Variational Autoregressive Decoder for Neural Response Generation

Jiachen Du[1], Wenjie Li[2], Yulan He[3], Lidong Bing[4], Ruifeng Xu[1]*, Xuan Wang[1]
[1]Department of Computer Science, Harbin Institute of Technology (Shenzhen), China
[2] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[3] Department of Computer Science, University of Warwick, United Kingdom
[4] Tencent AI Lab, China
dujiachen@stmail.hitsz.edu.cn, cswjli@comp.polyu.edu.hk
y.he@cantab.net, lyndonbing@tencent.com
xuruifeng@hit.edu.cn, wangxuan@cs.hitsz.edu.cn

Abstract

Combining the virtues of probability graphic models and neural networks, Conditional Variational Auto-encoder (CVAE) has shown promising performance in many applications such as response generation. However, existing CVAE-based models often generate responses from a single latent variable which may not be sufficient to model high variability in responses. To solve this problem, we propose a novel model that sequentially introduces a series of latent variables to condition the generation of each word in the response sequence. In addition, the approximate posteriors of these latent variables are augmented with a backward Recurrent Neural Network (RNN), which allows the latent variables to capture long-term dependencies of future tokens in generation. To facilitate training, we supplement our model with an auxiliary objective that predicts the subsequent bag of words. Empirical experiments conducted on the OpenSubtitle and Reddit datasets show that the proposed model leads to significant improvements on both relevance and diversity over state-of-the-art baselines.

1 Introduction

Recently, variational Bayesian models have shown attractive merits from both theoretical and practical perspectives (Kingma and Welling, 2013). As one of the most successful variational Bayesian models, Conditional Variational Auto-Encoder (CVAE) (Kingma et al., 2014) was proposed to improve upon the traditional Sequence-to-Sequence (Seq2Seq) dialogue models. The CVAE based models incorporate stochastic latent variables into decoders in order to generate more relevant and diverse responses (Serban et al., 2017; Zhao et al., 2017; Shen et al., 2017). However, existing CVAE based models normally rely on the unimodal distribution with a single latent variable to provide the global guidance to response generation, which is not sufficient to capture the complex semantics and high variability of responses. As a result, the autoregressive decoders used in response generation always tend to ignore these oversimple latent variables and degrade the CVAE based model to the simple Seq2Seq model (aka. the *model collapse* problem).

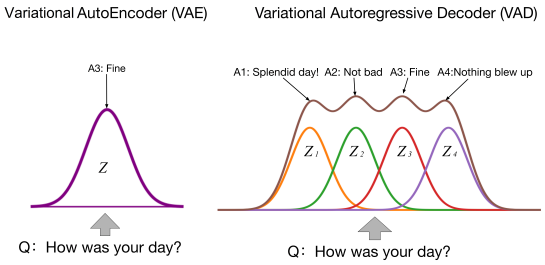

Figure 1: Distributions of latent variable

As illustrated in Figure 1, the unimodal latent variable z used in the conventional VAE usually captures simple unimodal pattern of responses. However, in open-domain conversations, an utterance may have various responses which form complex multimodal distributions. To overcome this problem and improve the quality of generated responses, we propose a novel model, named Variational Autoregressive Decoder (VAD) to iteratively incorporate a series of latent variables into the autoregressive decoder. In particular, a distinct latent variable sampled from CVAE is associated with each time step of the generation, and it is used to condition the next state of the autoregressive decoder (e.g., the hidden state of a RNN). These latent variables at different time steps are integrated by autoregressive decoder to model mutilmodal distribution of text sequences and capture variability of responses as depicted in Figure 1.

*Corresponding author

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3154–3163
Brussels, Belgium, October 31 - November 4, 2018.

Partially inspired by the sequential VAE-based models adopted in speech generation (Goyal et al., 2017; Bayer and Osendorfer, 2014), in our VAD the approximate posterior of the latent variable at each time step is augmented by the corresponding hidden state of a backward RNN running through the remaining response sequence. Since the hidden states of the backward RNN contain the information of the succeeding words in the response, they can be used as the guidance for the latent variables to capture the long-term dependency on the future content.

It has been found that auxiliary losses that predict another task-related objective could help latent variables capture more information from different perspectives when training the VAE based models (Zhao et al., 2017). To enhance VAD, we propose a purposely designed auxiliary loss to use the latent variable at each time step to predict the Bag-Of-Words (BOW) of the succeeding subsequence. The proposed auxiliary loss could essentially help VAD to generate more coherent responses.

