Ph.D. DISSERTATION

Deep Generative Data Augmentation for Natural Language Processing

딥러닝 기반 생성 모델을 이용한 자연어처리 데이터 증강 기법

BY

KANG MIN YOO

FEBRUARY 2020

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Deep Generative Data Augmentation for Natural Language Processing

딥러닝 기반 생성 모델을 이용한 자연어처리 데이터 증강 기법

BY

KANG MIN YOO

FEBRUARY 2020

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Deep Generative Data Augmentation for Natural Language Processing

딥러닝 기반 생성 모델을 이용한 자연어처리 데이터 증강 기법

지도교수 이 상 구

이 논문을 공학박사 학위논문으로 제출함

2020년 2월

서울대학교 대학원

컴퓨터공학부

유 강 민

유강민의 공학박사 학위 논문을 인준함

2020년 2월

위 원 장:	김 형 주	(인)
부 위 원 장:	이 상 구	(인)
위 원:	김 건 희	(인)
위 원:	송 현 오	(인)
위 원:	박 재 휘	(인)

Abstract

Recent advances in generation capability of deep learning models have spurred interest in utilizing deep generative models for unsupervised generative data augmentation (GDA). Generative data augmentation aims to improve the performance of a downstream machine learning model by augmenting the original dataset with samples generated from a deep latent variable model. This data augmentation approach is attractive to the natural language processing community, because (1) there is a shortage of text augmentation techniques that require little supervision and (2) resource scarcity being prevalent. In this dissertation, we explore the feasibility of exploiting deep latent variable models for data augmentation on three NLP tasks: sentence classification, spoken language understanding (SLU) and dialogue state tracking (DST), represent NLP tasks of various complexities and properties – SLU requires multi-task learning of text classification and sequence tagging, while DST requires the understanding of hierarchical and recurrent data structures. For each of the three tasks, we propose a task-specific latent variable model based on conditional, hierarchical and sequential variational autoencoders (VAE) for multi-modal joint modeling of linguistic features and the relevant annotations. We conduct extensive experiments to statistically justify our hypothesis that deep generative data augmentation is beneficial for all subject tasks. Our experiments show that deep generative data augmentation is effective for the select tasks, supporting the idea that the technique can potentially be utilized for other range of NLP tasks. Ablation and qualitative studies reveal deeper insight into the underlying mechanisms of generative data augmentation. As a secondary contribution, we also shed light onto the recurring posterior collapse phenomenon in autoregressive VAEs and, subsequently, propose novel techniques to reduce the model risk,

which is crucial for proper training of complex VAE models, enabling them to synthesize better samples for data augmentation. In summary, this work intends to demonstrate and analyze the effectiveness of unsupervised generative data augmentation in NLP. Ultimately, our approach enables standardized adoption of generative data augmentation, which can be applied orthogonally to existing regularization techniques.

Keywords: natural language processing, variational autoencoder, data augmentation, language generation, latent variable model, generative model, text classification, spoken language understanding, dialogue state tracking

Student Number: 2014-21763

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Recently, impressive advances in generative capabilities of deep learning models have been observed in both the vision domain (Karras et al., 2019) and the language domain (Radford et al.). Photo-realistic synthetic images from generative adversarial networks (GAN) and coherent texts generated from pre-trained transformer networks have become harder to distinguish from the real counterparts with human perceptions. The advances are attributed partly to the expansion of our knowledge in effective architectural choices and optimization techniques to maximize the expressive power of existing deep learning models (Vaswani et al., 2017; Arjovsky et al., 2017), while some other are attributed to even larger models realized through constantly improving computational hardwares (Devlin et al., 2018). That begs the question: *can we leverage ever-improving generative models to assist the training of supervised machine learning models?*

Data augmentation is defined as a class of data-oriented techniques for expanding existing datasets with class-preserving synthetic data points to alleviate the overfitting of resulting machine learning models (Tanner and Wong, 1987). It has been adopted in various forms (Krizhevsky et al., 2012; Dao et al., 2019) and in various domains (Van Dyk and Meng, 2001; Zhang et al., 2015) as an orthogonal approach to regularization techniques. With recent advances in deep general model, hypothetically, one might be able to construct a gen-

erator that perfectly mimics the true data distribution, which then can be used to sample as many data points as one wishes for augmenting a training set. In practice, such "perfect" data models are infeasible, but, with the recent emergence of deep generative models that achieve impressive realism, it is unclear whether near-perfect deep generative models are tangible tools for exploring novel samples for training set augmentation. Imperfect data models could probabilistically generate incoherent data samples, which might distort the augmented data distribution, negatively affecting the downstream model. However, if the benefit from discovering valuable novel samples from the generative model outweighs the degradation caused by data distribution distortion, then data augmentation using generative models could potentially be a feasible technique for optimizing machine learning models. This theoretical phenomenon had actually been preliminarily observed in a prior work (Hu et al., 2017), where the authors showed that a VAE-based text generative model can be used to synthesize class-preserving samples to achieve data augmentation for text classification.

The question is highly relevant for the natural language processing (NLP) community, as there are two major hurdles in NLP that are not as prounounced as in other domains. First, low-level data augmentation, for example introducing small transformative noises in the image space while maintaing the original label, is not readily available and not fully understood in the NLP domain. Class-invariant image scaling and transformation techniques have been widely adopted in image classification tasks (Krizhevsky et al., 2012), but the equivalent in the NLP domain has yet to be discovered. There have been various text augmentation techniques (Zhang et al., 2015; Wei and Zou, 2019) proposed in the past, but the techniques are mostly task-specific or domain-specific and are still in the infant stage in terms of adoption rate compared to image augmentation techniques.

Specifically, an ideal data augmentation technique must be able to generate (1) *class-preserving* and (2) *realistic* samples, where the latter property means that the synthetic samples must adhere to the true data distribution. Current approaches for data augmentation in NLP tasks largely revolve around thesaurus data augmentation (Zhang et al., 2015), in which words that belong to the same semantic role are substituted with one another using a preconstructed lexicon, and noisy data augmentation (Wei and Zou, 2019) where random editing operations such as insertion, deletion and substitution is applied to the language space. Thesaurus data augmentation satisfies both properties of an ideal technique, but it requires a set of handcrafted semantic dictionaries, which are costly to build and maintain; whereas noisy data augmentation does not guarantee synthetic samples to be realistic.

Second, linguistic resource scarcity is a persistent issue for many language-specific, task-specific, or domain-specific problems in NLP (Besacier et al., 2014; Banea et al., 2008), which exposes the downstream machine learning models to the risk of overfitting. The typical approaches to alleviate the issue can classified into two categories, in which (1) *resource augmentation* techniques aims to enrich the training set with better data construction methods or external resources (Barnard et al., 2009; Scannell, 2007) and (2) *model augmentation* techniques achieves the goal by developing complex models or training techniques to increase the generalization capabilities (Cai et al., 2014; Lu et al., 2011). Generator-based data augmentation could be classified as a resource augmentation technique with self-discovery of unseen samples through the explorative aspect of deep latent variable models - the main driving engine behind generative models. Standardization of generator-based data augmentation could potentially be indiscriminately used to alleviate data-related problems for any structured text datasets to improve the robustness of downstream models.

Before we delve into the feasibility of deep generative models for data aug-

mentation, we first turn to recent advances in *latent variable models*, a general class of probabilistic approach for modeling complex datasets with certain prior knowledge about the internal governing structure of the datasets. Latent variable models pertains modeling of a data distribution $\mathbf{x}$ through the inference of latent variables $\mathbf{z}$ that best explain the dataset. With the rise of deep neural networks, latent variable models became extremely effective in modeling and generating images (Goodfellow et al., 2014) and texts (Bowman et al., 2016). Variational inference methods, which are techniques for approximating the inference of latent variable models, have also contributed to the explosion of generative models, especially those that are based on variational autoencoders (VAE) (Kingma and Welling, 2013). Generative adversarial networks (GAN) (Goodfellow et al., 2014) are another prominent class of deep generative models due to its intuitive framework setup. Although both approaches have received significant attention from the overall machine learning community, VAEs have been more actively researched in the NLP community because of the difficulty of implementing GAN architectures for text generation in practice (Zhang et al., 2016c) and the fact that the NLP community is generally interested in the ability to map the feature space to the latent space (encoder/inference network) as well. Furthermore, the fact that VAEs allow explicit definition of distributional families makes VAEs more suitable for controlled text generation than GANs.

Synthesizing samples from latent variable models and using them as augmentational data can be helpful in two ways. First, we can understand the process of generator-based data augmentation as a form of data-side "regularization". The original dataset might contain sampling biases, which are one type of the major causes of overfitting (Vezhnevets and Barinova, 2007). By sampling from a latent variable model approximated with variational inference and using the generated samples to augment the original dataset, one could effectively "regularize" the training data distribution. Second, generative sampling from

deep latent variable models offers an automatic way to discover novel samples, thanks to the prior and distributional family assumptions. Because the true data distribution of real-world data can not be perfectly modeled using the usual choices of distributional families (e.g. Gaussian), the excess probabilities imperfectly inferred by the posterior or the prior distributions facilitate in exploration. Hence, the VAE-based model family is an attractive choice as the backbone generator framework: the existence of the inference network and the flexibility of choosing distributional families grants sample realism and the ability to explore novel data points.

Based on aforementioned intutions, we summarize our research questions as follows, for which this dissertation will be dedicated to provide insights.

1. **Can deep generative models be leveraged for data augmentation?**

2. **To what extent does the generator-driven data augmentation help in NLP tasks?**

3. **How can we maximize the benefit of generative data augmentation?**

By the end of the dissertation, we not only wish to offer insights for the feasibility of employing generative data augmentation as a general machine learning technique in NLP tasks, but we also wish to explore the underlying mechanisms and conditions on which downstream benefit of generative data augmentation is maximized. Fow now, we use the term *generative data augmentation* to define a class of data augmentation techniques that utilize deep generative models to generative synthetic samples for improving downstream models, although we elaborate related terms in Chapter 2,

1.2 Dissertation Overview

The contributions of the dissertation is summarized as follows:

1. We conduct a comprehensive survey on data augmentation techniques in current literature, by offering a fresh perspective on the categorization of existing data augmentation techniques. Based on the survey, we formalize the notion of *generative data augmentation*, a novel type of data augmentation technique that leverage deep generative models based on deep latent variables for improving downstream NLP tasks.

2. To demonstrate the effectiveness of generative data augmentation to NLP, we propose three deep latent variable models for learning to generate fully annotated text datasets specialized in following NLP tasks: *sentence classification*, *spoken language understanding* and *dialogue state tracking*. We conduct statistical tests to support our hypothesis that generative data augmentation can reliably yield positive results for the select tasks. In addition, we conduct qualitative analysis to show that the samples generated from our models are realistic.

3. Autoregressive VAEs, on which all of our proposed generative models are based, are known to be susceptible to the *posterior collapse* phenomenon. In order to reduce the risk, we propose various techniques, specifically I-VAE training policy (Chapter 3) and the mutual information trick (Chapter 5). Although these techniques are proposed with specific VAE structures in mind, these methods are applicable to other VAE-based models with similar structural characteristics.

The rest of the thesis is structured as follows. In Chapter 2, we provide an overview of the related research area and the relevant trends in recent years as the foundation of the dissertation. In Chapter 3, we begin our exploration

of generative data augmentation in NLP tasks with a relatively simple task: text classification, which serves as the demonstration of the feasibilty of generative data augmentation in real text corpora. We show that generative data augmentation, implemented by a VAE model designed specifically for modeling sentence-label pairs, yields positive results on downstream classifier models. In Chapter 4, we investigate generative data augmentation for the spoken language understanding task, which consists of two sub-tasks: intent classification and slot-filling (sequence tagging). We tackle the problem as a multi-task problem, and propose a VAE-based joint-learning model that is able to generate novel samples for improving the performance of baseline NLU models. We conduct additional experiments to examine the various conditions in which the generative data augmentation technique is most effective, such as data scarce scenarios and varying synthetic to real data ratios. In Chapter 5, we take the final challenge in applying generative data augmentation on a relatively complex task in the NLP domain – dialogue state tracking. We propose a novel VAE architecture that is able to fully model structured goal-oriented dialogues by leveraging its hierarchical and recurrent design. We realized the omplex VAE model with the help of two proposed techniques for reducing the risk of posterior collapse. Using state-of-the-art dialogue state trackers as the baseline, we empirically show that generative data augmentation is effective for the complex task. In Chapter 6, we review the implications and limitations of our findings, and we conclude the dissertation with suggestions on possible future research directions.

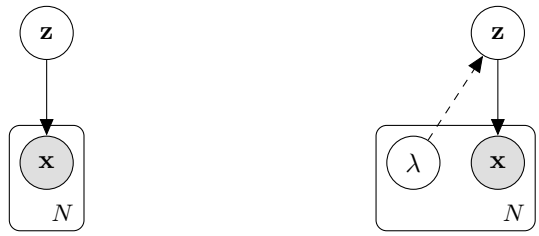
Chapter 2

Background and Related Work

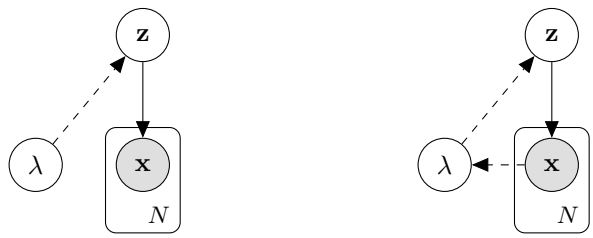
In this chapter, we provide an overview of the background material related to the subjects discussed in this thesis. Section 2.1 introduces the theoretical background and major concepts for latent variable models, as well as some recent developments in the field. Section 2.2 reviews various approaches to data augmentation in the current literature.

2.1 Deep Latent Variable Models

Latent variable models provide a concrete mathematical foundation for the fundamental problem of data analysis. The natural language processing domain has taken great advantage of the expressive flexibility of latent variable models, exploiting relevant prior knowledge about the data to achieve state of the art models for topic modeling (Blei et al., 2003), machine tranlsation (Blunsom et al., 2008; Zhang et al., 2016a), grammar induction (Klein and Manning, 2004; Kim et al., 2019), and dialogue modeling (Serban et al., 2017; Shen et al., 2017b). With recent advancements and rediscovery of the generalizing power of deep neural networks (Zhang et al., 2016b), existing latent variable models have been expanded to take advantage of the new paradigm. By exploiting of both methodologies, in which the probabilistic graphical model theory provides the flexibility of modeling complex dependencies among latent variables and deep neural networks realize those complex dependencies using parameterized architectures, the natural language processing domain has flourished with



(a) Latent Variable Model (b) Vanilla VI



(c) Stochastic VI (d) Amortized VI

Figure 2.1: Types of variational inference (VI) in latent variable models

record-breaking researches such as machine comprehension (Yang et al., 2019b) and machine translation (Wu et al., 2016).

The ultimate goal of designing and realizing latent variable models is to propose a set of latent variables $\mathbf{z}$ that best explain the observable data $\mathbf{x}$:

$$p(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) d\mathbf{z} \tag{2.1}$$

However, the integration over $\mathbf{z}$ is intractable and thus requires an approximate inference method, such as variational inference, to compute the solution. Variational inference, also known as variational bayes, is a set of appoximation methods that assume a family of distributions over the latent variables with variational parameters λ (Figure 2.1[1]).

Given a dataset $\mathbf{x}_1, \ldots, \mathbf{x}_N$, variational inference needs to find variational parameters $\lambda_1, \ldots, \lambda_N$ that can be used to define a variational family of distri-

[1]Note that the model parameters θ have been omitted for brevity.

butions $q_\lambda(\mathbf{z})$ parameterized by λ (Figure 2.1b). The objective for variational inference is to maximize the evidence lower bound (ELBO):

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\lambda(\mathbf{z})}[\log p(\mathbf{x} \mid \mathbf{z})] - D_{\mathrm{KL}}(q_\lambda(\mathbf{z}) \| p(\mathbf{z})) \tag{2.2}$$

Stochastic variational inference (SVI) (Hoffman et al., 2013) optimizes directly for instance-specific variational distributions by applying gradient ascent on the same set of parameters λ for all data (Figure 2.1c). In practice, the variational parameters λ and model parameters θ are optimized in alternation, potentially falling into local optima where the approximate posterior or the model posterior collapses onto the prior[2].

Amortized variational inference (AVI) improves upon SVI by using a global parameterized model to predict λ for each data point (Figure 2.1d). Hence, in AVI, λ is no longer a set of parameters but an intermediate factor, and we usually use ϕ to denote the set of parameters that are used to predict the latent variables: $q_\phi(\mathbf{z} \mid \mathbf{x})$.

2.1.1 Variational Autoencoder (VAE)

Variational autoencoders are a special case of latent variable models with amortized variational inference (Kingma and Welling, 2013) and it is a popular application of variational inference for deep latent variable models. The global parameterized model for predicting λ is called the *inference network* or the *encoder network*, and it is usually modeled by complex deep learning models which depend on the input modality. The objective of VAE is to maximize the following ELBO:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x} \mid \mathbf{z})] - D_{\mathrm{KL}}(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})) \tag{2.3}$$

[2]More details about the posterior collapse phenomenon can be found in Appendix A.

In the right-hand side of the ELBO, the first term is called the *reconstruction term* and the second term is called the *KL-divergence term*, which encourages the minimization of the distributional distance between the approximate posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ and the prior $p(\mathbf{z})$.

The usual choice for the prior is the unit multivariate Gaussian distribution, as the analytic solution for the KL-divergence term can be computed easily. The KL-divergence of two Gaussian distributions is given by

$$D_{\mathrm{KL}}(\mathcal{N}(\mu_x, \sigma_x) \| \mathcal{N}(\mu_y, \sigma_y)) = \log \frac{\sigma_y}{\sigma_x} + \frac{\sigma_x^2 + (\mu_x - \mu_y)^2}{2\sigma_y^2} - \frac{1}{2}. \tag{2.4}$$

If one of the distributions is a unit Gaussian, then the analytic solution can be rewritten to

$$D_{\mathrm{KL}}(\mathcal{N}(\mu_x, \sigma_x) \| \mathcal{N}(0, 1)) = -\log \sigma_x + \frac{\sigma_x^2 + \mu_x^2}{2} - \frac{1}{2}. \tag{2.5}$$

The approximate posterior is also applied with the mean field assumption, which states that each latent variable is independent from each other, allowing factorization of the joint posteriors. This is realized by choosing distributions in which the covariance of the multivariate Gaussian distribution is assumed to be diagonal: $q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\mu_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x})I)$. Due to the strong assumption of the family of distributions, some works have proposed using other classes of distributions such as Von Mises-Fisher (Xu and Durrett, 2018) or techniques that enable the modeling of arbitrary distributions such as normalizing flow (Rezende and Mohamed, 2015). However, due to the implementation and inference complexity of more advanced distributions, multivariate Gaussian distributions with diagonal covariance remain the more prevalent choice for VAEs.

2.1.2 Deep Generative Models and Text Generation

Many variants of VAEs have been explored in the language domain. Notably, VAEs with sequence-to-sequence architecture (Sutskever et al., 2014) as the backbone for the encoder and decoder has been explored with variational recurrent auto-encoders (VRAE) (Fabius and van Amersfoort, 2014). Generative adversarial networks (GAN) are another class of latent variable models with implicit latent distribution (Goodfellow et al., 2014), which has seen adaptations for texts (Yu et al., 2017; Fedus et al., 2018). Recently, some work have explored deep generative models in the context of controllable generation and style transfer (Hu et al., 2017; Shen et al., 2017a; Fu et al., 2018), specifically using variational generative models.

2.2 Data Augmentation

Many machine learning pipelines systematically employ augmentation for the training data in one way or the other to improve the model's generalizing capability. Adoption of data augmentation techniques has been steadily growing in domain-specific cases, especially in resource-scarce situations, such as medical imaging (Nguyen et al., 2019; Yoon et al., 2019) and domain-specific spoken language understanding tasks (Hou et al., 2018; Kurata et al., 2016b). To gain deeper understanding of data augmentation, this section provides a technical overview of the technique. Furthermore, we offer an systematic perspective on the classification of data augmentation, where we categorize techniques based on the source of augmentational knowledge and the type of supervision. The proposed perspective lays the foundation for investigating the mechanism behind data augmentation.

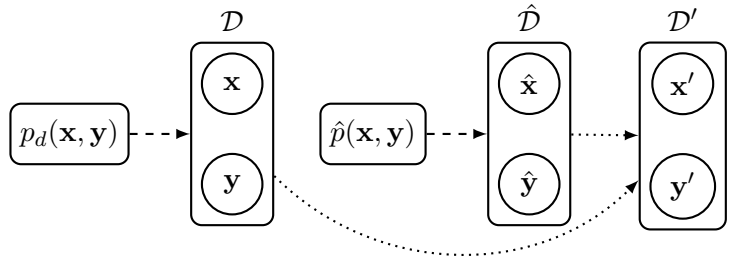


Figure 2.2: Depiction of general data augmentation.

2.2.1 General Description

Data augmentation has been widely adopted in various domains as a practical and powerful technique to alleviate data insufficiency for data-hungry deep learning models, as it has been shown to be effective for tasks including, but not limited to, multi-class image classification (Krizhevsky et al., 2012), visual object detection (Zhong et al., 2017), auditory detection (Takahashi et al., 2016), text classification (Zhu et al., 2003), spoken language understanding (Hou et al., 2018; Yoo et al., 2019), machine translation (Fadaee et al., 2017), and relation classification (Xu et al., 2016).

For the rest of this subsection, we establish a set of notations for describing data augmentation. For clarity without the loss of generality, we assume the task for which the data augmentation is used is text classification, in which we fit a classifier model $f_\theta(\mathbf{y} \mid \mathbf{x})$ parameterized by θ on a set of data points $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$. Here, $\mathbf{x} \in \mathcal{X}$ are the data point features defined over the feature space $\mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$ are the corresponding labels defined over the label space $\mathcal{Y}$. Data augmentation has been adopted in various forms, but the common idea can be illustrated in Figure 2.2. Given a dataset of input features and labels $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$, which was sampled from the empirical data distribution $p_d(\mathbf{x}, \mathbf{y})$, the goal of data augmentation is to enhance the original dataset $\mathcal{D}$ with a new augmentational dataset $\hat{\mathcal{D}}$ (which is

sampled or generated from the augmentational data distribution $\hat{p}(\mathbf{x}, \mathbf{y})$). The augmentational dataset $\hat{\mathcal{D}}$ is combined with the original dataset $\mathcal{D}$ to form a new dataset $\mathcal{D}'$ that follows an updated data distribution. By fitting the classifier model f_θ on $\mathcal{D}'$ instead of the original dataset $\mathcal{D}$, we expect less overfitting and improved robustness of the downstream classifier models, achieving better results on unseen test data. After applying data augmentation, the new data distribution that is learned by the downstream classifier models is a mixture of the original and the augmentational data distributions:

$$p^\star(\mathbf{x}, \mathbf{y}) = \frac{1}{1+\lambda} p_d(\mathbf{x}, \mathbf{y}) + \frac{\lambda}{1+\lambda} \hat{p}(\mathbf{x}, \mathbf{y}) \tag{2.6}$$

where λ is a hyperparameter that controls the mixture ratio between the two distributions. The motivation of employing data augmentation is to bring the new data distribution $p^\star$ closer to the true data distribution p by diluting the original distribution with augmentational data.

Complying to this common framework for data augmentation, various forms of data augmentation have been suggested and adopted in previous works. We categorize most data augmentation techniques based on the source of augmentational knowledge and the type of supervision. The following subsection goes through each category and provide detailed explanation and relevant cases.

2.2.2 Categorization of Data Augmentation

In order to synthesize or sample augmentational data that bring the final augmented data closer to the true data distribution, there must a mechanism which introduces some form of knowledge into the data. For example, knowledge might be injected through direct manipulation of the image feature space that is known to be class-preserving. Class-preserving feature transformations introduces the prior knowledge about the image spaces that maintains the class of the original data. For another examples, knowledge might be self-discovered through

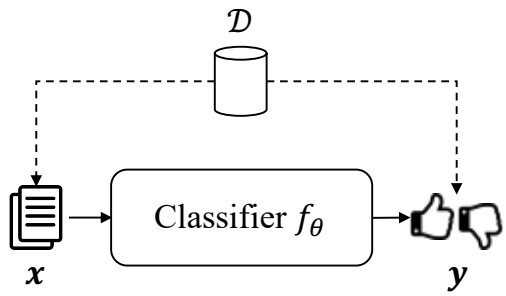


Figure 2.3: An example of supervised learning for text classification without data augmentation.

exploration in the latent space deep latent variable models. Hence, data augmentation techniques can be categorized based on how external knowledge is acquired and injected into the machine learning framework as the form of data. As an extension of the framework described in the previous subsection, we illustrate how different types of data augmentation interact with the main machine learning framework. The basic machine learning framework is shown in Figure 2.3.

Supervised Data Augmentation

Supervised data augmentation relies on external knowledge of the joint data-label distribution, whether it be from external datasets from other but similar domains or freshly constructed human-annotated datasets. Supervised data augmentation can be further sub-categorized based on the source of the augmentational data.

External Data Augmentation. This class of data augmentation techniques aims to acquire knowledge from o ther data sources that might or might not share the same domain as the original dataset (Figure 2.4). The augmentational data might contain data samples irrelevant to the target data distribution, however, the intuition is that deep learning-based classifier models are

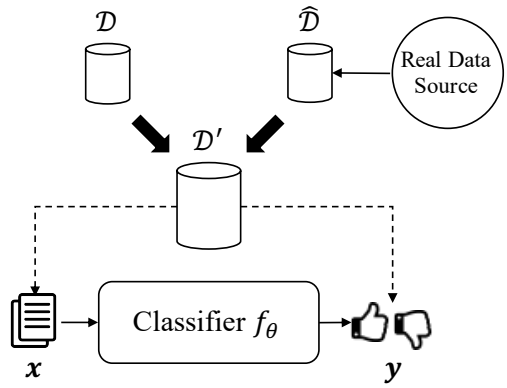


Figure 2.4: Supervised data augmentation with external data.

able to distinguish the relevant features from the noise, making the resulting model more robust. Data augmentation through expanding datasets from similar datasets have been used widely in various tasks, achieving state of the art models in problems such as image classification, visual question-answering (VQA) (Jiang et al., 2018) and text classification (Conneau et al., 2017).

In some cases, only the input features might be available as external data, hence an *oracle* might be required to annotate the labels based on the available input features (Nguyen et al., 2019), as depicted in Figure 2.5.

Transformational Data Augmentation. Data can be sourced from the original dataset itself, if the knowledge about the class-invariant transformations is known beforehand (Figure 2.6). An oracle that contains the external knowledge injects it into the augmented dataset by generating variations within the class-invariant space for each data point. The operations that introduce variations in the data space may be as simple as translations in images or complex - so complex that the operation may require parameterization (Cubuk et al., 2019). Particularly in the vision domain, utilization of class-invariant kernel filters, geometric transformations, random masking, color space transformations has been explored by various works (Krizhevsky et al., 2012; Perez and Wang,

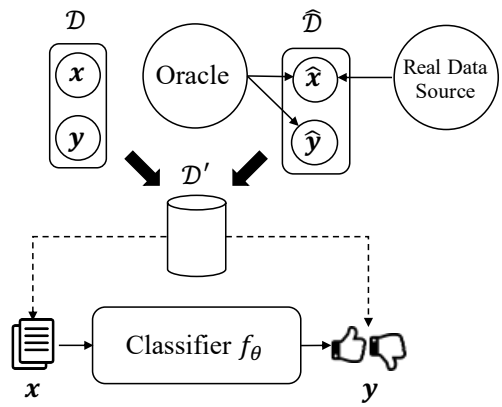


Figure 2.5: Supervised data augmentation with external data and an oracle annotator.

