

---

# Controllable Text Generation

---

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, Eric P. Xing

School of Computer Science, Carnegie Mellon University  
 {zhitingh,zichaoy,xiaodanl,rsalakhu,epxing}@cs.cmu.edu

## 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 natural language sentences, whose attributes are dynamically controlled by learning disentangled latent representations with designated semantics. We propose a new neural generative model which combines variational auto-encoders and holistic attribute discriminators for effective imposition of semantic structures. With differentiable approximation to discrete text samples, explicit constraints on independent attribute controls, and efficient collaborative learning of generator and discriminators, our model learns highly interpretable representations from even only word annotations, and produces realistic sentences with desired attributes. Quantitative evaluation validates the accuracy of sentence and attribute generation.

## 1. Introduction

There is a surge of research interest in deep generative models, such as Variational Autoencoders (VAEs) (Kingma & Welling, 2013), Generative Adversarial Nets (GANs) (Goodfellow et al., 2014), and auto-regressive 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. Generating realistic text 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). 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 *controllable* text generation. That is, we focus on generating realistic natural language sentences, whose attributes can be dynamically 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 & Hernandez-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 independence property on the full latent representation, and varying individual code may result in unexpected variation of other unspecified attributes in addition to 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 unexpected 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.

Both quantitative and qualitative 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 sentences.

## 2. Related Work

Recent research has made remarkable progress in deep generative models. 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 log-likelihood of observed data under the generative model. The vanilla VAEs are incompatible with discrete latent variables as they hinder differentiable reparameterization for learning the encoder. The wake-sleep algorithm (Hinton et al., 1995) introduced for training deep directed graphical models shares similarity with VAEs by also combining an inference network with the generator. The wake phase updates the generator with the samples generated from the inference network on the training data, while the sleep phase updates the inference network based on the samples from the generator. Our method effectively combines VAEs with an extended wake-sleep algorithm in which the sleep procedure updates both the generator and the inference network (i.e., discriminators), enabling efficient collaborative semi-supervised learning.

Besides reconstruction in the raw data space, discriminator-based metric provides a different way for learning the generator, in which the discriminator assesses synthesized samples from the generator 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); Gatys et al. (2015) use discriminator networks for measuring high-level perceptual similarity. Applying discriminators to text generation is difficult due to the non-differentiability of the discrete samples. Yu et al. (2017) resort to policy learning which tends to have high variance during training. Zhang et al. (2016); Kusner & Hernandez-Lobato (2016) apply continuous approximations with only preliminary qualitative results. Bowman et al. (2015); Tang et al. (2016) 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. For instance, InfoGAN (Chen et al., 2016), which resembles the sleep procedure of our extended VAE/wake-sleep algorithm, disentangles latent representation in an unsupervised manner, while the semantic of each dimension is observed after training rather than designated by users in a controllable way. Siddharth et al. (2017); Kulkarni et al. (2015); Kingma et al. (2014) build in the context of VAEs and obtain disentangled image representations with semi-supervised learning. In contrast, our model combines VAEs with discriminators which provide a better, holistic metric compared to element-wise reconstruction, and thus boost effective imposition of semantic structures. 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 latent code with attributes not explicitly encoded. We address this by introducing an independency constraint, and show its effectiveness for improving interpretability.

## 3. Controllable Text Generation

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 dis-

criminator 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. In the next, we first present the intuitions and overview of our framework (section 3.1), and then describe our model in detail (section 3.2).

### 3.1. Model Overview

We build our framework starting from variational auto-encoders (section 2) which have been used for text generation (Bowman et al., 2015; Semeniuta et al., 2017), where sentence  $\hat{x}$  is generated conditioned on latent code  $z$ . The vanilla VAE employs an unstructured vector  $z$  in which the dimensions are entangled. To model and control the attributes of interest in an interpretable way, we augment the unstructured variables  $z$  with a set of structured variables  $c$  each of which targets a salient and independent semantic feature of sentences.

