The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Data Augmentation for Spoken Language Understanding via Joint Variational Generation

Kang Min Yoo, Youhyun Shin, Sang-goo Lee
Seoul National University, Seoul 08826, Korea
{kangminyoo, shinu89, sglee}@europa.snu.ac.kr

Abstract

Data scarcity is one of the main obstacles of domain adaptation in spoken language understanding (SLU) due to the high cost of creating manually tagged SLU datasets. Recent works in neural text generative models, particularly latent variable models such as variational autoencoder (VAE), have shown promising results in regards to generating plausible and natural sentences. In this paper, we propose a novel generative architecture which leverages the generative power of latent variable models to jointly synthesize fully annotated utterances. Our experiments show that existing SLU models trained on the additional synthetic examples achieve performance gains. Our approach not only helps alleviate the data scarcity issue in the SLU task for many datasets but also indiscriminately improves language understanding performances for various SLU models, supported by extensive experiments and rigorous statistical testing.

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, Perozzi, and Skiena 2013), neural machine translation (NMT) (Zoph et al. 2016), and SLU (Kurata, Xiang, and Zhou 2016a). The issue is especially true for 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, Yousefhussien, and Kanan 2017; Kurata, Xiang, and Zhou 2016a; Hou et al. 2018).

In order to alleviate the data issues, data augmentation (DA) techniques that simply perform class-preserving transformation on data samples have been extensively used extensively (Simard, Steinkraus, and Platt 2003; Krizhevsky, Sutskever, and Hinton 2012; Fadaee, Bisazza, and Monz 2017). However, such DA methods require full supervision and generated datasets lack variety and robustness. To reduce reliance on handcrafted transformation functions, there has been growing interest in leveraging the generative power of latent variable models to facilitate DA. These line of works deserve a category of its own, to which we refer as *generative data augmentation* (GDA). Recent works have explored the idea for the SLU task (Kurata, Xiang, and Zhou 2016a; Hou et al. 2018).

In this paper, we formalize the notion of GDA by developing a general framework for the class of DA techniques in the SLU domain. Upon the framework, 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 three folds:

1. **The Generative DA Framework**: We develop a general framework of generative data augmentation specifically for the SLU task. During formulation, we posit the importance of prior approximation in generation sampling and propose a Monte Carlo-based method. Experiments show that the Monte Carlo-based estimation is superior compared to other approximation methods.
2. **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, Xiang, and Zhou 2016a).

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

Related Work

Deep Generative Models and Text Generation Variational autoencoders (VAE) (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014) are deep latent Gaussian models applied with stochastic variational inference, a method which makes the models scalable to large datasets. Since its inception, many variations of the VAE model have been explored in the language domain. Notably, variational recurrent auto-encoders (VRAE) were first proposed by (Fabius and van Amersfoort 2014). Generative adversarial networks (GAN) are another class of latent variable models with implicit latent distribution (Goodfellow et al. 2014). Advances have been made in applying the GAN model to text generation (Yu et al. 2017; Fedus, Goodfellow, and Dai 2018). Recently, much attention has been drawn to the tasks of controllable generation and style transfer, which have been successfully explored in (Hu et al. 2017; Shen et al. 2017) using variational models.

Data Augmentation and Regularization For data-hungry models, appropriate regularization is necessary to achieve high performance for many tasks. 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 noise 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) is a separate 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, Steinkraus, and Platt 2003; Krizhevsky, Sutskever, and Hinton 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, Yousefhussien, and Kanan 2017), general image classification (Ratner et al. 2017), and selected SLU tasks (Kurata, Xiang, and Zhou 2016a; Hou et al. 2018). However, relevant researches are hurdled by the architectural and experimental complexities. Nevertheless, our work is the first to explore idea of using variational generative models for DA.

