Improved Variational Autoencoders for Text Modeling using Dilated Convolutions

Zichao Yang | Zhiting Hu | Ruslan Salakhutdinov | Taylor Berg-Kirkpatrick

Abstract

Recent work on generative modeling of text has found that variational autoencoders (VAE) incorporating LSTM decoders perform worse than simpler LSTM language models (Bowman et al., 2015). This negative result is so far poorly understood, but has been attributed to the propensity of LSTM decoders to ignore conditioning information from the encoder. In this paper, we experiment with a new type of decoder for VAE: a dilated CNN. By changing the decoder’s dilation architecture, we control the effective context from previously generated words. In experiments, we find that there is a trade off between the contextual capacity of the decoder and the amount of encoding information used. We show that with the right decoder, VAE can outperform LSTM language models. We demonstrate perplexity gains on two datasets, representing the first positive experimental result on the use VAE for generative modeling of text. Further, we conduct an in-depth investigation of the use of VAE (with our new decoding architecture) for semi-supervised and unsupervised labeling tasks, demonstrating gains over several strong baselines.

1. Introduction

Generative modeling techniques play an important role in many machine learning application areas. Generative models allow for principled and effective use of unlabeled data and therefore facilitate unsupervised and semi-supervised learning. Recent use of deep neural networks inside of generative models has lead to model classes that are particularly flexible and can potentially model a wide range of data and modalities, including both images and text. We focus on a specific instance of this class: the variational autoencoder (VAE) (Kingma & Welling, 2013).

The generative story behind the VAE (to be described in detail in the next section) is simple: First, a continuous latent representation is sampled from a Gaussian. Then, an observed sample is generated from a neural decoder, conditioned on the latent representation. The latent representation (which must be marginalized out) is intended to give the model more expressive capacity when compared with simpler neural generative models—for example, conditional language models. Since effective variational techniques have been developed for learning VAEs (their namesake) (Kingma & Welling, 2013), these models have been successfully applied to image modeling and generation (Gregor et al., 2015; Salimans et al., 2015; Yan et al., 2016).

However, the application of VAEs to text data has been far less successful (Bowman et al., 2015; Miao et al., 2016). The obvious choice for decoding architecture for a textual VAE is an LSTM, a typical workhorse in the language processing community. Bowman et al. (2015) demonstrated negative results using VAEs for text modeling, finding that they perform worse than LSTM language models. In particular, they observe that the LSTM decoder does not make effective use of the latent representation (even when combined with more sophisticated training techniques) and as a result VAE collapses to a simple language model. Related work (Miao et al., 2016; Larochelle & Lauly, 2012; Mnih & Gregor, 2014) has used simpler decoders that model text as a bag of words. Their results indicate better use of latent representations, but their decoders are too simple to effectively model longer-range dependencies in text.

Motivated by these observations, we hypothesize that the contextual capacity of the decoder plays an important role in whether VAEs effectively condition on the latent representation when trained on text data. We propose the use of a dilated CNN as a decoder in VAE, inspired by the recent success of using CNN for audio, image and lan-
language modeling (van den Oord et al., 2016a; Kalchbrenner et al., 2016a; van den Oord et al., 2016b). In contrast with this prior work where extremely large CNNs are used, we exploit the dilated CNN for its flexibility in varying the amount of conditioning context. In the two extremes, depending on the choice of dilation, the CNN decoder cannot reproduce a simple MLP using a bag of words representation of text, or can reproduce the long-range dependence of recurrent architectures (like an LSTM) by conditioning on the entire history. Thus, by choosing a dilated CNN as the decoder, we are able to conduct experiments where we vary contextual capacity, finding a sweet spot where the decoder can accurately model text but does not yet overpower the latent representation produced by the encoder. We demonstrate that when this trade off is correctly managed, textual VAEs can perform substantially better than simple LSTM language models, a finding consistent with recent image modeling experiments using variational lossy autoencoders (Chen et al., 2016). We go on to show that VAEs with carefully selected CNN decoders can be quite effective for semi-supervised classification and unsupervised clustering, outperforming several strong baselines on both text categorization and sentiment analysis.

Our contributions are as follows: First, we propose the use of a dilated CNN as a new decoder for VAE. We then empirically evaluate several dilation architectures with different capacities, finding that reduced contextual capacity leads to stronger reliance on latent representations. By picking a decoder with suitable contextual capacity, we find our VAE performs better than LSTM language models on two data sets. We explore the use of dilated CNN VAEs for semi-supervised classification and find they perform better than strong baselines from Dai & Le (2015). Finally, we verify that the same framework can be used effectively for unsupervised clustering.

2. Model

In this section, we begin by providing background on the use of variational autoencoders for language modeling. Then we introduce the dilated CNN architecture that we will use in experiments as a new decoder for VAE. Finally, we describe the generalization of VAE that we will use to conduct experiments on semi-supervised classification and unsupervised clustering.

2.1. Variational Autoencoder for Language Modeling

Language models (Mikolov et al., 2011) typically generate each token $x_t$ conditioned on the entire history of previously generated tokens:

$$p(x) = \prod_t p(x_t|x_1, x_2, ..., x_{t-1}).$$

State-of-the-art language models generally parametrize these conditional probabilities using RNNs, which compute an evolving hidden state over the sequence and predicts $x_t$ based on the hidden state. Such models, though effective in modeling text, do not learn a vector that represents the full sequence (Bowman et al., 2015).

