Topic-Aware Variational Auto-Encoders for Controllable Text Generation

Anonymous ACL submission

Abstract

In this paper, we propose **T**opic-**A**ware **V**ariational **A**uto-**E**ncoders for Controllable Text Generation (**TA-VAE**). Distinct from existing VAE based approaches, we explicitly model document topic and sequence apart: a text variational auto-encoder (VAE) is utilized for sequence modeling, whose posterior is remolded by a Householder flow to be compatible with the non-isotropic allocation of texts (with diverse topics) in latent space; a variational topic model with its prior conditioned on well-crafted sequential posterior to take advantage from acquired text sequential information. Besides, an explicit discriminator (based on the topic encoder) as well as a mutual information maximization term (on topic latent code and observed data) are additionally added to enhance the utterance of topic behalf. Encouraging experimental results on real-world datasets demonstrate that the proposed model not only learns interpretable topic representations, but is fully capable of generating high-quality paragraphs that are grammatically reasonable and semantically consistent.

1 Introduction

In recent years, considerable advanced network architectures are employed to design robust and effective language models (LMs) for text generation. These language models gain apparent improvement in varied generation tasks, including machine translation (Bahdanau et al., 2014), summarization (Rush et al., 2015) and question answering (Iyyer et al., 2014). However, generating texts that fulfil expected attributions (e.g., topics, sentiment) remains a mountain to climb. However, methods incorporate explicit constraints (Mei et al., 2015; Wiseman et al., 2018; Jain et al., 2018) often face challenges like dull syntax, semantical discontinuity (Wiseman et al., 2017) and rigorous model requirement (Garbacea and Mei, 2020). Yet generation with implicit constraints is more compatible to produce authentic texts, and also in favor of downstream tasks by catching hold of high-quality linguistic representations.

Compared with other approaches to produce textual content, such as those based on generative adversarial networks (GANs) or plain recurrent neural network (RNN), VAE is suitable for text generation with implicit constraints, because its flexible latent representation is capable of capturing integral properties of input, such as style, topic, and high-level linguistic or semantic features (Fang et al., 2019). Nevertheless, a plain text VAE with one monopolistic latent space is faced with latent vacancy dilemma (Xu et al., 2020), which makes it notoriously unsuitable for controllable text generation. By infusing side knowledge to VAE-based LMs, techniques for generating desired sentences are widely explored (Wang et al., 2019; Tang et al., 2019; Rezaee and Ferraro, 2020).

However, other problems arise in practice may limit the modeling capacity and empirical performance of VAE-based models. KL collapse is one of the major challenges that are widely concerned (Bowman et al., 2015). Several approaches have been devised to handle this issue, including optimizing decoder architectures (Yang et al., 2017; Semeniuta et al., 2017; Li et al., 2020a), inventing auxiliary objectives (Zhao et al., 2017a,b; Xiao et al., 2018; Fang et al., 2019; Dai et al., 2020), novel encoder training schedule (Bowman et al., 2015; Fu et al., 2019), flexible latent code posterior (Wang et al., 2019), etc. These methods generally share a same goal: to impair the ability of powerful recurrent decoder and strengthen the expression of latent space. The second issue associated with a VAE to generate topic-specified texts is rooted in the assumption of its variational posterior, which usually accepts a spherical Gaussian distributions with diagonal co-variance matrices. Thus the true posterior can only be well approximated by the possible variational one when it is in the exact same family (Cremer et al., 2018). To address such plight, latent information with external help beyond only one single continuous space was considered (Xiao et al., 2018), but its training can not be regarded as end-to-end. As a fixup, methods that extract both text syntax and topic information simultaneously were proposed (Tang et al., 2019), but they suffered from an oversimplified representation in sequence component for analogous samples (i.e., isotropic Gaussian) for both hidden codes. Flexible latent modeling had also attracted attention (Wang et al., 2019; Dai et al., 2020), whereas it confused the text structure knowledge and topic information, which made the model less interpretable.

These methods (1) ignore the nature that topic-specified sentences are not analogous thus their representations are unlike to be fit in isotropic space; (2) neglect that modeling diverse topic information from scratch is harder than text sequential modeling using RNNs, so external help for topic learning benefits; (3) may confuse topic and sequence modeling in a holistic continuous space, which makes them suffer from interpretability and mode collapse issues for controllable generation.

In this paper we address these limitations and propose TA-VAE. As illustrated in Figure 1, our model essentially consists of a topic modeling part and a sequence modeling part, which equip their own continuous latent space and are both optimized based on VAE. In detail, TA-VAE discards the spherical Gaussian assumption of latent sequence component and replace its posterior with a more flexible Gaussian distribution using Householder flow. In order to maximize the utilization of coherent sequence latent space, we also condition the topic prior on expressive sequence posterior, which acts like a prophet in the topic learning process and brings about a leap forward on both language modeling and topic concentration level. Moreover, we estimate and maximize the mutual information between topic representations and input data to distill document topic knowledge, and also adjust the topic encoder as a discriminator to aggregate the topic expression.

Contributions. (1) We present TA-VAE, a novel approach to document topic modeling and controllable text generation based on VAE. (2) We clearly separate topic modeling and text generation process, propose to condition the topic latent on flexible sequence latent distribution parametrized by Householder flow. (3) We adapt a topic discriminator and a latent mutual information term to regularize topic learning, and further verify their effectiveness in multi-tasks. (4) The overall effectiveness is validated by consistently remarkable results on language modeling, topic modeling, classification and unsupervised style transfer tasks. Our model reaches the state-of-the-art performance on text perplexity for **better quality of output content**, and the topic latent classification accuracy for **higher interpretability of topic learning**.

2 TA-VAE Methodology

In this section, we will firstly introduce variational auto-encoder for text generation, then the proposed model. Since TA-VAE is essentially: a topic model for topic recognition and a conditional encoder-decoder frame for text generation, we will start from these two parts and then dive into their joint training stage and model enhanced components. A graphic illustration of the model is in the left part of Figure 1. Observed variables are in gray, while unseen variables are in white. Solid lines represent the inference process, dashed lines work during the training process. The corresponding model structure is in the right part of Figure 1.

2.1 Text Variational Auto-Encoder

Latent variable models (LVM) such as VAE-based models aim at minimizing the average negative log likelihood (NLL) of data $\boldsymbol{X}$. They achieve this goal by updating the evidence lower bound (ELBO) of $p_{\boldsymbol{\theta}}(\boldsymbol{X})$, which consists of a reconstruction loss and a regularization term on latent $\boldsymbol{z}$:

$$\begin{aligned} \log p(\boldsymbol{X}) \geq \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{X})}\left[\log p(\boldsymbol{X} \mid \boldsymbol{z})\right] \\ - \mathbb{D}_{\mathrm{KL}}\left(q(\boldsymbol{z} \mid \boldsymbol{X}) \| p(\boldsymbol{z})\right). \end{aligned} \tag{1}$$

Yet the existing LVM-related works mainly assume the latent code $\boldsymbol{z}$ follows an isotropic Gaussian with diagonal covariance matrix (Kingma and Welling, 2013; Rezende et al., 2014; Bowman et al., 2015), which can only be well and truly gained on if the actual latent distribution is exactly a Gaussian. This hypothesis leaves huge defects when it comes to modeling samples with obvious variations (e.g., topic-controlled sentences).

In recent years, *normalizing flow* (NF) (Rezende and Mohamed, 2015) as a practical framework has been widely employed to generative models (Dinh et al., 2014; Ziegler and Rush, 2019; Ding and Gimpel, 2021). By starting with a relatively simple distribution (e.g., Gaussian), it uses a series of

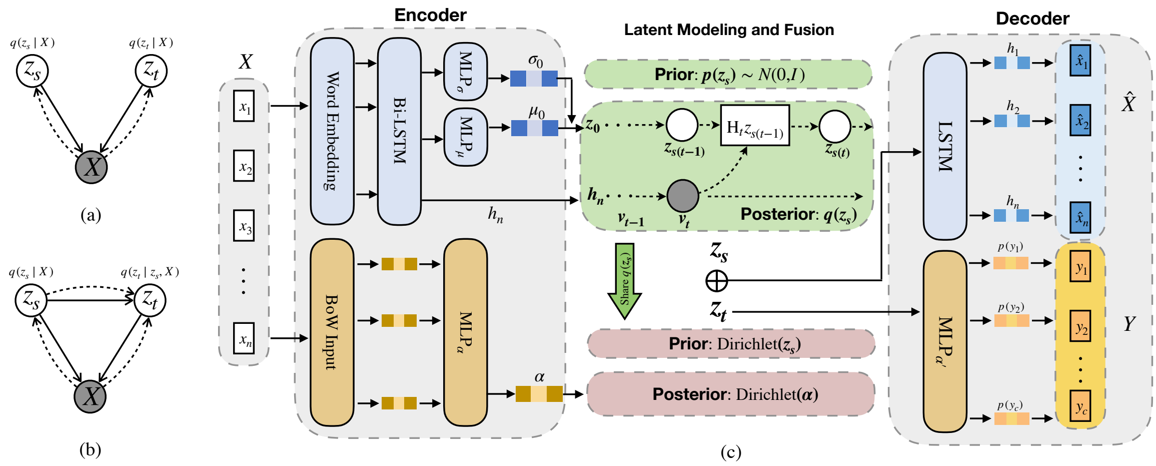

Figure 1: Graphic model of (a) modeling of sequence and topic latent codes without any dependency between $\boldsymbol{z_t}$ and $\boldsymbol{z_s}$ as most previous works, (b) conditional assumption between topic and sequence latent codes described in our model. (c) Model structure of TA-VAE. The overall architecture observes the Encoder-Decoder framework, which leverages two separate models for sequence and topic modeling.

invertible functions to form the overall transformation and obtain a flexible representation of data. Latent distributions parameterized with NF are no longer constrained to a specific distribution family, allowing more accurate estimation towards the data pattern. For a VAE-based generative model, the normalizing flow can be used to enrich the posterior of it with small or even none modifications in the architecture of the encoder and the decoder (Tomczak and Welling, 2016).

