Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Variational Autoencoder for Semi-Supervised Text Classification

Weidi Xu, Haoze Sun, Chao Deng, Ying Tan
Key Laboratory of Machine Perception (Ministry of Education),
School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, China
wead_hsu@pku.edu.cn, pkucissun@foxmail.com, cdspace678@pku.edu.cn, ytan@pku.edu.cn

Abstract

Although semi-supervised variational autoencoder (*SemiVAE*) works in image classification task, it fails in text classification task if using vanilla LSTM as its decoder. From a perspective of reinforcement learning, it is verified that the decoder's capability to distinguish between different categorical labels is essential. Therefore, *Semi-supervised Sequential Variational Autoencoder* (*SSVAE*) is proposed, which increases the capability by feeding label into its decoder RNN at each time-step. Two specific decoder structures are investigated and both of them are verified to be effective. Besides, in order to reduce the computational complexity in training, a novel optimization method is proposed, which estimates the gradient of the unlabeled objective function by sampling, along with two variance reduction techniques. Experimental results on Large Movie Review Dataset (IMDB) and AG's News corpus show that the proposed approach significantly improves the classification accuracy compared with pure-supervised classifiers, and achieves competitive performance against previous advanced methods. State-of-the-art results can be obtained by integrating other pretraining-based methods.

1 Introduction

Semi-supervised learning is a critical problem in the text classification task due to the fact that the data size nowadays is increasing much faster than before, while only a limited subset of data samples has their corresponding labels. Therefore lots of attention has been drawn from researchers over machine learning and deep learning communities, giving rise to many semi-supervised learning methods (Socher et al. 2013; Dai and Le 2015).

Variational autoencoder is recently proposed by Kingma and Welling; Rezende, Mohamed, and Wierstra, and it has been applied for semi-supervised learning (Kingma et al. 2014; Maaløe et al. 2016), to which we refer as *SemiVAE*. Although it has shown strong performance on image classification task, its application in sequential text classification problem has been out of sight for a long time. Since variational autoencoder has been verified to be effective at extracting global features from sequences (e.g., sentiment, topic and style) (Bowman et al. 2016), it is also promising in the semi-supervised text classification task.

In this paper, *Semi-supervised Sequential Variational Autoencoder* (*SSVAE*) is proposed for semi-supervised sequential text classification. The *SSVAE* consists of a Seq2Seq structure and a sequential classifier. In the Seq2Seq structure, the input sequence is firstly encoded by a recurrent neural network, e.g., LSTM network (Hochreiter and Schmidhuber 1997) and then decoded by another recurrent neural network conditioned on both latent variable and categorical label. However, if the vanilla LSTM network is adopted as the decoder, the *SSVAE* will fail to make use of unlabeled data and result in a poor performance.

The explanation is given by carefully analyzing the gradient of the classifier from a perspective of reinforcement learning (RL), which reveals how the classifier is driven by the decoder using unlabeled data. By comparing the gradient of the classifier w.r.t. unlabeled objective function to REINFORCE algorithm (Williams 1992), we realize that only if the decoder is able to make difference between correct and incorrect categorical labels, can the classifier be reinforced to improve the performance. Vanilla LSTM setting will mislead the decoder to ignore the label input and hence fails in the sequence classification task.

To remedy this problem, the influence of categorical information is increased by feeding label to the decoder RNN at each time step. This minor modification turns out to bring *SSVAE* into effect. Specifically, we made an investigation on two potential conditional LSTM structures. Experimental results on IMDB and AG's News corpus show that their performances are close and both of them are able to outperform pure-supervised learning methods by a large margin. When using only 2.5K labeled IMDB samples, 10.3% classification error can still be obtained, which outperforms supervised LSTM by 7.7%. The better one is able to achieve very competitive results compared with previous advanced methods. Combined with pretraining method, our model can obtain the current best results on IMDB dataset. Although LSTM is utilized as the classifier in this paper, it should be noted that the classifier can be easily replaced by other more powerful models to achieve better results.

In addition, motivated by the aforementioned interpretation, we reduce the computational complexity of *SemiVAE* by estimating the gradient of unlabeled objective function using sampling. In order to reduce the high variance caused by sampling, the baseline method from the RL literature is adopted.

For *SSVAE*, two kinds of baseline methods are studied, with which the training becomes stable.

