Abstract

Dialog response generation in open domain is an important research topic where the main challenge is to generate relevant and diverse responses. In this paper, we propose a new dialog pre-training framework called DialogVED, which introduces continuous latent variables into the enhanced encoder-decoder pre-training framework to increase the relevance and diversity of responses. With the help of a large dialog corpus (Reddit), we pre-train the model using the following 4 tasks, used in training language models (LMs) and Variational Autoencoders (VAEs) literature: 1) masked language model; 2) response generation; 3) bag-of-words prediction; and 4) KL divergence reduction. We also add additional parameters to model the turn structure in dialogs to improve the performance of the pre-trained model. We conduct experiments on PersonaChat, DailyDialog, and DSTC7-A VSD benchmarks for response generation. Experimental results show that our model achieves the new state-of-the-art results on all these datasets.

1 Introduction

Pre-trained language models (PLMs) have been widely explored both in natural language understanding (NLU) and generation (NLG) in recent years, this pre-training and fine-tuning paradigm sheds light on various downstream tasks in natural language processing (NLP). Compared with general pre-trained models, task-oriented pre-trained models (such as Summarization, Dialog and etc.), which is designed in line with task characteristics, may achieve better performance and be more robust. In this paper, we propose a novel pre-trained dialog response generation model based on previous research.

Dialogue Response Generation (DSG) in open domain is a challenging task with a wide range of application scenarios. Recent advances in DSG utilize pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) and GPT2 (Radford et al., 2019) in two major categories. The first one focuses on how to fine-tune PLMs in downstream tasks and address the various application-specific needs and challenges (Lin et al., 2020). The second one augments dialog specific tasks into the PLM training (Zhang et al., 2020; Bao et al., 2020) and then fine-tunes the new pre-trained model in downstream tasks. We study the latter in this paper.

There is a proverbial one-to-many problem in DSG, i.e., a single dialog context could be followed by multiple reasonable responses. Existing works introduce latent variables to model this problem. For example, VHRED (Serban et al., 2017) incorporates latent continuous variable into the sequence-to-sequence (Seq2Seq) RNN model to improve the diversity of generated responses. VAE-Seq2Seq (Bahuleyan et al., 2017) proposes variational attention to replace the vanilla encoder-decoder attention (Luong et al., 2015), to avoid attention to bypass the latent space and invalidate the latent variable. For controllability and interpretability, some discrete VAEs have also been proposed, such as (Oord et al., 2017; Vahdat et al., 2018).

Recently, PLATO (Bao et al., 2020) firstly introduces latent variables into their pre-training dialog model, where the authors introduce a $K$-way ($K = 20$) categorical latent variable, and the pre-trained model shows significant gains in multiple downstream response generation tasks. Continuous latent variables besides discrete latent variables is popularly used for modeling one-to-many mapping in dialog system, but the potential of incorporating continuous latent variables with large-scale language pretraining is less explored.

In this paper, we propose a pre-trained latent Variable Encoder-Decoder model for Dialog generation, which is called DialogVED. In this model, we introduce a continuous latent variable into the enhanced encoder-decoder pre-training framework and we adopt the optimization techniques based on
the VAEs literature to learn the model with continuous latent variables. More specifically, we conduct the pre-training by optimizing the following 4 pre-training objectives simultaneously: 1) masked language spans loss to enhance the encoder’s understanding of context, 2) response generation with n-gram loss to improve the decoder’s planning ability, 3) Kullback-Leibler divergence loss to minimize the difference between the posterior and prior distribution of the latent variables, and 4) bag-of-words loss to reduce posterior distribution collapse.

In addition, we also explore the effect of absolute and relative position embeddings specific for conversational data on the model performance.

We conduct experiments on three different kinds of conversation tasks: chit-chat, knowledge grounded conversation, and conversational question answering. Experimental results verify the effectiveness and superiority of our model compared with the previous state-of-the-art method. We further carry out ablation study to better understand the impact of different components in the DialogVED on model performance including latent space sizes, different decoding strategies, and position embeddings for turns and roles.

Our pre-trained models and source code will be released, hoping to facilitate further research progress in dialogue generation. The main contributions of this paper can be summarized as follows: 1) We propose a pretrained dialog model, which incorporates continuous latent variables into the enhanced encoder-decoder pre-training framework; We explore the impact of latent variable sizes, different decoding strategies, and position embeddings for turns and roles in our model; Extensive experiments show that the proposed model achieves the new state-of-the-art (SOTA) in multiple downstream tasks, and our model has better performance both on relevance and diversity than previous SOTA in response generation.

2 Method

2.1 Model Architecture

In response generation, there are three elements: dialogue context c, response r and latent variable z. The dialogue context c may consist of several history utterances (i.e., multi turns) and the response r is one piece of appropriate reply towards the given context. Additionally, the latent variable z in the latent space represents many unobserved factors associating the context and the response.