2.2 From Texts to Topic Knowledge Learning

Bag-of-Word (BoW) is a generally recognized input manner for a topic model, thus we utilize such method for our neural topic part. We define c to be the corpus size, and $d \in \mathbb{Z}_+^c$ as the BoW representation of a document $\boldsymbol{X} = [x_1, x_2, ..., x_n]$ with length n, which indicates that every document has c elements with non-negative count. As in a standard latent Dirichlet allocation (LDA) model, we first assume there are T topics, and ideally each is represented as one dimension of the document-level Dirichlet parameter. Although recurrent features of texts can be caught by RNNs preferably, sometimes topic knowledge is less faithful to be accurately modeled. To synchronously obtain both information from a text corpus, topic modeling component can surely benefit from some extra helps. As a result, **we share the well-expressive sequential posterior from the sequence part** to ease the burden of learning topic knowledge groundlessly for the topic component. The generative process of our topic part can then be accomplished via the output probability of each word token, which can be specified as: (1) draw $\boldsymbol{z_s}$ from the sequence posterior: $\boldsymbol{z_s} \sim q(\boldsymbol{z_s}|\boldsymbol{X})$; (2) generate topic prior condition on $\boldsymbol{z_s}$: $p(\boldsymbol{z_t} \mid \boldsymbol{z_s}) = f_z(\boldsymbol{z_s})$; (3) draw $\boldsymbol{z_t}$ from its learned prior: $\boldsymbol{z_t} \sim p(\boldsymbol{z_t} \mid \boldsymbol{z_s})$; (4) generate output probability of topic words from topic decoder: $[p(y_1), ..., p(y_c)] = g(\boldsymbol{z_t})$. Here $f_z(\cdot)$ and $g(\cdot)$ are two functions acting on $\boldsymbol{z_s}$ and $\boldsymbol{z_t}$ respectively, $\boldsymbol{X}$ is the original document and $\boldsymbol{Y} = [y_1, y_2, ..., y_c]$ is the reconstructed words from topic decoder, which are non-sequential. In detail, function $f_z(\cdot)$ is implemented with a neural linear layer with bias, while $g(\cdot)$ consists of one linear layer with batch normalization and a softmax function. The recovery process of topic model (see Appendix A.1.1 for the complete proof) can be specified as:

$$p(\boldsymbol{Y}) = \int_{z_t} \int_{z_s} p(\boldsymbol{Y}, \boldsymbol{z_s}, \boldsymbol{z_t}) d\boldsymbol{z_s} d\boldsymbol{z_t}. \tag{2}$$

To preferably depict the topic distribution of documents, $\boldsymbol{z_t}$ follows Dirichlet as mentioned above.

Since the neural topic component is constructed in the fashion of VAE, the ELBO of this component is in the following form:

$$\begin{aligned}\mathcal{L}_T =\ & \mathbb{E}_{q(\boldsymbol{z_s}|\boldsymbol{X})q(\boldsymbol{z_t}|\boldsymbol{X},\boldsymbol{z_s})} \left[\log(p(\boldsymbol{Y} \mid \boldsymbol{z_t}, \boldsymbol{z_s}))\right] \\ & - \lambda_T \mathbb{E}_{q(\boldsymbol{z_s}|\boldsymbol{X})} \left[\mathbb{D}_{\mathrm{KL}}(q(\boldsymbol{z_t} \mid \boldsymbol{X}, \boldsymbol{z_s}) \| p(\boldsymbol{z_t} \mid \boldsymbol{z_s}))\right],\end{aligned} \tag{3}$$

with $q(\boldsymbol{z_t} \mid \boldsymbol{X}, \boldsymbol{z_s})$ and $p(\boldsymbol{z_t} \mid \boldsymbol{z_s}))$ to be the posterior and conditional prior of $\boldsymbol{z_t}$ respectively.

2.3 From Latent Codes to Guided Text Generation

Text modeling stage can be roughly split into two phases under the framework of variational encoder-decoder, namely text recurrent feature capture and joint generation with obtained topic guidance. Recurrent structure of texts is sequentially correlated, thus we utilize a text variational auto-encoder (textVAE) (Bowman et al., 2015) to model the sequential features of textual sequences. To be specific, we assign variable $\boldsymbol{z_s}$ from a continuous latent space that follows non-isotropic Gaussian for sequential feature modeling. When it comes to conditional language generation, controlled LMs aim at generating attribute-specified contents, which requires applicable mix plans for topic knowledge and text sequential information. In our model, there are two moments for them to be fully integrated.

As mentioned above, a flexible posterior of $\boldsymbol{z_s}$ is utilized as a condition for topic latent $\boldsymbol{z_t}$ update. During training, this connection not only assists topic model to learn with the help of basic sentence understanding, but also pushes $\boldsymbol{z_s}$ to be updated in the direction of learned topic messages through backpropagation. For the recurrent decoder, we concatenate two obtained latent variables from separate components as the holistic code $\boldsymbol{z} = [\boldsymbol{z_s}, \boldsymbol{z_t}]$ and further feed to the decoder as its direct input. For a reconstructed document $\hat{\boldsymbol{X}}$ output from the proposed method, its probability likelihood can be calculated as follow:

$$p(\hat{\boldsymbol{X}} \mid \boldsymbol{z}) = \prod_{i=1}^{n} p(x_i \mid x_{1:i-1}, \boldsymbol{z}) = \prod_{i=1}^{n} p(x_i \mid \boldsymbol{h_i}, \boldsymbol{z}), \tag{4}$$

where $\boldsymbol{h}_i$ is the i-th hidden state of the decoder RNN that satisfies $\boldsymbol{h}_i = \text{Decoder}(\boldsymbol{h}_{i-1}, x_{i-1}, \boldsymbol{z})$. Overall, the ELBO of our customized sequence VAE is:

$$\begin{aligned}\mathcal{L}_S = &\mathbb{E}_{q(\boldsymbol{z_t},\boldsymbol{z_s}|\boldsymbol{X})} [\log(p(\boldsymbol{X} \mid \boldsymbol{z_t}, \boldsymbol{z_s}))] \\ &- \lambda_S \mathbb{D}_{\text{KL}}(q(\boldsymbol{z_s} \mid \boldsymbol{X}) \| p(\boldsymbol{z_s})).\end{aligned} \tag{5}$$

Note that, the ELBOs of these two separate components are essentially corelative and can be rewritten in a unified manner (see Appendix A.1.2).

2.4 Householder Flow for $q(\boldsymbol{z_s} \mid \boldsymbol{X})$ Approximation

Endowing sequence posterior $q(\boldsymbol{z_s} \mid \boldsymbol{X})$ with high flexibility, so TA-VAE can not only models topic-specified texts but provides timely help for $\boldsymbol{z_t}$ learning. We apply a linear normalizing flow: Householder flow (Tomczak and Welling, 2016; Zhang et al., 2018; Wang et al., 2019) to leverage this process. Householder flow is made up of a series of *Householder transformations*. When applying to distribution estimation, it is not only capable of generating more flexible sequential posteriors thanks to its nature as a flow, but significantly simplifies the objective of flow-based variational methods. Because there stands $\log \left|\det \frac{\partial \boldsymbol{H_k z_{k-1}}}{\partial \boldsymbol{z_{k-1}}}\right| = 0$ for $k \in [1, K]$. By starting from a simple posterior with the full covariance matrix $\boldsymbol{z_{s(0)}}$ from sequence encoder, a K-layer Householder flow is inflicted to it in order to better approximate the true posterior that befits various topics. The loss function of our sequence part in Eq. (5) should be modified as:

$$\begin{aligned}&\mathbb{E}_{q(\boldsymbol{z_t},\boldsymbol{z_{s(0)}}|\boldsymbol{X})} \left[\log(p(\boldsymbol{X} \mid \boldsymbol{z_t}, \boldsymbol{z_{s(K)}}))\right] \\ &- \lambda_S \mathbb{D}_{\text{KL}}(q(\boldsymbol{z_{s(0)}} \mid \boldsymbol{X}) \| p(\boldsymbol{z_{s(K)}})).\end{aligned} \tag{6}$$

Though we only use flow to directly produce sequence posterior, the approximation method is also conducive to the topic latent $\boldsymbol{z_t}$ due to its conditional assumption on $\boldsymbol{z_s}$. Note that, distinct from TGVAE (Wang et al., 2019), which also utilizes Householder flow but does not divide topic and sequence modeling and requires Gaussian mixture model (GMM) to parameterize the hidden space, our method is more simple and effective to employ (check Section 3.3 for experimental results). A detailed introduction about flow-based VAE models is in Appendix A.2.

2.5 Topic-Aware Objectives

2.5.1 Discriminator

In the explicit manner, we expect the generated sentences could approach to the input texts in terms of topic representation as much as possible. We resort to a discriminator that is similar to the one described in Tang et al. (2019) to fulfill this goal. Formally, we re-input the output from our generative scheme $\hat{\boldsymbol{X}}$ to the topic modeling part. The updated objective of our discriminator setting is:

$$\mathcal{L}_D = \mathbb{E}_{p(\boldsymbol{z_s})p(\boldsymbol{z_t})} \left[\log q(\boldsymbol{z_t} \mid \hat{\boldsymbol{X}})\right]. \tag{7}$$

However, topic discriminator in Tang et al. (2019) transfers tokens by word embedding and inevitably demands the same size between the hidden layers of topic encoder and word embedding, instead, we employs the BoW input as the embedding from topic encoder to avoid such dilemma.

Model	APNEWS	IMDB	BNC	PTB
LSTM LM	64.13	72.14	102.89	116.2
LSTM+LDA	57.05	69.58	96.42	-
Topic-RNN	56.77	68.74	94.66	97.3
TDLM	53.00	63.67	87.42	-
LSTM VAE	75.89	86.16	105.10	96.0
VAE+HF	71.60	83.67	104.82	-
TCNLM	52.75	63.98	87.98	-
TGVAE	48.73	57.11	87.86	-
DVAE	-	-	-	33.4
TATGM	47.23	52.01	80.78	-
rGBN-RNN	42.71	51.36	79.13	-
VRTM	47.78	51.08	86.33	55.82
iVAE	-	-	-	53.44
APo-VAE	-	-	-	53.02
Ours ↓	**36.35**	**36.53**	**76.34**	**27.25**

Table 1: Text quality analysis in terms of text perplexity (*PPL*). All topic language models remain the same topic latent size (if available) of 50.

Dataset	F=0	F=5	F=10	F=20
IMDB	52.01	37.48	36.53	35.75
PTB	49.06	27.40	27.25	26.94

Table 2: *PPL* of our models on test set with various number of flow layers (represented by F).

2.5.2 Mutual Information Maximization

Inspired by infoVAE (Zhao et al., 2017a), which adds a mutual information (MI) term between latent codes for direct output ($\boldsymbol{z_s}$) and the input data ($\boldsymbol{X}$) to avoid vanished representations, we encourage the model to explicitly maximize the MI term between input data and the conditioned topic latent code (instead of $\boldsymbol{z_s}$ for direct textual output) $I(\boldsymbol{X}; \boldsymbol{z_t} \mid \boldsymbol{z_s})$. Maximizing such MI term between observed data and conditioned topic latent can be factored into two items related to KL divergence $\mathbb{D}_{\text{KL}}(q(\boldsymbol{z_t} \mid \boldsymbol{X}, \boldsymbol{z_s}) \| p(\boldsymbol{z_t} \mid \boldsymbol{z_s}))$ and $\mathbb{D}_{\text{KL}}(q(\boldsymbol{z_t} \mid \boldsymbol{z_s}) \| p(\boldsymbol{z_t} \mid \boldsymbol{z_s}))$. A detailed proof can be found in Appendix A.1.3. Finally, we can rewrite the holistic ELBO of the proposed model into an equivalent form:

$$\begin{aligned} &\mathcal{L}_{\text{info}} = \mathbb{D}_{\text{KL}}(q(\boldsymbol{z_t} \mid \boldsymbol{z_s}) \| p(\boldsymbol{z_t} \mid \boldsymbol{z_s})), \\ &\mathcal{L} = \mathcal{L}_S + \mathcal{L}_T + \lambda_D \mathcal{L}_D - \lambda_{\text{info}} \mathcal{L}_{\text{info}}, \end{aligned} \tag{8}$$

λ_D and λ_{info} are weights of the discriminator loss and mutual information loss severally.