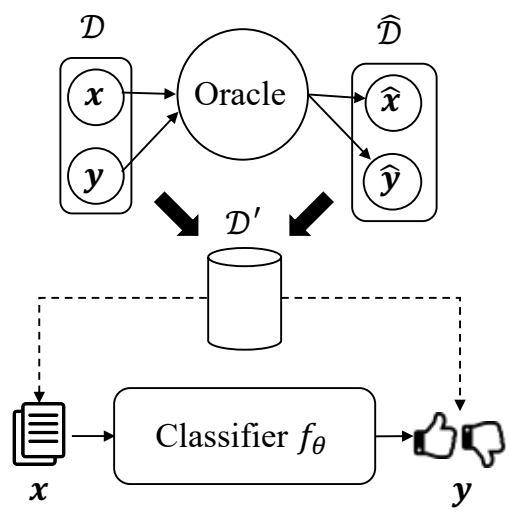


Figure 2.6: Transformational data augmentation.

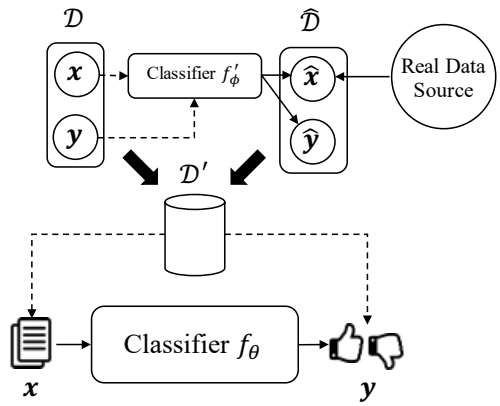


Figure 2.7: Semi-supervised data augmentation.

2017; Shorten and Khoshgoftaar, 2019). As for the NLP domain, noisy data augmentation (synonym replacement, random insertion, random swap and random deletion), which might behave as approximation to class-invariance transformations, had been adopted for text classification (Wei and Zou, 2019), and thesaurus data augmentation had been adopted for machine translation and spoken language understanding (Zhang et al., 2015; Ma et al., 2016). For speech recognition task, class-invariant audio wave manipulation was explored in (Ko et al., 2015). Not all transformation techniques are guaranteed to generate realistic samples, and in some cases, unwarranted noises might get introduced. For example, random insertions and random swaps in (Wei and Zou, 2019) could generate ungrammatical sentences; however, some studies find that obfuscation of the decision boundary prevents overfitting, enhancing the resulting classifier model to be more robust towards real-world data.

Semi-supervised Data Augmentation

The knowledge source for augmentation could be partially external, with the rest of the knowledge filled-in without supervision (Figure 2.7). The act of augmenting input features from external unannotated data sources, such as

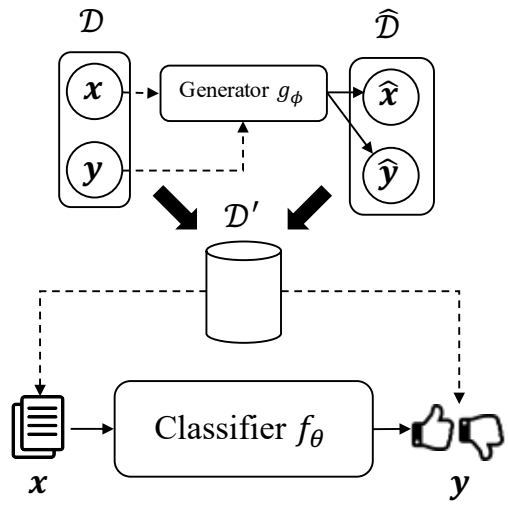


Figure 2.8: Zero-shot generative data augmentation.

large-scale text corpora, and using a classifier model trained from the original dataset to annotate and filter the external data source can be seen as a form of semi-supervised data augmentation (Ragni et al., 2014). Some of the notable early works in this line of approach (Zhu et al., 2003) exploits knowlede about the class prior to expand labeled dataset.

Unsupervised Data Augmentation

Zero-shot Data Augmentation. Unsupervised data augmentation enables the discovery of augmentational data without explicit injection of external knowledge or data sources. Typically, a generative model g_ϕ is trained on the original dataset $\mathcal{D}$ and new samples are obtained by generating plausible samples through perturbations (Figure 2.8), which we use the term *generative data augmentation* (GDA) to describe. Latent variable models with variational inference, such as VAEs, can be employed to learn the most likely distributions of the latent variable, and synthetic samples are generated by sampling from the prior of the latent variable distribution (Hu et al., 2017; Hsu et al., 2017). Due

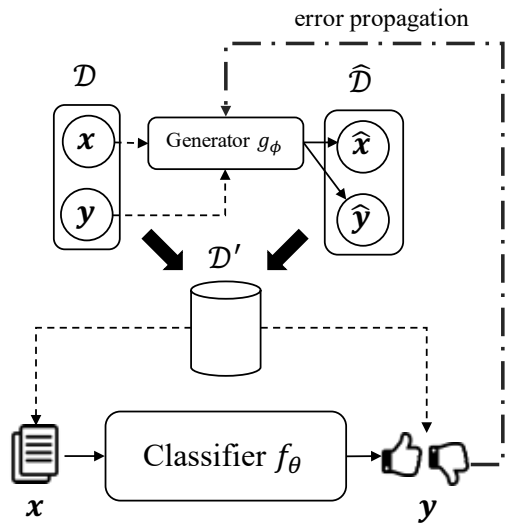


Figure 2.9: Self-supervised data augmentation.

to the robustness of deep learning models, prior works achieved a similar effect without employing variational inference (Kurata et al., 2016b; Hou et al., 2018). However, with advancements in variational deep models, we explore leveraging the expressive and explorative power of VAEs to enhance generative data augmentation in various tasks of NLP.

Self-supervised Data Augmentation. In contrast to performing zero-shot data augmentation through generative models, a recent line of work has suggested self-supervision mechanism as a way to further fine-tune the generators for the downstream classifier task. Adversarial learning combined with generators is a popular choice for achieving this effect (Tran et al., 2017; Antoniou et al., 2017). However, the main focus of our work is to explore the extent at which generative ata augmentation can be useful for various representative tasks in NLP, hence we limit the scope of the dissertation to zero-shot GDA and not incorporating adversarial learning into our models. As the extension of our work, we wish to explore whether employing task-specific fine-tuning policy further improves the performance of zero-shot generative data augmentation.

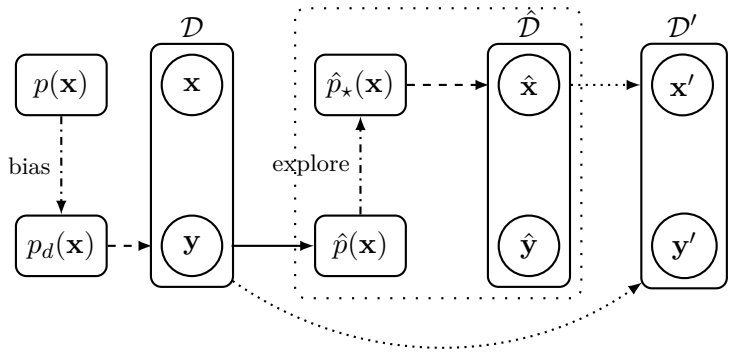


Figure 2.10: Relationships among data distributions in generative data augmentation framework.

On the other hand, some works have focused on searching the best transformational data augmentation strategy through self-learning(Cubuk et al., 2019), which could be viewed as a form of supervised data augmentation refined with self-supervision. Early works on *adversarial learning* have proposed injecting random perturbations in hidden representations of sequence-to-sequence models to improve text classification (Miyato et al., 2016), but unlike most works in generative data augmentation, these approaches do not involve generative models.

2.2.3 Theoretical Explanations

For data-hungry models, appropriate regularization is necessary to achieve high performance. Model regularizers such as dropout (Srivastava et al., 2014) and batch normalization (Ioffe and Szegedy, 2015) are widely accepted techniques to prevent model overfitting and promote robustness. Transfer learning is another regularization technique to enhance the generalization power of models that has achieved success across numerous domains and tasks (Pan and Yang, 2010).

Data augmentation (DA) can be considered an orthogonal class of regularization methods that create artificial training data to obtain better resulting

models. Most DA techniques proposed in the literature can be categorized into either *transformative* or *generative* methods. Transformative data augmentation relies on unparameterized data-transforming functions embued with external knowledge to synthesize new class-preserving data points (Dao et al., 2018). Transformative DA is widely used in the vision domain. For example, images are randomly perturbed with linear transformations (rotation, shifting etc.) to boost performances in many vision-related tasks (Simard et al., 2003; Krizhevsky et al., 2012).

On the other hand, Generative DA (GDA) exploits the generative power of latent variable models to artificially create convincing data samples. With advances in powerful generative models such as VAEs and GANs, the potential to leverage them for data augmentation has gained much attention recently. Particularly, performance gains from generated datasets have been studied and documented in the VQA task (Kafle et al., 2017), general image classification (Ratner et al., 2017), and few select SLU tasks (Kurata et al., 2016a; Hou et al., 2018). However, relevant researches are hurdled by the architectural and experimental complexities. Meanwhile, kernel theory has been suggested as a means of explaining transformational data augmentation (Dao et al., 2019). Data augmentation can also be seen as the act of calibrating the training set toward the true data distribution, which has been offset by various biases in the empirical data distribution p_d such as sampling biases.

Specifically, a general framework of generative data augmentation (GDA) is depicted in Figure 2.10. In the figure, solid arrows (—) denote training, dashed arrows (- -) denote generation, dot-dashed arrows (- · -) denote distortion, and dotted arrows (·····) denote data duplication. $\mathcal{D}'$ is the final augmented dataset for training SLU models. The goal of GDA (enclosed in loosely dotted lines) is to recover the true data distribution p through sampling, as if the samples are drawn from the corrected model distribution.

Suppose that IID samples $\mathbf{x} \in \mathcal{D}$ were intended to be sampled from a true but unknown language distribution $p(\mathbf{x}) \in \mathcal{P}$, where $\mathcal{P}$ is the probability function space for $\mathbf{x}$. However, in real world cases, the actual distribution represented by the $\mathcal{D}_w$ could be distorted due to biases introduced during erroneous data collection process or due to under-sampling variance (Torralba and Efros, 2011). Let such distortion be a function $\omega_b \in \Omega : \mathcal{P} \to \mathcal{P}$. The distorted data distribution $p^\star = \omega_b(p)$ diverges from the true distribution p, i.e. $d(p^\star, p) > 0$ where d is some statistical distance measure such as KL-divergence.

An ideal GDA counteracts the bias-introducing function ω_b and unearths the true distribution p through unsupervised explorative sampling. Suppose that a joint language understanding model $\hat{p}(\mathbf{x})$ is trained on $\mathbf{x} \sim p^\star(\mathbf{x})$. Without the loss of generality, suppose that the model is expressive enough to perfectly capture the underlying distribution, i.e. $\hat{p} = p^\star$. We collect m samples $\hat{\mathcal{D}} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_m\}$ drawn from $\hat{p}(\mathbf{x})$ and combine them with the original dataset $\mathcal{D}$ to form an augmented dataset $\mathcal{D}'$ of size $n + m$. Naïve DA will not yield better SLU results as synthetic data samples $\hat{\mathbf{x}}$ follow the distorted data distribution $p^\star$ in the best case. However, an ideal explorative sampling method and latent variable model's internal assumption about the data distribution could distort the sampling distribution, as if $\hat{\mathbf{x}}$ were sampled from another distribution $\hat{p}^\star$, such that the new distribution is closer to the true distribution (i.e. $d(\hat{p}^\star, p) < d(\hat{p}, p)$). There exists a distortion function ω_d such that $\hat{p}^\star = \omega_d(\hat{p})$. The ideal sampling method and the regularizing effect of distribution assumptions can be seen as a corrective function ω_d that undos the effect of ω_b. In this dissertation, we propose and investigate different sampling methods ω_d for the maximal DA effect. These methods are described in model description sections. The implementation details are covered in the experiments sections.

2.3 Summary

In this chapter, we have covered the background knowledge for deep latent variable models and different approaches for data augmentation related to the ideas proposed in this dissertations. Prior work has explored ways to construct complex variational latent variable models, while some other work has proposed various techniques to prevent posterior collapse that accompany during the training of such models. We have also provided a comprehensive survey on the current approaches to data augmentation and offered a fresh perspective on different takes on the technique. To the best of our knowledge, this dissertation is the first to formalize the notion of zero-shot generative data augmentation, not to mention in the context of NLP. In the subsequent chapters, we begin our journey in exploring zero-shot data augmentation for various NLP tasks, proposing various generative models and achieving meaningful results for sentence classification (Chapter 3), spoken language understanding (Chapter 4), and dialogue state tracking (Chapter 5).

Chapter 3

Basic Task: Text Classification

Text classification is one the fundemental tasks in the NLP domain. Numerous NLP tasks such as natural language understanding and sentiment analysis incorporates text classification to achieve the higher-level goal of the respective tasks. In this chapter, we explore generative data augmentation in a simple supervised learning setting where the goal is to classify a string of text into one of the predetermined number of classes. This simple and tractable setting allows us to perform various analysis on the underlying mechanism of generative data augmentation and the qualities of the generative model itself. This chapter will also serve as an introductory material for readers to familiarize with the typical generative data augmentation pipeline, where it starts out with architecture ideation of the generative model, followed by a definition of the downstream classifier/discriminator models. We will describe the training procedure of the generative model if necessary and also outline the datasets on which we test generative data augmentation. Experiments will always consist of a set of main data augmentation experiments, followed up by a series of ablation and qualitative studies that incrementally shed light on the properties of generative data augmentation.

3.1 Introduction

Variational autoencoders (VAE), on which our generative models proposed in this dissertation are based, are powerful latent variable models married with

variational inference that offer flexibility and expressiveness to model and generate text data. The NLP community has greatly benefitted from the advancement of VAEs, employing variational generative models for single sentence generation (Bowman et al., 2016; Semeniuta et al., 2017; Yang et al., 2017), paraphrase generation (Gupta et al., 2018), sentence matching (Xie and Ma, 2019; Choi et al., 2019), and dialog generation (Zhao et al., 2018; Shen et al., 2017b; Gu et al., 2018). Despite achieving great achievement in various NLP tasks, VAEs are still challenging to implement due to their sensitvity towards hyperparameter choices. The tendency of VAE models lose stability and spiral into local optima, also known as the posterior collapse phenomenon, is a well-known issue and has been studied extensively by various works (Bowman et al., 2016; Hu et al., 2017; Semeniuta et al., 2017). The phenomenon is more prominent when dealing with conditional or recurrent VAEs, which are requried for modeling joint distribution of text and some other annotation such as sentence classes and word-level sequence tags. In this chapter, in addition to presenting the standard generative data augmentation for text classification, we also intend to unearth and briefly elucidate the inner workings of VAE training and lay the foundation for hazards of developing complex VAE-based models in subsequent chapters.

Previous works have proposed weakening the decoders' expressive power (Semeniuta et al., 2017), intentionally impairing autoregressive training of the decoders (Bowman et al., 2016; Serban et al., 2017; Park et al., 2018), adjusting the KL-divergence regularizer constraint (Higgins et al.; Bowman et al., 2016; Razavi et al., 2019), and employing specific measures to empower the encoder training (He et al., 2019; Kim et al., 2018b). However, some of the measures are insufficient to reduce the hyperparameter search space to a manageable level and some others incur significant computational costs (Higgins et al.; Kim et al., 2018b). We propose a simple but effective training policy to mitigate

posterior collapse. The details about the intuition of our method is described in Appendix A.

With the effective training policy in hand, we devise a relatively simple conditional VAE structure for modeling sentences and their annotations. We show that the generative model synthesizes useful samples that improves the baseline sentence classifier on several text datasets. Qualitative analysis is also conducted to analyze the properties of the generative model itself.

In summary, the contribution of this chapter is as follows: using the novel VAE training technique, we realize two candidate conditional VAE models for modeling sentence classification datasets. We conduct statistically reliable experiments to preliminarily show that well-trained generative models can be used to boost the performance of the downstream classifiers. We also conduct comparative studies to study the difference in behavior betweeen the two candidate models. Finally, we demonstrate how conditional generative models can be not only be used for generative data augmentation but for other generation tasks such as paraphrasing and style transfer.

The reset of the chapter is structured as follows. In Section 3.2, we introduce two simple VAE-based models that can learn the distribution of language and the associated single label effectively. In this same section we present the simple training policy for mitigating posterior collapse. In Section 3.3, by utilizing the algorithm we proposed in this section, we conduct data augmentation experiments on various datasets to show that our VAE models is able to boost the performance of baseline classification models. We also present some interesting results of our qualitative assessment of the generated samples. We conclude the section with a summary and after-thoughts about the discoveries.

(a) LS-VAE (b) SL-VAE

Figure 3.1: Graphical representations of LS-VAE and SL-VAE

3.2 Our Approach

In this section, we propose a data augmentation approach for sentence classification tasks based on latent variable models.

3.2.1 Proposed Models

Given a pair of sentence and label variables $\mathbf{x} = \left(\mathbf{x}^{(s)}, \mathbf{x}^{(l)}\right)$, the goal of a latent variable model is to effectively model the joint distribution of the two data variables. There are two possible design choices depending on the point of view on the relationship between the two latent variables $\mathbf{z}^{(s)}$ and $\mathbf{z}^{(l)}$, as shown in Figure 3.1. The first point of view assumes that the label has precedence over the sentence: $p_\theta\left(\mathbf{z}^{(s)}, \mathbf{z}^{(l)}\right) = p_\theta\left(\mathbf{z}^{(s)} \mid \mathbf{z}^{(l)}\right) p_\theta\left(\mathbf{z}^{(l)}\right)$ (Figure 3.1a). The second point of view assumes that the sentence determines the label: $p_\theta\left(\mathbf{z}^{(s)}, \mathbf{z}^{(l)}\right) = p_\theta\left(\mathbf{z}^{(l)} \mid \mathbf{z}^{(s)}\right) p_\theta\left(\mathbf{z}^{(s)}\right)$ (Figure 3.1b), in which $p_\theta\left(\mathbf{z}^{(l)} \mid \mathbf{z}^{(s)}\right)$ can be considered an implicit classifier that overlaps with the baseline classification model. Both are theoretically plausible, thus the only effective way to compare them is to conduct downstream data augmentation for both models and analyze the results. More details about the model comparison experiments is discussed in subsequent sections.

We realize LS-VAE and SL-VAE as amortized hierarchical VAEs, where we apply conditional VAE to the dependent variable. The objective of LS-VAE as

follows:

$$\mathcal{L}_{\text{LS}} = \mathbb{E}_{\mathbf{z}\sim q_\phi}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - D_{\text{KL}}\Big(q_\phi\Big(\mathbf{z}^{(l)} \mid \mathbf{x}^{(l)}\Big)\Big\|p\left(\mathbf{z}^{(l)}\right)\Big) \\ - D_{\text{KL}}\Big(q_\phi\Big(\mathbf{z}^{(s)} \mid \mathbf{x}^{(s)}, \mathbf{z}^{(l)}\Big)\Big\|p_\theta\Big(\mathbf{z}^{(s)} \mid \mathbf{z}^{(l)}\Big)\Big) \tag{3.1}$$

where $p_\theta\Big(\mathbf{z}^{(s)} \mid \mathbf{z}^{(l)}\Big)$ is the conditional prior of the latent sentence variable, parameterized by θ. The conditional prior is learned from the data using standard feedforward neural networks.

Similarly, the objective of SL-VAE is as follows:

$$\mathcal{L}_{\text{SL}} = \mathbb{E}_{\mathbf{z}\sim q_\phi}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - D_{\text{KL}}\Big(q_\phi\Big(\mathbf{z}^{(s)} \mid \mathbf{x}^{(s)}\Big)\Big\|p\left(\mathbf{z}^{(s)}\right)\Big) \\ - D_{\text{KL}}\Big(q_\phi\Big(\mathbf{z}^{(l)} \mid \mathbf{x}^{(l)}, \mathbf{z}^{(s)}\Big)\Big\|p_\theta\Big(\mathbf{z}^{(l)} \mid \mathbf{z}^{(s)}\Big)\Big) \tag{3.2}$$

The implementation consists of the sentence encoder $\text{ENC}^{(s)}$, the sentence decoder $\text{ENC}^{(l)}$ (not to be confused with the approximate posterior) encodes the respective variables into hidden representations $\mathbf{h}^{(s)}$ and $\mathbf{h}^{(l)}$.

3.2.2 Training with I-VAE

Autoregressive VAEs are prone to posterior collapse (Appendix A), whose risk of occurrence can be minimized by employing word dropouts (Bowman et al., 2016). However, dropping autoregressive signals could cost deterioration of the decoder performance. We propose an early-stopping technique, called I-VAE, to mitigate posterior collapse without sacrificing the decoding power. I-VAE exploits the fact that inference collapse occurs at the final phase of VAE training, after the KL-divergence term of the ELBO (Equation 2.3) is optimized and can't be further decreased without forgoing the encoder's performance, which can be measured using the Monte Carlo estimation of the mutual information

Method	MR			SUBJ		
	NLL	**MI**	**KLD**	**NLL**	**MI**	**KLD**
Vanilla VAE	239.93	0.69	0.00	277.41	0.69	0.00
β-**VAE**						
$\beta = 0.1$,anneal=10	240.48	0.69	0.00	275.86	0.69	0.00
$\beta = 0.1$,anneal=40	218.89	3.52	4.18	257.68	2.90	3.08
$\beta = 0.1$,anneal=100	215.74	4.68	7.56	258.69	4.81	9.84
$\beta = 0.2$,anneal=10	239.93	0.91	0.00	276.79	0.80	0.00
$\beta = 0.2$,anneal=40	240.46	0.70	0.00	274.52	0.77	0.00
$\beta = 0.2$,anneal=100	222.95	0.87	0.65	259.04	0.73	0.63
$\beta = 0.5$,anneal=10	241.34	0.69	0.00	276.15	0.69	0.00
$\beta = 0.5$,anneal=40	237.88	0.69	0.00	277.15	0.69	0.00
$\beta = 0.5$,anneal=100	240.11	0.70	0.00	273.51	0.69	0.00
I-VAE						
$\alpha = 500, \beta = 1e-3$	**122.84**	4.76	8.33	**132.60**	4.71	7.37
$\alpha = 1000, \beta = 1e-3$	176.04	**4.80**	11.65	187.22	**4.82**	9.93
$\alpha = 2000, \beta = 1e-3$	222.28	**4.80**	12.99	227.35	**4.82**	12.14

Table 3.1: Training results of LS-VAE on the MR and SUBJ datasets.

between $\mathbf{x}$ and $\mathbf{z}$ (Cremer et al., 2018; He et al., 2019). When the mutual information level drops below a certain threshold, the I-VAE algorithm terminates the VAE training. The intuition is similar to (He et al., 2019), but we argue that dedicated training of the encoder is unnecessary, as the encoder might not be lagging depending on the nature of the data and the structure of the VAE. The details of our approach is described in Appendix A.

In this subsection, we explore the effectiveness of our VAE training algorithm on LS-VAE and SL-VAE. We measure the negative log-likelihood

of predicted sentences, the conditional mutual information between $\mathbf{x}$ and $\mathbf{z}^{(s)}$ under the approximate posterior, and the conditional KL-divergence term $D_{\mathrm{KL}}\Big(q_\phi\Big(\mathbf{z}^{(s)} \mid \mathbf{x}, \mathbf{z}^{(l)}\Big)\Big\|p_\theta\Big(\mathbf{z}^{(s)} \mid \mathbf{z}^{(l)}\Big)\Big)$ on the validation set. We do not report the measurements on the latent label variable, as it is trivial to learn the posterior. The results are shown in Table 3.1.

We compare our training algorithm with β-VAE (Burgess et al., 2018), where the KL-divergence term is weighted by a hyperparameter β to control the regularization of disentangled representation learning. We also apply KL-divergence annealing to β-VAE as well. Of all the usual choices of 9 hyperparameter combinations in β-VAE, only two of the cases (22%) showed signs of healthy training and the rest of them experienced posterior collapse (MI ≈ 0, KLD ≈ 0). On the other hand, models trained using I-VAE stayed healthy for an arbitrary choice of α, the KL-divergence annealing rate. NLL results of I-VAE were much lower than that of β-VAEs, while the conditional mutual information was higher than even the best peforming β-VAEs (4.80>4.68). This is achieved all while KLD was maintained at a healthy level, signifying that the I-VAEs models generate more realistic sentences while more efficiently encoding them into the prior belief.

3.3 Experiments

In this section, we describe our experiments and present their results to show that (1) our proposed solution for mitigating posterior collapse works for training generators in GDA, (2) the proposed VAE model for learning sentence-label datasets is effective at generating novel and plausible sentence-label pairs, and (3) to analyze the effects of various hyperparameter choices on the generation capability of the model.

Dataset	N	N_{train}	N_{valid}	N_{test}	K	$\mu\Big(\lvert\mathbf{x}^{(s)}\rvert\Big)$
MR	10,662	8,530	1,066	1,066	2	20.32
CR	3,770	3,016	377	377	2	18.58
SUBJ	10,000	8,000	1,000	1,000	2	23.13
MPQA	10,603	8,603	1,000	1,000	2	3.06

Table 3.2: Dataset statistics for sentence classification tasks.

3.3.1 Datasets

To validate our methodology in the sentence classification task, we choose the following four datasets as our testbed. The statistical summary of the datasets is shown in Table 3.2.

- **MR** (Pang and Lee, 2005): The movie review dataset is a collection of short sentences annotated with binary sentiment (`positive`, `negative`) crawled from Rotten Tomatoes, a movie review aggregation website (Pang and Lee, 2005). The domain of the movies is unrestricted.

- **CR** (Hu and Liu, 2006): This dataset contains a smaller number of single sentences crawled from online product reviews (Hu and Liu, 2006). The dataset contains binary sentiment annotations (`positive`, `negative`), and the range of the products is limited to digital cameras, DVD players, mp3 player and cellphones.

- **SUBJ** (Pang and Lee, 2004): This dataset contains a mixture of subjective and objective single short sentences (Pang and Lee, 2004) annotated with subjectivity tags. The subjective sentences were crawled from the review section of Rotten Tomoatoes and the objective sentences were crawled from plot summaries on the same website.

- **MPQA** (Wiebe et al., 2005): The MPQA dataset is a set of general-

domain corpora with fine-grained sentiment annotations for opinion mining (Wiebe et al., 2005). We only use human-annotated phrases for this task, hence the average length of the sentences or phrases is much shorter than that of the other three datasets (Table 3.2).