Experimental results show that the proposed VAD model outperforms the conventional response generation models when evaluated automatically and manually on the OpenSubtitle and Reddit datasets. The contributions in this work are two-fold:

- We propose a novel VAD model for response generation that can better capture the high variability of responses by sequentially associating latent variables to different time steps of autoregressive decoder and approximating the posterior of latent variables by augmenting the hidden states of a backward RNN.
- A BOW based auxiliary objective is proposed to help preserving the diversity of generated responses.

2 Related Work

2.1 Conversational Systems

As neural network based models dominate the research in natural language processing, Seq2Seq models have been widely used for response generation (Sordoni et al., 2015). However, Seq2seq models suffer from the problem of generating generic responses, such as *I don't know* (Li et al., 2016a). Various approaches have been proposed to address this problem, including adding additional information (Li et al., 2016b; Xing et al., 2017; Zhou et al., 2017b) and modifying the architecture of existing models (Li et al., 2016a; Xu et al., 2017; Zhou et al., 2017a).

Another solution to address this problem is to add stochastic latent variables in order to change the deterministic structure of Seq2Seq models. VAE (Kingma and Welling, 2013) is one of the most successful models (Serban et al., 2017; Zhao et al., 2017; Shen et al., 2017; Cao and Clark, 2017). However, VAE-based models only use a single latent variable to encode the whole response sequence, thus suffering from the ***model collapse*** problem (Bowman et al., 2016). To overcome this problem, we propose a novel model that based on the variational autoregressive decoder to better represent highly structural latent variables.

2.2 Variational Autoregressive Models

Recently, some works attempted to combine VAE with autoregressive models to better process input sequences. Broadly speaking, they can be categorized into two groups. Methods in the first group leverage autoregressive models to improve the inference of traditional VAEs. The most well-known model is Inverse Autoregressive Flow (IAF), which used a series of invertible transformations based on the autoregressive model to construct the latent variables (Kingma et al., 2016; Chen et al., 2017). Methods in the second group focus on improving autoregressive models like RNNs by adding variational inference (Bayer and Osendorfer, 2014; Chung et al., 2015; Fraccaro et al., 2016; Goyal et al., 2017). These models usually modeled continuous data such as images and audio signals. For dealing with discrete data such as text, (Li et al., 2017) applied variational recurrent neural networks (VRNN) for text summarization.

Our proposed framework is based on the second line of research, but is different from the previous research as it develops a new strategy of combining VAE with RNN for response generation.

3 Proposed VAD Model

As shown in Figure 2, we use the Seq2Seq model as the basic architecture. The Seq2Seq model is an encoder-decoder neural framework for mapping a source sequence to a target sequence (Sutskever et al., 2014). The input of Seq2seq response generation model is variable-length query sequence

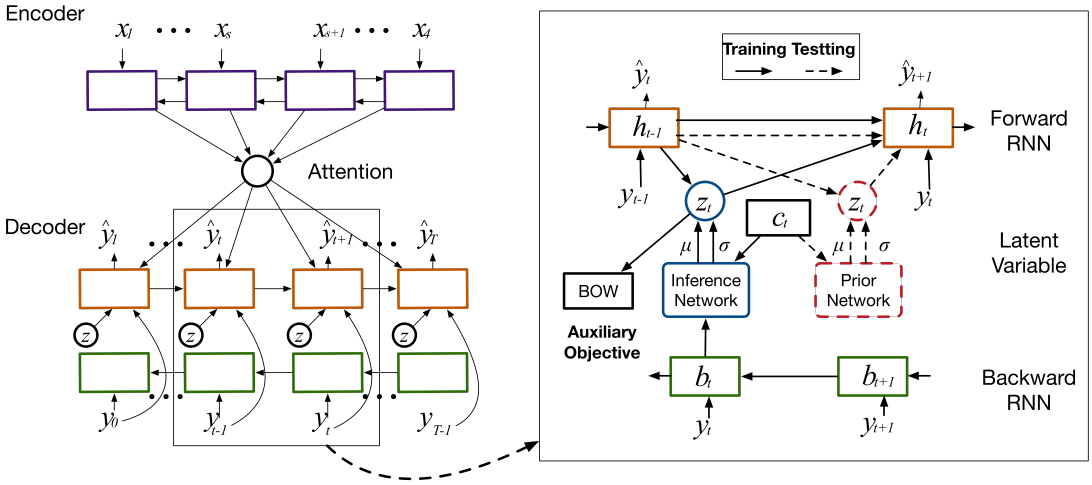

Figure 2: Sequence-to-sequence model using sequential variational decoder.