We assume the latent variable z is continuous, which is different from PLATO (Bao et al., 2020), and portrays a certain conditional probability distribution related to the response given context. We then define the conditional distribution \( p(r, z|c) = p(r|c, z)p(z|c) \) and our goal is to use encoder-decoder models (parameterized by \( \theta \)) to approximate \( p(r|c, z) \) and a multi-layer perceptron (parametrized by \( \phi \)) to estimate \( p(z|c) \), which is called the prior network in VAE literature. We call the final pre-trained model DialogVED, which is a transformer-based encoder-decoder model with an extra prior network for modeling the latent space. Figure 1 gives a overview of our model.

Notation of some parameters: We denote the same number of layers in the encoder and decoder as L, the hidden size as H, and the number of self-attention heads as A. We denote the dimension of latent variable as P.

2.2 Encoder

We use multi-layer Transformer-based (Vaswani et al., 2017) encoder to encode the dialogue context. First, an input sequence of tokens is mapped to a sequence of embeddings, which are then passed into the encoder. The encoder consists of a stack of “blocks”, each of which comprises two subcomponents: a self-attention layer followed by a small feed-forward network. Compared to the vanilla transformer encoder, our encoder has slight differences in position embeddings and self-attention layer in fine-tuning phase, which contains richer location information and will be introduced in § 2.7.

2.3 Decoder

Future predicting strategy has been concerned in recent research (Qi et al., 2020; Xiao et al., 2020), instead of predicting only the next token at each time step, the decoder using future predicting predicts n future tokens simultaneously.

Specifically, the original Seq2Seq model aims to optimize the conditional likelihood \( P(r_{<t}, c) \), while future strategy changes the optimization of predicting next single token to \( P(r_{t:t+n-1}|r_{<t}, c) \) at each time step t, where \( r_{t:t+n-1} \) denotes the next continuous n future tokens. The future n-gram prediction loss can explicitly encourage the model to plan for future token prediction and prevent over-fitting on strong local correlations (Qi et al., 2020).

We adopt the n-stream self-attention proposed in ProphetNet (Qi et al., 2020) in our decoder. The
n-stream self-attention mechanism incorporates $n$ extra self-attention predicting streams besides main stream to predict next $n$ continuous future tokens respectively at each time step.

**Memory Scheme** To incorporate the latent variable into decoder, we adopt a memory scheme similar to OPTIMUS (Li et al., 2020), where latent variable $z \in \mathbb{R}^P$ is mapped to a additional memory vector, denoted as $h_{\text{Mem}}$, which is an additional key-value pair for decoder to attend. We have memory vector

$$h_{\text{Mem}} = \begin{bmatrix} z_{\text{key}} \\ z_{\text{value}} \end{bmatrix} = W_M z$$  \hspace{1cm} (1)

where $W_M \in \mathbb{R}^{H \times P}$ is the weight matrix, and the memory vector is shared and propagated across all layers in decoder as:

$$H^{(k+1)} = \text{MultiHead}(H^{(k)}, h_{\text{Mem}}) = H^{(k)} + H^{(k)}$$

The memory vector is equivalent to adding a virtual token during decoding to participate in the calculation of self-attention main stream, and the predicting streams are implicitly affected by $h_{\text{Mem}}$ through interaction with the main stream. The latent variable guides the generation of each step of the decoder through the memory vector.

### 2.4 Latent Variable

Intuitively, introducing latent variables provides a hierarchical generation procedure: 1) sample a latent variable $z$ from the prior network $p(z|c)$; 2) generate $r$ through the decoder network $p(r|c, z)$. From previous research (Zhao et al., 2017a), $z \sim p(z|c)$ may determine the high-level semantics, and the auto-regressive decoding is followed to produce the output sentences with low-level syntactic and lexical details.

Similar to the Variational Autoencoders (VAEs), we learn the parameters $\theta$ by maximizing the marginal log likelihood:

$$\log p_\theta(r|c) = \log \int p_\phi(z|c)p_\theta(r|c, z)dz. \hspace{1cm} (2)$$

where $p_\phi$ involves an intractable marginalization over the latent variable $z$. (Kingma et al., 2016; Li et al., 2020). We will optimize its lower bound, which is equivalent to minimize the two terms below: reconstruction loss (or negative log-likelihood)

$$L_{rc} = -\mathbb{E}_{q(z)}[\log p_\theta(r|c, z)]$$

$$= -\mathbb{E}_{q(z)}[\log \prod_t p_\theta(r_{t:t+n-1}|r_{<t}, c)] \hspace{1cm} (2)$$

and K-L regularization term

$$L_{kl} = KL(q(z)||p_\phi(z|c)). \hspace{1cm} (3)$$

Here $q(z)$ is a multivariable normal distribution with mean $\mu \in \mathbb{R}^P$ and diagonal variance matrix with diagonal taking values $\sigma^2 \in \mathbb{R}^P$, denoted as $\text{diag}(\sigma^2)$. 