All datasets have been randomly split into training, validation and test sets while preserving the class distribution in the dataset. Compared to performing 10-split cross-validation as in (Conneau and Kiela, 2018), pre-determined splits enables us to perform large number of trials of data augmentation experiments in a controlled and feasible manner for the test of statistical significance.

The datasets have been carefully chosen to diversify domains (CR - shopping reviews, MPQA - general purpose), sentence lengths (SUBJ - 23.13, MPQA - 3.06) and target prediction schemes (sentiment polarity and subjectivity). All of the datasets have been studied extensively in the past (Zhao et al., 2015; Conneau and Kiela, 2018) for the sentence classification and sentiment analysis tasks, hence they provide sturdy ground for comparing our results with previous methods.

3.3.2 Experimental Settings

In this section, we describe the protocol for conducting GDA experiments for the sentence classification task. The following protocol pertains the experimental procedure for each hyperparameter setting of training a generative model and obtaining samples from it.

1. Given three dataset splits $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{valid}}, \mathcal{D}_{\text{test}}$, we train a generative model $\mathcal{G}$ on the training set $\mathcal{D}_{\text{train}} = \left\{\left(\mathbf{x}_1^{(s)}, \mathbf{x}_1^{(l)}\right), \ldots, \left(\mathbf{x}_n^{(s)}, \mathbf{x}_n^{(l)}\right)\right\}$, while validating the generative model against $\mathcal{D}_{\text{valid}}$ for our algorithm.

2. We use $\mathcal{G}$ to generate N_G sets of synthetic data samples $\mathcal{D}'_1, \ldots, \mathcal{D}'_{N_G}$ using different seeds. We employ ancestral sampling or approximate posterior

sampling for this experiment[1]. We use $\mathcal{D}_{\text{train}}$ as the empirical distribution of the dataset from which we sample $\mathbf{x}$. During generation, we apply the generation scaling factor γ to control how explorative we want our generator to be. Specifically, in our approaches, the approximate posterior distribution for the sentence latent variable $\mathbf{z}^{(s)}$ is modeled after a parameterized Gaussian distribution:

$$q_\phi\Big(\mathbf{z}^{(s)} \mid \mathbf{x}\Big) = \mathcal{N}(\mu_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x})I)$$

Latent variables $\mathbf{z}^{(s)}$ are then sampled from a non-biased scaled variant of the distribution $\mathcal{N}(\mu_\phi(\mathbf{x}), \gamma\sigma_\phi(\mathbf{x})I)$.

3. For each synthetic dataset $\mathcal{D}_i{}'$, we combine it with the training set $\mathcal{D}_{\text{train}}$ to form an augmented dataset $\mathcal{D}_i''$. We train N_C classifiers $\mathcal{C}$ on $\mathcal{D}_i''$ while validating against $\mathcal{D}_{\text{valid}}$. The final classifier model is tested against $\mathcal{D}_{\text{test}}$ and evaluation results are obtained for each trial. $N_G \times N_C$ classification results are aggregated, and we conduct statistical significance test of difference between current results and the baseline classification results, in which the classifiers are trained on the training set $\mathcal{D}_{\text{train}}$ only.

Overall, (1) we train a single generator $\mathcal{G}$, (2) sample N_G sets of different synthetic datasets from $\mathcal{G}$, (3) train N_C classifier $\mathcal{C}$ for each augmented dataset and aggregate the results. Multiple runs are conducted to ensure that the results account for variances in various steps of the pipeline. Figure 3.2 summarizes the experimental protocol.

3.3.3 Implementation Details

The general architecture of our VAE models is as follows. The sentence encoder is implemented using a stacked bidirectional LSTM cells of 2-layers with hidden

[1]We explore different sampling strategies in Chapter 4.

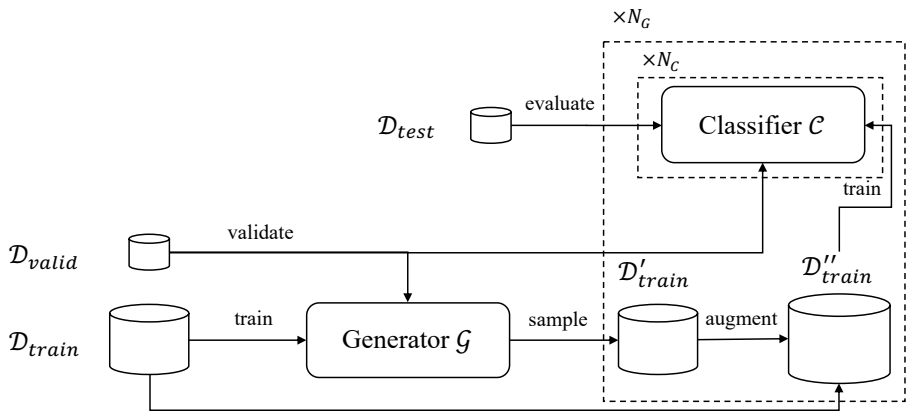


Figure 3.2: Experimental protocol for generative data augmentation in sentence classification.

dimensions of 1024 and dropout probability of 0.2 (Srivastava et al., 2014). We initialize the word embeddings with 840B 300d GloVe vectors (Pennington et al., 2014) for both the generator and the classifier. The sentence deocder is a single LSTM cell with hidden dimensions of 1024. The label encoder, decoder and Gaussian predictors (for approximate posterior inference) are implemented using 2-layer feedforward networks with dropout probability of 0.2 and are applied with batch normalization (Ioffe and Szegedy, 2015). We use Gaussian reparameterization trick (Kingma and Welling, 2013). Adam optimizer (Kingma and Ba, 2014) is used to optimize all networks with the initial learning rate of 0.001. Mini-batch size is 128 for both the generator and the classifier. Models were implemented on the PyTorch framework. Experiments were carried on a mixture of Nvidia GTX1080 and Titan Xp GPUs. During the training of SL-VAE models, we apply word dropout rate of 0.5 to the decoder during teacher-forcing training (Bowman et al., 2016). Similar to the intuition of word dropout, we also apply label dropout rate of 0.5 on the computation of the label posterior to discourage the label decoder from relying on the teacher-forced information.

We choose BiLSTM-Max (Figure 4.2) as our baseline classifier architecture

Method	Accuracy			
	MR	**CR**	**SUBJ**	**MPQA**
Baseline (w/o GDA)	76.62	78.54	92.12	85.85
LS-VAE				
$\alpha = 1000, \beta = 1e-3$	76.44	77.62	92.47	86.38
$\alpha = 2000, \beta = 3e-4$	**77.21**	**78.91**	**92.50**	**86.38**
SL-VAE				
$\alpha = 1000, \beta = 1e-3$	76.78	76.67	91.61	85.44
$\alpha = 2000, \beta = 3e-4$	76.85	77.54	91.72	85.46

Table 3.3: Comparison of GDA results between two candidate models for sentence-label pair datasets.

for its simplicity and effectiveness. The model is known to perform reasonably well for many sentence classification and understanding tasks (Conneau and Kiela, 2018). We perform standard machine learning protocol for training classifiers: a classifier $\mathcal{C}$ is trained on a training set $\mathcal{D}_{\text{train}}$, while the training progress is monitored by validating the model against the validation set $\mathcal{D}_{\text{valid}}$ every epoch. The best model is chosen based on the validation performance and evaluated using the test set $\mathcal{D}_{\text{test}}$. We employ the early stopping strategy where the patience is 10 training epochs and the maximum training steps is 800,000, which is reasonably tolerant for all datasets.

3.3.4 Data Augmentation Results

In this subsection, we conduct various GDA experiments to compare different strategies to maximize the benefits of data augmentation. The first experiment compares the two modeling approaches proposed in this chapter for sentence-label pair generation (SL-VAE and LS-VAE). We compare the performance

Dataset	BiLSTM-Max				
	Accuracy	pos/subj		neg/obj	
		Precision	Recall	Precision	Recall
MR	76.62 ± 1.57	73.79 ± 3.43	**80.10** ± 3.25	**80.15** ± 1.55	73.42 ± 5.48
MR+	**77.21** ± 1.36‡	**76.69** ± 3.62‡	76.03 ± 4.79‡	78.22 ± 2.49‡	**78.30** ± 5.50‡
CR	78.54 ± 1.06	78.23 ± 1.42	**89.03** ± 1.35	**79.33** ± 1.59	62.85 ± 3.40
CR+	**78.91** ± 0.23†	**78.98** ± 1.59†	88.50 ± 2.67†	79.23 ± 2.77†	**64.57** ± 4.45†
SUBJ	92.12 ± 0.79	92.40 ± 1.62	92.35 ± 1.99	91.94 ± 1.87	91.88 ± 2.00
SUBJ+	**92.50** ± 0.57†	**92.90** ± 0.83†	**92.52** ± 1.07†	**92.11** ± 1.00†	**92.47** ± 0.97†
MPQA	85.85 ± 1.22	83.27 ± 5.64	**70.09** ± 4.72	**87.15** ± 1.42	93.13 ± 3.42
MPQA+	**86.38** ± 0.51†	**84.84** ± 2.79†	69.48 ± 2.90†	87.00 ± 0.93†	**94.18** ± 1.51†

† $p < 0.1$ ‡ $p < 0.01$

Table 3.4: Data augmentation results on sentence classification tasks using LS-VAE.

of classifiers trained on datasets that are augmented by samples from SL-VAE and those that are augmented by LS-VAE (Table 3.3). Different hyperparameter settings were tested on the two models, and we found that LS-VAE performed marginally better for all datasets on average. We conjecture that the higher representation capacity possessed by LS-VAE allows it to have a slight edge over SL-VAE in GDA.

We present the effect of augmenting datasets using the generative latent variable model in the sentence classification task (Table 3.4). We use BiLSTM-Max (Figure 4.2) as the baseline classifier odel, keeping the hyperparameter settings constant across all experiments. We report classifier performances through the accuracy, along with the precision and the recall of each of the binary labels (`subj`/`obj` for the SUBJ dataset and `pos`/`neg` for others). The results are aggregated (mean and standard deviation) after being run multiple times as described in Section 3.3.2. We use + to denote the corresponding datasets that have been augmented with our proposed model for modeling sentence annotation datasets, LS-VAE.

Label	$\|\mathbf{V}\|$	$H(\mathbf{w})$	**Perplexity**
`pos`	11,100	6.835	930.2
`neg`	11,325	6.856	949.5

Table 3.5: 1-gram statistics of the MR dataset.

The results on the four datasets show that the generative data augmentation technique is beneficial, albeit with some varying results among the tasks. We observe three major improvements. First, the overall mean accuracy improved in all of the tasks with statistical significance $p < 0.1$, except for the CR dataset. We conjecture that the marginal improvement in the CR dataset is due to sparseness, causing the generative data augmentation to exacerbate the empirical bias in the dataset. Second, the classification performances stabilize with the data-augmented datasets, exhibiting less variance than the classifiers trained on the non-augmented datasets. In all datasets, the variance of the classifier accuracies have decreased when trained on the augmented datasets, with the smallest variance observed in the augmented CR dataset (0.23). The improvement in both bias and variance indicates that the classifiers are trained more optimally when they are subject to data-side regularization. Third, the detailed analysis of the per-label precision and recall scores shows that data regularization essentially normalizes the performance across all labels. For example, the MR dataset without data augmentation has a bias towards the postiive-sentiment samples. The bias comes from the asymmetry of the language distribution between the two class labels: the word-level perplexity of `neg`-labeled sentences is higher than that of `pos`-labeled sentences (Table 3.5). Having applied generative data augmentation, the precision and recall scores across all labels have normalized, improving the overall performance of the resulting classification models.

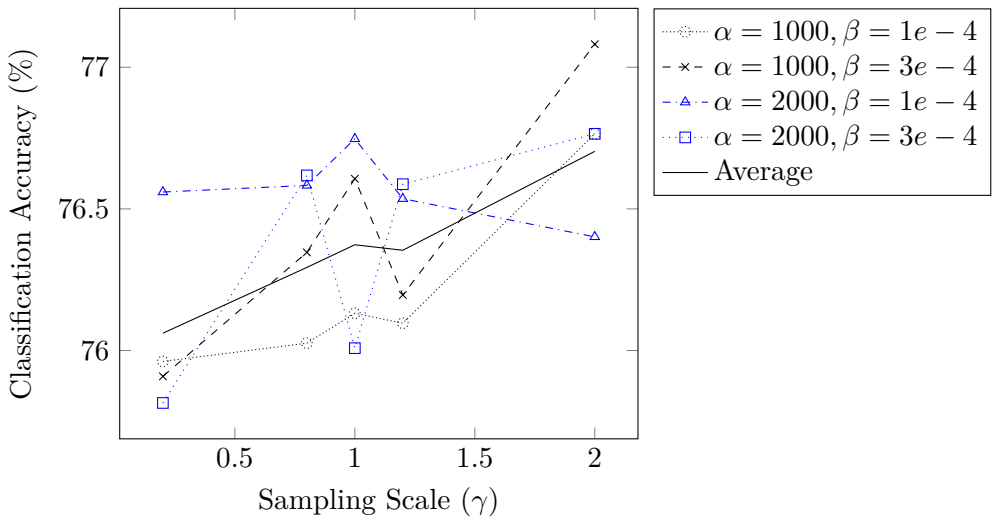


Figure 3.3: GDA results with varying hyperparameter choices for our algorithm

3.3.5 Ablation Studies

In this section, we explore the effects of different choices of hyperparameter settings have on the GDA results.

We vary the KL-divergence annealing period (α), the tolerance margin (β), and the posterior sampling scale (γ) by small margins and plot the GDA results in Figure 3.3. First, we observe that the sampling scale correlates with the data-augmented classifier performance, except for an edge case where $\alpha = 2000, \beta = 1e-4$. This is likely due to the high variance nature of the generative data augmentation process. Despite some irregularities in results, the plot of the average of each scale factor (solid line) shows that the two factors are postiively correlated (Pearson's correlation coefficient 0.959, $p < 0.01$). This finding supports our hypothesis that the positive effect on the performance of the classifier is attributed to the novel samples discovered through the exploratory power of posterior sampling. As the scaling factor increases above the standard rate ($\gamma > 1.0$), the model becomes more exploratory, increasing the chance to extract novel samples. When the scaling factor is below the standard

	Anchor x	Generated Sample x′
1	i have no complaints with this player	i have no problems with this player
2	i' ve been incredibly happy with this camera	i' m really happy with this purchase
3	this camera is bulletproof	the phone is awesome
4	it fits in your pocket	it looks sleek and modern
5	this dvd player is basically junk	the dvd player is not work
6	it is made of plastic	this is ridiculous

Table 3.6: Generation samples from LS-VAE trained on the CR dataset using ancestral sampling (differences highlighted in red).

rate ($\gamma < 1.0$), the model is not able to improve the performance of the classifier above the baseline (76.6), as little novel samples are explorable within the safe bounds of the predicted variance. On the contrary, the performance drops further as we decrease the scaling factor below 1.0, which can be explained by data distribution disruption. Data samples which exist only in the space between γ and 1.0 will never be generated, causing a shift in the augmented data distribution without any compensation through the introduction of novel instances.

3.3.6 Qualitative Analysis

Three qualitative experiments are conducted to visualize the generation quality of our approach, posterior sampling, **z**-interpolation and label inversion. These experiments are designed to demonstrate various aspects of our model's generation capability, such as grammaticality, accuracy and disentangled representation learning.

Posterior Sampling

In this experiment, we examine the quality of the generated samples that are obtained from ancestral sampling on a well-trained LS-VAE. We show that the

model trained with our algorithm applies meaning-preserving transformation that are learned from the dataset without explicit supervision. We find several nonlinear transformation strategies that the model has learned to adopt. The posterior sampling is conducted as follows.

Specifically, we choose an anchor data point from the dataset distribution: $\mathbf{x} \sim p_d(\mathbf{x})$, where p_d is the empirical distribution. Then, we sample a set of latent variable $\mathbf{z}$ from the encoded distribution of $\mathbf{x}$: $\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})$. We then obtain a generated sample $\mathbf{x}'$ that is most probable in the model posterior:

$$\mathbf{x}' = \arg\max_{\mathbf{x}} p_\theta(\mathbf{x} \mid \mathbf{z}) \tag{3.3}$$

Table 3.6 shows the sampling results. The results show that in terms of generation quality, the VAE is able to generate grammatically accurate samples. However, there were cases, such as the fifth example, that introduced subtle grammatical errors in the synthetic samples, but such noises had limited influence in classifier performances as seen in the data augmentation experients (Table 3.4).

Few observations are made from the samples in Table 3.6.

1. The VAE model has effectively learned to perform *word-level substitution* without much variation at the sentence-level semantics. As seen in the first example, the model has implicitly learned the similarity of words with overlapping contextual semantics, such as `complaint` and `problem`, allowing them to be generated interchangably.

2. The model has also learned to perform *phrase-level substitution*, allowing more complex phrases to be paraphrased, as evident in examples 2 and 4.

3. The model has learned to exhibit certain level of awareness in *higher-level compositionality*, such as the understanding of logical implication required

$^n/_N$	**Decoded Sentence**	**Decoded Label**
0%	a film of quiet delicacy and squalor	`pos`
20%	a triumph of astonishing delicacy and force	`pos`
40%	a sharp and engrossing character study	`pos`
60%	an intimate and a resonant work	`pos`
80%	smart edges and an engaging and a film	`pos`
100%	smart , fashioned , and a film that delivers a feast	`pos`

Table 3.7: **z**-interpolation between two data samples of the same label (pos), using LS-VAE trained on the MR dataset.

to generate paraphrases in example 4. The combination of these characteristics shown by the VAE model has contributed to the performance-boosting effect in generative data augmentation.

z-interpolation

Visualizing decoder samples from a linear interpolation of two points in the **z**-space is a popular method for showcasing the successful training of VAEs (Bowman et al., 2016). Given two data sample $\mathbf{x}_1$ and $\mathbf{x}_2$, we map the data points onto the latent variable space using the encoder q_ϕ to obtain $\mathbf{z}_1$ and $\mathbf{z}_2$. Multiple equidistant samples $\mathbf{z}'_1, ..., \mathbf{z}'_N$ are selected from the linear interpolation between the two points: $\mathbf{z}'_n = \mathbf{z}_1 + n\frac{\mathbf{z}_2 - \mathbf{z}_1}{N}$. Likelihood-maximizing samples $\mathbf{x}'_1, \ldots, \mathbf{x}'_N$ are chosen from the model posteriors given the $\mathbf{z}$ samples (Equation 3.3).

We showcase two interpolation results in Table 3.7 and Table 3.8. In the first experiment, we fix the sentiment of the sentences to `positive` such that the gradual differences between the two samples are limited to linguistic features. Results in Table 3.7 show that our model has effectively learned to encode and decode various levels of linguistic features into the latent variable space. Sub-

$^n/_N$	**Decoded Sentence**	**Decoded Label**
0%	the film is uniformly good	`pos`
20%	the movie is uniformly good	`pos`
40%	the film is worth seeing	`pos`
60%	this movie a lot	`neg`
80%	i hated every movie	`neg`
100%	i hated every minute of the film	`neg`

Table 3.8: $\mathbf{z}$-interpolation between two data samples of opposite labels using LS-VAE trained on the MR dataset.

stitutions of words with similar contextual meanings are observed between two adjacent steps. However, as we further move along the interpolation lineage, larger differences in syntactic and semantic structures of the sentences are observed. These observations suggest that the model is capable of generalizing different levels of linguistic variations into the latent space $\mathbf{z}^{(s)}$.

Label Inversion

We show that our model is able to disentangle linguistic representations into components that are relevant to the annotation and those that are not, enabling the model to perform bidirectional *style transfer* between the annotation classes. For this experiment, we encode a sentence and a label just as we did for the posterior sampling experiments, except that the encoded label is the complement of the original corresponding label. We decode a sample from the encoded $\mathbf{z}$ and compare it with the original sentence to observe any changes. If our model learned to represent intended information in the respective latent variables $\mathbf{z}^{(s)}$ and $\mathbf{z}^{(l)}$, only the linguistic features should be encoded into $\mathbf{z}^{(s)}$ and the annotation-related features (sentiment etc.) should be encoded into $\mathbf{z}^{(l)}$ and therefore, ideally, the decoded sample from aforementioned $\mathbf{z}$ should

	Original Sentence $\mathbf{x}^{(s)}$	$\mathbf{x}^{(s)}$	**Inverted Sentence $\mathbf{x}^{(s)\prime}$**	$\neg\mathbf{x}^{(l)}$
1	what a great film	`pos`	nothing about it fits	`neg`
2	a warm, funny, engaging film	`pos`	a bland, reluctant, thumbs down	`neg`
3	too silly to take seriously	`neg`	very very funny	`pos`
4	little more than a well mounted history lesson	`neg`	worth catching as providing some old fashioned spooks	`pos`

Table 3.9: Label-inverted samples from LS-VAE trained on the MR dataset

be a sentence with a similar syntactic structure with the opposite polarity, for example.

Specifically, let $\mathbf{x} = \left(\mathbf{x}^{(s)}, \mathbf{x}^{(l)}\right)$ be the data sample. We use the encoder q_ϕ to obtain the distribution of the latent variable representations of $\mathbf{x}' = \left(\mathbf{x}^{(s)}, \neg\mathbf{x}^{(l)}\right)$, where $\neg\mathbf{x}^{(l)}$ is the inverted label of $\mathbf{x}^{(l)}$: $q_\phi\left(\mathbf{z}^{(s)}, \mathbf{z}^{(l)} \mid \mathbf{x}^{(s)}, \neg\mathbf{x}^{(l)}\right)$. We take the mean of the approximate posterior distribution and decode samples from the model posterior given the mean using Equation 3.3. The results are shown in Table 3.9. The examples demonstrate that the model has learned to distinguish annotation-specific and general linguistic features that are orthogonal to each other. Those features are represented using the dedicated latent variables $\mathbf{x}^{(s)}$ and $\mathbf{x}^{(l)}$. For instance, the original syntactic structure of the original sentence *a warm, funny, engaging film* is nearly preserved in the inverted sentence *a bland, reluctant, thumbs down*, while the positive adjectives have been transformed to equivalent terms in the opposite polarity. As another example, example 4 demonstrates that the model is able to recognize sentiment-neutral concepts such as *the state of being old* from the original word *history*, as evident from the preservation of the meaning in the inverted sentence (*old fashioned spooks*).

3.4 Summary

In this chapter, we have taken a dive at the underlying mechanisms in training behaviors of VAEs and discussed the limitations of VAE training techniques in the current literature. To exploit our findings, we proposed I-VAE as an attempt to standardize VAE training.

In order to demonstrate that our training algorithm is effective, we not only conduct intrinsic evaluation, in which the training statistics of the VAE are examined, we also conduct extrinsic evaluation on downstream tasks. For the downstream tasks, we introduce the task of data-augmenting datasets for sentence classification. We also introduce LS-VAE, a architecture for modeling sentence-label pairs. Through statistically reliable experiments, we show that LS-VAE trained using I-VAE algorithm boosts the classification performance on four sentence classification tasks. We have also presented some interesting qualitative results that suggest other uses for our generative model, such as conditional generation and style transfer.

From this study we conclude that not only training VAEs using I-VAE is effective, but generative data augmentation for sentence classification is an effective data regularization strategy.

In the subsequent chapters, we explore whether generative data augmentation can be effective even for other NLP tasks, namely spoken language understanding and dialogue state tracking.

Chapter 4

Multi-task Learning: Spoken Language Understanding

In this chapter, we explore the feasibiilty of performing zero-shot data augmentation for the spoken language understanding task using an end-to-end latent variable model that takes advantages of recent advances in representation learning and text generation. As spoken language undertanding requires the joint learning of intent classification and sequence labeling, the success in applying generative data augmentation for the task demonstrates the feasibility of the data augmentation technique in multi-task settings.

4.1 Introduction

Spoken language understanding (SLU) in current literature refers to the study of models that parse spoken queries into semantic frames. Semantic frames contain pieces of semantic units that best represent the speaker's intentions and are essential for the development of human-machine interfaces, such as virtual assistants.

Scarcity of linguistic resources has been a recurring issue in many NLP tasks such as representation learning (Al-Rfou et al., 2013), neural machine translation (NMT) (Zoph et al., 2016), and SLU (Kurata et al., 2016a). The issue is especially prominent in SLU, because creating manually annotated SLU datasets is costly but the domain space that might require new labeled datasets

is near infinite.

Even for domains with existing datasets, they might suffer from the data sparsity issue, which have long been plaguing many NLP tasks that require annotated linguistic datasets (Lai et al., 2015). For example, most SLU datasets are not large enough cover all possible data label pairs. Furthermore, biased data collection methods could exacerbate the issue (Torralba and Efros, 2011).

Recent years, there have been significant advances in variational autoencoders (VAE) (Kingma and Welling, 2013) and other latent variable models for textual generation (Serban et al., 2017; Yu et al., 2017; Hu et al., 2017; Li et al., 2017), prompting investigations into the possibility of improving model performances through generative data augmentation (Kafle et al., 2017; Kurata et al., 2016a; Hou et al., 2018).

In this chapter, we propose a generative model specialized in the generation of SLU datasets. Finally, we wish to demonstrate the effectiveness of our approach through various experiments. In essence, our main contributions are two folds:

1. **A Novel Model for Labeled Language Generation**: We propose a novel generative model for jointly synthesizing spoken utterances and their semantic annotations (slot labels and intents). We show that the synthetic samples generated from the model are not only natural and accurately annotated, but they improve SLU performances by a significant margin when used in the generative data augmentation framework. We also show that our model is better than the previous work (Kurata et al., 2016a).

2. **Substantiation with Extensive Experimentation**: We substantiate the general benefits of generative data augmentation with experiments and statistical testing on various SLU models and datasets. Results show that our approach produces extremely competitive performances for exist-

ing SLU models in the ATIS dataset. Our ablation studies also bring some important insights such as the optimal synthetic dataset size to light.

4.2 Related Work

The SLU task is one of more mature research areas in NLP. Many works have focused on exploring neural architectures for the SLU task. Plain RNNs and LSTMs were first explored in (Mesnil et al., 2015; Yao et al., 2014). (Kurata et al., 2016b) proposed sequence-to-sequence (Seq2Seq) models. Hybrid models between RNNs and CRFs were explored in (Huang et al., 2015). Joint language understanding models that jointly predict slot labels and intents gained significant traction since they had been first proposed in (Guo et al., 2014; Goo et al., 2018). Some works focused on translating advances in other NLP areas to the SLU task (liu, 2016).

4.3 Model Description

In this section, we describe our generative data augmentation model and the underlying framework in detail.

4.3.1 Framework Formulation

We begin with some notations, then we formulate the overall generative data augmentation framework for the spoken language understanding task.

