# Semisupervised Text Classification by Variational Autoencoder

Weidi Xu and Ying Tan, *Senior Member, IEEE*

***Abstract*— Semisupervised text classification has attracted much attention from the research community. In this paper, a novel model, the semisupervised sequential variational autoencoder (SSVAE), is proposed to tackle this problem. By treating the categorical label of unlabeled data as a discrete latent variable, the proposed model maximizes the variational evidence lower bound of the data likelihood, which implicitly derives the underlying label distribution for the unlabeled data. Analytical work indicates that the autoregressive nature of the sequential model is the crucial issue that renders the vanilla model ineffective. To remedy this, two types of decoders are investigated in the SSVAE model and verified. In addition, a reweighting approach is proposed to circumvent the credit assignment problem that occurs during the reconstruction procedure, which can further improve performance for sparse text data. Experimental results show that our method significantly improves the classification accuracy compared with other modern methods.**

***Index Terms*— Generative models, semisupervised learning, text classification, variational autoencoder (VAE).**

## I. Introduction

RECENTLY, deep neural network models have seen tremendous success in areas such as speech recognition and image classification [2]. The models are able to learn useful abstractions using numerous parameters. However, to fully exploit the ability of deep neural models, huge amounts of labeled data are required during the supervised learning phase, to combat the problem of overfitting. Unfortunately, collecting an extensive amount of labeled data is both expensive and time-consuming in practice. Therefore, semisupervised learning, which aims to make use of unlabeled data to improve the model's performance, has increasingly drawn the attention of researchers.

In this paper, we propose to undertake semisupervised text classification by means of variational autoencoders (VAEs) [3], [4]. VAE is a powerful generative model that takes advantage of deep neural networks. It has been successfully applied in semisupervised image classification tasks [5], [6]. This paper follows the approach in [5], which breaks the latent representation into two parts. For textual data, these two parts correspond to the lexical information and the task-specific categorical information. The model consists of three learnable components: 1) a classifier; 2) an encoder; and 3) a decoder. The encoder and the classifier extract the lexical information and the categorical label from the input sequence, respectively. The decoder is responsible for reconstructing the data conditioned on both lexical feature and categorical label. Applying VAE in semisupervised text classification tasks seems to be straightforward when using the sequential models, e.g., long short-term memory (LSTM) [7] or gated recurrent unit (GRU) networks [8]. However, practical experiments have shown that VAE is ineffective for these tasks if the decoder is implemented by vanilla sequential models [1].

Our solution is the Semisupervised Sequential VAE (SSVAE), which is equipped with a novel decoder structure and method. Among the three components of SSVAE, the structure of the decoder has proven to be the key to properly handle the unlabeled data. The reason is given from the perspective of policy gradient [9], [10] by carefully investigating the VAE's intrinsic working mechanism. Due to the autoregressive nature of the sequential models, the decoder is prone to fitting the data by way of language modeling, i.e., predicting the words with previous ones within the context, regardless of the conditional categorical label. This phenomenon prevents the classifier from obtaining a valid gradient from the decoder. Therefore, in our model, the decoder is forced to perceive the conditional label by feeding the conditional label at each time step. This specific modification improves the VAE model to the point where it is effective for the semisupervised text classification problem. Two decoder structures are proposed, described, and tested empirically.

In addition, motivated by the observation that dependence on the conditional label varies across words in the data, a reweighting (RW) approach is proposed. This approach aims to alleviate the credit assignment problem that occurs during the text reconstruction process. The credit assignment problem concerns the determination of how the success of a systems overall performance results from the various contributions of the systems components [11]. In our method, the credit assignment problem occurs with the unlabeled data. When a portion of the sequence is independent of the categorical label, the classifier will face difficulty in obtaining a valid signal from the reconstruction loss. Specifically, it is unable

Manuscript received July 9, 2018; revised November 5, 2018 and January 6, 2019; accepted February 12, 2019. This work was supported in part by the Natural Science Foundation of China under Grant 61673025 and Grant 61375119, in part by Beijing Natural Science Foundation under Grant 4162029, and in part by the National Key Basic Research Development Plan through the 973 Plan Project of China under Grant 2015CB352302. *(Corresponding author: Ying Tan.)*

The authors are with the Key Laboratory of Machine Perception, Ministry of Education, Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: wead_hsu@pku.edu.cn; ytan@pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2900734

to determine which part is related to the categorical label. We show that by using our RW approach, the gradient of the classifier is theoretically equivalent to or better than the gradient of the classifier without RW. In practice, the RW approach most benefit models with a data set that has a considerable volume of irrelevant content in each sample. With a slight deviance in the terminology, we refer to this type of data as "*sparse*" text data. Furthermore, we present a sampling-based training method. This practice can reduce the training computational complexity by $O(|y|)$, where $|y|$ is the number of label categories. To reduce the gradient variance during training, two variance reduction techniques are employed and investigated in this paper.

The experimental results are obtained on four data sets, two of which contain sparse text samples. These results verify the effectiveness of our method. As an example, our model can achieve 10.28% test error on the Internet Movie Database (IMDB) data set using only 2.5K labeled instances. This outperforms the purely supervised learning model by 7.7%. Moreover, it even surpasses the purely supervised learning trained with 20K labeled samples. We have also found an improvement in results when the classifier is initialized using pretrained parameters. With the help of the sampling-based optimizer, the training speed is accelerated without loss of accuracy, and the proposed RW approach is able to further improve the performance of the sparse text data.

In summary, our main contributions are as follows.

1) We find that it is because the decoder tends to generate the sequence without the conditional categorical label that the VAE model fails for the semisupervised text classification problem.
2) Based on this observation, we make the VAE-based semisupervised method effective for text classification, by remedying the autoregressive problem of the sequential decoder. The proposed modification to the decoder is simple yet effective.
3) The analytical work motivates us to present two improved techniques, i.e., the RW approach and the sampling-based optimizer. The RW approach handles the credit assignment problem of SSVAE, and the sampling-based optimizer is proposed to increase the training efficiency. Experimental results show that both methods are able to further improve the performance of the model.
4) We empirically investigate the performance of SSVAE variants on regular and sparse data sets. Experimental results indicate that our model is able to achieve competitive results on the tested data sets. Finally, conclusions are drawn for future applications of this paper.

## II. Related Work

In the research domain, there is a large body of literature related to semisupervised learning. Here, we will only introduce the relevant topics and terminology.

A popular branch of semisupervised learning is based on regularization [12]–[14]. Faced with a large volume of unlabeled data, it is useless to directly estimate the discriminative probability $p(y|\mathbf{x})$ [15], where $\mathbf{x}$ and $y$ are the input and the output. However, unlabeled data can be used to regularize the feature space and make the prediction more robust. One way to utilize unlabeled data is to assign a target probabilistic label $q(y|\mathbf{x})$ to the unlabeled data $\mathbf{x}$ and train the classification model $p(y|\mathbf{x})$ using these labels. Based on the assumption that neighboring inputs are likely to have the same labels, label propagation methods [16] are proposed, which estimate the target labels by referring to the neighbors in the local region. However, the standard label propagation method is not very effective in text classification problems. This is primarily because the raw text is not semantically continuous by nature. For instance, in the sentiment analysis data set, two sentences may differ by only a single word, yet have contrasting labels.

With the expansion and widespread adoption of deep learning, many deep models have been proposed. One example is the ladder network [17], which uses an encoder–decoder structure to extract the representation feature from unlabeled data. For each layer, a consistency loss is computed between two branches, i.e., a clean encoding branch and a denoised decoding branch. The representation vectors are computed at each layer and the model is trained to minimize their distance. Another approach is the temporal ensembling method [18], which has two noisy branches. The outputs of these two branches are compared as the regularization term. This differs from the ladder network in removing the parametric nonlinearity and the denoising branch. The mean teacher method [19] further extends the temporal ensembling algorithm as it ensembles the models by averaging the network parameters during the training phase. This modification smoothens the target labels among different training epochs. All of these methods have demonstrated strong performance for the task of semisupervised image classification. However, their applications in the domain of text data are absent. The virtual adversarial training [20] algorithm is another efficient semisupervised method. It works to regularize the representation space by competing against the perturbation that alters the prediction maximally. It is compatible with the proposed method because the auxiliary loss can be added to the classifier. Combination of this algorithm with our method may yield better performance than either technique alone, but this is a subject work that is beyond the scope of this paper.

Another important branch of semisupervised learning is based on generative models. These generative models implicitly capture the similarity between the samples by modeling the data distribution. Commonly, a generative model learns the joint probability distribution $p(\mathbf{x}, y)$ for the labeled data and marginal probability distribution $p(\mathbf{x})$ for the unlabeled data, where the label is treated as a latent variable. Typical generative models that have been used in semisupervised learning include naïve Bayes [21], Gaussian mixture model [22], restricted Boltzmann machines [23], and latent Dirichlet allocation [24]. Recently, powerful deep generative models such as VAEs [3], [4] and generative adversarial networks (GANs) [25], [26] have been proposed. The VAE extracts the features for prediction by optimizing the variational lower bound, while the GAN estimates the generative probability via competition between the generative model

and discriminative model. Both VAE and GAN have shown promising performance in the semisupervised image classification tasks [5], [6], [27]–[29].

With respect to the general task of semisupervised text classification problem, many potential techniques have been explored [1], [30]–[33]. A simple yet efficient practice is to pretrain the model by learning to reconstruct the text sequence. The models proposed in [30] and [31] extract valuable features by learning to reproduce the textual data samples using a recursive neural network and recurrent neural network (RNN), respectively. These methods are complementary to our methods, as the classifier of SSVAE is an independent component. The model proposed in [32] is a parallel work to ours, which uses a dilated convolutional neural network (CNN) [34] instead of an RNN as the decoder. Their work focuses on analyzing the effects of utilizing the dilated CNN in VAE. Our work stands out in that we are the first to successfully make effective use of VAE for semisupervised text classification and illustrate that it is the autoregressive nature of the decoder that harms the VAE's performance. We also show the necessity of enhancing the influence of the conditional label in the decoding phase. The TopicRNN [33] model is an unsupervised sequential model for topic modeling. It extracts a global representation vector, rather than explicitly representing the data by two variables. To keep the representation clean, TopicRNN also removes the dependence between the latent variable and the unrelated words by dividing the tokens into the stop words and the others. Although TopicRNN is a general model, it may face difficulty in handling the fine-grained classification problem. The global topic vector extracted in an unsupervised manner may lose the fine-grain information for the subsequent classification task. We will show in the experiments that TopicRNN is less suitable for handling the sparse text data compared to the proposed model.