---

**Figure 1**: Pre-training and fine-tuning framework of DialogVED, the only difference between pre-training and fine-tuning is that in the fine-tuning stage, we do not mask the source thus the masked spans loss is discarded. It’s worth noting that, to facilitate drawing, we put [CLS] at the end of the context, although we actually put it at the beginning.
To connect to the hidden space, we add a special classification token ([CLS]) to the beginning of the context, and the first hidden state denoted as $h_{[CLS]} \in \mathbb{R}^H$ in last-layer is used to represent the global dialog context. We assume

$$
\begin{bmatrix}
\mu \\
\log(\sigma^2)
\end{bmatrix} = \text{MLP}_b h_{[CLS]}
$$

where $\text{MLP}_b$ is a multilayer perceptron and this multilayer perceptron is called the prior network in VAEs literature. We can then sample $P$ random variables with each variable is from standard normal distribution and via transformation, we obtain samples of $z \in \mathbb{R}^P$ from $\mathcal{N}(\mu, \text{diag}(\sigma^2))$, and feed them to the decoder.

2.5 Mask Language Spans

To improve the understanding ability of the encoder and the robustness to noise, we randomly mask part of the context before encoding. Recent research (Joshi et al., 2020; Lewis et al., 2020) on masked language models show the advantages of masking spans over masking individual words or subword units.

We adopt a simple method to mask spans: 1) randomly select $n$ tokens in context, denote as $S$; 2) for each token $t \in S$, extend it to a text span with a fixed length of $m$; 3) mask all selected tokens after sorting, deduplication and boundary checking.

Following BERT (Devlin et al., 2019), the total number of masked tokens in the context accounts for approximately 15%, and we replace the masked token with: 1) the [MASK] token 80% of the time; 2) a random token 10% of the time; 3) the unchanged masked token 10% of the time. Then, the last-layer hidden states $h_x \in \mathbb{R}^H$ of each masked token $x$ will be used to predict the original token and the encoder is trained to optimize the cross entropy loss:

$$
\mathcal{L}_M = -\sum_x \text{LSM}(W_2 \tanh(W_1 h_x + b_1)) (x)
$$

where $W_1 \in \mathbb{R}^{H \times H}, b_1 \in \mathbb{R}^H$ and $W_2 \in \mathbb{R}^{H \times |V|}$ denote the weight matrices of one fully-connected layer, $|V|$ is the vocabulary size, LSM is log softmax function and $\text{LSM}(\ldots) (x)$ means to take the log probability value corresponding to token $x$. In this paper, we share the parameters of $W_2$ with parameters of embedding layers in the encoder and decoder. Note that we only mask the context only the pre-training stage.

2.6 Reduce KL-vanishing

DialogVED allows the decoder to attend the hidden states of context (i.e., the output of the encoder), and thus direct training will cause the decoder to ignore the latent variable $z$, and the KL loss will rapidly decrease to 0 and the latent space loses its expressive power, which is called posterior collapse or KL-vanishing (Bowman et al., 2016). This paper adopts two methods developed in VAEs literature to reduce posterior collapse:

**Free Bits** (Kingma et al., 2016), which replaces the K-L regularization term in (3) with a hinge loss term that maximize each component of the original K-L term with a constant $\lambda$:

$$
\mathcal{L}_{kl'} = -\sum_i \max(\lambda, KL(q(z_i)||p_\phi(z_i|c)))
$$

**Bag-of-words Loss** (Zhao et al., 2017b), which is used to encourage the latent variable to predict the words in response $r$ in a non-autoregressive way:

$$
\mathcal{L}_{BOW} = -\sum_{t=1}^T \log f_{r_t}
$$

where $T$ is the number of tokens in response $r$, and $f_{r_t}$ denotes the estimated probability of word $r_t$. More specifically, $f$ is the function outputting the probability of words within the target response:

$$
f = \text{softmax}(\text{MLP}_z [z \oplus h_{[CLS]}]) \in \mathbb{R}^{|V|}
$$

where $\text{MLP}_z$ is a multilayer perceptron and $V$ refers to the whole vocabulary.

2.7 Position Embeddings

**Absolute Position Embeddings** Besides token-level learned position embeddings used in original Transformer, we also consider turn level and speaker-level position embeddings like PLATO (Bao et al., 2020). To better model the meaning of a turn in a dialog, We introduce embedding for turn position and role position in one conversation, the final input embedding of each token is the sum of corresponding turn, role and token embeddings.