Bowman et al. (2015) propose a different approach to generative text modeling. Instead of modeling the joint probability $p(x)$ directly as in Equation 1, we specify a generative process for which $p(x)$ is a marginal distribution. Specifically, we first generate a continuous latent vector representation $z$ from a Gaussian prior $p(z)$, and then generate the sequence $x$ from a conditional distribution (the decoder) $p(x|z)$. To estimate parameters for this model we would like to maximize the marginal probability $p(x) = \int p(z)p(x|z)dz$. The marginal probability is intractable, but the following variational lower bound is often used as an objective:

$$-\log p_\theta(x) = -\log \int p_\theta(z)p_\theta(x|z)dz$$

$$\leq \mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z) - \log p_\theta(z) + \log q_\phi(z|x)]$$

$$= -\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] + \text{KL}(q_\phi(z|x)||p_\theta(z)).$$

We optimize the lower bound w.r.t. the model parameters $\theta$ and the parameters of our approximation to posterior, $\phi$ (often called the recognition model or encoder.) In order for the bound to be tight, the posterior probability $p_\phi(z|x)$ needs to be close to the true posterior. $p_\phi(z|x)$ is typically assumed to be Gaussian so that the re-parametrization trick from Kingma & Welling (2013) can be used.

This model and inference procedure are often referred to as a VAE. In contrast with Equation 1, this distribution conditions on a latent representation $z$:

$$p(x|z) = \prod_t p(x_t|x_1, x_2, ..., x_{t-1}, z).$$

The desired result is that learned representations $z$ contains some high level information such as topic, which is helpful in predicting tokens $x_t$.

We can also view the VAE as a regularized version of the autoencoder. If only the first part of the lower bound objective $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$ is used as the objective function, the variance of the posterior probability $q_\phi(z|x)$ will be very small and it collapses to an autoencoder. With the regularization from the KL-divergence term $\text{KL}(q_\phi(z|x)||p_\theta(z))$, the variational autoencoder not just learns to encode $x$ as a single point $z$, it instead learns a distribution over the latent space.

The encoder (recognition model) and decoder (generative model) are typically parametrized with neural networks. For images, the encoder and decoder can be MLPs or
CNNs. For text, a RNN such as a LSTM is used as in (Bow-
man et al., 2015). However, the authors find the decoder de-
pends too much on context information and the latent rep-
resentation from the encoder is ignored. We suspect that it is the decoder model that plays an important role. If the de-
coder relies too much on context, the VAE tends to ignore the latent representation, turning into a standard RNN lan-
guage model. Hence, we propose to use a dilated CNN as the decoder. The architecture flexibility of CNNs allows us to change the contextual capacity, hence control the context information and latent representation trade-off. In two extreme cases, when the effective contextual width of a CNN is very large, it resembles the behavior of LSTM and when it is very small, it behaves like a bag of words model.

2.2. Dilated Convolutional Decoder

The CNN used for text modeling (Kalchbrenner et al.,
2016a) is similar to that used for images (Krizhevsky et al.,
2012; He et al., 2016), but with the convolution applied in
one dimension.

One Dimensional Convolution: Note that \( x_t \) can only
condition on past tokens \( x_{<t} \), applying the traditional con-
volution will break this and use tokens \( x_{>t} \) as inputs to pre-
dict \( x_t \). We can avoid this either by applying a mask on the
convolution filter or shift the input by several slots (van den
Oord et al., 2016b). Here we adopt the second approach.

The overall model architecture is shown in Figure 1.

Suppose we use convolution with filter size \( k \) and use \( n \)
layers, then the effective filter size (the number of past to-
kens to condition to in predicting \( x_t \)) is \((k-1) \times n + 1\).
The filter size grows linearly with the depth of the network.

Dilation: Dilated convolution (Yu & Koltun, 2015) was in-
troduced to greatly increase the effective receptive field size
without increasing the computational cost. With dilation \( d \),
the convolution is applied so that the inputs are skipped
\( d-1 \) values. Casual convolution can be seen a special
case with \( d = 1 \). With dilation, the effective receptive size
grows exponentially with network depth. In Figure 1 we

use dilation of size 2, 4 in the second and third layer. Sup-
pose the dilation size in the \( i \)-th layer is \( d_i \) and we use the
same filter size \( k \) in all layers, then the effective filter size is
\((k-1) \sum d_i + 1\). The dilations are typically set to double
every layer \( d_{i+1} = 2d_i \), hence the effective receptive field
size can grow exponentially. Hence, the contextual capac-
ity of a CNN can be controlled by manipulating the filter
size, dilation size and network depth.

Residual Connection: Residual connection (He et al.,
2016) is used in the decoder to speed up convergence and
enable us to train deep models. Our residual block is sim-
ilar to that of (Kalchbrenner et al., 2016a) and is shown in
Figure 2. We use three convolutional layers with filter size
\( 1 \times 1 \), \( 1 \times k \), \( 1 \times 1 \) respectively. ReLU activation function is
used between the convolutional layers. The residual block

can be more powerful by adding batch normalization and
gating mechanism (van den Oord et al., 2016b; Kalchbren-
ner et al., 2016a).

Overall architecture: Our VAE architecture is shown in
Figure 2. We use LSTM as the encoder to get the poste-
rior probability \( q(z|x) \), which we assume to be diagonal
Gaussian. We parametrize the mean \( \mu \) and variance \( \sigma \) with
LSTM output. We sample \( z \) from \( q(z|x) \), the decoder is con-
tioned on the sample by concatenating \( z \) with every
word embedding of the decoder input.