In summary our main contributions are:

- We make the *SSVAE* effective by using conditional LSTM that receives labels at each step, and give the explanation from the RL perspective. Two plausible conditional LSTMs are investigated.
- We propose an optimization method to reduce the computational complexity of *SSVAE* via sampling. And two different baseline methods are proposed to reduce the optimization variance. By sampling with these baselines, the model can be trained faster without loss of accuracy.
- We demonstrate the performance of our approach by providing competitive results on IMDB dataset and AG's news corpus. Our model is able to achieve very strong performance against current models.

The article is organized as follows. In the next section, we introduce several related works. And then our model is presented in section 3. In section 4, we obtain both quantitative results and qualitative analysis of our models. At last we conclude our paper with a discussion.

2 Preliminaries

2.1 Semi-supervised Variational Inference

Kingma et al. firstly introduced a semi-supervised learning method based on variational inference. The method consists of two objectives for labeled and unlabeled data. Given a labeled data pair (x, y), the evidence lower bound with corresponding latent variable z is:

$$\begin{aligned}\log p_\theta(x,y) \geq \mathbb{E}_{q_\phi(z|x,y)}[\log p_\theta(x|y,z)] + \log p_\theta(y) \\ - D_{KL}(q_\phi(z|x,y)||p(z)) = -\mathcal{L}(x,y)\,,\end{aligned} \tag{1}$$

where the first term is the expectation of the conditional log-likelihood on latent variable z, and the last term is the Kullback-Leibler divergence between the prior distribution $p(z)$ and the learned latent posterior $q_\phi(z|x,y)$.

For the unlabeled data, the unobserved label y is predicted from the inference model with a learnable classifier $q_\phi(y|x)$. The lower bound is hence:

$$\begin{aligned}\log p_\theta(x) &\geq \sum_y q_\phi(y|x)(-\mathcal{L}(x,y)) + \mathcal{H}(q_\phi(y|x)) \\ &= -\mathcal{U}(x)\,.\end{aligned} \tag{2}$$

The objective for entire dataset is now:

$$\begin{aligned}J = \sum_{(x,y)\in S_l} \mathcal{L}(x,y) + \sum_{x\in S_u} \mathcal{U}(x) \\ + \alpha\mathbb{E}_{(x,y)\in S_l}[-\log q_\phi(y|x)]\,,\end{aligned} \tag{3}$$

where S_l and S_u are labeled and unlabeled data set respectively, α is a hyper-parameter of additional classification loss of labeled data.

2.2 Semi-supervised Variational Autoencoder

This semi-supervised learning method can be implemented by variational autoencoder (Kingma and Welling 2014; Rezende, Mohamed, and Wierstra 2014) (*SemiVAE*). The *SemiVAE* is typically composed of three main components: an encoder network, a decoder network and a classifier, corresponding to $q_\phi(z|x,y)$, $p_\theta(x|y,z)$ and $q_\phi(y|x)$.

In the encoder network, each data pair (x, y) is encoded into a soft ellipsoidal region in the latent space, rather than a single point, i.e., the distribution of z is parameterized by a diagonal Gaussian distribution $q_\phi(z|x,y)$:

$$\hat{x} = f_{enc}(x)\,, \tag{4}$$

$$q_\phi(z|x,y) = \mathcal{N}(\mu(\hat{x},y), diag(\sigma^2(\hat{x},y)))\,, \tag{5}$$

$$z \sim q_\phi(z|x,y)\,. \tag{6}$$

The decoder is a conditional generative model that estimates the probability of generating x given latent variable z and categorical label y:

$$p_\theta(x|y,z) = D(x|f_{dec}(y,z))\,, \tag{7}$$

where $f_{dec}(y,z)$ is used to parameterize a distribution D, typically a Bernoulli or Gaussian distribution for image data.

In the applications, $f_{enc}(\cdot)$, $f_{dec}(\cdot)$ and the classifier $q_\phi(y|x)$ can be implemented by various models, e.g., MLP or CNN networks (Maaløe et al. 2016; Yan et al. 2016). Overall, the *SemiVAE* is trained end-to-end with reparameterization trick (Kingma and Welling 2014; Rezende, Mohamed, and Wierstra 2014).

3 Sequential Variational Autoencoder for Semi-supervised Learning

Based on *SemiVAE*, we propose *Semi-supervised Sequential Variational Autoencoder* (*SSVAE*) for semi-supervised sequential text classification, sketched in Fig. 1. In contrast to the implementation for image data, the sequential data is instead modelled by recurrent networks in our model. Concretely, the encoder $f_{enc}(\cdot)$ and the classifier $q_\phi(y|x)$ are replaced by LSTM networks.