**Relative Position Embeddings** It has recently become more common to use relative position embeddings, which produce a different learned embedding according to the offset between the “key” and “query” being compared in the self-attention mechanism (Shaw et al., 2018; Raffel et al., 2019). We
extend the element of the original relative distance matrix in T5 (Raffel et al., 2019) to two-tuple.

\[ e_{ij} = \frac{x_i W^Q (x_j W^K + a_{ij}^K) T}{\sqrt{d_z}}, \]

\[ a_{ij}^K = f(d_{token}, d_{turn}, x_i, x_j) \]

In the mapping function \( f \), we consider both token relative distance \( d_{token} \) and turn relative distance \( d_{turn} \), where these tuples are mapped through a bucket function, and then \( a_{ij}^K \) is queried in pre-defined embedding layers.

### 2.8 Pre-training Objectives

Combining the losses detailed in the Equations (2) (5) (6) and (7), we have pre-training objective, which we use to pre-train the DialogVED on the large-scale conversation corpus:

\[ \text{loss} = \mathcal{L}_M + \mathcal{L}_{rc} + \mathcal{L}_l^d + \mathcal{L}_{BOW} \quad (9) \]

To sum up, we mask text spans in the context \( c \), sample a latent variable \( z \) from prior network, and then let the encoder and decoder predict the masked spans and response \( r \) respectively with the guidance of the latent variable \( z \).

### 3 Experiments

In this section, we firstly introduce the pre-training datasets and fine-tuning benchmarks in § 3.1, and implement details in § 3.2. Then we present the main results in § 3.3. Lastly, we analyze the influence of parameters and position embeddings in § 3.4.

### 3.1 DataSets and Baselines

#### 3.1.1 Pre-training Corpus

Large-scale Reddit comments dataset (Zhou et al., 2018; Galley et al., 2019) is employed for pre-training our dialog language model. This dataset has been proved to be helpful in various conversation downstream tasks (Bao et al., 2020; Zhang et al., 2020). We use the script provided by DialogPT (Zhang et al., 2020) to obtain the latest Reddit comment data. We obtain 215 million\(^1\) training samples (42GB in total) for pre-training.

To accelerate the training process and accommodate GPU memory limitations, we adopt two methods. First, we sort the samples according to the length of the context. Samples with similar length (i.e. number of tokens in context) are assembled into a batch to minimize the amount of padding. Secondly, due to the uneven distribution of sample lengths, we divide the Reddit corpus into two sub-datasets: Reddit-Short and Reddit-Long according to the length of context and response, with some statistics in Table 1, and optimize the batch size for each sub-dataset to avoid reserving a large amount of memory for a few long response samples during the training process. Within an epoch, we first pre-train on Reddit-Short with a larger batch size, and then pre-train Reddit-Long with a smaller batch size. We split the reddit comment dataset here mainly for efficiency.

#### 3.1.2 Fine-tuning Benchmarks

Following PLATO (Bao et al., 2020), we select three datasets as our benchmarks: DailyDialog (Li et al., 2017), a chit-chat dataset, which contains high-quality human conversations about daily life.

Persona-Chat (Zhang et al., 2018), a knowledge-grounded conversation dataset. It provides both manually annotated conversations and corresponding persona profiles (background knowledge), where two participants chat naturally and try to get to know each other.

DSTC7-A VSD (Alamri et al., 2019a), a conversational question answering dataset, short for Audio Visual Scene-aware Dialog of the DSTC7 challenge. The system needs to generate an answer given dialogue context and background knowledge. There are multiple reference responses for each context in DSTC7-A VSD test set.

For evaluation, we use the same metrics as used in PLATO, except for knowledge-related metrics, since this paper does not focus on utilizing knowledge. So we will focus the following metrics:

<table>
<thead>
<tr>
<th>Task</th>
<th>Dataset</th>
<th># Examples</th>
<th># Turns</th>
<th># Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-train</td>
<td>Reddit-Short</td>
<td>214M</td>
<td>2.6</td>
<td>28.6/16.0</td>
</tr>
<tr>
<td></td>
<td>Reddit-Long</td>
<td>726K</td>
<td>6.9</td>
<td>137.1/21.2</td>
</tr>
<tr>
<td>Fine-tune</td>
<td>DailyDialog</td>
<td>76K</td>
<td>5.9</td>
<td>75.6/15.0</td>
</tr>
<tr>
<td></td>
<td>Persona-Chat</td>
<td>122K</td>
<td>8.4</td>
<td>95.6/12.2</td>
</tr>
<tr>
<td></td>
<td>DSTC7-A VSD</td>
<td>76K</td>
<td>10.9</td>
<td>102.1/10.7</td>
</tr>
</tbody>
</table>

\( ^1 \)Given an instance containing multiple turns of dialogue \( \{t_1, t_2, \ldots, t_n\} \), we extract \( n-1 \) samples (i.e. context-response pairs), where the context \( c \) is \( \{t_1, t_2, \ldots, t_{i-1}\} \), and the response \( r \) is \( \{t_i\} \), for \( i = \{2, 3, \ldots, n\} \).
3.1.3 Baselines

Vanilla sequence to sequence (Seq2Seq) models, dialog pre-training models, and general natural language pre-training models are used as our baselines: **Seq2Seq** (Vinyals and Le, 2015) is a sequence-to-sequence model with attention. **iVAE_MI** (Fang et al., 2019) is an implicit deep latent variable model based on Variational Autoencoder for better latent representations and diverse responses. **LIC** (Golovanov et al., 2019) obtains the best performance during the contest, and is one transformer based generation method. **PLATO** (Bao et al., 2020) utilizes a discrete latent variable for dialog generation pre-training to address the one-to-many problem. **ProphetNet** (Qi et al., 2020) is a pre-trained LM model with predicting more than one future tokens as the pre-training objective. We fine-tune ProphetNet-Large model released in (Qi et al., 2020) with downstream training data directly.

