Improving Variational Autoencoder for Text Modelling with Timestep-Wise Regularisation

Ruizhe Li[♠], **Xiao Li**[♠], **Guanyi Chen**[♡], **Chenghua Lin**[♠*]
[♠]Department of Computer Science, University of Sheffield
[♡]Department of Information and Computing Sciences, Utrecht University
{r.li, xiao.li, c.lin}@sheffield.ac.uk, g.chen@uu.nl

Abstract

The Variational Autoencoder (VAE) is a popular and powerful model applied to text modelling to generate diverse sentences. However, an issue known as posterior collapse (or KL loss vanishing) happens when the VAE is used in text modelling, where the approximate posterior collapses to the prior, and the model will totally ignore the latent variables and be degraded to a plain language model during text generation. Such an issue is particularly prevalent when RNN-based VAE models are employed for text modelling. In this paper, we propose a simple, generic architecture called Timestep-Wise Regularisation VAE (TWR-VAE), which can effectively avoid posterior collapse and can be applied to any RNN-based VAE models. The effectiveness and versatility of our model are demonstrated in different tasks, including language modelling and dialogue response generation.

1 Introduction

Variational Autoencoders (VAE) (Kingma and Welling, 2014; Rezende et al., 2014), together with other deep generative models, including Generative Adversarial Networks (Goodfellow et al., 2014) and autoregressive models (Oord et al., 2018), have attracted a mass of attention in the research community as they have shown their ability to learn compact representations from complex, high-dimensional unlabelled data. VAEs have been widely used in many NLP tasks, such as text modelling (Bowman et al., 2016; Yang et al., 2017; Xu and Durrett, 2018; Fang et al., 2019; Li et al., 2019b), style transfer (Fang et al., 2019), and response generation (Zhao et al., 2017; Fang et al., 2019). In addition, VAEs are also useful to several downstream tasks, e.g., classification (Xu et al., 2017; Zhao et al., 2017; Li et al., 2019c; Gururangan et al., 2019), transfer learning (Higgins et al., 2017), etc.

However, there is a challenging optimisation issue of VAEs known as posterior collapse (a.k.a. KL loss vanishing), where the variational posterior collapses to the prior and the latent variable is ignored by the model during generation (Bowman et al., 2016). This is particularly prevalent when employing VAE-RNN architectures for text modelling. When posterior collapse happens, the decoder will be downgraded to a simpler language model and the VAE cannot learn good latent representations of data (Sønderby et al., 2016; Yang et al., 2017). Different strategies have been proposed to address this issue, such as annealing the KL term in the VAE loss function (Bowman et al., 2016; Sønderby et al., 2016; Fu et al., 2019), replacing the recurrent decoder with convolutional neural networks (CNNs) (Yang et al., 2017; Semeniuta et al., 2017), using a sophisticated prior distribution such as the von Mises-Fisher (vMF) distribution (Xu and Durrett, 2018); and adding mutual information into the VAE objectives (Phuong et al., 2018). While the aforementioned strategies have shown effectiveness in tackling the posterior collapse issue to some extent, they either require careful engineering between the reconstruction loss and the KL loss (Bowman et al., 2016; Sønderby et al., 2016; Fu et al., 2019), or designing more sophisticated model structures (Yang et al., 2017; Semeniuta et al., 2017; Xu and Durrett, 2018; Phuong et al., 2018).

In this paper, we propose a simple and robust architecture called Timestep-Wise Regularisation VAE (TWR-VAE), which can effectively alleviate the VAE posterior collapse issue in text modelling. Existing

*Corresponding author

Proceedings of the 28th International Conference on Computational Linguistics, pages 2381–2397
Barcelona, Spain (Online), December 8-13, 2020

VAE-RNN models for text modelling only impose KL regularisation on the latent variable of the RNN encoder at the final timestep, forcing the latent variable to be close to a Gaussian prior. In contrast, our TWR-VAE imposes KL regularisation on the latent variables of every timestep of the RNN encoder, which we dub *timestep-wise regularisation*. We hypothesise that timestep-wise regularisation is crucial to avoid posterior collapse and to learn good representations of data, and allows a more robust model learning process. In addition, the proposed timestep-wise regularisation strategy is generic and in theory can be applied to any existing VAE-RNN models, e.g., vanilla RNN and GRU-based VAE models. TWR-VAE shares some similarity with existing VAE-RNN models, where the input to the decoder is the latent variable sample from the variational posterior at the final timestep of the encoder. While this is a reasonable design choice, we also explore two model variants of TWR-VAE, namely, TWR-VAE$_{\text{mean}}$ and TWR-VAE$_{\text{sum}}$. At each time step, both model variants sample a latent variable from the timestep dependent variational posterior of the encoder. TWR-VAE$_{\text{mean}}$ averages the sampled latent variables which is then used as input to the decoder, whereas TWR-VAE$_{\text{sum}}$ performs vector addition on the sampled latent variables instead.

To demonstrate the effectiveness of our method, we select a number of strong baseline models and conduct comprehensive evaluations in two benchmark tasks involving five public datasets. For the language modelling task, experimental results show that our TWR-VAE model can effectively alleviate the posterior collapse issue and consistently give better predictive performance than all the baselines as evidenced by both quantitative (e.g., negative log likelihood and perplexity) and qualitative evaluation. For the dialogue response generation task, our model yields better or comparable performance to the state-of-the-art baselines based on three evaluation metrics (i.e. BLEU (Zhao et al., 2017), BOW embedding (Liu et al., 2016) and Dist (Liu et al., 2016)). Manual examination also shows that the dialogue responses generated by our model are more diverse and contentful than the baselines, as well as being simpler in model design. Our two model variants also show comparable performance to the best baseline, although not as strong as TWR-VAE.

In summary, the contribution of our paper are three-fold: (1) we propose a simple and robust method, which can effectively alleviate the posterior collapse issue of VAE via timestep-wise regularisation; (2) our approach is generic which can be applied to any RNN-based VAE models; (3) our approach outperforms the state-of-art on language modelling and yields better or comparable performance on dialogue response generation. The code of TWR-VAE is available at: https://github.com/ruizheliUOA/TWR-VAE.

2 Related Work

Variational autoencoder (Kingma and Welling, 2014; Rezende et al., 2014) yields great performance when it was applied to image generation (Razavi et al., 2019), facial attribute style transfer (Hou et al., 2017; Klys et al., 2018; Li et al., 2020), etc. It also has been applied to many natural language processing tasks, including text generation (Bowman et al., 2016; Fang et al., 2019; Zhu et al., 2020), dialogue response generation (Serban et al., 2017; Zhao et al., 2017; Park et al., 2018; Gu et al., 2019; Fang et al., 2019), and style transfer (John et al., 2019; Fang et al., 2019; Xu et al., 2020) For all these applications, there is a common issue called posterior collapse (or KL loss vanishing) (Bowman et al., 2016).

Several different types of methods were proposed to address this issue. KL annealing is the most common and basic solution used in almost all works (Bowman et al., 2016; Sønderby et al., 2016; Semeniuta et al., 2017; He et al., 2019; Fu et al., 2019; Fang et al., 2019). Another type of approaches attempt to weaken the decoder of VAE to avoid posterior collapse, such as introducing word dropout and historyless decoding into the decoder (Bowman et al., 2016), replacing the decoder with different CNNs (Yang et al., 2017; Semeniuta et al., 2017), and adding skip connections in the decoder (Dieng et al., 2019). Others tried to solve this issue by introducing new regularisers (Zhao et al., 2019; Goyal et al., 2017; Tolstikhin et al., 2018), using more sophisticated prior distributions (Tomczak and Welling, 2018; Xu and Durrett, 2018), etc.