2.3. Semi-supervised VAE

In this section, we briefly review semi-supervised VAEs of
(Kingma et al., 2014) that can incorporate labels. Given the
labeled set \( (x, y) \sim D_L \) and the unlabeled set \( x \sim D_U \),

\[
\begin{align*}
    q(z|x) &= \mathcal{N}(\mu(x, y), \sigma(x, y)) \\
    z &= q(z|x) \\
    p(x|y, z) &= \mathcal{N}(\mu(y, z), \sigma(y, z)) \\
    l(x, y) &= \mathcal{L}(x, y) + \mathcal{D}(\mu(x, y), \sigma(x, y), \mu(y, z), \sigma(y, z))
\end{align*}
\]

\[
\begin{align*}
    \mathcal{L}(x, y) &= -\sum_{i=1}^{T} \log p(x_i|x_{<i}, y) \\
    \mathcal{D}(\mu(x, y), \sigma(x, y), \mu(y, z), \sigma(y, z)) &= \mathcal{D}(\mu(x, y), \sigma(x, y)) + \mathcal{D}(\mu(y, z), \sigma(y, z))
\end{align*}
\]
Improved Variational Autoencoders for Text Modeling using Dilated Convolutions

Kingma et al. [2014] proposed a semi-supervised VAE model whose latent representation contains both continuous variable $z$ and discrete label $y$:

$$p(x, y, z) = p(y)p(z)p(x|y, z). \quad (3)$$

The semi-supervised VAE trains a discriminative network $q(y|x)$, an inference network $q(z|x, y)$ and a generative network $p(x|y, z)$ jointly by minimizing the variational lower bound. For labeled data $(x, y)$, the variational lower bound is

$$- \log p(x, y) \leq - \mathbb{E}_{q(y|x)}[\log p(x|y, z)] + \text{KL}(q(z|x, y)||p(z)) - \log p(y) = L(x, y) - \log p(y).$$

For unlabeled data $x$, the label $y$ is treated as a latent variable and marginalized out in the training objective:

$$- \log p(x) \leq \mathbb{E}_{q(y|x)}[\mathbb{E}_{q(z|x, y)}[\log p(x|y, z)]] + \text{KL}(q(z|x, y)||p(z)) - \log q(y|x) + \log q(y|x) = \sum_y q(y|x)L(x, y) + \text{KL}(q(y|x)||p(y)).$$

Combining the labeled and unlabeled data loss, we have the overall objective as:

$$J = \mathbb{E}_{(x, y) \sim D_L}[L(x, y)] + \mathbb{E}_{x \sim D_U}[U(x)] + \alpha \mathbb{E}_{(x, y) \sim D_L}[\log q(y|x)],$$

where $\alpha$ controls the trade off between generative loss and discriminative loss.

Since $y$ is a discrete variable, we have to compute the marginal probability by iterating all classes. The computational cost scales linearly with the number of classes.

Gumbel-Softmax: Yang et al. [2016] and Maddison et al. [2016] propose a continuous approximation to the samples of categorical distribution. Let $u$ be a categorical distribution with probabilities $\pi_1, \pi_2, ..., \pi_c$, the samples from categorical distribution can be approximated using:

$$y_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^c \exp((\log(\pi_j) + g_j)/\tau)}, \quad (4)$$

where $g_i$ follows Gumbel(0, 1). We can obtain the samples from Gumbel distribution by first sample $u \sim \text{Uniform}(0, 1)$ and then compute $g = - \log(- \log(u))$. The approximation is accurate when $\tau \to 0$ and is smooth when $\tau > 0$. In experiments, we anneal $\tau$ so that it is large and sample variance is small at beginning and then gradually decrease $\tau$.

We use Gumbel-Softmax to approximate the samples from $p(y|x)$ to reduce the computational cost. We can directly back propagate the gradients of $U(x)$ to the discriminator network.

Unsupervised clustering: In this section we adapt the same framework for unsupervised clustering. We directly minimize the objective $U(x)$, which is consisted of two parts: reconstruction loss and KL regularization on $q(y|x)$. The first part encourages the model to assign $x$ to label $y$ such that the reconstruction loss is low. We find that the model can easily get stuck in two local optimum: the KL term is very small and $q(y|x)$ is close to uniform distribution or the KL term is very large and all samples collapse to one class. In order to make the model more robust, we modify the KL term by:

$$\text{KL}_y = \max(\gamma, \text{KL}(q(y|x)||p(y))). \quad (5)$$

That is, we only minimize the KL term when it is large enough.

3. Experiments

3.1. Data sets

Since we would like to investigate VAEs for language modeling and semi-supervised classification, the data sets should be suitable for both purposes. We use two large scale document classification data sets: Yahoo Answer and Yelp15 review, representing topic classification and sentiment classification data sets respectively [Tang et al., 2015; Yang et al., 2016; Zhang et al., 2015]. The original data sets contain millions of samples, of which we sample 100k as training and 10k as validation and test from the respective partitions. The detailed statistics of both data sets are in Table 1. Yahoo Answer contains 10 topics including Society & Culture, Science & Mathematics etc. Yelp15 contains 5 level of rating, with higher rating better.