For benchmark DSTC7-A VSD, we include **AVSD Baseline** (Alamri et al., 2019a) system provided by the the challenge organizer, as well as the best performing model developed by the team of CMU Sinbad’s (Sanabria et al., 2019).

### Table 2: Experimental results on DailyDialog and PersonaChat with automatic evaluations, with highest value written in bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>DailyDialog</th>
<th></th>
<th></th>
<th></th>
<th>PersonaChat</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU-1</td>
<td>BLEU-2</td>
<td>Distinct-1</td>
<td>Distinct-2</td>
<td>BLEU-1</td>
<td>BLEU-2</td>
<td>Distinct-1</td>
</tr>
<tr>
<td>Seq2Seq (Vinyals and Le, 2015)</td>
<td>0.336</td>
<td>0.238</td>
<td>0.030</td>
<td>0.128</td>
<td>0.448</td>
<td>0.353</td>
<td>0.004</td>
</tr>
<tr>
<td>iVAE_MI (Fang et al., 2019)</td>
<td>0.309</td>
<td>0.249</td>
<td>0.029</td>
<td>0.250</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LIC (Golovanov et al., 2019)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.405</td>
<td>0.320</td>
<td>0.019</td>
</tr>
<tr>
<td>PLATO w/o latent (Bao et al., 2020)</td>
<td>0.405</td>
<td>0.322</td>
<td>0.046</td>
<td>0.246</td>
<td>0.458</td>
<td>0.357</td>
<td>0.012</td>
</tr>
<tr>
<td>PLATO (Bao et al., 2020)</td>
<td>0.397</td>
<td>0.311</td>
<td>0.054</td>
<td>0.291</td>
<td>0.406</td>
<td>0.315</td>
<td>0.021</td>
</tr>
<tr>
<td>ProphetNet (Qi et al., 2020)</td>
<td>0.443</td>
<td>0.392</td>
<td>0.039</td>
<td>0.211</td>
<td>0.466</td>
<td>0.391</td>
<td>0.013</td>
</tr>
<tr>
<td>DialogVED w/o latent</td>
<td>0.461</td>
<td>0.407</td>
<td>0.041</td>
<td>0.222</td>
<td>0.459</td>
<td>0.380</td>
<td>0.010</td>
</tr>
<tr>
<td>DialogVED - Greedy</td>
<td>0.459</td>
<td>0.410</td>
<td>0.045</td>
<td>0.265</td>
<td>0.470</td>
<td>0.387</td>
<td>0.016</td>
</tr>
<tr>
<td>DialogVED - Sampling</td>
<td>0.431</td>
<td>0.370</td>
<td>0.058</td>
<td>0.372</td>
<td>0.428</td>
<td>0.357</td>
<td>0.032</td>
</tr>
<tr>
<td>DialogVED</td>
<td><strong>0.481</strong></td>
<td><strong>0.421</strong></td>
<td>0.042</td>
<td>0.232</td>
<td><strong>0.482</strong></td>
<td><strong>0.399</strong></td>
<td>0.015</td>
</tr>
</tbody>
</table>

**BLEU-1/2** (Papineni et al., 2002), which measures the relevance of generated text to the reference text by calculating the 1/2-gram overlapping between them.

**Distinct-1/2** (Li et al., 2016a), which measures the diversity of a generated sentence by focusing on the number of distinct 1/2-gram of a sentence and thus penalizing sentences with lots of repeated words.

Other word-overlap-based metrics, METEOR, ROUGE-L, and CIDEr, which are also reported for the DSTC7-A VSD dataset, same as DSTC7 reviews (Alamri et al., 2019b).

### 3.2 Model Configuration

DialogVED is composed of a 12-layer encoder and a 12-layer decoder, with 1024 embedding/hidden size and 4096 feed-forward filter size. The dimension $p$ of hidden states $z$ is set to 64 and we will analyze the effect of $P$ in § 3.4.1. We use Adam optimizer (Kingma and Ba, 2014) with a learning rate of $3 \times 10^{-4}$ for pre-training. We set ngram as 2 following ProphetNet (Qi et al., 2020). The pre-training of dialogue generation is carried out on 32 Nvidia Tesla V100 32G GPU (4 nodes) for 6 epochs, taking about 5 days to reach convergence. Mixed precision training is also adopted for efficiently training and inference, and we use the Fairseq (Ott et al., 2019) framework to conduct all experiments. We use the BERT-uncased dictionary, and replace some unused tokens to custom special symbols (such as [SOT], denoting the beginning of the conversation, which is suitable for conversation datasets containing knowledge, like PersonaChat and DSTC7-A VSD). We used package WordPiece (Devlin et al., 2019) for tokenization.