3 Experimental Results and Analysis

3.1 Datasets

We conduct our experiments on five publicly available datasets (APNEWS, IMDB, BNC, PTB and Yelp15). Details are listed in Appendix A.3.1.

3.2 Baselines

In our experiments, we compare against baseline methods that **mostly consider both topic and syntax information** into generation:

Language model (LM) based methods: LSTM LDA is a LSTM language model with learned LDA representations infuses into its hidden states. Topic-RNN (Dieng et al., 2016) blends topic distribution from an LDA component using gate mechanism, and trains jointly with the language model. TDLM (Lau et al., 2017) employs a convolutional network for topic model and also concatenates it with hidden states of RNN. rGBN-RNN (Guo et al., 2020) brings a gamma belief network as a topic model, infuses learned topic information into RNN to improve model capability.

VAE-based methods: TCNLM (Wang et al., 2018) utilizes a neural topic model based on the VAE paradigm, and a multiple experts network to generate texts. TGVAE (Wang et al., 2019) consists of the same topic model of TCNLM, but a textVAE with Gaussian mixture prior and a Householder flow to approximate its posterior. DVAE (Xiao et al., 2018) incorporates an external LDA model to improve textVAE. TATGM (Tang et al., 2019) applies multivariant Gaussian for both topic and sequence latent codes, and concatenates them for generation. VRTM (Rezaee and Ferraro, 2020) blends RNN hidden state with a binary vector sign to judge topic expression. iVAE (Fang et al., 2019) parameterizes hidden space with sample method and replace KL divergence with mutual information. APo-VAE (Dai et al., 2020) makes the latent space a Riemannian manifold with learnable prior and posterior. Note that, both iVAE and APo-VAE only equip latent codes for sequence modeling.

Though VAE-based models with mighty encoder/decoder (i.e., pre-trained language models such as GPT-2 (Radford et al., 2019)) are recently explored and show optimistic empirical results (Li et al., 2020a; Fang et al., 2021), they are not suitable for being baseline candidates because they neither derive topic latent space nor use RNN-based decoder trained from scratch for generation (fine-tuning two large pre-trained language models based

Metrics	Methods	APNEWS			IMDB			BNC			PTB		
		B-2	B-3	B-4	B-2	B-3	B-4	B-2	B-3	B-4	B-2	B-3	B-4
test-BLEU↑	VAE	0.564	0.278	0.192	0.597	0.315	0.219	0.479	0.266	0.169	0.5215	0.3633	0.2642
	VAE+HF	0.570	0.279	0.195	0.610	0.322	0.221	0.483	0.270	0.169	0.5565	0.3616	0.2529
	TGVAE(T=10)	0.584	0.327	0.202	0.621	0.357	0.223	0.518	0.283	0.173	-	-	-
	TGVAE(T=30)	0.627	0.335	0.207	0.655	0.369	0.243	0.528	0.291	0.182	-	-	-
	TGVAE(T=50)	0.629	0.340	0.210	0.652	0.372	0.239	0.535	0.290	0.188	-	-	-
	Ours(T=10)	0.6512	0.3862	0.2358	0.7202	0.4505	0.2470	0.6997	0.5947	0.4934	0.6824	0.4847	0.3564
	Ours(T=30)	0.6434	0.3776	0.2374	0.7037	0.4347	0.2566	0.6791	0.5473	0.4502	0.6705	0.4779	0.3438
	Ours(T=50)	**0.6757**	0.3983	0.2432	**0.7542**	**0.4753**	**0.2755**	**0.7681**	**0.6610**	**0.5672**	**0.6924**	**0.5076**	**0.3733**
	Ours w/o Dis (T=50)	0.6596	**0.4100**	**0.2497**	0.7447	0.4637	0.2678	0.7316	0.6234	0.5292	0.6484	0.4587	0.3297
BLEU-F1↑	VAE	0.2166	0.3491	0.3071	0.1843	0.3394	0.3364	0.2273	0.3448	0.2812	0.2033	0.4055	0.3843
	VAE+HF	0.2077	0.3439	0.3121	0.1689	0.3363	0.3401	0.2242	0.3456	0.2809	0.2174	0.4292	0.3692
	TGVAE(T=10)	0.2524	0.3916	0.3248	0.1883	0.3872	0.3446	0.2571	0.3645	0.2874	-	-	-
	TGVAE(T=30)	0.2904	0.4081	0.3324	0.2441	0.4014	0.3693	0.2837	0.3750	0.2998	-	-	-
	TGVAE(T=50)	0.2942	0.4124	0.3368	0.2544	0.4036	0.3651	0.2985	**0.3751**	0.3079	-	-	-
	Ours(T=10)	0.3720	0.4088	0.3362	0.3193	0.4265	0.3501	0.2875	0.3299	0.3513	0.3233	0.3998	0.4027
	Ours(T=30)	**0.4007**	0.4268	0.3484	**0.3371**	0.4337	0.3642	0.2933	0.3564	0.3845	**0.3562**	**0.4350**	0.4168
	Ours(T=50)	0.3813	**0.4281**	0.3487	0.3272	**0.4415**	**0.3809**	**0.3358**	0.3725	**0.3989**	0.3459	0.4246	**0.4241**
	Ours w/o Dis (T=50)	0.3842	0.4228	**0.3490**	0.3148	0.4310	0.3709	0.3284	0.3653	0.3850	0.3287	0.4093	0.3986

Table 3: Text quality analysis in terms of *test*-BLEU and BLEU-F1 score. T is the topic number.

Methods	APNEWS	IMDB	BNC	PTB	Yelp15
LDA	0.125	0.084	0.106	0.118	0.087
TDLM	0.149	0.104	0.102	-	-
Topic-RNN	0.134	0.103	0.102	-	-
TCNLM	0.159	0.106	0.114	-	-
TGVAE	0.157	0.105	0.113	-	-
TATGM	**0.171**	**0.121**	0.115	-	0.114
Ours	0.159	0.099	0.114	**0.148**	**0.135**
Ours w/o Dis	0.155	0.092	0.109	0.130	0.123
Ours w/o $\mathcal{L}_{\text{info}}$	0.165	0.084	**0.118**	0.142	0.127

Table 4: NPMI scores for topic coherence evaluation.

on VAE requires vast amount of resources). Among all forementioned baselines, the rGBN-RNN model performs currently the best in terms of text quality metrics, and the TATGM model reaches state-of-the-art values on metrics about topic coherence.

3.3 Evaluations and Analysis

3.3.1 Text Perplexity

One important role our model plays is language model. For any language model, quality of its generated sentences is of priority. We adopted text perplexity (PPL) to evaluate the model at the content level (whether the content is relevant and grammatical). The perplexity values of the baselines and our TA-VAE across four evaluation sets are shown in Table 1. We also present experiments demonstrating the performance of our methods with different layer settings in Table 2. From these tables, (1) TA-VAE outperforms other baselines across all benchmark datasets; (2) Householder flow in sequence latent level improves the PPL value by over 10 absolute points on both IMDB and PTB. Besides, with the increase of flow layers, the PPL value gradually decrease; (3) Our models without flow parametrization can still reach competitive PPL results on IMDB and PTB compared with baselines, which yields convincing effectiveness of the model design. The flow layer number was chosen to 10 for the rest experiments, more discussions are in Appendix A.5.1.

3.3.2 BLEU

Following Wang et al. (2019); Guo et al. (2020), we used *test*-BLEU to evaluate the quality of generated sentences with a set of texts from the test sets as reference, and *self*-BLEU to evaluate the diversity of generated contents (Zhu et al., 2018). It is well known that, there intrinsically exists a trade-off between text quality and text diversity. Motivated by Gu et al. (2018); Li et al. (2020b), we proposed to employ BLEU-F1 score to evaluate the overall metric involving text quality and diversity simultaneously:

$$\text{BLEU-F1} = \frac{2 \times \textit{test}\text{-BLEU} \times (1 - \textit{self}\text{-BLEU})}{\textit{test}\text{-BLEU} + (1 - \textit{self}\text{-BLEU})}. \quad (9)$$

For the baseline methods, three VAE-based topic language models were selected, among which VAE+HF and TGVAE are two systems utilizing Householder flow like the proposed TA-VAE does. Since BLEU-related indexes require specific word output and comparison, we believe the discriminator can play a more important role in this process, because it is optimized on the word-token-level, we report model performances with or without it. For-

Models	APNEWS	IMDB	BNC	PTB
LDA VB(T=10)	2.29*	2.29*	2.30*	1.75
VRTM(T=10)	2.15*	1.56*	1.76*	1.70
Ours(T=10) ↓	**1.32**	**1.46**	**1.59**	**1.46**
LDA VB(T=30)	3.39*	3.39*	3.39*	2.91
VRTM(T=30)	2.82	2.98	2.88	**2.77**
Ours(T=30) ↓	**2.57**	**2.73**	**2.68**	2.84
LDA VB(T=50)	3.90*	3.90*	3.90*	3.53
VRTM(T=50)	3.30	3.40	3.39	3.34
Ours(T=50)↓	3.01	**3.26**	**3.13**	**3.25**
Ours(T=50) w/o Dis ↓	**3.00**	3.32	3.17	3.28
Ours(T=50) w/o $\mathcal{L}_{\text{info}}$ ↓	3.02	3.30	3.15	3.26
Ours(T=50) w/o HF	3.25	3.40	3.31	3.32

Table 5: Inferred document topic entropy. Statistics with * are from Rezaee and Ferraro (2020).

mally, we carried out all the BLEU-related experiments using benchmark tool Texygen (Zhu et al., 2018). From the *test*-BLEU and BLEU-F1 scores in Table 3, we could see that our TA-VAE model is superior to the baselines in terms of BLEU-F1 as well as *test*-BLEU in most cases, and the **discriminator is a strong performer** in improving text quality (higher *test*-BLEU values in all circumstances). Moreover, values of TA-VAE on BLEU-F1 change much smoother than others from B-2 to B-3. One possible reason is that TA-VAE produces more coherent texts (under the framework of n-gram language model) than other baselines do. The full statistics, discussions, experimental settings are available in Appendix A.5.2.