3.2. Model configurations and Training details

We use an LSTM as an encoder for VAE and explore LSTMs and CNNs as decoders. For CNNs, we explore several different configurations. We set the convolution filter size to be 3 and gradually increase the depth and dilation from [1, 2, 4], [1, 2, 4, 8, 16] to [1, 2, 4, 8, 16, 1, 2, 4, 8, 16]. They represent small, medium and large model and we name them as SCNN, MCNN and LCNN. We also explore a very large model with dilations [1, 2, 4, 8, 16, 1, 2, 4, 8, 16, 1, 2, 4, 8, 16] and name it as VLCNN. The effective filter size are 15, 63, 125 and 187 respectively. We use the

| Data    | classes | documents | average #w | vocabulary |
|---------|---------|-----------|-------------|------------|
| Yahoo   | 10      | 100k      | 78          | 200k       |
| Yelp15  | 5       | 100k      | 96          | 90k        |

Table 1: Data statistics


Improved Variational Autoencoders for Text Modeling using Dilated Convolutions

### Table 2: Language modeling results on the test set.

| Model          | Size | NLL (KL) | PPL |
|----------------|------|----------|-----|
| LSTM-LM        | $< i$ | 334.9    | 66.2 |
| LSTM-VAE**     | $< i$ | 342.1 (0.0) | 72.5 |
| LSTM-VAE** + init | $< i$ | 339.2 (0.0) | 69.9 |
| SCNN-LM        | 15   | 345.3    | 75.5 |
| SCNN-VAE       | 15   | 337.8 (13.3) | 68.7 |
| SCNN-VAE + init | 15   | 335.9 (13.9) | 67.0 |
| MCNN-LM        | 63   | 338.3    | 69.1 |
| MCNN-VAE       | 63   | 336.2 (11.8) | 67.3 |
| MCNN-VAE + init | 63   | 334.6 (12.6) | 66.0 |
| LCNN-LM        | 125  | 335.4    | 66.6 |
| LCNN-VAE       | 125  | 333.9 (6.7) | 65.4 |
| LCNN-VAE + init | 125  | **332.1 (10.0)** | **63.9** |
| VLCNN-LM       | 187  | 336.5    | 67.6 |
| VLCNN-VAE      | 187  | 336.5 (0.7) | 67.6 |
| VLCNN-VAE + init | 187  | 335.8 (3.8) | 67.0 |
| LSTM-LM        | $< i$ | 362.7    | 42.6 |
| LSTM-VAE**     | $< i$ | 372.2 (0.3) | 47.0 |
| LSTM-VAE** + init | $< i$ | 368.9 (4.7) | 46.4 |
| SCNN-LM        | 15   | 371.2    | 46.3 |
| SCNN-VAE       | 15   | 365.6 (9.4) | 43.9 |
| SCNN-VAE + init | 15   | 363.7 (10.3) | 43.1 |
| MCNN-LM        | 63   | 366.5    | 44.3 |
| MCNN-VAE       | 63   | 363.0 (6.9) | 42.8 |
| MCNN-VAE + init | 63   | 360.7 (9.1) | 41.8 |
| LCNN-LM        | 125  | 363.5    | 43.0 |
| LCNN-VAE       | 125  | 361.9 (6.4) | 42.3 |
| LCNN-VAE + init | 125  | 359.1 (7.6) | 41.1 |
| VLCNN-LM       | 187  | 364.8    | 43.7 |
| VLCNN-VAE      | 187  | 364.3 (2.7) | 43.4 |
| VLCNN-VAE + init | 187  | 364.7 (2.2) | 43.5 |

(a) Yahoo

Table 2: Language modeling results on the test set. ** is from (Bowman et al., 2015). We report both negative log likelihood (NLL) and perplexity (PPL) on the test set. The KL cost of NLL is in the parenthesis. Size means effective filter size. init means we initialize the encoder of VAE with LSTM LM.

last hidden state of the encoder LSTM and feed it though an MLP to get the mean and variance of $q(z|x)$, from which we sample $z$ and then feed it through an MLP to get the starting state of decoder. For the LSTM decoder, we follow (Bowman et al., 2015) to use it as the initial state of LSTM and feed it to every step of LSTM. For the CNN decoder, we concatenate it with the word embedding of every decoder input.

The architecture of the Semi-supervised VAE basically follows that of the VAE. We feed the last hidden state of the encoder LSTM through a two layer MLP then a softmax to get $q(y|x)$. We use Gumbel-softmax to sample $y$ from $q(y|x)$. We then concatenate $y$ with the last hidden state of encoder LSTM and feed them through an MLP to get the mean and variance of $q(z|y, x)$. $y$ and $z$ together are used as the starting state of the decoder.

We use a vocabulary size of 20k for both data sets and set the word embedding dimension to be 512. The LSTM dimension is 1024. The number of channels for convolutions in CNN decoders is 512 internally and 1024 externally, as shown in Figure 3. We select the dimension of $z$ from [32, 64]. We find our model is not sensitive to this parameter.

We use Adam (Kingma & Ba, 2014) to optimize all models and the learning rate is selected from [2e-3, 1e-3, 7.5e-4] and $\beta_1$ is selected from [0.5, 0.9]. Empirically, we find learning rate 1e-3 and $\beta_1 = 0.5$ to perform the best. We select drop out ratio of LSTMs (both encoder and decoder) from [0.3, 0.5]. Following (Bowman et al., 2015), we also use drop word for the LSTM decoder, the drop word ratio is selected from [0, 0.1, 0.3, 0.5, 0.7]. For the CNN decoder, we use a drop out ratio of 0.1 at each layer. We do not use drop word for CNN decoders. We use batch size of 32 and all model are trained for 40 epochs. We start to half the learning rate every 2 epochs after epoch 30. Following (Bowman et al., 2015), we use KL cost annealing strategy. We set the initial weight of KL cost term to be 0.01 and increase it linearly until a given iteration $T$. We treat $T$ as a hyper parameter and select it from [10k, 40k, 80k].