We want our sentence generator to condition on the combined vector  $(z, c)$ , and generate samples that fulfill the attributes as specified in the structured code  $c$ . Conditional generation in the context of VAEs 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  $c$ , 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  $c$  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  $z$  and thus be separated from  $c$  that we will manipulate. To this end, we reuse the VAE encoder as an additional discriminator for recognizing the attributes modeled in  $z$ , 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

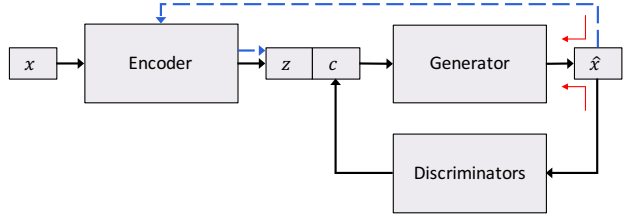


Figure 1. Structure of the generative model, where  $z$  is the unstructured latent code and  $c$  is the structured code targeting attributes of sentences to control. Blue dashed arrows denote the proposed independency constraint (section 3.2 for details), and red arrows denote gradient propagation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

attributes invariant as long as  $z$  is unchanged.

Figure 1 shows the overall model structure. Our complete model thus 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. This collaborative optimization mechanism can be regarded as resembling the wake-sleep algorithm for deep generative models. We show the combined VAE/wake-sleep learning system enables a highly efficient semi-supervised framework, which requires only a little supervision to construct 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 constructed with an LSTM-RNN for generating token sequence  $\hat{x} = \{\hat{x}_1, \dots, \hat{x}_T\}$  conditioned on the latent code  $(z, c)$ , which depicts a generative distribution:

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

where  $\hat{x}^{<t}$  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  $t$ .

That is,

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

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

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

$$\mathbf{z} \sim E(\mathbf{x}) = q_E(\mathbf{z}|\mathbf{x}). \quad (3)$$

Let  $\theta_G$  and  $\theta_E$  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  $p(\mathbf{z})$ :

$$\mathcal{L}_{\text{VAE}}(\theta_G, \theta_E; \mathbf{x}) = -\text{KL}(q_E(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})) + \mathbb{E}_{q_E(\mathbf{z}|\mathbf{x})q_D(\mathbf{c}|\mathbf{x})} [\log p_G(\mathbf{x}|\mathbf{z}, \mathbf{c})], \quad (4)$$

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

$$D(\mathbf{x}) = q_D(\mathbf{c}|\mathbf{x}). \quad (5)$$

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. 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  $\mathbf{c}$ . 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  $\tilde{G}_\tau(\mathbf{z}, \mathbf{c})$ , is fed into the discriminator<sup>1</sup> to measure the fitness to the target attribute, leading to the following loss for improving  $G$ :

$$\mathcal{L}_{\text{Attr}, \mathbf{c}}(\theta_G) = \mathbb{E}_{p(\mathbf{z})p(\mathbf{c})} [\log q_D(\mathbf{c}|\tilde{G}_\tau(\mathbf{z}, \mathbf{c}))]. \quad (6)$$

<sup>1</sup>The probability vector thus functions to average over the word embedding matrix to obtain a “soft” word embedding at each step.

The temperature  $\tau$  (Eq.(2)) is set to  $\tau \rightarrow 0$  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  $\mathbf{c}$  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  $\mathbf{c}$ , and thus varying a dimension of  $\mathbf{c}$  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  $\mathbf{c}$  by enforcing them to be fully captured by the unstructured part  $\mathbf{z}$ . Therefore, besides the attributes explicitly encoded in  $\mathbf{c}$ , we also train the generator so that other non-explicit attributes can be correctly recognized from the generated samples and match the unstructured code  $\mathbf{z}$ . Instead of building a new discriminator, we reuse the variational encoder  $E$  which serves precisely to infer the latents  $\mathbf{z}$  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  $q_E$ :

$$\mathcal{L}_{\text{Attr}, \mathbf{z}}(\theta_G) = \mathbb{E}_{p(\mathbf{z})p(\mathbf{c})} [\log q_E(\mathbf{z}|\tilde{G}_\tau(\mathbf{z}, \mathbf{c}))]. \quad (7)$$

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 objective of learning the generator:

$$\min_{\theta_G} \mathcal{L}_G = \mathcal{L}_{\text{VAE}} + \lambda_c \mathcal{L}_{\text{Attr}, \mathbf{c}} + \lambda_z \mathcal{L}_{\text{Attr}, \mathbf{z}}, \quad (8)$$

where  $\lambda_c$  and  $\lambda_z$  are balancing parameters. The variational encoder is trained by minimizing the VAE loss, i.e.,  $\min_{\theta_E} \mathcal{L}_{\text{VAE}}$ .

