A Topic Augmented Text Generation Model: Joint Learning of Semantics and Structural Features

Hongyin Tang[1,2], **Miao Li**[1,2], **Beihong Jin**[1,2,*]
[1]State Key Laboratory of Computer Sciences
Institute of Software, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences, Beijing China
{tanghongyin14,limiao17}@otcaix.iscas.ac.cn
Beihong@iscas.ac.cn

Abstract

Text generation is among the most fundamental tasks in natural language processing. In this paper, we propose a text generation model that learns semantics and structural features simultaneously. This model captures structural features by a sequential variational autoencoder component and leverages a topic modeling component based on a Gaussian distribution to enhance the recognition of text semantics. To make the reconstructed text more coherent to the topics, the model further adapts the encoder of the topic modeling component for a discriminator. The results of experiments over several datasets demonstrate that our model outperforms several states of the art models in terms of text perplexity and topic coherence. Moreover, the latent representations learned by our model is superior to others in a text classification task. Finally, given the input texts, our model can generate meaningful texts which hold similar structures but under different topics.

1 Introduction

Text generation is a fundamental task in natural language processing (NLP). Existing methods for text generation are mostly limited in supervised setting and designed for specific applications (e.g., machine translation (Bahdanau et al., 2015), text summarization (Rush et al., 2015)). Several research work (Yu et al., 2017; Zhang et al., 2016) attempts generic text generation based on deep generative models (e.g., GAN, VAE). However, models based on GAN(Generative Adversarial Network) are not able to generate explicit latent codes with salient features of texts. Different from GAN-based models, the VAE (Variational Autoencoder) and its variants can get latent codes of texts with reconstructing texts by its decoder, where it is assumed that the generation process is controlled by codes in a continuous latent space. A natural way to implement the VAEs is to adopt autoregressive networks (e.g., RNNs) as the decoder and encoder. This kind of implementation of VAEs considers sequential information of texts, and is able to model the linguistic structure of texts. But it is not good at modeling semantics (Dieng et al., 2017) and may cause the latent variable collapse problem (Bowman et al., 2016) which leads that the decoder ignores information from the inferred latent codes.

In general, texts inherently contain semantic features and structural features. Only when the latent codes of texts contain structural and semantic information can high-quality texts be generated from the codes. As far as the standard VAE is concerned, it assumes that the latent code is Gaussian distributed so that we cannot distinguish which part of code controls the structure and which part controls the semantics. In other words, it is difficult to generate effective texts by controlling text features directly.

To generate high-quality texts (i.e., with semantic and structural information), we propose a model named TATGM which adopts a sequential VAE to learn structural features of texts and builds a topic modeling component which extracts semantic features of texts. The main contributions of this paper are summarized as follows:

- TATGM can capture the semantics of texts by introducing the topic modeling component. The topic modeling component generates words of texts based on a Gaussian distribution which enables us to take full advantage of information shared by the word embeddings. Moreover, the encoder of the topic modeling component is served as a discriminator to force the decoder of the sequence modeling component to generate texts having the semantics as close to the original texts as

possible. Specifically, no extra training supervised by labeled data is needed for this discriminator.

- The latent code learned by TATGM is a concatenation of the latent variables of the topic modeling component and the sequence modeling component. By two separate parts of the code, TATGM can control structural and semantic information independently. Thus, we can get texts with the changed semantics or structure by changing one part of the code. One interesting practice is that we can get question-answering pairs with the same structure in different semantic spaces.

- The modeling ability of TATGM is evaluated by extensive experiments in terms of perplexity and the coherence of topics. Furthermore, to verify whether TATGM can learn salient features, a classification task using the latent codes is conducted. Experimental results show that TATGM achieves the best performance while compared with many existing models. Finally, several texts generated by TATGM are demonstrated, which indicates that TATGM can generate different expressions of texts of the same structure in different topics.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 describes the model TATGM in detail. Section 4 gives an experimental evaluation. Finally, the paper is concluded in Section 5.

2 Related Work

VAEs (Kingma and Welling, 2014) are a type of deep generative models, which learns explicit latent codes of input data in a continuous latent space and generates data with decoders. VAEs can extract the features of input texts by posterior inference with neural networks and reparameterization tricks. VAE-based models have been widely applied in image generation (Gregor et al., 2015), machine translation (Zhang et al., 2016), and knowledge graph reasoning (Zhang et al., 2018). VAE-based models can also be applied in different tasks of text processing, where the input document is treated as its bag of words (Miao et al., 2016; Srivastava and Sutton, 2017; Miao et al., 2017) or a sequential text (Bowman et al., 2016). The former type of models is always related to topics. For example, Miao et al. (2016) builds the connection by letting weights of the softmax decoder to indicate topic interpretability. Although these models overlook sequential information in texts, they can learn effective latent codes with global semantics. On the other hands, VAE-based models with the input of sequences usually adopt RNNs as encoders or decoders. However, latent variable collapse (i.e., the latent variable collapses to obey the distribution of the prior) makes the sequential VAE to generate texts without information of latent code so that the latent code cannot have enough information of text features.

