Unsupervised Discrete Sentence Representation Learning for Interpretable Neural Dialog Generation

Tiancheng Zhao, Kyusong Lee and Maxine Eskenazi
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
{tianchez, kyusongl, max+}@cs.cmu.edu

Abstract

The encoder-decoder dialog model is one of the most prominent methods used to build dialog systems in complex domains. Yet it is limited because it cannot output interpretable actions as in traditional systems, which hinders humans from understanding its generation process. We present an unsupervised discrete sentence representation learning method that can integrate with any existing encoder-decoder dialog models for interpretable response generation. Building upon variational autoencoders (VAEs), we present two novel models, DI-VAE and DI-VST that improve VAEs and can discover interpretable semantics via either auto encoding or context predicting. Our methods have been validated on real-world dialog datasets to discover semantic representations and enhance encoder-decoder models with interpretable generation.[1]

1 Introduction

Classic dialog systems rely on developing a meaning representation to represent the utterances from both the machine and human users (Larsson and Traum, 2000; Bohus et al., 2007). The dialog manager of a conventional dialog system outputs the system's next action in a semantic frame that usually contains hand-crafted dialog acts and slot values (Williams and Young, 2007). Then a natural language generation module is used to generate the system's output in natural language based on the given semantic frame. This approach suffers from generalization to more complex domains because it soon become intractable to man-

ually design a frame representation that covers all of the fine-grained system actions. The recently developed neural dialog system is one of the most prominent frameworks for developing dialog agents in complex domains. The basic model is based on encoder-decoder networks (Cho et al., 2014) and can learn to generate system responses without the need for hand-crafted meaning representations and other annotations.
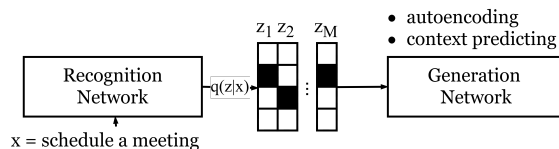

Figure 1: Our proposed models learn a set of discrete variables to represent sentences by either autoencoding or context prediction.

Although generative dialog models have advanced rapidly (Serban et al., 2016; Li et al., 2016; Zhao et al., 2017), they cannot provide interpretable system actions as in the conventional dialog systems. This inability limits the effectiveness of generative dialog models in several ways. First, having interpretable system actions enables human to understand the behavior of a dialog system and better interpret the system intentions. Also, modeling the high-level decision-making policy in dialogs enables useful generalization and data-efficient domain adaptation (Gašić et al., 2010). Therefore, the motivation of this paper is to develop an unsupervised neural recognition model that can discover interpretable meaning representations of utterances (denoted as *latent actions*) as a set of discrete latent variables from a large unlabelled corpus as shown in Figure 1. The discovered meaning representations will then be integrated with encoder decoder networks to achieve interpretable dialog generation while preserving

---

[1]Data and code are available at https://github.com/snakeztc/NeuralDialog-LAED.

all the merit of neural dialog systems.

We focus on learning discrete latent representations instead of dense continuous ones because discrete variables are easier to interpret (van den Oord et al., 2017) and can naturally correspond to categories in natural languages, e.g. topics, dialog acts and etc. Despite the difficulty of learning discrete latent variables in neural networks, the recently proposed Gumbel-Softmax offers a reliable way to back-propagate through discrete variables (Maddison et al., 2016; Jang et al., 2016). However, we found a simple combination of sentence variational autoencoders (VAEs) (Bowman et al., 2015) and Gumbel-Softmax fails to learn meaningful discrete representations. We then highlight the anti-information limitation of the evidence lowerbound objective (ELBO) in VAEs and improve it by proposing Discrete Information VAE (DI-VAE) that maximizes the mutual information between data and latent actions. We further enrich the learning signals beyond auto encoding by extending Skip Thought (Kiros et al., 2015) to Discrete Information Variational Skip Thought (DI-VST) that learns sentence-level distributional semantics. Finally, an integration mechanism is presented that combines the learned latent actions with encoder decoder models.

The proposed systems are tested on several real-world dialog datasets. Experiments show that the proposed methods significantly outperform the standard VAEs and can discover meaningful latent actions from these datasets. Also, experiments confirm the effectiveness of the proposed integration mechanism and show that the learned latent actions can control the sentence-level attributes of the generated responses and provide human-interpretable meaning representations.

2   Related Work

Our work is closely related to research in latent variable dialog models. The majority of models are based on Conditional Variational Autoencoders (CVAEs) (Serban et al., 2016; Cao and Clark, 2017) with continuous latent variables to better model the response distribution and encourage diverse responses. Zhao et al., (2017) further introduced dialog acts to guide the learning of the CVAEs. Discrete latent variables have also been used for task-oriented dialog systems (Wen et al., 2017), where the latent space is used to represent intention. The second line of related work

is enriching the dialog context encoder with more fine-grained information than the dialog history. Li et al., (2016) captured speakers' characteristics by encoding background information and speaking style into the distributed embeddings. Xing et al., (2016) maintain topic encoding based on Latent Dirichlet Allocation (LDA) (Blei et al., 2003) of the conversation to encourage the model to output more topic coherent responses.