3.1 Learning from Unlabeled Data

However, problem occurs if the vanilla LSTM is used for $f_{dec}(\cdot)$ in the decoder, i.e., the latent variable z and the categorical label y are concatenated as the initial state for a standard LSTM network and the words in x are predicted sequentially. With this setting, the resulting performance is poor and the training is very unstable (cf. Sec. 4).

To obtain a theoretical explanation, the gradient of the classifier $q_\phi(y|x;w_c)$, parameterized by w_c, is investigated. Its gradient w.r.t. Equ. 3 consists of three terms:

$$\begin{aligned}\Delta w_c = &\sum_{(x,y)\in S_l} \alpha\nabla_{w_c}\log q_\phi(y|x;w_c) \\ &+ \sum_{x\in S_u} \nabla_{w_c}\mathcal{H}(q_\phi(y|x;w_c)) \\ &+ \sum_{x\in S_u} \mathbb{E}_{q_\phi(y|x;w_c)}[(-\mathcal{L}(x,y))\nabla_{w_c}\log q_\phi(y|x;w_c)]\,.\end{aligned} \tag{8}$$

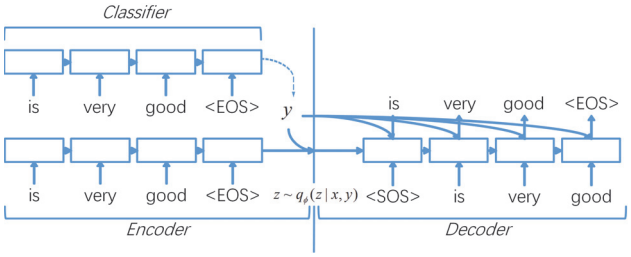

Figure 1: This is the sketch of our model. *Left Bottom*: The sequence is encoded by a recurrent neural network. The encoding and the label y are used to parameterize the the posterior $q_\phi(z|x,y)$. *Right*: A sample z from the posterior $q_\phi(z|x,y)$ and label y are passed to the generative network which estimates the probability $p_\theta(x|y,z)$. *Left Top*: When using unlabeled data, the distribution of y is provided by the sequential classifier (dashed line).

The first term comes from the additional classification loss of labeled data, and the other two terms come from the unlabeled objective function (Equ. 2). The first term is reliable as the gradient is provided by a standard classifier $q_\phi(y|x)$ with labeled data. The second term is a regularization term and it is negligible. The third term, with the summation omitted,

$$\mathbb{E}_{q_\phi(y|x;w_c)}[-\mathcal{L}(x,y)\nabla_{w_c}\log q_\phi(y|x;w_c)]\,, \tag{9}$$

is the expectation of $\nabla_{w_c}\log q_\phi(y|x;w_c)$ on classifier's prediction, multiplied by $-\mathcal{L}(x,y)$. Since the evidence lower bound $-\mathcal{L}(x,y)$ largely determines the magnitude of gradient for each label y, it is supposed to play an important role in utilizing the information of unlabeled data.

To verify this assumption, we investigate the term (Equ. 9) by analogy to REINFORCE algorithm. It adapts the parameters of a stochastic model to maximize the external reward signal which depends on the model's output. Given a policy network $P(a|s;\lambda)$, which gives the probability distribution of action a in current state s, and a reward signal $r(s,a)$, REINFORCE updates the model parameters using the rule:

$$\Delta\lambda \propto \mathbb{E}_{P(a|s;\lambda)}[r(s,a)\nabla_\lambda \log P(a|s;\lambda)], \tag{10}$$

which has the same format with Equ. 9. Comparing Equ. 9 to Equ. 10, the classifier $q_\phi(y|x)$ can be seen as the policy network while the variational autoencoder gives the reward signal $-\mathcal{L}(x,y)$. Actually, the *SemiVAE* can be seen as a generative model with continuous latent variable z and discrete latent variable y, combining both variational autoencoder (Kingma and Welling 2014) and neural variational inference learning (NVIL) (Mnih and Gregor 2014). The whole model is guided by labeled data and reinforced using unlabeled data.