3.3.3 Normalized PMI

Chang et al. (2009) argued that metrics for text quality (e.g., PPL, BLEU) are not suitable for measuring topic inference ability due to its low correlation with attribute knowledge. Hence we followed Lau et al. (2017) and tested our topic model using normalized PMI (NPMI). Detailed setup can be found in Appendix A.3.5. The numbers of topics remained 50 among all baselines. The flow layer number was 10 for all TA-VAE models. From Table 4, we find that **the discriminator gives more improvement than $\mathcal{L}_{\text{info}}$ does**. It is because NPMI calculation requires explicit topic word outputs, which indicates that discriminator is more adept at. While informative penalty is an implicit optimized proposal, that is, $\mathcal{L}_{\text{info}}$ helps reinforce the topic model in the latent spaces with more efficiency than the direct output of topic modeling part.

Though the primary goal of the proposed model is to generate sentences with matching attributes instead of topic words production (Wang et al., 2019).

Model	z_t	z_s	z
VAE	N/A	N/A	27.2
LDA	N/A	N/A	30.44
DVAE	N/A	N/A	42.4
TATGM	34.36	35.37	46.03
Ours(T=10)	43.81±0.78	46.97±0.29	47.28±0.58
Ours(T=30)	45.28±0.85	46.56±0.48	47.81±0.47
Ours(T=50)	46.25±0.59	**48.09±0.38**	**48.75±0.42**
Ours(T=50) w/o Dis	**47.09±0.27**	46.56±0.84	48.06±0.84
Ours(T=50) w/o $\mathcal{L}_{\text{info}}$	43.22±0.45	45.69±0.63	47.12±1.13

Table 6: Latent classification accuracy on Yelp15. N/A means not applicable for the current method.

Our model exhibits competitive scores compared with baselines. In result, the topic modeling component as an independent topic model to be a side product of our model is qualified.

3.3.4 Document-Level Topic Entropy

Topic entropy (Rezaee and Ferraro, 2020) reflects the concentration degree of a topic model. By calculating the entropy value of the topic latent representations, we can obtain the focus intensity of the topic modeling part with different documents. The lower entropy is, the less topics a topic model infers for one document, i.e., the higher concentration level for one script. From Table 5, we find that our model performs well among different baselines. Besides, both advanced objectives make efforts to form the topic modeling component a more dedicated one. To verify the validity of conditioning z_t on expressive z_s, we additionally display topic entropy value without flow approximation. It is very obvious that, flexible z_s largely prompts topic expression of the model. All in all, these make clear that TA-VAE is competent to provide consistent and accurate topic analyses.

3.3.5 Latent Codes Classification

Do latent codes really distinguish different text attributes? To answer that question, we conducted a supervised classification task on latent variables of various types on Yelp15. Higher the accuracy is, more precise topic guidance TA-VAE captures.

Specific experimental setting can be found in Appendix A.3.6. From Table 6 we can draw the following conclusions: firstly, the proposed TA-VAE model under different settings takes top positions regarding to the test accuracy, which demonstrates the advantage of our model to learn attribute knowledge from its latent spaces. Secondly, both topic-aware objectives contribute to distinct senti-

#1	#2	#3	#4	#5	#6	#7	#8	#9
gay	iraq	57-year	plane	tea	rain	deputies	mark	museum
marriage	soldier	19-year	crashed	gop	rains	deputy	staff	art
anti	syria	collision	miles	nomination	snow	commissioners	clinton	festival
ruling	troops	21-year	wildfire	democrat	unemployment	maricopa	lead	music
congress	forces	tractor	engine	challenger	storms	patrol	elections	zoo

Table 7: Top-5 topic words from nine topics generated by 50 topic TA-VAE models on APNEWS (cherry-picked).

Int. 1	•ok . the waiter was rude to us , we did n't know what we wanted to do with our food ... we were told that they were not busy at all
Int. 2	•very disappointing . the only thing that was not the best thing about this place is that they do n't care about the quality of the food ! ! ! we were not impressed with the service , food was bad , service was horrible .
Int. 3	•not very disappointed . the only thing that was not the best thing about this place is that they do n't care about the quality of the food ! ! ! we were not impressed with the service , food was good , service was horrible . we . will be back to try their <unk>
Int. 4	•not bad . the food was not bad , we had to ask for the <unk>sauce . we were told that they were not only to be able to get our food to be delivered . we were told that they were n't even busy , but we were not impressed with the service . we will be back to try this place again !
Int. 5	•not bad . the food was not bad , but the <unk>was not too salty . we were told that they were n't even able to get our food to be delivered to the kitchen . we were told that they were n't even busy . we had a great time to go to this place , the service was great !
Int. 6	•not bad at all ! the food was not bad at all ! the only thing i would say was that the service was great . we were greeted by the owner and he was very friendly and helpful . we will be back for sure .
Int. 7	•not sure what i wanted to say about this place but the service was great . we were in the area for a few minutes and they were very nice . they were very friendly and helpful . i would recommend this place to anyone who . likes the <unk>
Int. 8	•this place is amazing and the breakfast is delicious and the staff is very friendly . i will be back .
Int. 9	•this starbucks is my favorite breakfast spot , i have been to a few times . i have a good time and i have a good time . the coffee is very good and the staff is very friendly . i will be back .

Table 8: Text style transfer generation from **negative** to **positive** by traversing learned topic representation.

ments in sentences, but the **implicit informative penalty devotes more, which can be ascribed to the direct devotion in latent spaces of $\mathcal{L}_{\text{info}}$.** Moreover, statistics with only topic latent codes are sometimes inferior to accuracy inferred from sequence latent representations. We argue that, since labels in Yelp15 dataset are specified as sentiment attributes, a positive sentence may only differ from a negative sentence by several non-topic words (i.e., "happy" and "not happy"), which is more correlated with the sequential expression. Finally, different topic numbers give different outcomes. Models with 50 topic numbers reach the highest accuracy in three settings. While results with only topic representations get improved with the increase of topic numbers, results with only sequence latent seem to be less effected in this process. This can be naturally explained as a greater information capacity of z_t with a larger topic number.

3.3.6 Sentiment Transfer & Topic Words Generation

We expect each dimension of the latent representations derives a topic in texts. As a result, we conducted sentence generation tasks via latent traversal and interpolation to demonstrate the capability of learned knowledge of TA-VAE. As shown in Table 8, there is a sentiment transformation from negative to positive by traversing latent codes. Adjacent sentences share a similar context structure while gradually converted sentiment, that is to say, by manipulating expressive learned latent spaces, we could obtain effective implicit guidance for context generation while maintaining consistent structure. More textual examples are presented in Appendix A.4 due to the page length limit. Besides, we also selected 9 dimensions in the topic representation, and printed the top-5 topic words in Table 7.

4 Conclusion

We have proposed an unsupervised conditional text generation model TA-VAE, with theoretical justification on feasibility and remarkable empirical performance. TA-VAE proves a better generalization ability for language modeling with learned topic guidance based on the efficient latent dependency assumption and inference method of Householder flow. More importantly, TA-VAE demonstrates its superiority on validating the effectiveness of topic enhanced modifications with promising results in related tasks, and it can further derive meaningful learning representations to guide text generation.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296.

BNC Consortium et al. 2007. British national corpus. *Oxford Text Archive Core Collection*.

Chris Cremer, Xuechen Li, and David Duvenaud. 2018. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086. PMLR.

Shuyang Dai, Zhe Gan, Yu Cheng, Chenyang Tao, Lawrence Carin, and Jingjing Liu. 2020. Apo-vae: Text generation in hyperbolic space. *arXiv preprint arXiv:2005.00054*.

Adji B Dieng, Chong Wang, Jianfeng Gao, and John Paisley. 2016. Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*.

Xiaoan Ding and Kevin Gimpel. 2021. Flowprior: Learning expressive priors for latent variable sentence models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3242–3258.

Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. 2019. Implicit deep latent variable models for text generation. *arXiv preprint arXiv:1908.11527*.

Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. 2021. Transformer-based conditional variational autoencoder for controllable story generation. *arXiv preprint arXiv:2101.00828*.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *arXiv preprint arXiv:1903.10145*.

Cristina Garbacea and Qiaozhu Mei. 2020. Neural language generation: Formulation, methods, and evaluation. *arXiv preprint arXiv:2007.15780*.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. 2018. Dialogwae: Multimodal response generation with conditional wasserstein autoencoder. *arXiv preprint arXiv:1805.12352*.

Dandan Guo, Bo Chen, Ruiying Lu, and Mingyuan Zhou. 2020. Recurrent hierarchical topic-guided rnn for language generation. In *International Conference on Machine Learning*, pages 3810–3821. PMLR.

Alston S Householder. 1958. Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM (JACM)*, 5(4):339–342.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 633–644.

Parag Jain, Anirban Laha, Karthik Sankaranarayanan, Preksha Nema, Mitesh M Khapra, and Shreyas Shetty. 2018. A mixed hierarchical attention based encoder-decoder approach for standard table summarization. *arXiv preprint arXiv:1804.07790*.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020a. Optimus: Organizing sentences via pre-trained modeling of a latent space. *arXiv preprint arXiv:2004.04092*.

Jianqiao Li, Chunyuan Li, Guoyin Wang, Hao Fu, Yuhchen Lin, Liqun Chen, Yizhe Zhang, Chenyang Tao, Ruiyi Zhang, Wenlin Wang, et al. 2020b. Improving text generation with student-forcing optimal transport. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9144–9156.

Yujia Li, Kevin Swersky, and Rich Zemel. 2015. Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727. PMLR.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2015. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *arXiv preprint arXiv:1509.00838*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Mehdi Rezaee and Francis Ferraro. 2020. A discrete variational recurrent topic model without the reparametrization trick. *arXiv preprint arXiv:2010.12055*.

Danilo Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. *arXiv preprint arXiv:1702.02390*.

Hongyin Tang, Miao Li, and Beihong Jin. 2019. A topic augmented text generation model: Joint learning of semantics and structural features. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5090–5099.

Jakub M Tomczak and Max Welling. 2016. Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*.

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*, pages 356–365. PMLR.

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational auto-encoder for text generation. In *NAACL-HLT (1)*.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2018. Learning neural templates for text generation. *arXiv preprint arXiv:1808.10122*.

Yijun Xiao, Tiancheng Zhao, and William Yang Wang. 2018. Dirichlet variational autoencoder for text modeling. *arXiv preprint arXiv:1811.00135*.

Peng Xu, Jackie Chi Kit Cheung, and Yanshuai Cao. 2020. On variational learning of controllable representations for text without supervision. In *International Conference on Machine Learning*, pages 10534–10543. PMLR.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *International conference on machine learning*, pages 3881–3890. PMLR.

Ruiyi Zhang, Chunyuan Li, Changyou Chen, and Lawrence Carin. 2018. Learning structural weight uncertainty for sequential decision-making. In *International Conference on Artificial Intelligence and Statistics*, pages 1137–1146. PMLR.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2017a. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017b. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

Zachary Ziegler and Alexander Rush. 2019. Latent normalizing flows for discrete sequences. In *International Conference on Machine Learning*, pages 7673–7682. PMLR.