The proposed method also relates to sentence representation learning using neural networks. Most work learns continuous distributed representations of sentences from various learning signals (Hill et al., 2016), e.g. the Skip Thought learns representations by predicting the previous and next sentences (Kiros et al., 2015). Another area of work focused on learning regularized continuous sentence representation, which enables sentence generation by sampling the latent space (Bowman et al., 2015; Kim et al., 2017). There is less work on discrete sentence representations due to the difficulty of passing gradients through discrete outputs. The recently developed Gumbel Softmax (Jang et al., 2016; Maddison et al., 2016) and vector quantization (van den Oord et al., 2017) enable us to train discrete variables. Notably, discrete variable models have been proposed to discover document topics (Miao et al., 2016) and semi-supervised sequence transaction (Zhou and Neubig, 2017)

Our work differs from these as follows: (1) we focus on learning interpretable variables; in prior research the semantics of latent variables are mostly ignored in the dialog generation setting. (2) we improve the learning objective for discrete VAEs and overcome the well-known posterior collapsing issue (Bowman et al., 2015; Chen et al., 2016). (3) we focus on unsupervised learning of salient features in dialog responses instead of hand-crafted features.

3   Proposed Methods

Our formulation contains three random variables: the dialog context $\mathbf{c}$, the response $\mathbf{x}$ and the latent action $\mathbf{z}$. The context often contains the discourse history in the format of a list of utterances. The response is an utterance that contains a list of word tokens. The latent action is a set of discrete variables that define high-level attributes of $\mathbf{x}$. Before introducing the proposed framework, we first identify two key properties that are essential in or-

der for **z** to be *interpretable*:

1. **z** should capture salient sentence-level features about the response **x**.

2. The meaning of latent symbols **z** should be independent of the context **c**.

The first property is self-evident. The second can be explained: assume **z** contains a single discrete variable with K classes. Since the context **c** can be any dialog history, if the meaning of each class changes given a different context, then it is difficult to extract an intuitive interpretation by only looking at all responses with class $k \in [1, K]$. Therefore, the second property looks for latent actions that have *context-independent semantics* so that each assignment of **z** conveys the same meaning in all dialog contexts.

With the above definition of interpretable latent actions, we first introduce a recognition network $\mathcal{R} : q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})$ and a generation network $\mathcal{G}$. The role of $\mathcal{R}$ is to map an sentence to the latent variable **z** and the generator $\mathcal{G}$ defines the learning signals that will be used to train **z**'s representation. Notably, our recognition network $\mathcal{R}$ does not depend on the context **c** as has been the case in prior work (Serban et al., 2016). The motivation of this design is to encourage **z** to capture context-independent semantics, which are further elaborated in Section 3.4. With the **z** learned by $\mathcal{R}$ and $\mathcal{G}$, we then introduce an encoder decoder network $\mathcal{F} : p_{\mathcal{F}}(\mathbf{x}|\mathbf{z}, \mathbf{c})$ and and a policy network $\pi : p_{\pi}(\mathbf{z}|\mathbf{c})$. At test time, given a context **c**, the policy network and encoder decoder will work together to generate the next response via $\tilde{\mathbf{x}} = p_{\mathcal{F}}(\mathbf{x}|\mathbf{z} \sim p_{\pi}(\mathbf{z}|\mathbf{c}), \mathbf{c})$. In short, $\mathcal{R}$, $\mathcal{G}$, $\mathcal{F}$ and π are the four components that comprise our proposed framework. The next section will first focus on developing $\mathcal{R}$ and $\mathcal{G}$ for learning interpretable **z** and then will move on to integrating $\mathcal{R}$ with $\mathcal{F}$ and π in Section 3.3.

3.1 Learning Sentence Representations from Auto-Encoding

Our baseline model is a sentence VAE with discrete latent space. We use an RNN as the recognition network to encode the response **x**. Its last hidden state $h_{|\mathbf{x}|}^{\mathcal{R}}$ is used to represent **x**. We define **z** to be a set of K-way categorical variables $\mathbf{z} = \{\mathbf{z}_1...\mathbf{z}_m...\mathbf{z}_M\}$, where M is the number of variables. For each $\mathbf{z}_m$, its posterior distribution is defined as $q_{\mathcal{R}}(\mathbf{z}_m|\mathbf{x}) = \text{Softmax}(W_q h_{|\mathbf{x}|}^{\mathcal{R}} + b_q)$.