### 3.3. Language modeling results

The results for language modeling are shown in Table 2. We report the negative log likelihood (NLL) and perplexity (PPL) of the test set. For the NLL of VAEs, we decompose it into reconstruction loss and KL divergence and report the KL divergence in the parenthesis. To better visualize these results, we plot the results of Yahoo data set (Table 2a) in Figure 4.

We first look at the LM results for Yahoo data set. As we gradually increase the effective filter size of CNN from SCNN, MCNN to LCNN, the NLL decreases from 345.3, 338.3 to 335.4. The NLL of LCNN-LM is very close to the NLL of LSTM-LM 334.9. But VLCNN-LM is a little bit worse than LCNN-LM, this indicates a little bit of over-fitting.
The cases are different when we use the CNNs as decoders for VAEs. We can see that LSTM-VAE is worse than LSTM-LM in terms of NLL and the KL term is nearly zero, which verifies the finding of Bowman et al. (2015). When we use CNNs as the decoders for VAEs, we can see improvement over pure CNN LMs. For SCNN, MCNN and LCNN, the VAE results improve over LM results from 345.3 to 337.8, 338.3 to 336.2, and 335.4 to 333.9 respectively. The improvement is big for small models and gradually decreases as we increase the decoder model contextual capacity. When the model is as large as VLCNN, the improvement diminishes and the VAE result is almost the same with LM result. This is also reflected in the KL term. SCNN-VAE has the largest KL of 13.3 and VLCNN-VAE has the smallest KL of 0.7. When LCNN is used as the decoder, we obtain an optimal trade off between utilizing contextual information and latent representation. LCNN-VAE achieves a NLL of 333.9, which improves over LSTM-LM with NLL of 334.9.

We find that if we initialize the parameters of LSTM encoder with parameters of LSTM language model, we can improve the VAE results further. This indicates better encoder model is also a key factor for VAEs to work well. Combined with encoder initialization, LCNN-VAE improves over LSTM-LM from 343.9 to 332.1 in NLL and from 66.2 to 63.9 in PPL.

Similar observation is found for the sentiment data set Yelp in Table 2. LCNN-VAE improves over LSTM-LM from 362.7 to 359.1 in NLL and from 42.6 to 41.1 in PPL.

**Latent representation visualization:** In order to visualize the latent representation, we set the dimension of $z$ to be 2 and plot the mean of posterior probability $q(z|x)$, as shown in Figure 5. We can see distinct different characteristics of topic and sentiment representation. In Figure 5a, we can see that documents of different topics fall into different clusters, while in Figure 5b, documents of different ratings form a continuum, they lie continuously on the x-axis as the review rating increases. This is consistent with sentiment actually being real-valued.

| Model                | ACCU | NLL (KL)    |
|----------------------|------|-------------|
| LSTM-VAE-Semi        | 51.9 | 345.5 (9.3) |
| SCNN-VAE-Semi        | 65.5 | 335.7 (10.4)|
| MCNN-VAE-Semi        | 64.6 | 332.8 (7.2) |
| LCNN-VAE-Semi        | 57.2 | 331.3 (2.7) |

Table 3: Semi-supervised VAE ablation results on Yahoo. We report both the NLL and classification accuracy of the test data. Accuracy is in percentage. Number of labeled samples is fixed to be 500.

### 3.4. Semi-supervised VAE results

Motivated by the success of VAEs for language modeling, we continue to explore VAEs for semi-supervised learning. Following that of Kingma et al. (2014), we set the number of labeled samples to be 100, 500, 1000 and 2000 respectively.

**Ablation Study:** At first, we would like to explore the effect of different decoders for semi-supervised classification. We fix the number of labeled samples to be 500 and report both classification accuracy and NLL of the test set of Yahoo data set in Table 5. We can see that SCNN-VAE-Semi has the best classification accuracy of 65.5. The accuracy decreases as we gradually increase the decoder contextual capacity. On the other hand, LCNN-VAE-Semi has the best NLL result. This classification accuracy and NLL trade off once again verifies our conjecture: with small contextual window size, the decoder is forced to use the encoder information, hence the latent representation is better learned.

Comparing the NLL results of Table 5 with that of Table 2, we find that initializing the parameters of LSTM encoder with parameters of LSTM language model, we can improve the VAE results further. This indicates better encoder model is also a key factor for VAEs to work well. Combined with encoder initialization, LCNN-VAE improves over LSTM-LM from 343.9 to 332.1 in NLL and from 66.2 to 63.9 in PPL.
We find empirically that simply using the features does not perform well since the features are high dimensional. We run a PCA on these features, the dimension of PCA is selected from [8, 16, 32]. Since GMM can easily get stuck in poor local optimum, we run each model ten times and report the best result.

We find directly optimizing $U(x)$ does not perform well for unsupervised clustering and we need to initialize the encoder with LSTM language model. The model only works well for Yahoo data set. This is potentially because Figure 5b shows that sentiment latent representations does not fall into clusters. $\gamma$ in Equation 5 is a sensitive parameter, we select it from the range between 0.5 and 1.5 with an interval of 0.1.