To avoid the latent variable collapse problem, two types of methods are adopted. The first type tackles this problem by weakening the context modeling ability of decoders mostly in the model architecture level. For example, Bowman et al. (2016) adopts word dropout when feeding the decoder and Yang et al. (2017); Semeniuta et al. (2017) employ non-autoregressive networks (e.g., convolutional neural networks) in the decoder. The second type is to modify the original objective. For example, Bowman et al. (2016) adopts *KL* annealing which can be seen as gradually adding annealing weight into the objective function. Zhao et al. (2018) introduces an additional mutual information term to compensate for the objective function. Xiao et al. (2018) and Zhao et al. (2017) introduce the bag-of-words loss as an auxiliary loss, which measures how well predictions of words can be made from the latent codes. The ideas behind all these methods are same, i.e., to force the models to generate texts from the latent codes so as to make latent codes have more information on text features.

Improving the sequential model by incorporating topics have been explored (Lau et al., 2017; Wang et al., 2018; Dieng et al., 2017; Mikolov and Zweig, 2012). But these models do not have a code for the text structure. Taking TGVAE (Wang et al., 2018) as an example, it guides the generation of the VAE latent code by topic distribution. However, it does not separate the semantic and structure latent codes explicitly. In addition, the models mentioned above use a multinominal LDA which cuts off the possibility of leveraging the semantics in embeddings. As a remedy, Das et al. (2015) and Hu et al. (2012)

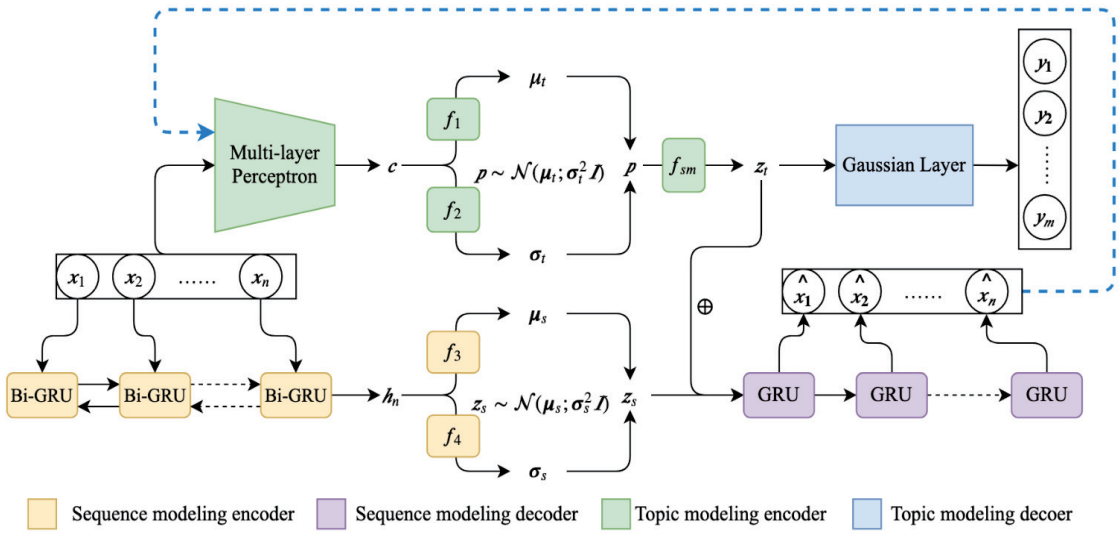

Figure 1: The holistic structure of TATGM. The dotted arrow denotes that the texts which are generated by the sequence modeling decoder are fed into the topic modeling encoder as the input of the discriminator.

propose to adopt a Gaussian-based topic model which assumes each word is generated by a Gaussian distribution. However, their learning algorithms are based on sampling and variational inference, which cannot be assembled in an end-to-end mode.

Hu et al. (2017) proposes a text generation model based on VAE, which aims at generating sentences with controllable styles by learning disentangled latent representations. It feeds generated texts into a discriminator to preserve that these texts have the given style and optimizes the generator with the signals backpropagated from the discriminator. Tian et al. (2018) uses a classifier which is trained with small datasets as the discriminator. Yang et al. (2018) adopts a language model as the discriminator to control the style transfer of generated texts.