During training, we use the Gumbel-Softmax trick to sample from this distribution and obtain low-variance gradients. To map the latent samples to the initial state of the decoder RNN, we define $\{e_1...e_m...e_M\}$ where $e_m \in \mathbb{R}^{K \times D}$ and D is the generator cell size. Thus the initial state of the generator is: $h_0^{\mathcal{G}} = \sum_{m=1}^{M} e_m(\mathbf{z}_m)$. Finally, the generator RNN is used to reconstruct the response given $h_0^{\mathcal{G}}$. VAEs is trained to maxmimize the evidence lowerbound objective (ELBO) (Kingma and Welling, 2013). For simplicity, later discussion drops the subscript m in $\mathbf{z}_m$ and assumes a single latent **z**. Since each $\mathbf{z}_m$ is independent, we can easily extend the results below to multiple variables.

3.1.1 Anti-Information Limitation of ELBO

It is well-known that sentence VAEs are hard to train because of the posterior collapse issue. Many empirical solutions have been proposed: weakening the decoder, adding auxiliary loss etc. (Bowman et al., 2015; Chen et al., 2016; Zhao et al., 2017). We argue that the posterior collapse issue lies in ELBO and we offer a novel decomposition to understand its behavior. First, instead of writing ELBO for a single data point, we write it as an expectation over a dataset:

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})}[\log p_{\mathcal{G}}(\mathbf{x}|\mathbf{z})] \\ - \text{KL}(q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))] \tag{1}$$

We can expand the KL term as Eq. 2 (derivations in Appendix A.1) and rewrite ELBO as:

$$\mathbb{E}_{\mathbf{x}}[\text{KL}(q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))] = \tag{2} \\ I(Z, X) + \text{KL}(q(\mathbf{z})\|p(\mathbf{z}))$$

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})p(\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] \\ - I(Z, X) - \text{KL}(q(\mathbf{z})\|p(\mathbf{z})) \tag{3}$$

where $q(\mathbf{z}) = \mathbb{E}_{\mathbf{x}}[q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})]$ and $I(Z, X)$ is the mutual information between Z and X. This expansion shows that the KL term in ELBO is trying to reduce the mutual information between latent variables and the input data, which explains why VAEs often ignore the latent variable, especially when equipped with powerful decoders.

3.1.2 VAE with Information Maximization and Batch Prior Regularization

A natural solution to correct the anti-information issue in Eq. 3 is to maximize both the data likeli-

1100

hood lowerbound and the mutual information between $\mathbf{z}$ and the input data:

$$\mathcal{L}_{\text{VAE}} + I(Z, X) =$$
$$\mathbb{E}_{q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})p(\mathbf{x})}[\log p_{\mathcal{G}}(\mathbf{x}|\mathbf{z})] - \text{KL}(q(\mathbf{z})\|p(\mathbf{z})) \quad (4)$$

Therefore, jointly optimizing ELBO and mutual information simply cancels out the information-discouraging term. Also, we can still sample from the prior distribution for generation because of $\text{KL}(q(\mathbf{z})\|p(\mathbf{z}))$. Eq. 4 is similar to the objectives used in adversarial autoencoders (Makhzani et al., 2015; Kim et al., 2017). Our derivation provides a theoretical justification to their superior performance. Notably, Eq. 4 arrives at the same loss function proposed in infoVAE (Zhao S et al., 2017). However, our derivation is different, offering a new way to understand ELBO behavior.

The remaining challenge is how to minimize $\text{KL}(q(\mathbf{z})\|p(\mathbf{z}))$, since $q(\mathbf{z})$ is an expectation over $q(\mathbf{z}|\mathbf{x})$. When $\mathbf{z}$ is continuous, prior work has used adversarial training (Makhzani et al., 2015; Kim et al., 2017) or Maximum Mean Discrepancy (MMD) (Zhao S et al., 2017) to regularize $q(\mathbf{z})$. It turns out that minimizing $\text{KL}(q(\mathbf{z})\|p(\mathbf{z}))$ for discrete $\mathbf{z}$ is much simpler than its continuous counterparts. Let $\mathbf{x}_n$ be a sample from a batch of N data points. Then we have:

$$q(\mathbf{z}) \approx \frac{1}{N} \sum_{n=1}^{N} q(\mathbf{z}|\mathbf{x}_n) = q'(\mathbf{z}) \quad (5)$$

where $q'(\mathbf{z})$ is a mixture of softmax from the posteriors $q(\mathbf{z}|\mathbf{x}_n)$ of each $\mathbf{x}_n$. We can approximate $\text{KL}(q(\mathbf{z})\|p(\mathbf{z}))$ by:

$$\text{KL}(q'(\mathbf{z})\|p(\mathbf{z})) = \sum_{k=1}^{K} q'(\mathbf{z}=k) \log \frac{q'(\mathbf{z}=k)}{p(\mathbf{z}=k)} \quad (6)$$

We refer to Eq. 6 as Batch Prior Regularization (BPR). When N approaches infinity, $q'(\mathbf{z})$ approaches the true marginal distribution of $q(\mathbf{z})$. In practice, we only need to use the data from each mini-batch assuming that the mini batches are randomized. Last, BPR is fundamentally different from multiplying a coefficient < 1 to anneal the KL term in VAE (Bowman et al., 2015). This is because BPR is a non-linear operation *log_sum_exp*. For later discussion, we denote our discrete infoVAE with BPR as DI-VAE.