## III. Background: Variational Inference for Semisupervised Learning

Let us consider a data set consisting of a set of labeled samples $S_l = (\mathbf{X}_l, \mathbf{Y}_l)$ and a set of unlabeled samples $S_u = (\mathbf{X}_u)$, where $(\mathbf{X}_l, \mathbf{Y}_l) = \{(\mathbf{x}_l^{(i)}, y_l^{(i)})\}_{i=1}^{N_l}$ and $\mathbf{X}_u = \{\mathbf{x}_u^{(i)}\}_{i=1}^{N_u}$. For clarity, the superscript and subscript will be omitted if they are unnecessary in the context.

The semisupervised learning method based on VAE was first introduced by Kingma *et al*. [5]. In their work, two solutions were proposed. We follow the solution that explicitly disentangles the latent variable and label information, which is also employed in [32]. In addition to the supervised objective using the labeled data $S_l$, the semisupervised learning method also utilizes the unlabeled samples $S_u$ to improve the classification accuracy. Specifically, there are two objectives for the labeled and unlabeled data. For the labeled data, the objective is to maximize the data log-likelihood, i.e., $\log p_\theta(\mathbf{x}, y)$, whose variational evidence lower bound (ELBO) is

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

where $\mathbf{z}$ is the latent variable and $D_{KL}$ is the Kullback-Leibler divergence.

With respect to the unlabeled data, the objective is to maximize the ELBO of marginal probability $p_\theta(\mathbf{x})$. By regarding the label $y$ as a latent discrete variable, the objective can be given as follows:

$$\log p_\theta(\mathbf{x}) \geq \sum_y q_\phi(y|\mathbf{x})(\mathcal{L}(\mathbf{x}, y)) + \mathcal{H}(q_\phi(y|\mathbf{x})) = \mathcal{U}(\mathbf{x}) \quad (2)$$

where $\mathcal{H}$ is the entropy function.

As mentioned in [5], it is also desirable to add a classification loss using the labeled data. Therefore, the overall objective function to be minimized is

$$J = \sum_{(\mathbf{x},y)\in S_l} -\mathcal{L}(\mathbf{x}, y) + \sum_{\mathbf{x}\in S_u} -\mathcal{U}(\mathbf{x}) + \gamma \sum_{(\mathbf{x},y)\in S_l} -\log q_\phi(y|\mathbf{x}) \quad (3)$$

where $\gamma$ is a hyperparameter that controls the weight of the additional classification loss.

This objective function can be implemented via VAE. VAE achieves the variational inference using deep neural networks. It amortizes the difficulty of variational inference into a parametric neural network by computing the posterior distribution $q_\phi$ of latent variables through a feed-forward network.

The implementation is comprised three parametric components: a classifier, an encoder, and a decoder. They correspond to $q_\phi(y|\mathbf{x})$, $q_\phi(\mathbf{z}|\mathbf{x}, y)$, and $p_\theta(\mathbf{x}|y, \mathbf{z})$. Previous works implement these components using a multilayer perception (MLP) or a CNN for image data [5], [6]. However, practical experiments suggest that it is infeasible to simply use a vanilla RNN for the textual data. Before explaining why vanilla implementation fails, the proposed model is first presented in the following.

## IV. Semisupervised Sequential Variational Autoencoder

In this section, the framework of our model is presented with details concerning implementation and structure. Then, the motivation for the proposed decoder is explained.

### A. Model Framework

The semisupervised sequential variational autoencoder (SSVAE) is proposed for the task of semisupervised text classification. Its framework is sketched in Fig. 1.

In fact, the classifier and the encoder can be implemented by any differential models. The classifier takes the sequence $\mathbf{x}$ as input, and the output is a target vector for the prediction. The encoder takes the same sequence $\mathbf{x}$ as input, as well as the conditional label $y$, to compute the parameters of latent distribution $q_\phi(\mathbf{z}|\mathbf{x}, y)$.

The key argument to note is that a special decoder is required to circumvent the autoregressive nature of the sequential models. In the decoder, the probability $p_\theta(\mathbf{x}|y, \mathbf{z})$ of reproducing the input sequence $\mathbf{x}$ is computed, conditioned on the latent variable $\mathbf{z}$, and the label $y$. The difficulty for SSVAE actually comes about during decoding a sequence using label $y$. The sequential generative models tend to generate words according to a small context in the form of language modeling, which are insensitive to other conditional input. Worse than in

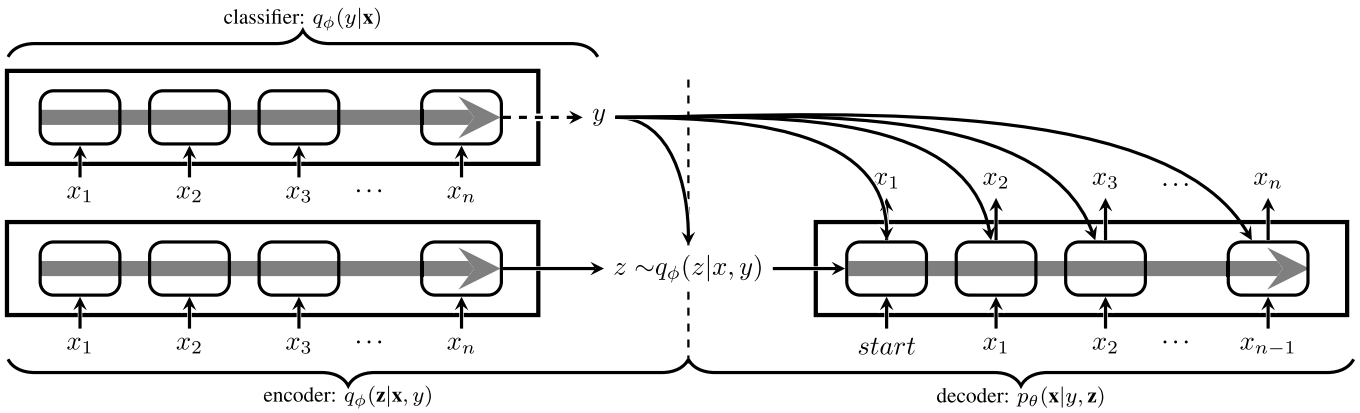


Fig. 1. This is the sketch of our model. The sequence is encoded by a RNN. The encoding and the label $y$ are used to parameterize the posterior $q_\phi(\mathbf{z}|\mathbf{x}, y)$ (bottom left). A sample $z$ from the posterior $q_\phi(\mathbf{z}|\mathbf{x}, y)$ and label $y$ are passed to the generative network, which estimates the probability $p_\theta(\mathbf{x}|y, \mathbf{z})$ (right). When using unlabeled data, the distribution of $y$ is provided by the sequential classifier (dashed line) (top left).

the case of the standard condition generation tasks, the training in SSVAE encounters an additional problem where it maximizes the log-likelihood of all possible labels [cf. (2)]. We will show that when the decoder cannot distinguish between nonidentical labels, minimizing the objective in (2) is not beneficial for improving the classification accuracy. To remedy this problem, we propose to increase the influence of the input label by utilizing a novel decoder. The decoder receives the label at every time-step and has the effect of transforming SSVAE into an effective model.

For the purpose of clarity, each word $x_t$ in a sequence $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$ denotes the word embedding vector whenever it is used as the network input. Otherwise, it represents the word index in the processed vocabulary. Since the size of the vocabulary is extensive, the word embedding matrix is shared by both the classifier and autoencoder to reduce the demand on computational resources.

### B. Classifier: $q_\phi(y|\boldsymbol{x})$

The classifier is used as a probability estimator over the label distribution, i.e., $y \sim q_\phi(y|\mathbf{x})$. As mentioned above, SSVAE functions correctly using a variety of choices for the classifier. Here, the LSTM or GRU network is utilized as the basic classifier unless we explicitly declare the use of another model. The choice of the RNN cell depends on which one is better for the task at hand.

Let $\mathbf{h}_t^c = f_c(x_t, \mathbf{h}_{t-1}^c)$ denote the RNN unit of the classifier mapping input $x_t$ and previous state $\mathbf{h}_{t-1}^c$ to the next state $\mathbf{h}_t^c$. The output at the final time-step is then used for the prediction by concatenating it with a projection layer to compute the distribution $p_\theta(y|\mathbf{x})$

$$\mathbf{h}_t^c = f_c\big(x_t, \mathbf{h}_{t-1}^c\big), \quad t = 1, \ldots, T \tag{4}$$

$$p_\theta(y|\mathbf{x}) = \text{softmax}\big(\mathbf{W}^c\big(\mathbf{h}_T^c\big)\big) \tag{5}$$

where $x_t \in \mathcal{R}^{d_x}$, $\mathbf{h}_t^c \in \mathcal{R}^{d_h}$. Throughout this paper, the notation $\mathbf{b} = \mathbf{W}(\mathbf{a})$ denotes a linear weight matrix with the bias transforming vector $\mathbf{a}$ into the vector $\mathbf{b}$. In practice, we use two layers between the last state $\mathbf{h}_T^c$ and the prediction because it has proven beneficial for improving the test accuracy [31].