More recently, Fu et al. (2019) used a cyclical annealing schedule to alleviate the KL loss vanishing issue. He et al. (2019) proposed a lagging inference network to update the encoder multiple times before a single decoder update to address the issue from the perspective of training dynamics. Zhu et al. (2020) applied the batch normalisation to the parameters of the approximate posterior and ensured that the lower

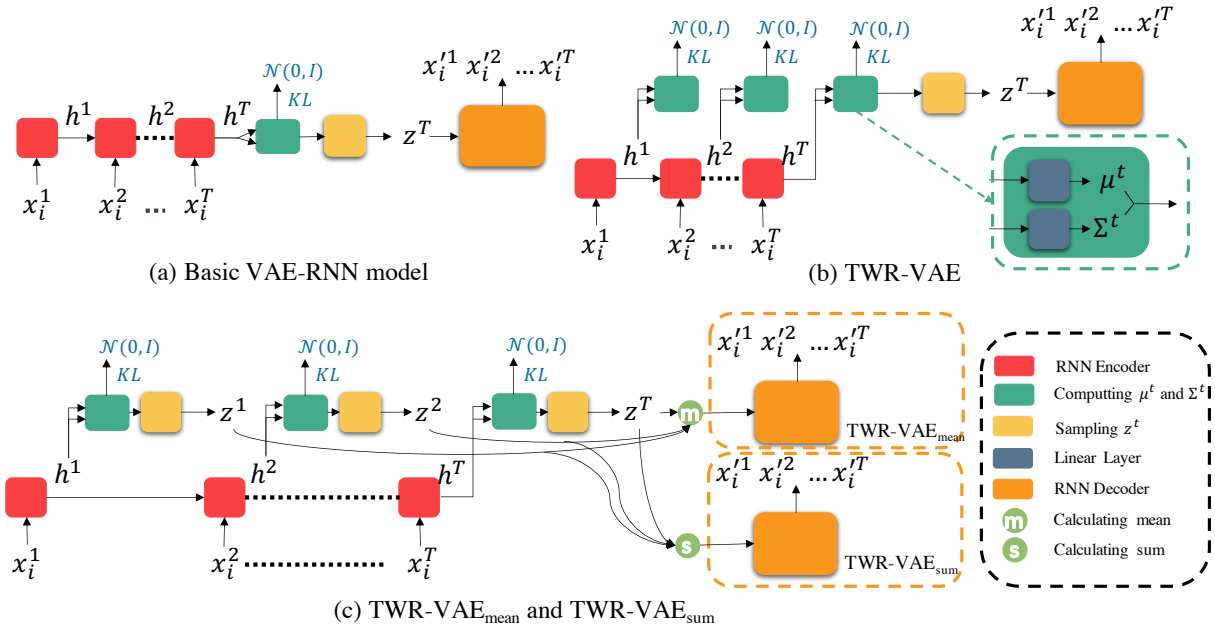


Figure 1: Architectures of the proposed TWR-VAE models and the basic VAE-RNN model.

bound of the expectation of the KL is positive to avoid posterior collapse. In contrast, our approach only imposes the KL regularisation on timestep-wise latent variables in the encoder, which is simpler without changing the VAE training mode, introducing more complicated posterior distributions or adding a KL annealing as a warm-up setup.

3 Methodology

In this section, we introduce the proposed Timestep-Wise Regularisation VAE (TWR-VAE) model as well as its two model variants. We briefly introduce the background of VAE before describing the technical details of the proposed models.

3.1 Background of VAE

A variational autoencoder is a generative model, which is designed to generate data via a latent variable $\mathbf{z}$. For a dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ with N i.i.d. data, there are two steps in the data generation process: (1) a latent variable $\mathbf{z}$ is sampled from a prior distribution $P_{\boldsymbol{\theta}}(\mathbf{z})$; (2) a data $\mathbf{x}_i$ is generated from the conditional distribution $P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z})$. We need to optimise the marginal likelihood $P_{\boldsymbol{\theta}}(\mathbf{x}_i) = \int P_{\boldsymbol{\theta}}(\mathbf{z})P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z})d\mathbf{z}$ using VAE. However, both of the marginal likelihood $P_{\boldsymbol{\theta}}(\mathbf{x}_i)$ and the true posterior distribution $P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_i) = P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z})P_{\boldsymbol{\theta}}(\mathbf{z})/P_{\boldsymbol{\theta}}(\mathbf{x}_i)$ are intractable. In order to train VAE, an encoder $Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)$ is used to approximate the true posterior $P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_i)$. In this way, a data $\mathbf{x}_i$ is encoded as a distribution of $\mathbf{z}$ via the encoder $Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)$ and the latent code $\mathbf{z}$ is fed into the decoder $P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z})$ to decode a distribution over some values of $\mathbf{x}_i$.

In general, the VAE is trained to maximise the marginal log likelihood $\log P_{\boldsymbol{\theta}}(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \sum_{i=1}^N \log P_{\boldsymbol{\theta}}(\mathbf{x}_i)$ for the whole training dataset. This is essentially equivalent to maximising the following evidence lower bound (ELBO)[1], which consists of two terms (Kingma and Welling, 2014):

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i) = \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z})] - D_{\mathrm{KL}}\left(Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)\|P(\mathbf{z})\right) . \quad (1)$$

The first term is the expected reconstruction error indicating how well the model can reconstruct data given a latent variable. The the second term is the KL-divergence of the approximate posterior from prior, i.e., a regularisation pushing the learned posterior to be as close to the prior as possible. The basic VAE-RNN model (Figure 1(a)) follows the aforementioned ELBO (i.e. Eq. 1). As the architecture of the

[1]See Appendix A for the full derivation.

encoder is a RNN, a latent variable (denoted as $\mathbf{z}^T$) is sampled from the variational posterior at the final timestep T, and then $\mathbf{z}^T$ is used as the input to the decoder. Therefore, the ELBO of a basic VAE-RNN model becomes:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i)_{\text{basic}} = \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T)] - D_{\text{KL}}\left(Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)\|P(\mathbf{z}^T)\right) . \tag{2}$$

Note that the total number of timestep T is also the length of the input sentence. As discussed, optimising ELBO (in Eq. 2) is prone to posterior collapsing to the prior (Bowman et al., 2016). This phenomenon happens when the second term of Eq. 2 would approach to its global minimum when $Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i) = P(\mathbf{z}^T)$, which results that $\mathbf{x}$ and $\mathbf{z}^T$ are two independent variables. As a result, the decoder (i.e., the reconstruction term) no longer depends on $\mathbf{z}^T$ and it fits the training data as a plain language model.

3.2 Variational Autoendoder with Timestep-Wise Regularisation (TWR-VAE)

In this section, we introduce the proposed Timestep-Wise Regularisation (TWR-VAE) model, a general architecture which can effectively mitigate the posterior collapse issue frequently observed in the VAE models with RNN-based backbone.

Our model design is motivated by one noticeable defect shared by the VAE-RNN based models in previous works (Bowman et al., 2016; Yang et al., 2017; Xu and Durrett, 2018; Dieng et al., 2019). That is, the general architecture of all these models, as shown in Figure 1(a), only impose a standard normal distribution prior on the last hidden state of the RNN encoder, which potentially leads to learning a suboptimal representation of the latent variable. In addition, to avoid posterior collapsing, it is important to learn good latent representations of data at the early stage of decoder training, so that the decoder can easily adopt them to generate controllable observations (Fu et al., 2019). Our hypothesis is that to learn a good representation of data, it is crucial to impose the standard normal prior on the hidden states of all timesteps of the RNN-based encoder, which will allow a better regularisation of the model learning process especially during the early stages.