3.2 Learning Sentence Representations from the Context

DI-VAE infers sentence representations by reconstruction of the input sentence. Past research in distributional semantics has suggested the meaning of language can be inferred from the adjacent context (Harris, 1954; Hill et al., 2016). The distributional hypothesis is especially applicable to dialog since the utterance meaning is highly contextual. For example, the dialog act is a well-known utterance feature and depends on dialog state (Austin, 1975; Stolcke et al., 2000). Thus, we introduce a second type of latent action based on sentence-level distributional semantics.

Skip thought (ST) is a powerful sentence representation that captures contextual information (Kiros et al., 2015). ST uses an RNN to encode a sentence, and then uses the resulting sentence representation to predict the previous and next sentences. Inspired by ST's robust performance across multiple tasks (Hill et al., 2016), we adapt our DI-VAE to Discrete Information Variational Skip Thought (DI-VST) to learn discrete latent actions that model distributional semantics of sentences. We use the same recognition network from DI-VAE to output $\mathbf{z}$'s posterior distribution $q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})$. Given the samples from $q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})$, two RNN generators are used to predict the previous sentence $\mathbf{x}_p$ and the next sentences $\mathbf{x}_n$. Finally, the learning objective is to maximize:

$$\mathcal{L}_{\text{DI-VST}} = \mathbb{E}_{q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})p(\mathbf{x}))}[\log(p_{\mathcal{G}}^{n}(\mathbf{x}_n|\mathbf{z})p_{\mathcal{G}}^{p}(\mathbf{x}_p|\mathbf{z}))]$$
$$- \text{KL}(q(\mathbf{z})\|p(\mathbf{z})) \quad (7)$$

3.3 Integration with Encoder Decoders

We now describe how to integrate a given $q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})$ with an encoder decoder and a policy network. Let the dialog context $\mathbf{c}$ be a sequence of utterances. Then a dialog context encoder network can encode the dialog context into a distributed representation $h^e = \mathcal{F}^e(\mathbf{c})$. The decoder $\mathcal{F}^d$ can generate the responses $\tilde{\mathbf{x}} = \mathcal{F}^d(h^e, \mathbf{z})$ using samples from $q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})$. Meanwhile, we train π to predict the aggregated posterior $\mathbb{E}_{p(\mathbf{x}|\mathbf{c})}[q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})]$ from $\mathbf{c}$ via maximum likelihood training. This model is referred as Latent Action Encoder Decoder (LAED) with the following objective.

$$\mathcal{L}_{\text{LAED}}(\theta_{\mathcal{F}}, \theta_{\pi}) =$$
$$\mathbb{E}_{q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})p(\mathbf{x},\mathbf{c})}[\log p_{\pi}(\mathbf{z}|\mathbf{c}) + \log p_{\mathcal{F}}(\mathbf{x}|\mathbf{z},\mathbf{c})] \quad (8)$$

Also simply augmenting the inputs of the decoders with latent action does not guarantee that the generated response exhibits the attributes of the give action. Thus we use the controllable text generation framework (Hu et al., 2017) by introducing $\mathcal{L}_{\text{Attr}}$, which reuses the same recognition network $q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})$ as a fixed discriminator to penalize the decoder if its generated responses do not reflect the attributes in $\mathbf{z}$.

$$\mathcal{L}_{\text{Attr}}(\theta_{\mathcal{F}}) = \mathbb{E}_{q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})p(\mathbf{c},\mathbf{x})}[\log q_{\mathcal{R}}(\mathbf{z}|\mathcal{F}(\mathbf{c},\mathbf{z}))] \quad (9)$$

Since it is not possible to propagate gradients through the discrete outputs at $\mathcal{F}^d$ at each word step, we use a deterministic continuous relaxation (Hu et al., 2017) by replacing output of $\mathcal{F}^d$ with the probability of each word. Let $\mathbf{o}_t$ be the normalized probability at step $t \in [1, |\mathbf{x}|]$, the inputs to $q_{\mathcal{R}}$ at time t are then the sum of word embeddings weighted by $\mathbf{o}_t$, i.e. $h_t^{\mathcal{R}} = \text{RNN}(h_{t-1}^{\mathcal{R}}, \mathbf{E}\mathbf{o}_t)$ and $\mathbf{E}$ is the word embedding matrix. Finally this loss is combined with $\mathcal{L}_{\text{LAED}}$ and a hyperparameter λ to have Attribute Forcing LAED.

$$\mathcal{L}_{\text{attrLAED}} = \mathcal{L}_{\text{LAED}} + \lambda\mathcal{L}_{\text{Attr}} \quad (10)$$