Notations

An utterance $\mathbf{w}$ is a sequence of words $(w_1, \ldots, w_{T_i})$, where T is the length of the utterance. For each utterance in a labeled dataset, an equally-long semantic slot sequence $\mathbf{s} = (s_1, \ldots, s_T)$ exists such that s_i annotates the corresponding

word w_i. The intent class of the utterance is denoted by y. A fully labeled language understanding dataset $\mathcal{D}$ is a collection of utterances and their respective annotations $\{(\mathbf{w}_1, \mathbf{s}_1, y_1), \ldots, (\mathbf{w}_n, \mathbf{s}_n, y_n)\}$, where n is the size of the dataset. A data sample in $\mathcal{D}$ is denoted by $\mathbf{x} = (\mathbf{w}, \mathbf{s}, y)$. The set of all utterances present in $\mathcal{D}$ is denoted by $\mathcal{D}_w = \{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$. Similarly, the set of slot label sequences and intent classes are denoted by $\mathcal{D}_s$ and $\mathcal{D}_y$.

Spoken Language Understanding

A spoken language understanding model is a discriminative model $\mathbf{S}$ fitted on labeled SLU datasets. Specifically, let ψ to denote parameters of the prediction model. Given a training sample $(\mathbf{w}, \mathbf{s}, y)$, the training objective is as follows:

$$\mathcal{L}_{LU}(\psi; \mathbf{w}, \mathbf{s}, y) = -\log p_\psi(\mathbf{s}, y|\mathbf{w}). \tag{4.1}$$

Given an utterance $\mathbf{w}$, predictions are made by finding the slot label sequence $\hat{\mathbf{s}}$ and the intent class $\hat{y}$ that maximize the loglikelihood: $(\hat{\mathbf{s}}, \hat{y}) = \arg\max_{\mathbf{s},y} \log p_\psi(\mathbf{s}, y|\mathbf{w})$. For non-joint SLU models, p_ϕ is factorizable: $p_\phi(\mathbf{s}, y|\mathbf{w}) = p_\phi(\mathbf{s}|\mathbf{w})\, p_\phi(y|\mathbf{w})$. In recent years, joint language understanding has become a popular approach, as studies show a synergetic effect of jointly training slot filling and intent identification (Guo et al., 2014; Chen et al., 2016b).

4.3.2 Joint Generative Model

In this subsection, we describe our generative model in detail. We begin with a standard VAE (Kingma and Welling, 2013) applied to utterances, then we extend the model by allowing it to generate other labels in a joint fashion.

Standard VAE

VAEs are latent variable models applied with amortized variational inference. Let θ be the parameters of the generator network (i.e. the decoder network), and let ϕ be the parameters of the recognition network (i.e. the encoder network). Specifically in the case of utterance learning, the goal is to maximize the log likelihood of sample utterances $\mathbf{w}$ in the dataset $\log p(\mathbf{w}) = \log \int p(\mathbf{w}, \mathbf{z})\, d\mathbf{z}$. However, since the marginalization is computationally intractable, we introduce a proxy network $q_\phi(\mathbf{z}|\mathbf{w})$ and subsequently minimize a training objective based on ELBO (Equation 2.3).

In Equation 2.3, the proxy distribution q_ϕ is kept close to the prior $p(\mathbf{z})$, which we assume to be the standard multivariate Gaussian. Since the KL-divergence term is always positive, $\mathcal{L}_{VAE}$ is the upper bound for the reconstruction error under the particular choice of a proxy distribution q_ϕ. The proposed generative model is based on VRAEs, in which the posterior of a sequence factorizes over sequence elements (i.e. words) based on the Markov Chain assumption: $p_\theta(\mathbf{w}|\mathbf{z}) = \prod_{i=1}^{T} p_\theta(w_i|w_1, \ldots, w_{i-1}, \mathbf{z})$. VAEs can be optimized using gradient-descent methods with the reparameterization trick (Kingma and Welling, 2013).

The Sampling Problem

Given the parameters $\theta_\mathcal{D}$ and $\phi_\mathcal{D}$ that are optimized for all $\mathbf{w} \in \mathcal{D}_w$, our goal is to sample plausible utterances $\hat{\mathbf{w}}$ from the distribution of $\mathbf{w}$ believed by the model:

$$\hat{\mathbf{w}} \sim p_{\theta_\mathcal{D}, \phi_\mathcal{D}}(\mathbf{w}) = \int p_{\theta_\mathcal{D}}(\mathbf{w}|\mathbf{z})\, p_{\theta_\mathcal{D}, \phi_\mathcal{D}}(\mathbf{z})\, d\mathbf{z} \tag{4.2}$$

As evident in Equation 4.2, the marginal likelihood estimation requires us to infer the marginal probability of the latent variable $p_{\theta_\mathcal{D}, \phi_\mathcal{D}}(\mathbf{z})$, which can be estimated by marginalizing the joint probability from the recognition network.

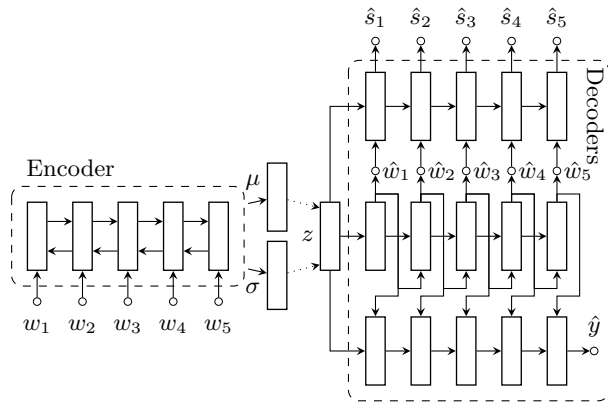


Figure 4.1: Joint language understanding variational autoencoder (JLUVA). The VAE model consists of a BiLSTM-Max encoder and three uni-directional decoders. Note that the fully connected layers and embedding layers are omitted for clarity.

$$p_{\theta_{\mathcal{D}},\phi_{\mathcal{D}}}(\mathbf{z}) = \mathbb{E}_{\mathbf{w}\sim p(\mathbf{w})}\left[q_{\phi_{\mathcal{D}}}(\mathbf{z}|\mathbf{w})\right] \tag{4.3}$$

However, Equation 4.3 cannot be solved analytically, as the true distribution of w is unknown. Hence, some form of approximation is required to sample utterances from the latent variable model. The approximation approach will likely have an impact on the quality of generated utterances, thereby determine the effect of data augmentation. Here, we describe two options.

The first is to approximate the marginal probability of the latent variable with the prior $p(\mathbf{z})$, the standard multivariate Gaussian. However, this naïve approximation will likely yield homogeneous and uninteresting utterances due to over-simplication of the latent variable space. In real world scenarios, the KLD loss term in Equation 2.3 is still large after convergence.

Alternatively, the other option is to approximate using the Monte Carlo method. Under the Monte Carlo approach (Algorithm 1), the marginal likelihood is calculated deterministically for each utterance w sampled from the

input : a sufficiently large number m

given : $\mathcal{D}_w$, θ, ϕ

output: synthetic utterance list $\mathbf{U}$

initialize $\mathbf{U}$ as an empty list;

while $\mathbf{U}$ *has less than* m *samples* **do**

  sample a real utterance $\mathbf{w}$ from $\mathcal{D}_w$;

  estimate the mean $\bar{\mathbf{z}}$ of the posterior $q_\phi(\mathbf{z}|\mathbf{w})$;

  sample $\hat{\mathbf{w}}$ from the likelihood $p_\theta(\mathbf{w}|\bar{\mathbf{z}})$;

  append $\hat{\mathbf{w}}$ to $\mathbf{U}$;

end

return $\mathbf{U}$

Algorithm 1: Monte Carlo posterior sampling.

dataset $\mathcal{D}$. According to the law of large numbers, the marginal likelihood $p_{\theta_\mathcal{D},\phi_\mathcal{D}}(\mathbf{w})$ converges to the empirical mean, thereby providing an unbiased distribution for sampling $\mathbf{w}$.

Exploratory Sampling

In our general framework for GDA, remind that the sampling method is required to be exploratory, such that the biases in datasets are counteracted. translating to better performances in resulting models. Hence, an ideal exploratory sampling approach is unbiased but has increased sampling variance. Intuitively, we can sample the latent variable $\mathbf{z}$ from the Gaussian encoded by the recognizer in place of analytically estimating the mean in Algorithm 1. Suppose that $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are mean and standard deviation vectors encoded by the recognizer. Then we sample $\mathbf{z}$ from $\mathcal{N}(\boldsymbol{\mu}(\mathbf{w}), \lambda_s \cdot \boldsymbol{\sigma}(\mathbf{w}))$, where the scaling hyperparameter λ_s controls the level of exploration exhibited by the generator. This unbiased empirical estimation of the posterior helps generate realistic but more varied utterances.

Dataset	#Splits	Train	Val	Test
ATIS-small	35	127 - 128	500	893
ATIS-medium	9	497 - 498	500	893
ATIS	1	4,478	500	893
Snips	1	13,084	700	700
MIT Movie Eng	1	8,798	977	2,443
MIT Movie Trivia	1	7,035	781	1,953
MIT Restaurant	1	6,894	766	1,521

Table 4.1: SLU Dataset statistics.

Joint Language Understanding VAE

Starting from a VAE for encoding and decoding utterances, Joint Language Understanding VAE (JLUVA) extends the model by predicting slot labels and intent classes. The generation of slot labels and intents are conditioned on the latent variable $\mathbf{z}$ and the generated utterance $\hat{\mathbf{w}}$ (Figure 4.1). The benefits of having conditional dependence on $\mathbf{z}$ during labeling is documented in (Kurata et al., 2016b). The modified training objective for the language understanding task is as follows.

$$\mathcal{L}_{LU}\left(\phi, \psi; \mathbf{w}, \mathbf{s}, y\right) = -\mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log p_\psi\left(\mathbf{s}, y|\hat{\mathbf{w}}, \mathbf{z}\right)\right] \tag{4.4}$$

The joint training objective of the entire model is specified in terms of the training objective of the VAE component (Equation 2.3) and the negative log-likelihood of the discriminatory component (Equation 4.4):

Model + Sampling Approach	Slot Filling (F1)		
	Small	Med.	Full
Baseline (No Augmentation)	72.57‡	88.28‡	95.34
Encoder-Decoder + Additive*	74.79†	89.13‡	95.20
JLUVA + Additive (Ours)	74.14‡	89.13‡	95.40
JLUVA + Standard Gaussian (Ours)	70.72‡	86.90‡	94.91‡
JLUVA + Posterior (Ours)	**74.92**	**89.27**	**95.55**

* (Kurata et al., 2016a) † $p < 0.1$ ‡ $p < 0.01$

Table 4.2: Data scarcity results of JLUVA on the ATIS dataset (Slot Filling).

Model + Sampling Approach	Intent (F1)		
	Small	Med.	Full
Baseline (No Augmentation)	82.65	90.59†	97.21
JLUVA + Additive (Ours)	83.46	**90.97**	97.04
JLUVA + Standard Gaussian (Ours)	78.67‡	86.90‡	96.90
JLUVA + Posterior (Ours)	**83.65**	90.95	**97.24**

* (Kurata et al., 2016a) † $p < 0.1$ ‡ $p < 0.01$

Table 4.3: Data scarcity results of JLUVA on the ATIS dataset (Intent Classification).

Model + Sampling Approach	Semantic Frame (Acc.)		
	Small	Med.	Full
Baseline (No Augmentation)	35.09‡	65.18‡	85.95
JLUVA + Additive (Ours)	38.58	66.75	85.81
JLUVA + Standard Gaussian (Ours)	32.46‡	61.12‡	84.62‡
JLUVA + Posterior (Ours)	**39.43**	**67.05**	**86.33**

* (Kurata et al., 2016a) † $p < 0.1$ ‡ $p < 0.01$

Table 4.4: Data scarcity results of JLUVA on the ATIS dataset (Semantic Frame).

$$\begin{aligned}\mathcal{L}(\theta, \phi, \psi; \mathbf{w}, \mathbf{s}, y) =& D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{w}) \| p_\theta(\mathbf{z}|\mathbf{w})) \\ &- \mathbb{E}_{\mathbf{z}\sim q_\phi}[\log p_\theta(\mathbf{w}|\mathbf{z})] \\ &- \mathbb{E}_{\mathbf{z}\sim q_\phi}[\log p_\psi(\mathbf{s}, y|\hat{\mathbf{w}}, \mathbf{z})]\end{aligned} \tag{4.5}$$

We obtain the optimal parameters θ^*, ϕ^*, ψ^* by minimizing Equation 4.5 (i.e. $\arg\min_{\theta,\phi,\psi} \mathcal{L}$) with respect to a real dataset $\mathcal{D}$. During the data generation process, we sample $z^\star$ from an approximated prior $p^\star(\mathbf{z})$ which depends on the approximation strategy (e.g. posterior sampling). Then we perform inference on the posterior network $p_\theta(\mathbf{w}|\mathbf{z}^\star)$ to estimate the language distribution. A synthetic utterance $\hat{\mathbf{w}}$ is sampled from said distribution and is used to infer the slot label and intent distribution from the relevant networks, i.e. $p(\mathbf{s}, y|\mathbf{z}, \hat{\mathbf{w}})$. The most probable $\hat{\mathbf{s}}$ and $\hat{y}$ are combined with $\hat{\mathbf{w}}$ to form a generated sample set $(\hat{\mathbf{w}}, \hat{\mathbf{s}}, \hat{y})$. This generation process is repeated until sufficient synthetic data samples are collected.

4.4 Experiments

In this section, we outline the design, execution, results and analysis of all experiments pertaining to testing the effectiveness of our GDA approach.

4.4.1 Datasets

In this paper, we carry out experiments on the following language understanding datasets.

- **ATIS**: Airline Travel Information System (ATIS) (Hemphill et al., 1990) is a representative dataset in the SLU task, providing well-founded comparative environment for our experiments.
- **Snips**: The snips dataset is an open source virtual-assistant corpus. The dataset contains user queries from various domains such as manipulating playlists or booking restaurants.
- **MIT Restaurant (MR)**: This single-domain dataset specializes in spoken queries related to booking restaurants.
- **MIT Movie**: The MIT movie corpus consists of two single-domain datasets: the movie eng (ME) and movie trivia (MT) datasets. While both datasets contain queries about film information, the trivia queries are more complex and specific.

All of the datasets are annotated with slot labels and intent classes except the MIT datasets. The detailed statistics of each dataset are shown in Table 4.1. In order to simulate a data scarce environment (similar to the experimental design proposed in (Chen et al., 2016b)), we randomly chunk the ATIS training set into equal-sized smaller splits. For the small dataset the training set is chunked into 35 pieces, and for the medium dataset it is chunked into 9 pieces.

The sizes of the small and medium training splits approximately correspond those mentioned in the previous work (Chen et al., 2016b).

4.4.2 Experimental Settings

Here, we describe the methodological and implementation details for testing the GDA approach under the framework.

General Experimental Flow

Since we observe a high variance in performance gains among different runs of the same generative model (e.g. Figure 4.5), we need to approach the experimental designs with a more conservative stance. The general experimental methodology is as follows.

- N_G identical generative models are trained with different initial seeds on the same training split.
- m utterance samples are drawn from each model to create N_G augmented datasets $\mathcal{D}'_1, \ldots, \mathcal{D}'_{N_G}$.
- N_L identical SLU models are train for *each* augmented dataset $\mathcal{D}'$. All models are validated against the evaluation results on the same validation split. Best model from each SLU model is evaluated on the test set.
- We collect the statistics of all $N_G \times N_L$ results and perform comparative analyses.

Implementation Details

For both models, the word (W_w), slot label (W_s), and intent (W_y) embeddings have dimensions of 300, 200, and 100 respectively and were trained jointly with the network. W_w had been initialized with the GloVe vectors (Pennington et al., 2014).

Generative Model

The encoder network, a single-layer BiLSTM-Max model (Conneau et al., 2017), encodes the word embeddings of word tokens $w_i \in \mathbf{w}$ in both directions and produces the final hidden state by applying max-pooling-over-time on combined encoder hidden outputs $\mathbf{h}_1^{(e)}, \ldots, \mathbf{h}_T^{(e)}$ (1024 hidden dimensions). The decoders are uni-directional single-layer LSTMs with the same hidden dimensions (1024). Let $\mathbf{h}_t^{(w)}$, $\mathbf{h}_t^{(s)}$, and $\mathbf{h}_t^{(y)}$ be the hidden outputs of word, slot label, and intent decoders at time step t respectively. We perform dot products between respective embeddings and the hidden outputs to obtain logits (e.g. $\mathbf{o}_t^{(w)} = W_w \mathbf{h}_t^{(w)}$ etc.). The likelihood of each token at each time step t is obtained by applying the softmax on the logits:

$$p\left(w_t | \mathbf{w}_{<t}, \mathbf{z}\right) = \frac{e^{o_{t,w_t}^{(w)}}}{\sum_{w' \in V_w} e^{o_{t,w'}^{(w)}}}.$$

Where V_w is the vocabulary set of utterance words. During generation, the beam search algorithm is used to search for the most likely sequence candidates using the conditional token distributions. The beam search size was set to 15 and the utterances were sampled from top-1 (k_b) candidate(s) to reduce variance. Exploratory hyperparameter λ_s was 0.18.

To feasibly train the model, we employ the teacher-forcing strategy, in which the LU network is trained on the ground truth utterance $\mathbf{w}$ instead of the predicted sequence $\hat{\mathbf{w}}$. We applied KLD annealing and the decoder word dropout (Bowman et al., 2016). KLD annealing rate (k_d) was 0.03 and word dropout rate p_w was 0.5. We used Adam (Kingma and Ba, 2014) optimizer with 0.001 initial learning rate. The code is available on github (`kaniblu/ludus-jluva`).

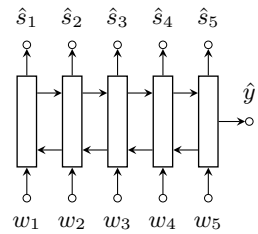


Figure 4.2: BiLSTM-Max architecture for joint language understanding

SLU Models

For the baseline SLU model, we implemented a relatively simple BiLSTM model (Figure 4.2), used as the control SLU model in the data scarcity and ablation studies. A bidirectional LSTM cell encodes an utterance into a fixed size representation h. A fully connected layer translates the hidden outputs h_t of the BiLSTM to slot scores for all time step t. The softmax function is applied to the logits to produce $p\left(s_t|\mathbf{w}_{\leq t}\right)$. The final hidden representation h of the input utterance is obtained by applying max-pooling-over-time on all hidden outputs. Another fully connected layer and a softmax function maps h_t to the intent distribution $p\left(y|\mathbf{w}\right)$. This simple baseline was able to achieve 95.32 in the slot filling f1-score.

For other SLU models, we consider the slot-gated SLU model (Goo et al., 2018), which incorporates the attention and the gating mechanism into the LU network. We found the model suitable for our task, as the model is reasonably complex and distinctive from our simple baseline. Furthermore, the code for running the model is publicly available and the results are readily reproducible. We were able to obtain similar or even better results on our environment (Table 4.6). This difference might be due to differing data preprocessing methods. SLU performance is measure by (1) slot filling f1-score (evaluated using the conlleval perl script), (2) intent identification f1-sore, and (3) semantic frame formulation. f1-score measures the correctness of predicted slot labels.

	Slot-Gated (Full)			Slot-Gated (Intent)		
Dataset	Slot	Intent	SF	Slot	Intent	SF
ATIS	95.3‡	94.9‡	84.3‡	95.4‡	94.7‡	83.5‡
ATIS+	**95.7**	**95.6**	**85.4**	**95.6**	**95.6**	**84.8**
Snips	88.2‡	97.0	74.9‡	88.2	**96.9**	74.6
Snips+	**89.3**	**97.3**	**76.4**	**88.3**	96.7	**74.6**
ME	82.2‡	-	63.6‡	81.8‡	-	62.1‡
ME+	**82.9**	-	**64.5**	**82.8**	-	**63.3**
MT	63.5‡	-	24.0‡	62.8‡	-	24.4‡
MT+	**65.7**	-	**27.4**	**65.0**	-	**27.5**
MR	72.6†	-	52.8†	72.1‡	-	51.8‡
MR+	**73.0**	-	**53.4**	**73.0**	-	**52.9**

† $p < 0.1$ ‡ $p < 0.01$

Table 4.5: Mean data augmentation results on various SLU tasks tested using the slot-gated (Goo et al., 2018) SLU models

Dataset	Model	Slot (F1)	Intent (F1)
ATIS	JLUVA	94.44	97.09
ATIS	BiLSTM (Baseline)	95.34	97.21
ATIS	Deep LSTM[a]	95.66	-
ATIS	Slot-Gated (Full)[b,d]	95.66	96.08
ATIS	Att. Encoder-Decoder[c]	95.87	**98.43**
ATIS	Att. BiRNN[c]	95.98	98.21
ATIS+	BiLSTM (Baseline)	95.75	97.54
ATIS+	Slot-Gated (Full)[b,d]	**96.04**	96.75

[a] (Kurata et al., 2016b) [b] (Goo et al., 2018) [c] (liu, 2016)

[d] run on our environment

Table 4.6: Comparisons of the best slot filling and intent detection results for the ATIS dataset.

4.4.3 Generative Data Augmentation Results

In this section, we describe and present two experiments that test the GDA approach under variety of experimental settings: data scarce scenarios, varied SLU models, and varied datasets.

Data Scarce Scenario

For the first experiment, we test whether our GDA approach performs better than the previous work 1) under the regular condition (full datasets) 2) and data scarce scenarios. We compare our model to a deterministic encoder-decoder model (Seq2Seq) proposed in (Kurata et al., 2016a). The two decoders of the model learn to decode utterances and slot labels from an encoded representation of the utterance. In an attempt to reproduce the results of the original models, we restrict the model from generating intents. However, we could still observe

some discrepancies between our results and the results reported in the paper (Table 4.4), possibly due to minor differences in experimental protocols.

For the full dataset, we conduct the standard experiments with $N_G = 3, N_L = 3$ and $m = 10000$, synthetic dataset size. For small and medium datasets, each experiment is repeated $N_L = 3$ times for *all* N_T training splits. The final result is aggregated from $N_T \times N_L$ runs (i.e. 105 runs for ATIS-small and 27 runs for ATIS-medium). Results are presented in 4.4. We use the baseline BiLSTM model as the control SLU model. Results are averaged over multiple runs and compared to the best of our approaches (JLUVA + Posterior). The differences are tested for statistical significance.

According to the results, our approach performed better than all other baselines at the statistically significant level for small and medium datasets. The performance gain of our approach diminishes for the full dataset. This is likely due to the homogeneous nature of the ATIS dataset, leaving little room for the GDA to explore. Although we could not achieve statistically significant improvement on the full dataset, we note that our approach never experiences performance degradation for any dataset size and evaluation measure.

GDA on Other SLU Models and Datasets

We test GDA with various combinations of SLU models and datasets (Table 4.5). Dataset that are augmented using our proposed generative model is denoted by +. The results have been aggregated and were tested for statistical significance. There were statistically significant improvements in language understanding performances across most datasets and SLU models. Comparing these results with the data scarcity results in Table 4.4, we observe two trends: (1) the more difficult the dataset is to model (e.g. MIT Movie Trivia) and (2) the more expressive the SLU model, the more drastic the improvements are. For example, the improvement rate between ATIS and ATIS+ for full

attention-based Slot-Gated model was only 0.39%, whereas the improvement rate increased nearly ten-fold (3.54%) between MIT Movie Eng and MIT Movie Eng+ for the same model.

We also observe a positive correlation between model complexity and performance gains. For example, the performance improvement was more significant for the slot-gated model than the simple baseline model for the ATIS dataset. This suggests that the performance-boosting benefits from synthetic datasets can be more easily captured by more expressive models. This is also supported by generally better performances achieved by the slot-gated full attention model, as the full attention variant is the more complex one.

4.4.4 Comparison to Other State-of-the-art Results

In this study, we compare the best LU performance achieved by our generative approach on the ATIS task to other state-of-the-art results in literature (Table 4.6). We chose the best performing run out of all runs carried out from the previous experiments ($N_G = 3, N_L = 3, m = 10000$) and report its results in Table 4.6. In the best case, our approach was able to boost the slot filling performance for the slot-gated (full) model by 0.38. Remarkably, our best results outpeformed more complex models, further supporting the idea of data-centric regularization. We also evaluate the SLU performance of JLUVA by performing deterministic inference (i.e. $\mathbf{z} = \boldsymbol{\mu}$). We find that the LU performance by itself is not competitive. This eliminates the possibility that the performance gains in our approach are attributed to JLUVA being a more expressive model and therefore acting as a teacher network.

4.4.5 Ablation Studies

In the ablation studies, we carry out two separate comparative experiments on variations of our generative model.

Sampling Methods

The following sampling approaches are considered.

- **Monte-Carlo Posterior Sampling (Ours)**: $\mathbf{z}$ is sampled from the empirical expectation of the model, which is estimated by inferring posteriors from random utterance samples. (Algorithm 1)
- **Standard Gaussian**: $\mathbf{z}$ is sampled from the assumed prior, the standard multivariate Gaussian.
- **Additive Sampling**: First, the latent representation $\mathbf{z_w}$ of a random utterance $\mathbf{w}$ is sampled. Then $\mathbf{z_w}$ is disturbed by a perturbation vector $\boldsymbol{\alpha} \sim \mathcal{U}(-0.2, 0.2)$. It was proposed for the deterministic model in (Kurata et al., 2016a).

The results in Table 4.4 confirm that exploratory Monte-Carlo sampling based on scaled posterior distribution ($\lambda_s = 0.18$) provides the greatest benefit to the language understanding models for the ATIS and the data-scarce datasets. We note that the additive perturbation, despite its simplicity in nature, performs reasonably well compared to our approach. This suggests the exploratory sampling approaches are not only limited to Gaussian distributions. On the other hand, over-simplified and biased approximation of the prior such as standard multivariate Gaussian, could rather cause performance degradation. This also highlights the fact that the choice of sampling approach has a significant impact on the generative quality and thereby the resulting performances.

Synthetic Data Ratio

To gain further insights into generative DA, we conduct regressional experiments to expose the underlying relationship between the relative synthetic data

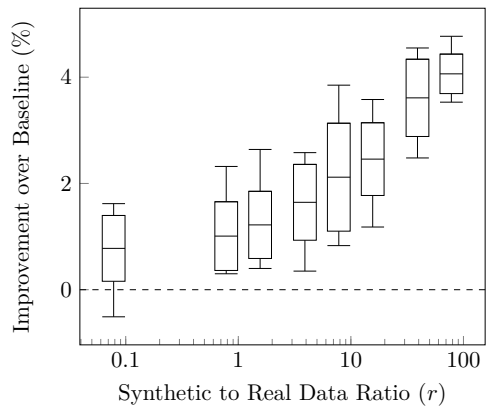


Figure 4.3: Plotting of data augmentation benefits on intent classification over synthetic data to real data ratio.

size and the performance improvements.

Let m be the size of the synthetic dataset used to augment the original dataset of size n. The *synthetic to real data ratio* r is m/n. For each run, we conduct the standard experiment procedure ($N_G = 10, N_L = 5$) on a ATIS-small dataset with JLUVA as the generative model and the simple BiLSTM as the SLU model. We repeat the experiments for all $r \in \{0.08, 0.78, 1.56, 3.90, 7.81, 15.6, 39.06, 78.13\}$.

The impact of synthetic data to real data ratio on the relative improvements in SLU performance are shown in Figures 4.3 to 4.5. In the figures, sashed horizontal lines illustrate the level at which the impact on the performance is no longer positive. The baseline is the SLU performance achieved without an augmenting dataset. For each box plot, the center line denotes μ, the top and bottom boundaries denote $\mu + \sigma$ and $\mu - \sigma$ respectively, and both whiskers denote the maximum and the minimum respectively.