The architecture of the proposed TWR-VAE model is shown in Figure 1(b), which is implemented using a one-layer LSTM for both the encoder and decoder. For each timestep t, we feed the hidden state $\mathbf{h}^t$ into two linear transformation layers for estimating $\boldsymbol{\mu}^t$ and $\boldsymbol{\Sigma}^t$, which are parameters of the variational posterior, i.e., a normal distribution corresponding to the $\mathbf{h}^t$. We then impose KL regularisation on all timestep-wise variational posteriors rather than posterior of the last timestep. Formally, given input $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, the ELBO of our model for each data pint $\mathbf{x}_i$ is defined as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i)_{\text{TWR}} = \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T)] - \frac{1}{T}\sum_{t=1}^{T} D_{\text{KL}}(Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})\|P(\mathbf{z}^t)) , \tag{3}$$

where T is the length of the input sentence, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are the parameters for the decoder and the encoder, respectively. Note that TWR-VAE is similar to existing VAE-RNN models (Xu and Durrett, 2018; Fu et al., 2019; He et al., 2019), which passes a single $\mathbf{z}^T$ at the final timestep to the decoder. However, there is a crucial difference that while existing models only impose KL regularisation on the last timestep, TWR-VAE imposes timestep-Wise KL regularisation and *average the KL loss over all timesteps*, i.e., the second term of Eq. 3. Such a strategy allows more robust model learning and can effectively mitigate posterior collapse (see §4 Experiment for detailed discussion). Compared to the HR-VAE of Li et al., (2019b), our model does not concatenate the cell state of the encoder at each timestep and the dimension of the latent variable of TWR-VAE is only 32, whereas for HR-VAE the dimension is 512 which is much larger. This enables the proposed TWR-VAE model to have fewer parameters than the HR-VAE. In addition, the training speed of the TWR-VAE is six times faster than the HR-VAE by paralleling the timestep-wise KL regularisation.

Following Kingma and Welling (2014), a reparameterisation trick is used to enable the timestep-wise latent variable sampling differentiable. During the gradients optimisation of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, we use Monte Carlo method (Metropolis and Ulam, 1949) to construct a Monte Carlo estimator, which can obtain unbiased

Dataset	Train	Dev.	Test	Vocab.
PTB	42,068	3,370	3,761	9.95K
Yelp15	100,000	10,000	10,000	19.76K
Yahoo	100,000	10,000	10,000	19.73K
SW	2,316	60	62	20K
DD	11,118	1,000	1,000	22K

Table 1: The statistics of the PTB, Yelp 2015, Yahoo, SW and DD datasets.

gradients of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ (see Appendices B and C for the detailed derivation):

$$\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}_i)_{\text{TWR}} \simeq \frac{1}{M}\sum_{m=1}^{M}\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_m^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}_m^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}_m^t)}\right)$$
$$\text{where} \quad \mathbf{z}_m^t = Q_{\boldsymbol{\phi}}(\mathbf{z}_m^t|\mathbf{x}_i^{1:t})\,, \tag{4}$$

Here M indicates the total number of times that we randomly sample $\mathbf{z}_m^t$ ($m \in [1:M]$) from the $Q_{\boldsymbol{\phi}}(\mathbf{z}_m^t|\mathbf{x}_i^{1:t})$ for approximation.

3.3 TWR-VAE$_{\text{mean}}$ and TWR-VAE$_{\text{sum}}$

In TWR-VAE, the input to the decoder is the latent variable sample from the variational posterior at the final timestep of the encoder. While this is a reasonable design choice, we also explore two model variants of TWR-VAE, namely, TWR-VAE$_{\text{mean}}$ and TWR-VAE$_{\text{sum}}$ (see Figure 1(c)). At each time step, both model variants sample a latent variable from the timestep dependent variational posterior of the encoder.

For TWR-VAE$_{\text{mean}}$, the timestep-wise latent variables $\{\mathbf{z}^t\}_{t=1}^T$ are sampled first and then they are averaged before feeding to the decoder. This leads to a different reconstruction loss of TWR-VAE$_{\text{mean}}$ compared to TWR-VAE (Eq. 3):

$$\mathbb{E}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\frac{1}{T}\sum_{t=1}^{T}\mathbf{z}^t)] \qquad \text{where } \mathbf{z}^t \sim Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t}) \tag{5}$$

For TWR-VAE$_{\text{sum}}$, it performs vector addition on the sampled latent variables $\{\mathbf{z}^t\}_{t=1}^T$ instead and the corresponding reconstruction loss is:

$$\mathbb{E}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\sum_{t=1}^{T}\mathbf{z}^t)] \qquad \text{where } \mathbf{z}^t \sim Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t}) \tag{6}$$

For both TWR-VAE$_{\text{mean}}$ and TWR-VAE$_{\text{sum}}$, their KL loss term is the same as TWR-VAE, i.e., $-\frac{1}{T}\sum_{t=1}^{T} D_{\text{KL}}(Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})\|P(\mathbf{z}^t))$.

4 Experiment

4.1 Language Modelling

We evaluate our TWR-VAE model on three public benchmark datasets, namely, Penn Treebank (PTB) (Marcus and Marcinkiewicz, 1993), Yelp15 (Yang et al., 2017), and Yahoo (Zhang et al., 2015), which have been widely used in previous work for text modelling (Bowman et al., 2016; Kim et al., 2018; Fu et al., 2019; He et al., 2019; Zhu et al., 2020). The statistics of the datasets are summarised in Table 1. We represent input data with 512-dimensional word2vec embeddings (Mikolov et al., 2013) and set the dimension of the hidden layers of both one-layer encoder and decoder to 256. Appendix D shows more details.

We compare our TWR-VAE model with five strong baselines[2]: **VAE-LSTM**: A VAE with LSTM and with KL annealing for tackling the posterior collapse issue (Bowman et al., 2016); (2) **SA-VAE**: A

[2]**VAE-LSTM**: https://github.com/timbmg/Sentence-VAE; **SA-VAE**: https://github.com/harvardnlp/sa-vae; **Cyclical VAE**: https://github.com/haofuml/cyclical_annealing; **Lagging VAE**: https://github.com/jxhe/vae-lagging-encoder; **BN-VAE**: https://github.com/valdersoul/bn-vae

Model	PTB				Yelp15				Yahoo			
	NLL↓	PPL↓	MI↑	KL	NLL↓	PPL↓	MI↑	KL	NLL↓	PPL↓	MI↑	KL
VAE-LSTM	101.2	101.4	0.0	0.0	357.9	40.6	0.0	0.0	328.6	61.2	0.0	0.0
SA-VAE	101.0	100.7	0.8	1.3	355.9	39.7	2.8	1.7	327.2	60.2	2.7	5.2
Cyc-VAE	102.8	109.0	1.3	1.4	359.5	41.3	1.0	2.0	330.6	65.3	2.0	2.1
Lag-VAE	100.9	99.8	0.8	0.9	355.9	39.7	2.4	3.8	326.7	59.8	2.9	5.7
BN-VAE (0.7)	100.2	96.9	**5.5**	7.2	355.9	39.7	**7.4**	9.1	327.4	60.2	**7.4**	8.8
TWR-VAE$_{\text{sum}}$	96.7	63.2	3.7	5.9	378.3	47.4	3.8	3.9	345.6	71.1	3.7	3.8
TWR-VAE$_{\text{mean}}$	95.6	60.4	3.9	4.9	361.7	40.0	3.9	3.5	324.8	55.0	4.1	4.8
TWR-VAE	**86.6**	**40.9**	4.1	5.0	**344.3**	**33.5**	4.1	3.1	**317.3**	**50.2**	4.1	3.3

Table 2: Language modelling results of all baselines and our models on the PTB, Yelp15 and Yahoo test datasets. The results of all baselines are reported based on (Li et al., 2019a; Zhu et al., 2020). ↓ denotes lower the better and ↑ higher the better.