### C. Encoder: $q_\phi(\boldsymbol{z}|\boldsymbol{x}, y)$

The encoder is used to map the data instance $(\mathbf{x}, y)$ to the latent variable $\mathbf{z}$, playing the role of $q_\phi(\mathbf{z}|\mathbf{x}, y)$. This can be realized in many ways. For example, the input sequence $\mathbf{x}$ can first be encoded by an LSTM network to compute the representation vector. Let $\mathbf{h}_t^e = f_e(x_t, \mathbf{h}_{t-1}^e)$ denote the encoder LSTM to process the given input $\mathbf{x}_t$ and the previous state $\mathbf{h}_{t-1}^e$. Afterward, the last state $\mathbf{h}_T^e$ is concatenated with the conditional label $y$ to derive the distribution of the stochastic variable $\mathbf{z} \in \mathcal{R}^{d_z}$

$$\mathbf{h}_t^e = f_e\big(x_t, \mathbf{h}_{t-1}^e\big), \quad t = 1, \ldots, T \tag{6}$$

$$\mathbf{z} \sim \mathcal{N}(\mu(\mathbf{x}, y), \quad \text{diag}(\sigma^2(\mathbf{x}, y))) \tag{7}$$

$$\mu(\mathbf{x}, y) = \mathbf{W}_\mu\big(\big[\mathbf{h}_T^e : \mathbf{y}\big]\big) \tag{8}$$

$$\log \sigma(\mathbf{x}, y) = \mathbf{W}_\sigma\big(\big[\mathbf{h}_T^e : \mathbf{y}\big]\big) \tag{9}$$

where $\mathbf{W}_\mu \in \mathcal{R}^{d_z \times (d_h + |y|)}$, $\mathbf{W}_\sigma \in \mathcal{R}^{d_z \times (d_h + |y|)}$, bold $\mathbf{y} \in \mathcal{R}^{|y|}$ is the one-hot vector of $y$ and [:] denotes the vector concatenation operation.

### D. Decoder: $p_\theta(\boldsymbol{x}|y, z)$

The decoder is a sequential conditional generative model, which estimates the probability of generating the sequence $p_\theta(\mathbf{x}|y, \mathbf{z})$ when the sampled latent variable $\mathbf{z}$ and label input $y$ are specified as input. For an input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_T)$, the probability is the product of conditional probabilities

$$p_\theta(\mathbf{x}|y, \mathbf{z}) = \prod_{t=1}^{T} p_\theta(x_t|x_{<t}, y, \mathbf{z}) \tag{10}$$

where a RNN is utilized.[1]

In the common practice of conditional generative model, the conditional input is provided as the initial state. Specifically, the label $y$ and the latent variable $\mathbf{z}$ are fed into a transformation layer. Then, this outputs a vector as the initial

[1]The model is optional. Yang *et al*. adopts the dilated CNN to model the sequential generative probability [32].

state of an LSTM network. Unfortunately, this implementation fails when tested empirically.

For the labeled data, the input $y$ is the ground-truth in the data set. In contrast to the unlabeled data, the label is unknown, and all possible values are iterated when computing the loss in (2). For instance, in a binary classification task, both $p_\theta(\mathbf{x}|y^{\text{pos}}, \mathbf{z})$ and $p_\theta(\mathbf{x}|y^{\text{neg}}, \mathbf{z})$ will be maximized in (2). This leads to a potential problem where the decoder will generate $\mathbf{x}$ on the basis of neighboring words within the context, and it will neglect the conditional input $y$. We will show in Section VI that it is indeed an issue in practice.

To address this problem, it is essential to increase the influence of label $y$, thereby forcing the decoder to recognize and consider the input label. Instead of just using it as the initial state, we supply the decoder with the input label at each time step in the decoder. The decoding process is as follows:

$$\mathbf{h}_0^d = \tanh(\mathbf{W}_d([\mathbf{y} : \mathbf{z}])) \tag{11}$$

$$\mathbf{h}_t^d = f_d\left(x_{t-1}, \mathbf{y}, \mathbf{h}_{t-1}^d\right), \quad t = 1, \ldots, T \tag{12}$$

$$p_\theta(x_t|x_{<t}, y, \mathbf{z}) = \text{softmax}\left(\mathbf{W}_p\left(\mathbf{h}_t^d\right)\right) \tag{13}$$

where $x_0$ is a predefined symbol indicating the start of the sentence, $\mathbf{W}_d \in \mathcal{R}^{(|y|+d_z)\times d_h}$, $\mathbf{W}_p \in \mathcal{R}^{d_h \times |x|}$ ($|x|$ denotes the size of the input vocabulary).

Two conditional LSTM (CLSTM) decoders that implement $f_d$ are investigated.

*1) CLSTM-I:* The first one simply concatenates the $x_t$ and input label vector as the input to the vanilla LSTM network at each time step. This implementation is also used in other conditional generative models [35]–[37]

$$\mathbf{h}_t^d = \text{LSTM}\left([x_{t-1} : \mathbf{y}], \mathbf{h}_{t-1}^d\right). \tag{14}$$

*2) CLSTM-II:* The second conditional LSTM network is motivated by Wen *et al.* [38]. In this implementation, the label is directly fed into the computation of the cell state. $f_d$ is presented by

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_{wi}(x_{t-1}) + \mathbf{W}_{hi}\left(\mathbf{h}_{t-1}^d\right)\right) \tag{15}$$

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_{wf}(x_{t-1}) + \mathbf{W}_{hf}\left(\mathbf{h}_{t-1}^d\right)\right) \tag{16}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_{wo}(x_{t-1}) + \mathbf{W}_{ho}\left(\mathbf{h}_{t-1}^d\right)\right) \tag{17}$$

$$\hat{\mathbf{c}}_t = \tanh\left(\mathbf{W}_{wc}(x_{t-1}) + \mathbf{W}_{hi}\left(\mathbf{h}_{t-1}^d\right)\right) \tag{18}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot c_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t + \tanh(\mathbf{W}_{yc}(\mathbf{y})) \tag{19}$$

$$\mathbf{h}_t^d = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{20}$$

where $\sigma$ is the sigmoid function and $\odot$ is the elementwise multiplication of vectors. This approach is similar to the vanilla LSTM except that the label is given in (19). Unlike CLSTM-I, CLSTM-II perceives the label information in a more straightforward fashion.

It turns out that both structures are valid and their performances vary in regular and sparse data sets. From this point on, we will denote the combination of SSVAE and CLSTM-I as SSVAE-I, and the model using CLSTM-II as SSVAE-II.

### *E. Analysis of SSVAE*

As mentioned above, the decoder structure is specifically designed in SSVAE. Although it is reasonable that $p_\theta(\mathbf{x}|y, \mathbf{z})$ should be modeled accurately, here we aim to provide a more thorough explanation.

The motivation derives from discoveries made during the investigation into the gradient of the classifier. Denote $w_c$ as the parameters in the classifier, and the gradient with regard to $w_c$ of (2) is

$$\Delta w_c = \sum_{x \in S_u} \nabla_{w_c} \mathcal{H}(q_\phi(y|\mathbf{x}; w_c)) + \sum_{x \in S_u} \mathbb{E}_{q_\phi(y|\mathbf{x}; w_c)}[(\mathcal{L}(\mathbf{x}, y)) \nabla_{w_c} \log q_\phi(y|\mathbf{x}; w_c)] \tag{21}$$

where the first term can be regarded as a regularization term to prevent the classifier from overfitting, and the second term has the same format as in the REINFORCE algorithm [9]. Correspondingly, $\mathcal{L}(\mathbf{x}, y)$ is equivalent to the reward signal while $q_\phi(y|\mathbf{x})$ acts as the policy module. This analogy gives us a hint regarding how the classifier is improved by the unlabeled data. That is, for the unlabeled data, the potentially true label probably leads to a higher reconstruction probability $\mathcal{L}(\mathbf{x}, y)$. Because the classifier is trained to maximize the reward $\mathcal{L}(\mathbf{x}, y)$, the potentially true label is given more weight, making it the more likely prediction.

In essence, SSVAE is a generative model containing both the discrete latent variable $y$ and the continuous latent variable $\mathbf{z}$, which combines VAE and neural variational inference learning [39]. More specifically, we give the following theorem.

*Theorem 1:* In SSVAE, if the classifier has infinite capacity and the encoder and decoder are both fixed, for any unlabeled data sample $\mathbf{x}$, the optimal classifier $q_\phi^*(y|\mathbf{x})$ is $q_\phi^*(y|\mathbf{x}) = (\exp \mathcal{L}(\mathbf{x}, y)/\sum_y \exp \mathcal{L}(\mathbf{x}, y))$.

The proof is provided in Appendix A. When the parameters of the autoencoder are fixed, the ELBO of $p_\theta(\mathbf{x}, y)$ remains constant. Theorem 1 states that for any unlabeled sample, the optimal label prediction is defined by the autoencoder. This provides an interpretation for the working mechanism of SSVAE. Since $\mathcal{L}(\mathbf{x}, y)$ is the approximation of $\log p_\theta(\mathbf{x}, y)$, the optimal target prediction can be approximately rewritten as $q_\phi^*(y|\mathbf{x}) = (p_\theta(\mathbf{x}, y)/\sum_y p_\theta(\mathbf{x}, y))$ where $\log p_\theta(\mathbf{x}, y) \approx \mathcal{L}(\mathbf{x}, y)$, which actually corresponds to the E-step in the expectation–maximization algorithm [40].

This also implies that, when $\mathcal{L}(\mathbf{x}, y)$ is estimated inaccurately, the classifier will be misled. When the standard sequential generative model ignores the conditional label, $\mathcal{L}(\mathbf{x}, y)$ becomes independent of $y$, i.e., $\mathcal{L}(\mathbf{x}, y) = \mathcal{L}(\mathbf{x}, \cdot)$. As such, the optimal classifier $q_\phi^*(y|\mathbf{x})$ will converge to predict each category with the same probability $(1/|y|)$. This problem motivates us to propose the novel decoder in SSVAE. By forcing the decoder to recognize the conditional input, the ELBO $\mathcal{L}(\mathbf{x}, y)$ will make adjustments between different possible labels. Consequently, SSVAE functions correctly.

## V. Improvements

The proposed model has been presented in Section III. In this section, we introduce two improved techniques for SSVAE. The RW approach aims to achieve a more robust reconstruction probability, in turn improving the

| |
|---|
| Description: June 15, 2006, 9 am, Wang (male, Tel: 139xxxxxxx, address: Temple Park 1x # 3 unit 20x), reported that a Sun-125 motorcycle was stolen, which was bought in the morning of April 1 (worth 3,500 CNY). |
| Case Category: Motorcycle theft |

Fig. 2. Example of a case record where only words in the color red are informative for the prediction.

test accuracy. In addition, a sampling-based optimizer is introduced to increase the training efficiency.