Different from existing work, our model combines a sequence modeling component and a topic modeling component so that the final latent code of our model contains both topic semantics and sequential structure of texts. In particular, our model can learn these two features simultaneously, since AutoEncoding Variation Bayes (AEVB) are employed to train these two components. Moreover, our models generate bags-of-words text representations from the latent codes, which avoids the latent variable collapse problem.

3 Model

TATGM is essentially a hybrid autoencoder which comprises a topic modeling component capturing topic information and a sequence modeling capturing structural features. The topic modeling component captures topic information through a Gaussian-based topic model while the sequence modeling component integrates the topic information and generates text. As shown in Fig. 1, the input texts are fed into the two components simultaneously. Each component contains an encoder and decoder, which are labeled by its own color.

3.1 Neural Topic Modeling Component

Similar to existing topic modeling methods, we treat bag-of-words representations of texts as input. However, since the encoder in the topic modeling component is expected to be as a discriminator and guarantee that texts generated by the sequence modeling decoder have specific topic information, the encoder cannot be with the discrete representations (e.g., one-hot representation) of texts as input. It is because such discrete representations of texts cannot be compatible with backward gradients from the discriminator. Therefore, we employ the embeddings of words as input, considering that the word embedding space is continuous and the similarity between every two words can be calculated by the Euclidean distance. Further, we assume that there are K topics in the corpus and each topic is represented as a multivariable Gaussian distribution with a mean and a variance (e.g., $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2 \boldsymbol{I})$ for a given topic k).

In our neural topic modeling component, a document with n words is represented as $\boldsymbol{x} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ where $\boldsymbol{x}_i \in \mathbb{R}^e$ denotes the embedding

of i-th word of the document. The generative process of the document is as follows.

1. For each document, draw the document-topic distribution $\boldsymbol{z}_t \sim Dir(\alpha)$.
2. For i-th word in the document.
   (a) Draw topic assignment $t_i \sim Cat(\boldsymbol{z}_t)$
   (b) Draw the word $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_{t_i}, \boldsymbol{\sigma}_{t_i}^2 \boldsymbol{I})$

According to the generative process above and the marginalization of t_i, the likelihood $p(\boldsymbol{x})$ of the document $\boldsymbol{x}$ can be derived as

$$\begin{aligned} &\int_{\boldsymbol{z}_t} p(\boldsymbol{z}_t|\alpha) \left(\prod_{i=1}^{n} \sum_{t_i} p(\boldsymbol{x}_i|\boldsymbol{\mu}_{t_i}, \boldsymbol{\sigma}_{t_i}^2 \boldsymbol{I}) p(t_i|\boldsymbol{z}_t) \right) d\boldsymbol{z}_t \\ &= \int_{\boldsymbol{z}_t} p(\boldsymbol{z}_t|\alpha) \left(\prod_{i=1}^{n} p(\boldsymbol{x}_i|\boldsymbol{\mu}, \boldsymbol{\sigma}^2 \boldsymbol{I}, \boldsymbol{z}_t) \right) d\boldsymbol{z}_t \end{aligned} \tag{1}$$

where $\boldsymbol{\mu} = \{\boldsymbol{\mu}\}_{k=1}^{K}$, $\boldsymbol{\sigma} = \{\boldsymbol{\sigma}\}_{k=1}^{K}$. Further, we adopt AEVB and the reparameterization trick to achieve posterior inference and parameter learning. Specifically, similar to (Srivastava and Sutton, 2017), we first draw a Gaussian random vector by a reparameterization trick and then pass it through a softmax function to parameterize the multinomial document topic distributions. Therefore, we can replace α with the parameters of Gaussian prior $\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2$.

Thus, we can get the ELBO of our topic modeling component.

$$\begin{aligned} \mathcal{L}_T =& \mathbb{E}_{q(\boldsymbol{z}_t|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z}_t)] \\ &- KL(q(\boldsymbol{z}_t|\boldsymbol{x})||p(\boldsymbol{z}_t|\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2 \boldsymbol{I})) \end{aligned} \tag{2}$$

In Eq. 2, $\boldsymbol{z}_t$ is inferred by the neural network in the encoder. The inference process is detailed as follows. Document $\boldsymbol{x}$ is fed into a two-layer perceptron with *ReLU* as the activation function. The transformation results of the MLP are then processed by max-pooling, getting fixed-length representation $\boldsymbol{c}$ of the document. Parameters of the posterior $\boldsymbol{\mu}_t$ and $\boldsymbol{\sigma}_t^2$ are obtained from two feedforward neural networks. Next, a Gaussian variable is sampled by the reparameterization trick. Finally, $\boldsymbol{z}_t$ is obtained by passing $\boldsymbol{p}$ to the softmax function. Our choices are specified as follows.

$$\begin{aligned} &\boldsymbol{c} = Pooling(MLP(\boldsymbol{x})) \\ &\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t = f_1(\boldsymbol{c}), f_2(\boldsymbol{c}) \\ &\boldsymbol{p} \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2 \boldsymbol{I}) \in \mathbb{R}^K \\ &\boldsymbol{z}_t = f_{sm}(\boldsymbol{W}_t \boldsymbol{p}) \end{aligned}$$

where f_1, f_2 denote two feedforward neural networks, $\boldsymbol{W}_t$ is a trainable parameter, f_{sm} denotes softmax function.