3.4 Relationship with Conditional VAEs

It is not hard to see $\mathcal{L}_{\text{LAED}}$ is closely related to the objective of CVAEs for dialog generation (Serban et al., 2016; Zhao et al., 2017), which is:

$$\mathcal{L}_{\text{CVAE}} = \mathbb{E}_q[\log p(\mathbf{x}|\mathbf{z},\mathbf{c})] - \text{KL}(q(\mathbf{z}|\mathbf{x},\mathbf{c})\|p(\mathbf{z}|\mathbf{c})) \quad (11)$$

Despite their similarities, we highlight the key differences that prohibit CVAE from achieving interpretable dialog generation. First $\mathcal{L}_{\text{CVAE}}$ encourages $I(\mathbf{x}, \mathbf{z}|\mathbf{c})$ (Agakov, 2005), which learns $\mathbf{z}$ that capture context-dependent semantics. More intuitively, $\mathbf{z}$ in CVAE is trained to generate $\mathbf{x}$ via $p(\mathbf{x}|\mathbf{z}, \mathbf{c})$ so the meaning of learned $\mathbf{z}$ can only be interpreted along with its context $\mathbf{c}$. Therefore this violates our goal of learning context-independent semantics. Our methods learn $q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})$ that only depends on $\mathbf{x}$ and trains $q_{\mathcal{R}}$ separately to ensure the semantics of $\mathbf{z}$ are interpretable standalone.

4 Experiments and Results

The proposed methods are evaluated on four datasets. The first corpus is Penn Treebank (PTB) (Marcus et al., 1993) used to evaluate sentence VAEs (Bowman et al., 2015). We used

the version pre-processed by Mikolov (Mikolov et al., 2010). The second dataset is the Stanford Multi-Domain Dialog (SMD) dataset that contains 3,031 human-Woz, task-oriented dialogs collected from 3 different domains (navigation, weather and scheduling) (Eric and Manning, 2017). The other two datasets are chat-oriented data: Daily Dialog (DD) and Switchboard (SW) (Godfrey and Holliman, 1997), which are used to test whether our methods can generalize beyond task-oriented dialogs but also to to open-domain chatting. DD contains 13,118 multi-turn human-human dialogs annotated with dialog acts and emotions. (Li et al., 2017). SW has 2,400 human-human telephone conversations that are annotated with topics and dialog acts. SW is a more challenging dataset because it is transcribed from speech which contains complex spoken language phenomenon, e.g. hesitation, self-repair etc.

4.1 Comparing Discrete Sentence Representation Models

The first experiment used PTB and DD to evaluate the performance of the proposed methods in learning discrete sentence representations. We implemented DI-VAE and DI-VST using GRU-RNN (Chung et al., 2014) and trained them using Adam (Kingma and Ba, 2014). Besides the proposed methods, the following baselines are compared. **Unregularized models**: removing the KL$(q|p)$ term from DI-VAE and DI-VST leads to a simple discrete autoencoder (DAE) and discrete skip thought (DST) with stochastic discrete hidden units. **ELBO models**: the basic discrete sentence VAE (DVAE) or variational skip thought (DVST) that optimizes ELBO with regularization term KL$(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))$. We found that standard training failed to learn informative latent actions for either DVAE or DVST because of the posterior collapse. Therefore, KL-annealing (Bowman et al., 2015) and bag-of-word loss (Zhao et al., 2017) are used to force these two models learn meaningful representations. We also include the results for VAE with continuous latent variables reported on the same PTB (Zhao et al., 2017). Additionally, we report the perplexity from a standard GRU-RNN language model (Zaremba et al., 2014).

The evaluation metrics include reconstruction perplexity (PPL), KL$(q(\mathbf{z})\|p(\mathbf{z}))$ and the mutual information between input data and latent vari-

ables $I(\mathbf{x}, \mathbf{z})$. Intuitively a good model should achieve low perplexity and KL distance, and simultaneously achieve high $I(\mathbf{x}, \mathbf{z})$. The discrete latent space for all models are M=20 and K=10. Mini-batch size is 30.

Dom	Model	PPL	KL($q\|p$)	$I(\mathbf{x}, \mathbf{z})$
PTB	RNNLM	116.22	-	-
	VAE	73.49	15.94*	-
	DAE	66.49	2.20	0.349
	DVAE	70.84	0.315	0.286
	DI-VAE	**52.53**	**0.133**	**1.18**
DD	RNNLM	31.15	-	-
	DST	$\mathbf{x}_p$:28.23	0.588	**1.359**
		$\mathbf{x}_n$:28.16		
	DVST	$\mathbf{x}_p$:30.36	**0.007**	0.081
		$\mathbf{x}_n$:30.71		
	DI-VST	$\mathbf{x}_p$:**28.04**	0.088	1.028
		$\mathbf{x}_n$:**27.94**		

Table 1: Results for various discrete sentence representations. The KL for VAE is KL($q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})$) instead of KL($q(\mathbf{z})\|p(\mathbf{z})$) (Zhao et al., 2017)