For fine-tuning, we use exactly the same hyperparameter settings in all three datasets, and they are slightly different from the hyperparameter in pre-training. The learning rate is set to $1 \times 10^{-4}$ and the batch size is fixed to 512. We also adopt an additional warmup strategy where we linearly increase the learning rate from initial learning rate ($1 \times 10^{-5}$), the number of warmup updates is set to 2000. For each dataset, we train 10 epochs, and select the checkpoint with the lowest validation loss for inference.

### 3.3 Main Results

In Table 2, we compare several DialogVED variants with baseline models. DialogVED represents inferencing DialogVED with beam search. Command
pared with DialogVED, DialogVED w/o latent is not equipped with latent variable, thus the loss function does not include bag-of-words loss and K-L loss. DialogVED Greedy means DialogVED inference with greedy search. For DialogVED Sampling, we sample from the top K tokens with the highest output probability at each decoding step. For the latent space, we always sample each latent variable from the prior distribution standard normal distribution. Here, beam size is set to 5 and K is set to 100.

As shown in Table 2 and Table 6 (in Appendix A), our model DialogVED is very competitive compared to PLATO and other models. In particular, decoding using Top-K (K = 100) sampling with DialogVED beats the PLATO in BLEU-1/2 and Distinct-1/2 on DailyDialog and PersonaChat (see in Table 2). In fact, as K increases, the overlap of n-grams decreases and the diversity increases. Based on our observations, K taking 100 is a good balance, Table 3 shows more detailed results.

There are 2 essential components that contribute greatly the success of our model: Firstly, We adopt a newly developed pretrained LM as the initializer and further continue its pretraining pipeline on our dialog dataset (Reddit) and thus we have a really powerful encoder-decoder. This is demonstrated in the fact that our model (DialogVED w/o latent variable) beat PLATO (w/o latent variable) in all metrics on all the three datasets.

Secondly, the special structure of our model combines the benefits of both seq2seq models and VAE models. Compared to general VAEs, DialogVED allows encoder-decoder interaction in the decoding, which avoids insufficient representation of low-dimensional latent variable. At the same time, compared with seq2seq model, predicting the bag of words pushes the latent variable to give extra guidance to decoder. This is demonstrated by the fact that when compared with DialogVED w/o latent variable, we observe the additional gains in terms of both accuracy and diversity (see Table 2).

Overall, our DialogVED achieves new state-of-the-art results in all three downstream tasks of dialogue response generation.

3.4 Parameters and Position Analysis
3.4.1 Balancing Accuracy and Diversity with Sampling
We investigate the effect of latent space sizes, $P$, defined as the dimension of the latent variable $z$ and the different $K$ in sampling.

The results in Table 3 show that smaller latent size ($P = 32$) is more dominant in n-gram based metrics (BLEU-1/2), while larger latent size generates more diverse texts. From the results of top-$K$ sampling, we see that the two metric (BLEU-1/2 and Distinct-1/2) have a negative correlation.

We can flexibly choose the decoding strategy depends on specific scene.

<table>
<thead>
<tr>
<th>$P$</th>
<th>Top-$K$</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>Distinct-1</th>
<th>Distinct-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>5</td>
<td>0.448</td>
<td>0.385</td>
<td>0.042</td>
<td>0.289</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.443</td>
<td>0.376</td>
<td>0.045</td>
<td>0.317</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.442</td>
<td>0.375</td>
<td>0.047</td>
<td>0.332</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.439</td>
<td>0.374</td>
<td>0.051</td>
<td>0.347</td>
</tr>
<tr>
<td>64</td>
<td>5</td>
<td>0.442</td>
<td>0.383</td>
<td>0.046</td>
<td>0.308</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.437</td>
<td>0.374</td>
<td>0.050</td>
<td>0.340</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.434</td>
<td>0.371</td>
<td>0.054</td>
<td>0.364</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.431</td>
<td>0.370</td>
<td><strong>0.058</strong></td>
<td><strong>0.372</strong></td>
</tr>
</tbody>
</table>

Table 3: The results of different latent size and Top $K$ on DailyDialog dataset. $P$ is the dimension of the latent variable.

3.4.2 Position Embeddings
We study the impact of position embeddings as described in section 2.7, we define two types of position embeddings: absolute position embeddings (APE) and relative position embeddings (RPE).