Besides, $KL(q(\boldsymbol{z}_t|\boldsymbol{x})||p(\boldsymbol{z}_t|\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2 \boldsymbol{I}))$ in Eq. 2 is obtained by calculating the *KL* divergence between $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2 \boldsymbol{I})$ and $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2 \boldsymbol{I})$.

In general, during the training of topic models, a smaller vocabulary is built after eliminating some specific words (e.g., stop words, frequent words, and rare words) in the corpus. This is a denoising preprocessing step that makes the model more reliable. However, to make the execution of the discriminator applicable, our model takes all of the words in the document as input while outputting the words in a smaller vocabulary as a topic model does. Besides the reason above, this setting can also be regarded as a regulation of the model which strengthen the ability of modeling documents.

Supposing we have a smaller vocabulary, the document represented by this vocabulary is $\boldsymbol{y} = \{\boldsymbol{y}_i\}_{i=1}^{m}$, where $m < n$ is the number of words and $\boldsymbol{y}_i \in \mathbb{R}^e$ is the word embedding of the word at position i.

The topic modeling decoder reconstructs the document from $\boldsymbol{z}_t$. We assume that the distribution of every topic is a Gaussian with identity variance matrix. Therefore, $p(\boldsymbol{x}|\boldsymbol{z}_t)$ of Eq. 2 is indeed $p(\boldsymbol{y}|\boldsymbol{z}_t)$ and can be expanded as:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{z}_t) &= \prod_{i=1}^{m} p(\boldsymbol{y}_i|\boldsymbol{\mu}_{t_i}, \boldsymbol{\sigma}_{t_i}^2 \boldsymbol{I}, \boldsymbol{z}_t) \\ &\propto \prod_{i=1}^{m} \boldsymbol{z}_t^T \exp{(\boldsymbol{y}_i - \boldsymbol{\mu})^T (\boldsymbol{y}_i - \boldsymbol{\mu})} \end{aligned} \tag{3}$$

To stabilize the training process, when calculate the likelihood, we first normalize each $\boldsymbol{y}_i$ and $\boldsymbol{\mu}_k$.

3.2 Sequence Modeling Component

Since the structural features are closely related to the word sequence information, we construct a sequential VAE. By the sequential VAE, the encoder infers the structural latent variable and the decoder reconstructs texts via integrating the topic latent variable and the structural latent variable. This process is depicted by the bottom half of Fig. 1. Since the semantic information is given by the topic modeling component, the reconstruction loss function makes the encoder of sequential VAE focusing on encoding the structural features. Let $\boldsymbol{z}_s$ denote the structural latent variable, the ELBO of this component is in the following form:

$$\begin{aligned}\mathcal{L}_S =& \mathbb{E}_{q(\boldsymbol{z}_t|\boldsymbol{x})q(\boldsymbol{z}_s|\boldsymbol{x})}\left[\log p(\boldsymbol{x}|\boldsymbol{z}_t, \boldsymbol{z}_s)\right] \\ & - KL(q(\boldsymbol{z}_s|\boldsymbol{x})||p(\boldsymbol{z}_s))\end{aligned} \tag{4}$$

We adopt a bi-directional GRU as the encoder. The last hidden state of the encoder is used to infer the parameters $\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s$ of the Gaussian distribution. Then, we get $\boldsymbol{z}_s$ by the reparameterization trick.

$$\begin{aligned}\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_n &= BiGRU(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n) \\ \boldsymbol{\mu}_s, \boldsymbol{\sigma}_s &= f_3(\boldsymbol{h}_n), f_4(\boldsymbol{h}_n) \\ \boldsymbol{z}_s &\sim \mathcal{N}(\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s^2 \boldsymbol{I})\end{aligned}$$

where f_3 and f_4 are the feedforward neural networks. Further, we obtain the holistic latent code $\boldsymbol{z}$ by concatenating $\boldsymbol{z}_t$ and $\boldsymbol{z}_s$, i.e., $\boldsymbol{z} = [\boldsymbol{z}_t; \boldsymbol{z}_s]$.