$\boldsymbol{x} = \{x_1, \ldots, x_m\}$, and the output is a response sequence $\boldsymbol{y} = \{y_1, \ldots, y_n\}$. Both the encoder and decoder are the Recurrent Neural Networks (RNN) with Gated Recurrent Units (GRU) (Chung et al., 2014).

The encoder is a bidirectional GRU that encodes the query sequence as the concatenation of the hidden states of a forward and a backward GRUs. The semantic of word t in the query sequence is represented by $h_t^e = [\overrightarrow{h_t^e}, \overleftarrow{h_t^e}]$, where

$$\begin{aligned} \overrightarrow{h_t^e} &= \overrightarrow{\text{GRU}}(x_t, \overrightarrow{h_{t-1}^e}) \\ \overleftarrow{h_t^e} &= \overleftarrow{\text{GRU}}(x_t, \overleftarrow{h_{t+1}^e}) \end{aligned} \tag{1}$$

The decoder is a GRU with hidden state h_t^d at each step. The input at step t is the concatenation of previous word in response sequence y_{t-1} and the context vector c_t computed by a neural attention model. The context vector c_t is the weighted sum of the whole encoder's hidden states computed by:

$$\begin{aligned} \alpha_{s,t} &= f_{attention}([h_s^e, h_{t-1}^d]) \\ c_t &= \sum_{s=1}^{m} \alpha_{s,t} h_s^e \end{aligned} \tag{2}$$

where $f_{attention}$ is a one-layer neural network that produces attention weights, $\alpha_{s,t}$ is the attention weight evaluating the correlation between encoder's hidden state h_s^e and hidden state of decoder h_{t-1}^d. The decoder predicts the next word $\hat{y}_t$ by jointly considering previous word y_{t-1}, attentional context c_t and previous hidden state h_{t-1}^d.

3.1 Conditional Variational Auto-Encoder

The decoder of VAD is based on the Conditional VAE (CVAE) framework (Kingma et al., 2014), which approximates the distribution of random variable $\boldsymbol{y}$ (response) conditioned on $\boldsymbol{x}$ (i.e., query) by incorporating an latent variable z. CVAE introduces a parameterized conditional posterior distribution $q_\theta(z|\boldsymbol{y}, \boldsymbol{x})$ to approximate true posterior distribution $p(z|\boldsymbol{y}, \boldsymbol{x})$. By injecting $q_\theta(z|\boldsymbol{y}, \boldsymbol{x})$, the conditional marginal distribution of $p(\boldsymbol{y}|\boldsymbol{x})$ can be maximized by approximating the Evidence Lower Bound (ELBO):

$$\log p_\phi(\boldsymbol{y}|\boldsymbol{x}) \geq \log p(\boldsymbol{y}|\boldsymbol{x}) - \text{KL}(q_\theta(z|\boldsymbol{y}, \boldsymbol{x})||p(z|\boldsymbol{y}, \boldsymbol{x}))$$

where KL denotes the Kullback-Leibler divergence. ELBO can be rewritten as a regularized auto-encoder function:

$$\mathcal{L} = \mathbb{E}_{q_\theta(z|\boldsymbol{y},\boldsymbol{x})}[p_\phi(\boldsymbol{y}|z, \boldsymbol{x})] - \text{KL}(q_\theta(z|\boldsymbol{y}, \boldsymbol{x})||p_\phi(z|\boldsymbol{x}))$$

where $p_\phi(\boldsymbol{y}|z, \boldsymbol{x})$ is the decoder that decodes $\boldsymbol{y}$ from the latent variable z and conditional variable $\boldsymbol{x}$, $q_\theta(z|\boldsymbol{y}, \boldsymbol{x})$ is the inference model that approximates the true posterior, $p_\phi(z|\boldsymbol{x})$ is the prior model that samples the latent variable from the prior distribution, θ, ϕ are the parameters of the inference and decoder models, respectively. All parameterized distributions are modeled by neural networks.

In the training phase, the latent variable z is sampled from both the inference model and the prior model. z from the inference model is then used to condition the generated distribution $p(\boldsymbol{y}|z, \boldsymbol{x})$. Meanwhile, CVAE minimizes the KL

divergence between the latent variables from these two models. This process makes it possible for CVAE to samples z from the prior model only when decoding in the testing phase.

Different from the previous work on CVAE-based response generation that only relys on a single latent variable (Serban et al., 2017; Zhao et al., 2017; Shen et al., 2017), our proposed model incorporates a series of latent variables into the autoregressive decoder. Inspired by the work on variational recurrent neural networks (Goyal et al., 2017; Bayer and Osendorfer, 2014), our model sequentially decodes the response sequence conditioned on the latent variable z_t at each time step by $p_\phi(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{x}) = \prod_t p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{z}_t, \boldsymbol{x})$.