A Appendix

A.1 Proofs

We do the mathematical proof of reconstruction process in the topic modeling part, decomposition of $I(\boldsymbol{X}; \boldsymbol{z_t} \mid \boldsymbol{z_s})$ and the separation of KL divergence of two modeling parts in this section.

A.1.1 Reconstruction Process in the Topic Modeling Part

We assume $\boldsymbol{X}$ is the input text data, $\boldsymbol{\alpha}$ is the document-level topic parameter, $\boldsymbol{Y}$ is the output of the topic modeling component. Then the reconstruction of topic modeling part is:

$$
\begin{aligned}
& p(\boldsymbol{X} \mid \boldsymbol{\alpha}) = p(\boldsymbol{Y}) = \\
& \int_{z_t}\int_{z_s} p(\boldsymbol{z_t}) \left(\prod_{i=1}^{m} p(y_i \mid \boldsymbol{z_t}) p(\boldsymbol{z_t} \mid \boldsymbol{z_s}) p(\boldsymbol{z_s}) \right) d\boldsymbol{z_s} d\boldsymbol{z_t} \\
& = \int_{z_t}\int_{z_s} p(\boldsymbol{z_t}) \left(\prod_{i=1}^{m} p(y_i, \boldsymbol{z_s} \mid \boldsymbol{z_t}) \right) d\boldsymbol{z_s} d\boldsymbol{z_t} \\
& = \int_{z_t}\int_{z_s} p(\boldsymbol{z_t}) p(\boldsymbol{Y}, \boldsymbol{z_s} \mid \boldsymbol{z_t}) d\boldsymbol{z_s} d\boldsymbol{z_t} \\
& = \int_{z_t}\int_{z_s} p(\boldsymbol{Y}, \boldsymbol{z_s}, \boldsymbol{z_t}) d\boldsymbol{z_s} d\boldsymbol{z_t} \\
& = \int_{z_t}\int_{z_s} p(\boldsymbol{X}, \boldsymbol{z_s}, \boldsymbol{z_t} \mid \boldsymbol{\alpha}) d\boldsymbol{z_s} d\boldsymbol{z_t},
\end{aligned} \tag{10}
$$

The relation between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is $\boldsymbol{Y} = \boldsymbol{X} \mid \boldsymbol{\alpha}$. The second equation above can stand because of the approximation method of the marginal probability of a word in documents: $p(y_i \mid \boldsymbol{z_t}) p(\boldsymbol{z_t} \mid \boldsymbol{z_s}) p(\boldsymbol{z_s}) = p(y_i \mid \boldsymbol{z_t}) p(\boldsymbol{z_t}, \boldsymbol{z_s}) = p(y_i, \boldsymbol{z_s} \mid \boldsymbol{z_t})$.

A.1.2 From the Overall KL to Separate Modes

We will give a more intuitive explanation of the derivation of KL terms from separate modeling component (sequence and topic) in TA-VAE. The overall KL term of TA-VAE model under the paradigm of two VAEs can be modeled as:

$$
\mathbb{D}_{\mathrm{KL}}(q(\boldsymbol{z_t}, \boldsymbol{z_s} \mid \boldsymbol{X}) \| p(\boldsymbol{z_t}, \boldsymbol{z_s})), \tag{11}
$$

where we treat two different latent representations as one and calculate its regularization penalty using KL divergence. However, Eq.(11) can be factorized into two terms with regard to sequence and topic latents respectively, that is:

$$
\begin{aligned}
& \mathbb{D}_{\mathrm{KL}}(q(\boldsymbol{z_t}, \boldsymbol{z_s} \mid \boldsymbol{X}) \| p(\boldsymbol{z_t}, \boldsymbol{z_s})) \\
& = q(\boldsymbol{z_t}, \boldsymbol{z_s} \mid \boldsymbol{X}) \log \left[q(\boldsymbol{z_t}, \boldsymbol{z_s} \mid \boldsymbol{X}) \right] - \log \left[p(\boldsymbol{z_t}, \boldsymbol{z_s}) \right] \\
& = q(\boldsymbol{z_t}, \boldsymbol{z_s} \mid \boldsymbol{X}) \log \left[\frac{q(\boldsymbol{z_t}, \boldsymbol{z_s}, \boldsymbol{X})}{q(\boldsymbol{z_s}, \boldsymbol{X})} \cdot \frac{q(\boldsymbol{z_s}, \boldsymbol{X})}{q(\boldsymbol{X})} \right] \\
& - q(\boldsymbol{z_t}, \boldsymbol{z_s} \mid \boldsymbol{X}) \log \left[\frac{p(\boldsymbol{z_t}, \boldsymbol{z_s})}{p(\boldsymbol{z_t})} \cdot p(\boldsymbol{z_t}) \right] \\
& = q(\boldsymbol{z_t}, \boldsymbol{z_s} \mid \boldsymbol{X}) \{ \log \left[q(\boldsymbol{z_t} \mid \boldsymbol{z_s}, \boldsymbol{X}) \right] - \log \left[p(\boldsymbol{z_t} \mid \boldsymbol{z_s}) \right] \} \\
& + q(\boldsymbol{z_s} \mid \boldsymbol{X}) \{ \log \left[q(\boldsymbol{z_s} \mid \boldsymbol{X}) \right] - \log p(\boldsymbol{z_s}) \} \\
& = q(\boldsymbol{z_s} \mid \boldsymbol{X}) q(\boldsymbol{z_t} \mid \boldsymbol{z_s}, \boldsymbol{X}) \log \frac{q(\boldsymbol{z_t} \mid \boldsymbol{z_s}, \boldsymbol{X})}{p(\boldsymbol{z_t} \mid \boldsymbol{z_s})} \\
& + q(\boldsymbol{z_s} \mid \boldsymbol{X}) \log \frac{q(\boldsymbol{z_s} \mid \boldsymbol{X})}{p(\boldsymbol{z_s})} \\
& = \underbrace{\mathbb{E}_{q(\boldsymbol{z_t} \mid \boldsymbol{X})} \left[\mathbb{D}_{\mathrm{KL}}(q(\boldsymbol{z_t} \mid \boldsymbol{X}, \boldsymbol{z_s}) \| p(\boldsymbol{z_t} \mid \boldsymbol{z_s})) \right]}_{\text{KL Term in Topic Modeling Component}} \\
& + \underbrace{\mathbb{D}_{\mathrm{KL}}(q(\boldsymbol{z_s} \mid \boldsymbol{X}) \| p(\boldsymbol{z_s}))}_{\text{KL Term in Sequence Modeling Component}} .
\end{aligned} \tag{12}
$$

The third equation can stand because we replace $q(\boldsymbol{z_t}, \boldsymbol{z_s} \mid \boldsymbol{X})$ with $q(\boldsymbol{z_s} \mid \boldsymbol{X})$ in the second term for the third equation. At last, we discover that the overall KL term of the system is well approximated by two distinct KL penalties related to components in TA-VAE model.

A.1.3 Decomposition of $I(\boldsymbol{X}; \boldsymbol{z_t} \mid \boldsymbol{z_s})$

To avoid inferring meaningless latent representations with regard to the true data $\boldsymbol{X}$, we add a mutual information maximization term between $\boldsymbol{X}$ and topic latent code $\boldsymbol{z_t}$. In practice, topic latent space is conditioned on sequence latent representation $\boldsymbol{z_s}$ in TA-VAE setup. So we calculate $I(\boldsymbol{X}; \boldsymbol{z_t} \mid \boldsymbol{z_s})$ instead.

$$
\begin{aligned}
& I(\boldsymbol{X}; \boldsymbol{z_t} \mid \boldsymbol{z_s}) \\
& = \int_{\boldsymbol{X}}\int_{\boldsymbol{z_t}} q((\boldsymbol{z_t} \mid \boldsymbol{z_s}), \boldsymbol{X}) \log \frac{q((\boldsymbol{z_t} \mid \boldsymbol{z_s}), \boldsymbol{X})}{q(\boldsymbol{z_t} \mid \boldsymbol{z_s}) q(\boldsymbol{X})} d\boldsymbol{z_t} d\boldsymbol{X} \\
& = \int_{\boldsymbol{X}}\int_{\boldsymbol{z_t}} q(\boldsymbol{z_t} \mid \boldsymbol{z_s}, \boldsymbol{X}) p(\boldsymbol{X}) \log \frac{q(\boldsymbol{z_t} \mid \boldsymbol{z_s}, \boldsymbol{X})}{q(\boldsymbol{z_t} \mid \boldsymbol{z_s})} d\boldsymbol{z_t} d\boldsymbol{X} \\
& = \int_{\boldsymbol{X}}\int_{\boldsymbol{z_t}} q(\boldsymbol{z_t} \mid \boldsymbol{z_s}, \boldsymbol{X}) p(\boldsymbol{X}) \left[\log \frac{q(\boldsymbol{z_t} \mid \boldsymbol{z_s}, \boldsymbol{X})}{p(\boldsymbol{z_t} \mid \boldsymbol{z_s})} \right] d\boldsymbol{z_t} d\boldsymbol{X} \\
& - \int_{\boldsymbol{X}}\int_{\boldsymbol{z_t}} q((\boldsymbol{z_t} \mid \boldsymbol{z_s}), \boldsymbol{X}) \left[\log \frac{q(\boldsymbol{z_t} \mid \boldsymbol{z_s})}{p(\boldsymbol{z_t} \mid \boldsymbol{z_s})} \right] d\boldsymbol{z_t} d\boldsymbol{X} \\
& = \mathbb{E}_{p(\boldsymbol{X})} \left[\int_{\boldsymbol{z_t}} q(\boldsymbol{z_t} \mid \boldsymbol{z_s}, \boldsymbol{X}) \left[\log \frac{q(\boldsymbol{z_t} \mid \boldsymbol{z_s}, \boldsymbol{X})}{p(\boldsymbol{z_t} \mid \boldsymbol{z_s})} \right] d\boldsymbol{z_t} \right] \\
& - \int_{\boldsymbol{z_t}} q(\boldsymbol{z_t} \mid \boldsymbol{z_s}) \left[\log \frac{q(\boldsymbol{z_t} \mid \boldsymbol{z_s})}{p(\boldsymbol{z_t} \mid \boldsymbol{z_s})} \right] d\boldsymbol{z_t} \\
& = \mathbb{E}_{p(\boldsymbol{X})} \left[\mathbb{D}_{\mathrm{KL}}(q(\boldsymbol{z_t} \mid \boldsymbol{X}, \boldsymbol{z_s}) \| p(\boldsymbol{z_t} \mid \boldsymbol{z_s})) \right] \\
& - \mathbb{D}_{\mathrm{KL}}(q(\boldsymbol{z_t} \mid \boldsymbol{z_s}) \| p(\boldsymbol{z_t} \mid \boldsymbol{z_s})).
\end{aligned} \tag{13}
$$