A prerequisite for RL is that the rewards between actions should make difference, i.e., more reward should be given when the agent takes the right action. Similarly, in the *SemiVAE*, only if $-\mathcal{L}(\cdot,y)$ can distinguish between correct and incorrect labels, can the classifier be trained to make better predictions. And since $-\mathcal{L}(x,y)$ is largely determined by the conditional generative probability $p_\theta(x|y,z)$, it requires us to design a conditional generative model that has to be aware of the existence of label y.

In vanilla LSTM setting, the label is fed only at the first time step. And it is found that the model tends to ignore the class feature y, because minimizing the conditional likelihood of each class according to language model (i.e., predicting next word according to a small context window) is the best strategy to optimize the objective function (Equ. 2).

3.2 Conditional LSTM Structures

To remedy this problem, the influence of label is increased by proposing a slight modification to the decoder RNN, i.e., feeding label y at each time step as in (Wen et al. 2015; Ghosh et al. 2016). Although this kind of implementation is simple, it turns out to bring the *SSVAE* into effect.

This paper studies two potential conditional LSTM structures. The first one concatenates word embedding and label vector at each time-step, which is widely used in (Ghosh et al. 2016; Serban et al. 2016). We call this structure CLSTM-I and its corresponding model *SSVAE-I*.

The second conditional LSTM network is motivated by Wen et al.. It is defined by the following equations:

$$i_t = \sigma(W_{wi}w_t + W_{hi}h_{t-1})\,, \tag{11}$$
$$f_t = \sigma(W_{wf}w_t + W_{hf}h_{t-1})\,, \tag{12}$$
$$o_t = \sigma(W_{wo}w_t + W_{ho}h_{t-1})\,, \tag{13}$$
$$\hat{c}_t = \tanh(W_{wc}w_t + W_{hi}h_{t-1})\,, \tag{14}$$
$$c_t = f_t \otimes c_{t-1} + i_t \otimes \hat{c}_t + \tanh(W_{yc}y)\,, \tag{15}$$
$$h_t = o_t \otimes \tanh(c_t)\,, \tag{16}$$

where the equations are the same as in standard LSTM networks except that Equ. 15 has an extra term about y. The label information is directly passed to the memory cell, without the process of four gates in LSTM. This structure is denoted as CLSTM-II and the model with this structure is called *SSVAE-II*.

3.3 Optimizing via Sampling

A limitation of the *SemiVAE* is that they scale linearly in the number of classes in the data sets (Kingma et al. 2014). It is an expensive operation to re-evaluate the generative likelihood for each class during training. Actually, the expectation term in Equ. 9 can be estimated using Monte Carlo sampling from the classifier to reduce the computational complexity of *SemiVAE*.

However, the variance of the sampling-based gradient estimator can be very high due to the scaling of the gradient inside the expectation by a potentially large term. To reduce the variance, the baseline method (Williams 1992; Weaver and Tao 2001), which has been proven to be very efficient for reinforcement learning tasks, is adopted. The baseline can be added without changing the expect gradient. With baseline b introduced, the Equ. 9 is transformed to:

$$\frac{1}{K}\sum_{k=1}^{K}[(-\mathcal{L}(x,y^{(k)}) - b(x))\nabla_{w_c}\log q_\phi(y^{(k)}|x;w_c)]\,, \tag{17}$$

where $y^{(k)} \sim q_\phi(y|x; w_c)$. We investigate two kinds of baseline methods in this paper for *SSVAE*:

- **S1** Since the term $\log p_\theta(x|y,z)$ in $-\mathcal{L}(x,y)$ is approximately proportional to the sentence length, we implement a sequence-length-dependent baseline $b(x) = c|x|$, where $|x|$ stands for the length of input sequence. During the training, the scalar c is learned by minimize MSE $(\log p_\theta(x|y,z)/|x| - c)^2$. In practice, the $\log p_\theta(x|y,z)$ is divided by $|x|$ and use c directly as the baseline to reduce the variance introduced by various sentence lengths.
- **S2** The second one samples $K \geq 2$ labels and simply use the averaged $-\mathcal{L}(x,\cdot)$ as the baseline, i.e., $b(x) = \frac{1}{K}\sum_{k=1}^{K} -\mathcal{L}(x, y^{(k)})$. Although it is a little more computationally expensive, this baseline is more robust.

To avoid confusion, the S1 and S2 tags are used to indicate that the model is trained by sampling using two different baselines, while the *SSVAE-I,II* are two implementations of *SSVAE* using two different conditional LSTMs.