3.2 Variational Autoregressive Decoder

Traditional CVAE-based models only use a single standard normal distribution to model the latent variable z. They are usually difficult to model the multi-modal distribution of responses $p(\boldsymbol{y}|z, \boldsymbol{x})$. To overcome this limitation, we propose a Variational Autoregressive Decoder (VAD) that decomposes $\boldsymbol{z}$ into sequential variables z_t at each time step t during response generation. Owing to the autoregressive structure of VAD, the hidden state of backward RNN $\overleftarrow{h}^d_t$ is used to condition the latent variable z_t, which can be seen as a long-term guidance to the generation. Moreover, we propose a novel auxiliary objective, which is specially designed for VAD, to avoid *model collapse*.

At each time step, the decoder uses a forward GRU to process the sequence and predicts the next token by a feed-forward network f_{output} with the softmax activation function. The input to GRU is the combination of the previous word's embedding y_{t-1}, the context vector produced by an attention model c_t and the latent variable z_t. The process is described by,

$$\overrightarrow{h}^d_t = \overrightarrow{\text{GRU}}([y_{t-1}, c_t, z_t], \overrightarrow{h}^d_{t-1}) \quad (3)$$

$$p_\phi(y_t|\boldsymbol{y}_{<t}, \boldsymbol{z}_t, \boldsymbol{x}) = f_{\text{output}}([\overrightarrow{h}^d_t, c_t]) \quad (4)$$

where, $\overrightarrow{h}^d_t$ is the hidden state produced by the forward GRU at time step t. c_t is the attentional weighted sum of the encoder's output.

Inference Model We use the hidden states of the backward RNN running through the response sequence as an additional input to the inference model. The backward RNN processes the sequence by,

$$\overleftarrow{h}^d_t = \overleftarrow{\text{GRU}}(y_{t+1}, \overleftarrow{h}^d_{t+1}) \quad (5)$$

The backward hidden state $\overleftarrow{h}^d_t$ contains the information of succeeding tokens, and it serves as a future plan for generation. By combining the information produced by the backward RNN, the inference model has a better capability of approximating the real posterior distribution.

Considering context variable c_t at each time step as a substitute of the condition variable $\boldsymbol{x}$ in (3.1), c_t is also fed to the inference model. The inference model is a feed-forward neural network f_{infer}. The approximated distribution $q(z_t|\boldsymbol{y}, \boldsymbol{x})$ is a normal distribution $\mathcal{N}(\mu^i, \sigma^i)$, which is parameterized by the output of f_{infer}:

$$[\mu^i, \sigma^i] = f_{\text{infer}}([\overrightarrow{h}^d_{t-1}, c_t, \overleftarrow{h}^d_t]) \quad (6)$$

$$q_\theta(z_t|\boldsymbol{y}, \boldsymbol{x}) = \mathcal{N}(\mu^i, \sigma^i) \quad (7)$$

where the sampling process of z_t is done by reparameterization (Kingma and Welling, 2013).

Prior Model The prior network can only use the observable variables in the testing phase to sample z_t. The observable variables include the previous hidden state $\overrightarrow{h}^d_{t-1}$ and the context variable c_t. The prior model is also modeled by a feed-forward network f_{prior} as follows.

$$[\mu^p, \sigma^p] = f_{\text{prior}}([\overrightarrow{h}^d_{t-1}, c_t]) \quad (8)$$

$$p_\phi(z_t|\boldsymbol{y}_{<t}, \boldsymbol{x}) = \mathcal{N}(\mu^p, \sigma^p) \quad (9)$$

where μ^p, σ^p are the parameters of prior normal distribution.

Auxiliary Objective As discussed in Section 1, the decoder based on the autoregressive model often ignores the latent variables and causes the model to collapse. One way to alleviate this problem is to add an auxiliary loss to the training objective (Zhao et al., 2017; Goyal et al., 2017). To allow the latent variables to capture the information from a different perspective, we use ***Sequential Bag of Word (SBOW)*** as the auxiliary objective for the proposed VAD model. The idea of the ***SBOW*** auxiliary objective is to sequentially predict the bag of succeeding words $\boldsymbol{y}_{bow(t+1,T)}$ in the response using the latent variable z_t at each

time step. This auxiliary objective can be seen as the prediction of candidate words for future generation.