The whole continued equality can stand because we make the following assumption: we assume the observed data $\boldsymbol{X}$ has no direct impact on latent variable $\boldsymbol{z_s}$, which can explain the second decomposition equation. This is also the main reason for adding the auxiliary mutual information maximization between observed data and latent codes for effective inference. Besides, we approximate KL term in topic modeling part $(\mathbb{E}_{q(\boldsymbol{z_t}|\boldsymbol{X})}\left[\mathbb{D}_{\mathrm{KL}}(q(\boldsymbol{z_t} \mid \boldsymbol{X}, \boldsymbol{z_s}) \| p(\boldsymbol{z_t} \mid \boldsymbol{z_s}))\right])$ by the first KL penalty in the last equation from Eq.(13), which helps upgrade the holistic model ELBO in a uniform way. Finally the holistic ELBO of TA-VAE model is

$$\begin{aligned} \mathcal{L}_{\text{info}} &= \mathbb{D}_{\mathrm{KL}}(q(\boldsymbol{z_t} \mid \boldsymbol{z_s}) \| p(\boldsymbol{z_t} \mid \boldsymbol{z_s})), \\ \mathcal{L} &= \mathcal{L}_S + \mathcal{L}_T + \lambda_D \mathcal{L}_D - \lambda_{\text{info}} \mathcal{L}_{\text{info}}. \end{aligned} \tag{14}$$

A.2 Introduction of Flow-based VAE and Householder Transformation

A.2.1 Flow-based VAE

In recent years, *normalizing flow* (NF) (Rezende and Mohamed, 2015) as a practical framework to approximate flexible posterior distributions by starting with a relatively simple one (e.g., Gaussian) has been widely employed to generative models (Dinh et al., 2014, 2016). Formally, given an initial distribution $\mathcal{D}_0$ and a data point $\boldsymbol{z_0} \sim \mathcal{D}_0$, we aim to find the true and complex distribution $\mathcal{D}_K$ of data by orienting a specific variable $\boldsymbol{z_K}$ from it. This process should be accomplished by an invertible and intuitively complex function $f(\cdot)$, such that $f(\boldsymbol{z_0}) = \boldsymbol{z_K}$. To build the powerful modeling function $f(\cdot)$, a series of invertible transformations $F = \{f_i\}_{i=1}^{K}$ are stacked into a chain and applied on $\boldsymbol{z_0}$. Methodologically, they play the same role as $f(\cdot)$ with $\mathcal{D}_0$, that is: $f(\boldsymbol{z_0}) = \boldsymbol{z_K} \triangleq f_K(\ldots f_2(f_1(\boldsymbol{z_0})))$. The last iterate gives a random variable $\boldsymbol{z}_K$ with more flexibility. For a VAE-based generative model, the normalizing flow can be used to enrich the posterior of it with small or even none modifications in the architecture of the encoder and the decoder.

Constant invertible transformations on a data point are equivalent to coordinate changes of the system. As a result, once we choose the transformation $f(\cdot)$ for which the Jacobian-determinant can be computed, the training objective from Eq. (1) should be refactored as follow:

$$\begin{aligned} &\log p(\boldsymbol{X}) \\ &\geq \mathbb{E}_{q(\boldsymbol{z_0}|\boldsymbol{X})}\left[\log p\left(\boldsymbol{X} \mid \boldsymbol{z_K}\right) + \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k}{\partial \boldsymbol{z_{k-1}}} \right| \right] \\ &- \mathbb{D}_{\mathrm{KL}}\left(q\left(\boldsymbol{z_0} \mid \boldsymbol{X}\right) \| p\left(\boldsymbol{z_k}\right)\right), \end{aligned} \tag{15}$$

here the original latent code $\boldsymbol{z}$ is replaced by $\boldsymbol{z_K}$, which is more competent to build a flexible posterior distribution.

A.2.2 Householder Transformation

The *Householder transformation* (Householder, 1958) is defined as follows. For a given vector $\boldsymbol{z_{k-1}}$, the reflection hyperplane can be defined by a vector $\boldsymbol{v_k} \in \mathbb{R}^n$ (also known as *Householder vector*), which is orthogonal to the hyperplane. Then the reflection of $\boldsymbol{z_{k-1}}$ to $\boldsymbol{z_k}$ regard to the hyperplane can be described as (Tomczak and Welling, 2016):

$$\boldsymbol{z_k} = \boldsymbol{H_k} \cdot \boldsymbol{z_{k-1}} = (I - 2\frac{\boldsymbol{v_k}\boldsymbol{v_k}^{\mathrm{T}}}{\|\boldsymbol{v_k}\|^2}) \cdot \boldsymbol{z_{k-1}}, \tag{16}$$

where $\boldsymbol{H_k} = I - 2\frac{\boldsymbol{v_k}\boldsymbol{v_k}^{\mathrm{T}}}{\|\boldsymbol{v_k}\|^2}$ is called the *Householder Matrix*. Householder matrix is orthogonal, so the absolute value of its Jacobian determinant is always 1. This property also makes a Householder transformation to be volume-preserving.

A.3 Experimental Details

A.3.1 Dataset Details

We evaluated the performance of TA-VAE on five public corpora, namely APNEWS[1], IMDB (Maas et al., 2011), BNC (Consortium et al., 2007), PTB (Marcus et al., 1993) and Yelp15[2]. The first three corpora are the same datasets including the train, validation and test splits, as used by prior works, which are publicly available[3] and widely used. For the first four datasets (APNEWS, IMDB, BNC, PTB), we fixed the maximum sequence length to 80 and maximum vocabulary size to 40,000. For Yelp15, we followed the work in (Tang et al., 2019) and set the maximum sentence length to 150 while maximum vocabulary size to 20,000. In the preprocess procedure, we first used the publicly provided tokenizer and followed past works (Lau et al., 2017; Xiao et al., 2018; Tang et al., 2019) to lowercase all texts, then mapped the most frequent

[1] https://www.ap.org/en-gb/
[2] https://www.yelp.com/dataset
[3] https://github.com/jhlau/topically-driven-language-model

Dataset	#SM. Voc	#TM. Voc	#Training Docs	#Val. Docs	#Test Docs	#Avg. Len
APNEWS	22,760	7,498	50k	2k	2k	21.4
IMDB	27,764	5,829	75k	12.5k	12.5k	22.5
BNC	22,154	7,700	15k	1k	1k	22.6
PTB	9,733	4,498	42k	3.8k	3.4k	24.8
Yelp15	20,004	7,575	74k	7.4k	7.4k	75.3

Table 9: Statistical summary of five datasets.

Dataset	#1	#2	#3	#4	#5	#6	#7	#8	#9
IMDB	reviewers ridiculous total considering highly	poorly cinematography romance dialogue directing	debut finest beautifully stage romance	oscar terrific poorly independent talented	finished remote aged maker pre	toronto independent maker oscar debut	happened screening makers camera reviewers	twice yesterday funniest cable viewed	grade sub flicks fu kung
BNC	yesterday night today young just	council britain environmental meeting title	conservation environmental pollution council species	voice yesterday night daily post	award pounds ref research holder	africa pacific council asia east	international east european europe british	england cup voice britain league	environmental pollution conservation council environment
PTB	cost fiscal spending budget senate	composite counter volume ounce pence	mortgages adjustable capped yields rise	gains rise inflation orders percentage	futures traders short gains selling	nov oct priced mature dec	benchmark points priced treasury point	tuesday notes october september oct	nasdaq counter s&p activity decline
Yelp15	casino hotels strip mgm rooms	avec c'est des en que	massage pedicure gel nail polish	beers buffet tap burgers bartender	matcha milk bagel vanilla cupcake	min mins tip dirty 40	spa tub shower pool massage	cons pros buffet rooms rental	rooms suite amenities stayed pool

Table 10: Top-5 topic words from nine topics generated from 50 topic TA-VAE models (cherry-picked).

and infrequent words (those in the top 0.03% of frequency and appear less than 100 documents) to a special token (i.e. ⟨UNK⟩ token). We set the minimum frequency to 2 for all corpus except BNC, which was 8 to avoid over-fitting (Dieng et al., 2016) and expedite training process. The full statistics of datasets is presented in Table 1.

A.3.2 Overall Model Settings

We used pre-trained GloVe (Pennington et al., 2014) word vector to initialize the 200-dimensional word embedding layer. Bag-of-Word (BoW) encoder was a 2-layer feedforward neural network with 200 hidden units. The sequential encoder level Bi-LSTM had 2×300 hidden states, while the decoder LSTM had 300. Weight decay was set as 10^{-5} with dropout ratio 0.2 for all RNNs. The size of $\boldsymbol{z_s}$ was fixed to 32. We employed the Adam (Kingma and Ba, 2014) optimizer using a batch of 32 training samples and learning rate of 10^{-4} for all the model training. All models were trained for 80 epochs except the ones on BNC (100 epochs for adequate training) on a single GeForce GTX 1080Ti GPU. We set the max clip norm of gradient to 5.0 for avoiding gradiant explosion. Moreover, to take full advantage of learned latent knowledge as well as making topic modeling part to be more concentrated, we trained the model with $\lambda_S : \lambda_T = 1 : 3$ and used cyclical schedule (Fu et al., 2019) with 4 cycles through all training epochs for KL annealing. The weight of discriminator λ_D and informative penalty λ_{info} were 0.3 and 500 respectively followed infoVAE (Zhao et al., 2017a). As for Householder flow implementation, we formally followed the experimental settings in (Tomczak and Welling, 2016), but with the change that $q(\boldsymbol{z_{s(0)}})$ was a simple Gaussian with full covariance matrix. Finally, we assigned the pre-defined parameter τ in discriminator to 0.02 during training and 1.0 at inference stage as described in (Tang et al., 2019). In the generation procedure, we calculated text perplexity as the negative exponential value of the negative log-likelihood (NLL) averaged over the sum of words. We adopted models that perform the best on validation sets and reported results on test sets.

A.3.3 Implementation of Discriminator

In detail, we employ Gumbel-Softmax (Jang et al., 2016) for the implementation because of the inhos-

Dataset	Sampled Sentences
APNEWS	●virginia ’s largest school system is getting ready to raise a new tax increase . ●red cross - area residents are being hit by the winter storm . ●san francisco police officers are investigating a suspected of marijuana and a car that killed a man and injured two others in a rural area of san diego county . ●plant destroyed and wind gusts of winter weather . ●wounded castle county police are looking for a missing boater .
IMDB	●in the late 1980 ’s , i was never able to say that the film industry made a great deal . ●so bad - the plot line was very bad , to me , i know what this is about . ●a thoroughly entertaining thriller from beginning , i have no idea what the hell . ●it made that a great cast - like this , well - acted film . ●this movie reminds more kind of sort of science fiction of an <unk>of science fiction and science fiction of crap .
BNC	●when rail comes to the <unk>world cup qualifying <unk>at the end of the season . ●europe albania <unk>, political correspondent the government ’s largest government has been launched. ●award title : the structure of <unk>and social services, award type : research grant (project), award ref no : <unk>/ <unk>, award holder : dr r <unk>
PTB	●the company had been working with the state and financial services ’ plan ●this is n’t more efficient for people who want to get out ●a spokesman said it would be able to reduce the tax rate on the market
Yelp15	●avoid this place ! ! ! ! i will never go back . ●great place ! the best part of the strip is the free . the price is reasonable . ●a great selection of beers . they have a lot of options . ●server was rude , rude owner was rude , rude and unhelpful . i would n’t recommend this place to anyone looking for a good chinese food , but i would n’t go back to this place . ●really good ! i would recommend this place to anyone looking for a quick car wash and a great price for a quick bite and will be back !

