University of Sheffield

Deep Latent Variable Models for Text Modelling

Ruizhe Li

September 2021

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy in Computer Science

in the

Department of Computer Science

Dedicated to my parents, who always unconditionally encourage me to pursue for my dream.

Acknowledgements

First of all, I would like to thank my supervisor Chenghua Lin, who helped me a lot by providing valuable suggestions from research ideas to writing polishing tips in the past four years of my PhD studies. I really enjoy working with him and I am extremely grateful to Chenghua's encouragement, support and patience during the difficult time of my research.

I also want to thank my second supervisor Matthew Collinson when I was in Aberdeen, whose input and advice on my work was very useful. Matthew is also very generous with his time for discussing research, for which I am grateful.

My gratitude also goes to my second supervisor Nikolaos Aletras, who also gives me several invaluable advice and helped me a lot during my PhD studies in Sheffield.

I really appreciated my PhD panel committee members in Aberdeen and Sheffield, Ehud Reiter, Wei Pang and Mauricio Álvarez, who provided many valuable suggestions at my early PhD stages.

I am very fortunate to have a number of great friends who have supported me both research wise and non-research wise, particularly Xiao Li, Guanyi Chen, Rui Mao, Xutan Peng, Chaozheng Wang, Milen Marev, Anthony Chapman, Sam Cauvin, Yida Mu, Haiyang Zhang, Mali Jin and Zehai Tu. Without them, my PhD would have been a less enjoyable one.

I would also like to thank all my colleagues in the Natural Language Processing group for having such a wonderful working environment. I pretty much enjoy our weekly reading group, helpful discussions in each seminar, lovely lunch breaks and coffee time.

Finally, I would like to express my deepest thanks to my parents. Their lifetime love and care help me regain my confidence and make me once again believe in myself.

Abstract

Deep latent variable models is a class of models that parameterise components of probabilistic latent variable models with neural networks. This class of models can capture useful high-level representations of information from the input data, and has been widely applied to many domains (e.g., images, speech, and texts), with tasks ranging from image synthesis to dialogue response generation.

For instance, implicit linguistic cues such as topic information are helpful for various text modelling tasks, e.g., language modelling, dialogue response generation. Being able to accurately recognising dialogue acts plays a key role to help generate relevant and meaningful responses for dialogue systems. However, existing deep learning models mostly focus on modelling the interactions between utterances during a conversation (i.e., contextual information), where important implicit linguistic cues (e.g., topic information of the utterances) for recognising dialogue acts have not been considered. This motivates our first model, which is a dual-attention hierarchical recurrent neural network model for dialogue act classification. Compared to other works which focus on modelling contextual information, our model considers, for the first time, both topic information and dialogue act using a dual-attention hierarchical deep learning framework. Experimental results show that our model achieves a better or comparable performance than other baselines.

When applying deep latent variable models in the text domain, one can generate diverse texts via randomly sampling latent codes from the trained latent space. However, several noticeable issues of deep latent variable models in the text domain remained unsolved, where one of such issues is KL loss vanishing and has serious effects on the quality of generated

texts. To tackle this challenge, we propose a simple and robust Variational Autoencoder (VAE) model to alleviate the KL loss vanishing issue. Specifically, a timestep-wise KL regularisation is proposed and imposed into the encoder of VAE at each timestep. This method does not require careful engineering the objective function of VAE or constructing a more complicated model architecture, as existing models do. In addition, our approach can be easily applied to any types of RNN-based VAEs. Our model is evaluated in the language modelling task and successfully alleviates the KL loss vanishing issue. Our model has also been tested on the dialogue response generation task, which not only avoids the KL loss vanishing issue, but also generates relevant, diverse and contentful responses.

Finally, we investigate the low-density latent regions (holes) of VAE in the text domain, a phenomenon which exists in the trained latent space of VAE and leads to low-quality outputs when latent variables are sampled from those areas. In order to provide an in-depth analysis of the holes issue, a novel and efficient tree-based decoder-centric algorithm for the low-density latent regions identification is developed. We further explore how the holes impact the performance of generated texts of VAE models. For instance, we analyse whether the holes are really vacant, which captures no useful information and how the holes are distributed in the latent space.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The research of modelling natural language from human beings has a long and rich history. One popular class of approaches to capture features from the natural language is generative models, especially the *deep latent variable models*, which models texts under the probabilistic framework with prior knowledge (Kingma and Welling, 2014; Rezende et al., 2014; Bowman et al., 2016). With the help of deep neural networks, Variational Autoencoder (VAE), a deep latent variable model, has recently achieved impressive success in several domains. For instance, VAE has been utilised to generate high-quality human faces or daily scenes in the computer vision domain (Huang et al., 2018; Vahdat and Kautz, 2020), to generate music using latent variables (Roberts et al., 2018), and to generate diverse dialogue responses to reduce the pressure of manual customer service in the dialogue systems (Fang et al., 2019; Li et al., 2020a).

1.1 Variational Autoencoder

Due to that Variational Autoencoder (VAE) is centric to this thesis, I will first give a brief discussion of VAEs. Basically, VAE is a generative model that is designed to generate data via a latent variable $\mathbf{z}$. As depicted in Figure 1.1, the input texts $\mathbf{x}$ are fed into the encoder (encoder is a typical recurrent neural network for texts) at first. Then the hidden representation

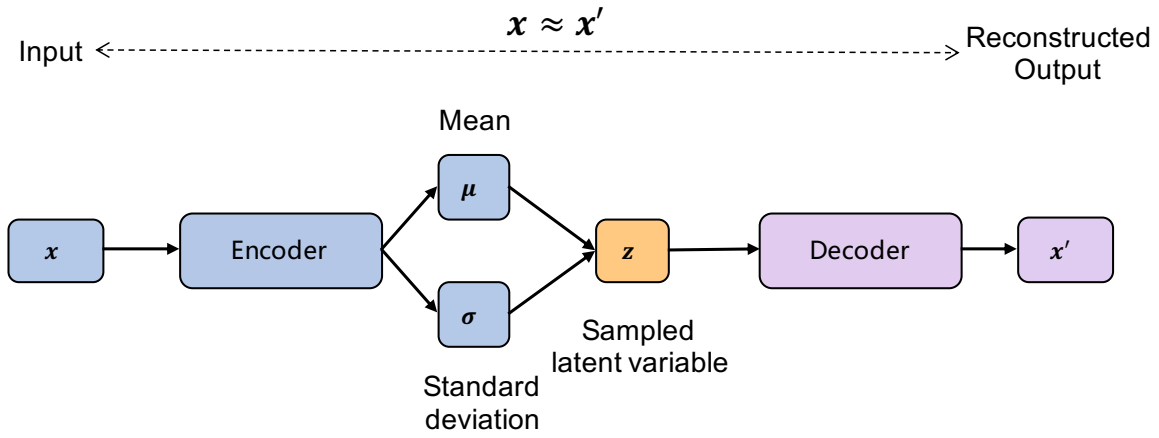


Figure 1.1: *The framework of variational autoencoder (VAE).*

of the input $\mathbf{x}$ is encoded using a linear layer as the mean $\boldsymbol{\mu}$ and standard deviation vectors $\boldsymbol{\sigma}$, respectively. A latent variable $\mathbf{z}$ is sampled from the latent space based on the $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, and finally $\mathbf{z}$ is fed into the decoder (decoder has the same structure as the encoder) to reconstruct output $\mathbf{x}'$.

In general, the objective function of VAE consists of two terms: the first term is the expected reconstruction error indicating how well the model can reconstruct data given a latent variable; 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.

With the help of explicit KL regularisation, the latent space of VAE can be well organised to avoid overfitting and can be easily manipulated to generate outputs. In recent years, VAE has gained much attention and it has been applied to several practical areas. For the dialogue response generation task, there are several relevant works (Zhao et al., 2017, 2018), where the dialogue act is utilised and captured by VAE to help dialogue response generation. In addition, pre-trained VAE language models based on the BERT and GPT-2 are also utilised in this task (Li et al., 2020a). VAE has been tested in different domains, e.g., finance, politics, work, health, etc. Given the previous context utterances from both speakers, the model can generate contextual and diverse responses. In the language modelling task, the latent variable of well-trained VAE can be smoothly manipulated in the latent space to generate diverse texts (Fang et al., 2019; Li et al., 2020a).

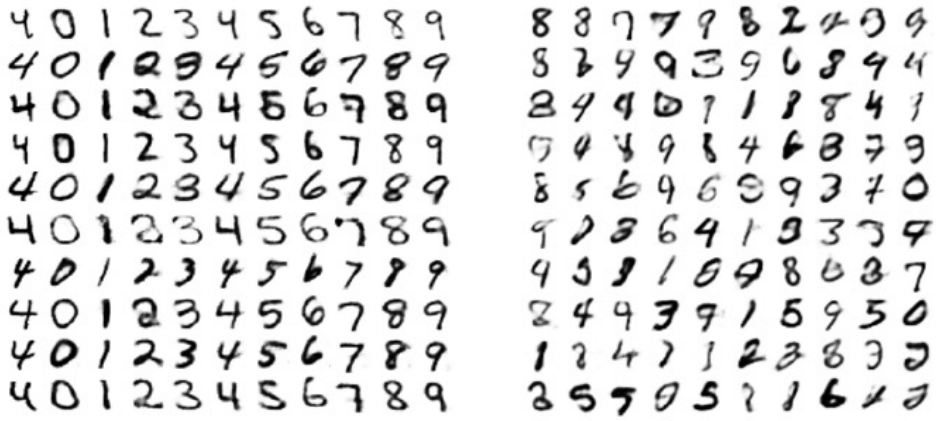


Figure 1.2: *The normal handwriting digits in the MNIST (a) and generated handwriting digits from latent holes (b).*

Although VAEs have been successfully applied in different domains, there still exist several key issues which have not been solved or explored yet, especially in the text domain, i.e., the KL loss vanishing (a.k.a. posterior collapse) (Bowman et al., 2016) and low-density latent regions (a.k.a. holes) (Xu et al., 2020). VAE (Kingma and Welling, 2014; Rezende et al., 2014), a typical encoder-decoder framework, was proposed and applied in the image domain at first. When the VAE is utilised in the text domain, the KL loss vanishing issue occurred (Bowman et al., 2016). Once the issue happens during text generation, the latent variable is ignored by the decoder and the VAE will be downgraded into a simple language model without the help of the latent variable. Consequently, the quality of the generated texts is seriously affected. For instance, generated texts are general and boring (e.g., *I don't know*) or the meaning of them makes no sense (e.g., *the bridge is an old man*).

As for the low-density latent regions, this issue has mainly been studied in the image domain (Davidson et al., 2018; Falorsi et al., 2018; Kalatzis et al., 2020). When the VAE is used to encode non-trivial high-dimensional data manifold into a low-dimensional latent manifold (i.e., latent space), there exists *manifold mismatch* (Davidson et al., 2018; Falorsi et al., 2018), which leads to low-density regions (a.k.a. latent holes) in the learned latent space. If the latent variable is sampled from those areas in the latent space for image synthesis, the generated outputs will be out of the control of the model and hence have low quality.

For example, Figure 1.2 shows that generated digits from latent holes in the MNIST dataset are difficult to recognise compared to normal handwriting digits in that dataset (Davidson et al., 2018). In the text domain, this situation is more severe because the text is discrete and generated texts based on the latent holes is likely to be unreadable (Xu et al., 2020), e.g., text is syntactically incorrect and semantically uninterpretable.

In addition, recognising DA labels is important for many natural language processing tasks. For instance, in dialogue systems, knowing the DA label of an utterance supports its interpretation as well as the generation of an appropriate response (Zhao et al., 2017, 2018). Moreover, VAE is commonly used in dialogue systems since its latent variable can capture high-level linguistic features, such as topics or dialogue acts (Zhao et al., 2018). Accurately recognising dialogue acts will help to regularise the latent space of VAE and cluster encoded utterances with similar dialogue act labels (Zhao et al., 2017). However, current works only focus on modelling contextual information in the conversation, with important linguistic cues such as topic information of the utterances ignored. The rationale behind is that the types of DA associated with a conversation is likely to be influenced by the topic of the conversation (Searle, 1969; Wallace et al., 2013), where DA captures the social act (e.g., promising) and topics describe the subject matter (Wallace et al., 2013). For instance, conversations relating to topics about *customer service* might be more frequently associated with DAs of type Wh-question (e.g., *Why my mobile is not working?*) and a complaining statement (Bhuiyan et al., 2018). However, such a reasonable source of information, surprisingly, has not been explored in the deep learning or deep latent variable models for DA classification. We assume that modelling the topics of utterances as additional contextual information may effectively support DA classification.

This thesis will investigate how to effectively incorporate topical information to improve dialogue act classification, how to avoid the KL loss vanishing issue in the text domain, and explore how the latent holes of VAE affect text generation for the first time.

1.2 Research Challenges

Based on the discussions above, I have identified the following research challenges.

The first research challenge is related to dialogue act (DA) classification, a key task for dialogue systems. Being able to accurately recognising dialogue acts in the conversation is very important as dialogue acts can represent the implicit intention of speakers (Litman and Allen, 1987; Raux et al., 2005) and can help dialogue system generate more relevant and meaningful responses (Zhao et al., 2017). Existing works mainly focus on developing various neural networks to recognise DA by modelling contextual information of utterances. For example, hierarchical CNN (Kalchbrenner and Blunsom, 2013), a deeper-layer LSTM (Khanpour et al., 2016), and hierarchical bidirectional LSTM (Liu et al., 2017; Kumar et al., 2018) have been utilised to encode contextual history in the conversation, where each utterance is encoded as a hidden vector by the models and then all utterance-level hidden vectors are sequentially modelled as a conversation-level hidden vector to classify dialogue acts. Different attention mechanisms have been applied to focus on key words or utterances in the conversation (Kumar et al., 2018; Chen et al., 2018; Raheja and Tetreault, 2019). However, all aforementioned works ignore important linguistic cues in the conversation, such as the topic for the utterance and conversation. Topics are key linguist features for dialogue conversations and they have a close relationship with the types of DA (Searle, 1969; Wallace et al., 2013). Furthermore, the multi-task training strategy has not been explored in the DA classification. The shared encoder within the multi-task learning mode can reduce the risk of overfitting and improve the performance of a single task by training an auxiliary task (Ruder, 2017). In summary, our first research challenge (**RC1**) is: *How can we effectively incorporate and model topical information of utterances for improving DA classification?*

Our second research challenge is grounded on the KL loss vanishing issue when applying VAEs in the text domain. 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). Therefore, our second research challenge (**RC2**) is: *How can we effectively alleviate the KL loss vanishing issue of VAEs by developing a generic solution without needing to design more sophisticated model architectures or carefully balancing the trade-off between the reconstruction loss and the KL divergence?*

The last research challenge is related to the investigations of the low-density regions (a.k.a. latent holes) of the latent space of VAEs. There are only a few prior works that study this problem and they mainly focus on the domain of computer vision. For instance, Davidson et al. (2018) introduced the von Mises-Fisher (vMF) distribution to replace the standard Gaussian distribution and conducted experiments on MNIST dataset; Kalatzis et al. (2020) assumed a Riemannian structure over the latent space by adopting the Riemannian Brownian motion prior. There is only one work, to our best knowledge, that explores the latent holes issue in the text domain. Xu et al. (2020) examined the obstacles that prevent sequence VAEs from performing well in unsupervised controllable text generation, and empirically discovered that manipulating the latent codes for semantic variations in text often leads to latent codes reside in some latent holes. As a result, the decoding network fails to properly decode or generalise when the sampled latent codes land in those low-density latent regions. However, existing works only focus on the low-density latent regions issue on the encoder network and they merely investigate the existence of latent holes on the decoder network without giving in-depth analysis of the issue, such as how the latent holes affect VAE's text generation performance; whether the holes are really vacant or not; how the holes are distributed in the latent space, etc. Therefore, our last research challenge (**RC3**) is: *How can we effectively detect latent holes from the latent space of VAEs and systematically analyse the properties of the holes detected?*

1.3 Research Objectives

In this thesis, I will tackle the three aforementioned challenges by achieving the following research objectives:

1. **RO1**: To propose a deep learning framework for DA classification, which can extract and incorporate important linguistic information (i.e., topics of the dialogue utterances) for improving DA classification performance. This objective tackles challenge **RC1**.

2. **RO2**: To develop a simple and genetic VAE model to alleviate the posterior collapse issue in the language modelling and dialogue response generation tasks. This objective tackles challenge **RC2**.

3. **RO3**: To develop algorithms for effectively detecting latent holes from the latent space of VAEs and systematically analyse the properties of the holes detected. This objective tackles challenge **RC3**.

1.4 Thesis Contributions

The thesis makes three main contributions to meet the three research objectives above:

1. Objective **RO1** is achieved by developing a dual-attention hierarchical recurrent neural network with a CRF, which respects the natural hierarchical structure of a conversation, and is able to incorporate rich context information for DA classification, achieving better or comparable performance to the state-of-the-art. To our knowledge, leveraging topic information of utterances under the multi-task learning mode has not previously been explored in existing deep learning models for DA classification. In addition, we further develop a simple topic labelling mechanism, showing that using the automatically acquired topic information for utterances can effectively improve DA classification.

2. Objective **RO2** is achieved by proposing a simple and robust method, which can effectively alleviate the posterior collapse issue of VAE via timestep-wise regularisation.

Our approach is generic which can be applied to any RNN-based VAE model. In addition, our approach outperforms the state-of-art on language modelling and yields better or comparable performance on dialogue response generation.

3. Objective **RO3** is achieved by proposing a novel tree-based decoder-centric (`TDC`) algorithm for latent hole identification, with a focus on the text domain. In contrast to existing works which are encoder-centric, our approach is centric to the decoder network, as a decoder has a direct impact on model's performance, e.g., for text generation. Our `TDC` algorithm is also highly efficient for latent hole searching when compared to existing approaches, owing to the dimension reduction and Breadth-First Search strategies. Another important technical contribution we have made is that we theoretically unify the two prior indicators for latent hole identification.

1.5 Publications

Chapter 3 is based on the published work:

- **Li R.**, Lin C., Collinson M., Li X. and Chen G. A Dual-Attention Hierarchical Recurrent Neural Network for Dialogue Act Classification, The SIGNLL Conference on Computational Natural Language Learning (CoNLL), Hong Kong, China, 2019.

Chapter 4 is based on the published works:

- **Li R.**, Li X., Chen G. and Lin C. Improving Variational Autoencoder for Text Modelling with Timestep-Wise Regularisation, The 28th International Conference on Computational Linguistics (COLING), Online, 2020.
- **Li R.**, Li X., Lin C, Collinson M. and Mao R. A Stable Variational Autoencoder for Text Modelling, The 12th International Conference on Natural Language Generation (INLG), Tokyo, 2019.

Chapter 5 is based on the work:

- **Li R.***, Peng X.*, Lin C. and Liu B. Understanding Latent Discontinuity of VAEs for Text Generation, Tenth International Conference on Learning Representations (ICLR),

2022, Online, under review.

Other publications:

- Zeng C., Chen G., Lin C., **Li R.** and Chen Z. Affective Decoding for Empathetic Response Generation, The 14th International Conference on Natural Language Generation (INLG), Aberdeen, UK, 2021.
- **Li R.***, Peng X.*, Lin C., Rong W. and Chen Z. On the low-density latent regions of VAE-based language models, NeurIPS 2020 Workshop on Pre-registration in Machine Learning, PMLR 148:343-357, 2021.
- Li X., Chen G., Lin C. and **Li R.** DGST: a Dual-Generator Network for Text Style Transfer, Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 2020.
- Li X., Lin C., **Li R.**, Wang C. and Guerin F. Latent Space Factorisation and Manipulation via Matrix Subspace Projection, The 37th International Conference on Machine Learning (ICML), Vienna, 2020.
- Mao R., Chen G., **Li R.** and Lin C. ABDN at SemEval-2018 Task 10: Recognising Discriminative Attributes using Context Embeddings and WordNet. The International Workshop on Semantic Evaluation at the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), New Orleans, 2018.

1.6 Thesis Overview

The remaining thesis is organised as follows:

1. **Chapter 2** introduces the basic knowledge of neural network and variational autoencoder (VAE) for the whole thesis, and the notation will also be provided. In addition, the related work of dialogue act classification, KL loss vanishing and the low-density regions detection will be summarised.

2. **Chapter 3** focuses on tackling **RO1** by developing a hierarchical model combining

the topical information into dialogue act classification. The methodologies, empirical evaluations and analyses are introduced in detail.

3. **Chapter 4** focuses on solving **RO2** by proposing a timestep-wise KL regularisation VAE for language modelling and dialogue response generation tasks. The methodologies, empirical evaluations and analyses are provided in detail.

4. **Chapter 5** focuses on dealing with **RO3** by proposing a novel tree-based decoder-centric algorithm for the latent hole identification. The theoretical and empirical evaluations are provided in the text domain.

5. **Chapter 6** summarises the highlights main contributions of this thesis. The future work is also discussed.

Chapter 2

Related Work

2.1 Preliminary

In this section, we will give the notation used across the whole thesis first. Then the basic knowledge of neural networks and variational autoencoder will be explained to help the understanding of the thesis.

2.1.1 Notation

This section will contain the most common notation used in this thesis and other additional notations will be introduced in each chapter.

Example	Explanation
x, y, z	The lowercase italic letter denotes a scalar random variable.
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	The lowercase bold letter denotes a vector random variable.
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$	The uppercase bold letter denotes a matrix random variable.
$\mathcal{L}$	The calligraphic letter $\mathcal{L}$ denotes the objective function.
R^N	A N-dimensional real space.
$\mathbf{x}_i$	The i-th element of vector $\mathbf{x}$.
$\mathbf{X}_{i,j}$	The i, j-th element of matrix $\mathbf{X}$.

Example	Explanation
$f_\theta(.), f(.;\theta)$	The functions with the explicit parameters θ (unless clear based on the context).
$p(.), q(.)$	Probability density functions (PDFs) or probability distributions are denoted by the lower-case letters $p(.)$ and $q(.)$, where $q(.)$ is commonly denoted as variational distributions. We use the same notation for probability mass functions (PMFs) of discrete random variables. The same letter will be used for marginals, joint distributions, and conditionals of the same probabilistic model.
$\mathcal{N}(.;\mu,\mathbf{\Sigma})$	A multivariate normal (or Gaussian) distribution with a vector of means μ and a covariance matrix $\mathbf{\Sigma}$.

2.1.2 Neural Network

In this section, we briefly introduce the neural network used throughout this thesis. A neural networks is a parameterised nonlinear network, which consists of multiple layers of nonlinear transformation functions $f^l(.)$ to map input data $\mathbf{x}$ into the hidden vectors $\mathbf{h}$:

$$\mathbf{h} = f^L(f^{L-1}(f^l(\mathbf{x}))), \quad l = 1, ..., L-2\,, \tag{2.1}$$

where f is a parameterised affine linear transformation function followed with a nonlinear function, such as ReLU, sigmoid or tanh functions:

$$f(\mathbf{x}|\boldsymbol{\theta}) = \mathbf{W}\mathbf{x} + \mathbf{b}\,, \tag{2.2}$$

Here $\boldsymbol{\theta}$ denotes all parameters of the neural networks including $\mathbf{W}$ and $\mathbf{b}$. In text domain, input text data $\mathbf{x}$ will be encoded as corresponding embedding vectors:

$$\mathbf{e}_t = \mathbf{E}(\mathbf{x}_t), \quad \mathbf{x}_t \in \mathbf{x}\,, \tag{2.3}$$

Where $\mathbf{E}$ is a word embedding matrix with the size of $|V| \times d$, and $|V|$ is the vocabulary size and d is the number of word embedding dimensions.

2.1.3 Variational Autoencoder

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 using VAE:

$$P_{\boldsymbol{\theta}}(\mathbf{x}_i) = \int P_{\boldsymbol{\theta}}(\mathbf{z}) P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}) d\mathbf{z}\,, \tag{2.4}$$

However, both of the marginal likelihood $P_{\boldsymbol{\theta}}(\mathbf{x}_i)$ and the true posterior distribution $P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_i)$ are intractable, where

$$P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_i) = \frac{P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}) P_{\boldsymbol{\theta}}(\mathbf{z})}{P_{\boldsymbol{\theta}}(\mathbf{x}_i)}\,, \tag{2.5}$$

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 for the whole training dataset:

$$\log P_{\boldsymbol{\theta}}\left(\mathbf{x}_1, \ldots, \mathbf{x}_N\right) = \sum_{i=1}^{N} \log P_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right)\,, \tag{2.6}$$

This is essentially equivalent to maximising the following evidence lower bound (ELBO), which consists of two terms (Kingma and Welling, 2014):

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i) = \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z})] - D_{\mathrm{KL}}\left(Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i) \| P(\mathbf{z})\right) , \tag{2.7}$$

The first term is the expected reconstruction error indicating how well the model can reconstruct data given a latent variable. 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.