We use the following evaluation protocol (Makhzani et al., 2015): after we finish training, for cluster $i$, we find out the validation sample $x_n$ from cluster $i$ that has the best $q(y_i|x)$ and assign the label of $x_n$ to all samples in cluster $i$. We then compute the test accuracy based on this assignment. The detailed results are in Table 5. We can see SCNN-VAE-Unsup + init performs better than other baselines.

### Table 5: Unsupervised clustering results for Yahoo data set.

| Model                        | ACCU |
|------------------------------|------|
| LSTM + GMM                   | 25.8 |
| SCNN-VAE + GMM               | 56.6 |
| SCNN-VAE + init + GMM        | 57.0 |
| SCNN-VAE-Unsup + init        | 59.9 |

We also explored using the same framework for unsupervised clustering. We compare with the baselines that extract the feature with existing models and then run Gaussian Mixture Model (GMM) on these features. We find empirically that simply using the features does not perform well since the features are high dimensional. We run a PCA on these features, the dimension of PCA is selected from [8, 16, 32]. Since GMM can easily get stuck in poor local optimum, we run each model ten times and report the best result.

We find directly optimizing $U(x)$ does not perform well for unsupervised clustering and we need to initialize the encoder with LSTM language model. The model only works well for Yahoo data set. This is potentially because Figure 5b shows that sentiment latent representations does not fall into clusters. $\gamma$ in Equation 5 is a sensitive parameter, we select it from the range between 0.5 and 1.5 with an interval of 0.1.

We use the following evaluation protocol (Makhzani et al., 2015): after we finish training, for cluster $i$, we find out the validation sample $x_n$ from cluster $i$ that has the best $q(y_i|x)$ and assign the label of $x_n$ to all samples in cluster $i$. We then compute the test accuracy based on this assignment. The detailed results are in Table 5. We can see SCNN-VAE-Unsup + init performs better than other baselines. LSTM+GMM performs very bad probably because the feature dimension is 1024 and is too high for GMM, even though we already used PCA to reduce the dimension.

### 3.6. Conditional text generation

With the semi-supervised VAE, we are able to generate text conditional on the label. Due to space limitation, we only...
1 star the food was good but the service was horrible. took forever to get our food. we had to ask twice for our check after we got our food. will not return.

2 star the food was good, but the service was terrible. took forever to get someone to take our order. had to ask 3 times to get the check. food was ok. nothing to write about.

3 star came here for the first time last night. food was good. service was a little slow. food was just ok.

4 star food was good, service was a little slow, but the food was pretty good. i had the grilled chicken sandwich and it was really good. will definitely be back!

5 star food was very good, service was fast and friendly. food was very good as well. will be back!

Table 6: Text generated by conditioning on sentiment label.

show one example of generated reviews conditioning on review rating in Table 6. More examples of text generated conditioning on topic and rating are shown in the Appendix. For each group of generated text, we fix z and vary the label y. We use beam search of size 10 in the generation process.

4. Related work

Variational inference through re-parameterization trick was initially proposed by (Kingma & Welling, 2013; Rezende et al., 2014) and since then, VAE has been widely adopted as generative model for images (Gregor et al., 2015; Yan et al., 2016; Salimans et al., 2015; Gregor et al., 2016).

Our work is in line with previous works on combining variational inferences with text modeling (Bowman et al., 2015; Miao et al., 2016; Serban et al., 2016; Zhang et al., 2016). (Bowman et al., 2015) is the first work to combine VAE with language model and they use LSTM as the decoder and find some negative results. On the other hand, (Miao et al., 2016) models text as bag of words, though improvement has been found, the model can not be used to generate text. Our work fills the gaps between them. (Serban et al., 2016; Zhang et al., 2016) applies variational inference to dialogue modeling and machine translation and found some improvement in terms of generated text quality, but no language modeling results are reported. (Chung et al., 2015; Bayer & Osendorfer, 2014; Fraccaro et al., 2016) embedded variational units in every step of a RNN, which is different from our model in using global latent variables to learn high level features.

Our use of CNN as decoder is inspired by recent success of PixelCNN model for images (van den Oord et al., 2016b), WaveNet for audios (van den Oord et al., 2016a), Video Pixel Network for video modeling (Kalchbrenner et al., 2016b) and ByteNet for machine translation (Kalchbrenner et al., 2016a). But in contrast to those works showing using a very deep architecture leads to better performance, CNN as decoder is used in our model to control the contextual capacity. We find a suitable CNN with VAE can have the best performance.

Our work is closed related the recently proposed variational lossy autoencoder (Chen et al., 2016) which is used to predict image pixels. They find that conditioning on a smaller window of a pixels leads to better results with VAE, which is similar to our finding. Much (Rezende & Mohamed, 2015; Kingma et al., 2016; Chen et al., 2016) has been done to come up more powerful prior/posterior distribution representations with techniques such as normalizing flows. We treat this as one of our future works. This work is largely orthogonal and could be potentially combined with a more effective choice of decoder to yield additional gains.

There are many previous works that explore unsupervised sentence encoding such as skip-thought vectors (Kiros et al., 2015), paragraph vector (Le & Mikolov, 2014) and sequence autoencoder (Dai & Le, 2015). (Dai & Le, 2015) applies the pre-trained model to semi-supervised classification and find significant gains, we use this as the baseline for our semi-supervised VAE.