Table 11: Generated sentences on five datasets from trained TA-VAE models (randomly sampled).

Int. 1	●have been here twice , and i have never had a bad experience . i had the chicken salad with garlic knots . the salad was delicious ! ! ! ! ! ! ! ! ! ! !
Int. 2	●i have been here twice , and i have never had a bad experience . i had the shrimp taco salad , which was delicious . i will be back ! ! ! ! ! ! ! ! ! !
Int. 3	●i have been here twice and have never been disappointed . the food was delicious , the fish tacos were delicious . i had the shrimp tacos , and the chicken was cooked perfectly .
Int. 4	●i have been to this location twice and have never been disappointed . the service is very friendly and helpful .
Int. 5	●i have n’t been to this location twice . the <unk>is very nice and helpful . the <unk>is located in the middle of the strip mall .
Int. 6	●i have n’t been to this location twice . pros : <unk>and <unk>. the <unk>was very nice and the service was great . i was in the area for a few days and it was n’t a bad experience .
Int. 7	●i have n’t been to this location twice . the <unk>was very nice and the service was great . i was n’t sure what to expect .
Int. 8	●i have n’t been to this location twice . i would have given a lot of money in the future , but i ’m not sure why the prices are reasonable .
Int. 9	●i think it ’s a bit overpriced . pros : <unk>:

Table 12: Text style transfer generation from positive to slightly negative by traversing learned topic representations (cherry-picked).

Type	Sentences
Org. I	**•the company and its executives deny the charges**
Rec. I	•the company had been working with the state and financial services and the government 's plan
Int. 1	•the company had no comment on the other hand and the state department said
Int. 2	•the company wants to keep the entire computer system says the agency
Int. 3	•these guys are a good idea he says
Int. 4	•these guys is an important and financial services he says
Rec. II	•you have a lot more efficient than he says
Org. II	**•our doors are open an nbc spokesman says**

Table 13: Generated sentences by interpolating latent codes.

pitality of discrete tokens for backpropagation. Our choice of discriminator can be depicted as follow:

- Gain the conditional probability of at the i-th time step $p(\hat{x}_i \mid \hat{x}_{1:i}, \boldsymbol{z}) = [p_1, p_2, ..., p_n]$,
- Obtain $a_i = \frac{\exp(\log(p_i)+g_i)/\tau}{\sum_{j=1}^{n}\exp(\log(p_j)+g_j)/\tau}$,
- Approximate the i-th reconstructed word by $\hat{x}_i = \boldsymbol{a}^T \boldsymbol{W_b}$,

here g_i and g_j are separately drawn from a Gumbel-Softmax distribution between 0 and 1. Parameter τ is set in advance during both training and inference stages. $\boldsymbol{a} = \{a_i\}_{i=1}^{n}$ is the vector for token approximation, while $\boldsymbol{W_b}$ denotes the BoW input from topic encoder. This setting has technical advantage compared with the discriminator in Tang et al. (2019), which transfers tokens by word embedding and inevitably demands the same size between the hidden layers of topic encoder and word embedding.

A.3.4 Implementation of Mutual Information Maximazation

In practice, we followed previous explorations, and replaced KL divergence in $\mathbb{D}_{\mathrm{KL}}(q(\boldsymbol{z_t} \mid \boldsymbol{z_s}) \| p(\boldsymbol{z_t} \mid \boldsymbol{z_s}))$ with another divergence Maximum-Mean Discrepancy (MMD) (Gretton et al., 2012; Li et al., 2015) that can be efficiently optimized over. Maximum-Mean Discrepancy efficiently quantifies the distance between two distributions using the kernel trick. For the given distributions q, p, and variables drawn from them $\boldsymbol{z} \sim p, \boldsymbol{z'} \sim q$ we approximated MMD term with the Gaussian kernel, that is:

$$\begin{aligned}\mathbb{D}_{\mathrm{MMD}}(p, q) = \mathbb{E}_{p(\boldsymbol{z})p(\boldsymbol{z'})}[k(\boldsymbol{z}, \boldsymbol{z'})] \\ + \mathbb{E}_{q(\boldsymbol{z})q(\boldsymbol{z'})}[k(\boldsymbol{z}, \boldsymbol{z'})] \\ - \mathbb{E}_{p(\boldsymbol{z})q(\boldsymbol{z'})}[k(\boldsymbol{z}, \boldsymbol{z'})],\end{aligned} \tag{17}$$

where the function $k(\cdot)$ is a Gaussian kernel.

A.3.5 NPMI Details

Given the top-n words of a topic, coherence is computed based on the sum of pairwise NPMI scores between topic words. We averaged topic coherence over the top $5/10/15/20$ topic words. To aggregate topic coherence scores, we calculated the mean coherence over topics (Dieng et al., 2016; Lau et al., 2017; Wang et al., 2019; Tang et al., 2019).

A.3.6 Classification Details

For any model to be tested, we first obtained the latent representations from a well-trained TA-VAE model with 10 flow layers of the training sets, then randomly sampled $2,000$ examples to train a 2-layer feedforward neural network with softmax function. As for final classification results, we recorded the model with highest accuracy on validation set for final result. We trained the classifier five times with every setup and reported the averaged classification accuracy as well as its standardized deviation.

A.4 Texts & Topic Words Generation

A.4.1 Generated Topics

For topic word generation, we used the decoder of topic modeling part to produce probability of each token in a corpora, and sorted words with the highest five probabilities as top-5 topic word output. We selected nine channels from TA-VAE models with 50 topic latent dimensions. And generated top-5 topic words from them severally. Results are shown in Table 10.

A.4.2 Sampled Texts

We randomly sampled sequence latent code $\boldsymbol{z_s}$ from its prior $N(0, I)$, and generated sentences from it on well-trained TA-VAE models on five datasets. Textual results are presented in Table 11.

A.4.3 Style Transfer Generation and Interpolated Sentences

For well-expressive attribute representation spaces, we expect they contain distinct attribute and can be easily manipulated. For sentence generation with transferred styles, we traversed the value in one latent dimension of latent variables from -10.0 to 10.0 by a step size of 2.0. Results in Table 12 show a transformation from positive sentiment to relatively negative (i.e., with negative expressions "n't been ... twice", "overpriced"). For interpolation task. We used linear interpolation strategy, this process can be specified as follows:

1. Given two samples x_i, x_j from train set.
2. Obtain their sequential latent code and topic latent code respectively $(z_{s(i)}, z_{t(i)}), (z_{s(j)}, z_{t(j)})$.
3. For both types of latent variables we use linear interpolation $z_{\text{type}} = z_{\text{type}(i)} \cdot (1-\tau) + z_{\text{type}(j)} \cdot \tau$ where $z_{\text{type}} \in \{z_s, z_t\}$ and τ increases from 0 to 1 by a step size of 0.2.

We can see there is a maintenance from the original text key phrases or structure (e.g., "the company", "lawmakers are consider", inverted form) and semantics (e.g., positive, business, law) as well as a transformation between two given examples. We can observe smooth and sensible interpolation results for almost arbitrary input pairs. This demonstrates our TA-VAE model learns meaningful latent spaces.

A.5 Full Statistical Results

A.5.1 Text Perplexity and KL Divergence

We present PPL values of models with varied flow layer numbers also with or without two auxiliary objectives respectively, as well as KL values of both modeling components (sequence and topic) from a top-down order in Table 15. For PPL results, our model outperforms all baselines on different settings. However, when flow layers are not elaborately designed (i.e., flow layer that is shallow for 5 layers or too deep for 20 layers), models with the proposed two auxiliary functions do not noticeably outperform models without them. As for observed KL values, firstly, models with medium-sized flow layers are more likely to reach a lower KL value in z_t, which is equivalent to a more competent topic modeling part. Secondly, sequential KL values are much lower than topic KL values. On the one hand, this can be attributed to a more powerful fitting tool (i.e., Householder flow) for sequential posterior to approximate the true distribution of its representation. On the other hand, as mentioned in (Tang et al., 2019), the topic information reveals much of the diversity of texts, which leads to higher KL values.

Type	Sentences
Org. 1	**•lawmakers are considering restrictions on harvesting a hawaii seafood <unk> known as <unk>.**
Rec. 1	•lawmakers are considering a bill that would link at least two dozen dogs dead inside a local airport .
Int. 1	•lawmakers are considering a bill that would link the south carolina town of marine corps on sunday night .
Int. 2	•the state 's government will be held on a las vegas strip - based weapons ring that killed in the u.s . house , but it does n't have a chance .
Int. 3	•the city of a florida man who died after being held by a fellow military veterans affairs in the nation 's largest valley .
Int. 4	•the man who died in a shooting that killed a tennessee valley business .
Rec. 2	•the man who shot a man in a downtown philadelphia house is now that he has received a plea deal .
Org. 2	**•a man who barricaded himself in his omaha home has surrendered without incident .**

Table 14: Generated sentences by interpolating latent codes.

A.5.2 Full Results of BLEU

We used benchmark tool Texygen (Zhu et al., 2018) to do all the BLEU-related calculations. We show results of our model only with or without discriminator, which we believe is more important for token-level upgrade, because the mutual information term is directly optimized in the topic latent space z_t, rather than in sequence embedding z_s or token level like the discriminator does. From the full results in Table 16, we can see that our model outperforms all baselines in *test*-BLEU metric, yet is only superior to other models on *self*-BLEU under B-2 in major cases. This phenomenon demonstrates that the proposed model is qualified to produce texts with high quality, but has difficulty in generating texts with high diversity. Nevertheless, the overall metric BLEU-F1 shows the superiority of TA-VAE model in a well weighted trade-off between text quality and diversity.