From the box plots of our results, we make two observations. First, the maximum marginal improvement is achieved around $10 \leq r \leq 20$ for all evaluation measures. Also, the improvements appear to plateau around $r = 50$. Second,

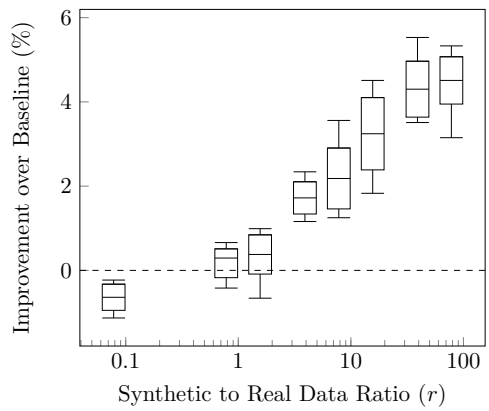


Figure 4.4: Plotting of data augmentation benefits on slot filling over synthetic data to real data ratio.

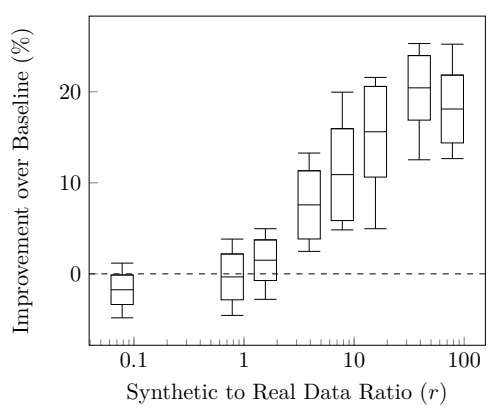


Figure 4.5: Plotting of data augmentation benefits on semantic frame parsing over synthetic data to real data ratio.

The variance starts off relatively small when $r < 1$, but it quickly grows as r increases and peaks around $5 \leq r \leq 20$. The variance appears to shrink again after $r > 20$. A plausible explanation for the apparent trend of the variance is that increasing r enhances the chance to generate performance-boosting key utterances, until no novel instances of such utterances are samplable from the generator, at which point further increasing r only increases the chance to generate already known utterances, thereby reducing the variance. This also explains the plateauing phenomenon.

4.5 Summary

In this paper, we formulated the generic framework for generative data augmentation (GDA) and derived analytically the most effective sampling approach for generating performance-boosting instances from our proposed generative model, Joint Language Understanding Variational Autoencoder (JLUVA). Based on the positive experimental results, we believe that our approach could bring immediate benefits to SLU researchers and the industry by reducing the cost of building new SLU datasets and improve performances of existing SLU models. Although our work has primarily been motivated by the data issues in SLU datasets, we would like to invite researchers to explore the potential of applying GDA in other NLP tasks, such as neural machine translation and natural language inference. Similar to the work done by Dao et al. on the analysis of class-preserving transformative DAs using the kernel theory (Dao et al., 2018), our work also calls for deeper theoretical analysis on the mechanism of data-centric regularization techniques. We wish to address these issues in our future work.

Chapter 5

Complex Data: Dialogue State Tracking

5.1 Introduction

In contrast to text classification or natural language understanding, the ability to understand conversations and generate appropriate responses given a conversational context can be considered the hallmark of artificial intelligence, as it transcends all tasks in the NLP domain (Bengio, 2017). As an intermediate step towards the ultimate goal of realizing conversational agents, the study of task-oriented dialogues had been proposed in the past. The practicality and the availability of datasets have garnered interest in task-oriented dialogue modeling from the NLP community in recent years.

Although fully annotated task-oriented dialogue datasets were made available by various researchers (Wen et al., 2017; El Asri et al., 2017) over the years, there will always be a shortage of labeled dialogue datasets due to the the necessity of having domain-specific datasets, sparseness of dialogue acts and the significant labor cost of collecting, annotating and validating dialogue corpora. The ability to generate novel dialogue samples without supervision could not only help with directly boosting the downstream supervised learning tasks but also be utilized for assisting data collection and annotation process, thereby reducing the construction cost.

In this chapter, we intend to explore the feasibility and limitation of bringing generative data augmentation in the domain of task-oriented dialogue modeling, initiating a new paradigm of generating fully annotated dialogue samples for

various tasks.

Recent approaches for language-driven dialogue modeling is largely based on the recurrent neural network (RNN) architecture (Vinyals and Le, 2015; Shang et al., 2015; Li et al., 2017), while some more recent works have employed variational inference (Serban et al., 2017; Park et al., 2018; Bak and Oh, 2019). Although VAEs offer us several advantanges, i.e. rich representations and the capability of modeling variability in linguistic or higher-level features, they can fall into a local optimum where they fail to encode meaningful representations, also known as the *posterior collapse* or the degeneration phenomenon (Bowman et al., 2016; Chen et al., 2016a; He et al., 2019). Posterior collapse, specifically inference collapse, is mainly caused by teacher-forcing training of autoregressive decoders, inducing the decoders' strong reliance on autoregressive signals rather than the information provided by the encoder. Dropout-based measures have been proposed to reduce the risk of posterior collapse such as word dropouts (Bowman et al., 2016) and utterance dropouts (Park et al., 2018); however, the problems of these measures are two-folds when applied to deep hierarchical latent variable models. First, the trade-off between posterior collapse mitigation and the decoder performance present in autoregressive dropouts exacerbates the optimization complexity. Second, the employment of dropout measures do not guarantee elimination of the risk of posterior collapse.

In this work, we not only propose a novel latent variable model that is capable of learning the joint distribution of lingusitic and the underlying structured features of goal-oriented dialogues, but we also devise novel training policies for degeneration alleviation. The summary of the contributions are as follows.

- We propose Variational Hierarchical Goal-oriented Dialogue Autoencoder (VHDA), a deep latent variable model for generating both dialogue utterances and the complete aspect of underlying dialogue acts and features. We show that not only the training of the deep hierarchical variational

model is feasible, but the model shows good generalizability, generating coherent and diverse samples.

- In the process of realizing the deep variational model, we propose and employ two novel measures for reducing the risk of posterior collapse, a phenomenon where VAEs fully or partially fail to autoencode.
- Experiments on multiple datasets and dialogue state tracking models are conducted to confirm the benefits of data augmentation on the state tracking task.

The rest of the chapter is structured as follows. Section 5.2 summarizes the related work on task-oriented dialogue modeling. Section 5.3 describes our proposed model and relevant techniques in details, including the novel measures related to reducing the risk of posterior collapse. Section 5.4 provides all details regarding the experiments on our model, including data augmentation experiments on dialogue state tracking and ablation studies. In the final section, we summarize the chapter and offer limitations and future work.

5.2 Background and Related Work

5.2.1 Task-oriented Dialogue

In this section, we formalize the problem setting of modeling task-oriented dialogues. Task-oriented dialogues (TOD) or goal-oriented dialogues are a subset of general conversational dialogues, in which two participants (a user and a system) partake in asymmetric interaction to resolve common goals. Interaction asymmetry arises from the difference in the information sources accessible by the two parties (e.g. the system has the privilege to access an internal database of restaurants or the system is able to take exclusive actions such as book restaurants on an internal reservation system), hence the parties are incentivized to

interact verbally. The goal of the user is to obtain particular information from the system or request the system to take certain actions that only the system is able to. On the other hand, the goal of the system is to provide appropriate information when requested and take appropriate system actions when they are deemed necessary. All communications of requests and transfer of information must be carried out via language, hence TODs usually take multiple interactive turns to completely resolve user goals in a session. In order to successfully carry out a task-oriented interactive session with the user, the system must be capable of performing several key factors:

1. **User Utterance Understanding.** Given a history of interactions (utterances, system acts, etc.) in a session, the system must be capable of infer context-sensitive semantics from a user utterance.

2. **User Goal Inference.** Based on the history of interactions and the semantics of the current user utterance, the system must also be able to infer the session-wide goals of the user.

3. **System Action Prediction.** Taking all the information into account, including past utterances, past system acts, inferred current user utterance semantics, and inferred user goals, the system must hold an optimal policy of system actions that are geared towards resolving the user goals as efficiently as possible.

As evident from these factors, task-oriented dialogue modeling encompasses multiple sub-tasks, such as **dialogue state tracking (DST)** and **user simulation**. In DST, the task is to extract semantics from a user utterance as well as the dialogue-wide goals of the user, which are called the **goal states** or **belief states** depending on the literature.

5.2.2 Dialogue State Tracking.

Dialogue state tracking (DST) is the task of predicting the user's current goals and dialogue acts given the context of the dialogue. Historically, DST models relied on hand-crafted finite-state automata to emulate humans in conversations (Dybkjær and Minker, 2008) or separate SLU modules to achieve dialogue tracking using a two-stage processs (Thomson and Young, 2010; Wang and Lemon, 2013; Henderson et al., 2014b). Recent approaches combine the two-stage process into one unified model to directly predict dialogue states from dialogue features (Zilka and Jurcicek, 2015; Mrkšić et al., 2017; Zhong et al., 2018; Nouri and Hosseini-Asl, 2018; Wu et al., 2019).

Among the integrated single-stage models, the earlier ones relied on delexicalization – the act of replacing entities in slots and values with generic tags using handcrafted semantic dictionaries – to improve generalization. Neural Belief Tracker (NBT) (Mrkšić et al., 2017) has been proposed to decrease reliance on handcrafted semantic dictionaries by reformulating the multi-class classification problem to multiple binary classification problems. GLAD (Zhong et al., 2018) improves upon NBT by introducing global modules (for sharing parameters among estimators for slot values) and local modules to learn slot-specific feature representations. GCE (Nouri and Hosseini-Asl, 2018) improves within the paradigm of neural belief tracking by forgoing the separation of global and local modules and letting the unified module to take slot embeddings as the condition, greatly reducing the number of parameters and improving the inference efficiency.

5.2.3 Conversation Modeling.

Since hierarchical modeling is naturally suitable for the task of dialogue modeling, recent works have explored hierarchically-structured deep networks for learning and generating conversations (Vinyals and Le, 2015; Serban et al.,

2016). There have also been considerable efforts to employ variational inference to increase the modeling capacity and the generation diversity of the models (Serban et al., 2017; Park et al., 2018; Gu et al., 2018; Shen et al., 2018; Bak and Oh, 2019). The prominent approach for dialogue modeling was the Markov assumption (Serban et al., 2017), but recent approaches have converged on utilizing global latent variables for representating the holistic properties of dialogues (Park et al., 2018; Gu et al., 2018; Bak and Oh, 2019), which preserves long term dependencies in the dialogue. In this work, we employ global latent variables to maximize the effectiveness in preserving dialogue semantics for data augmentation.

5.3 Variational Hierarchical Dialogue Autoencoder (VHDA)

In this section, we describe the proposed latent variable model for generating goal-oriented dialogue datasets complete with their annotations. To facilitate in describing our main work, we introduce a set of notations for representing dialogue-related concepts and offer a short description of the prior work that uses a hierarchical VAE structure to solely model the linguistic features (Park et al., 2018). In the rest of the section, we present details and inner workings of VHDA, which captures not only the linguistic features but also the underlying structural features simultaneously.

5.3.1 Notations

In this subsection, we establish a set of general notations for describing any type of goal-oriented dialogues. A goal-oriented dialogue dataset $\mathcal{D}$ is a set of N i.i.d goal-oriented dialogue samples $\{\mathbf{c}_1, \ldots, \mathbf{c}_N\}$, where each $\mathbf{c}$ is a sequence of dialogue turns $(\mathbf{v}_1, \ldots, \mathbf{v}_T)$. Each goal-oriented dialogue turn $\mathbf{v}$ is a

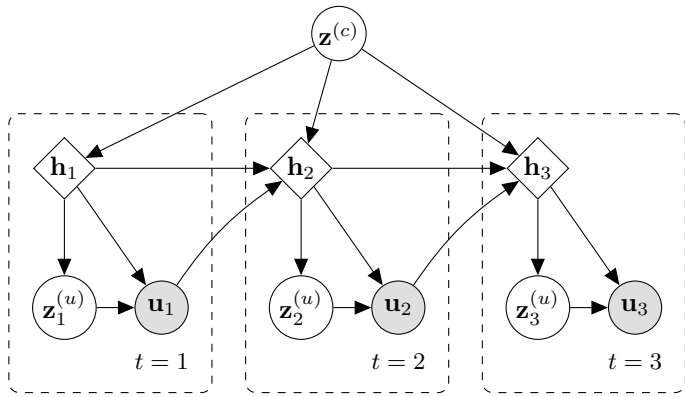


Figure 5.1: Graphical representation of VHCR (Park et al., 2018).

tuple of speaker information $\mathbf{r}$, the speaker's goals $\mathbf{g}$, dialogue state $\mathbf{s}$, and the speaker's utterance $\mathbf{u}$: $\mathbf{v} = (\mathbf{r}, \mathbf{g}, \mathbf{s}, \mathbf{u})$. Each utterance $\mathbf{u}$ is a sequence of words $\left(w_1, \ldots, w_{|\mathbf{u}|}\right)$. Each set of speaker goals $\mathbf{g}$ and each dialogue state $\mathbf{s}$ are defined as a set of the smallest unit of dialogue state specification a (Henderson et al., 2014a), which is a tuple of dialogue act, slot and value defined over the space of dialogue acts $\mathcal{A}$, slots $\mathcal{S}$, and values $\mathcal{V}$: $\mathbf{g} = \left\{a_1, \ldots, a_{|\mathbf{g}|}\right\}$, $\mathbf{s} = \left\{a_1, \ldots, a_{|\mathbf{s}|}\right\}$, where $a_i \in (\mathcal{A}, \mathcal{S}, \mathcal{V})$. In literature, such as (Henderson et al., 2014a), a is represented in the human-readable form as `<act>(<slot>=<value>)`, while multiple a that share the same dialogue act are represented in a similar format by listing slot-value pairs in the same parentheses, separated by commas: `<act>(<slot1>=<value1>, ..., <slotN>=<valueN>)`.

5.3.2 Variational Hierarchical Conversational RNN

Given a conversation $\mathbf{c}$, Variational Hierarchical Conversational RNN (VHCR) (Park et al., 2018) models the holistic features of the conversation as well as individual utterances $\mathbf{u}$ using a hierarchical and recurrent VAE model, as shown in Figure 5.1. In the figure, we assume that the length of a conversation T is 3. The model introduces global-level latent variables $\mathbf{z}^{(c)}$ for encoding the

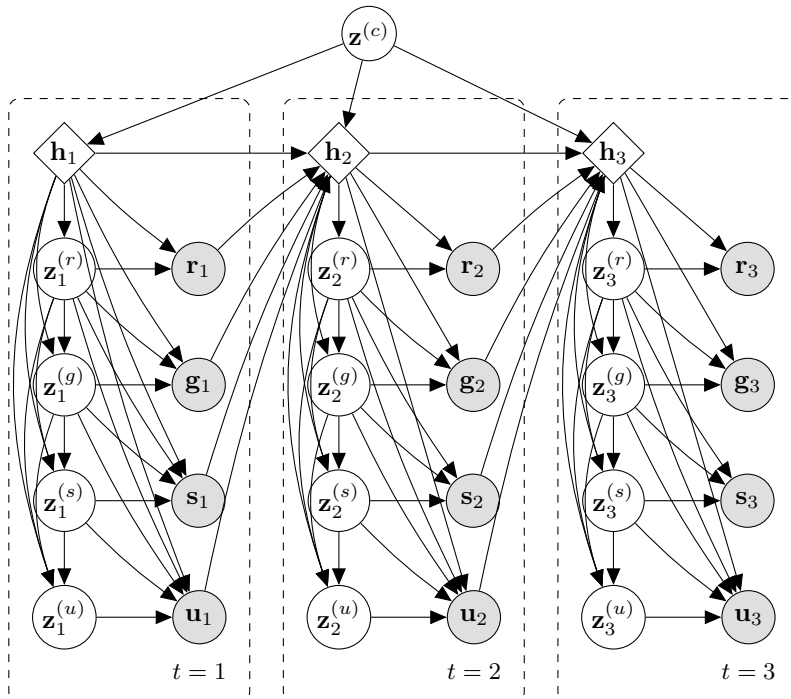


Figure 5.2: Graphical representation of VHDA.

high-level structure of the conversation, and local-level latent variables $\mathbf{z}_t^{(u)}$ responsible for encoding and generating utterances at each turn step t. The local latent variables $\mathbf{z}^{(u)}$ are designed to be conditionally dependent on $\mathbf{z}^{(c)}$ and the previous observations, forming a hierarchical structure. This model is realized by the hidden variables $\mathbf{h}_t$ that have conditional dependence on the global information and the hidden variables from the previous timestep $\mathbf{h}_{t-1}$.

5.3.3 Proposed Model

To achieve complete modeling of goal-oriented dialogues, we propose Variational Hierarchical Dialogue Autoencoder (VHDA) to generate dialogues and their underlying dialogue annotations simultaneously (Figure 5.2). Similar to VHCR, we employ a hierarchical latent structure to capture both the holistic dialogue semantics using the conversation latent variables $\mathbf{z}^{(c)}$ and individual

turn-level features $\mathbf{z}^{(r)}$ (speaker), $\mathbf{z}^{(g)}$ (goal), $\mathbf{z}^{(s)}$ (dialogue state), and $\mathbf{z}^{(u)}$ (utterance). Motivated by speech act theory (Perrault et al., 1978), we also employ a hierarchical structure for the turn-level latent variables, in which the utterance latent variables $\mathbf{z}^{(u)}$ are dependent on all other latent variables within the same turn. The model is not only capable of generating linguistic features and the relevant annotations from a single model, but it is also capable of generating languages of higher quality and diversity thanks to the effect of joint learning, which we discuss in Section 5.4.2.

VHDA consists of multiple encoder modules and multple decoder modules, each responsible for extracting features or generating a particular dialogue feature. However, multiple encoders share the same sequence-encoding architecture (but not parameters).

Sequence Encoder Architecture

Given a sequence of variable number of elements $\mathbf{X} = [\mathbf{x}_1; \ldots; \mathbf{x}_n]^\intercal \in \mathbb{R}^{n \times d}$, where n is the number of elements, the goal of a sequence encoder is to extract a fixed-size representation $\mathbf{h} \in \mathbb{R}^d$, where d is the dimensionality of the hidden representation. For our implementation, we employ a shallow self-attention mechanism over hidden outputs of bidirectional LSTM (Hochreiter and Schmidhuber, 1997) cells produced from taking inputs from the input sequence. We also allow the attention mechanism to be queried by external variables, enabling the sequence to be attended according to more specific external factors, such as attending over a word sequence of an utterance based on a dialogue context:

$$
\begin{aligned}
\overrightarrow{\mathbf{H}} &= \overrightarrow{\mathrm{LSTM}}(\mathbf{X}) \\
\overleftarrow{\mathbf{H}} &= \overleftarrow{\mathrm{LSTM}}(\mathbf{X}) \\
\mathbf{H} &= \left[\overrightarrow{\mathbf{H}}; \overleftarrow{\mathbf{H}}\right] \in \mathbb{R}^{n \times d} \\
\mathbf{a} &= \mathrm{softmax}(\mathbf{W}[\mathbf{H}; \mathbf{Q}]^\intercal + \mathbf{b}) \in \mathbb{R}^{n} \\
\mathbf{h} &= \mathbf{H}^\intercal \mathbf{a} \in \mathbb{R}^{d}
\end{aligned}
$$

Here, $\mathbf{Q} \in \mathbb{R}^{n \times d_q}$ is a collection of query vectors of dimensionality d_q that can query each element in the sequence; $\mathbf{W} \in \mathbb{R}^{d+d_q}$ and $\mathbf{b} \in \mathbb{R}^{d+d_q}$ are learnable parameters for inferring the attention weights $\mathbf{a}$ with given hidden outputs $\mathbf{H}$ and query vectors $\mathbf{Q}$. We encapsulate above operations using the notation ENC, which takes a sequence of input vectors and query vectors and returns a fixed sized representation and is defined as follows.

$$
\mathrm{ENC} : \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times d_q} \rightarrow \mathbb{R}^{d}
$$

The architecture of ENC is utilized repetitively for encoding various features in dialogues that have dynamic lengths (sequences of words, sequences of dialogue acts, sequences of turns, etc.).

VHDA Architecture

Our model architecture consists of five sequence encoders (dialogue act encoder $\mathrm{ENC}^{(a)}$, goal encoder $\mathrm{ENC}^{(g)}$, dialogue state encoder $\mathrm{ENC}^{(s)}$, utterance encoder $\mathrm{ENC}^{(u)}$, and conversation encoder $\mathrm{ENC}^{(c)}$), a context encoder $\mathrm{ENC}^{(\mathrm{ctx})}$, and four decoders for each dialogue feature (speaker decoder $\mathrm{DEC}^{(a)}$, goal decoder $\mathrm{DEC}^{(g)}$, state decoder $\mathrm{DEC}^{(s)}$, and utterance decoder $\mathrm{DEC}^{(u)}$, all parameterized separately. In addition to the conversational latent variables $\mathbf{z}^{(c)}$ and utterance

latent variables $\mathbf{z}^{(u)}$ introduced in the previous subsection, our model also consists of latent variables for the speaker $\mathbf{z}^{(r)}$, the goal $\mathbf{z}^{(g)}$, and the dialogue state $\mathbf{z}^{(s)}$ at each dialogue turn.

Initially, the global latent variable $\mathbf{z}^{(c)}$ is generated from a standard Gaussian prior: $p\Big(\mathbf{z}^{(c)}\Big) = \mathcal{N}(0, \mathbf{I})$. At dialogue turn step t, VHDA uses the context encoder $\text{ENC}^{(\text{ctx})}$ to encode the context information $\mathbf{h}_t$ using (1) the context information $\mathbf{h}$ encoded from the previous turn step $t-1$ and (2) the information about all dialogue features (the speaker $\mathbf{r}$, the goal $\mathbf{g}$, the dialogue state $\mathbf{s}$, and the utterance $\mathbf{u}$) at the current turn step:

$$\begin{aligned}
\mathbf{v}_{t-1} &= \Big[\mathbf{h}_{t-1}^{(r)}; \mathbf{h}_{t-1}^{(g)}; \mathbf{h}_{t-1}^{(s)}; \mathbf{h}_{t-1}^{(u)}\Big] \\
\mathbf{h}_t &= \text{ENC}^{(\text{ctx})}(\mathbf{h}_{t-1}, \mathbf{v}_{t-1})
\end{aligned}$$

where $\mathbf{v}_t$ is the concatenation of all feature representations at the turn step t. Note that context encoder $\text{ENC}^{(\text{ctx})}$ has a different structure than other sequence encoders in that it employs uni-directional sequence encoding and takes inputs from the previous turn step and returns hidden outputs one step at a time. Here, the hidden representations of the dialogue features $\mathbf{h}^{(r)}$, $\mathbf{h}^{(g)}$, $\mathbf{h}^{(s)}$, and $\mathbf{h}^{(u)}$ are encoded by the sequence encoders from the respective dialogue features, which we describe in subsequent subsections.

For the following step, VHDA successively generates latent variables using a series of inference networks for each turn step t:

$$\begin{aligned}
p_\theta\Big(\mathbf{z}_t^{(r)}\Big|\mathbf{v}_{<t}, \mathbf{z}^{(c)}\Big) &= \mathcal{N}\Big(\mu_t^{(r)}, \sigma_t^{(r)}\mathbf{I}\Big) \\
p_\theta\Big(\mathbf{z}_t^{(g)}\Big|\mathbf{v}_{<t}, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}\Big) &= \mathcal{N}\Big(\mu_t^{(g)}, \sigma_t^{(g)}\mathbf{I}\Big) \\
p_\theta\Big(\mathbf{z}_t^{(s)}\Big|\mathbf{v}_{<t}, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}\Big) &= \mathcal{N}\Big(\mu_t^{(s)}, \sigma_t^{(s)}\mathbf{I}\Big) \\
p_\theta\Big(\mathbf{z}_t^{(u)}\Big|\mathbf{v}_{<t}, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}, \mathbf{z}_t^{(s)}\Big) &= \mathcal{N}\Big(\mu_t^{(u)}, \sigma_t^{(u)}\mathbf{I}\Big)
\end{aligned}$$

Here, note that the gaussian distribution encoders (μ and σ) are implemented using multi-layer feedforward networks f that predict the parameters of gaussian distribution families given all previous conditions:

$$
\begin{aligned}
\mu_t^{(r)} &= f_\theta^{(r)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}\Big) \\
\sigma_t^{(r)} &= \text{softplus}\Big(f_\theta^{(r)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}\Big)\Big) \\
\mu_t^{(g)} &= f_\theta^{(g)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}\Big) \\
\sigma_t^{(g)} &= \text{softplus}\Big(f_\theta^{(g)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}\Big)\Big) \\
\mu_t^{(s)} &= f_\theta^{(s)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}\Big) \\
\sigma_t^{(s)} &= \text{softplus}\Big(f_\theta^{(s)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}\Big)\Big) \\
\mu_t^{(u)} &= f_\theta^{(u)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}, \mathbf{z}_t^{(s)}\Big) \\
\sigma_t^{(u)} &= \text{softplus}\Big(f_\theta^{(u)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}, \mathbf{z}_t^{(s)}\Big)\Big)
\end{aligned}
$$

We use the reparameterization trick (Kingma and Welling, 2013) to allow the samples of latent variables to be computed with standard backpropagation during training and optimization.