We adopt a GRU as the decoder to reconstruct document $\boldsymbol{x} = \{\boldsymbol{x}_i\}_{i=1}^n$ and latent code $\boldsymbol{z}$ as its initial state. So, the likelihood of the reconstructed document can be derived by Eq. 5.

$$\begin{aligned}p(\boldsymbol{x}|\boldsymbol{z}_t, \boldsymbol{z}_s) &= p(\boldsymbol{x}_1|\boldsymbol{z}) \prod_{i=1}^{n} p(\boldsymbol{x}_i|\boldsymbol{x}_{1:i-1}) \\ &= p(\boldsymbol{x}_1|\boldsymbol{z}) \prod_{i=1}^{n} p(\boldsymbol{x}_i|h_i)\end{aligned} \tag{5}$$

where $\boldsymbol{h}_i$ denotes the i-th hidden state of the GRU decoder.

3.3 Topic Encoder as a Discriminator

Although the holistic latent code contains semantics and structure features, the decoder may not fully leverage the semantic part of code. Besides the reconstruction loss which drives the generator to produce realistic sentences, we introduce a discriminator which enforces the generator to produce texts in a coherent topic with $\boldsymbol{z}_t$. Specifically, we let the encoder in our topic modeling component act as the discriminator. It is expected that topic distributions inferred from the generated texts are similar to the topic distributions of the original texts. However, if the outputs of the sequence decoder are discrete, it is impossible to propagate gradients from the discriminator through the discrete samples. We thus resort to a *Gumbel-Softmax* distribution as an approximation of the discrete samples.

In detail, in each step of the generative process of texts, we get the distribution of one word $p(\boldsymbol{x}_i|\boldsymbol{z}_s, \boldsymbol{z}_t) = [\pi_1, \pi_2, ..., \pi_{|V|}]$, and approximate the samples from $p(\boldsymbol{x}_i|\boldsymbol{z}_s, \boldsymbol{z}_t)$ by

$$u_i = \frac{\exp\left(\log(\pi_i) + g_i\right)/\tau}{\sum_{j=1}^{|V|} \exp\left(\log(\pi_j) + g_j\right)/\tau} \tag{6}$$

where g_i and g_j are samples from $Gumbel - Softmax(0, 1)$. As training proceeds, τ gets close to 0, yielding the increasingly peaked distribution that finally emulate the discrete cases. Thus, the i-th word generated by the decoder is

$$\widehat{\boldsymbol{x}}_i = \boldsymbol{u}^T \boldsymbol{W}_v \tag{7}$$

where $\boldsymbol{W}_v \in \mathbb{R}^{|V| \times e}$ denotes word embedding.

We feed the above generative texts into the topic modeling encoder and expect to get the maximum likelihood of the original topic distribution $\boldsymbol{z}_t$. So the loss function of the discriminator is specified as follows.

$$\mathcal{L}_D = \mathbb{E}_{p(\boldsymbol{z}_s)p(\boldsymbol{z}_t)} \log q(\boldsymbol{z}_t|\widehat{\boldsymbol{x}}) \tag{8}$$

Combining three parts of loss functions, we can get the loss function of the model as Eq. 9, where λ_D=0.1.

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_T + \mathcal{L}_S + \lambda_D \mathcal{L}_D \\ &= \mathbb{E}_{q(\boldsymbol{z}_t|\boldsymbol{x})}\left[\log p(\boldsymbol{x}|\boldsymbol{z}_t) + \mathbb{E}_{q(\boldsymbol{z}_s|\boldsymbol{x})}(\log p(\boldsymbol{x}|\boldsymbol{z_s}, \boldsymbol{z_t}))\right] \\ &- \lambda_t KL(q(\boldsymbol{z}_t|\boldsymbol{x})||p(\boldsymbol{z}_t)) - \lambda_s KL(q(\boldsymbol{z}_s|\boldsymbol{x})||p(\boldsymbol{z}_s)) \\ &+ \lambda_D \mathbb{E}_{p(\boldsymbol{z}_s)p(\boldsymbol{z}_t)} \log q(\boldsymbol{z}_t|\widehat{\boldsymbol{x}})\end{aligned} \tag{9}$$

We learn parameters of topic modeling and sequence modeling components alternatively in the multitask learning setting. To avoid the latent variable collapse problem, we add weights λ_t, λ_s to *KL* related terms in the loss function and make the weights increase slowly to 1 in training.

4 Experiments and Result Analyses

To evaluate the text generation ability, we do experiments from four perspectives. First, we evaluate the language modeling ability of the sequence modeling component and the topic coherence of the topic modeling component. Next, to evaluate the effectiveness of the learned latent codes of texts, we perform a semi-supervised classification task. Finally, we transfer the question-answering pairs on Yahoo dataset to different topics and demonstrate generated texts.