### A. Reweighting Approach

In SSVAE, the label is supplied at each time step as input to the decoder. This assumes that all of the words in the text are dependent on the label $y$; however, this assumption is not reasonable for the sparse text data sets. For instance, in Fig. 2, the text is a description of a criminal case. The task is to classify the crime by type, e.g., robbery, theft, and so on. In this case, only a small fraction of the text depicts the criminal act. This part is denoted as "relevant" to the classification task while the rest is not.

In this scenario, we assume that each word $x_t$ in the text sample has a corresponding latent binary variable $u_t$ and $u_t = 0$ denotes that $x_t$ is independent of $y$. For these irrelevant words, $p_\theta(x_t|x_{<t}, y, \mathbf{z}) = p_\theta(x_t|x_{<t}, \mathbf{z})$ holds and this requires the decoder to ignore the input labels. However, as the relevance information $\mathbf{u}$ is not present in the data set, we cannot explicitly determine where to discard the input label during the decoding process. This leads us to the credit assignment problem in the RL literature, i.e., to determine what parts the label variable is responsible for.

To alleviate this problem, an RW approach that transfers the issue from the decoder to the classifier is proposed. This method is based on the following observation.

*Theorem 2:* In SSVAE, denote $w_c$ as the parameters of the classifier, and assume that the latent variable $\mathbf{u}$ is given ($u_t = 0$ means that $x_t$ is independent of the label $y$); then, the gradient with regard to $w_c$ of (2) is equivalent to

$$\Delta w_c \sim \nabla_{w_c}\mathcal{H}(q_\phi(y|\mathbf{x}; w_c)) - \nabla_{w_c} D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}, y)||p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(y|\mathbf{x})}\left(\underbrace{\sum_t u_t \log p_\theta(x_t|x_{<t}, y, \mathbf{z})}_{\text{words with } u_t = 0 \text{ are discarded}}\right)\nabla_{w_c}\log q_\phi(y|\mathbf{x}; w_c) \tag{22}$$

where $\mathcal{L}(\mathbf{x}, y)$ is expanded here.

The proof is given in Appendix B. In contrast to (21), the words with $u_t = 0$ are explicitly discarded. This implies that if the relevance information $\mathbf{u}$ is obtained, the generative probability of these independent words can be omitted when calculating the reconstruction loss, and the gradient remains unchanged.

In this paper, we propose using the attention mechanism [41] to obtain the relevance information. A classifier with the attention mechanism can be regarded as a feature extractor that will select informative parts to determine the label $y$. To learn the model end to end, the soft attention method is implemented. By applying the attention mechanism, we can obtain an array of attention weights $\boldsymbol{\alpha}$ during the classification

$$\boldsymbol{\alpha} = f_{\text{att}}(\mathbf{x}) \tag{23}$$

where $f_{\text{att}}$ is the attention function of $q_\phi(y|\mathbf{x})$ and $\alpha_t \in [0, 1]$ is a scalar.

With attention weight $\boldsymbol{\alpha}$ obtained, we modify the third term of (22) by substituting the continuous variable $\boldsymbol{\alpha}$ for the discrete variable $\mathbf{u}$

$$\mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{x})}\left(\sum_t \hat{\alpha}_t \log p_\theta(x_t|x_{<t}, y, \mathbf{z})\right)\nabla_{w_c}\log q_\phi(y|\mathbf{x}) \tag{24}$$

where $\hat{\alpha}_t$ is the normalized weight $\alpha_t / \sum_j \alpha_j$.

Although using a continuous variable instead of $\mathbf{u}$ will introduce bias, it works well empirically. A continuous variable is also capable of representing the relevance and extracting the key content. Generally, the classifier will assign very small attention weights to the parts that are not useful for prediction. The irrelevant content can still be discarded when using soft attention. As long as the classifier is able to correctly focus on the relevant content, the soft version performs similar to the hard attention.

Note that we use $\sum_j \alpha_j$ to normalize the reconstruction loss. This normalization trick does not degrade the classification performance in practice. Averaging over $\alpha$ unifies the scale of loss and stabilizes the training when using a sampling-based optimization method. Consequently, this enables us to use various attention methods, as long as the classifier is able to determine the comparative importance of each word.

By RW the reconstruction loss, the reward signal for the classifier becomes more robust. Rather than having the decoder implicitly infer which part is informative by the decoder, it is explicitly modeled in the reweighted reconstruction loss. The decoder is, therefore, not required to derive a highly accurate probability $p_\theta(\mathbf{x}|y, \mathbf{z})$, and the deviation from the irrelevant parts will be eliminated by multiplying the attention weight. This effectively removes some of the responsibility from the decoder.

### B. Optimization Via Sampling

Optimizing the loss in (2) is computationally expensive because it scales linearly with the number of classes in the data sets [5]. The generative likelihood has to be reevaluated for each class during training. To accelerate the training speed, a sampling-based optimization method is utilized, which reduces the computational complexity by $O(|y|)$ in a single batch.

However, it is well known that sampling-based optimization methods suffer from the high training variance problem. Due to the scaling of the gradient inside the expectation by a potentially large term, the gradient derived from a single sampling will be ineffective. Therefore, the baseline methods, which are widely used in policy gradient methods [9], [10],

are adopted. These baseline methods are able to improve robustness without changing the expected gradient. Here, two types of baseline methods in the RL literature are utilized. Both of them are unbiased and independent of the label $y$. Denote $b(\mathbf{x})$ as the baseline; the second term in (21) can be represented as

$$\frac{1}{K}\sum_{k=1}^{K}[(\mathcal{L}(\mathbf{x}, y^{(k)}) - b(\mathbf{x}))\nabla_{w_c}\log q_\phi(y^{(k)}|\mathbf{x}; w_c)] \quad (25)$$

where $y^{(k)} \sim q_\phi(y|\mathbf{x}; w_c)$.

*1) S1:* The first baseline is a scalar that approximates the averaging $\mathcal{L}(\mathbf{x}, y)$ dynamically. Given that the conditional generative probability $\log p_\theta(\mathbf{x}|y, \mathbf{z})$ is proportional to the sentence length, the $\log p_\theta(\mathbf{x}|y, \mathbf{z})$ is normalized by the length to unify the scale of $\mathcal{L}(\mathbf{x}, y)$. The normalization technique is shown to be effective in stabilizing the training. Specifically, $\mathcal{L}(\mathbf{x}, y)$ in (25) is replaced by $\mathcal{L}'(\mathbf{x}, y)$, where

$$\mathcal{L}'(\mathbf{x}, y) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},y)}[\log p_\theta(\mathbf{x}|y, \mathbf{z})]/|\mathbf{x}| + \log p_\theta(y) - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}, y)||p_\theta(\mathbf{z})) \quad (26)$$

and $|\mathbf{x}|$ denotes the sequence length. Then, the baseline is updated by exponential moving averaging

$$b_t = (1-\rho)\mathcal{L}'(\mathbf{x}, y) + \rho b_{t-1} \quad (27)$$

where $\rho$ is a smoothing coefficient hyperparameter.

*2) S2:* The second method samples several labels from $q_\phi(y|\mathbf{x})$, and the averaging ELBO $(1/K)\sum_{k=1}^{K}\mathcal{L}(\mathbf{x}, y^{(k)})$ is used as the baseline. This method is usually referred to as variational inference for Monte Carlo objectives (VIMCO) [42], and it does not require learning additional parameters to perform variance reduction. Although this baseline is more computationally expensive, it is more robust.

## VI. Experiments

In this section, we present the experimental results of SSVAE, accompanied by an overview of the implementation details. We first show that the SSVAE is effective in improving the test accuracy on four data sets. Then, we thoroughly analyze the effect of the decoder structures, sampling-based optimizer, and RW approach. Finally, the generation ability of the SSVAE is also investigated with qualitative results.

### A. Hyperparameters and Implementation Details

The system is basically implemented using Theano [43]. The ADAM optimizer [44] is adopted in all experiments with a 3e-4 learning rate, 0.9 beta1, 0.999 beta2, and 1e-8 epsilon. In terms of the model regularization, the dropout [45] and batch normalization [46] are used. We apply dropout with a 0.5 rate to the word embedding and the outputs that connect to the MLP layers. The batch normalization, which plays a crucial role during training, is also used "vertically" in all outputs of the MLP layers, i.e., the input of the decoder, the output of the encoder, and the fully connected layer of the classifier. The word embedding is initialized using GloVe [47] pretrained vectors[2] with a dimension of 300. As reported in [48], the weight of the KL term in (1) should be carefully tuned to prevent the classifier from becoming stuck in a local optimum. In our system, the weight of the KL term is set to be $\sigma(75 + e/12)$, where $e$ is the number of epochs. The hyperparameter $\gamma$ in (3) is set to be $1 + N_u/N_l$, where $N_l$ ($N_u$) is the number of labeled (unlabeled) data samples. When using the S1 baseline method, the smoothing coefficient hyperparameter $\rho$ is set to be 0.99. In terms of model structure, we used 512 units for the RNN. The dimension of the latent variable $\mathbf{z}$ is 50. The gradients are clipped by $[-10, 10]$ and the norm of the gradient is constrained by 25 [49]. It is also noteworthy that the cell clip technique is essential when using CLSTM-II, without which the training becomes unstable.

In the following, the S1 and S2 tags are used to indicate that the model (SSVAE-I,II) is trained using the sampling-based optimizer with two variance reduction techniques. The RW tag is used to indicate that the model is trained with the RW approach and has the attention-based classifier adopted. Here, the simple self-attention model (SelfAtt) is used. This model utilizes the attention mechanism on the basis of the RNN states (see the details in Appendix C). SSVAE (SelfAtt) denotes that the SelfAtt is implemented as the classifier.