2.2 Dialogue Act Classification

Dialogue act classification is a task which focuses on predicting the correct DA label associated with the corresponding utterance in the conversation. Broadly speaking, methods for DA classification can be divided into two categories: traditional machine learning (e.g., support vector machine (SVM), hidden Markov model (HMM), dynamic Bayesian network (DBN), etc) and neural networks (e.g., recurrent neural networks (RNNs), convolutional neural networks (CNN), etc). In this section, we will introduce the related work from these directions.

2.2.1 Traditional Machine Learning Methods

Stolcke et al. (2000) was the first work to model DA using a statistical approach and also created a large human-annotated Switchboard dataset from the human-to-human telephone conversation. They utilised the 1st-order HMM to model the dialogue conversation, where the HMM states denote DAs, observations represent utterances, transition probabilities and observation probabilities are respectively calculated by the n-gram discourse model and the posterior likelihood of the DA. With the trigram discourse model and the HMM, their model achieved 71% DA classification accuracy in the Switchboard corpus. In contrast with Stolcke et al. (2000) which recognised DA classification as a structured prediction task,

Liu (2006) regarded this task as a multiclass work by breaking the direct multiclass task into multiple binary classification sub-tasks using SVMs and error correcting output codes (ECOC), and then combining their results together. Their experiment empirically showed that the combination of multiple binary SVM and the ECOC improve the DA classification accuracy compared with the direct multiclass task using SVM. Kim et al. (2010) investigated the effectiveness of various features for the DA classification, e.g., bag of words (BoW), the location of an utterance, author information, the dependency of different utterances, etc. The finding is that the structural information (i.e., the position of each utterance and the author information) and inter-utterance dependency help to improve the DA accuracy associated with CRF and 2-grams/3-grams works well with SVM.

In contrast with discriminative models only used in the aforementioned works, there is another research direction combining the discriminative model and the generative model to classify DA labels. Dielmann and Renals (2008) proposed a joint generative model for the DA segmentation and classification. In detail, various lexical and prosodic features including timing, intonation and energy are encoded using a Gaussian mixture model (GMM), a factored language model (FLM) and an interpolated FLM are used to model the dependency between DA labels and the corresponding utterances and a trigram language model is applied to model the probability of predicting a sequence of DA labels. Finally, a switching dynamic Bayesian network (DBN) integrates all components together and a CRF is built upon this architecture to classify DA labels. Wallace et al. (2013) utilised a generative joint sequential model to classify both DA and topics of patient-doctor conversations. Their model is similar to the factorial LDA model (Paul and Dredze, 2012), which generalises LDA to assign each token a K-dimensional vector of latent variables. Each utterance is generated conditioned on the previous and current topic/DA pairs in their model.

2.2.2 Deep Learning Neural Networks

In recent years, deep learning neural networks have been commonly applied in several NLP tasks, including DA classification. Kalchbrenner and Blunsom (2013) utilised CNN and RNN

to model the sentence and discourse information, respectively. A hierarchical convolutional neural network (HCNN) is used to encode each input sentence in the conversation, and then an RNN takes the current sentence vector from the HCNN and previously predicted DA label as input to predict the current DA label, which models the interactions between different utterances during the conversation. Khanpour et al. (2016) empirically confirmed that pre-trained word embedding, different dropout regularisation probabilities, fine-tuned decay rate and the number of LSTM layers have a significant effect on the DA classification accuracy. Different to the aforementioned works, Ji et al. (2016) developed a latent variable recurrent neural network for jointly modelling sequences of words and discourse relations between adjacent sentences. In their work, the shallow discourse structure is represented as a latent variable and the contextual information from preceding utterances are modelled with an RNN.

However, the above works only regard the conversation as a flat structure and other information has been not encoded yet, e.g., character information, the history utterance in the conversation, etc. Liu et al. (2017) proposed a hierarchical model for DA labels, where a CNN encoded the input utterances and then the previous predicted DA labels or the probability distributions are concatenated with the current CNN sentence vector for the current DA label prediction using an LSTM. Kumar et al. (2018) further proposed a hierarchical bidirectional LSTM with the CRF as the top layer to recognise DA labels. Compared to Liu et al. (2017), Kumar et al. (2018) encoded the entire history utterance and the corresponding DA labels using the CRF, which models the deeper dependency among utterances and DA labels in the conversation. Chen et al. (2018) proposed a CRF-Attentive Structured Network for DA classification. A memory mechanism is employed to jointly encode the current sentence vector and context information, and the CRF is improved by the internal structured attention network to consider the contextual information including nearing utterances and DA labels. In contrast with aforesaid works, Li et al. (2019c) proposed a multitask learning setting to parallel classify the DA label and topic label of each utterance. Specifically, the character-level, word-level, utterance-level and conversation-level information are jointly captured for DA classification. In addition, a dual attention mechanism was utilised to share the DA and

topic information for each task and a CRF layer was also added into the top layer to further model the dependency between utterances and DA labels in the conversation. Raheja and Tetreault (2019) instead utilised the context-aware self-attention to capture the dependency among different utterances, and they also used multiple word embeddings (e.g., pre-trained ELMO word embedding) to enhance the performance of their model.

2.3 KL Loss Vanishing

Variational autoencoder (VAE) (Kingma and Welling, 2014; Rezende et al., 2014) is a generative model which is commonly applied in image generation (Huang et al., 2018), text classification (Xu et al., 2017), text generation (Bowman et al., 2016; Yang et al., 2017; Li et al., 2020b), etc. However, an issue occurs called KL loss vanishing (or posterior collapse) when the VAE is used in the text generation. The latent variable will be ignored and the VAE is downgraded into a simple language model without the help of the latent variable. Consequently, the quality of the generated texts is seriously affected. Several works proposed methodologies for this issue from different views, e.g., changing the training strategy, weakening the decoder, replacing the Gaussian prior with other distributions, etc. A detailed literature review will be introduced from the aforementioned directions.

Bowman et al. (2016) was the first work to apply the VAE into the text generation task. When the VAE is utilised as a language model, a small reconstruction loss and a non-zero KL divergence represent that the latent variable $\mathbf{z}$ captures some useful information. However, they found that the posterior distribution $Q_\phi(\mathbf{z}|\mathbf{x})$ is equal to the prior distribution $P(\mathbf{z})$ (i.e., the KL divergence is zero) during training, which leads to the KL loss vanishing (or posterior collapse). This issue further downgrades the VAE into a simpler language model and causes bad-quality text generation. Bowman et al. (2016) argued that the sensitivity of the LSTM decoder to the subtle variation in its hidden vectors is a major cause of the KL loss vanishing, which optimises the model in an easier way by ignoring the latent variable $\mathbf{z}$. A KL cost annealing is thus proposed to tackle this issue, where an increasing weight is added to the KL divergence from 0 to 1 in order to prevent the VAE from ignoring $\mathbf{z}$. At first, the weight is set

as zero (i.e., the KL term disappears) and the information from the input $\mathbf{x}$ can be encoded into $\mathbf{z}$ as much as possible. Then the weight is gradually increased to 1 and the model is forced to smooth out the latent variable $\mathbf{z}$ in the latent space. Finally, the region of the trained $\mathbf{z}$ is allocated a high probability by the Gaussian prior distributions. The experiment empirically proves that the KL loss vanishing issue can be alleviated and this trick is commonly used in various work later. In addition, as a powerful decoder is another reason leading to the posterior collapse, a method to weaken the decoder is developed by randomly replacing the ground-truth words encoded in the decoder with the UNK token. Yang et al. (2017) instead proposed a dilated CNN to replace the powerful LSTM decoder by restricting the contextual capacity of the decoder. The contextual width of the dilated can be controlled based on the dilation size and filter size. Their experiment confirms that the VAE with a dilated CNN decoder can generate better sentences compared with the vanilla RNN-based VAE with lower NLL loss and perplexity. They further explored that there exists a trade-off between the contextual capacity of the decoder and how to effectively use the encoding information. Their model also achieved great performance for the semi-supervised text classification and unsupervised clustering of the text categorisation and sentiment analysis. However, the CNN decoder introduced by (Yang et al., 2017) needs a careful selection to achieve the best contextual capacity, Semeniuta et al. (2017) developed a hybrid convolutional and deconvolutional VAE combined with RNN for the text generation. The encoder in their model consists of a feed-forward network with a one-dimensional convolutional layer and the decoder includes a one-dimensional deconvolutional layer plus an RNN layer. This entire architecture not only converges the model faster but also captures longer dependencies in the text sequences. They further added an auxiliary regularisation loss by calculating the intermediate reconstruction loss with the hidden vectors from the deconvolutional layer. Compared to the aforementioned work weakening the decoder with different methods, Dieng et al. (2019) instead proposed a skip connection in the decoder side to enforce the connection between the latent variable $\mathbf{z}$ and different layers in the decoder, which avoids the ignorance of $\mathbf{z}$. They empirically and theoretically show that the skip connections increase the mutual information between the input $\mathbf{x}$ and the inferred $\mathbf{z}$, and the model outperforms several baselines in image synthesis

and text generation tasks.

Another direction to alleviate the KL loss vanishing is changing the training strategy. Kim et al. (2018) introduced a hybrid training approach to alleviate the posterior collapse issue, i.e., utilising amortized variational inference (AVI) to quickly initialise the whole variational parameters first and then applying stochastic variational inference (SVI) to optimise each data point. Specifically, AVI is used to update the variational parameters, where the encoder (i.e., inference network) along with the decoder (i.e., generative network) is optimised across the whole data using AVI and then SVI is employed to iteratively refine each data point with several numbers of steps. Their experiments show that their semi-amortized variational autoencoder outperforms the autoregressive/VAE/SVI baselines in texts and image generation tasks. Although the hybrid training improves the performance of the VAE, the effectiveness of their model is still limited (generally requiring more than 10x time to finish convergence). Fu et al. (2019) instead proposed a cyclical annealing schedule to repeatedly increase the weight β of the KL term from 0 to 1, which avoids the unwell-trained latent variables at the beginning of optimisation. This annealing schedule can be regarded as a cyclical warm restart, which gradually helps the latent variable $\mathbf{z}$ to encode the information of the input $\mathbf{z}$. In contrast with Fu et al. (2019) which cyclically anneals the weight β, He et al. (2019) found that the training for the inference network initially cannot approach the true posterior of the model, and subsequently conducted an aggressive optimisation for the inference network before updating the whole model. The mutual information between the latent variable $\mathbf{z}$ and the input $\mathbf{x}$ under the posterior distribution $Q_{\phi}(\mathbf{z}|\mathbf{x})$ to control the stopping criterion of the aggressive optimisation. Li et al. (2019a) recently proposed two simple approaches to improve the VAE in text modelling: (1) they pretrained the inference network first based on the autoencoder objective function to deviate the model from the local optimal where posterior collapse happens, and (2) the KL term is replaced with the free bits (FB) (Kingma et al., 2016) using a hinge loss to threshold the loss as a constant. The combination of them achieves superior performance in the language modelling task under the perplexity metric.

Apart from the aforementioned directions, there are also different attempts for the posterior collapse. Xu and Durrett (2018) replaced the Gaussian prior distribution with the

von Mises-Fisher (vMF) distribution. Specifically, the KL term only depends on the fixed concentration parameter κ in the unit hypersphere placed by the vMF distribution, and the KL term is constant and the posterior collapse can be avoided. Different from fixing the KL term by changing different prior distributions, Fang et al. (2019) introduced sample-based representations of input data and the aggregated posterior samples are matched to the prior distribution. Then the latent variable $\mathbf{z}$ is guided to encode more diverse and useful information of each input data. Although most works focus on weakening the decoder in VAE, a few works tried to improve the encoder instead. Li et al. (2019b) proposed an HR-VAE to impose a holistic KL regularisation into each concatenation of the hidden and cell states of the LSTM, which improves the encoding capacity of the encoder and avoids the posterior collapse. Li et al. (2020b) further proposed two variants of the HR-VAE by sampling $\mathbf{z}$ at each timestep and feeding the mean or sum of all $\mathbf{z}$ to the decoder. In addition, the dimension of $\mathbf{z}$ in the variants are much smaller than the one in HR-VAE, and the training speed is six-time faster than the HR-VAE. Instead of imposing the KL divergence into each timestep in the encoder, Zhu et al. (2020) forced the KL divergence to follow a distribution across the entire dataset and applied a fixed batch normalisation to the expectation of the KL distribution. In this way, a positive expectation of the KL distribution is confirmed and the posterior collapse consequently is alleviated.

2.4 Latent Hole Detection

VAE has shown its powerful capacity to unsupervisedly generate outputs by mapping the non-trivial high-dimensional data manifold to the learned low-dimensional manifold (i.e., the latent space). There are several successful applications of the VAE in a number of downstream tasks, e.g., image synthesis (Huang et al., 2018), language modelling (Bowman et al., 2016; Fang et al., 2019), dialogue generation (Zhao et al., 2017), music composition (Roberts et al., 2018), etc. However, when the high-dimensional non-trivial data manifold is mapped to the low-dimensional manifold using VAE, the low-dimensional manifold does have low-density regions (aka. latent holes), from which the generated outputs have low quality (Falorsi et al.,

2018; Xu et al., 2020). This section summarises relevant works which identified those areas and tried to alleviate the latent holes.

When a non-trivial high-dimensional data manifold is mapped to a low-dimensional manifold, the two manifolds are not homeomorphic[1] and there exists an irreversible mapping between these manifolds. Especially for VAE, the data manifold is mapped to the approximate posterior distribution in the latent space following the Gaussian prior distribution. Since the basic structure between the data manifold and the approximate posterior distribution in the latent space is different, the embedding of the data manifold is smoothly mapped to the latent space at the expense of leaving several low-density latent regions, which leads the bad-quality outputs. Davidson et al. (2018) proposed to use a von Mises-Fisher (vMF) distribution to replace the Gaussian prior for the VAE. The advantage of vMF is that a uniform distribution prior will be placed on the latent space and this prior would not force different clusters of the mapped data manifold to the origin and does not add extra directional bias into the distribution of the mapped distribution. Their model was applied to several tasks, e.g., semi-supervised classification on MNIST, link prediction on graphs, etc, which shows that the latent space using vMF learns a better latent representation than Gaussian distribution. Falorsi et al. (2018) instead investigated this issue in Lie groups and proposed to construct a VAE with the latent variables lying in the Lie groups with the help of reparameterised trick on the group of 3D rotation SO(3). They also developed an evaluation metric to measure the continuity of the latent representations sampled from the latent space based on the Lipschitz continuity assumption. In contrast with vMF used by (Davidson et al., 2018), Kalatzis et al. (2020) also argued that the Euclidean space is the heart of the failure for the low-density latent regions, but they employed Riemannian Brownian motion prior to replace the Gaussian prior. Especially, the identifiability issue was solved using the Riemannian Brownian motion prior and this prior also constrains the samples only from the mapped data manifold in the latent space, which effectively alleviates discontinuity in the latent space. However, the aforementioned works mainly focus on the image field, the discrete text space has rarely been explored yet. Xu et al. (2020) were the first work to explore the low-density latent regions in the text domain. They

[1] the topological structure should be preserved mapped from one space to another one.

proposed to learn a probability simplex to constrain the posterior mean in it and only sampled and manipulated latent representations in the simplex. In addition, the manipulated latent representations are evaluated using the NLL loss under the aggregated posterior distributions to distinguish which latent vector is a hole or not.

Chapter 3

A Dual-Attention Hierarchical Model for Dialogue Act Recognition

3.1 Introduction

Dialogue Acts (DA) are semantic labels of utterances, which are crucial to understanding communication: much of a speaker's intent is expressed, explicitly or implicitly, via social actions (e.g., questions or requests) associated with utterances (Searle, 1969). Recognising DA labels is important for many natural language processing tasks. For instance, in dialogue systems, knowing the DA label of an utterance supports its interpretation as well as the generation of an appropriate response (Searle, 1969; Chen et al., 2018). In the security domain, being able to detect intention in conversational texts can effectively support the recognition of sensitive information exchanged in emails or other communication channels, which is critical to timely security intervention (Verma et al., 2012).

A wide range of techniques have been investigated for DA classification. Early works on DA classification are mostly based on general machine learning techniques, framing the problem either as multi-class classification (e.g., using SVMs (Liu, 2006) and dynamic Bayesian networks (Dielmann and Renals, 2008)) or a structured prediction task (e.g., using Conditional Random Fields (Kim et al., 2010; Chen et al., 2018; Raheja and Tetreault, 2019,

CRF)). Recent studies to the problem of DA classification have seen an increasing uptake of deep learning techniques, where promising results have been obtained. Deep learning approaches typically model the dependency between adjacent utterances (Ji et al., 2016; Lee and Dernoncourt, 2016). Some researchers further account for dependencies among both consecutive utterances and consecutive DAs, i.e., both are considered factors that influence natural dialogue (Kumar et al., 2018; Chen et al., 2018). There is also work exploring different deep learning architectures (e.g., hierarchical CNN or RNN/LSTM) for incorporating context information for DA classification (Liu et al., 2017).

It has been observed that conversational utterances are normally associated with both a DA and a topic, where the former captures the social act (e.g., promising) and the latter describes the subject matter (Wallace et al., 2013). It is also recognised that the types of DA associated with a conversation are likely to be influenced by the topic of the conversation (Searle, 1969; Wallace et al., 2013). For instance, conversations relating to topics about *customer service* might be more frequently associated with DAs of type Wh-question (e.g., *Why my mobile is not working?*) and a complaining statement (Bhuiyan et al., 2018); whereas meetings covering administrative topics about resource allocation are likely to exhibit significantly more defending statements and floor grabbers (e.g., *Well I mean - is the handheld really any better?*) (Wrede and Shriberg, 2003). However, such a reasonable source of information, surprisingly, has not been explored in the deep learning literature for DA classification. We assume that modelling the topics of utterances as additional contextual information may effectively support DA classification.

We propose a dual-attention hierarchical recurrent neural network with a CRF (DAH-CRF) for DA classification. Our model is able to account for rich context information with the developed dual-attention mechanism, which, in addition to accounting for the dependencies between utterances, can further capture, for utterances, information about both topics and DAs. Topic is a useful source of context information which has not previously been explored in existing deep learning models for DA classification. Second, compared to the flat structure employed by existing models (Khanpour et al., 2016; Ji et al., 2016), our hierarchical recurrent neural network can represent the input at the character, word, utterance, and conversation

levels, preserving the natural hierarchical structure of a conversation. To capture the topic information of conversations, we propose a simple automatic utterance-level topic labelling mechanism based on LDA (Blei et al., 2003), which avoids expensive human annotation and improves the generalisability of our model.

We evaluate our model against several strong baselines (Wallace et al., 2013; Ji et al., 2016; Kumar et al., 2018; Chen et al., 2018; Raheja and Tetreault, 2019) on the task of DA classification. Extensive experiments conducted on three public datasets (i.e., Switchboard (SWDA), DailyDialog (DyDA), and the Meeting Recorder Dialogue Act corpus (MRDA)) show that by modelling the topic information of utterances as an auxiliary task, our model can significantly improve DA classification for all datasets compared to a base model without modelling topic information. Our model also yields better or comparable performance to state-of-the-art deep learning method (Raheja and Tetreault, 2019) in classification accuracy.

To summarise, the contributions of this work are three-fold:

1. we propose to leverage topic information of utterances, a useful source of contextual information which has not previously been explored in existing deep learning models for DA classification;

2. we propose a dual-attention hierarchical recurrent neural network with a CRF which respects the natural hierarchical structure of a conversation, and is able to incorporate rich context information for DA classification, achieving better or comparable performance to the state-of-the-art;

3. we develop a simple topic labelling mechanism, showing that using the automatically acquired topic information for utterances can effectively improve DA classification.

3.2 Methodology

Given a training corpus $\mathcal{D} = \langle (C_n, Y_n, Z_n) \rangle_{n=1}^{N}$, where $C_n = \langle u_t^n \rangle_{t=1}^{T}$ is a conversation containing a sequence of T utterances, $Y_n = \langle y_t^n \rangle_{t=1}^{T}$ and $Z_n = \langle z_t^n \rangle_{t=1}^{T}$ are the corresponding

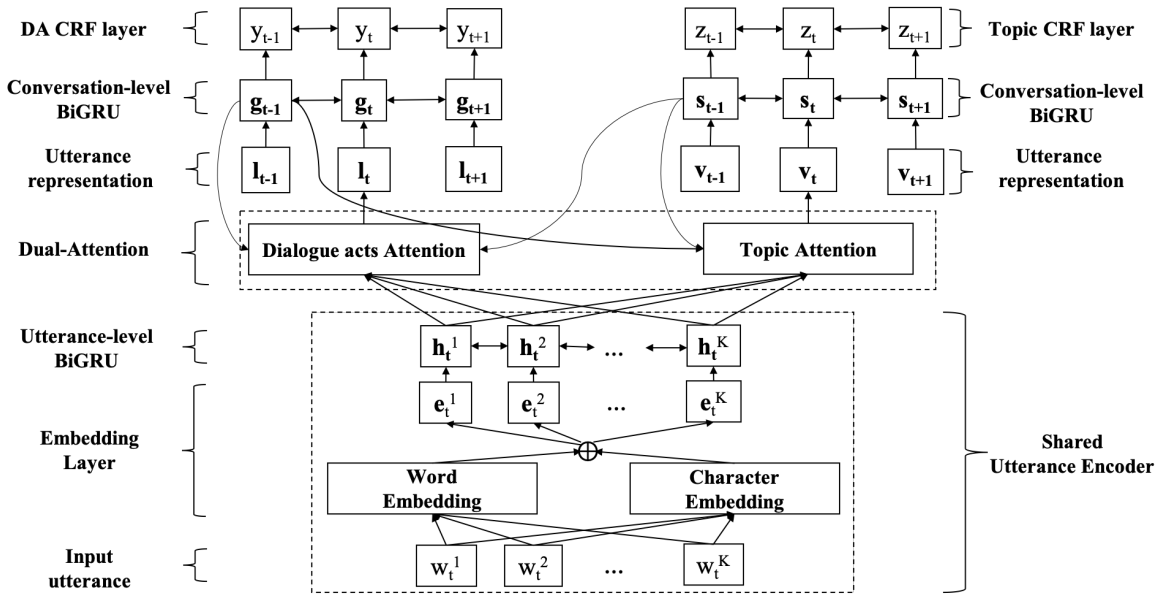


Figure 3.1: *Overview of the dual-attention hierarchical recurrent neural network with a CRF.*

labels of DA and topics for C_n, respectively. Each utterance $u_t = \langle w_t^i \rangle_{i=1}^K$ of C_n is a sequence of K words. Our goal is to learn a model from $\mathcal{D}$, such that, given an unseen conversation C_u, the model can predict the DA labels of the utterances of C_u.

Figure 3.1 gives an overview of the proposed Dual-Attention Hierarchical recurrent neural network with a CRF (DAH-CRF). A shared utterance encoder encodes each word w_t^i of an utterance u_t into a vector $\mathbf{h}_t^i$. The DA attention and topic attention mechanisms capture DA and topic information as well as the interactions between them. The outputs of the dual-attention are then encoded in the conversation-level sequence taggers (i.e., $\mathbf{g}_t$ and $\mathbf{s}_t$), based on the corresponding utterance representations (i.e., $\mathbf{l}_t$ and $\mathbf{v}_t$). Finally, the target labels (i.e., y_t and z_t) are predicted in the CRF layer.