### 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  $\mathbf{z}$  which is learned in an unsupervised



**Algorithm 1** Controllable Text Generation

**Input:** A large corpus of unlabeled sentences  $\mathcal{X} = \{\mathbf{x}\}$   
 A few sentence attribute labels  $\mathcal{X}_L = \{(\mathbf{x}_L, \mathbf{c}_L)\}$   
 Parameters:  $\lambda_c, \lambda_z, \lambda_u, \beta$  – balancing parameters  
 1: Initialize the base VAE by minimizing Eq.(4) on  $\mathcal{X}$  with  $\mathbf{c}$  sampled from prior  $p(\mathbf{c})$   
 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  $(\mathbf{z}, \mathbf{c})$

manner, the structured variable  $\mathbf{c}$  uses labeled examples to entail designated semantics. We derive an efficient semi-supervised learning method for the discriminator.

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

$$\mathcal{L}_s(\theta_D) = \mathbb{E}_{\mathcal{X}_L} [\log q_D(\mathbf{c}_L | \mathbf{x}_L)]. \quad (9)$$

Besides, the conditional generator  $G$  is also capable of synthesizing (noisy) sentence-attribute pairs  $(\hat{\mathbf{x}}, \mathbf{c})$  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:

$$\mathcal{L}_u(\theta_D) = \mathbb{E}_{p_G(\hat{\mathbf{x}}|\mathbf{z}, \mathbf{c})p(\mathbf{z})p(\mathbf{c})} [\log q_D(\mathbf{c} | \hat{\mathbf{x}}) + \beta \mathcal{H}(q_D(\mathbf{c}' | \hat{\mathbf{x}}))], \quad (10)$$

where  $\mathcal{H}(q_D(\mathbf{c}' | \hat{\mathbf{x}}))$  is the empirical Shannon entropy of distribution  $q_D$  evaluated on the generated sentence  $\hat{\mathbf{x}}$ ; and  $\beta$  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:

$$\min_{\theta_D} \mathcal{L}_D = \mathcal{L}_s + \lambda_u \mathcal{L}_u, \quad (11)$$

where  $\lambda_u$  is the balancing parameter.

**Summarization and Discussion**

We have derived our model and its learning procedure in the above sections. 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  $\mathbf{c}$  at this time sampled from the prior distribution  $p(\mathbf{c})$ . Learning of the whole model then proceeds by

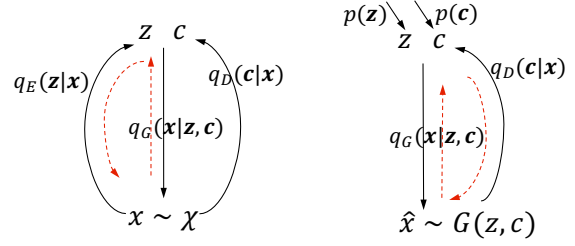


Figure 2. **Left:** The VAE and wake procedure, corresponding to Eq.(4). **Right:** The sleep procedure, corresponding to Eqs.(6)-(7) and (10). Black arrows denote inference and generation; red dashed arrows denote gradient propagation. The two steps in the sleep procedure, i.e., optimizing the discriminator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

alternating the optimization of the generator and the discriminator. The detailed learning procedure is summarized in Algorithm 1.

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 are used as targets for maximum likelihood training of the discriminator. This resembles the sleep phase of the wake-sleep algorithm. 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  $\mathbf{c}$  from the discriminator distribution  $q_D(\mathbf{c}|\mathbf{x})$  on observation  $\mathbf{x}$ , 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 phase, we show a little supervision is sufficient for learning structured representations.

**4. Experiments**

We evaluate our approach with both quantitative and qualitative experiments. Our model is demonstrated to be ef-

fective on sentence generation with controlled attributes of sentiment and tense. Our method improves over previous generative models on the accuracy of both attribute generation and classification. Highly disentangled representation is learned from a few labeled examples or even only word annotations. We validate the effect of the proposed independency constraint for interpretable generation, and show our model is able to produce convincing sentences with desired attributes.