We report the metrics of their different combinations, these independent components are TurnAPE (turn absolute embedding), RoleAPE (role absolute embedding), TokenRPE (token relative embedding) and TurnRPE (turn relative embedding) respectively.

As the results shown in Table 4, the combination of TurnAPE and RoleAPE achieve the best performance. Both absolute and relative position embeddings improve model performance, nevertheless, including them at the same time can be harmful.

<table>
<thead>
<tr>
<th>TurnAPE</th>
<th>RoleAPE</th>
<th>TokenRPE</th>
<th>TurnRPE</th>
<th>BLEU-1/2</th>
<th>Distinct-1/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>0.481/0.421</td>
<td>0.042/0.232</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>0.491/0.429</td>
<td>0.041/0.229</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>0.483/0.422</td>
<td>0.042/0.230</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td><strong>0.494/0.435</strong></td>
<td>0.042/0.232</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>0.485/0.424</td>
<td>0.039/0.216</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td><strong>0.480/0.422</strong></td>
<td>0.042/0.231</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>0.487/0.425</td>
<td>0.040/0.230</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.483/0.435</td>
<td>0.039/0.228</td>
</tr>
</tbody>
</table>

Table 4: Effect of position embeddings on model performance.
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640

<table>
<thead>
<tr>
<th></th>
<th>Group 1</th>
<th>Group 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Win</td>
<td>Lose</td>
</tr>
<tr>
<td>Fluency</td>
<td>0.16</td>
<td>0.11</td>
</tr>
<tr>
<td>Coherence</td>
<td>0.38</td>
<td>0.16</td>
</tr>
<tr>
<td>Informativeness</td>
<td>0.13</td>
<td>0.11</td>
</tr>
<tr>
<td>Overall</td>
<td>0.26</td>
<td>0.14</td>
</tr>
</tbody>
</table>

Table 5: Human evaluation of DialogVED vs. PLATO (Group 1) and DialogVED-sampling vs. PLATO (Group 2).

3.5 Human Evaluation
Automated metrics (BLEU 1/2, Distinct-1/2, etc.) have limitations for evaluating open-domain dialog tasks. To make it more convincing, we conduct a human evaluation. Specifically, we randomly select 100 dialogue contexts and generate responses with the following methods: PLATO, DialogVED and DialogVED-Sampling. Following PLATO, annotators are asked to compare the response (win, tie or lose) quality from four aspects: fluency, coherence, informativeness and overall.

The results of human comparison are shown in Table 5, where the average Cohen’s kappa (Kraemer, 2014) of group 1 and 2 is 0.729 and 0.743 respectively, indicating annotators have reached moderate agreement. It can be seen that most of the time they are tied, and the three models sometimes generate exactly the same response. For DialogVED, it beats Plato more in coherence but with close informativeness; while DialogVED-sampling beats Plato significantly in informativeness but with a slightly weaker coherence.

In general, DialogVED can generate both relevant and diverse response, we show some case study to help illustrate the effectiveness of our model in Appendix B.

4 Related Work

Encoder-Decoder dialog models: Encoder-decoder models are widely used in dialogue response generation, but it tends to generate generic responses and dull responses (e.g., I don’t know). To enhance encoder-decode models and generate diverse responses, researchers have tried different approaches: using diversity promotion objectives (Li et al., 2016a), using different decoding algorithms (Li et al., 2016b), adding additional contents (Xu et al., 2019), or introducing large-scale knowledge graphs into dialog generation (Liu et al., 2018; Wu et al., 2020).

Another class of methods is using the latent variable to address the one-to-many problem in response generation. These models introduce discourse-level diversity and are able to generate diverse dialog responses (Serban et al., 2017; Zhao et al., 2017a, 2018; Gao et al., 2019). In this paper, we also adopt this approach and further we incorporate the latent variables both in the pre-training and fine-tuning.

Pre-trained Dialog Models: Pre-trained language models have been successfully used in NLG and NLU tasks (Devlin et al., 2019; Radford et al., 2019). Recently, various new pre-trained language models have been pre-trained including BART (Lewis et al., 2020), ProphetNet (Qi et al., 2020), T5 (Raffel et al., 2020). In these papers, they demonstrate that better performance can be obtained with fine-tuning PLMs than training from scratch.

Due to the fact that there are many important applications in the dialog domain and the dialog corpus has different linguistic features from general documents, pre-trained dialog models with open domain dialog data such as Reddit is very important. DialoGPT (Zhang et al., 2020) continues to pre-train GPT-2 model directly on Reddit comments data, and the new pre-trained model achieves better performance on downstream tasks including several dialog response generation benchmarks.

PLATO (Bao et al., 2020) proposes a new model specifically for dialog generation, which introduces a discrete variable for one-to-many relationship modeling. The pre-trained model helps to achieve state-of-the-art results on several response generation tasks. This is the closest work in literature to ours. However, in our paper, we introduce continuous latent variables during pre-training on dialog corpus instead of a discrete latent variable.