4 Experimental Results and Analysis

The system was implemented using Theano (Bastien et al. 2012; Bergstra et al. 2010) and Lasagne (Dieleman et al. 2015). And the models were trained end-to-end using the ADAM (Kingma and Ba 2015) optimizer with learning rate of 4e-3. The cost annealing trick (Bowman et al. 2016; Kaae Sønderby et al. 2016) was adopted to smooth the training by gradually increasing the weight of KL cost from zero to one. Word dropout (Bowman et al. 2016) technique is also utilized and the rate was scaled from 0.25 to 0.5 in our experiments. Hyper-parameter α was scaled from 1 to 2. We apply both dropout (Srivastava et al. 2014) and batch normalization (Ioffe and Szegedy 2015) to the output of the word embedding projection layer and to the feature vectors that serve as the inputs and outputs to the MLP that precedes the final layer. The classifier was simply modelled by a LSTM network. In all the experiments, we used 512 units for memory cells, 300 units for the input embedding projection layer and 50 units for latent variable z.

4.1 Benchmark Classification

This section will show experimental results on Large Movie Review Dataset (IMDB) (Maas et al. 2011) and AG's News corpus (Zhang, Zhao, and LeCun 2015). The statistic of these two datasets is listed in Table 1. The data set for semi-supervised learning is created by shifting labeled data into unlabeled set. We ensure that all classes are balanced when doing this, i.e., each class has the same number of labeled points. In both datasets we split 20% samples from train set as valid set.

Table 1: The statistic for IMDB and AG's News dataset

Dataset	#labeled	#unlabeled	#testset	#classes
IMDB	25K	50K	25K	2
AG's News	120K	0	7.6k	4

Table 2 and Table 3 show classification results on IMDB and AG's News datasets respectively. The model using vanilla LSTM, referred as *SSVAE-vanilla*, fails to improve the classification performance. In contrast, our models, i.e., *SSVAE-I* and *SSVAE-II*, are able to outperform pure-supervised LSTM by a large margin, which verifies the *SSVAE* as a valid semi-supervised learning method for sequential data. With fewer labeled samples, more improvement can be obtained. When using 2.5K labeled IMDB samples, 10.3% classification error can still be obtained, in contrast to 10.9% error rate using full 20K labeled data for supervised LSTM classifier.

In addition we compare our models with previous state-of-the-art pretraining-based method (Dai and Le 2015). Since their codes have not been published yet, the LM-LSTM and SA-LSTM models were re-implemented. Although the LM-LSTM was successfully reproduced and equivalent performance reported in their paper was achieved, we are unable to reproduce their best results of the SA-LSTM. Therefore, the LM-LSTM was used as a baseline for this comparison. Experimental results show the *SSVAE*s perform worse than LM-LSTM, indicating that pretraining is very helpful in practice, considering the difficulty in optimizing the recurrent networks. Fortunately, since the classifier is separated in *SSVAE*, our method is compatible with pretraining methods. When integrating LM-LSTM, additional improvement can be achieved and the model obtains a tie result to the state-of-the-art result. A summary of previous results on IMDB dataset are listed in Table 4, including both supervised and semi-supervised learning methods. It should be noted that the classifier in our model can be easily replaced with other more powerful methods to get better results. Since only a subset of AG's News corpus is used as labeled data, there is no other comparative results on AG's News corpus.

Table 2: Performance of the methods with different amount of labeled data on IMDB dataset. LM denotes that the classifier is initialized by LM-LSTM.

Method	2.5K	5K	10K	20K
LSTM	17.97%	15.67%	12.99%	10.90%
SSVAE-vanilla	17.76%	15.81%	12.54%	11.86%
SSVAE-I	10.38%	9.93%	9.61%	9.37%
SSVAE-II	**10.28%**	**9.50%**	**9.40%**	**8.72%**
LM-LSTM	9.41%	8.90%	8.45%	7.65%
SSVAE-II,LM	**8.61%**	**8.24%**	**7.98%**	**7.23%**
SSVAE-II,S1	16.87%	15.28%	11.62%	9.75%
SSVAE-II,S1,LM	**9.40%**	**9.00%**	**8.00%**	**7.60%**

4.2 Analysis of Conditional LSTM Structures

From Table 2 and Table 3, the model with CLSTM-II outperforms CLSTM-I slightly. We suppose that CLSTM-II receives label information more directly than CLSTM-I and hence can learn to differentiate various categories much easier. Both of them surpass the model using vanilla LSTM evidently.