Our ***SBOW*** is specially designed for VAD. It is different from the Bag-of-Words (BOW) auxiliary loss used in the CVAE-based models (Zhao et al., 2017), which only uses the latent variable to predict the Bag-Of-Words of the whole sequence. VAD with ***SBOW*** sequentially produces the auxiliary loss for each time step of generation. The auxiliary loss at each time step is computed by

$$p_\xi(\boldsymbol{y}_{bow(t+1,T)}|z_{t:T}) = f_{\text{auxiliary}}(z_t) \quad (10)$$

where $\boldsymbol{y}_{bow(t+1,T)}$ is the bag-of-word vector of the words from $t+1$ to T in the response, and $f_{\text{auxiliary}}$ is a feed-forward neural network with the softmax output.

3.3 Learning

The loss function of our model is the sum of the losses at each time step, including the weighed sum of the ELBO loss $\mathcal{L}_{ELBO}(t)$ and the auxiliary loss $\mathcal{L}_{AUX}(t)$ where $\mathcal{L}_{ELBO}(t)$ can be further decomposed into a log-likelihood loss and the KL divergence:

$$\begin{aligned}\mathcal{L} &= \sum_t [\mathcal{L}_{ELBO}(t) + \alpha\mathcal{L}_{AUX}(t)] \\ &= \sum_t [(\mathcal{L}_{LL}(t) - \mathcal{L}_{KL}(t)) + \alpha\mathcal{L}_{AUX}(t)]\end{aligned} \quad (11)$$

Here, $\mathcal{L}_{LL}(t)$ denotes the log-likelihood loss when predicting y_t. $\mathcal{L}_{KL}(t)$ is the KL-divergence of the approximate posteriori q_θ and priori p_ϕ at time step t. $\mathcal{L}_{AUX}(t)$ is the auxiliary loss when predicting ***SBOW*** as described in Section 3.2. α is the weight controlling the auxiliary loss. The losses are computed by

$$\begin{aligned}\mathcal{L}_{LL}(t) &= \mathbb{E}_{q_\theta(z_t|\boldsymbol{y},\boldsymbol{z})}\left[\log p_\theta(y_t|\boldsymbol{y}_{<t}, z_t, x_t)\right] \\ \mathcal{L}_{KL}(t) &= \text{KL}(q_\theta(z_t|\boldsymbol{y},\boldsymbol{x})||p_\phi(z_t|\boldsymbol{y}_{<t},\boldsymbol{x})) \\ \mathcal{L}_{AUX}(t) &= \mathbb{E}_{q_\theta(z_t|\boldsymbol{y},\boldsymbol{z})}\left[\log p_\xi(\boldsymbol{y}_{bow(t+1,T)}|z_t)\right]\end{aligned}$$

All the parameters are learned by optimizing Equation (11) and updated with back-propagation.

4 Experimental Setup

4.1 Datasets

We evaluate the proposed model on two datasets: **OpenSubtitles** and **Reddit**. The OpenSubtitles dataset contains subtitles for movies in various languages. Here, we only choose the English version of OpenSubtiles. The Reddit dataset is crawled from comments of Reddit[1] which is an American social news discussion website. We collected more than 10 million single-turn dialogues from 100 topics posted in 2017. For each dataset, we randomly select 6 million conversations for training, 10k for validation and 5k for testing. For every conversation, we remove the sentences whose length is shorter than 6 words and only keep the first 40 words for sentences longer than 40. We keep top 15k frequent words as the vocabulary for OpenSubtitles and 20k frequent words for Reddit.

4.2 Hyper-parameters and Training Setup

We use the pre-trained GloVe 300-dimensional word embeddings for both the encoder and the decoder. The encoder is a bidirectional RNN with GRU with the size of the hidden state set to 512. The size of the hidden states of GRU in the decoder is also set to 512. We apply Layer Normalization when training the decoder. The size of the latent variables is set to 400. The inference network and the prior network are all one-layer feed-forward network. All weights are initialized by the xavier method (Glorot and Bengio, 2010). The model is trained end-to-end by Adam optimizer (Kingma and Ba, 2014) with the learning rate set to 10^{-4} and gradient clipped at 1. When generating text, we adopt the greedy strategy and the KL-annealing strategy, with the temperature varying from 0 to 1 and increased by 10^{-5} after each iteration of batch update.

4.3 Baselines

We compare our proposed model with the following three baselines:

- **Seq2Seq**: Sequence-to-Sequence model with attention (Sordoni et al., 2015).
- **CVAE**: Conditional Variational Auto-Encoder for generating responses (Serban et al., 2017). Different from our model, CVAE uses a unimodal Gaussian distribution to model the whole response and append the output of VAE as an additional input to decoder. We also use the KL annealing strategy when training CVAE with the same parameter setting as in our model.