Approximate Posterior Networks

A separate set of parameters, denoted by ϕ, approximates posterior distributions of all latent variables from evidence. The global latent variables $\mathbf{z}^{(c)}$ are estimated using the conversation encoder based on hidden representations of all dialogue features.

$$
\begin{aligned}
q_\phi\Big(\mathbf{z}^{(c)}\Big|\mathbf{v}_1,\ldots,\mathbf{v}_T\Big) &= \mathcal{N}\Big(\mu^{(c)},\sigma^{(c)}\mathbf{I}\Big)\\
\mathbf{h}^{(c)} &= \mathrm{ENC}^{(c)}([\mathbf{v}_1;\ldots,\mathbf{v}_T])\\
\mu^{(c)} &= f_\phi^{(c)}\Big(\mathbf{h}^{(c)}\Big)\\
\sigma^{(c)} &= \mathrm{softplus}\Big(f_\phi^{(c)}\Big(\mathbf{h}^{(c)}\Big)\Big)
\end{aligned}
$$

The rest of the turn-level latent variables are estimated similarly conditioned on conversation latent variables $\mathbf{z}^{(c)}$ and turn-level hidden factors $\mathbf{h}_t$:

$$
\begin{aligned}
q_\phi\Big(\mathbf{z}_t^{(r)}\Big|\mathbf{v}_{<t},\mathbf{z}^{(c)},\mathbf{h}_t^{(r)}\Big) &= \mathcal{N}\Big(\mu_t^{(r')},\sigma_t^{(r')}\mathbf{I}\Big)\\
&\vdots\\
q_\phi\Big(\mathbf{z}_t^{(u)}\Big|\mathbf{v}_{<t},\mathbf{z}^{(c)},\ldots,\mathbf{h}_t^{(u)}\Big) &= \mathcal{N}\Big(\mu_t^{(u')},\sigma_t^{(u')}\mathbf{I}\Big)
\end{aligned}
$$

$$
\begin{aligned}
\mu_t^{(r')} &= f_\phi^{(r)}\Big(\mathbf{h}_t,\mathbf{z}^{(c)},\mathbf{h}_t^{(r)}\Big)\\
\sigma_t^{(r')} &= \mathrm{softplus}\Big(f_\phi^{(r)}\Big(\mathbf{h}_t,\mathbf{z}^{(c)},\mathbf{h}_t^{(r)}\Big)\Big)\\
&\vdots\\
\mu_t^{(u')} &= f_\phi^{(u)}\Big(\mathbf{h}_t,\mathbf{z}^{(c)},\ldots,\mathbf{h}_t^{(u)}\Big)\\
\sigma_t^{(u')} &= \mathrm{softplus}\Big(f_\phi^{(u)}\Big(\mathbf{h}_t,\mathbf{z}^{(c)},\ldots,\mathbf{h}_t^{(u)}\Big)\Big)
\end{aligned}
$$

Common Encoder Networks

Apart from the recognition networks, common encoders are responsible for encoding dialogue features from their respective feature spaces to hidden representations that can be understood by the recognition and decoder networks. Hence the parameters are shared across the recognition and decoder networks. Specifically, each dialogue feature of $\mathbf{h}^{(r)}$, $\mathbf{h}^{(g)}$, $\mathbf{h}^{(s)}$, and $\mathbf{h}^{(u)}$ is encoded by the

respective sequence encoder. For speaker information $\mathbf{h}^{(r)}$, the encoding mechanism is achieved by a speaker embedding matrix $\mathbf{W}^{(r)} \in \mathbb{R}^{n^{(r)} \times d^{(r)}}$, where $n^{(r)}$ is the number of participants and $d^{(r)}$ is the dimensionality of the speaker embedding. Assuming that the speaker information is given as a one-hot encoded vector, the speaker embedding is obtained by $\mathbf{W}^{(r)}\mathbf{r}$.

For goal representation $\mathbf{h}^{(g)}$ and dialogue state (or act) representation $\mathbf{h}^{(s)}$, the encoding takes place over two-steps. In the first step, given a set of dialogue state specifications $\mathbf{g} = \left\{a_1, \ldots, a_{|\mathbf{g}|}\right\}$ or $\mathbf{s} = \left\{a_1, \ldots, a_{|\mathbf{s}|}\right\}$, a common dialogue act encoder $\text{ENC}^{(a)}$ encodes each dialogue act specification into a fixed size hidden representation $\mathbf{h}^{(a)}$. In the second step, the hidden representations of dialogue act specifications $\mathbf{h}^{(a)}$ are encoded by the respective encoder into a fixed size representation for the goal or the dialogue state:

$$\mathbf{h}^{(g)} = \text{ENC}^{(g)}\Big(\Big[\text{ENC}^{(a)}\Big(a_1^{(g)}\Big); \ldots; \text{ENC}^{(a)}\Big(a_{|g|}^{(g)}\Big)\Big]\Big)$$
$$\mathbf{h}^{(s)} = \text{ENC}^{(s)}\Big(\Big[\text{ENC}^{(a)}\Big(a_1^{(s)}\Big); \ldots; \text{ENC}^{(a)}\Big(a_{|s|}^{(s)}\Big)\Big]\Big)$$

The encoding of dialogue act specifications is realized by treating the dialogue acts as a sequence of tokens delimited by appropriate special words. In our implementation, we treat dialogue act specifications as sequences of tokens, e.g. `inform(food=indian)` becomes `inform`, `(`, `food`, `=`, `indian`, and `)`. Additionally, we use GloVe (Pennington et al., 2014) embeddings to obtain hints about the general token semantics. The utterances are encoded in a similar fashion, in which we apply the utterance encoder $\text{ENC}^{(u)}$ over arrays of word embeddings:

$$\mathbf{h}^{(u)} = \text{ENC}^{(u)}\Big(\Big[\mathbf{w}_1; \ldots; \mathbf{w}_{|u|}\Big]\Big)$$

Realization Networks

During the decoding step, the series of decoder networks successively decodes the latent variables into their respective feature spaces, with each successive decoding taking all latent variables up to the previous hierarchical level as inputs. Each decoder network is also conditioned on the global latent variable $\mathbf{z}^{(c)}$ and the turn-level hidden variable $\mathbf{h}_t$.

$$
\begin{aligned}
p_\theta\Big(\mathbf{r}_t\Big|\mathbf{v}_{<t}, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}\Big) &= \mathrm{DEC}^{(r)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}\Big) \\
p_\theta\Big(\mathbf{g}_t\Big|\mathbf{v}_{<t}, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}\Big) &= \mathrm{DEC}^{(g)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}\Big) \\
p_\theta\Big(\mathbf{s}_t\Big|\mathbf{v}_{<t}, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}, \mathbf{z}_t^{(s)}\Big) &= \mathrm{DEC}^{(s)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}, \mathbf{z}_t^{(s)}\Big) \\
p_\theta\Big(\mathbf{u}_t\Big|\mathbf{v}_{<t}, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}, \mathbf{z}_t^{(s)}, \mathbf{z}_t^{(u)}\Big) &= \mathrm{DEC}^{(u)}\Big(\mathbf{h}_t, \mathbf{z}^{(c)}, \mathbf{z}_t^{(r)}, \mathbf{z}_t^{(g)}, \mathbf{z}_t^{(s)}, \mathbf{z}_t^{(u)}\Big)
\end{aligned}
$$

5.3.4 Posterior Collapse

The tendency of VAE models to lose stabiliity during training and fail to effectively learn the data distribution is a well-known issue. This phenomenon is known as *posterior collapse*, and it has been perceived to be much more common and challenging to tackle for the NLP tasks than for the vision domain, due to the autoregressive nature of language modeling and the inherit difficulty of fitting linguistic representations into Gaussian priors (Xu and Durrett, 2018). The full details on the theory of posterior collapse in variational autoencoders are summarized in Appendix A.

In this work, we focus on the techniques to reduce the tendency of our model to ignore the encoder information and overly rely on the autoregressive hints.

Mutual Information Trick

In terms of information theory, the encoder performance level can be measured by the amount of information it passes through from the input data to the

latent variables using *mutual information* under the posterior distributions: $I_{p,q_\phi}(\mathbf{x}, \mathbf{z})$.

The mutual information trick involves modifying the VAE objective (Equation 2.3) to encourage the model to preserve the mutual information between the input data and the latent variables:

$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x} \mid \mathbf{z})] - D_{\mathrm{KL}}(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})) + I_{p,q_\phi}(\mathbf{x}, \mathbf{z}) \tag{5.1}$$

Since the KL-divergence term in the original ELBO can be decomposed into the KL-divergence between aggregate posterior $q_\phi(\mathbf{z})$ and the prior $p(\mathbf{z})$ and the mutual information term (Hoffman and Johnson, 2016), the third term in Equation 5.1 can be interpreted as counter-balancing the mutual information term in the KL-divergence term, which can rewritten as:

$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x} \mid \mathbf{z})] - D_{\mathrm{KL}}(q_\phi(\mathbf{z}) \| p(\mathbf{z})) \tag{5.2}$$

The detailed comparison of the proposed method and previous work on alleviating the inference collapse phenomenon is available in Appendex B.

Hierarchically-scaled Dropout Scheme

The common techniques for alleviating the inference collapse problem include (1) annealing the KL-divergence term weight during the initial stage of training and (2) employing word dropouts to the decoder inputs (Bowman et al., 2016). In a recent work, utterance-level dropouts were shown to be more effective than word dropouts (Park et al., 2018).

For our work, we notice that, due to the multi-level hierarchical structure, word or utterance dropouts are insufficient against inference collapse. However, employing dropouts for all feature inputs could deteriorate the learning of lower-level latent variables, as information dropouts stack multiplicatively along the vertical depth. Hence, we propose employing hierarchically-scaled dropout

scheme to effectively alleviate the information loss problem. For VHDA, we employ a dropout factor of 1.6, which is the ratio of the dropout rate of a particular type of feature over the dropout rate of another one depth above it. The initial dropout rate (speaker dropout rate) is set to 0.1. Similar ideas have been explored in the context of multi-class hierarchical classification (Wehrmann et al., 2018).

5.4 Experimental Results

This section describes the experimental settings for using VHDA as the generator in generative data augmentation framework for dialogue state tracking tasks. We conduct both quantitative and qualitative experiments to demonstrate that the samples generated by our model are not only beneficial towards training dialogue state trackers but exhibit variability and controlability at various levels of semantics.

5.4.1 Experimental Settings

The experimental protocol is based on the ones for used for sentence classification and spoken language understanding – the generator is trained once, but synthetic datasets are sampled multiple times with different seeds from the generator and, for each synthetic dataset, the dialogue state tracker is trained with the augmented dataset (the original training set and the synthetic dataset)) multiple times again to account for sampling and training variances (Figure 3.2). Training variances are relatively high, compared to other NLP tasks, because the dialogue state tracking task is evaluated using the joint goal accuracy, which is calculated based on the turn-level accuracy of accumulated predicted dialogue states. Specifically, under the same turn-level accuracy, accumulated dialogue state accuracy could fluctuate wildly, depending on the position of the

Metric	WoZ2.0	DSTC2	MWoZ-R	MWoZ-H
# Training Dialogues	600	1,612	1,265	1,282
# Validation Dialogues	200	506	52	120
# Test Dialogues	200	575	64	140
# Turns	4,472	23,354	10,980	6,752
# Tokens	50,264	199,431	132,991	95,562
Avg. Dialgue Length	7.45	14.49	8.68	5.27
Avg. Utternce Length	11.24	8.54	12.11	14.15
# Slots	4	8	24	24
# Values	99	212	335	194

Table 5.1: Statistics of goal-oriented dialogue datasets.

turn-level prediction errors in the dialogue [1]. During training of VHDAs, we employ KL-annealing period of 250,000 steps. For data synthesis, we employ ancestral sampling to generate from the empirical posterior distribution. The ratio of synthetic data to original data is 1.

Dataset

The standard framework for developing and evaluating dialogue state trackers has been provided by The Dialogue Systems Technology Challenge (DSTC) since the introduction of its first iteration (Williams et al., 2013). Under the framework, dialogue semantics (states and actions) are based on a task ontology, such as restaurant booking for DSTC2 (Henderson et al., 2014a) and WoZ2.0 (Wen et al., 2017) datasets. For this study, we analyze the quality of the generated task-oriented dialogues and their augmentation effect on dialogue state tracking specifically on four datasets: WoZ2.0 (or CamRest676), DSTC2, and two subsets of MultiWoZ (Budzianowski et al., 2018). A brief description

[1] In an extreme case, an error appearing in the first turn could invalidate all subsequent predictions.

of each dataset is as follows.

- **WoZ2.0**: Originally intended for the task of end-to-end dialogue systems (Wen et al., 2017), WoZ2.0 is a domain-specific goal-oriented dialogue corpus collected and annotated by crowdworkers using the Wizard-of-Oz (WOZ) methodology (Kelley, 1984). During collection, the main goal of the system was to assist users finding a restaurant in the Cambridge, UK area, according to the user requests and constraints that had to be communicated by the system through text. This dataset was collected using DSTC2 as the benchmark. Although the dataset is the smallest among the goal-oriented dialogue corpora, it has been studied extensively in the past, and thus the dataset provides a strong ground for comparative study of dialogue state tracking.

- **DSTC2**: This dataset was originally released as one of the tasks in Dialogue State Tracking Challenge (Henderson et al., 2014a). The structure of DSTC2 is considerably more complex than most other corpora, due to the extensive consideration gone through the collection and design process, including the consideration of automatic speech recognition (ASR) and dialogue manager. Different types of ASR models and dialogue managers have been considered during the collection, allowing models developed on top of the dataset to be trained and evaluated with different simulated environments. Note that DSTC2 had been collected from human-machine interactions, in contrast to human-human interactions as other datasets had been collected from. Hence, the system responses in DSTC2 are monotonic and lack linguistic variety. As one of the earlier fully annotated and structured dialogue corpora, the dataset has been the landmark corpus for dialogue state tracking.

- **MultiWoZ**: This relatively large goal-oriented dialogue corpora is currently the state-of-the-art of publicly available structured dialogue datasets in terms of size and domain diversity. Similar to WoZ2.0 and DSTC2, MultiWoZ (Budzianowski et al., 2018) contains task-oriented dialogues but the domain of the tasks is not limited to single item. As the name suggests, dialogues could switch from the hotel-booking domain to another such as taxi-calling and vice versa. However, in this paper, we are mainly focused on dialogue modeling for single domain scenarios, thus we extract single-domain dialogues from the dataset by checking all turns for active dialogue acts. Since the dataset does not explicitly denote active domains of a particular dialogue, we scan all turns in the dialogue for any activated domains and we consider a domain active if its related dialogue acts have been labeled in any of the turns. We extracted two single-domain subsets (`MultiWoZ-R` and `MultiWoZ-H`) that involve restaurant booking and hotel reservation tasks respectively. The statistics of the extracted datasets are shown in Table 5.1.

- **DialEdit**: This dialogue corpora contains conversations between a user and a wizard on the topic of image editing (Manuvinakurike et al., 2018).

The full statistics of the datasets are shown in Table 5.1.

Evaluation

Adhering to the evaluation framework proposed for past DSTC challenges, we utilize the following three evaluation measures to quantify the dialogue state tracking performance:

- **Turn Inform Accuracy**: the prediction accuracy of the user's turn-level dialogue acts, where the act type is `inform`. This is calculated by dividing the number of correctly predicted turns with the total number of turns.

- **Turn Request Accuracy**: the prediction accuracy of the user's turn-level dialogue acts, where the act ytpe is `request`. This is calculated by dividing the number of correctly predicted turns with the total number of turns.

- **Joint Goal Accuracy**: the prediction accuracy of the user's goals. User goals can be derived from the user's `inform`-type dialogue acts by accumulating slot-value pairs over the course of the dialogue. If an `inform` dialogue act has a slot that has already been specified in one of the previous turns, then the latter slot-value pair will overwrite the previous ones. For example, if the user goal in the previous turn contains `food=indian` and the current user turn specifies a different food type such as `italian`, then the user goal in the current turn is updated to `food=italian`. Joint goal accuracy can be calculated by dividing the total number of turns where the user goals are correctly identified with the total number of turns.

Dialogue State Tracker

Recent trend in deep learning-based dialogue state tracker is to reformulate the problem of multi-label classification problem as dynamic multiple binary classification problems (Mrkšić et al., 2017) by encoding the target dialogue act with a general-purpose pretrained word embeddings, such as GloVe (Pennington et al., 2014). This is largely motivated by the sparseness of dialogue act labels, where the number of possible combinations of slot and values far exceeds the number of annotated turns in a typical domain-specific dataset. Recent deep learning-based dialogue state trackers have been largely based on Neural Belief Tracker (Mrkšić et al., 2017), such as (Kim et al., 2018a; Zhong et al., 2018; Nouri and Hosseini-Asl, 2018; Wu et al., 2019). In this work, we select two state-of-the-art dialogue state trackers as our baseline classifiers: GLAD (Zhong et al.,

2018) and GCE (Nouri and Hosseini-Asl, 2018). We also consider a simpler RNN-based dialogue state tracker to explore the effect of data augmentation on less expressive dialogue state trackers.

Due to the difference in implementation environments, we could not reliably reproduce results for GLAD and GCE reported by the respective authors. Hence, we have made small modifications to the model architecture described in the original papers and denoted them with GLAD$^+$ and GCE$^+$) respectively. The description of each dialogue state tracker is as follows.

- **GLAD**: Motivated by the lack of parameter sharing across different slots in NBT, this dialogue state tracker had been proposed by (Zhong et al., 2018) to use a common sequence encoder architecture that jointly extract globally and locally aware representations for all sequence-like features such as dialogue acts and utterances.

- **GLAD$^+$**: This dialogue state tracker is a variant of GLAD, where we use FastText (Joulin et al., 2017) in addition to the GloVe and Kazuma (Hashimoto et al., 2017) embeddings employed by the original model. Additionally, a dropout layer (dropout rate 0.2) is added after the word embeddings layer to strengthen the model against noises introduced by synthetic samples.

- **GCE**: However, GLAD suffers from a scalability problem with respect to the number of slots, as the model dedicates an independent set of parameters for each slot type. GCE (Nouri and Hosseini-Asl, 2018) proposed an improvement on the previous state-of-the-art, where the common sequence encoder architecture (GLAD encoder) no longer manages two separate set of parameters for global and local-related features and uses a unified set of parameters conditioned by slot type embeddings (which are also encoded using the common sequence architecture). This

GDA	Model	WoZ2.0		DSTC2		MWoZ-R		MWoZ-H		DialEdit	
		Goal	Req	Goal	Req	Goal	Inf	Goal	Inf	Goal	Req
-	GLAD$^+$	87.8	**96.8**	74.5	96.4	58.9	76.3	33.4	58.9	35.9	96.7
VHDA	GLAD$^+$	**88.4**	96.6	**75.5**‡	**96.8**†	**61.5**†	**77.4**	**37.8**‡	**61.3**‡	**37.1**†	**96.8**
-	GCE$^+$	88.3	97.0	74.8	96.3	60.5	76.7	36.5	61.0	36.1	96.6
VHDA	GCE$^+$	**89.3**‡	**97.1**	**76.0**‡	**96.7**†	**63.3**	**77.2**	**38.3**	**63.1**†	**37.6**†	**96.8**
-	RNN	74.5	96.1	69.7	96.0	41.1	69.4	25.7	55.6	35.8	96.6
VHDA	RNN	**78.7**‡	**96.7**‡	**74.2**†	**97.0**‡	**49.6**†	**73.4**†	**31.0**†	**59.7**†	**36.4**†	**96.8**

$^\dagger p < 0.1$ $^\ddagger p < 0.01$

Table 5.2: Results of data augmentation using VHDA for dialogue state tracking on various datasets and trackers.

relatively small change had a huge impact on the model, hugely improving the time and space complexities and significantly improving the dialogue state tracking performance.

- **GCE$^+$**: Similar to **GLAD$^+$**, this variant also uses an additional source of pretrained word embeddings (FastText) and employs a dropout layer after the word embeddings.

- **RNN**: For this simple model, we largely follow the architectural design of GCE, but we consider a simpler common sequence encoder design, where variable length sequences are encoded using a set of bidirectional LSTM cells (Hochreiter and Schmidhuber, 1997) only (without the self-attention and local-conditioning mechanism). The purpose of this model is to demonstrate the effectiveness of data augmentation in settings where the expressive power of the model is not optimal.

5.4.2 Data Augmentation Results

Main Results

We conduct data augmentation experiments to explore the effect of augmenting dialogue state tracking datasets using synthetic samples generated from VHDA and report the results on various datasets and trackers as shown in Table 5.2.

GDA	Tracker	WoZ2.0		DSTC2	
		Goal	**Request**	**Goal**	**Request**
VHDA w/o goal	GLAD$^+$	86.5	**96.9**	74.7	**97.0**
VHDA	GLAD$^+$	**88.4**	96.6	**75.5**	96.8
VHDA w/o goal	GCE$^+$	86.4	96.3	75.5	96.7
VHDA	GCE$^+$	**89.3**	**97.1**	**76.0**	**96.7**
VHDA w/o goal	RNN	77.8	96.4	71.2	**97.2**
VHDA	RNN	**78.7**	**96.7**	**74.2**	97.0

Table 5.3: Comparison of data augmentation results between VHDA and VHDA without explicit goal tracking.

Results show that generative data augmentation for dialogue state tracking is a viable strategy for improving existing DST models without modifying the classifier and only augmenting the dataset using synthetic samples from our model. Regardless of the tracker model and the dataset, improvements were observed at a statistically significant level in most experimental settings. Surprisingly, due to the high-variance nature of the joint-goal metric for evaluating dialogue state tracking performance, some of the improvements with relatively largin margins (e.g. GCE$^+$ on MultiWoZ restaurant dataset) had weaker statistical significance. We posit that in correspondence to the high-variance, conducting much larger number of trials should improve the statistical confidence; however, our experiments were constrained by limited computational resources relative to the huge number of combinatory settings from all of the experiments. The full results of the data augmentation experiments are included in Appendix C.

Effect of Joint Goal Tracking

To evaluate and compare the effect of explicitly tracking goal annotations during training of dataset modeler, we train a variant of VHDA model, where the goal tracking mechanism is removed from the generator (denoted by *VHDA w/o goal*). In such a model, the latent variables for goal tracking $\mathbf{z}^{(g)}$ are detached from the graphical model and the utterance latent variables $\mathbf{z}^{(u)}$ are conditioned on $\mathbf{z}^{(c)}$, $\mathbf{z}^{(r)}$ and $\mathbf{z}^{(s)}$ only. Synthetic samples are generated from the variant model and the effect of data augmentation on dialogue state tracking is compared with the original VHDA model. We conduct experiments on WoZ2.0 and DSTC2 datasets (Table 5.3) and the results show that VDHA without explicit goal tracking suffers in joint goal accuracy but performs better in turn request accuracy at certain combinations of tracker and dataset. We conjecture that explicit goal tracking helps the model to reinforce the long-term goals of the dialogue participants; however, the model achieves better long-term tracking in the minor expense of short-term dialogue act tracking, such as request tracking, which only requires the model to only consider the preceding utterance.

VHDA Ablation Studies

As discussed in Section 5.3.4, the autoregressive nature of the decoder in our model poses the risk of the decoder ignoring the encoder signals and simply rely on the autoregressive inputs to predict each feature, essentially falling into the inference collapse phenomenon. In order to reduce the decoder's reliance on autoregressive signals, we have employed hierarchically scaled feature dropouts (word, utterance, goal, state, and speaker dropouts) (Park et al., 2018; Serban et al., 2017) and the mutual information trick, which modifies the VAE objective to encourage the model to preserve the encoder's information flow. We conduct ablation studies to show the effects of employing different combinations of the techniques that reduce the occurrence of inference collapse (Table 5.4). We

GDA	Dropout	Objective	$KL^{(c)}$	Tracker	WoZ2.0	
					Goal	Request
-	-	-	-	GCE^+	88.3±0.7	97.0±0.2
VHDA	0.00	Standard	5.63	GCE^+	84.1±0.9	95.9±0.6
VHDA	0.00	Modified	5.79	GCE^+	86.0±0.2	96.1±0.2
VHDA	0.25	Standard	10.44	GCE^+	88.5±1.4	96.9±0.1
VHDA	0.25	Modified	11.31	GCE^+	88.9±0.4	97.0±0.2
VHDA	0.50	Standard	14.68	GCE^+	88.6±1.0	96.9±0.2
VHDA	0.50	Modified	16.33	GCE^+	89.2±0.8	96.9±0.2
VHDA	hierarchical	Standard	14.34	GCE^+	88.2±1.0	97.1±0.2
VHDA	hierarchical	Modified	16.27	GCE^+	**89.3**±0.4	**97.1**±0.2

Table 5.4: Ablation studies for VDHA using GCE^+ as the baseline dialogue state tracker.

report the KL-divergence term of the conversation latent variables $\mathbf{z}^{(c)}$ of the test set data, along with the data augmentation results following the standard data augmentation protocol. The experimental results support our hypothesis that the regularization measures have drastically reduced the encoder from collapsing, maintaing the magnitude of the KL-divergence term at higher-levels, while achieving better data augmentation results due to improved exploratory power from having healthier encoders. The worst results were shown by the VHDA model without any measures, while the best results were achieved by applying hierarchically scaled dropout scheme in combination with the modified VAE objective.

Note that the KL-divergence term for the hierarchically scaled dropouts was not higher than next best performing measures (16.27 and 16.33), suggesting that having higher KL-divergence terms do not always correlate with the data augmentation performance or the generation quality. On the contrary, we interpret the results as the hierarchical scaling improving the decoder's ability

Model	WoZ2.0			DSTC2		
	BLEU	**ROUGE**	**ENT**	**BLEU**	**ROUGE**	**ENT**
VHCR	0.301	0.476	0.193	0.494	0.680	0.153
VHDA w/o goal	0.307	0.473	**0.195**	0.590	0.743	**0.162**
VHDA	**0.326**	**0.499**	0.193	**0.637**	**0.781**	0.154

Table 5.5: Evaluation of models on language quality and diversity.

to generate more coherent data that benefits data augmentation, while encouraging the latent variables to be encoded in a tighter sphere (unit Gaussian prior).

5.4.3 Intrinsic Evaluation - Language Evaluation

To gain deeper insight into the generation capability of VHDA, we compare the language quality and diversity of generated dialogues with those generated by the previous state-of-the-art model for dialogue generation (Park et al., 2018). Following the evaluation protocol employed by previous work (Bak and Oh, 2019; Li et al., 2016; Wen et al., 2017), we use BLEU score (Papineni et al., 2002) post-processed with the smoothing-7 method (Chen and Cherry, 2014) and ROUGE-L f1-score (Lin, 2004) to evaluate the linguistic quality of generated utterances and utterance-level unigram cross-entropy (Serban et al., 2017) (with respect to the training corpus distribution) for evaluating the information-level and linguistic diversity of the utterances. Note that our evaluation is conducted on the dialogue-level [2] rather than on the turn-level, hence turn-level utterance modeling approaches (Serban et al., 2017) are excluded from the comparison. Results are shown in Table 5.5.

Compared to the previous state-of-the-art model on conversation modeling,

[2] meaning that the utterances are generated and evaluated in the unit of complete dialogues.

VHDA is able to generate utterances of higher quality in terms of BLEU and ROUGE scores. This supports our hypothesis that learning jointly with the explicit annotation of underlying dialogue structure helps with generating more realistic utterances and thereby synthesizing realistic dialogues. In terms of linguistic diversity, it is well-known that the generation quality and diversity have a trade-off relationship (Huszár, 2015), hence it is expected that our model is not able to generate the most diverse responses among the other models. However, we argue that our model is able to generate utterances of best quality without scrificing linguistic diversity, compared to the language-based model.

5.4.4 Qualitative Results

In order to gain deeper insight into the exploratory power of our model, we dedicate this subsection to the qualitative analysis on generated samples produced by VHDA in multiple aspects.

We first examine the linguistic variability of the generated samples by analyzing variations of the linguistic patterns of random samples. In the following subsection, we demonstrate that the conversational latent variables encode holistic features of dialogues and exhibit controlability when novel dialogs are decoded from latent variables sampled from the space.

Linguistic Variations

We notice our model generates samples with several different levels of linguisitc variations. On the *word-level*, we observe that phrases such as "I'd like to" had been paraphrased to similar forms that preserve the dialogue semantics, such as "I want to". In other cases, "is there anything else I can help you with" was parapharased to "is there anything else I can help you find" and simple phrases that express gratitude were used interchangably (e.g. "thank you . bye", "thank you . bye bye"). The word-level linguistic variability of generative models had

Turn	Speaker	Utterance
1	User	i am looking for a moderately priced restaurant in the ...
2	Wizard	<name> <location> ... would you like their information
3	User	i do nt want it . show me another one .
4	Wizard	the restaurant <name> is a moderately priced ....
5	User	what is the address please
6	Wizard	<address>
7	User	thanks . could i get the phone number , too
8	Wizard	restaurant <name> s phone number is <number>
9	User	thank you , goodbye

Table 5.6: A example of utterance-level variation synthesized by VHDA. The utterance added by our model is underlined.

been observed in language-based VAE models (Gupta et al., 2018; Yang et al., 2019a), thus exhibition of such characteristics shows that our model, trained to jointly generate language and relevant annotations, is capable of learning paraphrastic linguistic patterns as other pure-language generative models.