To obtain a better understanding of these structures, we investigated the model using vanilla LSTM, CLSTM-I or CLSTM-II as its decoder quantitatively. At first we define the following index for the decoder to explore its relationship

Table 3: Performance of the methods with different amount of labeled data on AG's News dataset.

Method	8K	16K	32K
LSTM	12.74%	10.97%	9.28%
SSVAE-vanilla	12.69%	10.62%	9.49%
SSVAE-I	10.22%	9.32%	8.54%
SSVAE-II	**9.71%**	**9.12%**	**8.30%**
LM-LSTM	9.37%	8.51%	7.99%
SSVAE-II,LM	**8.97%**	**8.33%**	**7.60%**
SSVAE-II,S1	11.92%	10.59%	9.28%
SSVAE-II,S2	9.89%	9.25%	8.49%
SSVAE-II,S1,LM	9.74%	8.92%	8.00%
SSVAE-II,S2,LM	**9.05%**	**8.35%**	**7.68%**

Table 4: Performance of the methods on the IMDB sentiment classification task.

Model	Test error rate
LSTM (2015)	13.50%
LSTM initialize with word2vec (2015)	10.00%
Full+Unlabeled+BoW (2011)	11.11%
WRRBM+BoW (bnc) (2011)	10.77%
NBSVM-bi (2012)	8.78%
seq2-bow*n*-CNN (2015)	7.67%
Paragraph Vectors (2014)	7.42%
LM-LSTM (2015)	7.64%
SA-LSTM (2015)	7.24%
SSVAE-I	9.37%
SSVAE-II	8.72%
SSVAE-II,LM	7.23%

with classification performance:

$$\mathcal{D} = \frac{1}{N_l} \sum_{i=1}^{N_l} 1\{\arg\max_y -\mathcal{L}(x^{(i)}, y) = y^{(i)}\}, \qquad (18)$$

where $(x^{(i)}, y^{(i)})$ is a sample in labeled set, N_l is the number of total labeled data and $-\mathcal{L}(x, y)$ is the lower bound in Equ. 1. This equation denotes the ratio of samples that the decoder can produce higher evidence lower bound of generative likelihood with correct labels, in other words, "how many rewards are given correctly". We use this index to evaluate decoder's discrimination ability. The curves of models using these conditional LSTMs, together with classification accuracy $\mathcal{A}$, are shown in Fig. 2.

By using CLSTMs, the accuracy improves rapidly as well as $\mathcal{D}$ index, which indicates the strong correlation between the accuracy of classifier and discrimination ability of conditional generative model. At the early phase of training, the accuracy of vanilla LSTM improves quickly as well, but diverges at epoch 13. Meanwhile the $\mathcal{D}$ index improves very slowly, indicating that not enough guiding information is provided by the decoder. Therefore, the classifier is unable to utilize the unlabeled data to improve the accuracy, resulting in an unstable performance.

4.3 Sampling with Baseline Methods

Table 2 and 3 also list the results of models trained by sampling, as described in Sec. 3.3. In the implementation, sampling number K is set to 1 when using S1 and 2 for S2. The results of S2 for IMDB dataset are omitted, since the models using S2 are similar with the *SSVAE*s on the datasets with only two classes. The *SSVAE-II* is used as the basic model for this comparison.

Experimental results demonstrate that the sampling-based optimization is made available by using two proposed baselines. However, the models using S1 perform worse than the models without sampling, indicating that the variance is still high even using this baseline. By using S2, the models achieve the performance on par with the *SSVAE*s without sampling, which verifies the S2 as an efficient baseline method for *SSVAE*s. Besides, the time cost using S1 or S2 is less than that without sampling on both IMDB dataset and AG's News corpus (cf. Table 5). For both S1 and S2, the adoption of pre-trained weights makes the optimization more stable during the experiments.

Table 5: Time cost of training 1 epoch using different optimization methods on Nvidia GTX Titan-X GPU.