VAE using stochastic variational inference to refine the variational parameters initialised by Amortized variational inference (Kim et al., 2018); (3) **Cyclical VAE**: A VAE employing cyclical annealing to alleviate the posterior collapse issue (Fu et al., 2019); (4) **Lagging VAE**: A VAE updating the encoder more times than updating the decoder (He et al., 2019); (5) **BN-VAE**: A VAE utilising Batch Normalisation for the KL distribution (Zhu et al., 2020).

We report the performance on four metrics: negative log likelihood (NLL), perplexity (PPL), KL-divergence which measures the distance between two probability distributions, and the mutual information of the input $\mathbf{x}$ and the latent variable $\mathbf{z}$, which measures how much information of $\mathbf{x}$ is obtained by $\mathbf{z}$. Following Dieng et al. (2019) and He et al. (2019), the mutual information is formulated as $I(\mathbf{x}, \mathbf{z}) = \mathbb{E}_{\mathbf{x}}[D_{\text{KL}}\left(Q_\phi(\mathbf{z}^T|\mathbf{x})\|P(\mathbf{z}^T)\right)] - D_{\text{KL}}(Q_\phi(\mathbf{z}^T)\|P(\mathbf{z}^T))$, where $Q_\phi(\mathbf{z}^T)$ is an aggregated posterior and $D_{\text{KL}}(Q_\phi(\mathbf{z}^T)\|P(\mathbf{z}^T))$ is the KL divergence between the aggregated posterior and the prior estimated by Monte Carlo estimators (see Appendix E for the whole derivation).

Results. As depicted in Table 2, our TWR-VAE outperforms all baselines on all datasets. Compared to the strongest baseline BN-VAE, our model reduces NLL by 11.8 and PPL by 24.1 on average across three datasets, showing superior performance in reconstructing input sentences. As shown in Table 2, the two variants of TWR-VAE also yields better performance to the baselines. For instance, TWR-VAE$_{\text{mean}}$ outperforms all baselines on PTB and Yahoo datasets and yield comparable results to BN-VAE on Yelp. This shows the effectiveness of our strategy of regularising timestep-wise variational posteriors.

Model generalisability and Ablation studies. We also evaluate the model's generalisability by looking at how well our timestep-wise regularisor works in different RNN architectures. To this end, we tested Basic-VAE$_{\text{RNN}}$ and Basic-VAE$_{\text{GRU}}$ (i.e., vanilla RNN and GRU model with KL annealing), as well as TWR-VAE$_{\text{RNN}}$ and TWR-VAE$_{\text{GRU}}$ (vanilla RNN and GRU with the timestep-wise regularisor). Experimental results in Table 3 show that our TWR models outperform the corresponding basic models on all evaluation metrics, regardless the encoder architecture. This shows the generalisability of our proposed architecture.

In addition, to understand how the proportion of timesteps that are imposed with KL regularisation impacts the performance of our model, we run a battery of experiments with varying proportion settings. Concretely, we impose KL regularisation on the last 25%, 50%, and 75% timesteps of the encoder of TWR-VAE, respectively. (**NB**: the KL regularisation is imposed on the final timestep for all model variants). The results in Table 3 show that TWR-VAE$_{\text{LSTM-last25}}$ has the lowest performance on NLL and PPL and the performance goes up along with higher proportion of timesteps being imposed with KL regularisation. In addition, when comparing these three model variants with the baseline VAE-LSTM (which only imposes the KL regularisation on the final timestep), our models can effectively mitigate posterior collapse. This observation embodies that imposing the KL regularisation on earlier timesteps is an effective strategy for mitigating posterior collapse. Moreover, the more timesteps we impose the KL regularisation on, the better performance the model can yield (in terms of NLL and PPL).

Latent representation interpolation. We perform latent representation interpolation to assess how well the latent space ($\mathbf{z}$) can be learned by TWR-VAE comparing to the strongest baseline BN-VAE. Given a pair of sentences $\mathbf{x}_1$ and $\mathbf{x}_2$, we sample their latent codes $\mathbf{z}_1^T$ and $\mathbf{z}_2^T$ from the encoder, and interpolate

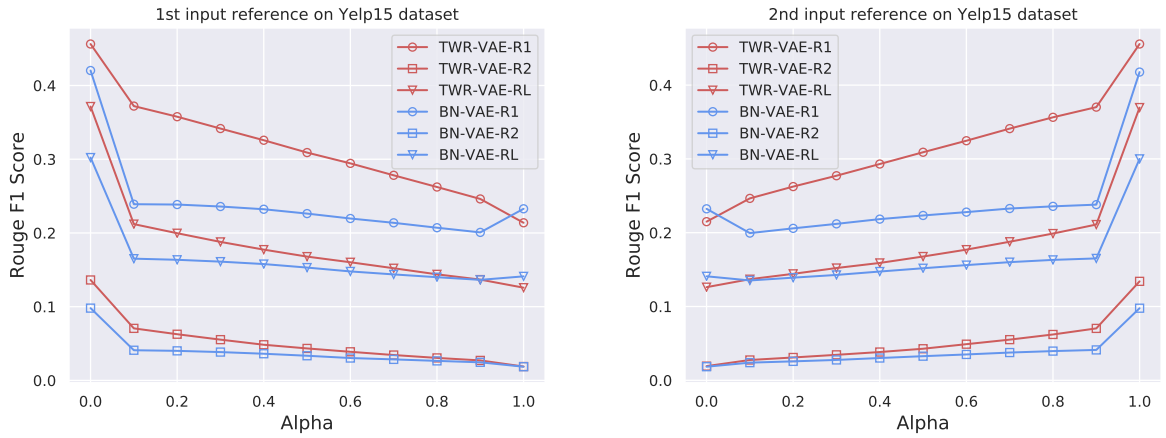
Model	Yelp15				Yahoo			
	NLL↓	PPL↓	MI↑	KL	NLL↓	PPL↓	MI↑	KL
Basic-VAE$_{\text{RNN}}$	399.2	58.7	0.0	0.0	363.9	89.1	0.0	0.1
TWR-VAE$_{\text{RNN}}$	395.4	56.4	3.9	0.5	363.0	88.2	4.1	0.6
Basic-VAE$_{\text{GRU}}$	389.6	53.2	0.6	0.6	355.0	79.9	2.3	2.6
TWR-VAE$_{\text{GRU}}$	360.9	39.7	**4.2**	3.3	336.9	63.9	**4.2**	3.7
TWR-VAE$_{\text{LSTM-last25}}$	360.4	39.5	4.1	8.3	338.2	64.9	**4.2**	8.4
TWR-VAE$_{\text{LSTM-last50}}$	356.2	37.9	4.1	5.1	331.7	59.9	**4.2**	5.3
TWR-VAE$_{\text{LSTM-last75}}$	352.6	36.5	4.1	3.7	321.0	52.5	4.1	4.1
TWR-VAE	**344.3**	**33.5**	4.1	3.1	**317.3**	**50.2**	4.1	3.3

Table 3: Ablation study results of all variants of our model on the Yelp15 and Yahoo test datasets.