3.2.1 Shared Utterance Encoder

In our model, we adopt a shared utterance encoder to encode the input utterances. Such a design is based on the rationale that the shared encoder can transfer parameters between two tasks and reduce the risk of overfitting (Ruder, 2017). Specifically, the shared utterance

encoder is implemented using the bidirectional gated recurrent unit (Cho et al., 2014, BiGRU), which encodes each utterance $u_t = \langle w_t^i \rangle_{i=1}^K$ of a conversation C_n as a series of hidden states $\langle \mathbf{h}_t^i \rangle_{i=1}^K$. Here, i indicates the timestamp of a sequence, and we define $\mathbf{h}_t^i$ as follows

$$\mathbf{h}_t^i = \overrightarrow{\mathbf{h}}_t^i \oplus \overleftarrow{\mathbf{h}}_t^i \tag{3.1}$$

where $\oplus$ is an operation for concatenating two vectors, and $\overrightarrow{\mathbf{h}}_t^i$ and $\overleftarrow{\mathbf{h}}_t^i$ are the i-th hidden state of the forward gated recurrent unit (Cho et al., 2014, GRU) and backward GRU for w_t^i, respectively. Formally, the forward GRU $\overrightarrow{\mathbf{h}}_t^i$ is computed as follows

$$\overrightarrow{\mathbf{h}}_t^i = \mathrm{GRU}(\overrightarrow{\mathbf{h}}_t^{i-1}, \mathbf{e}_t^i) \tag{3.2}$$

where $\mathbf{e}_t^i$ is the concatenation of the word embedding and the character embedding of word w_t^i. Finally, the backward GRU encodes u_t from the reverse direction (i.e. $w_t^K \rightarrow w_t^1$) and generates $\langle \overleftarrow{\mathbf{h}}_t^i \rangle_{i=1}^K$ following the same formulation as the forward GRU.

3.2.2 Task-specific Attention

Recall that one of the key challenges of our model is to capture for each utterance, information about both DAs and topics, as well as information about the interactions between them. We address this challenge by incorporating into our model a novel task-specific dual-attention mechanism, which accounts for both DA and topic information extracted from utterances. In addition, DAs and topics are semantically relevant to different words in an utterance. With the proposed attention mechanism, our model can also assign different weights to the words of an utterance by learning the degree of importance of the words to the DA or topic labelling task, i.e., promoting the words which are important to the task and reducing the noise introduced by less important words.

For each utterance u_t, the DA attention calculates a weight vector $\langle \alpha_t^i \rangle_{i=1}^K$ for $\langle \mathbf{h}_t^i \rangle_{i=1}^K$, the

hidden states of u_t. u_t can then be represented as an attention vector $\mathbf{l}_t$ computed as follows

$$\mathbf{l}_t = \sum_{i=1}^{K} \alpha_t^i \mathbf{h}_t^i \tag{3.3}$$

In contrast to the traditional attention mechanism (Bahdanau et al., 2015), which only depends on one set of hidden vectors from the Seq2Seq decoder, the DA attention of our model relies on two sets of hidden vectors, i.e., $\mathbf{g}_{t-1}$ of the conversation-level DA tagger and $\mathbf{s}_{t-1}$ of the conversation-level topic tagger, where dual attention mechanism can capture, for utterances, information about both DAs and topics as well as the interaction between them. Specifically, the weights $\langle \alpha_t^i \rangle_{i=1}^K$ for the DA attention are calculated as follows:

$$\alpha_t^i = \mathrm{softmax}(o_t^i) \tag{3.4}$$

Where the hidden vector o_t^i is calculated using a linear neural network:

$$o_t^i = \mathbf{w}_a^\top \tanh\left(\mathbf{W}^{(\mathrm{act})}(\mathbf{s}_{t-1} \oplus \mathbf{g}_{t-1} \oplus \mathbf{h}_t^i) + \mathbf{b}^{(\mathrm{act})}\right) \tag{3.5}$$

The topic attention layer has a similar architecture to the DA attention layer, which takes as input both $\mathbf{s}_{t-1}$ and $\mathbf{g}_{t-1}$. The weight vector $\langle \beta_t^i \rangle_{i=1}^K$ for the topic attention output $\mathbf{v}_t$ can be calculated similar to Eq. 3.3 and Eq. 3.4. Note that $\mathbf{w}_a^\top$, $\mathbf{W}^{(\mathrm{act})}$, and $\mathbf{b}^{(\mathrm{act})}$ are vectors of parameters that need to be learned during training.

3.2.3 Conversational Sequence Tagger

3.2.3.1 CRF Sequence Tagger for DA

The conversational CRF sequence tagger for DA predicts the next DA y_t conditioned on the conversational hidden state $\mathbf{g}_t$ and adjacent DAs (c.f. Figure 3.1). Formally, this conditional

probability of the whole conversation can be formulated as

$$p\left(y_{1:T}|C;\theta\right) = \frac{\prod_{t=1}^{T}\Psi\left(y_{t-1},y_t,\mathbf{g}_t;\theta\right)}{\sum_{Y}\prod_{t=1}^{T}\Psi\left(y_{t-1},y_t,\mathbf{g}_t;\theta\right)} \tag{3.6}$$

$$\begin{aligned}\Psi\left(y_{t-1},y_t,\mathbf{g}_t;\theta\right) &= \Psi_{emi}\left(y_t,\mathbf{g}_t\right)\Psi_{tran}\left(y_{t-1},y_t\right) \\ &= \mathbf{g}_t\left[y_t\right]\mathbf{P}_{y_t,y_{t-1}}\end{aligned} \tag{3.7}$$

Here the feature function $\Psi(\cdot)$ includes two score potentials: emission and transition. The emission potential Ψ_{emi} regards utterance representation $\mathbf{g}_t$ as the unary feature. The transition potential Ψ_{tran} is a pairwise feature constructed from a $T \times T$ state transition matrix $\mathbf{P}$, where T is the number of DA classes, and $\mathbf{P}_{y_t,y_{t-1}}$ is the probability of transiting from state y_{t-1} to y_t. $C = \langle u_t \rangle_{t=1}^{T}$ is the sequence of all utterances seen so far, θ is the parameters of the CRF layer. $\mathbf{g}_t$ is calculated in a BiGRU similar to Eq. 3.1 and Eq. 3.2:

$$\mathbf{g}_t = \overrightarrow{\mathbf{g}}_t \oplus \overleftarrow{\mathbf{g}}_t \tag{3.8}$$

$$\overrightarrow{\mathbf{g}}_t = \mathrm{GRU}(\overrightarrow{\mathbf{g}}_{t-1}, \mathbf{l}_t) \tag{3.9}$$

3.2.3.2 CRF Sequence Tagger for Topic

The conversational CRF sequence tagger for topic is designed to predict topic z_t conditioned on $\mathbf{v}_t$ and adjacent topics, which can be calculated similar to the formulation of the CRF tagger for DA.

3.2.3.3 Training the Model

Let Θ be all the model parameters that need to be estimated for DAH-CRF. Θ then is estimated based on $\mathcal{D} = \langle (C_n, Y_n, Z_n) \rangle_{n=1}^{N}$ (i.e., a corpus with N conversations) by maximising the following objective function

$$\mathcal{L} = \sum_{n=1}^{N} \left[\log \left(p \left(y_{1:T}^{n} | C_n; \Theta \right) \right) + \alpha \log \left(p \left(z_{1:T}^{n} | C_n; \Theta \right) \right) \right] \tag{3.10}$$

The hyper-parameter α controls the contribution of the conversational topic tagger towards the objective function. In our experiments, $\alpha = 0.5$ is determined using the validation datasets. During the test, the optimal DA or topic sequence is calculated using the Viterbi algorithm (Viterbi, 1967).

$$Y' = \arg \max_{y_{1:T} \in Y} p(y_{1:T} | C, \Theta) \tag{3.11}$$

3.2.4 Automatically Acquiring Topic Labels

To avoid expensive human annotation and to improve the generalisability of our model, we propose to label the topic of each utterance of the datasets using LDA (Blei et al., 2003). While perplexity has been widely used for model selection for LDA, we employ a topic coherence measure proposed by (Röder et al., 2015) to determine the optimal topic number for each dataset, which combines the indirect cosine measure with the normalised pointwise mutual information (Bouma, 2009, NPMI) and the Boolean sliding window. Empirically, we found the latter yields much better topic clusters than perplexity for supporting DA classification.

We treat each conversation as a document and train topic models using Gensim with topic number settings ranging from 10 to 100 (using an increment step of 10). Gibbs sampling is used to estimate the model posterior and for each model we run 1,000 iterations. For each trained model, we calculate the averaged coherence score of the extracted topics using

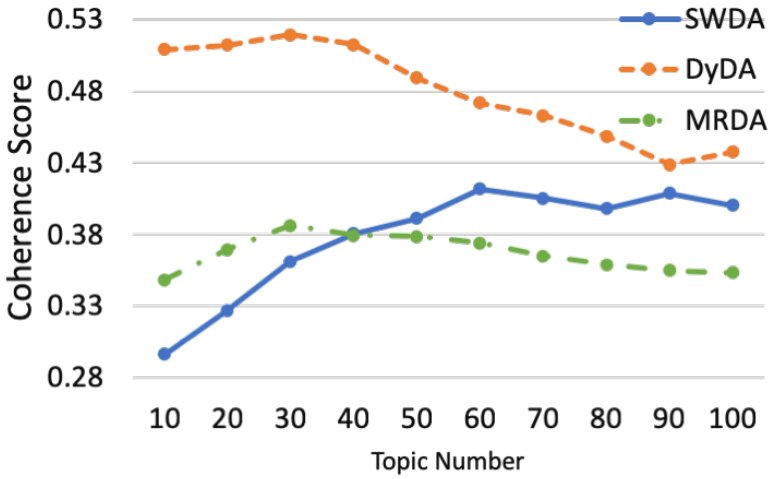


Figure 3.2: *Coherence score of LDA on three datasets.*

Gensim[1], an implementation following (Röder et al., 2015). Figure 3.2 shows the topic coherence score for each topic number setting for all datasets, from which we determine that the optimal topic number setting for SWDA, DyDA, and MRDA are 60, 30, and 30, respectively.

Based on the optimal models (i.e., a trained LDA model using the optimal topic number setting), we assign topic labels to the datasets with two different strategies, i.e., conversation-level labelling ($conv$) and utterance-level labelling (utt). For conversation-level labelling, we assign the topic label with the highest marginal probability to the conversation based on the corresponding per-document topic proportion estimated by LDA. Every utterance of the conversation then shares the same topic label of the conversation. Topic shift is common within the written and dialogue conversations when a current topic finishes and a new one starts (Qian and Jaeger, 2011; Xu and Reitter, 2016). Utterance-level topic labels are essential to track the status of the topic shift. Manually labelling utterance-level topics is time-consuming and a lack of generalisability for different tasks. Therefore, the utterance-level labelling using LDA to assign topic labels for each utterance in the datasets is a crucial step to alleviate the issue of manual topic labels. For utterance-level labelling, there is an additional step to perform inference on every utterance based on the corresponding optimal model (e.g., for every utterance of SWDA, we do inference using the LDA trained on SWDA with 60

[1]https://radimrehurek.com/gensim/models/coherencemodel.html

Dataset	$\|C\|$	$\|T\|$	$\|V\|$	Training	Validation	Testing
SWDA	42	66	20K	1003/193K	112/23K	19/5K
DyDA	4	10	22K	11K/92.7K	1K/8.5K	1K/8.2K
MRDA	5	-	15K	51/77.9K	11/15.8K	11/15.5K

Table 3.1: *$|C|$ is the number of DA classes, $|T|$ is the number of manually labelled conversation-level topic classes, $|V|$ is the vocabulary size. Training, Validation and Testing indicate the number of conversations/utterances in the respective splits.*

topics), and assign the topic label with the highest marginal probability to the utterance. Therefore, the topic labels of the utterances of the same conversation could be different for utterance-level labelling.

3.3 Experimental Settings

3.3.1 Datasets

We evaluate the performance of our model on three public DA datasets with different characteristics, namely, Switchboard (Jurafsky, 1997, SWDA), Dailydialog (Li et al., 2017b, DyDA), and the Meeting Recorder Dialogue Act corpus (Shriberg et al., 2004, MRDA).

3.3.1.1 Switchboard Dataset

SWDA[2] consists of 1,155 two-sided telephone conversations manually labelled with 66 conversation-level topics (e.g., *taxes*, *music*, etc.) and 42 utterance-level DAs (e.g., *statement-opinion*, *statement-non-opinion*, *wh-question*). The average speaker turns per conversation, tokens per conversation, and tokens per utterance are 195.2, 1,237.8, and 7.0, respectively. This dataset is highly imbalanced in terms of DAs: the DA labels *statement-non-opinion*, *acknowledge (backchannel)* and *statement-opinion* account for over 65% of the whole.

[2]http://compprag.christopherpotts.net/swda.html

3.3.1.2 Dailydialog Dataset

DyDA[3] contains 13,118 human-written daily conversations, manually labelled with 10 conversation-level topics (e.g., *tourism*, *politics*, *finance*) as well as four utterance-level DA classes, i.e., *inform*, *question*, *directive* and *commissive*. The former two classes are information transfer acts, while the latter two are action discussion acts. The dataset is also labelled with seven emotion labels and four dialogue act classes in the utterance level, such as *inform*, *question*, *directive* and *commissive*. The average speaker turns per conversation, tokens per conversation, and tokens per utterance are 7.9, 114.7, and 14.6, respectively. The definition of the four mutually-exclusive categories of DAs is as follows Li et al. (2017b).

3.3.1.3 Meeting Recorder Dialogue Act Dataset

MRDA[4] contains 75 meeting conversations annotated with 5 DAs, i.e., Statement (S), Question (Q), Floorgrabber (F), Backchannel (B), and Disruption (D). The average number of utterances per conversation is 1,496. There are no manually annotated topic labels available for this dataset. There are 11 general tags and 39 specific tags in the original MRDA tagset, but the most common usage is to group 11 general tags into 5 DAs, i.e., Statements (S), Questions (Q), Floorgrabber (F), Backchannel (B), and Disruption (D).

3.3.2 Implementation Details

For all experimental datasets, the top 85% highest frequency words were indexed. For SWDA and MRDA, we split training/validation/testing datasets following (Stolcke et al., 2000; Lee and Dernoncourt, 2016). For DyDA, we used the standard split from the original dataset (Li et al., 2017b). The statistics of the experimental datasets are summarised in Table 3.1. We represented input data with 300-dimensional Glove word embeddings (Pennington et al., 2014) and 50-dimensional character embeddings (Ma and Hovy, 2016). We set the dimension of the hidden layers (i.e., $\mathbf{h}_t^i$, $\mathbf{g}_t$ and $\mathbf{s}_t$) to 256 and applied a dropout layer to both the shared

[3] http://yanran.li/dailydialog
[4] http://www1.icsi.berkeley.edu/~ees/dadb/

encoder and the sequence tagger at a rate of 0.2. The Adam optimiser (Kingma and Ba, 2015) was used for training with an initial learning rate of 0.001 and a weight decay of 0.0001. Each utterance in a mini-batch was padded to the maximum length for that batch, and the maximum batch-size allowed was 50.

3.3.3 Baselines

We compare the proposed DAH-CRF model incorporating utterance-level topic labels extracted by LDA (denoted as DAH-CRF+LDA$_{utt}$) against five strong baselines and two variants of our own models:

JAS[5]: A generative joint, additive, sequential model of topics and speech acts in patient-doctor communication (Wallace et al., 2013);

DRLM-Cond[6]: A latent variable recurrent neural network for DA classification (Ji et al., 2016);

Bi-LSTM-CRF[7]: A hierarchical Bi-LSTM with a CRF to classify DAs (Kumar et al., 2018);

CRF-ASN: An attentive structured network with a CRF for DA classification (Chen et al., 2018);

SelfAtt-CRF: A hierarchical Bi-GRU with self-attention and CRF (Raheja and Tetreault, 2019);

DAH-CRF+MANUAL$_{conv}$: Use the manually annotated conversation-level topic labels (i.e., each utterance of the conversation shares the same topic) for DAH-CRF model training rather than the topic labels automatically acquired from LDA;

DAH-CRF+LDA$_{conv}$: Use conversation-level topic labels automatically acquired from LDA for DAH-CRF model training.

Note that only JAS (a non-deep-learning model) has attempted to model both DAs and topics, whereas all the deep learning baselines do not model topic information as a source of context for DA classification. All the baselines mentioned above use the same test dataset as

[5]https://github.com/bwallace/JAS

[6]https://github.com/jiyfeng/drlm

[7]https://github.com/YanWenqiang/HBLSTM-CRF

	Model	SWDA	MRDA	DyDA
Baselines	JAS	71.2	81.3	75.9
	DRLM-Cond	77.0†	88.4	81.1
	Bi-LSTM-CRF	79.2†	90.9†	83.6
	CRF-ASN	80.8†	91.4†	-
	SelfAtt-CRF	**82.9**†	91.1†	-
Ours	DAH-CRF + MANUAL$_{conv}$	80.9	-	86.5
	DAH-CRF + LDA$_{conv}$	80.7	91.2	86.4
	DAH-CRF + LDA$_{utt}$	82.3	**92.2**	**88.1**
	Human Agreement	84.0	-	-

Table 3.2: *DA classification accuracy.* † *indicates the results which are reported from the prior publications.*

our models for all experimental datasets.

3.4 Experimental Results

3.4.1 Dialogue Acts Classification

Table 3.2 shows the DA classification accuracy of our models and the baselines on three experimental datasets. We fine-tuned the model parameters for JAS, DRLM-Cond and Bi-LSTM-CRF in order to make the comparison as fair as possible. The implementation of CRF-ASN and SelfAtt-CRF are not available so we can only report their results for SWDA and MRDA based on the original papers (Chen et al., 2018; Raheja and Tetreault, 2019).

It can be observed that by jointly modelling DA and topics, DAH-CRF+LDA$_{utt}$ outperforms the two best baseline models SelfAtt-CRF and CRF-ASN around 1% on the MRDA dataset. Our model also gives similar performance to SelfAtt-CRF, the baseline which achieved the state-of-the-art performance on the SWDA dataset (i.e., 82.3% vs. 82.9%). While both manually annotated and automatically acquired topic labels are effective, we see that DAH-CRF+LDA$_{utt}$ outperforms both DAH-CRF+MANUAL$_{conv}$ and DAH-CRF+LDA$_{conv}$, i.e., with over 1.6% gain on DyDA and over 1.4% on SWDA (significant; paired t-test $p < .01$). It is also observed that DAH-CRF+MANUAL$_{conv}$ and DAH-CRF+LDA$_{conv}$ perform very similar to each other.

Model	SWDA	MRDA	DyDA
SAH	76.2	88.5	82.5
SAH-CRF	78.4	89.6	84.1
DAH + LDA$_{utt}$	79.5	91.1	86.0
DAH-CRF + LDA$_{utt}$ (without Dual-Att)	81.0	91.3	86.3
DAH-CRF + LDA$_{utt}$	82.3	92.2	88.1

Table 3.3: *Ablation studies of DA classification.*

3.4.2 Ablation Study Results

We conducted ablation studies (see Table 3.3) in order to evaluate the contribution of the components of our DAH-CRF+LDA$_{utt}$ model, and more importantly, the effectiveness of leveraging topic information for supporting DA classification.

DAH-CRF+LDA$_{utt}$ (without Dual-Att) removes the dual-attention component from DAH-CRF+LDA$_{utt}$, and DAH+LDA$_{utt}$ removes the CRF from DAH-CRF+LDA$_{utt}$ but retaining the dual-attention component. SAH is a Single-Attention Hierarchical RNN model without a CRF, i.e., a simplified version of DAH+LDA$_{utt}$ that only models DAs with topical information omitted. As can be seen in Table 3.3, DAH+LDA$_{utt}$ achieves over 3% averaged gain on all datasets when compared to SAH, which clearly shows that leveraging topic information can effectively support DA classification. It is also observed that both the dual-attention mechanism and the CRF component are beneficial, but are more effective on the SWDA and DyDA datasets than MRDA.

In summary, while all the analysed model components are beneficial, the biggest gain is obtained by jointly modelling DAs and topics.

3.4.3 Analysing the Effectiveness of Joint Modelling Dialogue Act and Topic

In this section, we provide detailed analysis on why DAH-CRF+LDA$_{utt}$ can yield better performance than SAH-CRF by jointly modelling DAs and topics.

Figure 3.3 shows the normalized confusion matrix derived from 10 DA classes of

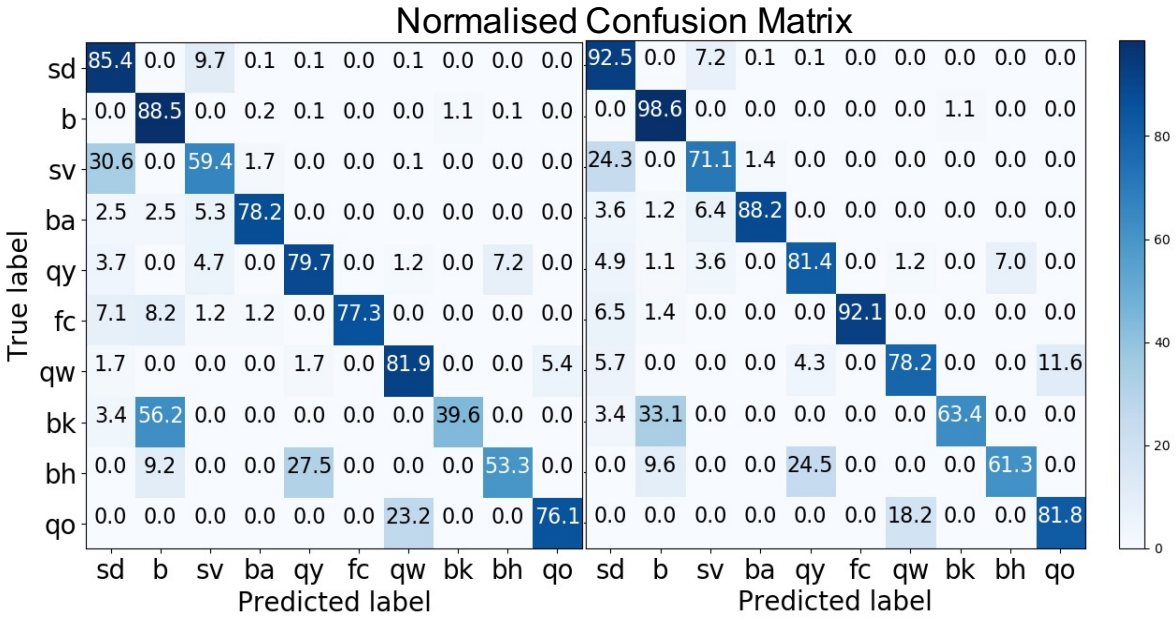


Figure 3.3: *The normalized confusion matrix of DAs using SAH-CRF (left) and DAH-CRF+LDA$_{utt}$ (right) on SWDA dataset.*

SWDA for both SAH-CRF and DAH-CRF+LDA$_{utt}$ models. It can be observed that DAH-CRF+LDA$_{utt}$ yields improvement on recall for many DA classes compared to SAH-CRF, e.g., 23.8% improvement on *bk* and 11.7% on *sv*. For *bk* (`Response Acknowledge`) which has the highest improvement level, we see that the improvement largely comes from the reduction of misclassifing *bk* to *b* (`Acknowledge Backchannel`). The key difference between *bk* and *b* is that an utterance labelled with *bk* has to be produced within a question-answer context, whereas *b* is a "continuer" simply representing a response to the speaker (Jurafsky, 1997). It is not surprising that SAH-CRF makes poor prediction on the utterances of these two DAs: they share many syntactic cues, e.g., indicator words such 'okay', 'oh', and 'uh-huh', which can easily confuse the model. When comparing the topic distribution of the utterances under the *bk* and *b* categories (cf. Figure 3.5), we found topics relating to personal leisure (e.g., buying cars, music, and exercise) are much more prominent in *bk* than *b*. By leveraging the topic information, DAH-CRF+LDA$_{utt}$ can better handle the confusion cases and hence improve the prediction for *bk* significantly.

