Wood, Frank, Goodman, Noah D., Kohli, Pushmeet, and Torr, Philip H.S. Learning disentangled representations in deep generative models. 2017.

Socher, Richard, Perelygin, Alex, Wu, Jean Y, Chuang, Jason, Manning, Christopher D, Ng, Andrew Y, Potts, Christopher, et al. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, pp. 1642. Citeseer, 2013.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Taigman, Yaniv, Polyak, Adam, and Wolf, Lior. Unsupervised cross-domain image generation. In *ICLR*, 2017.

Tang, Shuai, Jin, Hailin, Fang, Chen, and Wang, Zhaowen. Unsupervised sentence representation learning with adversarial auto-encoder. 2016.

van den Oord, Aaron, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks. In *ICML*, 2016.

Vinyals, Oriol, Toshev, Alexander, Bengio, Samy, and Erhan, Dumitru. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2015.

Wilson, Theresa, Wiebe, Janyce, and Hoffmann, Paul. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pp. 347–354. Association for Computational Linguistics, 2005.

Yang, Zichao, Hu, Zhiting, Salakhutdinov, Ruslan, and Berg-Kirkpatrick, Taylor. Improved variational autoencoders for text modeling using dilated convolutions. In *ICML*, 2017.

Young, Steve, Gašić, Milica, Thomson, Blaise, and Williams, Jason D. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101 (5):1160–1179, 2013.

Yu, Lantao, Zhang, Weinan, Wang, Jun, and Yu, Yong. SeqGAN: sequence generative adversarial nets with policy gradient. In *AAAI*, 2017.

Zhang, Yizhe, Gan, Zhe, and Carin, Lawrence. Generating text via adversarial training. In *NIPS Workshop on Adversarial Training*, 2016.

Zhou, Chunting and Neubig, Graham. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. In *ACL*, 2017.

Zhu, Jun-Yan, Krähenbühl, Philipp, Shechtman, Eli, and Efros, Alexei A. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pp. 597–613. Springer, 2016.