#### 4.1. Setup

##### 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 16 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  $\leq 16$ , 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 contains 5K/1K/10K sentences in the train/dev/test sets, respectively.

**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<sup>2</sup> 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

We implement both the generator and the encoder as single-

Model	Dataset		
	SST-full	SST-small	Lexicon
S-VAE	0.822	0.679	0.660
Ours	<b>0.851</b>	<b>0.707</b>	<b>0.701</b>

Table 1. Accuracy of generated sentences measured by a pre-trained sentiment classifier (Hu et al., 2016a). Models are trained on the three sentiment datasets and generate 30K sentences, respectively. S-VAE denotes the semi-supervised VAE model (Kingma et al., 2014).

layer LSTM RNNs with input and hidden dimension of 300 and maximum sample length of 16. Discriminators are parameterized as convolutional networks. We give the detailed structure configurations in the supplements. To avoid vanishingly small KL term in the VAE (Bowman et al., 2015), we use a KL term weight linearly annealing from 0 to 1 during training. Neural parameters are trained using SGD with the Adam update rule (Kingma & Ba, 2014). We set the balancing parameters to  $\lambda_c = \lambda_z = \lambda_u = 0.1$ , and  $\beta$  is selected on the dev sets. All experiments were performed on a TITAN-X GPU and 128GB RAM. We implemented neural networks based on Tensorflow<sup>3</sup>.

#### 4.2. Accuracy of Generated Attributes

We quantitatively measure the performance of attribute control for sentences, by evaluating the accuracy of generating designated sentiment, and the effect of using the generated samples for sentiment classification. We compare our model with the semi-supervised variational auto-encoder (S-VAE) (Kingma et al., 2014), one of the few existing deep generative models capable of conditional sentence generation. S-VAE learns to reconstruct observed sentences given the attribute code, and no discriminators are used.

We use a state-of-the-art sentiment classifier (Hu et al., 2016a), which exhibits around 90% accuracy on the SST test set, to automatically evaluate the sentiment generation accuracy. Specifically, we generate sentences given sentiment code  $c$ , and then use the pre-trained sentiment classifier to assign sentiment labels to the generated sentences. We treat the predicted labels as ground truth, and compare with the sentiment code  $c$  to compute the attribute generation accuracy. Table 1 shows the results on 30K sentences produced by the two models which are trained with the SST-full, SST-small, and Lexicon datasets, respectively. It can be observed that our method consistently outperforms S-VAE on all datasets. In particular, trained with only 250 labeled examples provided in SST-

<sup>2</sup>www.timeml.org

<sup>3</sup>www.tensorflow.org

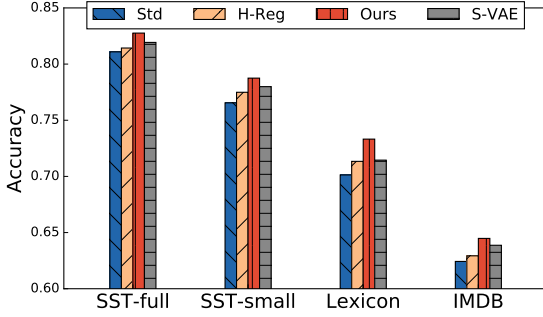


Figure 3. Test-set accuracy of classifiers trained on four sentiment datasets augmented with different methods (see text for details). The first three datasets use the SST-full test set for evaluation.

small, our model achieves reasonable generation accuracy, demonstrating the ability of our model to learn disentangled representations with very little supervision. More importantly, given only word-level annotations in Lexicon, our model successfully transfers the knowledge onto the 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 the sentiment semantic only by reconstructing labeled words, which is less efficient and results in 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 convolutional network 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 for enhancing the sentiment classifier. “Ours” incorporates the minimum entropy regularization and the sentiment attribute code  $c$  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 semi-supervised VAE model. Accuracy comparison in Figure 3 demonstrates that our method consistently gives the best performance on four datasets. For instance, on the Lexicon dataset, 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 sam-

ples, 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.