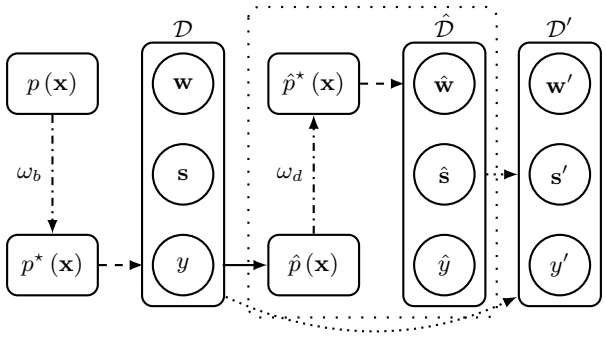


Figure 1: The general framework for generative language understanding data augmentation. Solid arrows (—) denote training, dashed arrows (- -) denote generation, dot-dashed arrows (- · -) denote distortion, and dotted arrows (·····) denote data duplication. $\mathcal{D}'$ is the final augmented dataset 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.

Spoken Language Understanding 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, Xu, and Yu 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 and Lane 2016).

Model Description

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

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:

$$\mathcal{L}_{LU}(\psi; \mathbf{w}, \mathbf{s}, y) = -\log p_\psi(\mathbf{s}, y|\mathbf{w}). \quad (1)$$

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. 2016).

Generative Data Augmentation A general framework of generative data augmentation (GDA) is depicted in Figure 1. 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 undersampling 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 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 can be seen as a corrective function ω_d that undos the effect of ω_b. In this paper, 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.

Algorithm 1: Monte Carlo posterior sampling.

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 $\mathbf{U}$

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 evidence lower bound (ELBO):

$$\begin{aligned}\mathcal{L}_{VAE}(\theta, \phi; \mathbf{w}) = &\mathrm{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{w}) \parallel p(\mathbf{z})) \\ &- \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{w})}[\log p_\theta(\mathbf{w}|\mathbf{z})]\end{aligned} \quad (2)$$

In Equation 2, 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).

6

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:

$$\hat{\mathbf{w}} \sim p_{\theta_\mathcal{D}, \phi_\mathcal{D}}(\mathbf{w}) = \int p_{\theta_\mathcal{D}}(\mathbf{w}|\mathbf{z})\, p_{\theta_\mathcal{D}, \phi_\mathcal{D}}(\mathbf{z})\, d\mathbf{z} \quad (3)$$

As evident in Equation 3, the marginal likelihood estimation requires us to infer the marginal probability of the latent

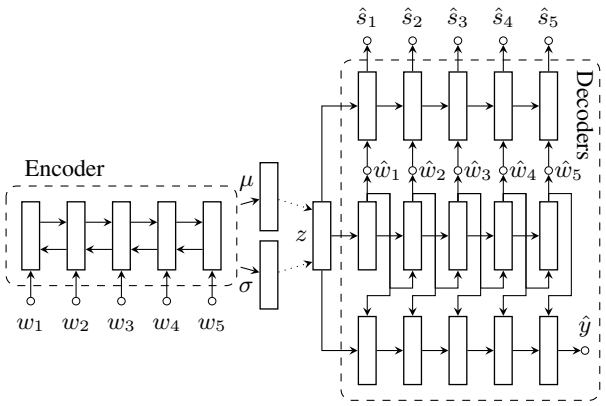


Figure 2: Joint language understanding variational autoencoder (JLUVA). The VAE model consists of a BiLSTM-Max encoder and three uni-directional decoders. Note that the fully connected layers and embedding layers are omitted for clarity.

variable $p_{\theta_\mathcal{D},\phi_\mathcal{D}}(\mathbf{z})$, which can be estimated by marginalizing the joint probability from the recognition network.

$$p_{\theta_\mathcal{D},\phi_\mathcal{D}}(\mathbf{z}) = \mathbb{E}_{\mathbf{w}\sim p(\mathbf{w})}\left[q_{\phi_\mathcal{D}}(\mathbf{z}|\mathbf{w})\right] \tag{4}$$

However, Equation 4 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 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 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 1: Dataset statistics. Training sets of ATIS (Small) and ATIS (Medium) have been chunked from the training set of ATIS (Full).