Table 1 shows that all models achieve better perplexity than an RNNLM, which shows they manage to learn meaningful $q(\mathbf{z}|\mathbf{x})$. First, for auto-encoding models, DI-VAE is able to achieve the best results in all metrics compared other methods. We found DAEs quickly learn to reconstruct the input but they are prone to overfitting during training, which leads to lower performance on the test data compared to DI-VAE. Also, since there is no regularization term in the latent space, $q(\mathbf{z})$ is very different from the $p(\mathbf{z})$ which prohibits us from generating sentences from the latent space. In fact, DI-VAE enjoys the same linear interpolation properties reported in (Bowman et al., 2015) (See Appendix A.2). As for DVAEs, it achieves zero $I(\mathbf{x}, \mathbf{z})$ in standard training and only manages to learn some information when training with KL-annealing and bag-of-word loss. On the other hand, our methods achieve robust performance without the need for additional processing. Similarly, the proposed DI-VST is able to achieve the lowest PPL and similar KL compared to the strongly regularized DVST. Interestingly, although DST is able to achieve the highest $I(\mathbf{x}, \mathbf{z})$, but PPL is not further improved. These results confirm the effectiveness of the proposed BPR in terms of regularizing $q(\mathbf{z})$ while learning meaningful posterior $q(\mathbf{z}|\mathbf{x})$.

In order to understand BPR's sensitivity to batch size N, a follow-up experiment varied the batch size from 2 to 60 (If N=1, DI-VAE is equivalent to DVAE). Figure 2 show that as N increases,
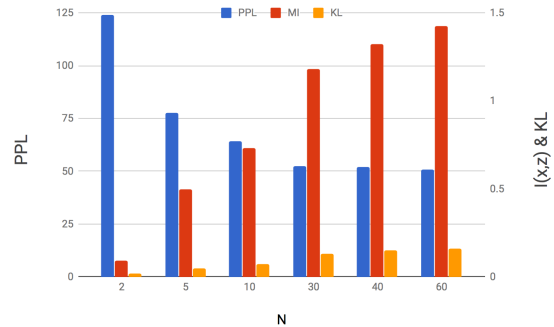

Figure 2: Perplexity and $I(\mathbf{x}, \mathbf{z})$ on PTB by varying batch size N. BPR works better for larger N.

perplexity, $I(\mathbf{x}, \mathbf{z})$ monotonically improves, while KL($q\|p$) only increases from 0 to 0.159. After $N > 30$, the performance plateaus. Therefore, using mini-batch is an efficient trade-off between $q(\mathbf{z})$ estimation and computation speed.

The last experiment in this section investigates the relation between representation learning and the dimension of the latent space. We set a fixed budget by restricting the maximum number of modes to be about 1000, i.e. $K^M \approx 1000$. We then vary the latent space size and report the same evaluation metrics. Table 2 shows that models with multiple small latent variables perform significantly better than those with large and few latent variables.

K, M	K^M	PPL	KL($q\|p$)	$I(\mathbf{x}, \mathbf{z})$
1000, 1	1000	75.61	0.032	0.335
10, 3	1000	71.42	0.071	0.607
4, 5	1024	68.43	0.088	0.809

Table 2: DI-VAE on PTB with different latent dimensions under the same budget.

4.2 Interpreting Latent Actions

The next question is to interpret the meaning of the learned latent action symbols. To achieve this, the latent action of an utterance $\mathbf{x}_n$ is obtained from a greedy mapping: $a_n = \arg\max_k q_{\mathcal{R}}(\mathbf{z} = k|\mathbf{x}_n)$. We set M=3 and K=5, so that there are at most 125 different latent actions, and each $\mathbf{x}_n$ can now be represented by a_1-a_2-a_3, e.g. "How are you?" $\rightarrow$ 1-4-2. Assuming that we have access to manually clustered data according to certain classes

(e.g. dialog acts), it is unfair to use classic cluster measures (Vinh et al., 2010) to evaluate the clusters from latent actions. This is because the uniform prior $p(\mathbf{z})$ evenly distribute the data to all possible latent actions, so that it is expected that frequent classes will be assigned to several latent actions. Thus we utilize the *homogeneity* metric (Rosenberg and Hirschberg, 2007) that measures if each latent action contains only members of a single class. We tested this on the SW and DD, which contain human annotated features and we report the latent actions' homogeneity w.r.t these features in Table 3. On DD, results show DI-VST

	SW		**DD**	
	Act	Topic	Act	Emotion
DI-VAE	0.48	0.08	0.18	0.09
DI-VST	0.33	0.13	0.34	0.12

Table 3: Homogeneity results (bounded [0, 1]).

works better than DI-VAE in terms of creating actions that are more coherent for emotion and dialog acts. The results are interesting on SW since DI-VST performs worse on dialog acts than DI-VAE. One reason is that the dialog acts in SW are more fine-grained (42 acts) than the ones in DD (5 acts) so that distinguishing utterances based on words in $\mathbf{x}$ is more important than the information in the neighbouring utterances.