Yelp15	Input 1	this is the worst restaurant experience i 've ever had ! not only is this place super slow in service but the food was not fresh !
	Input 2	i went to this place last month with my best friend and the food was good i love the coffee designs and the service was friendly .
BN-VAE	$\alpha = 0$	this place the worst restaurant i i have ever had . i only was the restaurant a overpriced , the , the food is not good and i
	$\alpha = 0.2$	this place joke ! the food was ok the was horrible . i ask for drink and came back to me . i will go back .
	$\alpha = 0.4$	this place joke ! the food was good horrible . i ask for a drink and check on me . i ask for a drink and check on me .
	$\alpha = 0.6$	i was try this place. disappointed . the food was not good it was just ok . the service was good the food was not price .
	$\alpha = 0.8$	i went lunch and the chicken and waffles . the food was good the service was horrible . i will go back .
	$\alpha = 1$	i went here this place for night and my family friend and i food was great . had the atmosphere and and the service was great . i
TWR-VAE	$\alpha = 0$	this is the worst restaurant i 've ever been ! service only was we restaurant was slow service but the food was not fresh !
	$\alpha = 0.2$	i love this place the food was very slow ! service is always slow and the food is not a good value so this was not my first choice .
	$\alpha = 0.4$	i have never been in this restaurant before the food was just ok and the service is very slow ! i will not continue to go back to this place .
	$\alpha = 0.6$	i have been here a few times now and the food was good ! ! ! the food is good and i would recommend to and return
	$\alpha = 0.8$	i went here this past weekend to see how good the food was and my husband had the same thing i would recommend for the price .
	$\alpha = 1$	i went to this place for night and my family friend and the food was good and would the service and the service was friendly .

Table 4: An example of interpolating the latent representation of two input sentences using BN-VAE and TWR-VAE in Yelp15 testset (see the example of Yahoo testset in appendix G).

them with $\mathbf{z}_\alpha^T = \mathbf{z}_1^T \cdot (1-\alpha) + \mathbf{z}_2^T \cdot \alpha$. Table 4 shows an example outputs by varying mixture weight α. It can be observed that our model learns representations which are more smooth than BN-VAE, where the sentences generated based on continuous samples from the latent code space preserve more consistent topical information in the neighbourhood of the path. There are less _UNK tokens occurring in generated sentences of our model, which implies that the quality of representations learned in our model is better than ones in BN-VAE. In addition to qualitative evaluation, we also evaluate the outputs quantitatively with ROUGE (Lin, 2004), which compares the generated sentences against the human references. Concretely, for each sentence pair, we compute the ROUGE-1, ROUGE-2 and ROUGE-L F1 scores between two input sentences (i.e., references) and each interpolation sentence. The averaged ROUGE scores over all sentence pairs in the test set versus different α settings are sketched in Figure 2. It can observed that as the mixture weight α increases, the ROUGE values of our model smoothly decrease w.r.t. the first reference and increase for the second one, showing a smooth transition of sentence interpolation. One can also note that our model has higher ROUGE scores than BN-VAE at $\alpha = 0$ for reference one and at $\alpha = 1$ for reference two, showing that our model is able to better learning latent representations and reconstructing the input sentences.



(a) (b)

Figure 2: The average ROUGE-1, ROUGE-2 and ROUGE-L F1 scores between two input references and 11 interpolations of each group using BN-VAE and TWR-VAE on Yelp15 test dataset (Appendix G shows the results on Yahoo dataset).

4.2 Dialogue Response Generation

In addition to language modelling, we further evaluate how well our proposed architecture could help alleviating the problem of "generic response" in Dialogue Systems (Huang et al., 2020; Wang et al., 2020). Dialogue systems that are built upon the sequence-to-sequence (seq2seq) model were found tend to generate generic and dull responses, such as "*I don't know*" or "*thank you*" (Li et al., 2016). One effective solution is using a more flexible intermediate representation between the encoder and the decoder of a seq2seq model with the help of a VAE, which models dialogue as a one-to-many problem and, therefore, can generate less generic responses. Such VAE-based dialogue response generators, similar to Shen et al. (2018), also face the problem of posterior collapse. Zhao et al. (2017) first addressed this issue by proposing the conditional VAE (CVAE) model which utilises KL annealing and Bag-of-Word loss. To test TWR-VAE on the dialogue response generation task, we extend TWR-VAE following the architecture of CVAE.

We represent each dialogue conversation as a combination of the dialogue context $\mathbf{c}$ (context window size J), the response utterance $\mathbf{x}$ (the $J+1^{th}$ utterance), and a latent representation $\mathbf{z}$ which encodes the information of the context and captures a latent distribution of valid responses. The dialogue response generation can then be defined as $P_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{c}) = \int P_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z},\mathbf{c})P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{c})d\mathbf{z}$. Here, a vatiational posterior $Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x},\mathbf{c})$ is used to approximate the true prior $P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{c})$. The ELBO of TWR-VAE can then be written as:

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}_i)_{\text{TWR}} = \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^J|\mathbf{x}_i,\mathbf{c})}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^J,\mathbf{c})] - \frac{1}{J}\sum_{j=1}^{J} D_{\text{KL}}(Q_{\boldsymbol{\phi}}(\mathbf{z}^j|\mathbf{x}_i,\mathbf{c})\|P_{\boldsymbol{\theta}}(\mathbf{z}^j|\mathbf{c}))\,. \quad (7)$$

Setup. We conducted experiment based on two popular benchmark datasets, namely, Switchboard (**SW**) (Godfrey and Holliman, 1997) and Dailydialog (**DD**) (Li et al., 2017b). For dataset statistics, please refer to Table 1. Following the implementation of CVAE, we pair each response with 10 context utterances (i.e. $J = 10$) from both speakers. The utterance encoder is a one-layer bidirectional GRU with 300 hidden size; both the context encoder and the decoder use a one-layer GRU with 300 hidden size. The dimension of the latent variable is 200. Appendix F shows more details.

Apart from comparing TWR-VAE to CVAE and iVAE, we further report the results of two other competitive models for dialogue response generation[3], i.e., **SeqGAN** (Li et al., 2017a) and a conditional

[3]**SeqGAN**:https://github.com/jiweil/Neural-Dialogue-Generation; **CVAE**:https://github.com/snakeztc/NeuralDialog-CVAE; **WAE**:https://github.com/guxd/DialogWAE; **iVAE**:https://github.com/fangleai/Implicit-LVM

Metrics	Switchboard					Dailydialog				
	SeqGAN	CVAE	WAE	iVAE	TWR-VAE	SeqGAN	CVAE	WAE	iVAE	TWR-VAE
BLEU-R↑	0.282	0.295	0.394	**0.427**	0.395	0.270	0.265	0.341	0.355	**0.407**
BLEU-P↑	**0.282**	0.258	0.254	0.254	0.258	0.270	0.222	0.278	0.239	**0.281**
BLEU-F1↑	0.282	0.275	0.309	**0.319**	0.312	0.270	0.242	0.306	0.285	**0.333**
BOW-A↑	0.817	0.836	0.897	**0.930**	0.921	0.918	0.923	0.948	0.951	**0.952**
BOW-E↑	0.515	0.572	0.627	**0.670**	0.654	0.495	0.543	0.578	**0.609**	0.603
BOW-G↑	0.748	0.846	0.887	**0.900**	**0.900**	0.774	0.811	0.846	**0.872**	0.865
Intra-dist1↑	0.705	0.803	0.713	0.828	**0.860**	0.747	**0.938**	0.830	0.897	0.921
Intra-dist2↑	0.521	0.415	0.651	0.692	**0.849**	0.806	0.973	0.940	0.975	**0.990**
Inter-dist1↑	0.070	0.112	0.245	0.391	**0.470**	0.075	0.177	0.327	**0.501**	0.497
Inter-dist2↑	0.052	0.102	0.413	0.668	**0.766**	0.081	0.222	0.583	**0.868**	0.817

Table 5: Dialogue response generation results of baselines and our model on SW and DD datasets.