4.1 Dataset Description & Experiment Setup

We conduct experiments on five benchmark text datasets: APNEWS[1], IMDB (Maas et al., 2011),

[1]https://www.ap.org/en-gb/

Data	#SM Voc	#TM Voc	#Training Docs	#Development Docs	#Test Docs	#Avg Len
APNEWS	33k	6k	700k	27.4k	26.3k	21.4
IMDB	34k	8k	900k	200k	200k	22.5
BNC	40k	9k	800k	44k	52k	22.6
Yahoo	20k	8k	100k	10k	10k	75.3
Yelp15	20k	7k	100k	10k	10k	97.8

Table 1: Summary statistics for datasets. SM Voc and TM Voc stand for vocabularies of corpus in the sequence modeling component and the topic modeling component, respectively.

BNC (BNC Consortium, 2007), Yahoo Answer (Yahoo) and Yelp15. We randomly sample 100k as training data and 10k as validation and testing data, respectively. For all datasets, we first tokenize the texts using Stanford CoreNLP (Manning et al., 2014). Then we lowercase all word tokens, and filter words that occur less than 10 times. For Yahoo and Yelp15, we truncate the vocabulary to 20k words for fast training. For the bag-of-words input in the topic modeling component, we further remove stop words, and exclude the top 0.1% most frequent words and also words that appear in less than 100 documents. Table 1 shows summary statistics of all datasets.

We fix the max sequence length to 50 for the texts in APNEWS, IMDB, BNC and 150 for Yahoo and Yelp15. The 300-dimentional embeddings of words are shared by two components in our model. For the topic modeling component, we adopt a 2-layer MLP with 200 hidden units and *ReLU* as its activation function. We set the size of $\boldsymbol{z}_t$ to 50. For the sequence modeling component, we adopt a bidirectional single layer GRU with 600 hidden units (300 in each direction) as the encoder and a unidirectional GRU with 300 hidden units as the decoder. The size of $\boldsymbol{z}_s$ is set to 20. We use a batch size of 32 and train the model up to 40 epochs. Linear scheduling is used in the *KL* annealing and the weight grows from 0 at the beginning to 1 at 40k steps.

4.2 Sequence Modeling Evaluation

By the experiments on five datasets, we evaluate the language modeling ability in terms of perplexity (*PPL*). We compare several baselines including models based on language model (i.e., LSTM LM, LSTM+LDA, Topic-RNN, TDLM) and models based on VAE (i.e., LSTM VAE, VAE+HF, TGVAE, DCNNVAE, DVAE). LSTM LM is a plain language model implemented in LSTM. LSTM+LDA concatenates the hidden states with the topic distribution learned by a pre-trained LDA. Different from ours, LDA in LSTM+LDA is trained separately. Topic-RNN (Dieng et al., 2017) learns an LDA with a language model jointly and incorporates the topic distribution by a gate mechanism. TDLM (Lau et al., 2017) incorporates a convolutional topic model and also leverages the topic distribution in the same way that LSTM+LDA does. LSTM VAE is a standard VAE whose encoder and decoder are implemented by two LSTMs, respectively. VAE+HF (Wang et al., 2019) is a VAE with a mixture-of-Gaussians prior with Householder Flow. TGVAE (Wang et al., 2019) is a VAE guided by a Gaussian mixture distribution as prior with a jointly-trained LDA. DCNN-VAE (Yang et al., 2017) is a VAE using dilated CNN as its decoder. DVAE (Xiao et al., 2018) uses a Dirichlet latent variable to improve VAE. Besides the model we propose, we also evaluate our model without the discriminator (i.e., Ours w/o Dis).

The perplexity of VAE-based models is estimated in ELBO approximately which is comprised of a reconstruction term and a *KL* term. Besides the perplexity, we report the *KL* term in the VAE based models. For our model, we report the *KL* values of the sequence modeling component and the topic modeling component in the first and second rows, respectively.

For a fair comparison, the compared results are picked from the models with 50 topics. The results are shown in Table 2. We find that our model and our model without a discriminator occupy the top-2 positions. We attribute the improvements to the decoupling of semantic and structural features. From the results, we also verify that the discriminator in our model can help in decreasing the perplexity. In addition, the *KL* values in the topic modeling component is much larger than the sequential one. One possible reason is that the topic information reveals much of the diversity of texts.