### 4.3. Disentangled Representation

We now investigate the interpretable representations learned by our model. We demonstrate the importance of the explicit independency constraint we introduce for disentangled attribute control and interpretable text generation. In addition, we show our model produces convincing natural language sentences and enables convenient and accurate control of the generated attributes.

We first study the effect of the independency constraint on unstructured code  $z$  as introduced in Eq.(7). 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  $c$  while instead implicitly encoded in the unstructured code  $z$ . Varying the sentiment code precisely changes the sentiment of the generated sentences (and paraphrases slightly when necessary 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 result in variations of other aspects not expected to controlled, 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 of the generated sentences, 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  $z$  in different settings of structured attribute factors. We obtain samples that are diverse in content while consistent in sentiment and tense. This further validates that our model is able to generate convincing sentences and attributes of interest in a controlled manner.

# Controllable Text Generation

w/ independency constraint	w/o independency constraint
the film is strictly routine ! the film is full of imagination .	the acting is bad . the movie is so much fun .
after watching this movie , i felt that disappointed . after seeing this film , i 'm a fan .	none of this is very original . highly recommended viewing for its courage , and ideas .
the acting is uniformly bad either . the performances are uniformly good .	too bland highly watchable
this is just awful . this is pure genius .	i can analyze this movie without more than three words . i highly recommend this film to anyone who appreciates music .
nothing about this film is amazing nothing about this film is terrible	a movie version of a paint by numbers a backstage must see for true fans of the bard

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  $z$ . The SST-full dataset is used for learning the sentiment representation.

Varying the code of sentiment	Varying the code of tense
this movie was awful and boring . this movie was funny and touching .	this was one of the outstanding thrillers of the last decade this is one of the outstanding thrillers of the all time this will be one of the great thrillers of the all time
jackson is n’t very good with documentary jackson is superb as a documentary productions	i thought the movie was too bland and too much i guess the movie is too bland and too much i guess the film will have been too bland
you will regret it you will enjoy it	

Table 3. Samples by varying one attribute code while fixing the others. Left column: each pair of sentences is generated by varying the sentiment code while fixing the tense code and  $z$ . Right column: each triple of sentences is generated by varying the tense code while fixing the sentiment code and  $z$ .

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

Table 4. Samples by varying the unstructured code  $z$  given sentiment (“positive” or “negative”) and tense (“past”, “present”, or “future”) code.



## 5. Conclusions & Discussions

We have proposed a deep generative model capable of learning interpretable latent representations and generating realistic sentences with specified attributes. Our approach combines VAEs with attribute discriminators and imposes explicit independency constraints on attribute controls, enabling learning disentangled latent code. Semi-supervised learning within the extended VAE/wake-sleep framework is effective with little or incomplete supervision. We show meaningful generation results and improved generation accuracy. Our method shows strong promise for generating more complex text and structured data.

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.

## 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 *NIPS*, 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 *KDD*, 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.
- Gatys, Leon A, Ecker, Alexander S, and Bethge, Matthias. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.
- Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *NIPS*, 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.
- Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- 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 *NIPS*, pp. 3581–3589, 2014.
- Kulkarni, Tejas D, Whitney, William F, Kohli, Pushmeet, and Tenenbaum, Josh. Deep convolutional inverse graphics network. In *NIPS*, pp. 2539–2547, 2015.
- Kusner, Matt and Hernandez-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.
- 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.
- Semeniuta, Stanislaw, Severyn, Aliaksei, and Barth, Erhardt. A hybrid convolutional variational autoencoder for text generation. *arXiv preprint arXiv:1702.02390*, 2017.
- 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 *EMNLP*, volume 1631, pp. 1642. Citeseer, 2013.
- 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 *CVPR*, pp. 3156–3164, 2015.
- Wilson, Theresa, Wiebe, Janyce, and Hoffmann, Paul. Recognizing contextual polarity in phrase-level sentiment analysis. In *EMNLP*, pp. 347–354. Association for Computational Linguistics, 2005.
- 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.
- Zhu, Jun-Yan, Krähenbühl, Philipp, Shechtman, Eli, and Efros, Alexei A. Generative visual manipulation on the natural image manifold. In *ECCV*, pp. 597–613. Springer, 2016.