[1] http://www.reddit.com

- **CVAE+*BOW* loss**: CVAE model with the auxiliary bag-of-words loss (Zhao et al., 2017).

4.4 Metrics

We employ three types of commonly used automatic evaluation metrics and human evaluation in our experiments:

Embedding Similarity: Embedding-based metrics compute the cosine similarity between the sentence embedding of a ground-truth response and that of the generated one. There are various ways to derive the sentence-level embedding from the constituent word embeddings. In our experiments, we apply three most commonly used strategies to obtain the sentence-level embeddings. $\textbf{EMB}_\textbf{A}$ calculates the average of word embeddings in a sentence. $\textbf{EMB}_\textbf{E}$ takes the most extreme value among all words for each dimension of word embeddings in a sentence. $\textbf{EMB}_\textbf{G}$ greedily calculates the maximum of cosine similarity of each token in two sentences and take the average of them to get the final matching score (Liu et al., 2016).

RUBER Score: RUBER (Referenced metric and Unreferenced metric Blended Evaluation Routine) is a newly proposed metric for evaluating the quality of response in conversations that show high correlation with human annotation (Tao et al., 2017). RUBER evaluates the generated responses by taking into account both the ground-truth responses and the given queries. For the *referenced* metric, RUBER calculates the embedding-based cosine similarity between a generated response and its corresponding ground-truth. For the *unreferenced* metric, RUBER firstly trains a neural network by a response retrieval task and evaluates the relatedness between a generated response and its query. Evaluating RUBER score can be treated as a rough simulation to the well-known *Turing Test*. For blending the two metrics, there are two strategies: taking the geometric mean ($\textbf{RUB}_\textbf{G}$) or the arithmetic mean ($\textbf{RUB}_\textbf{A}$). The RUBER score ranges between 0 and 1 and higher scores imply better relatedness.

Diversity: Diversity metrics evaluate the informativeness and diversity of generated responses. In our experiments, we use $\textbf{Dist}_\textbf{1}$ and $\textbf{Dist}_\textbf{2}$ (Li et al., 2016a) to evaluate the diversity and **Entropy** to measure the informativeness. $\textbf{Dist}_\textbf{1}$ (or $\textbf{Dist}_\textbf{2}$) calculates the ratio of the number of unique unigrams (or bigrams) against the total number of unigrams (or bigrams). Higher $\textbf{Dist}_\textbf{1}$ (or $\textbf{Dist}_\textbf{2}$) implies more diverse vocabularies used in responses. **Entropy** as a metric proposed by (Serban et al., 2017) calculates the average entropy in a generated response. According to information theory, it is known that low-frequent words have higher entropy and carries more information. Therefore, we use this **Entropy** to measure the informativeness and diversity of the generated responses. The unit of **Entropy** is *bit* and Higher **Entropy** correlates to more informative response.

Human Evaluation: In human evaluation, 10 research students are arranged to rate the generated responses generated by CVAE with *BOW* auxiliary loss and our model. We randomly selected 100 queries from the Reddit dataset[2] and used each model to generate the best responses. Each query with its ground-truth response and the two generated responses are simultaneously shown to the human evaluators. The evaluators are asked to rate the responses based on grammatical correctness, coherence and relevance to queries (tie is permitted).

5 Results

5.1 Quantitative Analysis

The experimental results evaluated by automatic metrics on the OpenSubtitles and the Reddit datasets are shown in Table 1 and 2, respectively. It is observed that both CVAE-based models and our proposed models outperform Seq2Seq by a large margin, showing the effectiveness of adding variational latent variable for response generation. However, using different structure of variational models leads to differences in performance on both plausibility and diversity. Our model with or without the SBOW auxiliary loss outperforms CVAE as observed by the significant boost in semantic relevance-oriented metrics (embedding similarities and RUBER score) and diversity-oriented metrics. This is mainly due to the different strategy employed for representing latent variables. CVAE only uses a unimodal latent variable as the semantic signal of the whole response sequence which limits its capability of capturing

[2]The reason of not conducting the human evaluation on the OpenSubtitles dataset is that query-response pairs in the OpenSubtitles dataset are extracted from movie scripts and hence are more difficult to evaluate without the context information.