There are also cases where DAH-CRF+LDA$_{utt}$ performs worse than SAH-CRF. Take the

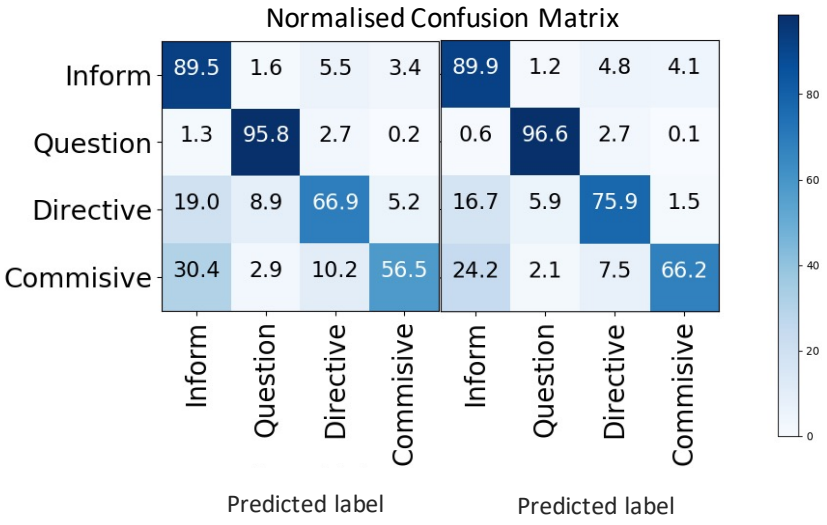


Figure 3.4: *The normalized confusion matrix of DAs using SAH-CRF (left) and DAH-CRF+LDA$_{utt}$ (right) on DyDA dataset.*

DA pair of *qo* (`Open Question`) and *qw* (`wh-questions`) as an example. *qo* refers to questions like '*How about you?*' and its variations (e.g., '*What do you think?*'), whereas *qw* represents wh-questions which are much more specific in general (e.g. '*What other long range goals do you have?'*). SAH-CRF gives quite decent performance in distinguishing *qw* and *qo* classes. This is somewhat reasonable, as linguistically the utterances of these two classes are quite different, i.e., the *qw* utterance expresses very specific question and is relatively lengthy, whereas *qo* utterances tends to be very brief. We see that DAH-CRF+LDA$_{utt}$ performs worse than SAH-CRF: a greater number of *qw* utterances are misclassified by DAH-CRF+LDA$_{utt}$ as *qo*. This might be attributed to the fact that topic distributions of *qw* and *qo* are similar to each other (see Figure 3.5), i.e., incorporating the topic information into DAH-CRF may cause these two DAs to be less distinguishable for the model.

We also conducted a similar analysis on the DyDA dataset. As can be seen from the confusion matrices shown in Figure 3.4, DAH-CRF+LDA$_{utt}$ gives improvement over SAH-CRF for all the four DA classes of DyDA. In particular, `Directives` and `Commissive` achieve higher improvement margin compared to the other two classes, where the improvement are largely attributed to less number of instances of the `Directives` and `Commissive`

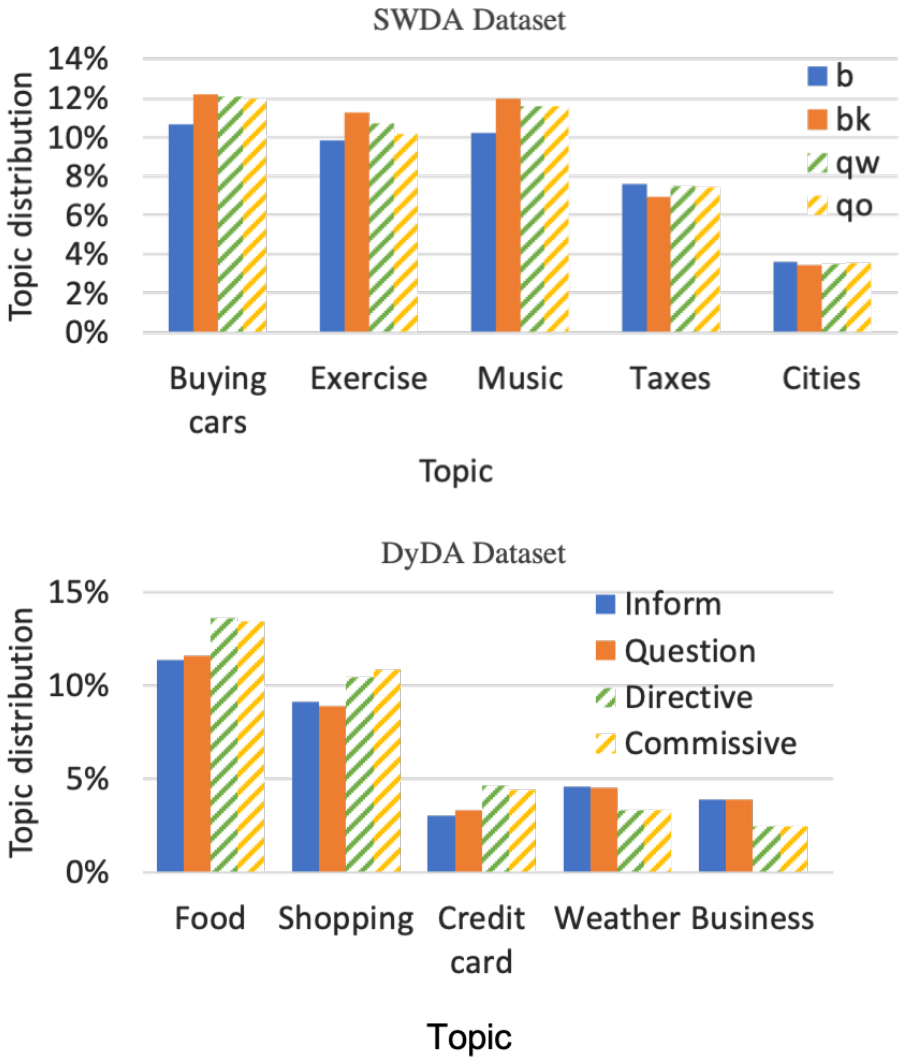


Figure 3.5: *We highlight the prominent topics for some example DAs. The topic distribution of a topic k under a DA label d is calculated by averaging the marginal probability of topic k for all utterances with the DA label d.*

classes being mis-classified into `Inform` and `Questions`. Examining the topic distributions in Figure 3.5 reveals that `Directives` and `Commissive` classes are more relevant to the topics such as *food*, *shopping*, and *credit card*. In contrast, the topics of `Inform` and `Questions` classes are more about *business*, and *weather*.

Finally, Figure 3.6 shows the DA attention visualisation examples of SAH-CRF and DAH-CRF+LDA$_{utt}$ for an utterance from SWDA and DyDA. For SWDA, it can be seen that

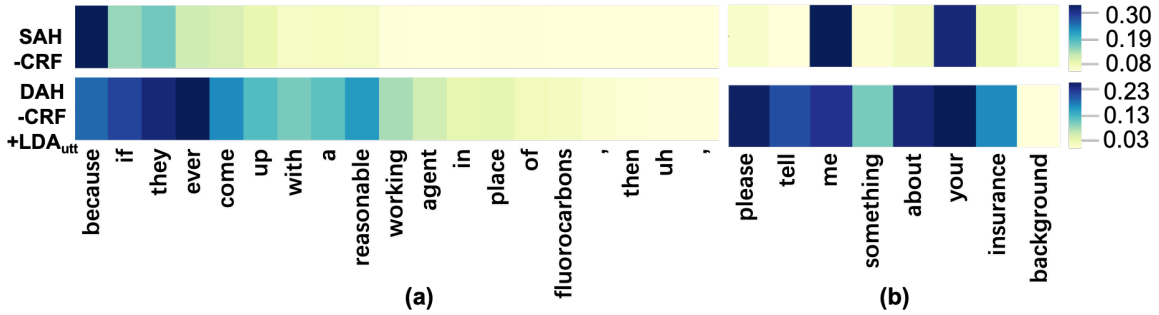


Figure 3.6: *DA Attention visualisation using SAH-CRF and DAH-CRF+LDA$_{utt}$ on (a) SWDA and (b) DyDA datasets. The true labels of the utterances above are sd (statement-non-opinion) and Directive, respectively. SAH-CRF misclassified the DA as sv (statement-opinion) and Inform whereas DAH-CRF+LDA$_{utt}$ gives correct prediction for both cases.*

SAH-CRF gives very high weight to the word "because" and de-emphasizes other words. However, DAH-CRF+LDA$_{utt}$ can capture more important words (e.g., "if", "reasonable", etc.) and correctly predicts the DA label as *sd*. For DyDA, SAH-CRF only focuses on "me" and "your", but DAH-CRF+LDA$_{utt}$ captures more words relevant to `Directive`, such as "please", "tell", etc. To summarise, DAH-CRF+LDA$_{utt}$ can capture more significant words related to the corresponding DA, by modelling both DAs and topic information with the dual-attention mechanism.

3.5 Conclusion

In this chapter, we developed a dual-attention hierarchical recurrent neural network with a CRF for DA classification. With the proposed task-specific dual-attention mechanism, our model is able to capture information about both DAs and topics, as well as information about the interactions between them. Moreover, our model is generalised by leveraging an unsupervised model to automatically acquire topic labels. Experimental results based on three public datasets show that modelling utterance-level topic information as an auxiliary task can effectively improve DA classification, and that our model is able to achieve better or comparable performance to the state-of-the-art deep learning methods for DA classification.

We envisage that our idea of modelling topic information for improving DA classification can be adapted to other DNN models, e.g., to encode topic labels into word embeddings and then concatenate with the utterance-level or conversation-level hidden vectors of our baselines, e.g. SelfAtt-CRF. It will also be interesting to explicitly take into account speaker's role in the future.

Chapter 4

Improving Variational Autoencoder for Text Modelling with Timestep-Wise Regularisation

4.1 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., 2017b), 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).

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 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 this work 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.

4.2 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.

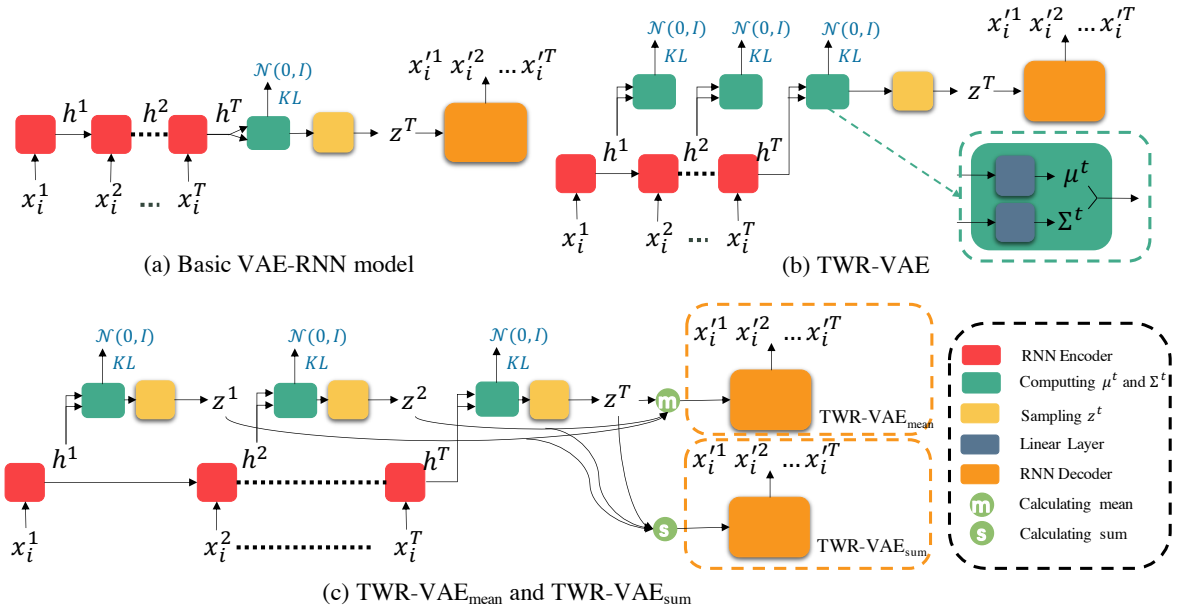


Figure 4.1: *Architectures of the proposed TWR-VAE models and the basic VAE-RNN model.*

4.2.1 Background of VAE

As introduced in § 2.1.3, VAE is trained to maximise the ELBO, which consists of two terms (Kingma and Welling, 2014):

$$
\begin{aligned}
\log P_{\boldsymbol{\theta}}(\mathbf{x}_i) =& \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}\left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i)\right] \\
=& \mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x}_i)}\left[\log\left[\frac{P_\theta(\mathbf{x}_i,\mathbf{z})}{P_\theta(\mathbf{z}|\mathbf{x}_i)}\right]\right] \\
=& \mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x}_i)}\left[\log\left[\frac{P_\theta(\mathbf{x}_i,\mathbf{z})}{Q_\phi(\mathbf{z}|\mathbf{x}_i)}\frac{Q_\phi(\mathbf{z}|\mathbf{x}_i)}{P_\theta(\mathbf{z}|\mathbf{x}_i)}\right]\right] \\
=& \underbrace{\mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}\left[\log\left[\frac{P_{\boldsymbol{\theta}}(\mathbf{x}_i,\mathbf{z})}{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}\right]\right]}_{=\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}_i)\ \text{(ELBO)}} + \underbrace{\mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}\left[\log\left[\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}{P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_i)}\right]\right]}_{=D_{\mathrm{KL}}\left(Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)\|P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_i)\right)},
\end{aligned}
\tag{4.1}
$$

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i) =& \log P_{\boldsymbol{\theta}}(\mathbf{x}_i) - \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}\left[\log\left[\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}{P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_i)}\right]\right] \\
=& \int Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i) \log P_{\boldsymbol{\theta}}(\mathbf{x}_i) d\mathbf{z} - \int Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i) \log \frac{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}{P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_i)} d\mathbf{z} \\
=& \int Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i) \log \frac{P_{\boldsymbol{\theta}}(\mathbf{x}_i) P_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_i)}{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)} d\mathbf{z} \\
=& \int Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i) \log \frac{P_{\boldsymbol{\theta}}(\mathbf{z}) P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z})}{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)} d\mathbf{z} \\
=& \int Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i) \log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}) d\mathbf{z} + \int Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i) \log \frac{P_{\boldsymbol{\theta}}(\mathbf{z})}{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)} d\mathbf{z} \\
=& \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z})] - D_{\mathrm{KL}}\left(Q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i) \| P(\mathbf{z})\right) , \qquad (4.2)
\end{aligned}
$$

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 4.1(a)) follows the aforementioned ELBO (i.e. Eq. 4.2). As the architecture of the 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:

$$
\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i)_{\text{basic}} = \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T)] - D_{\mathrm{KL}}\left(Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i) \| P(\mathbf{z}^T)\right) , \qquad (4.3)
$$

Note that the total number of timestep T is also the length of the input sentence. As discussed, optimising ELBO (in Eq. 4.3) is prone to posterior collapsing to the prior (Bowman et al., 2016). This phenomenon happens when the second term of Eq. 4.3 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.

4.2.2 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 4.1(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 4.1(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:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i)_{\text{TWR}} = \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T)] - \frac{1}{T}\sum_{t=1}^{T} D_{\text{KL}}(Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})\|P(\mathbf{z}^t))\,, \tag{4.4}$$

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. 4.4. Such a strategy allows more robust model learning and can effectively mitigate posterior collapse (see §4.3 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.

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:

$$\begin{aligned}\mathbf{z}^t =& Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t}) \\ =& g_{\boldsymbol{\phi}}(\mathbf{h}^t, \boldsymbol{\epsilon}^t|\mathbf{x}_i^{1:t}) \\ =& \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{h}^t|\mathbf{x}_i^{1:t})^{1/2}\boldsymbol{\epsilon}^t + \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{h}^t|\mathbf{x}_i^{1:t}) \,,\end{aligned} \tag{4.5}$$

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 $\mathbf{h}^t$.

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 $\boldsymbol{\phi}$:

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

$$= \nabla_{\boldsymbol{\theta}}\left(\mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}\left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T)\right] - \frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}\left(Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})\|P(\mathbf{z}^t)\right)\right) \tag{4.6}$$

$$= \nabla_{\boldsymbol{\theta}}\left(\mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}\left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right]\right) \tag{4.7}$$

$$= \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}\left[\nabla_{\boldsymbol{\theta}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right)\right] \tag{4.8}$$

$$\simeq \frac{1}{M}\sum_{m=1}^{M}\nabla_{\boldsymbol{\theta}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_m^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}_m^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}_m^t)}\right) \quad \text{where} \quad \mathbf{z}_m^T \sim Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i) \tag{4.9}$$

$$= \frac{1}{M}\sum_{m=1}^{M}\nabla_{\boldsymbol{\theta}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_m^T)\right) \quad \text{where} \quad \mathbf{z}_m^T \sim Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)\,, \tag{4.10}$$

which is an unbiased Monte Carlo gradient estimator to approximate the expectation (Eq. 4.6), 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:

$$\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}_i) = \nabla_{\boldsymbol{\phi}}\left(\mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}\left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right]\right) \tag{4.11}$$

$$\neq \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^T|\mathbf{x}_i)}\left[\nabla_{\boldsymbol{\phi}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)}\right)\right], \tag{4.12}$$

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. 4.5). We choose $\mathcal{N}(\mathbf{0},\mathbf{I})$ as $P(\boldsymbol{\epsilon})$ and we can use the Monte Carlo estimator approximate

Eq. 4.11:

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

$$= \nabla_{\boldsymbol{\phi}} \left(\mathbb{E}_{P(\boldsymbol{\epsilon})} \left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T} \sum_{t=1}^{T} \log \frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)} \right] \right) \tag{4.13}$$

$$= \mathbb{E}_{P(\boldsymbol{\epsilon})} \left[\nabla_{\boldsymbol{\phi}} \left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T} \sum_{t=1}^{T} \log \frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)} \right) \right] \tag{4.14}$$

$$\simeq \frac{1}{M} \sum_{m=1}^{M} \nabla_{\boldsymbol{\phi}} \left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_m^T) - \frac{1}{T} \sum_{t=1}^{T} \log \frac{Q_{\boldsymbol{\phi}}(\mathbf{z}_m^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}_m^t)} \right) \tag{4.15}$$

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

$$\text{where} \quad \mathbf{z}_m^t = g_{\boldsymbol{\phi}}^t \left(\boldsymbol{\epsilon}_m, \mathbf{x}_i^{1:t} \right) \quad \text{and} \quad \boldsymbol{\epsilon}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{4.16}$$

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

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

$$= \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \left(\mathbb{E}_{P(\boldsymbol{\epsilon})} \left[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T} \sum_{t=1}^{T} \log \frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)} \right] \right) \tag{4.17}$$

$$= \mathbb{E}_{P(\boldsymbol{\epsilon})} \left[\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^T) - \frac{1}{T} \sum_{t=1}^{T} \log \frac{Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}^t)} \right) \right] \tag{4.18}$$

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

$$\text{where} \quad \mathbf{z}_m^t = g_{\boldsymbol{\phi}}^t \left(\boldsymbol{\epsilon}_m, \mathbf{x}_i^{1:t} \right) \quad \text{and} \quad \boldsymbol{\epsilon}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \,, \tag{4.19}$$

$$\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}_i)_{\text{TWR}} \simeq \frac{1}{M}\sum_{m=1}^{M}\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\left(\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_m^T) - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_{\boldsymbol{\phi}}(\mathbf{z}_m^t|\mathbf{x}_i^{1:t})}{P(\mathbf{z}_m^t)}\right)$$
$$\text{where} \quad \mathbf{z}_m^t = Q_{\boldsymbol{\phi}}(\mathbf{z}_m^t|\mathbf{x}_i^{1:t})\,, \tag{4.20}$$

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.

4.2.3 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 4.1(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. 4.4):

$$\mathbb{E}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\frac{1}{T}\sum_{t=1}^{T}\mathbf{z}^t)] \qquad \text{where } \mathbf{z}^t \sim Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t}) \tag{4.21}$$

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:

$$\mathbb{E}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\sum_{t=1}^{T}\mathbf{z}^t)] \qquad \text{where } \mathbf{z}^t \sim Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t}) \tag{4.22}$$

For both TWR-VAE$_{\text{mean}}$ and TWR-VAE$_{\text{sum}}$, their KL loss term is the same as TWR-VAE,

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 4.1: *The statistics of the PTB, Yelp 2015, Yahoo, SW and DD datasets.*

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 4.2: *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.*

i.e., $-\frac{1}{T}\sum_{t=1}^{T} D_{\text{KL}}(Q_{\boldsymbol{\phi}}(\mathbf{z}^t|\mathbf{x}_i^{1:t})\|P(\mathbf{z}^t))$.

4.3 Experiment

4.3.1 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 4.1. 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. 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.

We compare our TWR-VAE model with five strong baselines:

VAE-LSTM[1]: A VAE with LSTM and with KL annealing for tackling the posterior collapse issue (Bowman et al., 2016);

SA-VAE[2]: A VAE using stochastic variational inference to refine the variational parameters initialised by Amortized variational inference (Kim et al., 2018);

Cyclical VAE[3]: A VAE employing cyclical annealing to alleviate the posterior collapse issue (Fu et al., 2019);

Lagging VAE[4]: A VAE updating the encoder more times than updating the decoder (He et al., 2019);

BN-VAE[5]: 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 $I(\mathbf{x}, \mathbf{z}) = \mathbb{E}_{\mathbf{x}}[D_{\mathrm{KL}}\left(Q_\phi(\mathbf{z}^T|\mathbf{x})\|P(\mathbf{z}^T)\right)] - D_{\mathrm{KL}}(Q_\phi(\mathbf{z}^T)\|P(\mathbf{z}^T))$, where $Q_\phi(\mathbf{z}^T)$ is an aggregated posterior and $D_{\mathrm{KL}}(Q_\phi(\mathbf{z}^T)\|P(\mathbf{z}^T))$ is the KL divergence between the aggregated posterior and the prior estimated by Monte Carlo estimators:

[1]https://github.com/timbmg/Sentence-VAE
[2]https://github.com/harvardnlp/sa-vae
[3]https://github.com/haofuml/cyclical_annealing
[4]https://github.com/jxhe/vae-lagging-encoder
[5]https://github.com/valdersoul/bn-vae

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

$$\mathbb{E}_{\mathbf{x}}[D_{\text{KL}}\left(Q_\phi(\mathbf{z}^T|\mathbf{x})\|P(\mathbf{z}^T)\right)] \tag{4.23}$$

$$= \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{Q_\phi(\mathbf{z}^T|\mathbf{x})}[\log Q_\phi(\mathbf{z}^T|\mathbf{x})]] - \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{Q_\phi(\mathbf{z}^T|\mathbf{x})}[\log P(\mathbf{z}^T)]] \tag{4.24}$$

$$= -H(Q_\phi(\mathbf{z}^T|\mathbf{x})) - \mathbb{E}_{Q_\phi(\mathbf{z}^T)}[\log P(\mathbf{z}^T)] \tag{4.25}$$

$$= -H(Q_\phi(\mathbf{z}^T|\mathbf{x})) + H(Q_\phi(\mathbf{z}^T)) - H(Q_\phi(\mathbf{z}^T)) - \mathbb{E}_{Q_\phi(\mathbf{z}^T)}[\log P(\mathbf{z}^T)] \tag{4.26}$$

$$= I(\mathbf{x}, \mathbf{z}^T) + \mathbb{E}_{Q_\phi(\mathbf{z}^T)}[\log Q_\phi(\mathbf{z}^T)] - \mathbb{E}_{Q_\phi(\mathbf{z}^T)}[\log P(\mathbf{z}^T)] \tag{4.27}$$

$$= I(\mathbf{x}, \mathbf{z}^T) + D_{\text{KL}}(Q_\phi(\mathbf{z}^T)\|P(\mathbf{z}^T))\,, \tag{4.28}$$

Therefore:

$$I(\mathbf{x}, \mathbf{z}^T) = \mathbb{E}_{\mathbf{x}}[D_{\text{KL}}\left(Q_\phi(\mathbf{z}^T|\mathbf{x})\|P(\mathbf{z}^T)\right)] - D_{\text{KL}}(Q_\phi(\mathbf{z}^T)\|P(\mathbf{z}^T))\,, \tag{4.29}$$

4.3.1.1 Results

As depicted in Table 4.2, 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.

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 4.4: *An example of interpolating the latent representation of two input sentences using BN-VAE and TWR-VAE in Yelp15 testset.*

As shown in Table 4.2, 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.