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 2). 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.

$$\mathcal{L}_{LU}(\phi, \psi; \mathbf{w}, \mathbf{s}, y) = -\mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log p_\psi(\mathbf{s}, y|\hat{\mathbf{w}}, \mathbf{z})\right] \tag{5}$$

The joint training objective of the entire model is specified in terms of the training objective of the VAE component (Equation 2) and the negative log-likelihood of the discriminatory component (Equation 5):

$$\begin{aligned}\mathcal{L}(\theta, \phi, \psi; \mathbf{w}, \mathbf{s}, y) = &\mathrm{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{w}) \parallel p_\theta(\mathbf{z}|\mathbf{w})) \\ &- \mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log p_\theta(\mathbf{w}|\mathbf{z})\right] \\ &- \mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log p_\psi(\mathbf{s}, y|\hat{\mathbf{w}}, \mathbf{z})\right]\end{aligned} \tag{6}$$

We obtain the optimal parameters θ^*, ϕ^*, ψ^* by minimizing Equation 6 (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.

Experiments

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

Model + Sampling Approach	Slot Filling (F1)			Intent (F1)			Semantic (Acc.)		
	Small	Med.	Full	Small	Med.	Full	Small	Med.	Full
Baseline (No Augmentation)	72.57‡	88.28‡	95.34	82.65	90.59†	97.21	35.09‡	65.18‡	85.95
Encoder-Decoder + Additive*	74.79†	89.13‡	95.20	-	-	-	-	-	-
JLUVA + Additive (Ours)	74.14‡	89.13‡	95.40	83.46	**90.97**	97.04	38.58	66.75	85.81
JLUVA + Standard Gaussian (Ours)	70.72‡	86.90‡	94.91‡	78.67‡	86.90‡	96.90	32.46‡	61.12‡	84.62‡
JLUVA + Posterior (Ours)	**74.92**	**89.27**	**95.55**	**83.65**	90.95	**97.24**	**39.43**	**67.05**	**86.33**

* (Kurata, Xiang, and Zhou 2016a) † $p < 0.1$ ‡ $p < 0.01$

Table 2: Data scarcity results for the ATIS dataset. 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.

Datasets

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

- **ATIS**: Airline Travel Information System (ATIS) (Hemphill, Godfrey, and Doddington 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 1. In order to simulate a data scarce environment (similar to the experimental design proposed in (Chen et al. 2016)), 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. 2016).

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 3), 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, Socher, and Manning 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

Dataset	Slot-Gated (Full)			Slot-Gated (Intent)		
	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 3: Mean data augmentation results on various SLU tasks tested using the slot-gated (Goo et al. 2018) SLU models. Datasets are augmented (prefixed by +) using our proposed generative model. The results have been aggregated and are tested for statistical significance.

Ba 2014) optimizer with 0.001 initial learning rate. The code is available on github (`kaniblu/ludus-jluva`).

SLU Models For the baseline SLU model, we implemented a simple BiLSTM model. 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 $p(s_t|\mathbf{w}_{\leq t})$. 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 $p(y|\mathbf{w})$. 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). 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.

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, Xiang, and Zhou 2016a). The two decoders of the model learn to decode utterances and slot labels from an encoded representation of the utterance.

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 and Lane 2016) [d] run on our environment

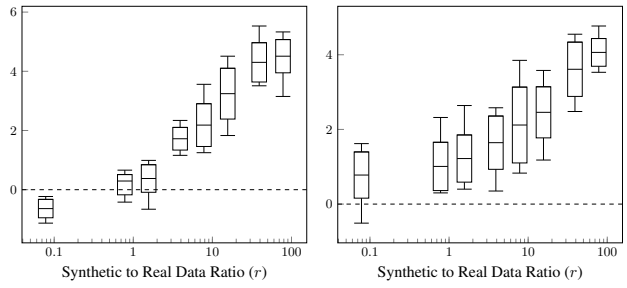
Table 4: Comparisons of the best slot filling and intent detection results for the ATIS dataset.

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 2.

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 3). 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 2, 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.



(a) Slot Filling (b) Intent Classification

Figure 3: The impact of synthetic data to real data ratio on the relative improvements in SLU performance. The vertical axis shows the relative performance gains, compared to the non-augmented baseline (dashed horizontal lines). For each box plot, the height of the box depicts the variance and outer whiskers mark the minimum and the maximum.