Example 1: **Topic**: Care for the elderly **Context**: to have the responsibility of putting someone in a nursing home whose mind was not good and could not tell you if they were being < unk > or something it just would all be so different **Target**: uh - huh	
iVAE	TWR-VAE
1. yeah uh - huh	1. uh - huh
2. yeah and then go back up and go back and forth and go back again	2. i see yeah and they have to go back to work and it's really sad
3. right oh that makes	3. oh gosh they don't have to worry about
4. she's not	4. hm how do you feel
Example 2: **Topic**: Relationship **Context**: what happened , john ? **Target**: nothing .	
iVAE	TWR-VAE
1. oh , i am .	1. i can't sleep well .
2. what can we do for you ?	2. working overtime . i have been working on the weekend for a long time . i was terrified of getting a lot of headaches and i had a terrible hangover .
3. oh what's wrong ? i didn't know anyone .	3. oh , i am sorry . i had a terrible pain in the morning . i was so nervous . i couldn ' t find a chance to memorize the class . i was hoping to see you
4. i have to get my phone .	4. well , i am not sure of it .

Table 6: Four sample responses generated by iVAE and our model on SW (top) and DD (bottom) datasets, given context as input. Corresponding topic and target response (gold standard) are also listed. The generated utterances are different possible responses from two models. We only show the last utterance of the dialogue context here due to space limit (the actual context window is 10).

Wasserstein autoencoder called **WAE** (Gu et al., 2019). Following prior works (Gu et al., 2019; Fang et al., 2019), we report performance on three evaluation metrics including: (1) *BLEU* scores proposed by Zhao et al. (2017), which evaluates how many n-grams multiple generated responses match the references. Zhao et al. (2017) defined BLUE precision (BLEU-P) and recall (BLEU-R) as the average and maximum BLUE score, respectively, and define BLEU-F as combination of BLEU-P and BLEU-R. $n < 4$ is used in our evaluation; (2) *BOW embedding* (Liu et al., 2016), a cosine similarity of bag-of-words embeddings between the generated response and the reference. Three different variants of BOW embedding were tested: (1) Greedy: the average cosine similarities between word embeddings of the two utterances which are greedily matched (Rus and Lintean, 2012); (2) Average: the cosine similarity between the averaged word embeddings in the two utterances (Mitchell and Lapata, 2008); (3) Extreme: the cosine similarity between the largest extreme values in the word embeddings of the two utterances (Pennington et al., 2014); (3) *Dist* (Gu et al., 2019), which measures the diversity of the generated dialogue responses by calculating the ratio of unique n-grams (n=1,2) over all n-grams in the generated dialogue responses. Two types of dist (*intra-dist* and *inter-dist*) were tested, which are calculated within a single sampled response and between different responses, respectively. For each context in the testset, we generate 10 responses with each model and calculate aforementioned metrics averaged over all responses.

Experiment Results. As shown in Table 5, our model yields a stable improvement over most evaluation metrics compared to baselines. Specially, there is a significant improvement on *Dist* for SW and the *BLEU* for DD, respectively, indicating that our model can generate relevant, contentful and diverse dialogue responses. There are some metrics where our model does not outperform the state-of-art baselines, but the difference is small. We also show in Table 6 two example responses generated by TWR-VAE and the best

baseline iVAE. In the first example, our model can generate more topical relevant responses compared to the responses by iVAE, which implies that the latent variable of TWR-VAE can capture a hidden topic information in the dialogue conversation. In the second example, the generated responses of TWR-VAE are more diverse and contentful than the baseline, and the content of those responses can also provide more topics and facilitate the continuation of the conversation.

5 Conclusion

In this paper, in order to solve posterior collapse issue of VAE in text modelling, we propose a simple and generic model called Timestep-Wise Regularisation VAE, which imposes the KL regularisation on the latent variables of every timestep of the encoder. Empirical results in language modelling show that our model can give better performance than all baselines while avoiding posterior collapse. Ablation studies show that the timestep-wise regularisation can be easily applied into different RNN-based VAE models and improve their performance. In addition, we evaluate the timestep-wise regularisation in dialogue response generation task, and the results suggest that our model yields better or comparable performance to the state-of-the-art and can generate relevant, contentful and diverse responses.

Acknowledgements

This work is supported by the award made by the UK Engineering and Physical Sciences Research Council (Grant number: EP/P011829/1).

References

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning (CoNLL)*.

Adji B Dieng, Yoon Kim, Alexander M Rush, and David M Blei. 2019. Avoiding latent variable collapse with generative skip models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2397–2405.

Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. 2019. Implicit deep latent variable models for text generation. 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 3937–3947.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 240–250.

John J Godfrey and Edward Holliman. 1997. Switchboard-1 release 2. *Linguistic Data Consortium, Philadelphia*, 926:927.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Anirudh Goyal Alias Parth Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. 2017. Z-forcing: Training stochastic recurrent networks. In *Advances in neural information processing systems*, pages 6713–6723.

Xiaodong Gu, Kyunghyun Cho, Jung Woo Ha, and Sunghun Kim. 2019. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. In *International Conference on Learning Representations*.

Suchin Gururangan, Tam Dang, Dallas Card, and Noah A Smith. 2019. Variational pretraining for semi-supervised text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5880–5894.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations*.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. 2017. Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1480–1490. JMLR.org.

Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. 2017. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1133–1141. IEEE.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434.

Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. 2018. Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, pages 2683–2692.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations*.

Jack Klys, Jake Snell, and Richard Zemel. 2018. Learning latent subspaces in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 6444–6454.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017b. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.

Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. 2019a. A surprisingly effective fix for deep latent variable modeling of text. 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 3594–3605.

Ruizhe Li, Xiao Li, Chenghua Lin, Matthew Collinson, and Rui Mao. 2019b. A stable variational autoencoder for text modelling. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 594–599.

Ruizhe Li, Chenghua Lin, Matthew Collinson, Xiao Li, and Guanyi Chen. 2019c. A dual-attention hierarchical recurrent neural network for dialogue act classification. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 383–392.

Xiao Li, Chenghua Lin, Ruizhe Li, Chaozheng Wang, and Frank Guerin. 2020. Latent space factorisation and manipulation via matrix subspace projection. In *Proceedings of Machine Learning and Systems 2020*, pages 3211–3221.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.

Mitchell P Marcus and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2).

Nicholas Metropolis and Stanislaw Ulam. 1949. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *proceedings of ACL-08: HLT*, pages 236–244.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Yookoon Park, Jaemin Cho, and Gunhee Kim. 2018. A hierarchical latent structure for variational conversation modeling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1792–1801.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*.

Mary Phuong, Max Welling, Nate Kushman, Ryota Tomioka, and Sebastian Nowozin. 2018. The mutual autoencoder: Controlling information in latent code representations. *International Conference on Learning Representations*.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pages 14837–14847.

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.

Vasile Rus and Mihai Lintean. 2012. An optimal assessment of natural language student input using word-to-word similarity metrics. In *International Conference on Intelligent Tutoring Systems*, pages 675–676. Springer.

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*, pages 627–637.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg. 2018. Improving variational encoder-decoders in dialogue generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. 2016. How to train deep variational autoencoders and probabilistic ladder networks. In *33rd International Conference on Machine Learning (ICML 2016)*.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2018. Wasserstein auto-encoders. In *International Conference on Learning Representations*.

JM Tomczak and M Welling. 2018. Vae with a vampprior. *Proceedings of Machine Learning Research*, 84:1214–1223.

Dingmin Wang, Chenghua Lin, Li Zhong, and Kam-Fai Wong. 2020. Dialogue state tracking with pretrained encoder for multi-domain trask-oriented dialogue systems. *arXiv preprint arXiv:2004.10663*.

Jiacheng Xu and Greg Durrett. 2018. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4503–4513.

Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. 2017. Variational autoencoder for semi-supervised text classification. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3358–3364.

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

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-Volume 70*, pages 3881–3890. JMLR. org.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 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 (Volume 1: Long Papers)*, pages 654–664.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2019. Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 5885–5892.