4.3 Topic Coherence Evaluation

Topic models are traditionally evaluated using perplexity. However, (Chang et al., 2009) show that

	APNEWS		IMDB		BNC		Yahoo		Yelp15	
	PPL	*KL*	*PPL*	*KL*	*PPL*	*KL*	*PPL*	*KL*	*PPL*	*KL*
LSTM LM	64.13	-	72.14	-	102.89	-	66.2	-	42.6	-
LSTM+LDA	57.05	-	69.58	-	96.42	-	53.5	-	37.2	-
Topic-RNN	56.77	-	68.74	-	94.66	-	-	-	-	-
TDLM	53.00	-	63.67	-	87.42	-	-	-	-	-
LSTM VAE	75.89	1.78	86.16	2.78	105.10	0.13	65.6	0.4	45.5	0.5
VAE+HF	71.60	0.83	83.67	1.51	104.82	0.17	-	-	-	-
TGVAE	48.73	3.55	57.11	5.02	87.86	4.57	-	-	-	-
DCNN-VAE	-	-	-	-	-	-	63.9	10.0	41.1	7.6
DVAE	-	-	-	-	-	-	47.6	31.9	34.7	30.5
Ours w/o DIS	47.80	2.13 7.32	54.50	3.24 9.82	83.92	1.32 7.83	42.92	3.12 11.45	**31.87**	2.43 8.72
Ours	**47.23**	2.88 8.18	**52.01**	3.87 9.34	**80.78**	2.54 7.76	**40.80**	4.25 11.51	32.90	2.92 8.34

Table 2: Language modeling results in the terms of *PPL* and *KL*.

Model	APNEWS	IMDB	BNC	Yahoo	Yelp15
LDA	0.125	0.084	0.106	0.148	0.087
TDLM	0.149	0.104	0.102	-	-
Topic-RNN	0.134	0.103	0.102	-	-
TGVAE	0.157	0.105	0.113	-	-
Ours w/o DIS	0.170	**0.121**	**0.115**	0.182	**0.114**
Ours	**0.171**	0.120	0.114	**0.182**	0.113

Table 3: Topic coherence over the datasets in the term of *NPMI*.

perplexity does not correlate with the coherence of the generated topic. We adopt normalized PMI (NPMI) to evaluate the topic coherence following (Lau et al., 2017). Given the top-n words of a topic, coherence is computed based on the sum of pairwise NPMI scores between topic words. We average topic coherence over the top 5/10/15/20 topic words. To aggregate topic coherence scores, we calculate the mean coherence over topics. In the experiments, the number of topics remains 50 among all baselines. From Table 3, we find that the discriminator gives little improvement. It is because that the role of the discriminator is to take the topic distributions as the supervised signals to improve the generation so that the sequence decoder can generate more topic relevant texts. That is, the discriminator does not improve the topic modeling component itself.

Besides topic coherence values, to understand the topics concretely, we also provide top five topic words from eight randomly chosen topics on each dataset in the supplementary material.

4.4 Semi-supervised Classification

To evaluate whether latent codes incorporate text features, we perform a semi-supervised classification task and compare our model with the other models. To make a comprehensive comparison,

Model	20NEWS	Yahoo	Yelp15
VAE	52.29	53.2*	27.2*
LDA	55.38	56.72	30.44
DVAE	-	57.6*	42.4*
TDLM	60.6*	-	-
Ours w/o DIS (topic)	59.31	60.51	35.11
Ours w/o DIS(seq)	28.25	31.42	34.39
Ours w/o DIS	61.28	63.98	45.34
Ours (topic)	59.06	61.34	34.36
Ours (seq)	30.05	30.31	35.37
Ours	**61.47**	**64.75**	**46.03**

* are from the original papers.

Table 4: Test split accuracy of classifiers trained with learned representations.

we use Yahoo, Yelp15 as well as 20NEWS for text classification. Here, 20NEWS is a collection of forum-like messages from 20 news-groups categories.

For any model to be evaluated, we first train it by the training documents of the dataset, and then executing the well-trained model to obtain latent codes of all documents in the dataset. Next, we sample 2,000 documents from the training data and train a 2-layer softmax classifier using these documents and their category labels. The accuracy on the testing set of documents is shown in Table 4. Since in our TATGM the latent code comprises two latent variables, we explore the accuracy on the two latent variables solely and collectively.

As shown in Table 4, our model which combines the two latent variables achieves the highest accuracy. For 20NEWS and Yahoo, the model only using the topic latent variable is better than that only using the structural latent variable. Since these two datasets are labeled by the topics, the combination improves little. For Yelp15 dataset, the performances of our two latent variables are similar whereas the combination improves up to

Topic	Question	Answer
Society&Culture	what is the average age of the homeless population?	i think they are younger then they look. i 'd guess 30 years old.
Science&Mathematics	what is the meaning of life in the world?	i know that i can find out about it.
Health	what is the average weight to lose of a week?	i think it varies to lose weight.
Education&Reference	what is the formula for energy?	i have about 10 points for asking questions.
Sports	which team has the most winning world cup?	i think brazil will win the world cup finals. i think they will win.
computers	what is the problem with the yahoo messenger?	i think yahoo messenger is not working on yahoo messenger.
Business&Finance	what is the minimum wage for a tax income?	i do n't know if they are illegal to pay taxes.
Entertainment&Music	what is the way of listen songs?	i think listen a song on internet.
Family&Relationships	what is the average age of have sex?	i am 14 years old and i 'm not sure what age is it?
Politics&Government	what states is the president of the united states?	i do n't know if it is illegal to be illegal.