Method	Embedding Similarity			RUBER		Diversity		
	EMB_A	EMB_E	EMB_G	Rub_G	Rub_A	$Dist_1$	$Dist_2$	Entropy
Ground Truth	1.000	1.000	1.000	0.872	0.881	0.091	0.423	11.886
Seq2Seq	0.572	0.493	0.487	0.441	0.462	0.015	0.053	6.730
CVAE	0.639	0.531	0.578	0.562	0.580	0.026	0.102	8.215
CVAE+BOW loss	0.659	0.530	0.526	0.602	0.597	0.041	0.302	9.519
Ours (without *SBOW*)	0.678	0.520	0.563	0.591	0.604	0.031	0.259	8.815
Ours	**0.714**	**0.582**	**0.642**	**0.635**	**0.642**	**0.053**	**0.404**	**10.976**

Table 1: Experimental results on the OpenSubtitles dataset.

Method	Embedding Similarity			RUBER		Diversity		
	EMB_A	EMB_E	EMB_G	Rub_G	Rub_A	$Dist_1$	$Dist_2$	Entropy
Ground Truth	1.000	1.000	1.000	0.842	0.869	0.083	0.399	10.089
Seq2Seq	0.520	0.382	0.377	0.371	0.386	0.007	0.042	6.003
CVAE	0.602	0.496	0.531	0.541	0.555	0.019	0.097	7.010
CVAE+BOW loss	0.659	0.531	0.578	0.591	0.604	0.026	0.282	9.215
Ours (without *SBOW*)	0.628	0.540	0.563	0.607	0.610	0.021	0.216	8.222
Ours	**0.692**	**0.556**	**0.598**	**0.622**	**0.629**	**0.046**	**0.391**	**10.043**

Table 2: Experimental results on the Reddit dataset.

Models	OpenSubtitles	Reddit
Ground Truth	15.31	17.48
CVAE+*BOW*	9.66	10.83
Ours	11.81	14.09

Table 3: The average length of responses.

Models	Wins	Loses	Ties
CVAE+*BOW*	0.207	0.678	0.115
Ours	0.685	0.200	0.115

Table 4: Results of human judgment on the generated responses.

variability of response sequences. By incorporating a series of time-varying latent variables into each step of autoregressive decoder, our model is able to model more complicated multimodal distributions of response sequences and capture more detailed semantic information.

Since adding the auxiliary loss could alleviate the *model collapse* problem, we found that CVAE model with the *BOW* auxiliary loss outperforms our basic model without auxiliary loss, especially on the diversity metrics. When adding the proposed *SBOW* auxiliary loss into our model, we found that our generated responses have shown better diversity compared to those generated by CVAE+*BOW* loss. The encouraging improvement is attributed to the autoregressive structure of our variational inferences, which makes it possible to gradually introduce additional information of *SBOW*. To better demonstrate the impact of *SBOW*, we calculate the average length of the generated responses of our model and CVAE with *BOW* loss and show the results in Table 3. It is observed that our model with *SBOW* can generate longer responses than CVAE+*BOW*. The results validate the effectiveness of adding the *SBOW* auxiliary objective into our model.

The evaluation results of human judgment is shown in Table 4. It is observed that the responses generated by our proposed VAD is more plausible than CVAE+*BOW* from human perspectives. We also conduct t-test to compare our model with CVAE+*BOW*. The results show that the improvement of VAD over CVAE+*BOW* is statistically significant ($p < 0.01$).

5.2 Qualitative Analysis

Case Study To empirically analyze the quality of the generated responses, we show some example responses generated by our model and two baselines (Seq2Seq and CVAE+*BOW*) in Table 5. It is observed that Seq2seq often generates generic responses that starting with '*I don't know*' or '*I am not sure*', since the deterministic structure of Seq2seq limits the diversity of generation. Injecting variational latent variables avoids dull responses as can be seen from the responses gen-

Dataset	Query	Seq2seq	CVAE+*BOW*	Ours
OpenSub	why he is not here ?	i do n't know where	he 's not present .	maybe he 's been caught in the rain too .
	that wasn ' t easy for him .	i ' m not so sure i know.	did he do good job and good job ?	it 's a tough job but it 's a great deal of money .
	what is the alternative solution ?	i ' m not interested.	i am the solution.	i ' m afraid there is no alternative .
Reddit	he looks exactly like my australian uncle .	he is NUM NUM	lol , you know him , he is my uncle .	nope , he 's a young man and was born in LOC .
	why no windows ?	i ' m not so sure you 're right .	do you like windows NUM ?	cuz linux can be a great os .
	i hope the grand tour will make an episode .	i ca n't commit post though .	i 'm wondering getting it .	i would hope that it will be on netflix as well .