### B. Data Sets

The benchmarks of focus are text understanding tasks, including sentiment analysis (Large Movie Review data set from the IMDB [50]), text classification (AG's News corpus [51], CaseType data set), and relation classification (Google Extraction data set.[3]) Among these four data sets, the IMDB data set and AG's News data set are regular data sets. Conversely, the samples in the Google Extraction data set and the CaseType data set are sparse. Each data sample in the IMDB data set is a movie review, and the task for the AG's News data set is to categorize the news into one of five topics. The Google Extraction data set consists of 27K records, each of which describes several aspects related to a person. Although it is a relation classification data set, we treat it as a general text classification problem because the entity information is not given in the data. Therefore, it is the model's duty to determine whether the mention of a relation exists within the input text. This data are considered sparse because the relation mention usually corresponds to a single sentence in a data sample. The CaseType data set[4] is a Chinese classification data set, where each sample is a record of an ordinary security case with a manually labeled target (cf., Fig. 2). The record depicts several aspects with only a few words informing the case category. The statistic of these four data sets is provided in Table I.

To verify how the performance of the model varies with different amounts of labeled data, we create several data sets by shifting the labeled data into unlabeled data. When doing this, we ensure that the class distributions are the same in both labeled data and unlabeled data. For the IMDB data set, the text is truncated with a maximum length of 400, while for

[2]http://nlp.stanford.edu/data/glove.640B.300d.zip

[3]https://github.com/google-research-data sets/relation-extraction-corpus

[4]The processed Google Extraction and CaseType data sets are available from https://github.com/wead-hsu/sssp

TABLE I
STATISTIC OF THE DATA SETS

| **Dataset** | IMDB | AG | GE | CT |
|---|---|---|---|---|
| **#labeled** | 20K | 96K | 24K | 1K |
| **#unlabeled** | 50K | 0 | 0 | 4.4K |
| **#validation** | 5K | 24K | 1.4K | 0.7K |
| **#test** | 25K | 7.6k | 1.4K | 0.7K |
| **#classes** | 2 | 4 | 5 | 12 |
| **avg. length** | 269 | 35 | 81 | 58 |
| **#vocab** | 20K | 20K | 20K | 20K |
| **Sparse** | No | No | Yes | Yes |

AG=AG's News, GE=Google Extraction, CT=CaseType

TABLE II
PERFORMANCE OF THE METHODS WITH DIFFERENT AMOUNTS OF LABELED DATA ON THE IMDB DATA SET

| **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%** |

the other three data sets, the maximum sentence length is set to be 200. In the following, we first show the classification accuracy of the SSVAE on these four data sets and then analyze the effect of the components in detail. The results shown reflect the performance of each model at the training step at which the model performs best on the development set.

### C. Benchmark Classification

Table II illustrates the classification accuracy on the IMDB data set. The model using vanilla LSTM, referred to as SSVAE-vanilla, fails to improve the classification performance. In contrast, our models, i.e., SSVAE-I and SSVAE-II, improve the test accuracy over the pure-supervised classifier by a large margin. With fewer instances of labeled data, the improvement is more evident. When using only 2.5K labeled data ($\sim$3% of all training samples), our model is able to achieve a test accuracy of 89.7%. This is a net gain of 7.7% over the pure-supervised LSTM. In fact, it is superior even when the pure-supervised LSTM is trained using 20K labeled instances. Given our results, we feel that SSVAE is an effective semisupervised learning model for text classification tasks.

In addition to the pure-supervised models, we compare our model with LSTM that is initialized by language modeling (LM-LSTM), which is proposed by Dai and Le [31]. The LM-LSTM is an effective semisupervised method for the text classification problem that works by learning the language model to pretrain the classifier. After pretraining, the model can extract useful semantic features and the optimization is easier as a result of the parameters being better initialized [52]. For the IMDB data set, SSVAEs perform slightly worse than the LM-LSTM, which is an indicator that the role of initialization is crucial for the RNN classifier. Recall that in SSVAE, the classifier is an independent component. As such, it is complementary with pretraining-based methods and can take advantage of the benefits offered through pretraining. When the LM-LSTM is used for the parameter initialization, the classification accuracy is further improved. Table III demonstrates the results of AG's News data set. A summary of previous results on the IMDB and AG's News data sets are given in Tables IV and V. The experiments on the AG's News data set were originally conducted with 8/16/32K labeled data in the original paper [1]. Recently, comparable results were reported in the literature [53] with 12K labeled samples. To compare with these results, the performance of the SSVAE with 12K labeled data samples is evaluated and reported in Table V, together with several other supervised results using the full AG's News from [54]. It is worth noting that the classifier in SSVAE is compatible with other more powerful models to possibly obtain better results.

Tables VI and VII demonstrate the results on the Google Extraction, and CaseType data sets, respectively. All of these results demonstrate that the SSVAE shows a significant improvement in test accuracy when compared to baseline methods.

The TopicRNN paper [33] only reported the result of the IMDB data set. Since the code for TopicRNN has not yet been released, we reimplemented it with the same configuration and

TABLE III
PERFORMANCE OF THE METHODS WITH DIFFERENT AMOUNTS OF LABELED DATA ON THE AG'S NEWS DATA SET

| **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%** |
| SelfAtt | 13.24% | 11.21% | 9.50% |
| SSVAE-II(SelfAtt) | 10.33% | 9.55% | 8.37% |
| SSVAE-II(SelfAtt),RW | 10.63% | 9.77% | 8.69% |

TABLE IV
PERFORMANCE OF THE METHODS ON THE IMDB SENTIMENT CLASSIFICATION TASK

| **Model** | **Test error rate** |
|---|---|
| LSTM [31] | 13.50% |
| LSTM initialize with word2vec [31] | 10.00% |
| Full+Unlabeled+BoW [50] | 11.11% |
| WRRBM+BoW (bnc) [50] | 10.77% |
| NBSVM-bi [55] | 8.78% |
| seq2-bow*n*-CNN [56] | 7.67% |
| Paragraph Vectors [57] | 7.42% |
| LM-LSTM [31] | 7.64% |
| SA-LSTM [31] | 7.24% |
| TopicRNN [33] | 6.28% |
| Virtual Adversarial [20] | 5.91% |
| SSVAE-I | 9.37% |
| SSVAE-II | 8.72% |
| SSVAE-II,LM | 7.23% |

TABLE V
PERFORMANCE OF THE METHODS ON THE AG'S NEWS TASK

| Model | Test error rate |
|---|---|
| Standard Co-training (12K labeled) ‡ [53] | 26.5% |
| Reinforced Co-training (12K labeled) ‡ [53] | 16.6% |
| LM-LSTM (12K labeled) § [31] | 8.8% |
| TopicRNN (12K labeled) § [33] | 8.6% |
| Adversarial SSL (12K labeled) [20] ‡ | 8.5% |
| CNN-rand (full 120K)† [58] | 7.8% |
| CNN-static (full 120K)† [58] | 8.6% |
| CNN-non-static (full 120K)† [58] | 7.7% |
| CL-CNN (full 120K)† [59] | 7.7% |
| VD-CNN (full 120K)† [60] | 8.7% |
| Capsule-B (full 120K)† [54] | 7.4% |
| SSVAE-I (12K labeled) | 9.7% |
| SSVAE-II (12K labeled) | 9.4% |
| SSVAE-II,LM (12K labeled) | 8.6% |

‡: results are retrieved from [53].
†: results are retrieved from [54].
§: results are obtained with our implementation.

TABLE VI
PERFORMANCE ON THE GOOGLE EXTRACTION DATA SET

| Method | Ratio of labeled data | | | |
|---|---|---|---|---|
| | 5% | 10% | 20% | 40% |
| LinearSVM | 24.8% | 23.7% | 25.5% | 25.3% |
| Naïve Bayes | 31.0% | 30.7% | 30.1% | 30.5% |
| LSTM | 30.8% | 28.3% | 24.3% | 22.8% |
| GRU | 30.1% | 25.6% | 23.8% | 21.0% |
| TopicRNN § [33] | 20.4% | 19.5% | 19.0% | 18.7% |
| LM-LSTM § [31] | 19.7% | 18.3% | 17.7% | 16.4% |
| SelfAtt | 27.2% | 26.7% | 23.1% | 21.0% |
| SSVAE-I(SelfAtt) | 20.6% | 17.6% | 14.6% | 12.9% |
| SSVAE-II(SelfAtt) | 21.7% | 18.5% | 15.5% | 14.0% |
| SSVAE-I(SelfAtt),RW | **19.0%** | **16.8%** | **14.2%** | **12.8%** |

§: results are obtained with our implementation.

evaluated its performance on other data sets. When comparing with TopicRNN, SSVAE is outperformed on the regular text data set. However, on the sparse text data set, SSVAE becomes superior. Given these results, we suggest that TopicRNN is suitable for coarse-grained text classification problem, e.g., topic modeling. For fine-grained tasks, TopicRNN may suffer from capturing task-related information, because the general representation vector extracted by unsupervised learning is task-agnostic. This conclusion is in line with that when comparing SSVAE with LM-LSTM. On the sparse data sets, LM-LSTM performs worse than SSVAE, indicating that the general features learned from training the language model are not very helpful in the fine-grained classification task. In contrast, SSVAE performs well because it models the joint probability; hence, the features related to the tasks can be captured. Overall, we conclude that, if the data are sparse, SSVAE is preferred. Otherwise, it may be a better choice to extract a global representation vector in an unsupervised manner, e.g., LM-LSTM and TopicRNN.

### *D. Analysis of Conditional LSTM Structures*

The importance of the decoder structure is emphasized in Section IV-D. The comparison between different decoders is

TABLE VII
PERFORMANCE ON THE CASETYPE DATA SET

| Method | Test error rate |
|---|---|
| LinearSVM | 30.7% |
| Naïve Bayes | 31.0% |
| TopicRNN § [33] | 30.0% |
| LSTM | 28.5% |
| LM-LSTM § [31] | 27.8% |
| SSVAE-I | 27.2% |
| SelfAtt | 28.2% |
| SSVAE-I(SelfAtt) | 26.1% |
| SSVAE-I(SelfAtt),RW | **24.4%** |

§: results are obtained with our implementation.