Qile Zhu, Wei Bi, Xiaojiang Liu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. 2020. A batch normalized inference network keeps the KL vanishing away. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2636–2649, Online, July. Association for Computational Linguistics.

A The derivation of the ELBO (Eq. 1)

The ELBO can be directly derivated from the marginal log likelihood $\log P_{\boldsymbol{\theta}}(\mathbf{x}_i)$:

$$\begin{aligned}
\log P_{\boldsymbol{\theta}}(\mathbf{x}_i) =& \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}\left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i)\right] && (8)\\
=& \mathbb{E}_{Q_{\phi}(\mathbf{z}|\mathbf{x}_i)}\left[\log\left[\frac{P_{\theta}(\mathbf{x}_i,\mathbf{z})}{P_{\theta}(\mathbf{z}|\mathbf{x}_i)}\right]\right] && (9)\\
=& \mathbb{E}_{Q_{\phi}(\mathbf{z}|\mathbf{x}_i)}\left[\log\left[\frac{P_{\theta}(\mathbf{x}_i,\mathbf{z})}{Q_{\phi}(\mathbf{z}|\mathbf{x}_i)}\frac{Q_{\phi}(\mathbf{z}|\mathbf{x}_i)}{P_{\theta}(\mathbf{z}|\mathbf{x}_i)}\right]\right] && (10)\\
=& \underbrace{\mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}\left[\log\left[\frac{P_{\boldsymbol{\theta}}(\mathbf{x}_i,\mathbf{z})}{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}\right]\right]}_{\substack{=\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}_i)\\ \text{(ELBO)}}} + \underbrace{\mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}\left[\log\left[\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}{P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_i)}\right]\right]}_{=D_{\mathrm{KL}}\left(Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)\|P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_i)\right)}, && (11)
\end{aligned}$$

B The reparameterisation trick for our timestep-wise latent variables

If TWR-VAE directly samples $\mathbf{z}^t$ from the $Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})$, this sampling behaviour is undifferentiable. A reparameterisation trick was proposed by Kingma and Welling (2014) to solve this issue. Nevertheless, our TWR-VAE samples multiple $\mathbf{z}^t$ at different timesteps, and we modify the form of each $Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})$, where the mean and covariance do not directly depend on $\mathbf{z}^{t-1}$. After using the reparameterisation trick with $\boldsymbol{\epsilon}^t \sim \mathcal{N}(\mathbf{0},\mathbf{I})$, $\mathbf{z}^t$ can be sampled as:

$$\begin{aligned}
\mathbf{z}^t =& Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})\\
=& g_{\boldsymbol{\phi}}(\mathbf{h}^t,\boldsymbol{\epsilon}^t|\mathbf{x}_i^{1:t})\\
=& \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{h}^t|\mathbf{x}_i^{1:t})^{1/2}\boldsymbol{\epsilon}^t + \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{h}^t|\mathbf{x}_i^{1:t}), && (12)
\end{aligned}$$

where $\boldsymbol{\epsilon}^t \sim \mathcal{N}(\mathbf{0},\mathbf{I})$, and $\mathbf{h}^t$ is the hidden state of the LSTM at t timestep. The mean and covariance are calculated via two linear transformation layers with the $\mathbf{h}^t$.

C The derivation of the gradients optimisation of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ (Eq. 4)

When optimising the $\boldsymbol{\theta}$ and the $\boldsymbol{\phi}$, we use Monte Carlo method (Metropolis and Ulam, 1949) in order to construct a Monte Carlo estimator, which can obtain unbiased gradients of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$:

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}_i)\\
&= \nabla_{\boldsymbol{\theta}}\left(\mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}\left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T)\right] - \frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}\left(Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})\|P(\mathbf{z}^t)\right)\right) && (13)\\
&= \nabla_{\boldsymbol{\theta}}\left(\mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}\left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right]\right) && (14)\\
&= \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}\left[\nabla_{\boldsymbol{\theta}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right)\right] && (15)\\
&\simeq \frac{1}{M}\sum_{m=1}^{M}\nabla_{\boldsymbol{\theta}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_m^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}_m^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}_m^t)}\right) \quad \text{where} \quad \mathbf{z}_m^T \sim Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i) && (16)\\
&= \frac{1}{M}\sum_{m=1}^{M}\nabla_{\boldsymbol{\theta}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_m^T)\right) \quad \text{where} \quad \mathbf{z}_m^T \sim Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i), && (17)
\end{aligned}$$

which is an unbiased Monte Carlo gradient estimator to approximate the expectation (Eq. 13), and M indicates the total number of times that we randomly sample $\mathbf{z}_m^T$ from the $Q_{\boldsymbol{\phi}}(\mathbf{z}_m^T|\mathbf{x}_i^{1:t})$ for approximation.

When applying the similar method to obtain the unbiased gradients of $\boldsymbol{\phi}$, there is an obstacle to finishing the gradients:

$$\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}_i) = \nabla_{\boldsymbol{\phi}}\left(\mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}\left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right]\right) \tag{18}$$

$$\neq \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}\left[\nabla_{\boldsymbol{\phi}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right)\right], \tag{19}$$

However, we can tackle this issue by using the reparameterisation trick proposed by (Kingma and Welling, 2014). Normally, we choose a differentiable and invertible function $g_{\boldsymbol{\phi}}(\mathbf{z},\boldsymbol{\epsilon})$ with the random variable $\boldsymbol{\epsilon}$ to replace $Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)$, namely $\mathbf{z} = g_{\boldsymbol{\phi}}(\mathbf{x},\boldsymbol{\epsilon})$, where $\boldsymbol{\epsilon}\sim P(\boldsymbol{\epsilon})$ (see Eq. 12). We choose $\mathcal{N}(\mathbf{0},\mathbf{I})$ as $P(\boldsymbol{\epsilon})$ and we can use the Monte Carlo estimator approximate Eq. 18:

$$\nabla_{\boldsymbol{\phi}}\left(\mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}\left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right]\right)$$

$$= \nabla_{\boldsymbol{\phi}}\left(\mathbb{E}_{P(\boldsymbol{\epsilon})}\left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right]\right) \tag{20}$$

$$= \mathbb{E}_{P(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\phi}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right)\right] \tag{21}$$

$$\simeq \frac{1}{M}\sum_{m=1}^{M}\nabla_{\boldsymbol{\phi}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_m^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}_m^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}_m^t)}\right) \tag{22}$$

$$= \frac{1}{M}\sum_{m=1}^{M}\nabla_{\boldsymbol{\phi}}\left(-\frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}_m^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}_m^t)}\right)$$

$$\text{where}\quad \mathbf{z}_m^t = g_{\boldsymbol{\phi}}^t\left(\boldsymbol{\epsilon}_m,\mathbf{x}_i^{1:t}\right)\quad\text{and}\quad \boldsymbol{\epsilon}_m\sim\mathcal{N}(\mathbf{0},\mathbf{I}) \tag{23}$$

Overall, the gradients of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ of the ELBO can be re-formed as:

$$\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}_i)$$

$$= \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\left(\mathbb{E}_{P(\boldsymbol{\epsilon})}\left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right]\right) \tag{24}$$

$$= \mathbb{E}_{P(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right)\right] \tag{25}$$

$$\simeq \frac{1}{M}\sum_{m=1}^{M}\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_m^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}_m^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}_m^t)}\right)$$

$$\text{where}\quad \mathbf{z}_m^t = g_{\boldsymbol{\phi}}^t\left(\boldsymbol{\epsilon}_m,\mathbf{x}_i^{1:t}\right)\quad\text{and}\quad \boldsymbol{\epsilon}_m\sim\mathcal{N}(\mathbf{0},\mathbf{I})\,, \tag{26}$$

D Training Details for Language Modelling

We represent input data with 512-dimensional word2vec embeddings (Mikolov et al., 2013) and set the dimension of the hidden layers of both 1-layer encoder and decoder to 256. The dimension of the latent variable is 32. There is no gradient clipped during training. The Adam optimiser (Kingma and Ba, 2015) is used for training with an initial learning rate of 1e-4 and a weight decay of 1e-5. Each sentence in a mini-batch is padded to the maximum length for that batch, and the maximum batch-size allowed is 64.