We also observe context-aware *utterance-level* variations as shown in Table 5.6. Our model generates variations of utterances (marked by underline) that preserve the overall dialogue semantics. The example in the table shows that the utterance "would you like their information", which was absent in the original dialogue piece, was added by our model, complementing the subsequent response "i don't want it", improving the dialogue fluency. This suggests that the model is capable of comprehending basic level of dialogue semantics.

Finally, we also observe *dialogue-level* variations in the synethetic samples, noting the noises in the dialogue progression while preserving the user goal, as shown in Table 5.8. The synthetic sample has a dialogue progression where the user seems requests for a cheap restaurant at first, then in the later part of the dialogue, the users specifies an additional constraint of food type (Vietnamese).

Turn	Speaker	Utterance
1	User	what s good and cheap ?
2	Wizard	there are <number> selections in the cheap price range . what kind of food were you looking for ?
3	User	well how many in the west ?
4	Wizard	there are two cheap options in the west side one is an italian restaurant and the other is vietnamese .
5	User	hmm , what s the name of the vietnamese restaurant ?
6	Wizard	<name>
7	User	is there anything just plain american ?
8	Wizard	there are no american places in the west .
9	User	can i have the phone number and address of <name>
10	Wizard	of course . <name> is located at <location> and can be reached at <phone>
11	User	thank you !

Table 5.7: A real dialogue sample (#27) in WoZ2.0.

Turn	Speaker	Utterance
1	User	i m looking for a cheap restaurant .
2	Wizard	i m sorry there are no restaurants matching your preferences .
3	User	how about vietnamese food ?
4	Wizard	yes there are four restaurants with one cheap price range . do you have a preference ?
5	User	i do nt care
6	Wizard	how about <name> in the <location> .
7	User	can i have the address and postocde ?
8	Wizard	the address is at <location> their phone is <phone>
9	User	thank you goodbye .

Table 5.8: A synthetic dialogue sample similar to the real dialogue #27.

This dialogue progression shares the same pattern as the real sample #27 (Table 5.7), which we can compare our synthetic sample with to examine the difference. From the comparison, we note two things: (1) our model is able to introduce noises to the dialogue progression while preserving the dynamics of belief states (`price=cheap` → `food=vietnamese`) and linguistic consistency; however, (2) as noted in prior works on text generation using VAEs (Shin et al., 2019; Yoo et al., 2019), our model also exhibits the tendency to generate simpler samples, as we observe that the generated sample in Table 5.8 is shorter and less complex in terms of dialogue dynamics than the comparable real sample.

Additionally, in the example above, we note that the model has introduced a subtle logical inconsistency that spans several turns. Specifically, the system initially does not find any results satisfying the user's request for a cheap restaurant from turn 1, but the system was able to do so for the same request with an *additional* constraint of `food=vietnamese` after few turns later. Such logical fallacies are not critical in the task of data augmentation for dialogue state tracking, as the usual problem formulation of DST is to predict dialogue states given the user utterance in the current turn and the system actions in the turn preceding the current turn; however, for other downstream applications, such fallacies might not be ideal and an improvement to the model might be desired.

Turns	Speaker	Utterance	Goal	Turn Act
1	User	i 'm looking for a mediterranean place for any price . what is the phone and postcode ?	inform(food=mediterranean) inform(price=dont care)	inform(food=mediterranean) inform(price=dont care) request(slot=phone, slot=postcode)
2	Wizard	i found a few places . the first is <name> with a phone number of <number> and postcode of <postcode>		
3	User	That will be fine . thank you .	inform(food=mediterranean) inform(price=dont care)	

Table 5.9: The first anchor point in the $\mathbf{z}^{(c)}$-interpolation experiment.

Turns	Speaker	Utterance	Goal	Turn Act
1	User	i want to find a cheap restaurant in the north part of town .	inform(area=north) inform(price range=cheap)	inform(area=north) inform(price range=cheap)
2	Wizard	what food type are you looking for ?		request(slot=food)
3	User	any type of restaurant will be fine .	inform(area=north) inform(food=dontcare) inform(price range=cheap)	inform(food=dontcare)
4	Wizard	the <place> is a cheap indian restaurant in the north . would you like more information ?		
5	User	what is the number ?	inform(area=north) inform(food=dontcare) inform(price range=cheap)	request(slot=phone)
6	Wizard	<place> 's phone number is <number> . is there anything else i can help you with ?		
7	User	no thank you . goodbye .	inform(area=north) inform(food=dontcare) inform(price range=cheap)	

Table 5.10: A sample generated from the midpoint between two latent variables in the $\mathbf{z}^{(c)}$ space encoded from two anchor data points.

Turns	Speaker	Utterance	Goal	Turn Act
1	User	i am looking for a cheap restaurant in the north part of town .	inform(area=north) inform(price range=cheap)	inform(area=north) inform(price range=cheap)
2	Wizard	there are two restaurants that fit your criteria would you prefer italian or indian food ?		request(slot=food)
3	User	let s try indian please	inform(area=north) inform(price range=cheap) inform(food=indian)	inform(food=indian)
4	Wizard	<name> serves indian food in the cheap price range and in the north part of town . is there anything else i can help you with ?		
5	User	what is the name of the italian restaurant ?	inform(area=north) inform(price range=cheap) inform(food=indian)	inform(food=italian) request(slot=name)
6	Wizard	<name>		
7	User	what is the address and phone number ?	inform(area=north) inform(price range=cheap) inform(food=indian)	request(slot=address) request(slot=phone)
8	Wizard	the address for <name> is <address> and the phone number is <phone> .		
9	User	thanks so much .	inform(area=north) inform(price range=cheap) inform(food=indian)	

Table 5.11: The second anchor point in the $\mathbf{z}^{(c)}$-interpolation experiment.

$\mathbf{z}^{(c)}$-Interpolation

We conduct $\mathbf{z}^{(c)}$-interpolation experiments to demonstrate that our model is able to generalize the dataset space and learn to decode plausible samples from unseen latent space. First, we encode two random dialog samples $\mathbf{x}'$ and $\mathbf{x}''$ onto the latent space of dialog $\mathbf{z}^{(c)}$ using the our model's approximate posterior. The conversation latent vectors of the two data points are interpolated and subject to equidistant samples $\mathbf{z}_1^{(c)}, \ldots, \mathbf{z}_n^{(c)}$. Then, we observe the decoder outputs generated from the interpolated latent variable samples (Table 5.9-5.11). The first anchor sample is shown in Table 5.9 and the second anchor sample is shown in Table 5.11. The generated sample (shown in Table 5.10) demonstrates that our model is able to generalize key dialogue features, such as the user goal and the dialogue length, to generate novel dialogues. One specific example of such generalization is that since, for the first sample, the user's goal is to search for a Mediterranean restaurant and, for the second sample, the user's goal is an Indian restaurant, the midpoint between the two latent variables results in a novel dialogue with no specific preference for food type (`food=dontcare`). This behavior by our model supports our hypothesis that the model is capable of generalizing the latent conversation space and generate novel and coherent synthetic samples that is beneficial towards data augmentation.

5.5 Summary

In this chapter, we have proposed a novel VAE-based architecture that is both hierarchical and recurrent in order to accurately capture the semantics of complex datasets, such as fully annotated goal-oriented dialogue datasets. Due to the highly autoregressive nature of our model's decoder, the model was prone to inference collapse. Hence, we devised and employed a simple technique called the mutual information trick, based on the manipulation of VAE training objec-

tive, for reducing the chance of inference collapse occurrence without sacrificing the performance of the decoder. Utilizing the mutual information trick, our proposed model VHDA was able to achieve significant improvement in the task of generatively augmentating training datasets for dialogue state tracking, using various competitive dialogue state trackers on various domains. Through qualitative analysis of the generated samples, we have gained a deeper insight into the characteristics of linguistic variations and assured that the latent variables are controllable, allowing us to better understand the exploratory mechanism of our model.

Recent works on end-to-end goal-oriented dialogue systems have proposed using reinforcement learning to incorporate system databases and their interaction with the system responses in an end-to-end fashion (Wen et al., 2017; Williams et al., 2017; Lei et al., 2018). As our qualitative analysis on one of the generated samples has unearthed the possibility of our model introducing long-term logical inconsistencies, an end-to-end variational modeling methodology that incorporates knowledgebase is inevitable for achieving a truly dialogue-generative model. As future work, we wish to explore an end-to-end knowledge-based self-supervised generative model that is capable of generating more coherent goal-oriented dialogues. We also wish to explore how goal-oriented dialogue modeling could benefit other dialogue-related tasks, such as user simulation and data construction, and not only dialogue state tracking.

Chapter 6

Conclusion

6.1 Summary

Data augmentation has been one of the well-adopted techniques for boosting performances of supervised learning tasks without the knowledge of the model being used. Advances in generative models in recent years due to the wide adoption of deep variational models has opened up the possibility of leveraging these models for generating novel samples that could improve supervised learning. We formalized the notion of generative data augmentation, where samples generated from latent variable models are used as augmentational material for the training set. This dissertation started off with an in-depth survey and theoretical analysis of data augmentation and subsequently proposed utilizing deep latent variable models as the generator backbone in generative data augmentation for NLP tasks. We proposed three distinct VAE-based models for modeling joint distributions of multi-modal (text and annotations) datasets, achieving improvements in the respective tasks on various competitive downstream models and datasets.

However, autogressive VAEs are notorious for their hyperparameter sensitivity during training and the susceptibility to posterior collapse when learning text datasets due to the teacher-forcing strategy employed for training linguistic sequences. With the need to expand VAEs to accomodate structured annotations as additional latent variables, training these complex conditional VAEs became even a bigger challenge. We proposed three training algorithms and

policies that could largely reduce the hyperparameter search space and make training of VAEs for NLP datasets much more feasible. We successfully applied the techniques to the training of VAEs for sentence classification datasets and dialogue state tracking datasets, further improving the generative data augmentation results.

In terms of task exploration, we conducted experiments on spoken language understanding datasets to investigate whether VAE-based latent variable models are capable of learning tasks that are inherently require joint-learning (intent classification and sequence tagging). For the dialog state tracking problem, we took on the challenge of modeling complex annotated datasets by designing a hierarchical and recurrent VAE-based model for jointly learning goal-oriented dialogue distributions. Despite the complexity, we showed that our model, applied with the proposed training technique for VAEs, was able to learn appropriate representations into the dedicated latent variables. Ablation studies in spoken language understanding experiments unearthed the existence of a hidden ceiling of the potential benefit that generative data augmentation can bring to the downstream models. The studies on artifical data scarcity experiments highlighted how generative data augmentation could help alleviate resource-scarce scenarios. Furthermore, another set of ablation studies also revealed the relationship between augmentational ratio and the improvement margin.

6.2 Limitations

There are few limitations of the approaches proposed in this work.

First, generative data augmentation requires two individual training stages, which could impose heavy computational cost depending on the application. However, generative data augmentation allows direct interpretation of intermediate augmentation results. Thus, it opens up new ways of utilizing the generated samples such as the assistance of human construction of NLP datasets.

Second, due to the fact that deep learning models are generally data-hungry, certain level of data abundance is required for the latent variable model to function properly as an ideal generator for data augmentation. This statement may seem contradictory to one of the motivations of employing generative data augmentation - to combat data scarcity (Chapter 4). Although we did not observe any performance deterioration during artificial data scarcity experiments, we conjecture that, below certain levels of available training dataset size, the generative model could significantly suffer from overfitting.

Third, developing and implementing task-specific latent variable models is costly, which is the main obstacle to standardized adoption of generative data augmentation.

6.3 Future Work

In lieu of the limitations, we propose several directions this work could take in order to further advance the GDA technique for NLP. Since the regularization effect is partly attributed to the assumed distribution choice for the prior, choosing better distributional families for the prior and the posterior of VAE models could allow further improvements into the generative quality. Some alternative distribution families include Von Mises Fischer (VMF) and arbitrary distribution modeling using normalizing flow.

On the other hand, the prior knowledge of whether GDA would benefit the resulting model or not could be useful for performing cost analysis and decision making. By collecting more samples on the positive cases of generative data augmentation in various NLP tasks, it would be possible create a predictor that predicts the magnitude of the benefit a dataset would enjoy from generative data augmentation based on the dataset features.

As mentioned in the limitations, the intermediate generation samples could also be used for supervised data collection. It would be interesting to research

how artificial novel samples could help annotators and data collectors in improving diversity and reducing biases in the constructed datasets (Pannucci and Wilkins, 2010).

Bibliography

Attention-based recurrent neural network models for joint intent detection and slot filling. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 685–689, 2016.

R. Al-Rfou, B. Perozzi, and S. Skiena. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CoNLL)*, pages 183–192, 2013.

A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken elbo. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 159–168, 2018.

A. Antoniou, A. Storkey, and H. Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

J. Bak and A. Oh. Variational hierarchical user-based conversation model. 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 1941–1950, 2019.

C. Banea, R. Mihalcea, and J. Wiebe. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. In *Language Resources and Evaluation Conference*, volume 8, pages 2–764, 2008.

E. Barnard, M. Davel, and C. v. Heerden. Asr corpus design for resource-scarce languages. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.

Y. Bengio. The consciousness prior. *arXiv preprint arXiv:1709.08568*, 2017.

L. Besacier, E. Barnard, A. Karpov, and T. Schultz. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56: 85–100, 2014.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research (JMLR)*, 3(Jan):993–1022, 2003.

P. Blunsom, T. Cohn, and M. Osborne. A discriminative latent variable model for statistical machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL: HLT)*, pages 200–208, 2008.

S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning (SIGNLL)*, pages 10–21, 2016.

P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gasic. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5016–5026, 2018.

C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

M. Cai, Y. Shi, J. Kang, J. Liu, and T. Su. Convolutional maxout neural networks for low-resource speech recognition. In *The 9th International Symposium on Chinese Spoken Language Processing*, pages 133–137. IEEE, 2014.

B. Chen and C. Cherry. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, 2014.

X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016a.

Y.-N. Chen, D. Hakanni-Tür, G. Tur, A. Celikyilmaz, J. Guo, and L. Deng. Syntax or semantics? knowledge-guided joint semantic frame parsing. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 348–355. IEEE, 2016b.

J. Choi, T. Kim, and S.-g. Lee. A cross-sentence latent variable model for semi-supervised text sequence matching. *arXiv preprint arXiv:1906.01343*, 2019.

A. Conneau and D. Kiela. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, 2018.

A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680, 2017.

C. Cremer, X. Li, and D. Duvenaud. Inference suboptimality in variational autoencoders. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1086–1094, 2018.

E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 113–123, 2019.

T. Dao, A. Gu, A. J. Ratner, V. Smith, C. De Sa, and C. Ré. A kernel theory of modern data augmentation. *arXiv preprint arXiv:1803.06084*, 2018.

T. Dao, A. Gu, A. Ratner, V. Smith, C. De Sa, and C. Re. A kernel theory of modern data augmentation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 1528–1537, 2019.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

L. Dybkjær and W. Minker. *Recent Trends in Discourse and Dialogue*, volume 39. Springer Science & Business Media, 2008.

L. El Asri, H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, and K. Suleman. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 207–219, 2017.

O. Fabius and J. R. van Amersfoort. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014.

M. Fadaee, A. Bisazza, and C. Monz. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 567–573, 2017.

W. Fedus, I. Goodfellow, and A. M. Dai. Maskgan: Better text generation via filling in the _. *arXiv preprint arXiv:1801.07736*, 2018.

Z. Fu, X. Tan, N. Peng, D. Zhao, and R. Yan. Style transfer in text: Exploration and evaluation. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL: HLT)*, volume 2, pages 753–757, 2018.

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

X. Gu, K. Cho, J.-W. Ha, and S. Kim. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. *arXiv preprint arXiv:1805.12352*, 2018.

D. Guo, G. Tur, W.-t. Yih, and G. Zweig. Joint semantic utterance classification and slot filling with recursive neural networks. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 554–559. IEEE, 2014.

A. Gupta, A. Agarwal, P. Singh, and P. Rai. A deep generative framework for paraphrase generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

K. Hashimoto, Y. Tsuruoka, R. Socher, et al. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1923–1933, 2017.

J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*, 2019.

C. T. Hemphill, J. J. Godfrey, and G. R. Doddington. The atis spoken lan-

guage systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

M. Henderson, B. Thomson, and J. D. Williams. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, 2014a.

M. Henderson, B. Thomson, and S. Young. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, 2014b.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

M. D. Hoffman and M. J. Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Proceedings of the NIPS Workshop in Advances in Approximate Bayesian Inference*, volume 1, 2016.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research (JMLR)*, 14(1):1303–1347, 2013.

Y. Hou, Y. Liu, W. Che, and T. Liu. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 1234–1245, 2018.

W.-N. Hsu, Y. Zhang, and J. Glass. Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In

Proceedings of the 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pages 16–23. IEEE, 2017.

M. Hu and B. Liu. Opinion extraction and summarization on the web. In *Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI)*, 2006.

Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing. Toward controlled generation of text. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1587–1596, 2017.

Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

F. Huszár. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*, 2015.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh. Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 427–431, 2017.

K. Kafle, M. Yousefhussien, and C. Kanan. Data augmentation for visual question answering. In *Proceedings of the 10th International Conference on Natural Language Generation (INLG)*, pages 198–202, 2017.

T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019.

J. Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41, 1984.

A.-Y. Kim, H.-J. Song, and S.-B. Park. A two-step neural dialog state tracker for task-oriented dialog processing. *Computational Intelligence and Neuroscience*, 2018, 2018a.

Y. Kim, S. Wiseman, A. C. Miller, D. Sontag, and A. M. Rush. Semi-amortized variational autoencoders. *arXiv preprint arXiv:1802.02550*, 2018b.

Y. Kim, A. M. Rush, L. Yu, A. Kuncoro, C. Dyer, and G. Melis. Unsupervised recurrent neural network grammars. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL: HLT)*, pages 1105–1117, 2019.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

D. Klein and C. D. Manning. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL)*, page 478. Association for Computational Linguistics, 2004.

T. Ko, V. Peddinti, D. Povey, and S. Khudanpur. Audio augmentation for speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

G. Kurata, B. Xiang, and B. Zhou. Labeled data generation with encoder-decoder lstm for semantic slot filling. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 725–729, 2016a.

G. Kurata, B. Xiang, B. Zhou, and M. Yu. Leveraging sentence-level information with encoder lstm for semantic slot filling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2077–2083, 2016b.

S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, pages 2267–2273. AAAI Press, 2015.

W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, and D. Yin. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1437–1447, 2018.

J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. 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 (NAACL: HLT)*, pages 110–119, 2016.

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

C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, 2004.

L. Lu, A. Ghoshal, and S. Renals. Regularized subspace gaussian mixture models for cross-lingual speech recognition. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 365–370. IEEE, 2011.

Y. Ma, J.-j. Kim, B. Bigot, and T. M. Khan. Feature-enriched word embeddings for named entity recognition in open-domain conversations. In *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6055–6059. IEEE, 2016.

R. Manuvinakurike, J. Brixey, T. Bui, W. Chang, R. Artstein, and K. Georgila. Dialedit: Annotations for spoken conversational image editing. In *Proceedings 14th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 1–9, 2018.

G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu, and G. Zweig. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 530–539, 2015.

T. Miyato, A. M. Dai, and I. Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

N. Mrkšić, D. Ó. Séaghdha, T.-H. Wen, B. Thomson, and S. Young. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th*

Annual Meeting of the Association for Computational Linguistics (ACL), pages 1777–1788, 2017.

K. P. Nguyen, C. C. Fatt, A. Treacher, C. Mellema, M. H. Trivedi, and A. Montillo. Anatomically-informed data augmentation for functional mri with applications to deep learning. *arXiv preprint arXiv:1910.08112*, 2019.

E. Nouri and E. Hosseini-Asl. Toward scalable neural dialogue state tracking model. *arXiv preprint arXiv:1812.00899*, 2018.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, pages 1345–1359, 2010.

B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL)*, page 271. Association for Computational Linguistics, 2004.

B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 115–124. Association for Computational Linguistics, 2005.

C. J. Pannucci and E. G. Wilkins. Identifying and avoiding bias in research. *Plastic and reconstructive surgery*, 126(2):619, 2010.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 311–318. Association for Computational Linguistics, 2002.

Y. Park, J. Cho, and G. Kim. A hierarchical latent structure for variational conversation modeling. In *Proceedings of the 2018 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL: HLT), pages 1792–1801, 2018.

J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

C. R. Perrault, J. F. Allen, and P. R. Cohen. Speech acts as a basis for understanding dialogue coherence. In *Proceedings of the 1978 Workshop on Theoretical Issues in Natural Language Processing*, pages 125–132. Association for Computational Linguistics, 1978.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners.

A. Ragni, K. Knill, S. Rath, and M. Gales. Data augmentation for low resource languages. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 810–814, 2014.

A. J. Ratner, H. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré. Learning to compose domain-specific transformations for data augmentation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3236–3246, 2017.

A. Razavi, A. v. d. Oord, B. Poole, and O. Vinyals. Preventing posterior collapse with delta-vaes. *arXiv preprint arXiv:1901.03416*, 2019.

D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1530–1538. JMLR. org, 2015.

K. P. Scannell. The crúbadán project: Corpus building for under-resourced

languages. In *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop*, volume 4, pages 5–15, 2007.

S. Semeniuta, A. Severyn, and E. Barth. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 627–637, 2017.

I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, pages 3776–3783. AAAI Press, 2016.

I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, and Y. Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, pages 3295–3301. AAAI Press, 2017.

L. Shang, Z. Lu, and H. Li. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 1577–1586, 2015.

T. Shen, T. Lei, R. Barzilay, and T. Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6830–6841, 2017a.

X. Shen, H. Su, Y. Li, W. Li, S. Niu, Y. Zhao, A. Aizawa, and G. Long. A conditional variational framework for dialog generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–509, 2017b.

X. Shen, H. Su, S. Niu, and V. Demberg. Improving variational encoder-decoders in dialogue generation. In *Proceeding of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 5456–5463. AAAI, 2018.

Y. Shin, K. M. Yoo, and S.-G. Lee. Utterance generation with variational auto-encoder for slot filling in spoken language understanding. *IEEE Signal Processing Letters*, 26(3):505–509, 2019.

C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data (JBD)*, 6(1):60, 2019.

P. Simard, D. Steinkraus, and J. Platt. Best practices for convolutional neural networks applied to visual document analysis. *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR)*, pages 958–962, 2003.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, pages 3104–3112. MIT Press, 2014.

N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool. Deep convolutional neural networks and data augmentation for acoustic event recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2982–2986. International Speech and Communication Association, 2016.

M. A. Tanner and W. H. Wong. The calculation of posterior distributions by

data augmentation. *Journal of the American Statistical Association*, 82(398): 528–540, 1987.

B. Thomson and S. Young. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4): 562–588, 2010.

A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528. IEEE, 2011.

T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid. A bayesian data augmentation approach for learning deep models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2797–2806, 2017.

D. A. Van Dyk and X.-L. Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, 2017.

A. Vezhnevets and O. Barinova. Avoiding boosting overfitting by removing confusing samples. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 430–441. Springer, 2007.

O. Vinyals and Q. Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

Z. Wang and O. Lemon. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of The 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 423–432, 2013.

J. Wehrmann, R. Cerri, and R. Barros. Hierarchical multi-label classification networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 5225–5234, 2018.

J. W. Wei and K. Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gasic, L. M. R. Barahona, P.-H. Su, S. Ultes, and S. Young. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 438–449, 2017.

J. Wiebe, T. Wilson, and C. Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210, 2005.

J. Williams, A. Raux, D. Ramachandran, and A. Black. The dialog state tracking challenge. In *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 404–413, 2013.

J. D. Williams, K. Asadi, and G. Zweig. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 665–677, 2017.

C.-S. Wu, A. Madotto, E. Hosseini-Asl, C. Xiong, R. Socher, and P. Fung. Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743*, 2019.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation

system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Z. Xie and S. Ma. Dual-view variational autoencoders for semi-supervised text matching. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5306–5312. AAAI Press, 2019.

J. Xu and G. Durrett. Spherical latent spaces for stable variational autoencoders. *arXiv preprint arXiv:1808.10805*, 2018.

Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Y. Lu, and Z. Jin. Improved relation classification by deep recurrent neural networks with data augmentation. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers (COLING)*, pages 1461–1470, 2016.

Q. Yang, D. Shen, Y. Cheng, W. Wang, G. Wang, L. Carin, et al. An end-to-end generative architecture for paraphrase 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 3123–3133, 2019a.

Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 3881–3890. JMLR. org, 2017.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019b.

K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, and Y. Shi. Spoken language understanding using long short-term memory neural networks. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 189–194. IEEE, 2014.

K. M. Yoo, Y. Shin, and S.-g. Lee. Data augmentation for spoken language understanding via joint variational generation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 7402–7409, 2019.

H. J. Yoon, Y. J. Jeong, D.-Y. Kang, H. Kang, K. K. Yeo, J. E. Jeong, K. W. Park, G. E. Choi, and S.-W. Ha. Effect of data augmentation of f-18-florbetaben positron-emission tomography images by using deep learning convolutional neural network architecture for amyloid positive patients. *Journal of the Korean Physical Society*, 75(8):597–604, 2019.

L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 2852–2858, 2017.

B. Zhang, D. Xiong, J. Su, H. Duan, and M. Zhang. Variational neural machine translation. *arXiv preprint arXiv:1605.07869*, 2016a.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016b.

X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 649–657, 2015.

Y. Zhang, Z. Gan, and L. Carin. Generating text via adversarial training. In *Proceedings of the NIPS Workshop on Adversarial Training*, volume 21, 2016c.

H. Zhao, Z. Lu, and P. Poupart. Self-adaptive hierarchical sentence model. In *Proceedings of the 24th International Conference on Artificial Intelligence (ICOAI)*, pages 4069–4076. AAAI Press, 2015.

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

V. Zhong, C. Xiong, and R. Socher. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1458–1467, 2018.

Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning (ICML)*, pages 912–919. AAAI Press, 2003.

L. Zilka and F. Jurcicek. Incremental lstm-based dialog state tracker. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 757–762. IEEE, 2015.

B. Zoph, D. Yuret, J. May, and K. Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1575, 2016.

Appendices

A Posterior Collapse in VAEs

In this appendix, we summarize the current findings on VAE training behaviors and offer our own insight by decomposing the objective and exploring the training dynamics with real examples. Based on the intuition gained from our findings, we offer a simple methodology to greatly increase the chance of mitigating posterior collapse. We compare our training results with previous work on a hierarchical VAE model and show empirically that VAEs trained with our algorithm behave stably and produces highly controllable samples. Our algorithm has only two hyperparameters, greatly reducing the hyperparameter search space compared to the previous work.