4.3.1.2 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

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

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 4.3 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 4.3 show that TWR-VAE$_{\text{LSTM-last25}}$ has the lowest performance on NLL and PPL and the performance goes

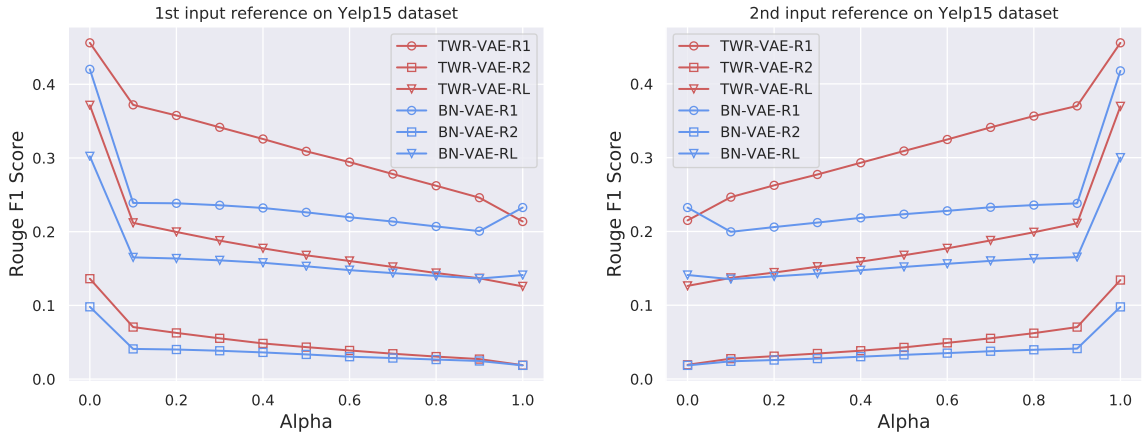
Metrics	Switchboard					Dailydialog				
	SeqGAN	CVAE	WAE	iVAE	TWR-VAE	SeqGAN	CVAE	WAE	iVAE	TWR-VAE
BLEU-R↑	0.282	0.295	0.394	**0.427**	0.395	0.270	0.265	0.341	0.355	**0.407**
BLEU-P↑	**0.282**	0.258	0.254	0.254	0.258	0.270	0.222	0.278	0.239	**0.281**
BLEU-F1↑	0.282	0.275	0.309	**0.319**	0.312	0.270	0.242	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 4.6: *Dialogue response generation results of baselines and our model on SW and DD datasets.*

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).

4.3.1.3 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 them with $\mathbf{z}_\alpha^T = \mathbf{z}_1^T \cdot (1-\alpha) + \mathbf{z}_2^T \cdot \alpha$. Table 4.4 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



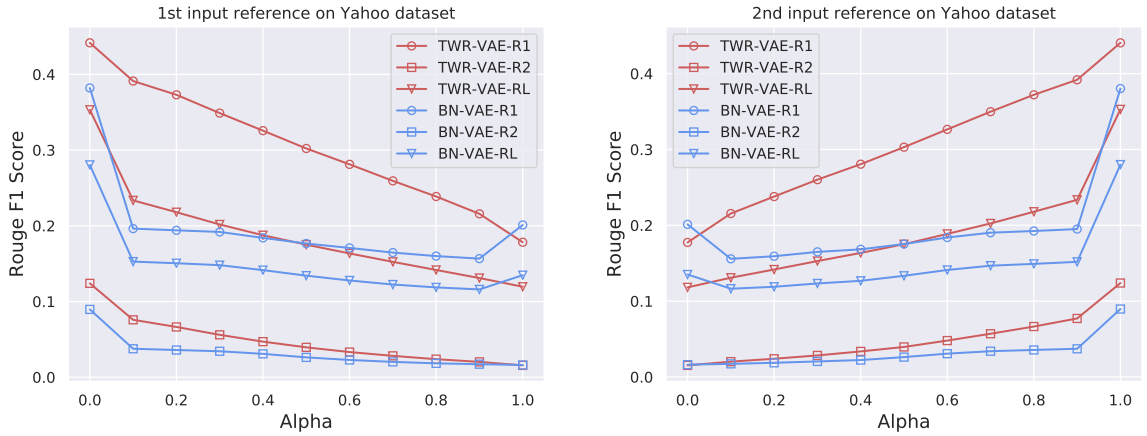
(a) **(b)**

Figure 4.2: *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.*

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 4.2. 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.

4.3.2 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



(a) **(b)**

Figure 4.3: *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.*

(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:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i)_{\text{TWR}} = \mathbb{E}_{Q_{\boldsymbol{\phi}}(\mathbf{z}^J|\mathbf{x}_i,\mathbf{c})}[\log P_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}^J, \mathbf{c})] - \frac{1}{J}\sum_{j=1}^{J} D_{\text{KL}}(Q_{\boldsymbol{\phi}}(\mathbf{z}^j|\mathbf{x}_i, \mathbf{c})\|P_{\boldsymbol{\theta}}(\mathbf{z}^j|\mathbf{c}))\,. \tag{4.30}$$

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
	1. yeah uh - huh 2. yeah and then go back up and go back and forth and go back again 3. right oh that makes 4. she's not
	TWR-VAE
	1. uh - huh 2. i see yeah and they have to go back to work and it's really sad 3. oh gosh they don't have to worry about 4. hm how do you feel
Example 2: **Topic**: Relationship **Context**: what happened , john ? **Target**: nothing .	iVAE
	1. oh , i am . 2. what can we do for you ? 3. oh what's wrong ? i didn't know anyone . 4. i have to get my phone .
	TWR-VAE
	1. i can't sleep well . 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 , 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. well , i am not sure of it .

Table 4.7: *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 and the actual context window is 10.*

4.3.2.1 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 4.1. 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.

Apart from comparing TWR-VAE to CVAE and iVAE, we further report the results of two other competitive models for dialogue response generation[6], i.e., **SeqGAN** (Li et al., 2017a) and a conditional 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;

[6] **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

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:

   (a) Greedy: the average cosine similarities between word embeddings of the two utterances which are greedily matched (Rus and Lintean, 2012);

   (b) Average: the cosine similarity between the averaged word embeddings in the two utterances (Mitchell and Lapata, 2008);

   (c) 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.

4.3.2.2 Experiment Results

As shown in Table 4.6, 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 4.7 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.

4.4 Conclusion

In this chapter, 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.

Chapter 5

Understanding Latent Discontinuity of VAEs for Text Generation

5.1 Introduction

Variational Auto-Encoders (VAEs) are powerful unsupervised models for learning low-dimensional manifolds (aka. a latent space) from non-trivial high-dimensional data manifolds (Kingma and Welling, 2014; Rezende et al., 2014). They have found successes in a number of downstream tasks across different application domains such as text classification (Xu et al., 2017), transfer learning (Higgins et al., 2017b), image synthesis (Huang et al., 2018; Razavi et al., 2019), language generation (Bowman et al., 2016; He et al., 2019), and music composition (Roberts et al., 2018).

Various effort has been made to improve the capacity of VAEs, where the majority of the extensions are focused on increasing the flexibility of the prior and approximating posterior. For instance, Davidson et al. (2018) introduced the von Mises-Fisher (vMF) distribution to replace the standard Gaussian distribution; Kalatzis et al. (2020) assumed a Riemannian structure over the latent space by adopting the Riemannian Brownian motion prior. A small number of recent studies attempted to investigate the problem more fundamentally, and revealed that there exist discontinuous regions (we refer them as "*latent holes*" following past

literature) in the latent space of VAEs, which have a detrimental effect on model capacity. Falorsi et al. (2018) approached the problem from a theoretical perspective of *manifold mismatch* and showed that this undesirable phenomenon is due to the latent space's topological incapability of accurately capturing the properties of a dataset. Xu et al. (2020) examined the obstacles that prevent sequence VAEs from performing well in unsupervised controllable text generation, and empirically discovered that manipulating the latent variables for semantic variations in text often leads to latent variables to reside in some latent holes. As a result, the decoding network fails to properly decode or generalise when the sampled latent variables land in those low-density latent regions.

Although the works on investigating latent holes are still relatively sparse, they have opened up new opportunities for improving VAE models, where one can design mechanisms directly engineered for mitigating the hole issue. However, it should be noted that existing works (Falorsi et al., 2018; Xu et al., 2020) exclusively focus on the encoder network when investigating holes in the latent space, and they merely explored its *existence* without providing further in-depth analysis of the phenomenon. It has also been revealed that the hole issue is more severe on text compared to the image domain, due to the discreteness of text data (Xu et al., 2020).

We tackle the aforementioned issues by proposing a novel tree-based decoder-centric (`TDC`) algorithm for latent hole identification, with a focus on the text domain. In contrast to existing works which are encoder-centric, our approach is centric to the decoder network, as a decoder has a direct impact to model's performance, e.g., for text generation. Our `TDC` algorithm is also highly efficient for latent hole searching when compared to existing approaches, owing to the dimension reduction and Breadth-First Search strategies. Another important technical contribution we have made is that we theoretically unify the two prior indicators for latent hole identification, and evidence that the one of Falorsi et al. (2018) is more accurate, which forms the basis of our algorithm detailed in § 5.3.

In terms of analysing the latent hole phenomenon, we provide, for the first time, an in-depth empirical analysis which examines three important aspects: (i) how the holes impact VAE models' performance on text generation; (ii) whether the holes are really *vacant*, i.e., no

useful information is captured by the holes; and (iii) how the holes are distributed in the latent space. To validate our theory and to demonstrate the generalisability of our proposed `TDC` algorithm, we pre-train five strong and representative VAE models for producing sentences, including the state-of-the-art model. Comprehensive experiments on the language generation task involving four large-scale public datasets show that the performance of text generation is strongly correlated with the density of latent holes; that from the perspective of the decoder, the Latent Vacancy Hypothesis proposed by Xu et al. (2020) does not hold empirically; and that holes are ubiquitous and densely distributed in the latent space.

5.2 Preliminaries

5.2.1 Existing Latent Hole Indicators

To our knowledge, there are only two prior works which directly determine whether a latent region is continuous or not. One work formalises latent holes based on the relative distance of pairwise points taken from the latent space and the sample space (Falorsi et al., 2018). Concretely speaking, given a pair of vectors $\mathbf{z}_i$ and $\mathbf{z}_{i+1}$ which are closely located on a latent path, and their corresponding samples $\mathbf{x'}_i$ and $\mathbf{x'}_{i+1}$ in the sample space, a latent hole indicator is computed as

$$\mathcal{I}_{\text{Lipschitz}}(i) := \mathbb{D}_{\text{sample}}(\mathbf{x'}_i, \mathbf{x'}_{i+1}) / \mathbb{D}_{\text{latent}}(\mathbf{z}_i, \mathbf{z}_{i+1}), \tag{5.1}$$

where $\mathbb{D}_{\text{sample}}$ and $\mathbb{D}_{\text{latent}}$ respectively denote the metrics measuring the sample and latent spaces (Note: $\mathbb{D}_{\text{latent}}$ can be represented as any distance metric, such as Euclidean distance metric, Riemannian distance metric and so on). Falorsi et al. (2018) focused on the image domain and utilised Euclidean distance for both spaces. Based on the concept of Lipschitz continuity, Falorsi et al. (2018) then proposed to measure the continuity of a latent region as follows: under the premise that $\mathbf{z}_{i+1}$ does not land on a hole, $\mathbf{z}_i$ *is recognised as belonging to a hole if the corresponding* $\mathcal{I}_{\text{Lipschitz}}(i)$ *is a large outlier*[1].

[1]Unless otherwise stated, outliers are detected by comparing the subject data point with a fixed bound, which is pre-determined based on a percentile of all data points.

Another line of work (Xu et al., 2020) signals latent holes based on the so-called *aggregated posterior*, with a focus on sequence VAEs for language modelling. This approach interpolates a series of vectors on a latent path at a small interval, and then scores the i-th latent vector $\mathbf{z}_i$ as

$$\mathcal{I}_{\text{Aggregation}}(i) := \sum_{t=1}^{M} \text{NLL}(\mathbf{z}_i, \mathbf{Z}^{(t)})/M, \tag{5.2}$$

where $\mathbf{Z}^{(t)}$ is the sample of the posterior distribution of the t-th out of the total M training samples, e.g., when studying holes on the encoder side, this distribution can be computed using $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ in Eq. (2.7) (Xu et al., 2020). $\mathbf{Z}^{(t)}$ serves as the reference when calculating the Negative Log-Likelihood (NLL). After all the interpolated vectors on the latent path are traversed, similar to the first method, vectors with large outlier indicators ($\mathcal{I}_{\text{Aggregation}}$) are identified as in latent holes.

When comparing these two indicators, one can see that they actually stem from different intuitions. For $\mathcal{I}_{\text{Lipschitz}}$, there is an underlying assumption that a mapping between the sample and latent spaces should have good stability in terms of *relative* distance change in order to guarantee good continuity in the latent space. In contrast, $\mathcal{I}_{\text{Aggregation}}$ is based on the belief that small perturbations on the non-hole regions should not lead to large offsets on the *absolute* dissimilarity between posterior samples $\mathbf{Z}^{(\cdot)}$ and the sample $\mathbf{z}_i$, and hence the calculation is performed only in the latent space and only around one single latent position. While seemly different, we show that (in § 5.3.2) both indicators actually have tight underlying connections and can be unified in a shared mathematical framework. Moreover, the first indicator ($\mathcal{I}_{\text{Lipschitz}}$) is proofed to be more comprehensive than the second ($\mathcal{I}_{\text{Aggregation}}$) and thus can reduce false negatives when identifying holes in the latent space. This forms the basis of our algorithm in § 5.3, which is the first attempt to identify a VAE decoder's latent holes for language generation.

5.3 Methodology

In this section, we describe our tree-based decoder-centric (TDC) algorithm for latent hole identification, which consists of three main components. We first introduce our heuristic-based Breadth-First Search (BFS) algorithm for highly efficient latent space searching (§ 5.3.1). We then theoretically proof, for the first time, that two existing holes indicators can be unified under the same framework and that $\mathcal{I}_{\text{Lipschitz}}$ is a more accurate choice for identifying latent holes (§ 5.3.2). Finally, we extend $\mathcal{I}_{\text{Lipschitz}}$ to the text domain by incorporating the Wasserstein distance for the sample space (§ 5.3.3).

5.3.1 Tree-based Decoder-Centric Latent Hole Identification

As discussed earlier, existing works for investigating latent holes of VAEs all exclusively focus on the encoder network (Falorsi et al., 2018; Xu et al., 2020), and they cannot be trivially applied to the decoders (which play ultimately important roles on generation tasks) due to metric incompatibility, especially for VAEs in the text domain (see detailed discussion in § 5.3.3). Another drawback of existing indicators is that they have very limited efficiency. Theoretically, their time complexity for traversing a d-dimensional latent space with I interpolations per path is $\mathcal{O}(I^d)$ at the *optimal efficiency*, which is computationally prohibitive as typically d and I are larger than 30 and 50 for VAEs in practice. Each path is parallel with one axis of the traversed latent space[2]. Empirically, we observe that even finding *a handful of* latent holes has been shown to be difficult for existing methods (Falorsi et al., 2018; Xu et al., 2020). Therefore, we tackle both challenges by proposing a highly efficient algorithm for decoder-centric latent hole identification. The pipeline of our TDC algorithm is described in Algorithm 1 and we give a detailed discussion as follows. For the visualisation of TDC's working process in practice, please see Figure. 5.1.

[2]For example, we traverse a 3-dimensional latent space with 4 interpolations per path. There will be $4^3 = 64$ points to traverse because each point will be passed by 3 paths. The whole space will be equally divided into 64 cubes.

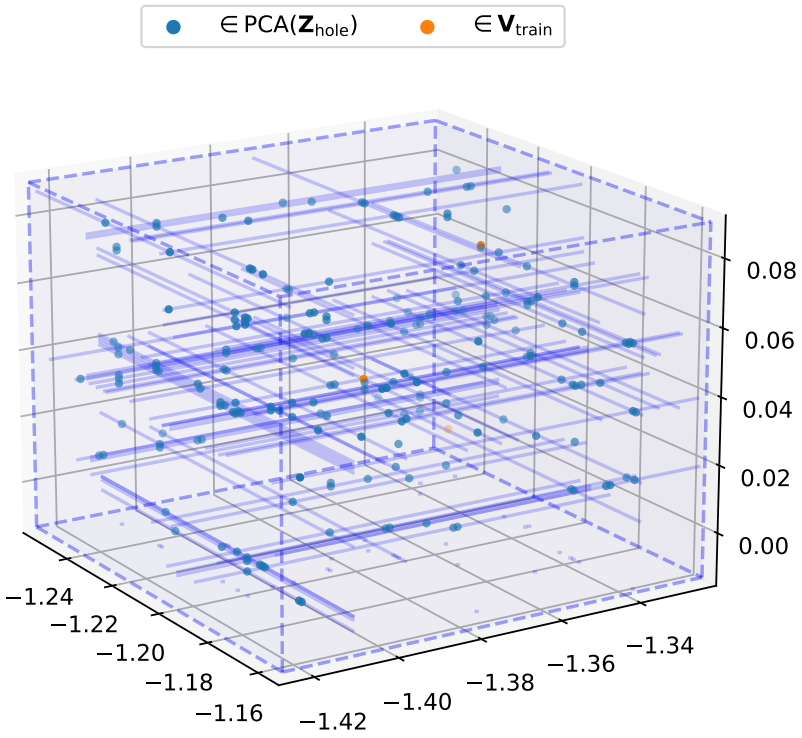


Figure 5.1: *A cubic fence C in the latent space of Vanilla-VAE trained on the Wiki dataset, with d_r set at 3 to facilitate visualisation (cf. § 5.4). C, whose 12 edges are illustrated by dashed lines, surrounds the dimensionally reduced expectation of three encoded training samples. Traversed paths are illustrated by the solid lines within the cube.*

5.3.1.1 Dimensionality Reduction

One problem for the current indicators is their limited searching capacity (as evidenced by their time complexity $\mathcal{O}(I^d)$) over the target space. Concretely speaking, both indicators rely on signalling latent holes through 1-dimensional traversal, but a latent space normally has dozens of dimensions to guarantee modelling capacity. To alleviate this issue, after feeding all training samples in $\mathbf{X}$ to the forward pass of a trained VAE and storing the encoded latent variables in $\mathbf{Z}$ (**Step 2**), we perform dimension reduction using Principal Component Analysis (PCA) (Jolliffe, 1986) and conduct a search in the resulting d_r-dimensional space instead of the original d-dimensinoal space (**Step 5**). We further save the mathematical expectation and standard deviation of the latent variables in $\mathbf{V}_{\text{train}}$ (**Step 3**) and $\mathbf{D}_{\text{train}}$ (**Step 4**), respectively.

Algorithm 1 TDC for latent hole identification

Input: Trained VAE model w/ a d-dimensional latent space; original training set $\mathbf{X}$; reduced dimension d_r; the desired number of detected vectors in latent holes N_{hole}
Output: $\mathbf{Z}_{\text{hole}}$

1: $\mathbf{Z} \leftarrow \emptyset$; $\mathbf{V'}_{\text{train}} \leftarrow \emptyset$; $\mathbf{D'}_{\text{train}} \leftarrow \emptyset$
2: $\forall \mathbf{x} \in \mathbf{X}$, $\mathbf{Z} \leftarrow \mathbf{Z} \cup \{\mathbf{z}\}$ *// $\mathbf{z}$ is the encoded $\mathbf{x}$*
3: $\forall \mathbf{z} \in \mathbf{Z}$, $\mathbf{V}_{\text{train}} \leftarrow \mathbf{V}_{\text{train}} \cup \{\mathbb{E}(\mathbf{z})\}$ *// $\mathbb{E}(\cdot)$ yields the expectation*
4: $\forall \mathbf{z} \in \mathbf{Z}$, $\mathbf{D}_{\text{train}} \leftarrow \mathbf{D}_{\text{train}} \cup \{\sigma(\mathbf{z})\}$ *// $\sigma(\cdot)$ yields the standard deviation*
5: $\mathbf{Z}' \leftarrow \text{PCA}(\mathbf{Z})$ *// Dimension reduced from d to d_r*
6: $C \leftarrow$ a randomly-picked closed cube which contains d_r vectors of $\mathbf{Z}'$, w/ edges parallel to d_r dimensions
7: $\mathbf{Z}_{\text{hole}} \leftarrow \emptyset$; $\mathbf{Z'}_{\text{hub}} \leftarrow \emptyset$; $\Pi \leftarrow \emptyset$
8: **while** $|\mathbf{Z}_{\text{hole}}| \leq N_{\text{hole}}$ **do**
9: **if** $\mathbf{Z'}_{\text{hub}} == \emptyset$ **then**
10: $\mathbf{Z'}_{\text{hub}} \leftarrow$ {a random point in C} *// Restart BFS*
11: **end if**
12: $\Pi \leftarrow$ unvisited line segments: passing through vectors in $\mathbf{Z'}_{\text{hub}}$ $\bigwedge$ parallel to one of the d_r dimensions $\bigwedge$ w/ endpoints on C *// Depth increases by 1*
13: $\mathbf{Z'}_{\text{hub}} \leftarrow \emptyset$
14: **for** each *path* (cf. § 5.2.1) in Π **do**
15: Sample $\mathbf{z'}_i$ on path at an interval of $0.01 * \min(\mathbf{D}_{\text{train}})$
16: $\forall i,\ \mathbf{z}_i \leftarrow \text{INVERSE\_PCA}(\mathbf{z'}_i)$
17: $\forall i$, decode $\mathbf{z}_i$ to compute $\mathcal{I}(i)$ w/ $\mathbf{V}_{\text{train}}$ and $\mathbf{D}_{\text{train}}$ *// Based on* Eq. (5.1) *in § 5.2.1*
18: **if** $\mathcal{I}(i)$ is an outlier **then**
19: $\mathbf{Z}_{\text{hole}} \leftarrow \mathbf{Z}_{\text{hole}} \cup \{\mathbf{z}_i\}$; $\mathbf{Z'}_{\text{hub}} \leftarrow \mathbf{Z'}_{\text{hub}} \cup \{\mathbf{z'}_i\}$
20: **end if**
21: **end for**
22: **end while**

In addition, instead of traversing unconstrained paths like past studies, we only visit latent vectors through paths parallel to the d_r dimensions (see **Step 12** and the next paragraph). Such a setup is based on the intuition that these top principal components contain more information about the latent space, and thus they are more likely to be useful when capturing latent holes.

5.3.1.2 Initialising Infrastructures for Search

To further boost efficiency, we propose to conduct a search on a tree-based structure within a pre-established cubic fence. To be more concrete, at **Step 6** we first locate a cube C which surrounds d_r encoded training samples from $\mathbf{Z}'$ (i.e., $\mathbf{Z}$ after dimension reduction).

These d_r posterior vectors serve as references when analysing the distribution of latent holes[3] (cf. § 5.4.2). We restrict the edges of the d_r-dimensional C to be parallel to the d_r latent dimensions and treat C as the range of our search. Next, we regard each sampled latent vector after dimension reduction $\mathbf{z}'_i$ as a node, and in order to expand the search regions rapidly, we need to visit these nodes following a BFS-based procedure (Skiena, 2008). Therefore, our algorithm maintains a set $\mathbf{Z}'_{\text{hub}}$ to keep track of all untraversed *hub nodes*, where the root (aka. the first hub node) is randomly initialised in C (**Step 10**). For each hub node, we define d_r orthogonal *paths*, each of which is a line segment that passes through the hub node and is parallel to one dimension. At **Step 12**, we log paths having not been previously processed in a set Π (see the next paragraph for more detail).

5.3.1.3 Identifying Latent Holes

Following the principle of BFS, the `TDC` algorithm processes all nodes at the same depth (i.e., all nodes on the paths in Π) before moving to the next depth. On each path, following Falorsi et al. (2018) and Xu et al. (2020), at **Step 15** we sequentially sample a series of $\mathbf{z}'_i$. To ensure the sampling is fine-grained, we set the interpolation interval at the 0.01 times minimum standard deviation of all elements in $\mathbf{D}_{\text{train}}$ (see **Step 4**). After that, we utilise the inverse transformation of PCA[4] to reconstruct $\mathbf{z}'_i$ to the original d-dimensional latent space at **Step 16** and generate output samples through the decoder at **Step 17**. One core question raised is how to choose the indicator $\mathcal{I}$ between the two existing ones which seem quite distinct (cf. § 5.2.1). We eventually select the scheme of Falorsi et al. (2018) (i.e., $\mathcal{I}_{\text{Lipschitz}}$ in Eq. (5.1)) and further adopt the Wassertein distance as the metric for the sample space. Detailed justifications are provided in § 5.3.2 and § 5.3.3, respectively. After all paths in Π are investigated, our algorithm pushes the tree search to its next depth by reloading the emptied $\mathbf{Z}'_{\text{hub}}$ with newly identified latent variables in the holes (**Step 19**). The motivation for treating them as new hub nodes comes from our observation that holes tend to gather as

[3]We select d_r as the number of contained $\mathbf{z}' \in \mathbf{Z}'$, aiming to avoid cherry-picking and deredundancy hyper-parameters.