Yahoo	Input 1	wher can i find a poem called " in flight " ? it has something to do with death dunno
	Input 2	where can i find dinosaur books for my 3 yr old son ? just check with your local library .
BN-VAE	$\alpha = 0$	can can i find a list about " _UNK the " ? i is to to do with the . .
	$\alpha = 0.2$	can tell me what is the name of the song on the _UNK and the _UNK ? i think it is a _UNK song .
	$\alpha = 0.4$	where can i find a list of all the _UNK in the world ? i need to find a list of the _UNK and _UNK of the _UNK .
	$\alpha = 0.6$	where can i find a list of all the _UNK in the world ? i need to find a list of the _UNK and _UNK of the _UNK .
	$\alpha = 0.8$	where can i find a list of all the _UNK in the world ? i need to find a list of the _UNK and _UNK of the _UNK .
	$\alpha = 1$	where can i find a _UNK ? free son year old son ? i go out the local library . they
TWR-VAE	$\alpha = 0$	where can i find a pic in " in touch attendant ? it has been to do with someone and what
	$\alpha = 0.2$	in my opinion what can be done ? it 's a poem for me on myspace .com and some people have no clue
	$\alpha = 0.4$	where can i find an old testament to find out how old it was ? i 'm looking at a photograph of albert einstein .
	$\alpha = 0.6$	where can i find an old book for someone who has an old son ? i need to know how to do it ! !
	$\alpha = 0.8$	where can i find info on my research for an anatomy book ? try these links to your local newspaper . good luck
	$\alpha = 1$	where can i find info for my son year old son ? try be out your local library . good

Table 7: The example of interpolating the latent representation of two input sentences using BN-VAE and TWR-VAE in Yahoo test dataset.

E The derivation of $I(\mathbf{x}, \mathbf{z})$

$$\mathbb{E}_{\mathbf{x}}[D_{\mathrm{KL}}\left(Q_\phi(\mathbf{z}^T|\mathbf{x})\|P(\mathbf{z}^T)\right)] \tag{27}$$
$$= \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{Q_\phi(\mathbf{z}^T|\mathbf{x})}[\log Q_\phi(\mathbf{z}^T|\mathbf{x})]] - \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{Q_\phi(\mathbf{z}^T|\mathbf{x})}[\log P(\mathbf{z}^T)]] \tag{28}$$
$$= -H(Q_\phi(\mathbf{z}^T|\mathbf{x})) - \mathbb{E}_{Q_\phi(\mathbf{z}^T)}[\log P(\mathbf{z}^T)] \tag{29}$$
$$= -H(Q_\phi(\mathbf{z}^T|\mathbf{x})) + H(Q_\phi(\mathbf{z}^T)) - H(Q_\phi(\mathbf{z}^T)) - \mathbb{E}_{Q_\phi(\mathbf{z}^T)}[\log P(\mathbf{z}^T)] \tag{30}$$
$$= I(\mathbf{x}, \mathbf{z}^T) + \mathbb{E}_{Q_\phi(\mathbf{z}^T)}[\log Q_\phi(\mathbf{z}^T)] - \mathbb{E}_{Q_\phi(\mathbf{z}^T)}[\log P(\mathbf{z}^T)] \tag{31}$$
$$= I(\mathbf{x}, \mathbf{z}^T) + D_{\mathrm{KL}}(Q_\phi(\mathbf{z}^T)\|P(\mathbf{z}^T))\,, \tag{32}$$

Therefore:

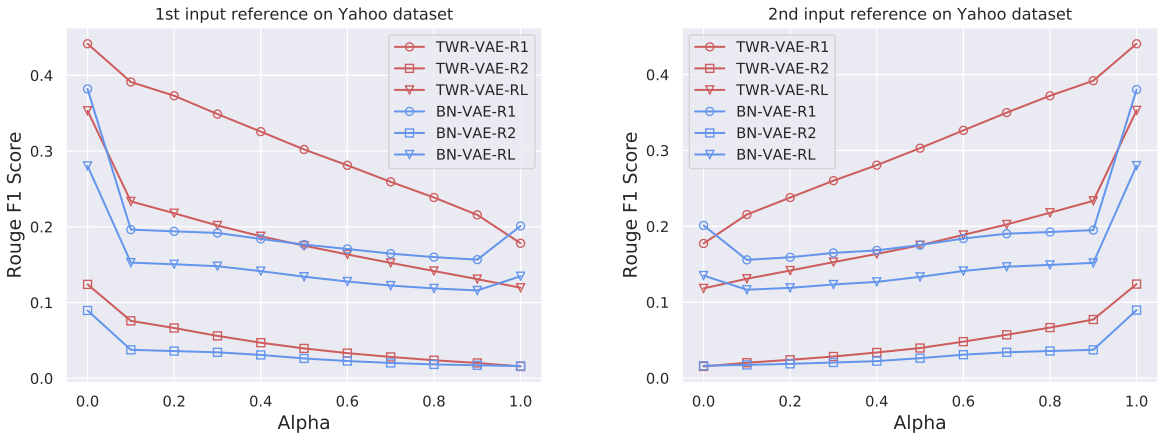
$$I(\mathbf{x}, \mathbf{z}^T) = \mathbb{E}_{\mathbf{x}}[D_{\mathrm{KL}}\left(Q_\phi(\mathbf{z}^T|\mathbf{x})\|P(\mathbf{z}^T)\right)] - D_{\mathrm{KL}}(Q_\phi(\mathbf{z}^T)\|P(\mathbf{z}^T))\,, \tag{33}$$

F Training Details for Dialogue Response Generation

Our model follows the implementation details of the CVAE (Zhao et al., 2017). The size of word embedding is 200 and it is initialised from a pre-trained Glove embedding on Twitter (Pennington et al., 2014). The utterance encoder is a one-layer bidirectional GRU with 300 hidden size, and both of the context encoder and the decoder use a one-layer GRU with 300 hidden size. The recognition network is 1-layer feed-forward network and prior network is 2-layer feed-forward network plus a tanh non-linearity for Gaussian prior sampling. The dimension of the latent variable is 200. The context window size J is 10. The initial weights for recognition and prior networks are sampled from a uniform distribution [-0.02, 0.02]. The vocabulary size is 10,000 and all out-of-vocabulary words are defined as "$<$ unk $>$" token. A greedy decoding mode is used to sample dialogue responses in order to ensure that the randomness comes from the latent variables. The entire model is trained using Adam optimiser with an initial learning rate of 1e-4 and a weight decay of 1e-5. Gradient clipping is not used.

G Examples of the latent representation interpolation on the Yahoo test dataset

There are less _UNK tokens and repeated words occurring in the interpolated sentences generated by our model compared to BN-VAE, as shown in Table 7. Figure 3 shows that our model has higher ROUGE scores than BN-VAE at $\alpha = 0$ for reference one and at $\alpha = 1$ for reference two. Moreover, the ROUGE-L scores of our model are even higher than the ROUGE-1 scores of BN-VAE at $\alpha = \{0.1, 0.2, 0.3\}$ for reference one and at $\alpha = \{0.7, 0.8, 0.9\}$ for reference two.



(a) (b)

Figure 3: The average ROUGE-1, ROUGE-2 and ROUGE-L F1 scores between two input references and 11 interpolations of each group using BN-VAE and TWR-VAE on Yahoo test dataset.