Method	SSVAE-II	SSVAE-II,S1	SSVAE-II,S2
IMDB,2.5K	4050(s)	2900(s)	-
AG,8K	1880(s)	1070(s)	1205(s)

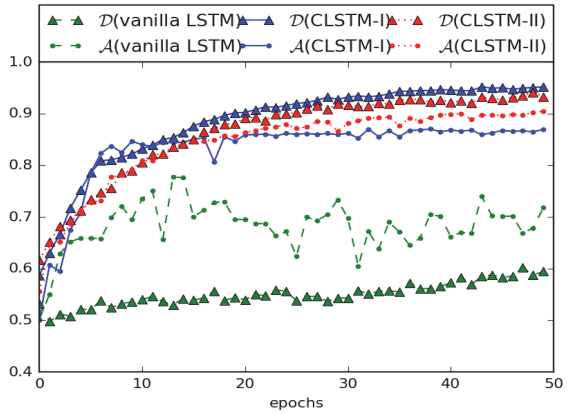

Figure 2: The discrimination index of decoder and classification accuracy between models using vanilla LSTM and conditional LSTMs, with 5K labeled data samples.

4.4 Generating Sentences from Latent Space

To investigate whether the model has utilized the stochastic latent space, we calculated the KL-divergence for each latent variable unit z_i during training, as seen in Fig. 3. This term is zero if the inference model is independent of the input data, i.e., $q_\phi(z_i|x, y) = p(z_i)$, and hence collapsed onto the prior carrying no information about the data. At the end of training process, about 10 out of 50 latent units in our model keep an obviously non-zero value, which may indicate that the latent variable z has propagated certain useful information to the generative model.

To qualitatively study the latent representations, t-SNE (Maaten and Hinton 2008) plots of $z \sim q_\phi(z|x, y)$ from IMDB dataset are seen in Fig. 4. The distribution is Gaussian-like due to its normal prior $p(z)$ and the distribu-

Table 6: Nice generated sentences conditioned on different categorical label y and same latent state z.

Negative	**Positive**
this has to be one of the worst movies I've seen in a long time.	this has to be one of the best movies I've seen in a long time.
what a waste of time ! ! !	what a great movie ! ! !
all i can say is that this is one of the worst movies i have seen.	anyone who wants to see this movie is a must see ! !
UNK is one of the worst movies i've seen in a long time .	UNK is one of my favorite movies of all time.
if you haven't seen this film , don't waste your time ! ! !	if you haven't seen this film , don't miss it ! ! !
suffice to say that the movie is about a group of people who want to see this movie , but this is the only reason why this movie was made in the united states .	suffice to say that this is one of those movies that will appeal to children and adults alike , but this is one of the best movies i have ever seen .

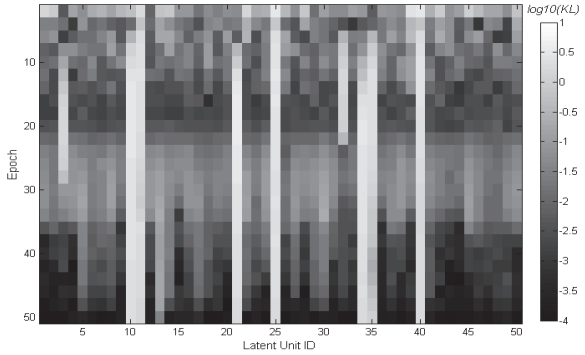

Figure 3: $\log D_{KL}(q_\phi(z|x,y)||p(z))$ for each latent unit is shown at different training epochs. High KL (white) unit carries information about the input text x.

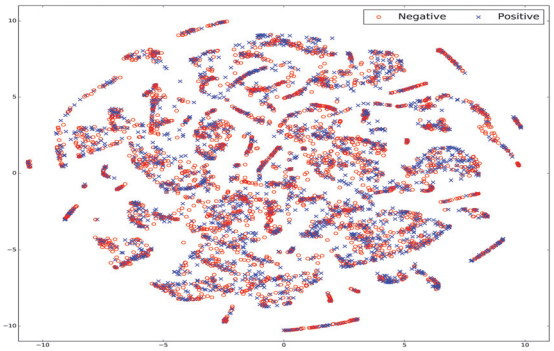

Figure 4: The distribution of IMDB data set in latent space using t-SNE.

tions of two classes are not separable. When digging into some local areas (cf. supplementary materials), it's interesting to discover that sentences sharing similar syntactic and lexical structures are learned to cluster together, which indicates that the shallow semantic context and the categorical information are successfully disentangled.