[4]https://tinyurl.com/inverse-pca

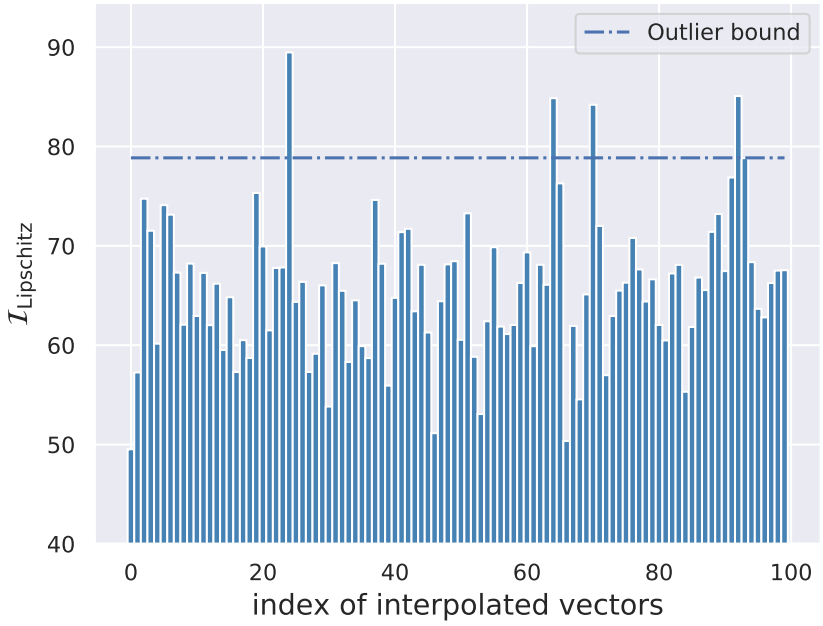


Figure 5.2: *$\mathcal{I}_{\text{Lipschitz}}$ of traversed vectors on one latent path of Vanilla-VAE trained on the Yahoo dataset.*

clusters. Figure. 5.2 exhibits one observation where multiple outlier $\mathcal{I}_{\text{Lipschitz}}$ are identified after visiting just 100 latent vectors on a path. Such example confirms the motivation of the TDC algorithm, i.e., latent holes often gather in small regions and the principal components tend to pass through them. In case that no hub node is added, which suggests the end of current BFS, TDC will bootstrap another tree by randomly picking a new root. The algorithm halts when more than N_{hole} holes are identified.

In practice, we find that our tree-search strategy with dimension reduction not only boosts the efficiency from an algorithmic perspective, but is also highly parallelisable by nature[5] and thus can duce computational time. In theory, the time complexity of TDC can be reduced to $\mathcal{O}({I_r}^{d_r})$, where d_r can be as small as 3 (cf. § 5.4) and I_r is typically less than 2, thanks to the parallelism of our algorithm. In experiments, when the device is equipped with a Nvidia GTX Titan-X GPU and a Intel i9-9900K CPU, in most cases TDC (with d_r at 8) can return more

[5]Our implementation parallelises the processing of different paths at the same BFS depth and different $\mathbf{z}'$ on the same path.

than 200 holes in less than 5 minutes, whereas the methods of Falorsi et al. (2018) and Xu et al. (2020) often need at least 30 minutes to find a hole in the same setup as our `TDC`.

5.3.2 Picking Indicator for `TDC`

Obviously, the indicator used by `TDC` (**Step 17** in Algorithm 1) plays a crucial role as it directly affects the effectiveness of identifying latent holes. By analysing the two existing indicators in § 5.2.1, we demonstrate that (1) although developed under different intuitions, they can actually be unified within a common framework; (2) although both indicators have been tested successfully in validating the presence of latent holes, the indicator of Falorsi et al. (2018) ($\mathcal{I}_{\text{Lipschitz}}$) is more accurate as it has better completeness and is thus more suitable to our algorithm. To begin with, we prove the following lemma:

Lemma 1. $\text{NLL}(\mathbf{x}, P)$*, the* NLL *of a data point* $\mathbf{x}$ *under a multivariate normal distribution with independent dimensions* $P = \mathcal{N}(\mu, \mathbf{K}_P)$ *(with* μ *as the mean and* $\mathbf{K}_P$ *as the covariance matrix) can be* ***numerically*** *equal to a distance measure* $\mathbb{D}_{\text{NLL}}$ *as*

$$\mathbb{D}_{\text{NLL}}(\mathbf{x}, \mu) = \frac{1}{2}\mathbb{D}_{\text{G}}(\mathbf{x}, \mu) + \delta(\mathbf{K}_P), \tag{5.3}$$

where $\mathbb{D}_{\text{G}}$ *is the so-called Generalized Squared Interpoint Distance (Gnanadesikan and Kettenring, 1972) and* $\delta(\cdot)$ *is a single value function.*

Remark 1. $\mathbb{D}_{\text{G}}$ *is the squared value of the Mahalanobis distance (Mahalanobis, 1936) and measures dissimilarity between two random vectors of the same distribution.*

Proof. The probability density of P at observation $\mathbf{x}$ can be computed as Prince (2012)

$$\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^\intercal \mathbf{K}_P^{-1}(\mathbf{x}-\mu)\right) / \left((2\pi)^{\frac{d}{2}}|\mathbf{K}_P|^{\frac{1}{2}}\right). \tag{5.4}$$

Therefore, the NLL of $\mathbf{x}$ under P becomes

$$\mathrm{NLL}(\mathbf{x}, P) = \frac{1}{2}\left[(\mathbf{x}-\mu)^{\intercal}\mathbf{K}_P^{-1}(\mathbf{x}-\mu) + \log(|\mathbf{K}_P|) + \log(2\pi)d\right]. \tag{5.5}$$

Additionally, by defining function $\delta(\cdot)$ as

$$\delta(\cdot) := \frac{1}{2}\left[\log(|\cdot|) + \log(2\pi)d\right], \tag{5.6}$$

we can see that

$$\mathrm{NLL}(\mathbf{x}, P) = \frac{1}{2}\left[(\mathbf{x}-\mu)^{\intercal}\mathbf{K}_P^{-1}(\mathbf{x}-\mu)\right] + \delta(\mathbf{K}_P). \tag{5.7}$$

As $\mathbb{D}_{\mathrm{G}}$ between $\mathbf{x}$ and μ is written as

$$\mathbb{D}_{\mathrm{G}}(\mathbf{x}, \mu) = (\mathbf{x}-\mu)^{\intercal}\mathbf{K}_P^{-1}(\mathbf{x}-\mu), \tag{5.8}$$

By substituting Eq. (5.8) into Eq. (5.7) we have

$$\mathrm{NLL}(\mathbf{x}, P) = \frac{1}{2}\mathbb{D}_{\mathrm{G}}(\mathbf{x}, \mu) + \delta(\mathbf{K}_P) = \mathbb{D}_{\mathrm{NLL}}(\mathbf{x}, \mu). \tag{5.9}$$

□

Based on this lemma, we find Eq. (5.2) is **numerically** equivalent to directly calculating $\mathbb{D}_{\mathrm{NLL}}(\mathbf{z}_i, \mu^{(t)})$, yielding

$$\mathcal{I}_{\text{Aggregation}}(i) = \sum_{t=1}^{M} \left[\frac{1}{2} \mathbb{D}_{\text{G}}(\mathbf{z}_i, \mu^{(t)}) + \delta(\mathbf{K}_{\mathbf{Z}^{(t)}}) \right] / M, \tag{5.10}$$

where $\mu^{(t)}$ and $\mathbf{K}_{\mathbf{Z}^{(t)}}$ are the mean and covariance matrix of posterior $\mathbf{Z}^{(t)}$, respectively. Note that as $\mathbf{Z}^{(t)}$ is deterministic, $\delta(\mathbf{K}_{\mathbf{Z}^{(t)}})$ settles as a constant term. By specifying $\mathbb{D}_{\text{sample}}$ as $\mathbb{D}_{\text{NLL}}$, *w.l.o.g.*, we theoretically prove that *if a latent position is signalled to be discontinuous by the indicator of Xu et al. (2020), it will be identified using that of Falorsi et al. (2018).*

Proof. For a latent position $\mathbf{z}_i$, if it is classified as *continuous* with a continuous neighbour $\mathbf{z}_{i+1}$ (i.e., based on $\mathcal{I}_{\text{Lipschitz}}(i+1)$ and the outlier criterion as discussed in § 5.2.1), we know that the indicator $\mathcal{I}_{\text{Lipschitz}}(i+1)$ is not a large outlier and thus is bounded (considering the original formalisation of Lipschitz continuity).

As mentioned, we implement $\mathbb{D}_{\text{space}}$ in Eq. (5.1) with $\mathbb{D}_{\text{NLL}}$ in the proofed lemma, yielding

$$\mathbb{D}_{\text{NLL}}(\mathbf{x'}_i, \mathbf{x'}_{i+1}) / \mathbb{D}_{\text{latent}}(\mathbf{z}_i, \mathbf{z}_{i+1}) < \lambda_{\text{Lipschitz}}, \tag{5.11}$$

where $\lambda_{\text{Lipschitz}}$ is a pre-defined threshold (e.g., Falorsi et al. (2018) set $\lambda = 10$). Note that $\mathbb{D}_{\text{latent}}(\mathbf{z}_i, \mathbf{z}_{i+1})$ is now a constant term because the positions of $\mathbf{z}_i$ and $\mathbf{z}_{i+1}$ are determinate. Similarly, as its neighbour $\mathbf{z}_{i+1}$ is continuous as given, we have $\mathcal{I}_{\text{Aggregation}}(i+1)$ is bounded and thus there exists a threshold $\lambda_{\text{Aggregation}}$, such that

$$\begin{aligned} \mathcal{I}_{\text{Aggregation}}(i+1) &= \sum_{t=1}^{M} \left[\frac{1}{2} \mathbb{D}_{\text{G}}(\mathbf{z}_{i+1}, \mu^{(t)}) + \delta(\mathbf{K}_{\mathbf{Z}^{(t)}}) \right] / M \\ &= \sum_{t=1}^{M} \mathbb{D}_{\text{NLL}}(\mathbf{z}_{i+1}, \mu^{(t)}) / M \\ &< \lambda_{\text{Aggregation}} - \lambda_{\text{Lipschitz}}\, \mathbb{D}_{\text{latent}}(\mathbf{z}_i, \mathbf{z}_{i+1}) < \lambda_{\text{Aggregation}}. \end{aligned} \tag{5.12}$$

where there must exist a larger upper bound (i.e., the threshold $\lambda_{\text{Aggregation}}$) and a smaller

one (i.e., $\lambda_{\text{Aggregation}} - \lambda_{\text{Lipschitz}}\, \mathbb{D}_{\text{latent}}(\mathbf{z}_i, \mathbf{z}_{i+1})$). Note that both of $\lambda_{\text{Lipschitz}}$ and $\mathbb{D}_{\text{latent}}(\mathbf{z}_i, \mathbf{z}_{i+1})$ are constant terms mentioned above.

By definition, the Triangle Inequality always holds for established metrics such as $\mathbb{D}_{\text{G}}$. Therefore, taking $\mathbf{z}_{i+1}$ as an anchor point we can show that

$$\begin{aligned}\text{Eq. (5.10)} &\leq \sum_{t=1}^{M} \left[\frac{1}{2}\big(\mathbb{D}_{\text{G}}(\mathbf{z}_i, \mathbf{z}_{i+1}) + \mathbb{D}_{\text{G}}(\mathbf{z}_{i+1}, \mu^{(t)})\big) + \delta(\mathbf{K}_{\mathbf{Z}^{(t)}})\right] /M \\ &< \sum_{t=1}^{M} \mathbb{D}_{\text{NLL}}(\mathbf{z}_{i+1}, \mu^{(t)})/M + \sum_{t=1}^{M} \mathbb{D}_{\text{NLL}}(\mathbf{z}_i, \mathbf{z}_{i+1})/M. \end{aligned} \tag{5.13}$$

Further incorporating Eq. (5.13) with Eq. (5.11) and Eq. (5.12) finally yields

$$\begin{aligned}\mathcal{I}_{\text{Aggregation}}(i) &< \lambda_{\text{Aggregation}} - \lambda_{\text{Lipschitz}}\, \mathbb{D}_{\text{latent}}(\mathbf{z}_i, \mathbf{z}_{i+1}) + \sum_{t=1}^{M} \lambda_{\text{Lipschitz}}\, \mathbb{D}_{\text{latent}}(\mathbf{z}_i, \mathbf{z}_{i+1})/M \\ &= \lambda_{\text{Aggregation}} - \lambda_{\text{Lipschitz}}\, \mathbb{D}_{\text{latent}}(\mathbf{z}_i, \mathbf{z}_{i+1}) + \lambda_{\text{Lipschitz}}\, \mathbb{D}_{\text{latent}}(\mathbf{z}_i, \mathbf{z}_{i+1}) \\ &= \lambda_{\text{Aggregation}}, \end{aligned} \tag{5.14}$$

which suggests a fixed upper bound for $\mathcal{I}_{\text{Aggregation}}(i)$. Therefore, $\mathbf{v}_i$ is continuous under the criterion of Xu et al. (2020). This demonstrates that $\forall$ *latent positions, if they are not identified as in holes under the criterion of Xu et al. (2020), they will not be identified as in holes under the criterion of Falorsi et al. (2018).* □

Apart from theoretical proof, empirically we also observe cases showing $\mathcal{I}_{\text{Lipschitz}}$ has better completeness than $\mathcal{I}_{\text{Aggregation}}$. As illustrated by Figure. 5.3, $\mathbf{z}_4$ is in a discontinuous latent region as its corresponding $\mathbf{x}'_4$ greatly departs from the samples of other latent vectors on the same path. However, when $\mathbf{x}'_4$ and $\{\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3, \mathbf{x}'_5\}$ are roughly symmetric to the posteriors ($\sim$ normal distributions with same standard deviation) of $M = 4$ test samples, $\mathcal{I}_{\text{Aggregation}}(4)$ is not a large outlier and the hole may thus be ignored. However, this hole can be identified using the other indicator as $\mathcal{I}_{\text{Lipschitz}}(4)$ makes a large outlier in this scenario. To

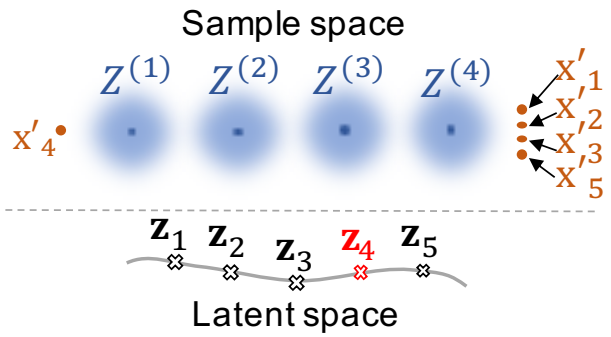


Figure 5.3: *A toy example where* $\mathbf{z}_4$ *is in a latent hole but may be falsely ignored by* $\mathcal{I}_{\text{Aggregation}}$.

conclude, $\mathcal{I}_{\text{Lipschitz}}$ should be adopted to reduce the false-negative rate of `TDC`.

5.3.3 Picking Sample Space Metric

We find that it is impossible to directly apply the indicator of Falorsi et al. (2018) ($\mathcal{I}_{\text{Lipschitz}}$) for VAEs for NLP tasks: the Euclidean distance is used as $\mathbb{D}_{\text{sample}}$ in the original study which is on vision VAEs, but it cannot be used to measure the distance between sentences[6]. One straightforward solution is to directly follow Xu et al. (2020) who select NLL, a long-standing and popular metric in past VAE studies on NLP tasks (Bowman et al., 2016; Fu et al., 2019; Zhu et al., 2020). However, it does not make a valid metric for the decoder of VAEs for language generation. To be more concrete, while on the encoder side NLL can be calculated as $q_\phi(\mathbf{z}|\mathbf{x})$ in Eq. (4.3) (Xu et al., 2020) and is thus normal and thus has a metric-based numerical equivalent $\mathbb{D}_{\text{NLL}}$ (cf. the proofed lemma in § 5.2.1), on the decoder side the posterior distribution of a output sentence is generally computed by a `logsoftmax` layer in practice and is thus *no longer* normal. Instead, coupling the `logsoftmax` layer with NLL yields cross-entropy[7] as

[6] In principle, by simply adopting metrics such as Euclidean distance, `TDC` can also be applied on VAEs for image generation. We will explore this direction in the future.

[7] https://tinyurl.com/CE-LSM-NLL

$$\mathbb{H}(P, Q) := \mathbb{H}(P) + \mathbb{D}_{\mathrm{KL}}(P||Q), \tag{5.15}$$

where P and Q are two probability distributions and $\mathbb{H}(P)$ is the entropy of P. It is obvious that $\mathbb{H}(P, Q)$ does not qualify as a statistical metric, because it does not satisfy symmetry nor Triangle Inequality. A workaround which adopts the symmetric cross entropy (Wang et al., 2019) and replaces KL-divergence with the positive squared root of its smoothed version, JS-divergence, can somehow alleviate the issues (Osán et al., 2018). Nonetheless, the resulting formula may dramatically lose its measurement capacity when there is no overlap between P and Q (Lin, 1991) (which is common when testing a VAE for language generation) and is thus unsuitable neither.

Finally, we refer to the Wasserstein distance of finite first moment as our final candidate, which is defined as

$$\mathbb{D}_{\mathrm{W1}}(\nu_P, \nu_Q) := \inf_{\Gamma \in \mathcal{P}(P \sim \nu_P, Q \sim \nu_Q)} \mathbb{E}_{(P,Q)\sim\Gamma}||P, Q||_1, \tag{5.16}$$

where $\mathcal{P}(P \sim \nu_P, Q \sim \nu_Q)$ is a set of all joint distributions of (P, Q) with marginals ν_P and ν_Q, respectively. $\mathbb{D}_{\mathrm{W1}}$ has been adopted in a large body of recent VAE studies, such as Chewi et al. (2021); Tolstikhin et al. (2018); Wu et al. (2019). Moreover, to further enhance efficiency, following Patrini et al. (2020), we select the lightspeed Sinkhorn algorithm (Cuturi, 2013) to compute $\mathbb{D}_{\mathrm{W1}}$.

5.4 Empirical Studies

In this section, we describe our experiment for validating the effectiveness of the proposed `TDC` algorithm for latent hole identification. We first describe our setup, followed by three empirical studies investigating the impact of latent holes on text generation, the vacancy of

	Description	Training	Validation
Yelp15	Restaurant reviews	100K	10K
Yahoo	Web Q&A on daily knowledge	100K	10K
SNLI	Constructed based on a real-world image captioning dataset	100K	10K
Wiki	Wikipedia articles	1.13M	141K

Table 5.1: *Corpora descriptions and statistics.*

holes, and how the holes are distributed.

5.4.1 Experimental Setup

5.4.1.1 Models

To demonstrate the generalisability of our proposed `TDC` algorithm, we pretrain five strong and representative VAE models for language generation, including the state-of-the-art iVAE$_{\text{MI}}$ model:

Vanilla-VAE (Bowman et al., 2016), which uses LSTM and KL annealing for mitigating the posterior collapse issue;

β-VAE (Higgins et al., 2017a), which utilises an adjustable β to balance the reconstruction loss and the KL term;

Cyc-VAE (Fu et al., 2019), which employs cyclical annealing for the KL term;

iVAE$_{\text{MI}}$ (Fang et al., 2019), which replaces the Gaussian-based posteriors with the sample-based distributions;

BN-VAE (Zhu et al., 2020), which leverages the batch normalisation for the variational posterior's parameters.

5.4.1.2 Datasets

We consider four large-scale datasets, three of which have been commonly used in previous studies for testing VAEs on the language generation task: **Yelp15** (Yang et al., 2017), **Yahoo** (Zhang et al., 2015; Yang et al., 2017), and a downsampled version of **SNLI** (Bowman

et al., 2015; Li et al., 2019a). We additionally constructed a dataset (called **Wiki**) by downloading the latest English Wikipedia articles[8] and then randomly sampling 1% sentences from the whole set. The size of **Wiki** is 10 times larger than other datasets and it contains more training samples which can cover more areas of the latent space during training VAEs. The statistics of four datasets are shown in Table 5.1. For **Yahoo**, **Yelp15** and **SNLI**, their training and validation sets are all 100K and 10K, respectively. For **Wiki**, the training and validation sets are 1.13M and 141K, respectively.

5.4.1.3 Hyper-parameter Settings

We adopt the official code of each tested models and apply the same pretraining hyper-parameters to all models. To be concrete, the encoders and decoders of all models are constructed using one-layer LSTM with 1024 hidden units and 512D word embeddings. The dimension of the latent space is 32. KL annealing (Bowman et al., 2016) is applied to all models, and the scalar weight of the KL term linearly increases from 0 to 1 during the first 10 epochs. Dropout layers with a probability 0.5 are installed on the encoder's both input-to-hidden and hidden-to-output layers. All baselines are trained with Adam optimiser with an initial learning rate of 8e-4. Parameters of all models are initialised using a uniform distribution $U(-0.01, 0.01)$ except for word embeddings with $U(-0.1, 0.1)$. The gradients are clipped at 5.0. During training, we set patience at 5 epochs, and adopt early stopping based on Perplexity (PPL) with standard validation splits. For β-VAE and BN-VAE, the corresponding β and γ are set at 0.4 and 0.7, respectively.

5.4.1.4 Configurations of the `TDC` Algorithm

As discussed earlier, the dimensions of the original latent space d is 32. When performing dimension reduction, we experiment with $d_r = \{3, 4, 8\}$ for all setups. Empirically, we observe that results for different d_r setting show very similar trends. We report the results based on $d_r = 8$ in the main body and provide the results for other settings in Appendices A.1,

[8] https://tinyurl.com/LatestWiki

A.3, and A.4. When computing our hole indicator (Eq. (5.1)), we follow Falorsi et al. (2018) and adopt the Euclidean distance for $\mathbb{D}_{\text{latent}}$ (NB: for sample space ($\mathbb{D}_{\text{sample}}$) we adopt the Wasserstein distance as discussed in § 5.3.3). Following Hoaglin et al. (1986), at **Step 18** of `TDC` we adopt the popular Inter-Quartile Range measure that defines large outliers as data points falling above $\text{Q3} + 1.5 \cdot (\text{Q3} - \text{Q1})$, where Q1 and Q3 respectively denote the lower and upper quartile. In all runs, we set $N_{\text{hole}} = 200$, i.e., the program halts when more than 200 holes are identified and we store the first 200 holes in $\mathbf{Z}_{\text{hole}}$ for evaluation. For stochastic analysis, we run `TDC` 50 times for each setup, yielding $50 \times 200 = 10$K latent holes per setup. Recalling that there are 5 models and 4 datasets, we totally have 20 setups.

5.4.2 Results and Analysis

5.4.2.1 Impact of Latent Holes on Text Generation

In this experiment, we investigate how latent holes impact VAE models' performance on text generation. To our knowledge, this is the first such study as prior works (Falorsi et al., 2018; Xu et al., 2020) merely explored the existence of holes and their schemes are incapable to discover a sufficient amount of holes for quantitative analysis due to algorithm inefficiency (cf. § 5.3).

Our analysis is established on the correlation between models' performance on text generation and the density of latent holes. As discussed in § 5.4.1, we identify 10K holes for each setup using our `TDC` algorithm, based on which 10K sentences were decoded. We then calculate the average PPL of those 10K sentences using a pre-trained GPT model (Radford et al., 2018) following the practice of Dathathri et al. (2019)[9]. As for the density estimation of latent holes, we utilise the average number of paths traversed before the number of identified holes reaches the algorithm halting threshold $N_{\text{hole}} = 200$. Intuitively, the fewer paths visited, the denser the holes are distributed, and vice versa.