The tendency of VAE models to lose stabiliity during training and fail to effectively learn the data distribution is a well-known issue. This phenomenon is known as *posterior collapse*, and it has been perceived to be much more common and challenging to tackle for the NLP tasks than for the vision domain, due to the autoregressive nature of language modeling and the inherit difficulty of fitting linguistic representations into Gaussian priors (Xu and Durrett, 2018).

A.1 A Brief Background

Posterior collapse arises when the approximate posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ or the model posterior $p_\theta(\mathbf{z} \mid \mathbf{x})$ collapses to the prior distribution $p(\mathbf{z})$, causing the KL-divergence term in the ELBO to become 0. The ELBO consists of the reconstruction term and the KL-divergence term that pressures the approximate posterior of the latent representation of $\mathbf{z}$ to mimic the prior. The KL-divergence term effectively acts as a regularizer for representation disentangle-

ment (Burgess et al., 2018). When the VAE is over-applied with the regularization, it falls in a local optimum where the KL-divergence loss term is minimized to 0 and no meaningful data is encoded into latent variable models.

There are two types of posterior collapse: (1) *inference collapse* occurs when the approximate posterior collapses with the prior, i.e. $q_\phi(\mathbf{z}) = p(\mathbf{z})$; while (2) *model collapse* is when the model posterior collapses with the prior, i.e. $p_\theta(\mathbf{z}) = p(\mathbf{z})$. Inference collapse has been discussed in previous literature (He et al., 2019; Cremer et al., 2018; Kim et al., 2018b), and the general consensus is that the approximate posterior usually "lags" behind the model posterior in terms of training progress(He et al., 2019). When the disparity between the two distributions worsens, the encoder will not be motivated to encode the data into meaningful representations, so the optimizer will gravitate towards a local optimum where the KL-divergence between the approximate posterior and the prior is minimal but the reconstruction loss is unoptimized.

This has been a huge problem especially in the NLP domain, as the usual teacher-forcing technique employed in training recurrent neural networks might encourage the decoder to overlook the information flow form the encoder and optimize autoregressive objectives instead. Hence, it is imperative to be equipped with appropriate knowledge and tools to closely monitor the training process of VAEs. The complexity of VAE training unmanageably increases when the VAE architecture is compositional and contains multiple levels of hierarchicity and recurrence, such as the ones that are proposed in this chapter and subsequent chapters.

A.2 Identifying Posterior Collapse

There have been considerable efforts on quantifying and diagnosing posterior collapse in VAEs, notably (He et al., 2019; Razavi et al., 2019; Cremer et al., 2018). The simplest method is to examine the KL-divergence term of the ELBO

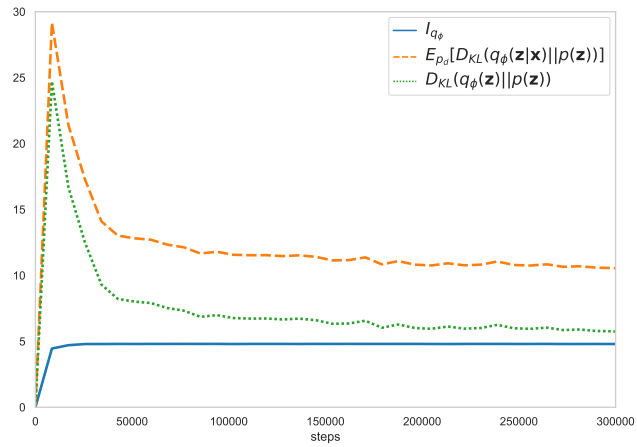


Figure 1: An example of ideal training behavior of VAE, which was obtained from the actual training behavior of LS-VAE, the proposed model for sentence classification data augmentation.

equation (Bowman et al., 2016), as shown in Figure 1. If the KL-divergence term is approximately zero, we can suspect with reasonable certainty that inference collapse has occurred. Joint visualization of mean approximate posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ and mean model posterior $p_\theta(\mathbf{z} \mid \mathbf{x})$ is another intuitive method for detecting posterior collapses(He et al., 2019). In the figure (not shown), the two distributions are plotted against each other for every data sample. The diagram can not only be used to detect inference collapse but can be used to detect model collapse and inference gap, in which the discrepancy between the two posteriors is quantified. However, when one of the posteriors approaches zero, its diminishing magnitude does not necessarily imply inference or model collapse, as we need to take the scale of the variance into account. As such, a recent work (Park et al., 2018) has proposed the utilization of *inverse relative variance*; by normalizing the variance of the means to the expected variance, one could get a clearer picture of the posterior distribution: $\mathrm{Var}\left[\mu\right]/\mathbb{E}\left[\sigma^2\right]$. The measure is

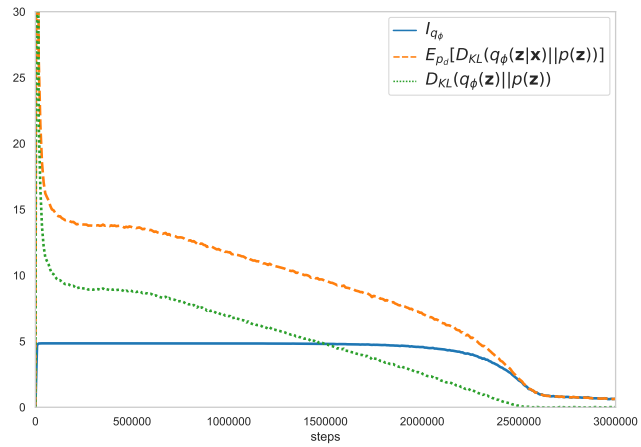


Figure 2: An example of posterior collapse

thought to be closely related to ANOVA (Analysis of Variance). Despite the diversity of metrics that have been suggested in the past to provide insights into the variational inference, they only paint one portion of the the entire picture of variational behaviors. One of the key to understanding training behaviors of VAEs, is to examine the following re-write of the ELBO KL-divergence term (Hoffman and Johnson, 2016):

$$\mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}[D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))] = D_{\mathrm{KL}}(q_\phi(\mathbf{z})\|p(\mathbf{z})) + I_{p_d,q_\phi}(\mathbf{x},\mathbf{z}) \tag{1}$$

where p_d is the empirical distribution of $\mathbf{x}$. The derivation of Equation 1 has been covered in (Hoffman and Johnson, 2016; Alemi et al., 2018). The KL-divergence term rewrite states that it is equivalent to the sum of the mutual information term $I_{p_d,q_\phi}(\mathbf{x},\mathbf{z})$ and the KL-divergence between the *aggregate approximate posterior* $q_\phi(\mathbf{z})$ and the prior of $\mathbf{z}$. As discussed in the previous subsection, the most common form of detector for posterior collapse is the measurement of the KL-divergence term; however, from the rewrite, it is apparent that a decreasing KL-divergence term $D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))$ might not always

indicate the occurrence of posterior collapse, as the decrease depends on the dynamics of the two decomposed terms. In order to gain complete picture of the training behavior of VAEs, we must observe at least two of the three terms that constitute the KL-divergence term rewrite (Equation 1).

When we minimize the KL-divergence term, what we *really* want to do is to keep the mutual information between $\mathbf{x}$ and $\mathbf{z}$ maximized while pressuring the other term to be minimized, i.e. keeping the aggregate of the posterior distributions $q_\phi(\mathbf{z})$ to be as similar to $p(\mathbf{z})$ as possible. However, optimization based on ELBO does not discriminate the constituting terms, causing unintended side effects such as coalescing minimization of the mutual information term. Although the reconstruction term in the ELBO should encourage the maximization of the mutual information term and that it should counter-balance the minimization pressure from the KL-divergence term, improper balance between the reconstruction term and the KL-divergence term could cause the optimizer to continue to even sacrifice the mutual information in order to meet the objectives.

An example of posterior collapse caused by mutual information collapse is illustrated in Figure 2. In the figure, the VAE is able to achieve a healthy level of mutual information for the majority of the training session. However, as the KL-divergence term between the aggregate posterior and the prior approaches zero, the mutual information level starts to fall below its previous optimal level, eventually collapsing to zero after a short period of destabilization. This observation has inspired us to propose the following idea for completely mitigating posterior collapse for any training behaviors of VAEs.

A.3 Estimating Mutual Information

The mutual information I_{p,q_ϕ} of the data $\mathbf{x}$ and the latent variable $\mathbf{z}$ under the empirical distribution of $\mathbf{x}$ and the approximate posterior can be rewritten in

different forms as follows:

$$I_{p,q_\phi}(\mathbf{x},\mathbf{z}) = D_{\mathrm{KL}}(q_\phi(\mathbf{x},\mathbf{z})\|q_\phi(\mathbf{z})\,p(\mathbf{x})) \tag{2}$$

$$= \mathbb{E}_p\left[D_{\mathrm{KL}}(q_\phi(\mathbf{z}\mid\mathbf{x})\|q_\phi(\mathbf{z}))\right] \tag{3}$$

$$= \mathbb{E}_p\left[D_{\mathrm{KL}}(q_\phi(\mathbf{z}\mid\mathbf{x})\|p(\mathbf{z}))\right] - D_{\mathrm{KL}}(q_\phi(\mathbf{z})\|p(\mathbf{z})) \tag{4}$$

The mutual information equation can be rewritten as the expected KL-divergence between the posterior and the aggregate posterior (Equation 3) or as a rearrangement of the ELBO KL-divergence term rewrite in Equation 1 (Equation 4). Both rewrites of the mutual information can be estimated using Monte Carlo estimation. The key difference between the two approaches is whether the approach utilizes analytic estimation of the ELBO KL-divergence term (the first term in Equation 4. In either cases, it are required to estiamte the *aggregate approximate posterior* $q_\phi(\mathbf{z})$, which can be estimated using sample data:

$$\begin{aligned} q_\phi(\mathbf{z}) &= \mathbb{E}_{\mathbf{x}\sim p_d}\left[q_\phi(\mathbf{z}\mid\mathbf{x})\right] \\ &\approx \frac{1}{N}\sum_i^N q_\phi(\mathbf{z}\mid\mathbf{x}_i) \end{aligned} \tag{5}$$

Using the aggregate posterior estimation, the KL-divergence between the aggregate posterior and the prior can be estimated as well using Monte Carlo sampling:

$$D_{\mathrm{KL}}(q_\phi(\mathbf{z})\|p(\mathbf{z})) = \mathbb{E}_{\mathbf{z}\ q_\phi(\mathbf{z})}\left[\log q_\phi(\mathbf{z}) - \log p(\mathbf{z})\right] \tag{6}$$

$$\approx \frac{1}{N}\sum_i^N \left(\log q_\phi(\mathbf{z}_i) - \log p(\mathbf{z}_i)\right)$$

$$\approx \frac{1}{N}\sum_i^N \left(\log \sum_j^M q_\phi(\mathbf{z}_i\mid\mathbf{x}_j) - \log M - \log p(\mathbf{z}_i)\right) \tag{7}$$

Hence, using Equation 7 and the analytic estimate of the ELBO KL-divergence term, we can estimate the mutual information between the data and the latent variables:

$$
\begin{aligned}
I_{p,q_\phi}(\mathbf{x}, \mathbf{z}) =& \mathbb{E}_p \left[D_{\mathrm{KL}}(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})) \right] - D_{\mathrm{KL}}(q_\phi(\mathbf{z}) \| p(\mathbf{z})) && (8) \\
\approx& \frac{1}{N} \sum_i^N D_{\mathrm{KL}}(q_\phi(\mathbf{z} \mid \mathbf{x}_i) \| p(\mathbf{z})) \\
& - \frac{1}{M} \sum_j^M \left(\log \sum_k^L q_\phi(\mathbf{z}_j \mid \mathbf{x}_k) - \log L - \log p(\mathbf{z}_j) \right) && (9)
\end{aligned}
$$

On the other hand, mutual information estimation can be carried out without utilizing the analytic estimate of ELBO KL-divergence term. We propose a second approach for mutual information estimation using Equation 3:

$$
\begin{aligned}
I_{p,q_\phi}(\mathbf{x}, \mathbf{z}) =& \mathbb{E}_p \left[D_{\mathrm{KL}}(q_\phi(\mathbf{z} \mid \mathbf{x}) \| q_\phi(\mathbf{z})) \right] && (10) \\
\approx& \frac{1}{NM} \sum_i^N \sum_j^M \left(\log q_\phi(\mathbf{z}_j \mid \mathbf{x}_i) - \log \sum_k^L q_\phi(\mathbf{z}_j \mid \mathbf{x}_k) + \log L \right)
\end{aligned}
$$

The key difference between the two approaches is that the second approach does not take the prior into account. We compare the two estimation methods on estimating the mutual information of a VAE trained on a simple dataset. The side-by-side comparison is shown in Figure 3. In the figure, `MI-1` and `MI-2` denote mutual information estimated using the first approach (Equation 8) and the second approach (Equation 10) respectively. The figure shows that the variance is lower in the mutual information estimated using the second approach. This is attributed to the fact that the first estimation method utilizes the KL-divergence with the *prior*, which in turn introduces more sample variances into the estimation. Hence, for the rest of the paper, we employ the second estimation method to analyze posterior collapse in VAEs.

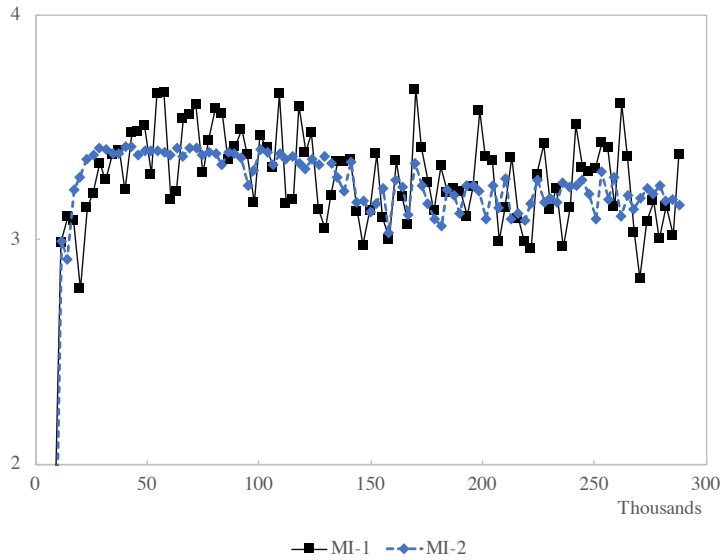


Figure 3: Mutual information $I_{p,q_\phi}(\mathbf{x}, \mathbf{z}_c)$ estimations

In practice, we estimate mutual information using data samples in a mini-batch. We further approximate the estimation by only sampling the latent variable $\mathbf{z}$ once for each data sample ($M = 1$):

$$I_{p,q_\phi}(\mathbf{x}, \mathbf{z}) \approx \frac{1}{N} \sum_{i}^{N} \left[\log q_\phi(\mathbf{z} \mid \mathbf{x}_i) - \log \sum_{j}^{N} q_\phi(\mathbf{z} \mid \mathbf{x}_j) + \log N \right]_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}_i)} \tag{11}$$

The complete algorithm is shown in Algorithm 2.

A.4 A Simple Technique for Mitigating Posterior Collapse

Based on the intuitions uncovered from the previous section that VAE training based on the ELBO is in risk of mutual information collapse, we propose a simple but effective algorithm for mitigating posterior collapse altogether. The algorithm, called **I-VAE**, essentially terminates the training session when it detects certain level of deterioration in the mutual information between $\mathbf{x}$ and $\mathbf{z}$, as shown in Algorithm 3. The main idea is that we want to slowly increase

input : data samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$

given : encoder parameters ϕ

output: mutual information estimate I

Compute posteriors $q_\phi(\mathbf{z} \mid \mathbf{x}_1), \ldots, q_\phi(\mathbf{z} \mid \mathbf{x}_N)$;

Sample latent variables from the posteriors

$\mathbf{z}_1, \ldots, \mathbf{z}_N \sim q_\phi(\mathbf{z} \mid \mathbf{x}_1), \ldots, q_\phi(\mathbf{z} \mid \mathbf{x}_N)$;

Let M be a matrix, where $m_{ij} = \log q_\phi(\mathbf{z}_i \mid \mathbf{x}_j)$; `// cross logprobs`

Initialize $s \leftarrow 0$;

foreach *i in* $1, \ldots, N$ **do**

  Compute $d \leftarrow \log \sum_j^N e^{m_{ij}}$;

  Update $s \leftarrow s + m_{ii} - d$;

end

return $s/N + \log N$

Algorithm 2: Algorithm for mutual information estimation

the KL-divergence regularization where the weight is determined by α and the current training iteration, encouraging the encoder and the docoder to freely explore optimal representations for reconstruction. As we increase the regularization weight, there is a certain point in the training dynamics where the mutual information between the data and the latent variables deteriorate, as described in Appendix A.2 and illustrated in Figure 2. For each training step, the algorithm estimates the mutual information using a Monte Carlo estimation method and keeps track of the maximum mutual information $I_{\max}$ for the entire training session. When the current mutual information level falls below a certain tolerance level β relative to the maximum MI $I_{\max}$, the algorithm will prepare for training termination. However, to account for turbulence in mutual information estimation, we employ the patience technique where we wait for τ steps before we confirm that the mutual information level has fallen. The algorithm returns the parameters θ and ϕ of the last model where the mutual

input : KL-divergence annealing rate α, tolerance β, patience τ

output: model parameters θ, ϕ

Initialize parameters θ, ϕ;

Initialize $I_{\text{max}} \leftarrow 0$, $I_{\text{final}} \leftarrow 0$;

Initialize patience counter $c \leftarrow 0$;

while *stopping criterion unfulfilled* **do**

Sample a mini-batch $\mathbf{x}_1, \ldots, \mathbf{x}_N \sim p_d(\mathbf{x})$; `// standard VAE training`

Update θ, ϕ using gradients from $\nabla_{\theta,\phi} \sum_i^N \mathcal{L}(\mathbf{x}_i; \alpha, \theta, \phi)$;

`/* check for stopping criterion */`

Compute MC estimation of $I \leftarrow \text{EstimateMI}(\mathbf{x}_1, \ldots, \mathbf{x}_N; \phi)$;

if $I > I_{max}$ **then**

Update $I_{\text{max}} \leftarrow I$;

Reset the patience counter $c \leftarrow 0$;

end

if $I < (1 - \beta) \cdot I_{max}$ **then**

Update $I_{\text{final}} \leftarrow I$ if the patience counter is zero ;

if $c > \tau$ **then**

terminate training ;

end

$c \leftarrow c + 1$;

end

else

Reset the patience counter $c \leftarrow 0$; `// rebound`

end

end

return θ, ϕ *of the model when* I_{final} *was last updated*

Algorithm 3: Algorithm for training I-VAE

information was within the tolerance bounds $(1-\beta)\cdot I_{\max} < I < I_{\max}$. The detailed method for estimating mutual information is discussed in Appendix A.3.

B Relation of the Mutual Information Trick to Other Methods

As a remedy to posterior collapse, (Bowman et al., 2016) proposed KL-divergence annealing, gradually increasing the weight of the KL-divergence term, allowing the model to develop autoencoding capability in the early stage of training. Orthogonal to the KL-divergence annealing technique is the modified training objective proposed for β-VAE (Higgins et al.), where the KL-divergence term is weighted a hyperparameters β. Our algorithm encompasses both intuitions of (Higgins et al.) and (Bowman et al., 2016), as the algorthm usually terminates before complete anneal of the KL-divergence term, producing the effect of applying the KL-divergence regularization pressure $\beta < 1$. However, we could increase α in I-VAE to the point where the effect is equivalent to $\beta > 1$, but, in practice, text modeling using VAEs suffers from posterior collapse without early termination for such large magnitudes of β.

A recent work by (He et al., 2019) proposed subdividing each iteration of VAE training into aggresive and non-aggresive training stages. In the aggressive training stage, only the encoder parameters θ is optimized, while keeping the decoder parameters constant. The criterion for switching from the aggressive to the non-aggresive stage is the plateauing of the mutual information level. The method shares the idea of using mutual information estimations to monitor the VAE training behavior, but the motivation and the resulting algorithm differs from ours. Previous work focused on bringing the training progress of encoder up to par with the decoder, and thus the MI monitoring is only used for

aggresive training stage. Some major drawbacks of Aggressive-VAE also limit its applicability in certain areas. Namely, the increased training complexity in time and code in implementing Aggressive-VAE and the sufficiency of employing teacher-forcing (or autoregression) dropout measures for alleviating inference collapse could be its limitations.

C Full Results on GDA for Dialogue State Tracking

The full results, including the standard deviation and the maximal value of each set of trials, of conducting generative data augmentation experiments on various dialogue state tracking datasets are presented in Table 1 and Table 2. In the table, we have included the full results on the reproduction of dialogue state trackers as well, allowing the comparison between the original models and the variants we have modified for our study (Section 5.4.1).

Generator	Tracker	WoZ2.0						DSTC2					
		Joint Goal			Turn Request			Joint Goal			Turn Request		
		μ	σ	max	μ	σ	max	μ	σ	max	μ	σ	max
-	GLAD[a]	*88.1*	*0.4*	-	*97.1*	*0.2*	-	*74.5*	*0.2*	-	*97.5*	*0.1*	-
-	GLAD[b]	*87.2*	*0.9*	*88.8*	*96.8*	*0.3*	*97.1*	*74.1*	*0.5*	*74.6*	*96.1*	*0.2*	*96.5*
-	GLAD$^+$	87.8	0.8	88.8	96.8	0.3	97.3	74.5	0.5	75.0	96.4	0.2	96.7
VHDA w/o goal	GLAD$^+$	86.5	0.6	87.0	**96.9**	0.2	97.2	74.7	0.5	75.1	**97.0**	0.2	97.2
VHDA	GLAD$^+$	**88.4**	0.3	88.7	96.6	0.2	97.0	**75.5**	0.5	75.9	96.8	0.5	97.1
-	GCE[a]	*88.5*	-	-	*97.4*	-	-	-	-	-	-	-	-
-	GCE[b]	*87.4*	*0.9*	*89.2*	*97.1*	*0.2*	*97.5*	*74.7*	*0.2*	*75.1*	*95.9*	*0.5*	*96.5*
-	GCE$^+$	88.3	0.7	89.3	97.0	0.2	97.4	74.8	0.6	75.5	96.3	0.2	96.6
VHDA w/o goal	GCE$^+$	86.4	1.2	87.1	96.3	0.2	96.6	75.5	0.3	75.7	**96.7**	0.7	97.2
VHDA	GCE$^+$	**89.3**	0.4	89.6	**97.1**	0.2	97.2	**76.0**	0.2	76.1	**96.7**	0.4	97.0
-	RNN	74.5	0.8	75.2	96.1	0.3	96.4	69.7	7.2	75.3	96.0	0.4	96.5
VHDA w/o goal	RNN	77.8	1.5	76.9	96.4	0.3	96.6	71.2	1.3	72.5	**97.2**	0.5	97.5
VHDA	RNN	**78.7**	2.1	80.2	**96.7**	0.1	96.8	**74.2**	0.9	74.8	97.0	0.2	97.1

[a] reported by the authors [b] reproduced on our environment

Table 1: Data augmentation results for dialogue state tracking for WoZ2.0 and DSTC2 datasets.

Generator	Tracker	MultiWoZ(Restaurant)						MultiWoZ(Hotel)					
		Joint Goal			Turn Inform			Joint Goal			Turn Inform		
		μ	σ	max	μ	σ	max	μ	σ	max	μ	σ	max
-	GLAD$^+$	58.9	2.5	62.9	76.3	1.4	77.7	33.4	2.4	38.5	58.9	1.5	60.5
VHDA	GLAD$^+$	**61.5**	2.4	64.6	**77.4**	2.0	79.0	**37.8**	2.2	40.3	**61.3**	1.0	62.3
-	GCE$^+$	60.5	3.4	65.0	76.7	1.2	78.7	36.5	2.4	41.3	61.0	1.2	63.0
VHDA	GCE$^+$	**63.3**	3.9	67.7	**77.2**	3.3	80.4	**38.3**	4.1	43.0	**63.1**	1.4	64.1
-	RNN	41.1	12.1	56.4	69.4	5.7	76.0	25.7	4.1	30.1	55.6	2.3	58.0
VHDA	RNN	**49.6**	3.1	59.1	**73.4**	1.8	76.3	**31.0**	5.0	35.7	**59.7**	3.1	62.3

[a] reported by the authors [b] reproduced on our environment

Table 2: Full data augmentation results for dialogue state tracking on MultiWoZ dataset.

초 록

최근 딥러닝 기반 생성 모델의 급격한 발전으로 이를 이용한 생성 기반 데이터 증강 기법(generative data augmentation, GDA)의 실현 가능성에 대한 기대가 커지고 있다. 생성 기반 데이터 증강 기법은 딥러닝 기반 잠재변수 모델에서 생성된 샘플을 원본 데이터셋에 추가하여 연관된 태스크의 성능을 향상시키는 기술을 의미한다. 따라서 생성 기반 데이터 증강 기법은 데이터 공간에서 이뤄지는 정규화 기술의 한 형태로 간주될 수 있다. 이러한 딥러닝 기반 생성 모델의 새로운 활용 가능성은 자연어처리 분야에서 더욱 중요하게 부각되는 이유는 (1) 범용 가능한 텍스트 데이터 증강 기술의 부재와 (2) 텍스트 데이터의 희소성을 극복할 수 있는 대안이 필요하기 때문이다. 문제의 복잡도와 특징을 골고루 채집하기 위해 본 논문에서는 텍스트 분류(text classification), 순차적 레이블링과 멀티태스킹 기술이 필요한 발화 이해(spoken language understanding, SLU), 계층적이며 재귀적인 데이터 구조에 대한 고려가 필요한 대화 상태 추적(dialogue state tracking, DST) 등 세 가지 문제에서 딥러닝 기반 생성 모델을 활용한 데이터 증강 기법의 타당성에 대해 다룬다. 본 연구에서는 조건부, 계층적 및 순차적 variational autoencoder (VAE)에 기반하여 각 자연어처리 문제에 특화된 텍스트 및 연관 부착 정보를 동시에 생성하는 특수 딥러닝 생성 모델들을 제시하고, 다양한 하류 모델과 데이터셋을 다루는 등 폭 넓은 실험을 통해 딥 생성 모델 기반 데이터 증강 기법의 효과를 통계적으로 입증하였다. 부수적 연구에서는 자기회귀적(autoregressive) VAE에서 빈번히 발생하는 posterior collapse 문제에 대해 탐구하고, 해당 문제를 완화할 수 있는 신규 방안도 제안한다. 해당 방법을 생성적 데이터 증강에 필요한 복잡한 VAE 모델에 적용하였을 때, 생성 모델의 생성 질이 향상되어 데이터 증강 효과에도 긍정적인 영향을 미칠 수 있음을 검증하였다. 본 논문을 통해 자연어처리 분야에서 기존 정규화 기법과 병행 적용 가능한 비지도 형태의 데이터 증강 기법의 표준화를 기대해 볼 수 있다.

주요어: 자연어처리, variational autoencoder, 데이터 증강, 자연어 생성, 잠재 변수 모델, 생성 모델, 텍스트 분류, 발화 이해, 대화 상태 추적
학번: 2014-21763