Table 5: Example responses generated by our model and two baselines (Seq2Seq and CVAE+*BOW*) from the OpenSubtitles and the Reddit Datasets.

erated by CVAE+*BOW* and our model. However, we found that CVAE+*BOW* tends to copy the given queries (the first and fourth example in Table 4) and repeatedly generate redundant tokens (the second example). The generated responses of our model are more fluent and relevant to queries. Also, our model generates longer responses compared to the baselines.

KL Divergence Visualization In order to demonstrate that our model is able to alleviate the ***model collapse*** problem of VAE, we visualize the KL divergence between the approximate posterior distribution $q_\theta(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{x})$ and priori $p_\phi(\boldsymbol{z}|\boldsymbol{x})$ during the training process of our models and CVAE with *BOW* loss in Figure 3. As we know, when variational models ignore the latent variable, the generated value $\boldsymbol{y}$ will be independent of the latent variable $\boldsymbol{z}$ which causes the KL divergence in Equation (3.1) to approach 0. The higher KL value during training means more dependence between $\boldsymbol{y}$ and $\boldsymbol{z}$. In this experiment, we use the same KL annealing strategies for our model and CVAE+*BOW* as described in Section 4.2. The KL divergence of the two models on the OpenSubtitles and the Reddit datasets during training is plotted in Figure 3. It is observed that the KL divergence of our model converges to a higher value compared to that of CVAE+*BOW*. It shows that our model could better alleviate the *model collapse* problem.

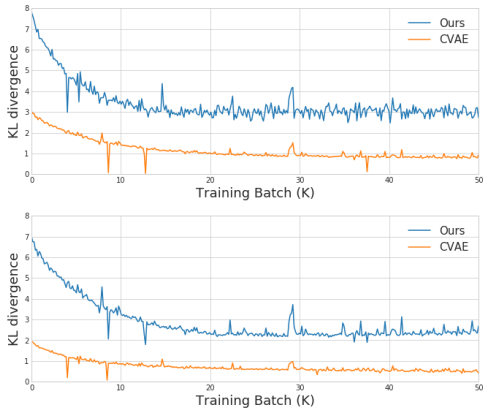

Figure 3: KL divergence during training.

6 Conclusion

In this paper, a novel variational autoregressive decoder is proposed to improve the performance of VAE-based models for open-domain response generation. By injecting the variational inference into the RNN-based decoder and applying carefully designed conditional variables and auxiliary objective for latent variables, the proposed model is expected to better modeling semantic information of text in conversations. Quantitative and qualitative experimental results show clear performance improvement of the proposed model over competitive baselines. In future works, we will explore the use of other attributes of responses such as Part-of-Speech (POS) tags and chunking sequences as additional conditions for better response generation.

Acknowledgements

This work was supported by National Natural Science Foundation of China U1636103, 61632011, Key Technologies Research and Development Program of Shenzhen JSGG20170817140856618, Shenzhen Foundational Research Funding 20170307150024907, Research Grants Council of Hong Kong (PolyU 152036/17E, 152040/18E), The Hong Kong Polytechnic University (G-YBJP, G-YBP6). and Innovate UK (grant no. 103652).

References

Justin Bayer and Christian Osendorfer. 2014. Learning stochastic recurrent networks. In *NIPS 2014 Workshop on Advances in Variational Inference*.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, page 10.

Kris Cao and Stephen Clark. 2017. Latent variable dialogue models and their diversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 182–187.

Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2017. Variational lossy autoencoder. In *5th International Conference on Learning Representations (ICLR)*.

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*.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. 2015. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988.

Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. 2016. Sequential neural models with stochastic layers. In *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS)*, pages 2199–2207.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 249–256.

Anirudh Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. 2017. Z-forcing: Training stochastic recurrent networks. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*, pages 6697–6707.

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

Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. Improved variational inference with inverse autoregressive flow. In *Proceedings of the 30th Advances in Neural Information Processing Systems*, pages 4743–4751.

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

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of NAACL-HLT*, pages 110–119.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 994–1003.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100.

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

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pages 3295–3301.

Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. 2017. A conditional variational framework for dialog generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (EMNLP)*, pages 504–509.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 196–205.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2017. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. *arXiv preprint arXiv:1701.03079*.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*.

Zhen Xu, Bingquan Liu, Baoxun Wang, SUN Chengjie, Xiaolong Wang, Zhuoran Wang, and Chao Qi. 2017. Neural response generation via gan with an approximate embedding layer. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 617–626.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 654–664.

Ganbin Zhou, Ping Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. 2017a. Mechanism-aware neural machine for dialogue response generation. In *Proceddings of 31st AAAI Conference on Artificial Intelligence*.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2017b. Emotional chatting machine: emotional conversation generation with internal and external memory. *arXiv preprint arXiv:1704.01074*.