We then apply the proposed methods to SMD which has no manual annotation and contains task-oriented dialogs. Two experts are shown 5 randomly selected utterances from each latent action and are asked to give an action name that can describe as many of the utterances as possible. Then an Amazon Mechanical Turk study is conducted to evaluate whether other utterances from the same latent action match these titles. 5 workers see the action name and a different group of 5 utterances from that latent action. They are asked to select all utterances that belong to the given actions, which tests the homogeneity of the utterances falling in the same cluster. Negative samples are included to prevent random selection. Table 4 shows that both methods work well and DI-VST achieved better homogeneity than DI-VAE.

Since DI-VAE is trained to reconstruct its input and DI-VST is trained to model the context, they group utterances in different ways. For example, DI-VST would group "Can I get a restaurant", "I am looking for a restaurant" into one action where

Model	Exp Agree	Worker κ	Match Rate
DI-VAE	85.6%	0.52	71.3%
DI-VST	93.3%	0.48	74.9%

Table 4: Human evaluation results on judging the homogeneity of latent actions in SMD.

DI-VAE may denote two actions for them. Finally, Table 4.2 shows sample annotation results, which show cases of the different types of latent actions discovered by our models.

Model	**Action**	**Sample utterance**
DI-VAE	scheduling	- sys: okay, scheduling a yoga activity with Tom for the 8th at 2pm. - sys: okay, scheduling a meeting for 6 pm on Tuesday with your boss to go over the quarterly report.
	requests	- usr: find out if it 's supposed to rain - usr: find nearest coffee shop
DI-VST	ask schedule info	- usr: when is my football activity and who is going with me? - usr: tell me when my dentist appointment is?
	requests	- usr: how about other coffee? - usr: 11 am please

Table 5: Example latent actions discovered in SMD using our methods.

4.3 Dialog Response Generation with Latent Actions

Finally we implement an LAED as follows. The encoder is a hierarchical recurrent encoder (Serban et al., 2016) with bi-directional GRU-RNNs as the utterance encoder and a second GRU-RNN as the discourse encoder. The discourse encoder output its last hidden state $h_{|\mathbf{x}|}^e$. The decoder is another GRU-RNN and its initial state of the decoder is obtained by $h_0^d = h_{|\mathbf{x}|}^e + \sum_{m=1}^M e_m(\mathbf{z}_m)$, where $\mathbf{z}$ comes from the recognition network of the proposed methods. The policy network π is a 2-layer multi-layer perceptron (MLP) that models $p_\pi(\mathbf{z}|h_{|\mathbf{x}|}^e)$. We use up to the previous 10 utterances as the dialog context and denote the LAED using DI-VAE latent actions as AE-ED and the one uses DI-VST as ST-ED.

First we need to confirm whether an LAED can generate responses that are consistent with the semantics of a given $\mathbf{z}$. To answer this, we use a pre-trained recognition network $\mathcal{R}$ to check if a generated response carries the attributes in

the given action. We generate dialog responses on a test dataset via $\tilde{\mathbf{x}} = \mathcal{F}(\mathbf{z} \sim \pi(\mathbf{c}), \mathbf{c})$ with greedy RNN decoding. The generated responses are passed into the $\mathcal{R}$ and we measure *attribute accuracy* by counting $\tilde{\mathbf{x}}$ as correct if $\mathbf{z} = \operatorname{argmax}_k q_{\mathcal{R}}(k|\tilde{\mathbf{x}})$. Table 4.3 shows our generated

Domain	AE-ED	$+L_{\text{attr}}$	ST-ED	$+L_{\text{attr}}$
SMD	93.5%	94.8%	91.9%	93.8%
DD	88.4%	93.6%	78.5%	86.1%
SW	84.7%	94.6%	57.3%	61.3%

Table 6: Results for attribute accuracy with and without attribute loss.

responses are highly consistent with the given latent actions. Also, latent actions from DI-VAE achieve higher attribute accuracy than the ones from DI-VST, because $\mathbf{z}$ from auto-encoding is explicitly trained for $\mathbf{x}$ reconstruction. Adding $\mathcal{L}_{attr}$ is effective in forcing the decoder to take $\mathbf{z}$ into account during its generation, which helps the most in more challenging open-domain chatting data, e.g. SW and DD. The accuracy of ST-ED on SW is worse than the other two datasets. The reason is that SW contains many short utterances that can be either a continuation of the same speaker or a new turn from the other speaker, whereas the responses in the other two domains are always followed by a different speaker. The more complex context pattern in SW may require special treatment. We leave it for future work.