5. Conclusion

We propose to use dilated CNNs as decoders for VAEs for text modeling. We studied the contextual information and latent representation trade off by varying the decoder contextual capacity through changing CNN architectures. We find with a decoder with a small context window, the VAE is forced to use information from the latent representation. By selecting a suitable decoder, the VAE can perform better than simple LSTM language models. We find a similar trade off between classification accuracy and NLL for semi-supervised VAEs. We show our semi-supervised VAEs perform better than strong baselines with proper decoders are selected. There are several future directions to explore based on our work. The first is to use more sophisticated prior/posterior probability representations such as inverse autoregressive flow to further improve the VAE results. Another direction is to come up with better models for sentiment analysis with VAE since it has shown rather different code structure with topic.
References

Bayer, Justin and Osendorfer, Christian. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 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, Kingma, Diederik P, Salimans, Tim, Duan, Yan, Dhariwal, Prafulla, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

Chung, Junyoung, Kastner, Kyle, Dinh, Laurent, Goel, Kratarth, Courville, Aaron C, and Bengio, Yoshua. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pp. 2980–2988, 2015.

Dai, Andrew M and Le, Quoc V. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pp. 3079–3087, 2015.

Fraccaro, Marco, Sønderby, Søren Kaae, Paquet, Ulrich, and Winther, Ole. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems*, pp. 2199–2207, 2016.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, Rezende, Danilo Jimenez, and Wierstra, Daan. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kalchbrenner, Nal, Espeholt, Lasse, Simonyan, Karen, Oord, Aaron van den, Graves, Alex, and Kavukcuoglu, Koray. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016a.

Kalchbrenner, Nal, Oord, Aaron van den, Simonyan, Karen, Danihelka, Ivo, Vinyals, Oriol, Graves, Alex, and Kavukcuoglu, Koray. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016b.

Kalchbrenner, Nal, Oord, Aaron van den, Simonyan, Karen, Danihelka, Ivo, Vinyals, Oriol, Graves, Alex, and Kavukcuoglu, Koray. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016a.

Kalchbrenner, Nal, Oord, Aaron van den, Simonyan, Karen, Danihelka, Ivo, Vinyals, Oriol, Graves, Alex, and Kavukcuoglu, Koray. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016b.
Rezende, Danilo Jimenez and Mohamed, Shakir. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Salimans, Tim, Kingma, Diederik P, Welling, Max, et al. Markov chain monte carlo and variational inference: Bridging the gap. In *ICML*, volume 37, pp. 1218–1226, 2015.

Serban, Iulian Vlad, Sordoni, Alessandro, Lowe, Ryan, Charlin, Laurent, Pineau, Joelle, Courville, Aaron, and Bengio, Yoshua. A hierarchical latent variable encoder-decoder model for generating dialogues. *arXiv preprint arXiv:1605.06069*, 2016.

Tang, Duyu, Qin, Bing, and Liu, Ting. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, pp. 1422–1432, 2015.

van den Oord, Aaron, Dieleman, Sander, Zen, Heiga, Simonyan, Karen, Vinyals, Oriol, Graves, Alex, Kalchbrenner, Nal, Senior, Andrew, and Kavukcuoglu, Koray. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*, 2016a.

van den Oord, Aaron, Kalchbrenner, Nal, Espeholt, Lasse, Vinyals, Oriol, Graves, Alex, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pp. 4790–4798, 2016b.

Yan, Xinchen, Yang, Jimei, Sohn, Kihyuk, and Lee, Honglak. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pp. 776–791. Springer, 2016.

Yang, Zichao, Yang, Diyi, Dyer, Chris, He, Xiaodong, Smola, Alex, and Hovy, Eduard. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pp. 1480–1489, 2016.

Yu, Fisher and Koltun, Vladlen. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Zhang, Biao, Xiong, Deyi, Su, Jinsong, Duan, Hong, and Zhang, Min. Variational neural machine translation. *arXiv preprint arXiv:1605.07869*, 2016.

Zhang, Xiang, Zhao, Junbo, and LeCun, Yann. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pp. 649–657, 2015.
| Society          | do you think there is a god ? |
|-----------------|--------------------------------|
| Science         | how many orbitals are there in outer space ? how many orbitals are there in the solar system ? |
| Health          | what is the difference between _UNK and _UNK |
| Education       | what is the difference between a computer and a _UNK ? |
| Computers       | how can i make flash mp3 files ? i want to know how to make a flash video so i can upload it to my mp3 player ? |
| Sports          | who is the best soccer player in the world ? |
| Business        | what is the best way to make money online ? |
| Music           | who is the best artist of all time ? |
| Relationships   | how do i know if a guy likes me ? |
| Politics        | what do you think about Iran ? |
| Society         | what is the meaning of life ? |
| Science         | what is the difference between kinetic energy and heat ? |
| Health          | what is the best way to get rid of migrane headaches ? |
| Education       | what is the best way to study for a good future ? |
| Computers       | what is the best way to install windows xp home edition ? |
| Sports          | who do you think will win the super bowl this year ? |
| Business        | i would like to know what is the best way to get a good paying job ? |
| Entertainment   | what do you think is the best movie ever ? |
| Relationships   | what is the best way to get over a broken heart ? |
| Politics        | what do you think about the war in iraq ? |
| Society         | what would you do if you had a million dollars ? |
| Mathematics     | i need help with this math problem ! |
| Health          | what is the best way to lose weight ? |
| Education       | what is the best college in the world ? |
| Computers       | what is the best way to get a new computer ? |
| Sports          | who should i start ? |
| Business        | what is the best way to get a good paying job ? |
| Entertainment   | who do you think is the hottest guy in the world ? |
| Relationships   | what should i do ? |
| Politics        | who do you think will be the next president of the united states ? |
| Society         | do you believe in ghosts ? |
| Science         | why is the sky blue ? |
| Health          | what is the best way to get rid of a cold ? |
| Reference       | what do you do when you are bored ? |
| Computers       | why ca n’t i watch videos on my computer ? when i try to watch videos on my computer , i can’t get it to work on my computer , can anyone help ? |
| Sports          | what do you think about the _UNK game ? |
| Business        | what is the best way to get a job ? |
| Entertainment   | what is your favorite tv show ? |
| Relationships   | how do you know when a guy likes you ? |
| Politics        | what do you think about this ? |
| Society         | what is the name of the prophet muhammad ( pbuh ) ? i do n’t know if he is a jew or not . |
| Science         | where can i find a picture of the _UNK _UNK _UNK _UNK ? i need to know the name of the insect that has the name of the whale . |
| Health          | what is the best way to get rid of a _UNK mole ? |
| Reference       | does anyone know where i can find info on _UNK _UNK _UNK ? i am looking for the name of the _UNK _UNK . |
| Computers       | does anyone know where i can find a picture of a friend ’s cell phone ? |
| Sports          | does anyone know where i can find a biography of _UNK ? |
| Business        | does anyone know where i can find a copy of the _UNK ? |
| Music           | does anyone know the name of the song and who sings it ? |
| Relationship    | how do i tell my boyfriend that i love him ? he is my best friend , but i dont know how to tell him . please help ! ! ! ! ! ! |
| Politics        | where is osama bin laden ? |