Figure. 5.4 shows the average PPL versus the number of paths traversed (when reaching 200 identified holes) for each setup. It can be observed that there is a strong negative

[9]https://git.io/HF-gpt

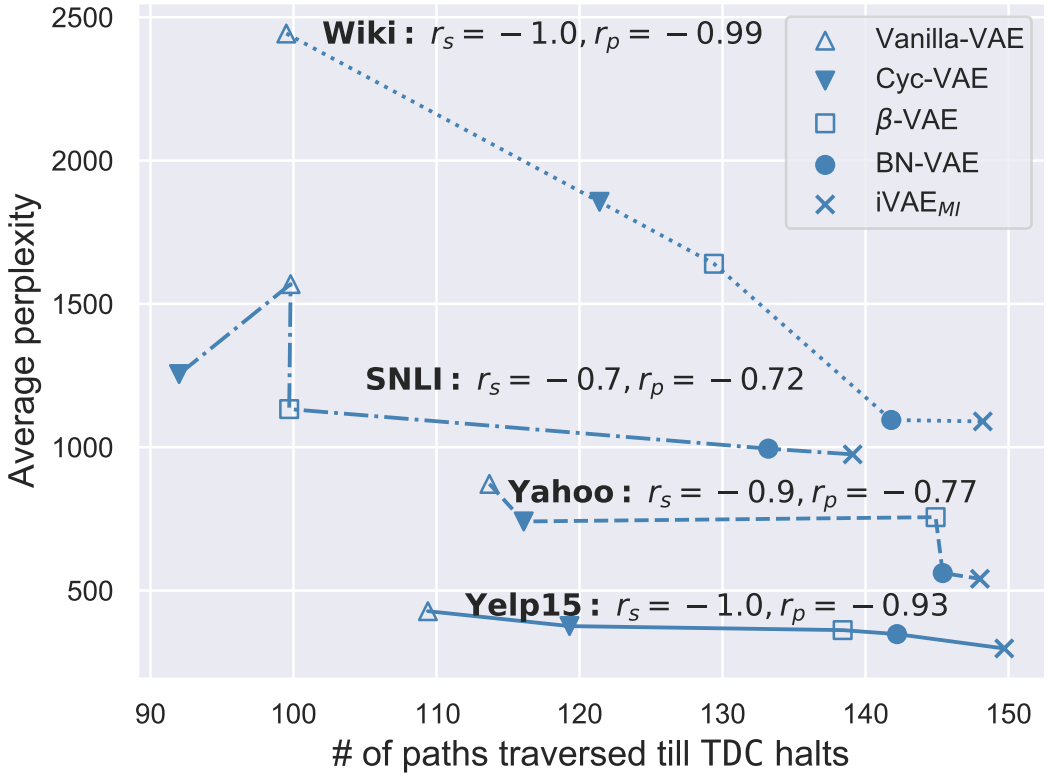


Figure 5.4: *Average PPL and the number of paths traversed until more than N_{hole} holes are identified. Correlation coefficients r_s and r_p are marked corpus-wise.*

correlation between the average PPL (lower the better) and the number of visited paths, where the corpus-wise Spearman's correlation coefficient r_s is consistently below or equal to -0.70. It can also be observed that the Person's correlation coefficient r_p is below -0.72 for all datasets, showing a certain degree of linearity for the correlation. In summary, the above observations verify the intuition that denser latent hole distribution leads to higher average PPL, and hence worse performance of VAEs for text generation.

As shown in Table. 5.2, for all cubes with different dimension in all datasets, iVAE$_{\text{MI}}$ needs to search much more paths and depths than other models to reach the halt condition, and it performs best. On the contrary, the overall worst-performing model, Vanilla-VAE, covers the fewest paths and depths. In addition, when d_r increases, we find that the quantity of traversed path gradually increases but the quantity of reached depths decreases, indicating

Model	Yelp15		Yahoo		SNLI		Wiki	
	path	depth	path	depth	path	depth	path	depth
VAE	109.4	3.1	113.7	3.4	99.8	4.2	99.5	2.8
Cyc-VAE	119.3	3.7	116.1	3.5	92.0	3.7	121.4	3.4
β-VAE	138.4	5.0	144.9	8.9	99.7	3.1	129.4	4.8
BN-VAE	142.2	5.6	145.4	6.6	133.2	6.5	141.8	5.8
iVAE_{MI}	149.7	7.4	148.0	11.4	139.1	14.7	148.2	6.4

Table 5.2: *Average quantities of traversed paths and reached depths in each C of 8D until 200 latent holes are identified.*

that the distribution of holes is denser in a lower-dimensional cube. By comparing results across different datasets, the distribution of holes is denser in Wiki dataset for VAEs, which agrees with our finding in Figure. 5.4 (see similar results in Appendix A.1).

Corpus-wise, we notice that models trained on the Wiki dataset, i.e., our largest training dataset, do not seem to yield improvement for hole reduction when comparing to the much smaller datasets such as Yelp15. Furthermore, sentences decoded by models trained on Wiki have lower quality than those decoded by the corresponding models trained on Yelp15 and Yahoo. One plausible explanation is that the complexity (e.g., topic coverage) of datasets plays a more important role than the corpus size when training VAEs for language generation. For instance, while SNLI contains the same number of sentences as Yelp15 and Yahoo, models trained on SNLI are substantially inferior to the models trained on the other two datasets in terms of average PPL. Manually examining the datasets reveals that the topics covered topics in Yelp and Yahoo datasets are less diverse than that of SNLI and Wiki, e.g., SNLI was constructed based on Flickr30k (Young et al., 2014), which includes captions for real-world images across a wide range of categories.

5.4.2.2 Probing the Vacancy of Latent Holes

The previous experiment empirically shows that latent holes indeed have a detrimental effect on VAEs's generation performance. A recent study (Xu et al., 2020) proposed the so-called Latent Vacancy Hypothesis, assuming holes are *vacant* with no meaningful information encoded. This motivates us to further probe the vacancy of latent holes. Specifically, we

		Vanilla-VAE	Cyc-VAE	β-VAE	BN-VAE	iVAE$_{\text{MI}}$
Yelp15	H	0.428	0.376	0.362	0.348	0.298
	N	0.386†	0.339 †	0.356	0.303†	0.294
	R	18.241‡	18.293‡	18.349‡	18.234‡	18.211‡
Yahoo	H	0.872	0.741	0.756	0.561	0.541
	N	0.831†	0.704†	0.710†	0.527†	0.519†
	R	18.736‡	18.576‡	19.027‡	20.343‡	18.556‡
SNLI	H	1.569	1.255	1.133	0.995	0.975
	N	1.529†	1.129†	1.068†	0.947†	0.911†
	R	41.247‡	41.026‡	40.781‡	40.774‡	40.692‡
Wiki	H	2.443	1.856	1.640	1.095	1.090
	N	2.357†	1.721†	1.587†	1.041†	1.039†
	R	5.377‡	5.354‡	5.338‡	5.347‡	5.320‡

Table 5.3: *Average PPL (divided by 1K) of sentences decoded via vectors of* HOLE *(H),* NORM *(N) , and* RAND *(R) in all setups.* † *indicates the PPL of a model via N significantly lower than via H (with* $p < .05$*);* ‡ *indicates the the PPL of a model via R significantly larger than via H (with* $p < .005$*).*

conduct analysis by comparing the sentences decoded by latent vectors from an untrained decoder and by the hole vectors from a VAE decoder trained following the setup in § 5.4.1. For completeness, we also show the sentence decoded by normal (not in a hole) vectors from a trained VAE. We detailed these three different types of vectors below:

- **Hole vectors (HOLE)**, those being investigated in our previous experiments.
- **Normal vectors (NORM)**, sampled from the continuous regions near a hole, i.e., $\mathbf{z}_{i+1}$ is a normal vector if $\mathbf{z}_i$ is identified to be in a hole in $\mathcal{I}_{\text{Lipschitz}}$.
- **Vectors from the latent space of an untrained VAE (RAND)**. For controlled analysis, we randomly initialised a VAE model and pick latent vectors whose coordinates are the *same* as those of HOLE vectors. As this VAE is untrained, its latent vectors should carry zero information by nature.

We compute the PPL of the sentences generated by the vectors of each of the above categories. As expected, results in Table. 5.3 show that the sentences decoded via HOLE vectors are significantly inferior to those via NORM vectors in almost all setups tested (two-tailed t-test with Bonferroni correction (Dror et al., 2018); $p < .05$). It can also be observed

Vanilla-VAE $\times$ *SNLI*	
HOLE	the bridge was an old gentleman .
NORM	a married couple is resting .
RAND	waling speedo ever vehicle birdhouse supports tahoe vacant commute
HOLE	a crowd smiles at people .
NORM	an old man plays with his dogs .
RAND	inspect rioting shivering entrance back-to-back seeker wheeling
iVAE$_{MI}$ $\times$ *Yahoo*	
HOLE	it 's _UNK to do it or you just put home sick in the a back .
NORM	i 'm thinking of buying the _UNK on the internet from pennsylvania .
RAND	drin ;-lrb- parker vastly san ripped fountain tais compared gratuit
HOLE	this is not a place of all or more specifically my life .
NORM	is that what you want to do when your _UNK exceeds ?
RAND	rr selves t-mobile sad nondescript up-sell dominos concern newly

Table 5.4: *Examples of sentences decoded via vectors of* HOLE*,* NORM*, and* RAND *from SNLI and Yahoo datasets.*

that sentences decoded via HOLE vectors are a lot better than the *random output* generated via the RAND vectors ($p < .005$). This observation suggests that the Latent Vacancy Hypothesis proposed by Xu et al. (2020) does not hold empirically, i.e., the regions containing HOLE vectors are not vacant, which do capture some information from the training corpus.

Finally, we qualitatively analyse some sentence examples generated by different types of vectors, as shown in Table 5.4 and Table 5.5. First, we observe that although topologically adjacent in the latent space, HOLE and NORM vectors are decoded into completely irrelevant sentences semantically, indicating that holes, due to severely harming the smoothness of latent continuity, do have a detrimental effect on model's generation quality. Second, it can be observed that the output sentences generated via RAND vectors are neither syntactically correct, nor making any sense semantically. In contrast, although sentences decoded via HOLE vectors tend to have problematic word matching and contain content which is against common sense, at least they still follow basic grammars in most cases, which once again verifies that HOLE vectors some useful information. Based on this finding, one implication of the future work is to introduce a novel regularisation term in the objective function and utilise

BN-VAE $\times$ *Yelp15*	
HOLE	it free vip and ate well some of the sushi options around to _UNK you in the guest !! there 's more wine that an awesome hot chocolate cake then fair grade .
NORM	so i tend to get some good red salsa when i go to the restaurant . i always get the turkey wings , cornbread , risotto . the fries are very good as well !
RAND	told 18th maintenance crappy awsome devoured confit mosh sorely expiration cinnamon compassion refused abroad perfectly cant hokkaido
HOLE	$ the dude working back was great . if your perfectly _UNK then try it there . a safe bet " with light fluffy slices and some new soul .
NORM	if you 're a regular , this is really a good place to go with your family . its vegetarian dishes , no more like shredded beef . what do you want : there is a lot of onions on the side , but the noodles are a bit
RAND	excelent styrofoam thighs extra scots roadside poof cart massaman meters miracles boneless cannon oxymoron spoiled maui retain 12.50 dating
β-*VAE* $\times$ *Wiki*	
HOLE	from the _UNK that 's considered religious adventures were evolutionary lived of definition .
NORM	the first section of the " _UNK " , in the late 14th century , relief efforts were accomplished .
RAND	eviction abbe cultural biannual highfield aqua 27.7 ieyasu slowed gretchen fb raping charadriiformesfamily cleaner municipal
HOLE	fully investing by means in kyiv and enough budget genetic compliance egypt .
NORM	that they had a girl to set up the system , it seems to be " _UNK " .
RAND	£3 albrecht rendell dubstep elland sinhalese pediments namely anxieties amrita nootka worked brownish tatars luxury analogues europe/africa

Table 5.5: *Examples of sentences decoded via vectors of* HOLE*,* NORM*, and* RAND *from Yelp15 and Wiki datasets.*

the detected latent holes to regularise the latent space. In addition, TDC is a plugin for other existing VAE models. During training, TDC can be regarded as a data augmentation approach to treat the detected latent holes as negative samples under contrastive learning framework.

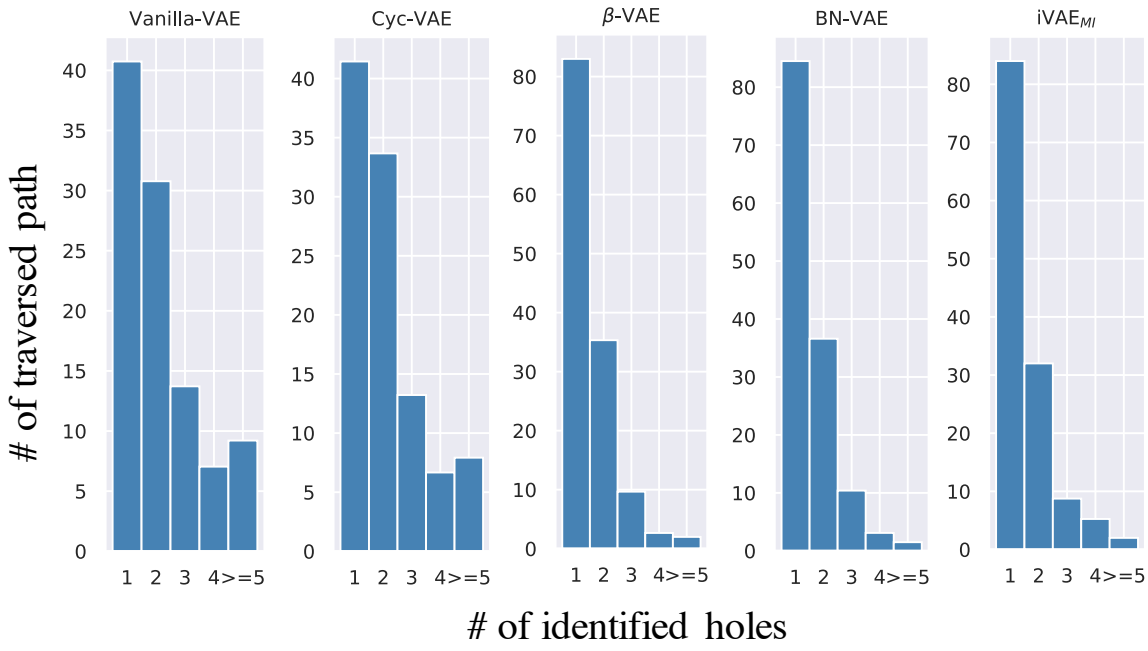


Figure 5.5: *Distribution of quantity of identified holes per latent path for models trained on the Wiki dataset when $d_r = 8$. Results for other datasets are in Appendix A.4.*

5.4.2.3 The Distribution of Latent Holes

Finally, we explore how the latent holes are distributed in the latent space. While a prior study (de Haan and Falorsi, 2018) proposed a theoretical hypothesis that latent holes should be densely distributed, it has never been investigated empirically.

We visualise one run of `TDC` in Figure. 5.1. As described in § 5.3.1, C is the *minimum* cube which can surround the 3 encoded training samples on a local latent region and thus spans quite narrowly (with a side length being around 0.1, while the width of the latent space is more than 5). However, even in this small search space, `TDC` still successfully halted and identified more than 200 (defined by N_{hole}, cf. § 5.3.1) latent holes, showing that the distance between these holes is tiny and their distribution is very dense. Moreover, all these latent holes are detected by traversing only 85 paths, meaning that more than 2 latent holes exist on each path, on average. Similar finding can be obtained in Figure. 5.5 (we further investigate the fine-grained quantity distribution of identified holes per latent path in Appendix A.4).

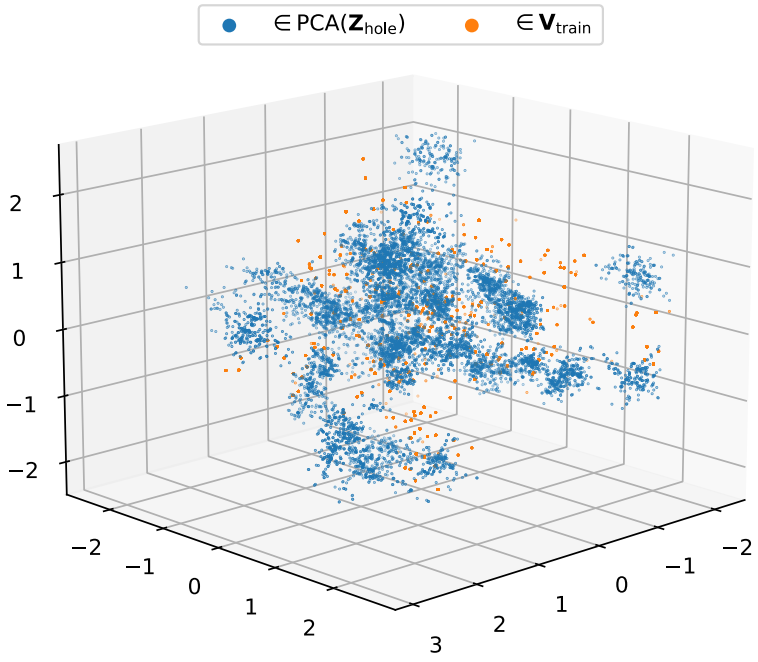


Figure 5.6: *The entire latent space of Vanilla-VAE trained on the Wiki dataset (please see Appendix A.2 for other setups). In total 50 runs of* `TDC` *are independently performed, each of which yields a cubic search space C like the one visualised in Figure. 5.1.*

Similar finding can be obtained in Figure. 5.6 that holes look ubiquitous in the entire latent space. In addition, once again we can see that in the 50 explored regions (the spaces which have been surrounded by C of each run of `TDC`), the identified latent holes are very close to each other and even form clusters.

5.5 Conclusion

In this chapter, we provide a comprehensive study on the discontinuities (aka. *holes*) in the latent space of VAEs, a phenomenon which has been shown to have a detrimental effect on model capacity. In contrast to existing works which exclusively focus on the encoder network and which merely explored the existence of holes, we propose a highly efficient tree-based decoder-centric (`TDC`) algorithm for latent hole identification. Comprehensive experiments on the language generation task show that the performance of text generation is strongly

correlated with the density of latent holes, that from the perspective of the decoder, the Latent Vacancy Hypothesis proposed by Xu et al. (2020) does not hold empirically; and that holes are ubiquitous and densely distributed in the latent space.

Chapter 6

Conclusions and Future Work

This thesis proposes various novel methods to tackle the dialogue act classification, the KL loss vanishing issue and the low-density latent region problem. This chapter will summarise the key contributions of each chapter and present possible directions of the future work.

6.1 Overview of Thesis

Chapter 3 proposes a dual-attention hierarchical recurrent neural network for dialogue act classification. Our model makes use of topic information in the dialogue conversation and utilises the multi-task learning strategy to combine dialogue act classification and topic information together. In addition, an automatic topic labelling scheme is introduced to avoid the time-consuming manual topic labelling. Our model is evaluated over three popular dialogue conversation datasets and compared with several strong baselines. Experimental results show that our model is able to yield better or comparable performance than baselines. In addition, there are two limitations of this work. At first, topic labels used in this model are only treated as discrete tokens, which has difficulties in capture topical semantic similarity in the corresponding label space. The topical word embedding might be a better choice. Second, although the classification accuracy of the dialogue act is improved under the multi-task learning mode, the classification accuracy on the topic side is not as high as the dialogue act

side.

Chapter 4 introduces a simple and robust VAE model, which imposes the KL regularisation into each timestep in the recurrent neural network encoder. Moreover, our method is generic and can be applied into any RNN-based VAE models in the text generation domain. By comparing with different strong VAE baselines in the language modelling task, our model can alleviate the KL loss vanishing well and generate high-quality sentences. In the dialogue response generation task, our model achieves better or comparable performance and can generate relevant, contentful and diverse responses. One limitation of this work is that the training convergence of the TWR-VAE is slow, especially in large-scale datasets. One possible reason might be that the timestep-wise KL regularisation always imposes small gradient values during optimisation, which further slows the training convergence down.

Chapter 5 provides a comprehensive research on the low-density latent region of the VAE in the text domain for the first time. We propose a highly efficient tree-based decoder algorithm to identify latent holes. In addition, comprehensive experiments show that the latent holes are densely distributed in the latent space and the latent hole is not really vacant. One limitation of this work is that we only tested our hypothesis and conducted the latent hole detection experiments in the Euclidean space. However, the latent space might be distorted via the mapping from non-trivial high-dimensional data manifolds to the learned low-dimensional latent space. Therefore, different space might need to be tested, e.g., Riemannian latent space.

6.2 Future Work

The current research work might be extended via many directions introduced below.

6.2.1 Topic Embeddings and Speaker's Information

As for the dialogue act classification work, our current model directly uses the topic labels for each conversation as a true labels for classification. However, there are several methods to impose the topic information into the model, such as encoding the topic labels as topic

embeddings and imposing them at character, word and utterance levels. In addition, the speaker's information is helpful for the dialogue act classification, because the implicit intention in each utterance varies based on different speaker's role. Apart from recognising dialogue acts, DA can help dialogue response generation as well, and combining DA and VAE for generation task is a meaningful direction.

6.2.2 Abstract Meaning Representation with Dialogue Acts

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is a type of sentence semantics representation which focuses on the main concepts (e.g., *phone*) and semantic relations (e.g., *ARG0*) and removes irrelevant syntactic information using a rooted directed acyclic graph. In addition, dialogue act, a representation of the speaker intent, has a strong connection with the main concepts and semantic relations in dialogue conversations. For example, a request for information (e.g., *How much is this phone?*) expresses distinct meaning compared to a request for action (e.g., *Could you please hand this phone to me?*). Constructing an AMR with dialogue act labels using graph neural network is a worthy direction to explore.

6.2.3 Pre-trained VAE Language Models

Recently, transformer-based pre-trained language models have shown its powerful performance in several downstream tasks, e.g., text classification, dialogue response generation, text summarisation, etc. However, variational pre-trained language models have not been well investigated yet. With the help of the transformer pretrained in large scale corpus, the latent space in the variational transformer can be organised better than RNN-based VAE, and different applications of the variational transformer is worth investigating.

6.2.4 The Impact of Latent Holes in Different Downstream Tasks

After investigating that latent holes are densely distributed in the latent space and they are not really vacant, an interesting direction is how the latent holes affect different NLP downstream

tasks, e.g., text summarisation, dialogue response generation, etc. In addition, a novel VAE model which avoids creating latent holes in the latent space is worth considering after identifying where the latent holes are.

6.2.5 Contrastive Learning using Detected Latent Holes

With the help of our highly efficient latent hole detection algorithm `TDC`, we find that the latent holes are actually not vacant, although there exist word matching and anti-common sense issues in the decoded sentences via the latent holes. However, one possible future work based on this finding is that we can treat these latent holes as negative samples and train a VAE model under the contrastive learning framework.

Bibliography

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

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Bhuiyan, M., Misra, A., Tripathy, S., Mahmud, J., and Akkiraju, R. (2018). Don't get Lost in Negation: An Effective Negation Handled Dialogue Acts Prediction Algorithm for Twitter Customer Service Conversations. In *arXiv preprint arXiv:1807.06107*.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

Bowman, S., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Bowman, S., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2016).

Generating sentences from a continuous space. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning (CoNLL)*.

Chen, Z., Yang, R., Zhao, Z., Cai, D., and He, X. (2018). Dialogue act recognition via crf-attentive structured network. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 225–234. ACM.

Chewi, S., Clancy, J., Gouic, T. L., Rigollet, P., Stepaniants, G., and Stromme, A. J. (2021). Fast and smooth interpolation on wasserstein space. In *International Conference on Artificial Intelligence and Statistics*, pages 3061–3069. PMLR.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103.

Cuturi, M. (2013). Sinkhorn distances: lightspeed computation of optimal transport. In *Advances in neural information processing systems*, volume 26, pages 2292–2300.

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. (2019). Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. (2018). Hyperspherical variational auto-encoders. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 856–865. Association For Uncertainty in Artificial Intelligence (AUAI).

de Haan, P. and Falorsi, L. (2018). Topological constraints on homeomorphic auto-encoding. *arXiv preprint arXiv:1812.10783*.

Dielmann, A. and Renals, S. (2008). Recognition of dialogue acts in multiparty meetings using a switching DBN. *IEEE transactions on audio, speech, and language processing*, 16(7):1303–1314.

Dieng, A. B., Kim, Y., Rush, A. M., and Blei, D. M. (2019). Avoiding latent variable collapse with generative skip models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2397–2405.

Dror, R., Baumer, G., Shlomov, S., and Reichart, R. (2018). The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392.

Falorsi, L., de Haan, P., Davidson, T. R., De Cao, N., Weiler, M., Forré, P., and Cohen, T. S. (2018). Explorations in homeomorphic variational auto-encoding. In *arXiv preprint arXiv:1807.04689*.

Fang, L., Li, C., Gao, J., Dong, W., and Chen, C. (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.

Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., and Carin, L. (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.

Gnanadesikan, R. and Kettenring, J. R. (1972). Robust estimates, residuals, and outlier detection with multiresponse data. *Biometrics*, pages 81–124.

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

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Gu, X., Cho, K., Ha, J. W., and Kim, S. (2019). Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. In *International Conference on Learning Representations*.

Gururangan, S., Dang, T., Card, D., and Smith, N. A. (2019). Variational pretraining for semi-supervised text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5880–5894.