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). 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. 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.

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, Xiang, and Zhou 2016a).

The results in Table 2 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 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 $r \in \{0.08, 0.78, 1.56, 3.90, 7.81, 15.6, 39.06, 78.13\}$. From the box plots of our results (Figure 3), 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, 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.

Conclusion

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.

References

Al-Rfou, R.; Perozzi, B.; and Skiena, S. 2013. Polyglot: Distributed word representations for multilingual nlp. In *CoNLL*, 183–192.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A.; Jozefowicz, R.; and Bengio, S. 2016. Generating sentences from a continuous space. In *CoNLL*, 10–21.

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

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*, 670–680.

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

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

Fadaee, M.; Bisazza, A.; and Monz, C. 2017. Data augmentation for low-resource neural machine translation. In *ACL*, volume 2, 567–573.

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

Goo, C.-W.; Gao, G.; Hsu, Y.-K.; Huo, C.-L.; Chen, T.-C.; Hsu, K.-W.; and Chen, Y.-N. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *NAACL*, volume 2, 753–757.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

Guo, D.; Tur, G.; Yih, W.-t.; and Zweig, G. 2014. Joint semantic utterance classification and slot filling with recursive neural networks. In *SLT Workshop*, 554–559.

Hemphill, C. T.; Godfrey, J. J.; and Doddington, G. R. 1990. The atis spoken language systems pilot corpus. In *The Workshop on Speech and Natural Language*, 96–101.

Hou, Y.; Liu, Y.; Che, W.; and Liu, T. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *ICCL*, 1234–1245.

Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017. Toward controlled generation of text. In *ICML*, 1587–1596.

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

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 448–456.

Kafle, K.; Yousefhussien, M.; and Kanan, C. 2017. Data augmentation for visual question answering. In *INLG*, 198–202.

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

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

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

Kurata, G.; Xiang, B.; Zhou, B.; and Yu, M. 2016b. Leveraging sentence-level information with encoder lstm for semantic slot filling. In *EMNLP*, 2077–2083.

Kurata, G.; Xiang, B.; and Zhou, B. 2016a. Labeled data generation with encoder-decoder lstm for semantic slot filling. In *INTERSPEECH*, 725–729.

Lai, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, 2267–2273.

Li, J.; Monroe, W.; Shi, T.; Jean, S.; Ritter, A.; and Jurafsky, D. 2017. Adversarial learning for neural dialogue generation. In *EMNLP*, 2157–2169.

Liu, B., and Lane, I. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *INTERSPEECH*, 685–689.

Mesnil, G.; Dauphin, Y.; Yao, K.; Bengio, Y.; Deng, L.; Hakkani-Tur, D.; He, X.; Heck, L.; Tur, G.; Yu, D.; and Zweig, G. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM TASLP* 23:530–539.

Pan, S. J., and Yang, Q. 2010. A Survey on Transfer Learning. *IEEE TKDE* 22:1345–1359.

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

Ratner, A. J.; Ehrenberg, H.; Hussain, Z.; Dunnmon, J.; and Ré, C. 2017. Learning to compose domain-specific transformations for data augmentation. In *NIPS*, 3236–3246.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, II–1278.

Serban, I. V.; Sordoni, A.; Lowe, R.; Charlin, L.; Pineau, J.; Courville, A. C.; and Bengio, Y. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, 3295–3301.

Shen, T.; Lei, T.; Barzilay, R.; and Jaakkola, T. 2017. Style transfer from non-parallel text by cross-alignment. In *NIPS*, 6830–6841.

Simard, P.; Steinkraus, D.; and Platt, J. 2003. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, 958.

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

Torralba, A., and Efros, A. A. 2011. Unbiased look at dataset bias. In *CVPR*, 1521–1528.

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

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2852–2858.

Zoph, B.; Yuret, D.; May, J.; and Knight, K. 2016. Transfer learning for low-resource neural machine translation. In *EMNLP*, 1568–1575.