Table 5: A topic transfer example of one question-answering pair on Yahoo dataset.

15%. The reason may be the fact that the Yelp15 dataset is labeled by the sentiment which is determined by not only single terms but also n-grams. For example, *good* and *not good* represents opposite sentiments. Our sequence modeling component can capture such kind of features more accurately.

4.5 Topic Transfer between Question-answering Pairs

In our model, each dimension of the latent variables in the topic modeling component corresponds to a topic, therefore we can manipulate the variables manually to verify whether the generation can express a given topic while remaining the same structure. Specifically, given a document and its latent code, we change the topic part of the code and keep the structural part of the code unchanged. Then we can check the text the decoder generated from the whole latent code.

We conduct the experiments on the Yahoo dataset where each item contains one question-answering pair and one label corresponds to its topic. At first, we treat each question as a single sentence to check whether the generated texts satisfy our assumption. From the second column of Table 5, we find that the generation can express a different topic while maintaining the original question structure.

We further try to transfer the question-answering pair from its original topic to target topics. We find that the model not only transfers topic of two sentences but also produces reasonable question-answering pairs, i.e., the new answer is meaningful to the new question.This is helpful for the automatic question-answering scenarios. Table 5 shows an example of topic transfer. The first row lists the original question-answering pair which are in the topic of *Society&Culture*. The rest rows show the generated question-answering pairs when we change the original topic to a target topic while keeping the structure latent variable unmodified. Specifically, we modify the original topic distribution to a new distribution whose dimension of the target topic is 1 and others are 0s. From the results, we can observe that while we change the topic to another, the generated questions remain almost the same structure as the original ones. Moreover, the generated answers are also transferred to the target topic and they together with the generated questions compose reasonable question-answering pairs. More examples are shown in the supplementary material.

5 Conclusion

In the paper, we present the text generation model TATGM. The model can learn semantics and structural features simultaneously. Moreover, the model employs a discriminator to ensure that the generated text is more coherent with the given topic information. Experimental results show that our model has a better text modeling ability than several state-of-the-art methods and learns disentangled latent representations for texts which shows the superiority in a classification task. Specifically, our model can generate meaningful question-answering pairs, which provides an alternative transfer learning way and helps to broaden

the knowledge in other fields.

Acknowledgments

This work has been supported by National Key R&D Program of China (No. 2017YFC0803300).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

BNC BNC Consortium. 2007. The british national corpus, version 3 (bnc xml edition). In *Distributed by Bodleian Libraries, University of Oxford, on behalf of the BNC Consortium*.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 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*, pages 10–21.

Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 288–296.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 795–804.

Adji B. Dieng, Chong Wang, Jianfeng Gao, and John W. Paisley. 2017. Topicrnn: A recurrent neural network with long-range semantic dependency. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. 2015. DRAW: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1462–1471.

Pengfei Hu, Wenju Liu, Wei Jiang, and Zhanlei Yang. 2012. Latent topic model based on gaussian-lda for audio retrieval. In *Chinese Conference on Pattern Recognition*, pages 556–563. Springer.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1587–1596.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Jey Han Lau, Timothy Baldwin, and Trevor Cohn. 2017. Topically driven neural language model. *arXiv preprint arXiv:1704.08012*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 2410–2419.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International conference on machine learning*, pages 1727–1736.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *2012 IEEE Spoken Language Technology Workshop (SLT), Miami, FL, USA, December 2-5, 2012*, pages 234–239.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 627–637.

Akash Srivastava and Charles A. Sutton. 2017. Autoencoding variational inference for topic models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Youzhi Tian, Zhiting Hu, and Zhou Yu. 2018. Structured content preservation for unsupervised text style transfer. *CoRR*, abs/1810.06526.

Wenlin Wang, Zhe Gan, Wenqi Wang, Dinghan Shen, Jiaji Huang, Wei Ping, Sanjeev Satheesh, and Lawrence Carin. 2018. Topic compositional neural language model. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, pages 356–365.

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational autoencoders for text generation. *CoRR*, abs/1903.07137.

Yijun Xiao, Tiancheng Zhao, and William Yang Wang. 2018. Dirichlet variational autoencoder for text modeling. *CoRR*, abs/1811.00135.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 7298–7309.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3881–3890.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 2852–2858.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 521–530.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6069–6076.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskénazi. 2018. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1098–1107.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 654–664.