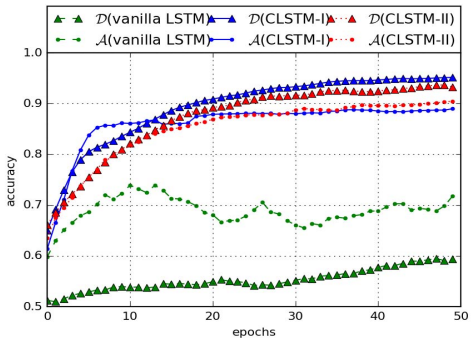


Fig. 3. Classification accuracy and the discrimination index of the decoder between models using vanilla LSTM and conditional LSTMs, with 5K labeled data samples on the IMDB data set.

provided in Tables II, III, and VI. From these results, it is seen that the SSVAEs using the proposed decoders, i.e., SSVAE-I,II, outperforms SSVAE-vanilla remarkably. In fact, when using vanilla LSTM as the decoder, the accuracy curve quickly diverges after several epochs. As aforementioned, this is because the SSVAE-I,II can obtain a more accurate generative probability compared to SSVAE-vanilla. We make the claim that the relative difference of the generative probabilities $\mathcal{L}$ between different given labels is almost identical to the classification accuracy. To verify this, an index about $\mathcal{L}$ is defined to explore the relationship between the classifier and the autoencoder

$$\mathcal{D} = \frac{1}{N_l} \sum_{i=1}^{N_l} \mathbb{1}\{\arg\max_y \mathcal{L}(\mathbf{x}^{(i)}, y) = y^{(i)}\} \tag{28}$$

where $(\mathbf{x}^{(i)}, y^{(i)})$ is a data sample in the training labeled set, $N_l$ is the number of total labeled instances, and $\mathcal{L}(\mathbf{x}, y)$ is defined in (1), $\mathbb{1}$ is the indicator function that takes on a value of 1 if its argument is true, and 0 otherwise ($\mathbb{1}\{\text{True}\} = 1$, $\mathbb{1}\{\text{False}\} = 0$). It can be regarded as the predictive accuracy using the generative probability $p_\theta(\mathbf{x}, y)$ (cf. Theorem. 1). The smoothed curves $\mathcal{D}$ of models using these decoders, together with the test classification accuracy $\mathcal{A}$, are shown in Fig. 3.

When using CLSTMs, the classification accuracy increases consistently with the $\mathcal{D}$ index, which indicates a strong correlation between $q_\phi(y|\mathbf{x})$ and the objective $\mathcal{L}$. During the early training phase, the classification accuracy of vanilla LSTM improves quickly as well. However, it diverges after 13 epochs. Meanwhile, $\mathcal{D}$ improves very slowly, showing that $\mathcal{L}$ is not

TABLE VIII

TIME COST (SECONDS) OF TRAINING 1 EPOCH USING DIFFERENT OPTIMIZATION METHODS ON AN NVIDIA GTX TITAN-X GPU

| Method | SSVAE-II | SSVAE-II,S1 | SSVAE-II,S2 |
|---|---|---|---|
| IMDB,2.5K | 4050 | 2900 (1.40x) | - |
| AG,8K | 1880 | 1070 (1.76x) | 1205 (1.56x) |

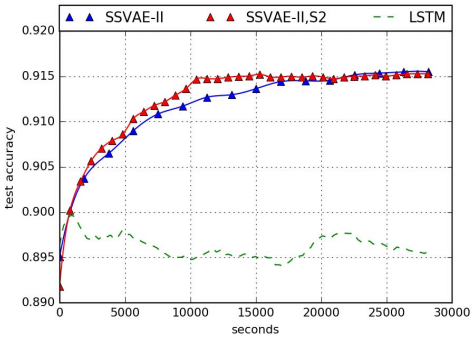


Fig. 4. Test accuracy curves of the SSVAE with or without the sampling-based optimizer with 32K labeled data on the AG's News data set.

discriminative. Therefore, the classifier will be misled by the false signal and fail to properly utilize the unlabeled data.

Both CLSTM-I and CLSTM-II are compatible within the SSVAE and their performances vary over the different types of tasks. In the AG's News data set II, the CLSTM-I is slightly outperformed by the CLSTM-II. In contrast, the CLSTM-I is better for the Google Extraction task VI. Given our observations, we conclude that the appropriate choice of decoder structure depends on the data set used in the experiment. The CLSTM-II receives conditional label information more straightforwardly; hence, it has no option to ignore the given labels for the irrelevant text. Therefore, it is less suitable for sparse text data.

### *E. Sampling-Based Optimizer*

Here, the effectiveness of the sampling-based optimizer, described in Section V-B, is analyzed. Tables II and III also list the results using different optimizers. In the implementation, the number of sampling $K$ is set to 1 when using S1 and 2 for S2. The S2 results in the IMDB data set are omitted since there are only two categories.

In terms of performance, the sampling-based optimizer is able to achieve similar accuracy compared to that without sampling. From the results, we observe that S2 outperforms S1 in all experiments, indicating that the baseline $b(\mathbf{x})$ obtained by S2 is more accurate than S1. Furthermore, S2 is able to achieve accuracy on par with the SSVAEs without using the sampling method, which verifies that S2 is an efficient baseline method for SSVAE. The adoption of pretrained weights has the beneficial effect of stabilizing the training for both S1 and S2. The computational effort using the sampling-based optimizer, measured in terms of time, is less on both the IMDB and the AGs News data sets (cf. Table VIII and Fig. 4).

### *F. Analysis of the Reweighting Approach*

The RW approach is motivated by the credit assignment problem. For the sparse textual data, the dependency on the label $y$ varies across words in the sentence. To tackle this problem, the RW approach aims to eliminate noise from the irrelevant contents by multiplying a relevance coefficient in the reconstruction loss. The RW approach is verified on the Google Extraction and CaseType data sets, shown in Tables VI and VII. To incorporate the RW, the attention-based classifier SelfAtt is utilized as the basic classifier. From the experimental results, we see that SSVAE works well for the sparse text data, and the RW approach is able to further improve the classification accuracy consistently.

To investigate whether the RW approach works on regular text data, we conducted experiments on the AG's News data set, as shown in Table III. The results reflect that the RW approach does not benefit the SSVAE on the AG's News data set. This is reasonable given that the RW approach is designed to resolve the credit assignment problem that occurs only in sparse text data sets. Nonetheless, the impact of using the RW approach in the regular data is negligible. The deterioration of the final accuracy is mainly attributed to the ability of the classifier.

TABLE IX

PREFERRED SETTING FOR SSVAE

| **Dataset Type** | Decoder | Optimizer | RW approach |
|---|---|---|---|
| Regular | CLSTM-II | S2/standard | No |
| Sparse | CLSTM-I | S2/standard | Yes |

A summary of the suggested configuration for SSVAE is listed in Table IX. From the reported results, the choice of components mainly depends on the type of data used in the experiments. A simple way to determine whether a data set is sparse to see whether the simple attention mechanism helps for the classifier.

### *G. Analysis of Latent Space*

To investigate whether the model has utilized the stochastic latent space, KL-divergence is calculated for each latent variable unit $z_i$ during training, as shown in Fig. 5. This term is zero if the inference model is independent of the input data, i.e., $q_\phi(z_i|\mathbf{x}, y) = p(z_i)$, and hence collapsed onto the prior carrying no information about the data. At the end of the training process, approximately 10 out of 50 latent units in our model retain an obviously nonzero value, indicating that the latent variable $\mathbf{z}$ has propagated useful information to the generative model.

To qualitatively study the latent representations, t-SNE [61] plots of $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}, y)$ from the IMDB data set are seen in Fig. 6. The distribution is Gaussian-like due to its normal prior $p(\mathbf{z})$. The distributions of the two classes are not separable, which indicates that the latent variable $\mathbf{z}$ does not carry information for the classification. When digging into some local areas, it is interesting to discover that sentences sharing similar syntactic and lexical structures are clustered together. This suggests that the shallow semantic context and the categorical information are successfully disentangled.

Another good explorative evaluation of the model's capability to comprehend the data manifold is to evaluate the

TABLE X

GOOD GENERATED SENTENCES CONDITIONED ON THE DIFFERENT CATEGORICAL LABEL $y$ AND SAME LATENT STATE $z$ ON THE IMDB DATA SET

| **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 . |

TABLE XI

GOOD GENERATED SENTENCES CONDITIONED ON THE DIFFERENT CATEGORICAL LABEL $y$ AND SAME LATENT STATE $z$ ON THE AG'S NEWS DATA SET

| **World** | **Sports** | **Business** | **Sci/Tech** |
|---|---|---|---|
| JERUSALEM - An Israeli helicopter fired a missile into the Jabalya refugee camp in the northern Gaza Strip on Thursday , witnesses said . | FOXBOROUGH – Peyton Manning threw for 254 yards and two touchdowns as the Indianapolis Colts beat the Tennessee Titans 31-17 . | WASHINGTON - With the economy slowly turning up , upgrading hardware has been on businesses radar in the past 12 months as their number two priority . | MacCentral - Apple Computer Inc. |
| Two bombs exploded at a gathering of Sunni Muslim radicals in central Bangladesh on Saturday , killing at least 10 people and injuring more than 100 . | Two Greek sprinters Kostas Kenteris and Katerina Thanou have been charged with avoiding a drug test on the eve of the Olympics . | Two former executives of Enron Corp. retire. | Two new moons have been discovered orbiting Saturn's moon Titan . |
| A militant group led by al Qaeda ally Abu Musab al-Zarqawi claimed responsibility for an American hostage in Iraq . | A Greek weightlifter has been stripped of his bronze medal in the men's all-around gymnastics competition . | A federal judge has approved Oracle Corp. 's $ 7.7 billion takeover bid for PeopleSoft Inc . | A scientific dig has uncovered four pomegranates believed to be 2,500 years old preserved inside a woven basket nestled in a bronze vessel . |
| Iran agrees to suspend its uranium enrichment programme . | Arsenal defender Sol Campbell will miss the rest of the season after undergoing arthroscopic surgery . | Oracle has extended its $ 7.7 billion hostile tender offer for PeopleSoft Inc . | IBM has agreed to pay $ 92 million to settle a patent infringement lawsuit brought by Eastman Kodak Co . |
| The brother of British hostage Kenneth Bigley has been kidnapped in Iraq . | The Mild Seven Renault F1 Team preview today . | The number of Americans filing first-time claims for unemployment benefits fell last week . | The software juggernaut that conquered the desktop is racing to get your PC . |

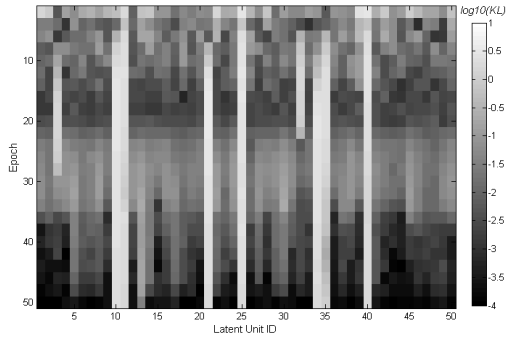


Fig. 5. $\log D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}, y)||p(\mathbf{z}))$ for each latent unit is shown at different training epochs. The high KL (white) unit carries information about the input text $\mathbf{x}$.