Model	APNEWS		IMDB		BNC		PTB	
	PPL	**KL**	**PPL**	**KL**	**PPL**	**KL**	**PPL**	**KL**
LSTM LM	64.13	-	72.14	-	102.89	-	116.2	-
LSTM+LDA	57.05	-	69.58	-	96.42	-	-	-
Topic-RNN	56.77	-	68.74	-	94.66	-	97.3	-
TDLM	53.00	-	63.67	-	87.42	-	-	-
LSTM VAE	71.60	0.83	86.16	2.78	105.10	0.13	79.8	9.6
TCNLM	52.75	-	63.98	-	87.98	-	-	-
TGVAE	48.73	3.55	57.11	5.02	87.86	4.57	-	-
DVAE	-	-	-	-	-	-	33.4	23.3
TATGM	47.23	2.90 8.18	52.01	3.87 9.34	80.78	2.54 7.76	-	- -
rGBN-RNN	42.71	-	51.36	-	79.13	-	-	-
VRTM	47.78	-	51.08	-	86.33	8.64	55.82	1.64
Ours(F=5)	36.48	0.20 4.86	37.48	0.16 13.9	78.11	2.30 23.00	27.40	0.28 13.74
Ours(F=5) w/o Dis	36.50	0.20 7.59	37.25	0.15 10.9	80.25	2.70 31.85	26.84	0.30 9.82
Ours(F=5) w/o $\mathcal{L}_{\text{info}}$	37.11	0.20 5.24	37.87	0.16 13.1	79.44	2.18 26.91	27.76	0.30 11.88
Ours(F=10)	36.35	0.20 5.31	36.53	0.14 12.3	76.34	4.68 9.58	27.25	0.26 8.18
Ours(F=10) w/o Dis	36.11	0.23 5.75	37.26	0.16 8.73	78.31	2.88 17.17	27.67	0.30 10.07
Ours(F=10) w/o $\mathcal{L}_{\text{info}}$	36.42	0.24 8.71	37.09	0.15 11.7	79.60	2.71 16.32	26.98	0.25 7.78
Ours(F=20)	36.08	0.21 5.58	35.75	0.12 8.18	78.45	2.88 9.45	26.94	0.27 13.04
Ours(F=20) w/o Dis	36.09	0.22 9.36	34.95	0.12 7.03	79.93	2.97 10.40	26.96	0.35 9.67
Ours(F=20) w/o $\mathcal{L}_{\text{info}}$	36.42	0.23 4.30	35.92	0.13 9.21	77.49	2.36 9.52	26.74	0.33 11.78

Table 15: Text quality analysis in terms of perplexity and KL value. Sequence and topic KL values are arranged in the top-down order.

Metrics	Methods	APNEWS				IMDB				BNC				PTB			
		B-2	B-3	B-4	B-5	B-2	B-3	B-4	B-5	B-2	B-3	B-4	B-5	B-2	B-3	B-4	B-5
test-BLEU↑	**VAE**	0.564	0.278	0.192	0.122	0.597	0.315	0.219	0.147	0.479	0.266	0.169	0.117	0.5215	0.3633	0.2642	0.1728
	VAE+HF	0.570	0.279	0.195	0.123	0.610	0.322	0.221	0.147	0.483	0.270	0.169	0.110	0.5565	0.3616	0.2529	0.1653
	TGVAE(F=10, T=10)	0.584	0.327	0.202	0.126	0.621	0.357	0.223	0.159	0.518	0.283	0.173	0.119	-	-	-	-
	TGVAE(F=10, T=30)	0.627	0.335	0.207	0.131	0.655	0.369	0.243	0.165	0.528	0.291	0.182	0.119	-	-	-	-
	TGVAE(F=10, T=50)	0.629	0.340	0.210	0.132	0.652	0.372	0.239	0.160	0.535	0.290	0.188	0.120	-	-	-	-
	Ours(F=10, T=10)	0.6512	0.3862	0.2358	0.1458	0.7202	0.4505	0.2470	0.1404	0.6997	0.5947	0.4934	0.3327	0.6824	0.4847	0.3564	0.2307
	Ours(F=10, T=30)	0.6434	0.3776	0.2374	0.1468	0.7037	0.4347	0.2566	0.1529	0.6791	0.5473	0.4502	0.3151	0.6705	0.4779	0.3438	0.2070
	Ours(F=10, T=50)	**0.6757**	0.3983	0.2432	**0.1514**	**0.7542**	**0.4753**	**0.2755**	**0.1620**	**0.7681**	**0.6610**	**0.5672**	0.4176	**0.6924**	**0.5076**	0.3733	0.2408
	Ours w/o Dis (F=10, T=50)	0.6596	**0.4100**	**0.2497**	0.1464	0.7447	0.4637	0.2678	0.1502	0.7316	0.6234	0.5292	**0.4215**	0.6484	0.4587	0.3297	0.2028
	Ours(F=5, T=50)	0.6449	0.3801	0.2241	0.1335	0.7136	0.4323	0.2444	0.1399	0.7397	0.6422	0.5521	0.3896	0.6599	0.4710	0.3407	0.2175
	Ours w/o Dis (F=5, T=50)	0.6531	0.3845	0.2204	0.1335	0.7221	0.4456	0.2498	0.1382	0.7283	0.6247	0.5323	0.4157	0.6870	0.5064	**0.3889**	**0.2604**
	Ours(F=20, T=50)	0.6558	0.3809	0.2187	0.1260	0.7374	0.4660	0.2543	0.1411	0.6744	0.5660	0.4818	0.3670	0.6790	0.5001	0.3661	0.2376
	Ours w/o Dis (F=20, T=50)	0.6522	0.3943	0.2274	0.1311	0.7255	0.4305	0.2418	0.1403	0.6538	0.5324	0.4265	0.2722	0.6391	0.4486	0.3149	0.1836
self-BLEU↓	**VAE**	0.866	0.531	0.233	-	0.891	0.632	0.275	-	0.851	0.510	0.163	-	0.8737	0.5411	0.2952	0.2359
	VAE+HF	0.873	0.552	0.219	-	0.902	0.648	0.262	-	0.845	0.520	0.163	-	0.8649	**0.4720**	**0.3162**	**0.2181**
	TGVAE(F=10, T=10)	0.839	0.512	0.172	-	0.889	0.577	0.242	-	0.829	0.488	0.151	-	-	-	-	-
	TGVAE(F=10, T=30)	0.811	0.478	0.157	-	0.850	0.560	0.231	-	0.806	0.473	0.150	-	-	-	-	-
	TGVAE(F=10, T=50)	0.808	**0.476**	**0.150**	-	0.842	**0.559**	**0.227**	-	0.793	**0.469**	**0.150**	-	-	-	-	-
	Ours(F=10, T=10)	0.7396	0.5659	0.4146	0.2927	0.7948	0.5950	0.3989	0.2423	0.8191	0.7718	0.7272	0.6798	0.7882	0.6598	0.5372	0.4149
	Ours(F=10, T=30)	**0.7091**	0.5093	0.3457	**0.2173**	**0.7783**	0.5674	0.3729	0.2298	0.8130	0.7358	0.6644	0.5924	**0.7575**	0.6009	0.4707	0.3413
	Ours(F=10, T=50)	0.7344	0.5373	0.3839	0.2309	0.7911	0.5878	0.3827	0.2346	**0.7851**	0.7407	0.6924	0.6297	0.7695	0.6350	0.5092	0.3806
	Ours w/o Dis (F=10, T=50)	0.7289	0.5635	0.4212	0.2792	0.8004	0.5973	0.3967	0.2475	0.7882	0.7417	0.6974	0.6420	0.7798	0.6305	0.4961	0.3663
	Ours(F=5, T=50)	0.7434	0.5599	0.3863	0.2590	0.7834	0.5745	0.3795	0.2392	0.8033	0.7565	0.7101	0.6524	0.7641	0.6097	0.4792	0.3546
	Ours w/o Dis (F=5, T=50)	0.7588	0.5891	0.4215	0.2773	0.7943	0.5796	0.3648	**0.2107**	0.8142	0.7549	0.6942	0.6306	0.7678	0.6320	0.5178	0.3854
	Ours(F=20, T=50)	0.7516	0.5799	0.4147	0.2768	0.8109	0.6008	0.3914	0.2443	0.8107	0.7552	0.7097	0.6650	0.7606	0.6127	0.4810	0.3461
	Ours w/o Dis (F=20, T=50)	0.7512	0.5735	0.4118	0.2708	0.7949	0.5723	0.3674	0.2262	0.8312	0.7788	0.7267	0.6728	0.7764	0.6442	0.5257	0.4055
BLEU-F1↑	**VAE**	0.2166	0.3491	0.3071	-	0.1843	0.3394	0.3364	-	0.2273	0.3448	0.2812	-	0.2033	0.4055	0.3843	0.2819
	VAE+HF	0.2077	0.3439	0.3121	-	0.1689	0.3363	0.3401	-	0.2242	0.3456	0.2809	-	0.2174	0.4292	0.3692	0.2729
	TGVAE(F=10, T=10)	0.2524	0.3916	0.3248	-	0.1883	0.3872	0.3446	-	0.2571	0.3645	0.2874	-	-	-	-	-
	TGVAE(F=10, T=30)	0.2904	0.4081	0.3324	-	0.2441	0.4014	0.3693	-	0.2837	0.3750	0.2998	-	-	-	-	-
	TGVAE(F=10, T=50)	0.2942	0.4124	0.3368	-	0.2544	0.4036	0.3651	-	0.2985	**0.3751**	0.3079	-	-	-	-	-
	Ours(F=10, T=10)	0.3720	0.4088	0.3362	0.2418	0.3193	0.4265	0.3501	0.2369	0.2875	0.3299	0.3513	0.3264	0.3233	0.3998	0.4027	0.3309
	Ours(F=10, T=30)	**0.4007**	0.4268	0.3484	0.2473	**0.3371**	0.4337	0.3642	0.2551	0.2933	0.3564	0.3845	0.3554	**0.3562**	0.4350	0.4168	0.3149
	Ours(F=10, T=50)	0.3813	**0.4281**	0.3487	**0.2530**	0.3272	**0.4415**	**0.3809**	**0.2673**	**0.3358**	0.3725	**0.3989**	**0.3925**	0.3459	0.4246	0.4241	0.3468
	Ours w/o Dis (F=10, T=50)	0.3842	0.4228	**0.3490**	0.2434	0.3148	0.4310	0.3709	0.2505	0.3284	0.3653	0.3850	0.3872	0.3287	0.4093	0.3986	0.3072
	Ours(F=5, T=50)	0.3671	0.4079	0.3283	0.2262	0.3323	0.4289	0.3507	0.2364	0.3108	0.3531	0.3802	0.3674	0.3475	0.4269	0.4119	0.3254
	Ours w/o Dis (F=5, T=50)	0.3523	0.3973	0.3193	0.2255	0.3203	0.4326	0.3586	0.2352	0.2960	0.3521	0.3884	0.3912	0.3471	0.4263	**0.4335**	**0.3658**
	Ours(F=20, T=50)	0.3603	0.3996	0.3185	0.2145	0.3010	0.4300	0.3587	0.2379	0.2956	0.3418	0.3623	0.3503	0.3540	**0.4364**	0.4293	0.3486
	Ours w/o Dis (F=20, T=50)	0.3602	0.4098	0.3280	0.2222	0.3197	0.4290	0.3498	0.2376	0.2683	0.3125	0.3330	0.2972	0.3313	0.3969	0.3785	0.2806

Table 16: Full BLEU result in terms of *test*-BLEU, *self*-BLEU and BLEU-F1 scores.