Table 7: Text generated by conditioning on topic label.
| Sentiment | Text |
|-----------|------|
| 1 star    | the food is good , but the service is terrible . i have been here three times and each time the service has been horrible . the last time we were there , we had to wait a long time for our food to come out . when we finally got our food , the food was cold and the service was terrible . i will not be back . |
| 2 star    | this place used to be one of my favorite places to eat in the area . |
| 3 star    | i 've been here a few times , and the food has always been good . |
| 4 star    | this is one of my favorite places to eat in the phoenix area . the food is good , and the service is friendly . |
| 5 star    | my husband and i love this place . the food is great , the service is great , and the prices are reasonable . |
| 1 star    | this is the worst hotel i have ever been to . the room was dirty , the bathroom was dirty , and the room was filthy . |
| 2 star    | my husband and i decided to try this place because we had heard good things about it so we decided to give it a try . the service was good , but the food was mediocre at best . |
| 3 star    | we came here on a saturday night with a group of friends . we were seated right away and the service was great . the food was good , but not great . the service was good and the atmosphere was nice . |
| 4 star    | my husband and i came here for brunch on a saturday night . the place was packed so we were able to sit outside on the patio . we had a great view of the bellagio fountains and had a great view of the bellagio fountains . we sat at the bar and had a great view of the bellagio fountains . |
| 5 star    | my husband and i came here for the first time last night and had a great time ! the food was amazing , the service was great , and the atmosphere was perfect . we will be back ! |
| 1 star    | this is the worst place i have ever been to . i will never go back . |
| 2 star    | i was very disappointed with the quality of the food and the service . i will not be returning . |
| 3 star    | this was my first time at this location and i have to say it was a good experience . |
| 4 star    | this is a great place to grab a bite to eat with friends or family . |
| 5 star    | i am so happy to have found a great place to get my nails done . |
| 1 star    | my wife and i have been going to this restaurant for years . the last few times i have been , the service has been terrible . the last time we were there , we had to wait a long time for our food to arrive . the food is good , but not worth the wait . |
| 2 star    | the food is good , but the service leaves something to be desired . |
| 3 star    | i have been here a few times . the food is consistently good , and the service is good . |
| 4 star    | my wife and i have been here a few times . the food is consistently good , and the service is friendly . |
| 5 star    | my husband and i have been coming here for years . the food is consistently good and the service is always great . |
| 1 star    | the food was good but the service was terrible . we had to wait 45 minutes for our food to come out and it was cold . i will not be back . |
| 2 star    | the food was good but the service was terrible . we had a party of 6 and the food took forever to come out . the food was good but not worth the price . |
| 3 star    | the food was good but the service was a little slow . we had to wait a while for our food and it was n’t even busy . |
| 4 star    | i have been here a few times and have never been disappointed . the food was great and the service was great . we will be back . |
| 5 star    | my husband and i have been here a few times and have never been disappointed . the food was great and the service was great . i will definitely be back ! |
| 1 star    | if i could give this place zero stars i would . i do not recommend this place to anyone ! |
| 2 star    | i do n’t know what all the hype is about this place , but i do n’t think i will be back . |
| 3 star    | i do n’t know what all the hype is about this place , but i do n’t think i ’ll be back . |
| 4 star    | i ’ve been here a couple of times and have never been disappointed . the food is fresh , the service is friendly , and the prices are reasonable . |
| 5 star    | this is the best ramen i ’ve ever had in my life , and i ’ve never had a bad meal here ! |
| 1 star    | this is the worst company i have ever dealt with . they do n’t know what they are doing . |
| 2 star    | this is the worst buffet i have ever been to in my life . the food was just ok , nothing to write home about . |
| 3 star    | not a bad place to stay if you ’re looking for a cheap place to stay . |
| 4 star    | this is a great place to stay if you ’re looking for a quick bite . |
| 5 star    | i love this place ! the staff is very friendly and helpful and the price is right ! |

Table 8: Text generated by conditioning on sentiment label.