The second experiment checks if the policy network π is able to predict the right latent action given just the dialog context. We report both accuracy, i.e. $\operatorname{argmax}_k q_{\mathcal{R}}(k|\mathbf{x}) = \operatorname{argmax}_{k'} p_\pi(k'|\mathbf{c})$ and perplexity of $p_\pi(\mathbf{z}|\mathbf{c})$. The perplexity measure is more useful for open domain dialogs because decision-making in complex dialogs is often one-to-many given a similar context (Zhao et al., 2017). Table 7 shows the prediction scores on

	SMD	
AE-ED	3.045 (51.5% sys 52.4% usr 50.5%)	
ST-ED	1.695 (75.5% sys 82.1% usr 69.2%)	
	DD	SW
AE-ED	4.47 (35.8%)	4.46 (31.68%)
ST-ED	3.89 (47.5%)	3.68 (33.2%)

Table 7: Performance of policy network. $\mathcal{L}_{attr}$ is included in training.

the three dialog datasets. These scores provide

useful insights to understand the complexity of a dialog dataset. For example, accuracy on open-domain chatting is harder than the task-oriented SMD data. Also, it is intuitive that predicting system actions is easier than predicting user actions on SMD. Also, in general the prediction scores for ST-ED are higher the ones for AE-ED. The reason is related to our previous discussion about the granularity of the latent actions. Since latent actions from DI-VST mainly model the the type of utterances used in certain types of context, it is easier for the policy network to predict latent actions from DI-VST. Therefore, choosing the type of latent actions is a design choice and depends on the type of interpretability that is needed. We finish with an example generated from the two variants of LAED on SMD as shown in Table 8. Given a dialog context, our systems are able to output a probability distribution over different latent actions that have interpretable meaning along with their natural language realizations.

c	**usr**: Where does my friend live?	
Model	Action	Generated Responses
AE-ED	give loc info	- Tom is at 753 University Ave, and a road block.
	$p(\mathbf{z}|\mathbf{c})$=0.34	- Comfort Inn is at 7 miles away.
	give user info	- Your home address is 5671 barringer street.
	$p(\mathbf{z}|\mathbf{c})$=0.22	- Your home is at 10 ames street.
ST-ED	give loc info	- Jill's house is 8 miles away at 347 Alta Mesa Ave.
	$p(\mathbf{z}|\mathbf{c})$=0.93	- Jill lives at 347 Alta Mesa Ave.

Table 8: Interpretable dialog generation on SMD with top probable latent actions. AE-ED predicts more fine-grained but more error-prone actions.

5 Conclusion and Future Work

This paper presents a novel unsupervised framework that enables the discovery of discrete latent actions and interpretable dialog response generation. Our main contributions reside in the two sentence representation models DI-VAE and DI-VST, and their integration with the encoder decoder models. Experiments show the proposed methods outperform strong baselines in learning discrete latent variables and showcase the effectiveness of interpretable dialog response generation. Our findings also suggest promising future research directions, including learning better context-based latent actions and using reinforce-

ment learning to adapt policy networks. We believe that this work is an important step forward towards creating generative dialog models that can not only generalize to large unlabelled datasets in complex domains but also be explainable to human users.

References

Felix Vsevolodovich Agakov. 2005. *Variational Information Maximization in Stochastic Environments*. Ph.D. thesis, University of Edinburgh.

John Langshaw Austin. 1975. *How to do things with words*. Oxford university press.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

Dan Bohus, Antoine Raux, Thomas K Harris, Maxine Eskenazi, and Alexander I Rudnicky. 2007. Olympus: an open-source framework for conversational spoken language interface research. In *Proceedings of the workshop on bridging the gap: Academic and industrial research in dialog technologies*. Association for Computational Linguistics, pages 32–39.

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

Kris Cao and Stephen Clark. 2017. Latent variable dialogue models and their diversity. *arXiv preprint arXiv:1702.05962* .

Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731* .

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* .

Mihail Eric and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. *arXiv preprint arXiv:1705.05414* .

Milica Gašić, Filip Jurčíček, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Gaussian processes for fast policy optimisation of pomdp-based dialogue managers. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, pages 201–204.

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

Zellig S Harris. 1954. Distributional structure. *Word* 10(2-3):146–162.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483* .

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*. pages 1587–1596.

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

Yoon Kim, Kelly Zhang, Alexander M Rush, Yann LeCun, et al. 2017. Adversarially regularized autoencoders for generating discrete structures. *arXiv preprint arXiv:1706.04223* .

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

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

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*. pages 3294–3302.

Staffan Larsson and David R Traum. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural language engineering* 6(3-4):323–340.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155* .

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957* .

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712* .

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644* .

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics* 19(2):313–330.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International Conference on Machine Learning*. pages 1727–1736.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*. volume 2, page 3.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. A hierarchical latent variable encoder-decoder model for generating dialogues. *arXiv preprint arXiv:1605.06069* .

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics* 26(3):339–373.

Aaron van den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*. pages 6309–6318.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research* 11(Oct):2837–2854.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve Young. 2017. Latent intention dialogue models. *arXiv preprint arXiv:1705.10229* .

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language* 21(2):393–422.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2016. Topic augmented neural response generation with a joint attention mechanism. *arXiv preprint arXiv:1606.08340* .

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* .

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

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

Chunting Zhou and Graham Neubig. 2017. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. *arXiv preprint arXiv:1704.01691* .