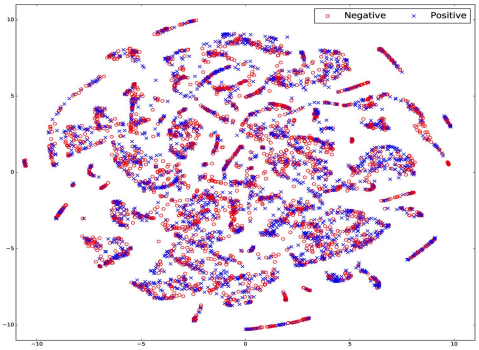


Fig. 6. Distribution of the IMDB data set in latent space $z$ using t-SNE.

generative model. Several $\mathbf{z}$ are selected to generate sentences using the trained conditional generative model $p_\theta(\mathbf{x}|y, \mathbf{z})$. Table X demonstrates several cases using the same latent variable $\mathbf{z}$ but with opposite sentimental labels for IMDB. Sentences generated by the same $\mathbf{z}$ share a similar syntactic structure and dictionary, but their sentimental implications are much different from each other. The model appears to be able to recognize the frequent sentimental phrases and remember them according to the categorical label $y$. While faced with the difficulty of a model understanding complex sentiment implication, it is interesting that some sentences can express 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 also be generated on the AG's News data sets (cf. Table XI).

## VII. CONCLUSION

The SSVAE has been proposed for the semisupervised text classification on the basis of VAE. Both analytical and experimental work has been carried out to explain why a novel structure is necessary to achieve functionality and effectiveness with SSVAE. In particular, the proposed model is validated using four data sets and it is able to achieve competitive results when compared against recent baselines. In addition, two improved techniques are put forward and analyzed: a sampling-based optimizer and an RW approach. The sampling-based optimizer successfully increases the training

efficiency, while the RW approach benefits SSVAE for sparse text data. To complement our model, a usage guide is provided to assist with configuring SSVAE on new tasks.

The effectiveness of SSVAE in semisupervised text classification tasks has been validated. One of our future goals is to continue exploration into other semisupervised natural language processing (NLP) tasks that have more complex output structures. In the future, we expect to extend the field of SSVAE by integrating more information, e.g., syntactic structure and grammar. Our intention is to discover, implement, and present novel and powerful language comprehension tools that will be of use in many domains and provide solutions for both new and existing problems.

## Appendix A
## Proof of Theorem 1

*Proof:* Since the autoencoder is fixed, $\mathcal{L}$ is constant. Therefore, the optimal classifier for the unlabeled data in (2) is

$$\arg\max_{q_\phi(y|\mathbf{x})} \sum_y q_\phi(y|\mathbf{x})\mathcal{L}(\mathbf{x}, y) + \mathcal{H}(q_\phi(y|\mathbf{x})). \tag{29}$$

By the Lagrange multiplication theorem, the problem becomes

$$\min_\lambda \max_{q_\phi(y|\mathbf{x})} \sum_y q_\phi(y|\mathbf{x})\mathcal{L}(\mathbf{x}, y) + \mathcal{H}(q_\phi(y|\mathbf{x})) + \lambda\left(\sum_y q_\phi(y|\mathbf{x}) - 1\right). \tag{30}$$

Let the gradient with regard to $q_\phi(y|\mathbf{x})$ be equal to 0, we then have

$$q_\phi(y|\mathbf{x}) = \exp(-1 - \lambda - \mathcal{L}(\mathbf{x}, y)). \tag{31}$$

Since $\sum_y q_\phi(y|\mathbf{x}) = 1$, $\lambda = \log\sum_y \exp(-1 - \mathcal{L}(\mathbf{x}, y))$. Therefore,

$$q_\phi^*(y|\mathbf{x}) = \exp(\mathcal{L}(\mathbf{x}, y))/\sum_y \exp(\mathcal{L}(\mathbf{x}, y)). \tag{32}$$

□

## Appendix B
## Proof of Theorem 2

*Proof:* We show that the content with $u_t = 0$ can be removed from the reconstruction loss without changing the expected gradient. Recall that $u_i = 0$ denotes that the corresponding input is not relevant to the classification task. We can express the equation by splitting $p(\mathbf{x}|y, \mathbf{z})$ into two parts

$$\begin{aligned}&\mathbb{E}_{q_\phi(y|\mathbf{x})}\left(\sum_t \log p(x_t|x_{<t}, y, \mathbf{z})\right)\nabla_{w_c} q_\phi(y|\mathbf{x}) \\ &= \mathbb{E}_{q_\phi(y|\mathbf{x})}\left(\sum_{u_t=1} \log p(x_t|x_{<t}, y, \mathbf{z})\right)\nabla_{w_c} q_\phi(y|\mathbf{x}) \\ &\quad + \mathbb{E}_{q_\phi(y|\mathbf{x})}\left(\sum_{u_t=0} \log p(x_t|x_{<t}, y, \mathbf{z})\right)\nabla_{w_c} q_\phi(y|\mathbf{x}).\end{aligned} \tag{33}$$

The second part can be subtracted, because

$$\begin{aligned}&\mathbb{E}_{q_\phi(y|\mathbf{x})} b(\mathbf{x}, \mathbf{z})\nabla_{w_c} \log q_\phi(y|\mathbf{x}; w_c) \\ &= \sum_y b(\mathbf{x}, \mathbf{z})\nabla_{w_c} q_\phi(y|\mathbf{x}; w_c) \\ &= b(\mathbf{x}, \mathbf{z})\nabla_{w_c}\sum_y q_\phi(y|\mathbf{x}; w_c) = 0.\end{aligned} \tag{34}$$

where $b(\mathbf{x}, \mathbf{z}) = \sum_{u_t=0} \log p(x_t|x_{<t}, y, \mathbf{z})$. Hence,

$$\begin{aligned}&\mathbb{E}_{q_\phi(y|\mathbf{x})}\left(\sum_t \log p(x_t|x_{<t}, y, \mathbf{z})\right)\nabla_{w_c} \log q_\phi(y|\mathbf{x}) \\ &= \mathbb{E}_{q_\phi(y|\mathbf{x})}\left(\sum_t u_t \log p(x_t|x_{<t}, y, \mathbf{z})\right)\nabla_{w_c} \log q_\phi(y|\mathbf{x}).\end{aligned} \tag{35}$$

□

## Appendix C
## *SelfAtt*: Classifier With the Attention Mechanism

The attention mechanism is typically implemented as per [41]. Similar to the model in [62], a GRU network first takes as input the sequence $\mathbf{x} = \{x_t\}_{t=1}^l$ and outputs a state sequence representing the semantic annotation at each time step

$$\mathbf{h}_t = \text{GRU}(\mathbf{h}_{t-1}, x_t) \tag{36}$$

where $\mathbf{h}_t$ denotes the state at time $t$. As each word contributes differently to the prediction, the context vector $\mathbf{c}$ for the prediction is defined as

$$\mathbf{c} = \sum_t \alpha_t \mathbf{h}_t. \tag{37}$$

The attention weight $\alpha_t$ of each annotation $\mathbf{h}_t$ is computed by

$$\alpha_t = \frac{\exp(f_a(\mathbf{h}_t, \mathbf{q}))}{\sum_j \exp(f_a(\mathbf{h}_j, \mathbf{q}))}, \tag{38}$$

where $\mathbf{q}$ is a learnable vector denoting the classification task, and $f_a$ is a standard attention score function as in [41]. Then, the weighted context $\mathbf{c}$ is supplied as input to a fully connected layer with a softmax function about $y$. This implementation is denoted as SelfAtt.

## Acknowledgment

This paper is an expansion of the publication by Xu *et al.* [1]. It contains additional materials that provide a more comprehensive description, including details on the implementation. In addition, a novel reweighting approach is employed to obtain a more robust gradient for the classifier. Finally, new experiments are added to verify the model's ability to handle different types of data sets. An open-source implementation of our semisupervised text classifier is available at https://github.com/wead-hsu/ssvaehttps://github.com/wead-hsu/ssvae.

## REFERENCES


[1] W. Xu, H. Sun, C. Deng, and Y. Tan, "Variational autoencoder for semi-supervised text classification," in *Proc. 21st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Feb. 2017, pp. 3358–3364. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14299

[2] I. J. Goodfellow, Y. Bengio, and A. C. Courville, *Deep Learning* (Adaptive Computation and Machine Learning Series). Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org/

[3] D. P. Kingma and M. Welling. (2014). "Auto-encoding variational Bayes." [Online]. Available: https://arxiv.org/abs/1312.6114

[4] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. 31th Int. Conf. Mach. Learn. (ICML)*, Beijing, China, Jun. 2014, pp. 1278–1286. [Online]. Available: http://jmlr.org/proceedings/papers/v32/rezende14.html

[5] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3581–3589.

[6] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther, "Auxiliary deep generative models," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, Jun. 2016, pp. 1445–1453. [Online]. Available: http://jmlr.org/proceedings/papers/v48/maaloe16.html

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[8] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proc. 8th Workshop Syntax, Semantics Struct. Stat. Transl. SSST@EMNLP*, Doha, Qatar, Oct. 2014, pp. 103–111. [Online]. Available: http://aclweb.org/anthology/W/W14/W14-4012.pdf

[9] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, 1992.