Another good explorative evaluation of the model's ability to comprehend the data manifold is to evaluate the generative model. We selected several z and generate sentences for IMDB using trained conditional generative model $p_\theta(x|y,z)$. Table 6 demonstrates several cases using the same latent variable z but with opposite sentimental labels. Sentences generated by the same z share a similar syntactic structure and words, but their sentimental implications are much different from each other. The model seems to be able to recognize the frequent sentimental phrases and remember them according to categorical label y. While faced with the difficulty for a model to understand real sentiment implication, it is interesting that some sentences can even express the sentimental information beyond the lexical phrases, e.g., "*but this is the only reason why this movie was made in the United States*". Similar interesting sentences can be also generated on AG's News dataset.

5 Conclusion

The *SSVAE* has been proposed for semi-supervised text classification problem. To explain why *SSVAE* fails if using vanilla LSTM as its decoder, we provided an angle for *SemiVAE* from the perspective of reinforcement learning. Based on this interpretation, the label information is enhanced in the *SSVAE* by feeding labels to the decoder RNN at each time step. This minor modification brings the *SSVAE* into effect. Two specific conditional LSTMs, i.e., CLSTM-I and CLSTM-II, are investigated. Experimental results on IMDB dataset and AG's News corpus demonstrate that our method can achieve competitive performance compared with previous advanced models, and achieve state-of-the-art results by combining pretraining method. In addition, the sampling-based optimization method has been proposed to reduce the computational complexity in training. With the help of the baseline methods suggested in this paper, the model can be trained faster without loss of accuracy.

Acknowledgments

This work was supported by National Key Basic Research Development Plan (973 Plan) Project of China under grant no. 2015CB352302, and partially supported by the Natural Science Foundation of China (NSFC) under grant no. 61375119 and no. 61673025, and Beijing Natural Science Foundation (4162029).

References

Bastien, F.; Lamblin, P.; Pascanu, R.; Bergstra, J.; Goodfellow, I. J.; Bergeron, A.; Bouchard, N.; and Bengio, Y. 2012. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

Bergstra, J.; Breuleux, O.; Bastien, F.; Lamblin, P.; Pascanu, R.; Desjardins, G.; Turian, J.; Warde-Farley, D.; and Bengio, Y. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*. Oral Presentation.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Józefowicz, R.; and Bengio, S. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, 10–21.

Dai, A. M., and Le, Q. V. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, 3061–3069.

Dieleman, S.; Schlüter, J.; Raffel, C.; Olson, E.; Sønderby, S.; Nouri, D.; Maturana, D.; Thoma, M.; Battenberg, E.; Kelly, J.; et al. 2015. Lasagne: First release. *Zenodo: Geneva, Switzerland*.

Ghosh, S.; Vinyals, O.; Strope, B.; Roy, S.; Dean, T.; and Heck, L. 2016. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 448–456.

Johnson, R., and Zhang, T. 2015. Effective use of word order for text categorization with convolutional neural networks. In *NAACL HLT 2015, Denver, Colorado, USA, May 31 - June 5, 2015*, 103–112.

Kaae Sønderby, C.; Raiko, T.; Maaløe, L.; Kaae Sønderby, S.; and Winther, O. 2016. How to train deep variational autoencoders and probabilistic ladder networks. *arXiv preprint arXiv:1602.02282*.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. In *The International Conference on Learning Representations (ICLR)*.

Kingma, D. P.; Mohamed, S.; Rezende, D. J.; and Welling, M. 2014. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, 3581–3589.

Le, Q. V., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 1188–1196.

Maaløe, L.; Sønderby, C. K.; Sønderby, S. K.; and Winther, O. 2016. Auxiliary deep generative models. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, 1445–1453.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *NAACL HLT 2011*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research* 9(Nov):2579–2605.

Mnih, A., and Gregor, K. 2014. Neural variational inference and learning in belief networks. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 1791–1799.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 1278–1286.

Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, 3776–3784.

Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1631, 1642. Citeseer.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

Wang, S., and Manning, C. D. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, 90–94. Association for Computational Linguistics.

Weaver, L., and Tao, N. 2001. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, 538–545. Morgan Kaufmann Publishers Inc.

Wen, T.; Gasic, M.; Mrksic, N.; Su, P.; Vandyke, D.; and Young, S. J. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 1711–1721.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Yan, X.; Yang, J.; Sohn, K.; and Lee, H. 2016. Attribute2image: Conditional image generation from visual attributes. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, 776–791.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 649–657.