He, J., Spokoyny, D., Neubig, G., and Berg-Kirkpatrick, T. (2019). Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations*.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017a). beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conferenceon Learning Representations*.

Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. (2017b). 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.

Hoaglin, D. C., Iglewicz, B., and Tukey, J. W. (1986). Performance of some resistant rules for outlier labeling. *Journal of the American Statistical Association*.

Huang, H., He, R., Sun, Z., Tan, T., et al. (2018). Introvae: Introspective variational autoencoders for photographic image synthesis. In *Advances in Neural Information Processing Systems*, pages 52–63.

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

Ji, Y., Haffari, G., and Eisenstein, J. (2016). A latent variable recurrent neural network for discourse-driven language models. In *Proceedings of the 2016 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 332–342.

Jolliffe, I. T. (1986). *Principal Component Analysis*. Springer.

Jurafsky, D. (1997). Switchboard swbd-damsl shallow-discourse-function annotation coders manual. *Institute of Cognitive Science Technical Report*.

Kalatzis, D., Eklund, D., Arvanitidis, G., and Hauberg, S. (2020). Variational autoencoders with riemannian brownian motion priors. In *International Conference on Machine Learning*, pages 5053–5066. PMLR.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126.

Khanpour, H., Guntakandla, N., and Nielsen, R. (2016). Dialogue act classification in domain-independent conversations using a deep recurrent neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2012–2021.

Kim, S. N., Cavedon, L., and Baldwin, T. (2010). Classifying dialogue acts in one-on-one live chats. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 862–871. Association for Computational Linguistics.

Kim, Y., Wiseman, S., Miller, A., Sontag, D., and Rush, A. (2018). Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, pages 2683–2692.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751.

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

Kumar, H., Agarwal, A., Dasgupta, R., and Joshi, S. (2018). Dialogue act sequence labeling using hierarchical encoder with crf. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Lee, J. Y. and Dernoncourt, F. (2016). Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–520.

Li, B., He, J., Neubig, G., Berg-Kirkpatrick, T., and Yang, Y. (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.

Li, C., Gao, X., Li, Y., Peng, B., Li, X., Zhang, Y., and Gao, J. (2020a). Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (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.

Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., and Jurafsky, D. (2017a). Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.

Li, R., Li, X., Chen, G., and Lin, C. (2020b). Improving variational autoencoder for text modelling with timestep-wise regularisation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2381–2397.

Li, R., Li, X., Lin, C., Collinson, M., and Mao, R. (2019b). A stable variational autoencoder for text modelling. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 594–599.

Li, R., Lin, C., Collinson, M., Li, X., and Chen, G. (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.

Li, Y., Su, H., Shen, X., Li, W., Cao, Z., and Niu, S. (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.

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

Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE TIT*.

Litman, D. J. and Allen, J. F. (1987). A plan recognition model for subdialogues in conversations. *Cognitive science*, 11(2):163–200.

Liu, C.-W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., and Pineau, J. (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.

Liu, Y. (2006). Using SVM and error-correcting codes for multiclass dialog act classification in meeting corpus. In *Ninth International Conference on Spoken Language Processing*.

Liu, Y., Han, K., Tan, Z., and Lei, Y. (2017). Using Context Information for Dialog Act Classification in DNN Framework. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2170–2178.

Ma, X. and Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. National Institute of Science of India.

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

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Osán, T. M., Bussandri, D. G., and Lamberti, P. W. (2018). Monoparametric family of metrics derived from classical jensen–shannon divergence. *Physica A*.

Patrini, G., van den Berg, R., Forre, P., Carioni, M., Bhargav, S., Welling, M., Genewein, T., and Nielsen, F. (2020). Sinkhorn autoencoders. In *Uncertainty in Artificial Intelligence*, pages 733–743. PMLR.

Paul, M. and Dredze, M. (2012). Factorial LDA: Sparse multi-dimensional text models. In *Advances in Neural Information Processing Systems*, pages 2582–2590.

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

Phuong, M., Welling, M., Kushman, N., Tomioka, R., and Nowozin, S. (2018). The mutual autoencoder: Controlling information in latent code representations. *International Conference on Learning Representations*.

Prince, S. (2012). *Computer Vision: Models Learning and Inference*. CUP.

Qian, T. and Jaeger, T. F. (2011). Topic shift in efficient discourse production. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *openai.com*.

Raheja, V. and Tetreault, J. (2019). Dialogue act classification with context-aware self-attention. 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 3727–3733.

Raux, A., Langner, B., Bohus, D., Black, A. W., and Eskenazi, M. (2005). Let's go public! taking a spoken dialog system to the real world. In *Ninth European conference on speech communication and technology*.

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

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286.

Roberts, A., Engel, J., Raffel, C., Hawthorne, C., and Eck, D. (2018). A hierarchical latent vector model for learning long-term structure in music. In *International Conference on Machine Learning*, pages 4364–4373.

Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Rus, V. and Lintean, M. (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.

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press.

Semeniuta, S., Severyn, A., and Barth, E. (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.

Shen, X., Su, H., Niu, S., and Demberg, V. (2018). Improving variational encoder-decoders in dialogue generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Shriberg, E., Dhillon, R., Bhagat, S., Ang, J., and Carvey, H. (2004). The icsi meeting recorder dialog act (mrda) corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004*, pages 97–100.

Skiena, S. S. (2008). *The Algorithm Design Manual*. Springer.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. (2016). How to train deep variational autoencoders and probabilistic ladder networks. In *33rd International Conference on Machine Learning (ICML 2016)*.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Ess-Dykema, C. V., and Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. (2018). Wasserstein auto-encoders. In *International Conference on Learning Representations*.

Vahdat, A. and Kautz, J. (2020). Nvae: A deep hierarchical variational autoencoder. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19667–19679. Curran Associates, Inc.

Verma, R., Shashidhar, N., and Hossain, N. (2012). Detecting phishing emails the natural language way. In *European Symposium on Research in Computer Security*, pages 824–841. Springer.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Wallace, B. C., Trikalinos, T. A., Laws, M. B., Wilson, I. B., and Charniak, E. (2013). A generative joint, additive, sequential model of topics and speech acts in patient-doctor communication. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1765–1775.

Wang, D., Lin, C., Zhong, L., and Wong, K.-F. (2020). Dialogue state tracking with pretrained encoder for multi-domain trask-oriented dialogue systems. *arXiv preprint arXiv:2004.10663*.

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330.

Wrede, B. and Shriberg, E. (2003). Relationship between dialogue acts and hot spots in meetings. In *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721)*, pages 180–185. IEEE.

Wu, J., Huang, Z., Acharya, D., Li, W., Thoma, J., Paudel, D. P., and Gool, L. V. (2019). Sliced wasserstein generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3713–3722.

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

Xu, P., Cheung, J. C. K., and Cao, Y. (2020). On variational learning of controllable representations for text without supervision. In *International Conference on Machine Learning*, pages 10534–10543. PMLR.

Xu, W., Sun, H., Deng, C., and Tan, Y. (2017). Variational autoencoder for semi-supervised text classification. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3358–3364.

Xu, Y. and Reitter, D. (2016). Entropy converges between dialogue participants: Explanations from an information-theoretic perspective. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 537–546.

Yang, Z., Hu, Z., Salakhutdinov, R., and Berg-Kirkpatrick, T. (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.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Zhao, T., Lee, K., and Eskenazi, M. (2018). Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1098–1107.

Zhao, T., Zhao, R., and Eskenazi, M. (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.

Zhu, Q., Bi, W., Liu, X., Ma, X., Li, X., and Wu, D. (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. Association for Computational Linguistics.

Appendices

Appendix A

Identified Holes in Different Dimensions

A.1 Paths Traversed and Depths Reached till TDC Halts

Table A.1: *Average quantities of traversed paths and reached depths in each C of 4D until 200 latent holes are identified.*

Model	Yelp15		Yahoo		SNLI		Wiki	
	path	depth	path	depth	path	depth	path	depth
VAE	101.0	4.7	102.9	5.2	99.6	11.2	95.3	3.7
Cyc-VAE	112.5	8.5	113.8	5.4	90.7	10.5	119.3	6.4
β-VAE	128.3	13.5	135.5	16.6	89.6	4.8	127.3	7.4
BN-VAE	135.4	19.2	136.1	19.9	121.1	8.9	139.4	9.9
iVAE$_{\text{MI}}$	142.1	22.7	140.8	21.1	132.5	13.5	140.4	15.8

Table A.2: *Average quantities of traversed paths and reached depths in each C of 3D until 200 latent holes are identified.*

Model	Yelp15		Yahoo		SNLI		Wiki	
	path	depth	path	depth	path	depth	path	depth
VAE	99.9	8.0	99.3	7.3	99.7	38.6	85.5	4.9
Cyc-VAE	110.0	9.3	106.0	11.3	88.3	14.4	118.7	11.9
β-VAE	120.8	14.9	133.4	13.7	88.2	9.4	125.0	13.5
BN-VAE	132.4	15.8	134.6	14.6	120.5	10.5	131.5	14.3
iVAE$_{\text{MI}}$	141.9	16.4	137.0	15.0	131.4	17.7	134.2	16.4

As shown in Tabs. A.1, and A.2, the similar trend can be found as the Tab. 5.2 in § 5.4.2.

A.2 Latent Space Visualisation

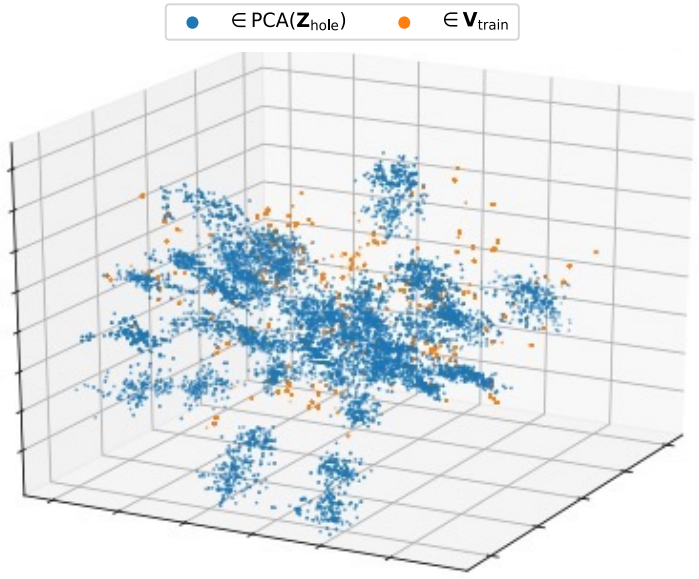


Figure A.1: *Visualisation of the latent space of Cyc-VAE (trained on the Yelp15 dataset).*

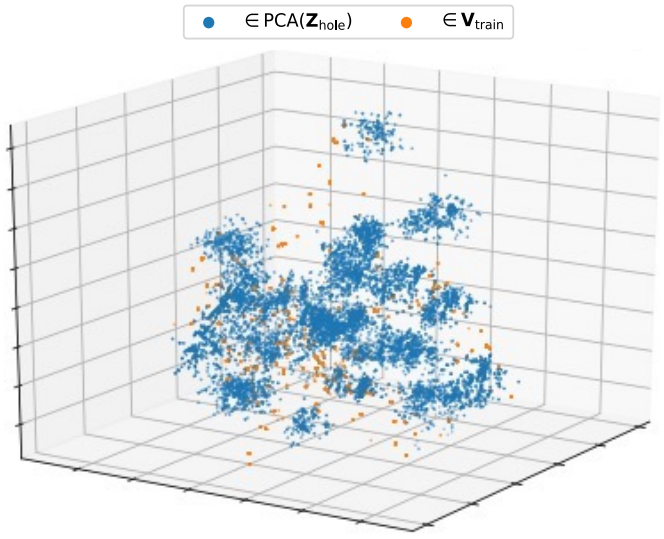


Figure A.2: *Visualisation of the latent space of β-VAE (trained on the Yahoo dataset).*

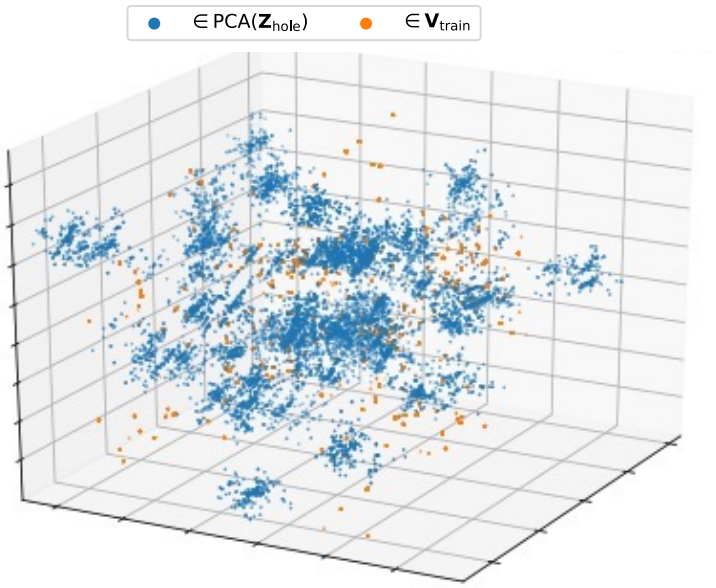


Figure A.3: *Visualisation of the latent space of BN-VAE (trained on the SNLI dataset).*

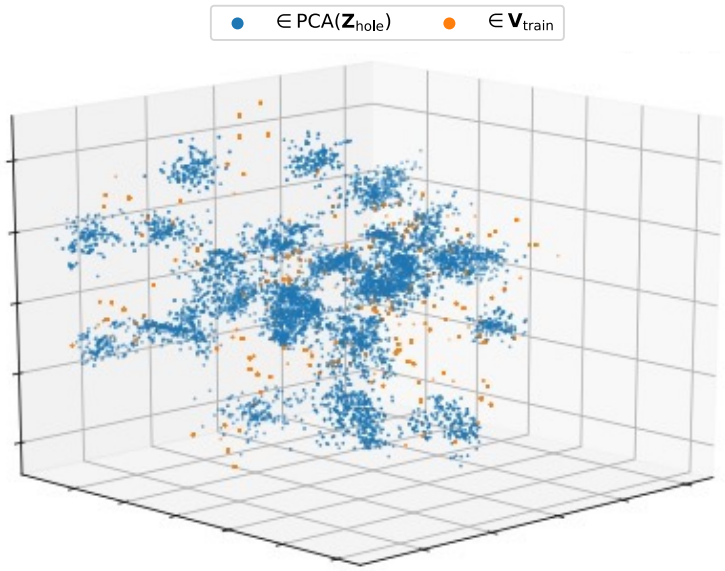


Figure A.4: *Visualisation of the latent space of $iVAE_{MI}$ (trained on the Wiki dataset).*

Figure A.1, A.2, A.3 and A.4 show that holes are ubiquitously distributed in the entire latent space for different baselines.

A.3 Impact of Latent Holes When $d_r \in \{3, 4\}$

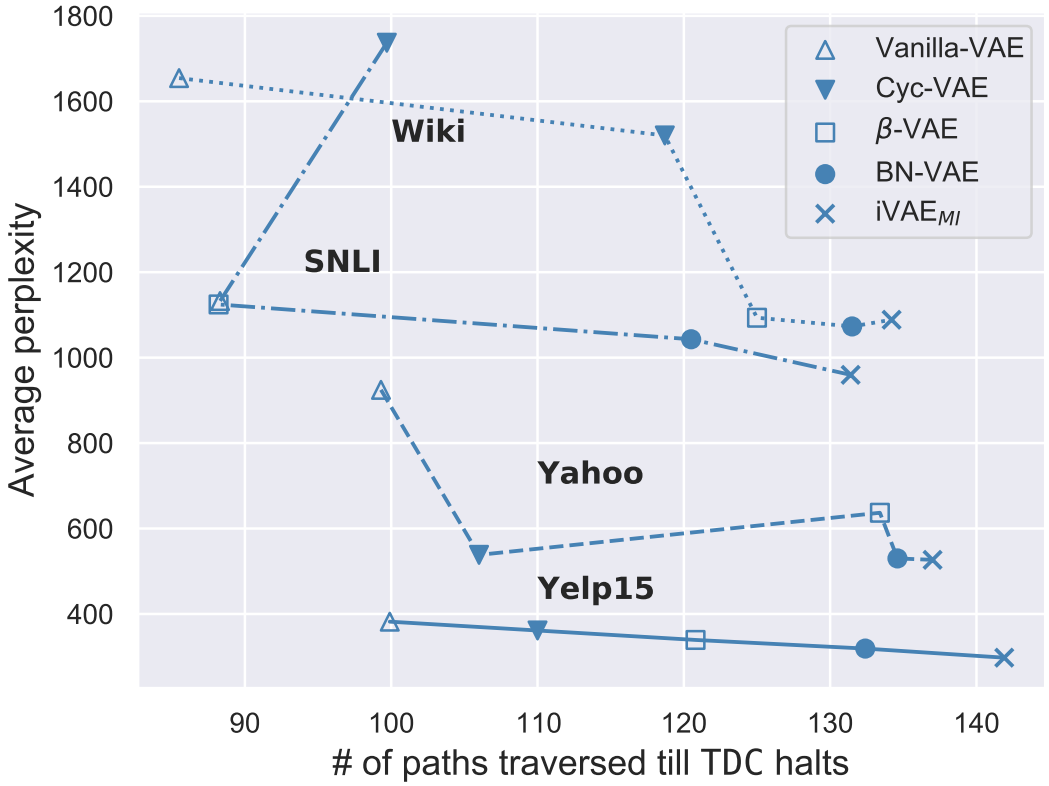


Figure A.5: *Average PPL and the number of paths traverseduntil* `TDC` *halts for all setups* ($d_r = 3$).

Figure A.5 and A.6 show the similar trend as Figure 5.4 in § 5.4.2.

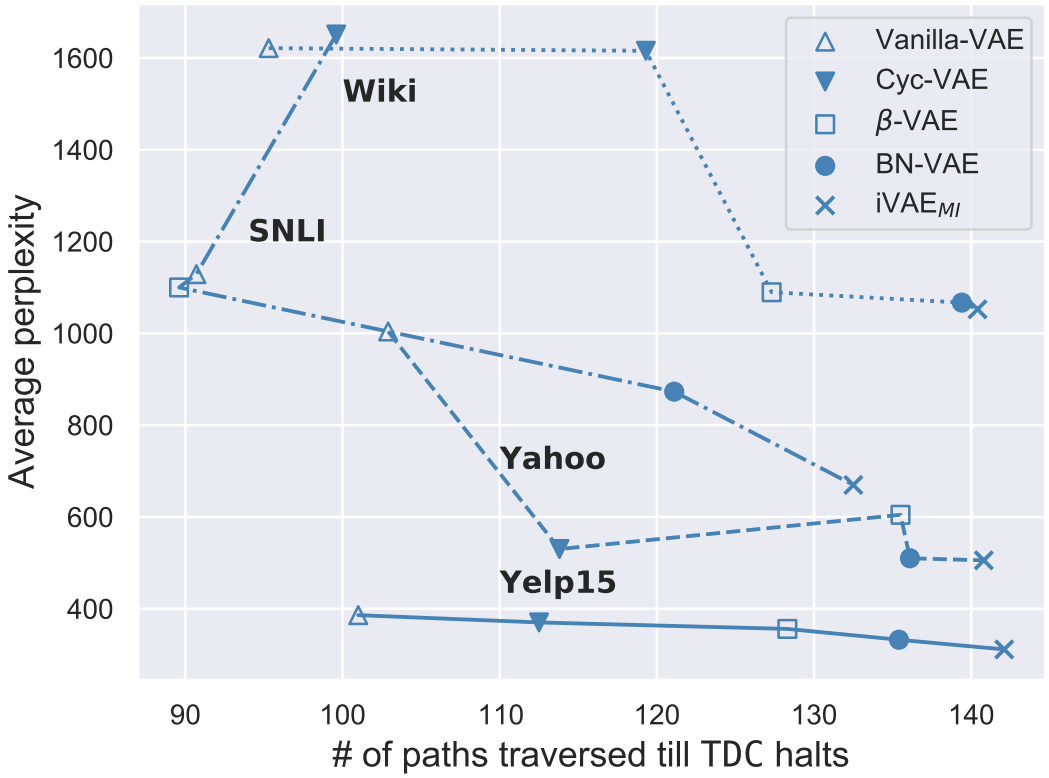


Figure A.6: *Average PPL and the number of paths traverseduntil* `TDC` *halts for all setups* ($d_r = 4$).

A.4 Quantity Distribution of Identified Holes

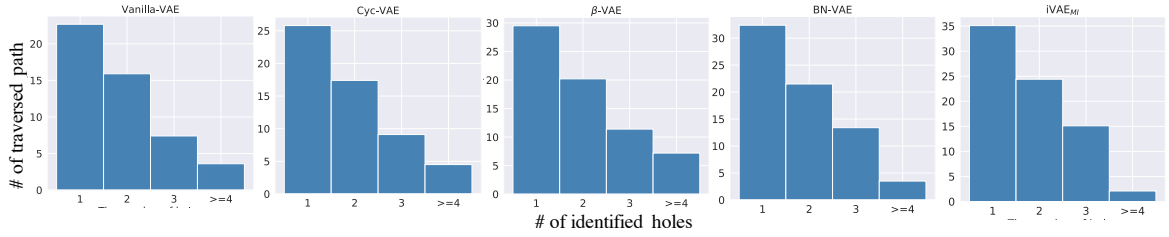


Figure A.7: *Quantity distribution of identified holes per discontinuous latent path for models trained on the Wiki dataset when* $d_r = 3$.

Figure A.8: *Quantity distribution of identified holes per discontinuous latent path for models trained on the Wiki dataset when* $d_r = 4$.

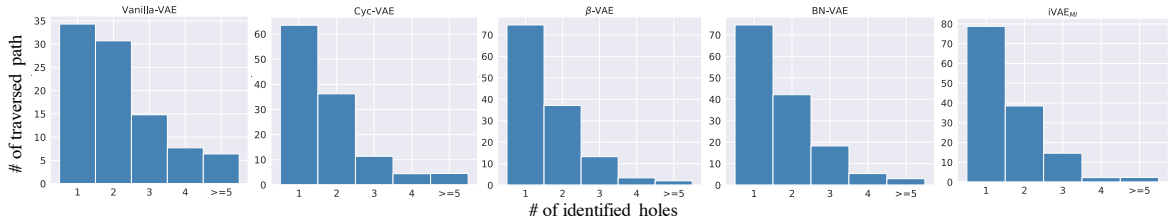


Figure A.9: *Quantity distribution of identified holes per discontinuous latent path for models trained on the Yelp15 dataset when* $d_r = 8$.

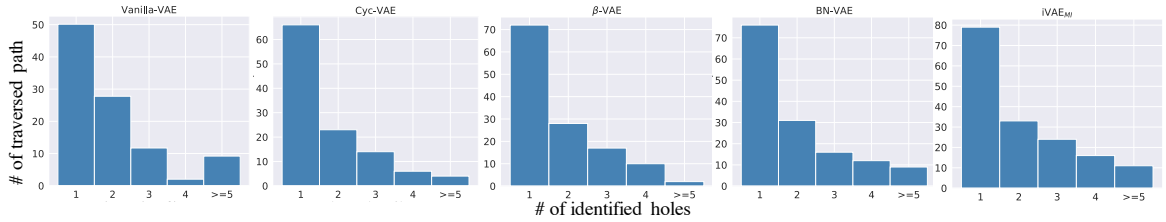


Figure A.10: *Quantity distribution of identified holes per discontinuous latent path for models trained on the Yahoo dataset when* $d_r = 8$.

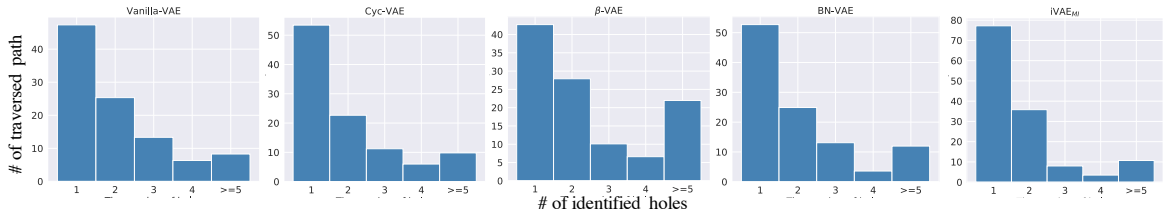


Figure A.11: *Quantity distribution of identified holes per discontinuous latent path for models trained on the SNLI dataset when* $d_r = 8$.