[10] L. Weaver and N. Tao, "The optimal reward baseline for gradient-based reinforcement learning," in *Proc. 17th Conf. Uncertainty Artif. Intell.* San Mateo, CA, USA: Morgan Kaufmann, 2001, pp. 538–545.

[11] M. Minsky, "Steps toward artificial intelligence," *Proc. IRE*, vol. 49, no. 1, pp. 8–30, Jan. 1961.

[12] Y. Wang and S. Chen, "Safety-aware semi-supervised classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 11, pp. 1763–1772, Nov. 2013.

[13] J. Ortigosa-Hernandez, I. Inza, and J. A. Lozano, "Semisupervised multiclass classification problems with scarcity of labeled data: A theoretical study," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2602–2614, Dec. 2016.

[14] M. Loog and A. C. Jensen, "Semi-supervised nearest mean classification through a constrained log-likelihood," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 995–1006, May 2015.

[15] T. Zhang and F. Oles, "The value of unlabeled data for classification problems," in *Proc. 17th Int. Conf. Mach. Learn.*, Langley, LA, USA, 2000, pp. 1191–1198.

[16] M. Szummer and T. S. Jaakkola, "Partially labeled classification with Markov random walks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, Dec. 2001, pp. 945–952. [Online]. Available: http://papers.nips.cc/paper/1967-partially-labeled-classification-with-markov-random-walks

[17] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3532–3540.

[18] S. Laine and T. Aila. (2016). "Temporal ensembling for semi-supervised learning." [Online]. Available: https://arxiv.org/abs/1610.02242

[19] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Annu. Conf. Neural Inf. Process. Syst. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 1195–1204.

[20] T. Miyato, A. M. Dai, and I. J. Goodfellow. (2016). "Adversarial training methods for semi-supervised text classification." [Online]. Available: https://arxiv.org/abs/1605.07725

[21] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Transferring Naïve Bayes classifiers for text classification," in *Proc. 22nd AAAI Conf. Artif. Intell.*, Vancouver, BC, Canada, Jul. 2007, pp. 540–545. [Online]. Available: http://www.aaai.org/Library/AAAI/2007/aaai07-085.php

[22] F. G. Cozman, I. Cohen, and M. C. Cirelo, "Semi-supervised learning of mixture models," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, Washington, DC, USA, Aug. 2003, pp. 99–106. [Online]. Available: http://www.aaai.org/Library/ICML/2003/icml03-016.php

[23] K. Veselý, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Olomouc, Czech Republic, Dec. 2013, pp. 267–272. doi: 10.1109/ASRU.2013.6707741.

[24] K. Toutanova and M. Johnson, "A Bayesian LDA-based model for semi-supervised part-of-speech tagging," in *Proc. 21st Annu. Conf. Neural Inf. Process. Syst. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2007, pp. 1521–1528. [Online]. Available: http://papers.nips.cc/paper/3317-a-bayesian-lda-based-model-for-semi-supervised-part-of-speech-tagging

[25] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[26] A. Creswell and A. A. Bharath, "Inverting the generator of a generative adversarial network," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–8, 2018. doi: 10.1109/TNNLS.2018.2875194.

[27] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 2226–2234. [Online]. Available: http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans

[28] J. T. Springenberg. (2015). "Unsupervised and semi-supervised learning with categorical generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1511.06390

[29] A. Creswell and A. A. Bharath, "Denoising adversarial autoencoders," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–17, 2018. doi: 10.1109/TNNLS.2018.2852738.

[30] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.

[31] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2015, pp. 3079–3087. [Online]. Available: http://papers.nips.cc/paper/5949-semi-supervised-sequence-learning

[32] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick, "Improved variational autoencoders for text modeling using dilated convolutions," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, Sydney, NSW, Australia, Aug. 2017, pp. 3881–3890. [Online]. Available: http://proceedings.mlr.press/v70/yang17d.html

[33] A. B. Dieng, C. Wang, J. Gao, and J. Paisley. (2016). "TopicRNN: A recurrent neural network with long-range semantic dependency." [Online]. Available: https://arxiv.org/abs/1611.01702

[34] A. van den Oord *et al.*, "WaveNet: A generative model for raw audio," in *Proc. 9th ISCA Speech Synth. Workshop*, Sunnyvale, CA, USA, Sep. 2016, p. 125. [Online]. Available: http://www.isca-speech.org/archive/SSW_2016/abstracts/ssw9_DS-4_van_den_Oord.html

[35] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, and L. Heck. (2016). "Contextual LSTM (CLSTM) models for large scale NLP tasks." [Online]. Available: https://arxiv.org/abs/1602.06291

[36] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. 13th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, Feb. 2016, pp. 3776–3784. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11957

[37] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective LSTMs for target-dependent sentiment classification," in *Proc. 26th Int. Conf. Comput. Linguistics, Conf., Tech. Papers (COLING)*, Osaka, Japan Dec. 2016, pp. 3298–3307. [Online]. Available: http://aclweb.org/anthology/C/C16/C16-1311.pdf

[38] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. J. Young, "Semantically conditioned LSTM-based natural language generation for spoken dialogue systems," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Lisbon, Portugal, Sep. 2015, pp. 1711–1721. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1199.pdf

[39] A. Mnih and K. Gregor, "Neural variational inference and learning in belief networks," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, Beijing, China, Jun. 2014, pp. 1791–1799. [Online]. Available: http://jmlr.org/proceedings/papers/v32/mnih14.html

[40] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.

[41] D. Bahdanau, K. Cho, and Y. Bengio. (2014). "Neural machine translation by jointly learning to align and translate." [Online]. Available: https://arxiv.org/abs/1409.0473

[42] A. Mnih and D. J. Rezende, "Variational inference for Monte Carlo objectives," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, Jun. 2016, pp. 2188–2196. [Online]. Available: http://jmlr.org/proceedings/papers/v48/mnihb16.html

[43] J. Bergstra *et al.*, "Theano: A CPU and GPU math expression compiler," in *Proc. Python Sci. Comput. Conf. (SciPy)*, Jun. 2010, pp. 1–25.

[44] D. P. Kingma and J. Ba. (2015). "Adam: A method for stochastic optimization." [Online]. Available: https//arxiv.org/abs/1412.6980

[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[46] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Lille, France, Jul. 2015, pp. 448–456. [Online]. Available: http://jmlr.org/proceedings/papers/v37/ioffe15.html

[47] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1532–1543. [Online]. Available: http://aclweb.org/anthology/D/D14/D14-1162.pdf

[48] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, and R. Józefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn. (CoNLL)*, Berlin, Germany, Aug. 2016, pp. 10–21. [Online]. Available: http://aclweb.org/anthology/K/K16/K16-1002.pdf

[49] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 3104–3112. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks

[50] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. NAACL HLT*. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 142–150.

[51] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2015, pp. 649–657. [Online]. Available: http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification

[52] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Feb. 2010. doi: 10.1145/1756006.1756025.

[53] J. Wu, L. Li, and W. Y. Wang. (2018). "Reinforced co-training." [Online]. Available: https://arxiv.org/abs/1804.06035

[54] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, and Z. Zhao. (2018). "Investigating capsule networks with dynamic routing for text classification." [Online]. Available: https://arxiv.org/abs/1804.00538

[55] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2012, pp. 90–94.

[56] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," in *Proc. NAACL HLT*, Denver, CO, USA, May/Jun. 2015, pp. 103–112.

[57] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31th Int. Conf. Mach. Learn. (ICML)*, Beijing, China, Jun. 2014, pp. 1188–1196.

[58] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1746–1751. [Online]. Available: http://aclweb.org/anthology/D/D14/D14-1181.pdf

[59] A. Conneau, H. Schwenk, L. Barrault, and Y. LeCun, "Very deep convolutional networks for text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics (EACL)*, Valencia, Spain, vol. 1, Apr. 2017, pp. 1107–1116. [Online]. Available: https://aclanthology.info/papers/E17-1104/e17-1104

[60] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2015, pp. 649–657. [Online]. Available: http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification

[61] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[62] Z. Lin *et al.* (2017). "A structured self-attentive sentence embedding." [Online]. Available: https://arxiv.org/abs/1703.03130

**Weidi Xu** was born in 1992. He received the B.S. degree from the School of Software Engineering, South China University of Technology, Guangzhou, China. He is currently pursuing the Ph.D. degree with the School of Electronics Engineering and Computer Science, Peking University, Beijing, China.

His current research interests include semisupervised learning and deep generative models, and their applications in NLP problems, such as text classification, tagging, sequence generation, and parsing.

**Ying Tan** (SM'02) received the B.Eng., M.S., and Ph.D. degrees from Southeast University, Nanjing, China, in 1985, 1988, and 1997, respectively.

He is a Full Professor and the Ph.D. Advisor with the Electrical Engineering and Computer Science Department and the Director of Computational Intelligence Laboratory, Peking University, Beijing, China, and a Guest Professor with Kyushu University, Fukuoka, Japan. His current research interests include swarm intelligence, machine learning, data mining and their applications, computational intelligence, data mining, and intelligent information processing for information security. He has authored or co-authored more than 280 papers in refereed journals and conferences in these areas, 11 books, and 16 chapters in the book. He holds four invention patents.

Dr. Tan is a member of the Emergent Technologies Technical Committee of IEEE Computational Intelligence Society since 2010. He is the Founder and the Chair of the ICSI International Conference Series. He was the General Chair of Joint General Chair of First/Second BRICS Congress of Computational Intelligence and the Program Committee Co-Chair of IEEE WCCI 2014. He was a recipient of the Second-Class Natural Science Award of China in 2009. He serves as the Editor-in-Chief for the *International Journal of Computational Intelligence and Pattern Recognition*, the Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and so on. He also served as an Editor of Springers *Lecture Notes on Computer Science* for more than 28 volumes and a Guest Editor of several referred journals, including IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, *Information Science*, *Neurocomputing*, *Natural Computing*, *Softcomputing*, and so on.