5 Conclusion
This paper proposes a new pre-training framework for dialogue response generation called DialogVED. The latent variable is incorporated into the sequence-to-sequence framework based on Transformer, and obtains a robust and diverse response generation model through 4 training targets. Our pre-trained model has achieved new state-of-the-art in multiple downstream tasks of dialogue response generation. Extensive experiments prove the effectiveness of our model. Additional human evaluation demonstrates the advantages of our proposed model.
**Ethical Statement**

In this paper, different ethical restrictions deserve discussion.

All data used in our pre-training are available online and other dialog corpus in this paper are publicly available sources. We strictly followed the platform's policies and rules when crawling data from web platforms. We did not employ any author-specific information in our research.

Our corpus may includes some bias, such as political bias and social bias, and our model might have inherited some forms of these bias. In order to limit these bias as much as possible, we filter controversial articles and removed data with offensive information when possible.

**References**


Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In ACL (1).


Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi.
2017a. Learning discourse-level diversity for neural
dialog models using conditional variational autoen-
coders. In *Proceedings of the 55th Annual Meeting of
the Association for Computational Linguistics (Vol-
ume 1: Long Papers)*, pages 654–664, Vancouver,
Canada. Association for Computational Linguistics.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao,
Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense
knowledge aware conversation generation with graph

A Results on DSTC7-A VSD

On the DSTC7-A VSD, the diversity of the re-
sponses is not as important as the accuracy. From
Table 6, We observe that DialogVED w/o latent
variable perform the best in overall metrics. How-
ever, DialogVED equipped with beam search or
greedy search, can still easily beat PLATO even
though it has a post-generation ranking component.

B Case Study

We demonstrate the responses generated from our
model as well as other baseline models in Table 7,
8 and 9, respectively. The results in Table 8 and 9
show that our model accurately outputs the knowl-
edge information contained in context although
we do not model knowledge explicitly. Compared
with beam search or greedy decoding, decoding
with top-K sampling not only generates bolder and
more diverse response, but also can maintain good
relevance, as showed in Table 7 and 8.
<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
<th>BLEU-4</th>
<th>METEOR</th>
<th>ROUGH-L</th>
<th>CIDEr</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVSD Baseline (Alamri et al., 2019a)</td>
<td>0.629</td>
<td>0.485</td>
<td>0.383</td>
<td>0.309</td>
<td>0.215</td>
<td>0.487</td>
<td>0.746</td>
</tr>
<tr>
<td>CMU Sinbad’s (Sanabria et al., 2019)</td>
<td>0.718</td>
<td>0.584</td>
<td>0.478</td>
<td>0.394</td>
<td>0.267</td>
<td>0.563</td>
<td>1.094</td>
</tr>
<tr>
<td>PLATO (Bao et al., 2020)</td>
<td>0.784</td>
<td>0.637</td>
<td>0.525</td>
<td>0.435</td>
<td>0.286</td>
<td>0.596</td>
<td>1.209</td>
</tr>
<tr>
<td>ProphetNet (Qi et al., 2020)</td>
<td>0.824</td>
<td>0.691</td>
<td>0.582</td>
<td>0.487</td>
<td>0.313</td>
<td>0.635</td>
<td>1.382</td>
</tr>
<tr>
<td>DialogVED w/o latent</td>
<td>0.832</td>
<td>0.705</td>
<td>0.598</td>
<td>0.506</td>
<td>0.314</td>
<td>0.638</td>
<td>1.386</td>
</tr>
<tr>
<td>DialogVED - Greedy</td>
<td>0.817</td>
<td>0.685</td>
<td>0.575</td>
<td>0.481</td>
<td>0.306</td>
<td>0.629</td>
<td>1.356</td>
</tr>
<tr>
<td>DialogVED</td>
<td>0.822</td>
<td>0.692</td>
<td>0.582</td>
<td>0.489</td>
<td>0.312</td>
<td>0.636</td>
<td>1.391</td>
</tr>
</tbody>
</table>

Table 6: Experimental results on DSTC7-A VSD with automatic evaluations, with highest value written in bold.

**Table 7: Case analysis of response generation on DailyDialog**

<table>
<thead>
<tr>
<th>Persona Profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>i love trying barbeque restaurants .</td>
</tr>
<tr>
<td>i am an administrative assistant .</td>
</tr>
<tr>
<td>i am looking forward to retiring in five years .</td>
</tr>
<tr>
<td>i like ot play racquetball .</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Context</th>
<th>[P2] have you seen the new cosmo magazine that i was reading this morning ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[P1] i think i may have seen it on the kitchen table .</td>
<td></td>
</tr>
<tr>
<td>[P2] you’re right , thanks . there’s an interesting interview with a top director about his latest film that i want to finish .</td>
<td></td>
</tr>
<tr>
<td>[P1] who’s the director ?</td>
<td></td>
</tr>
</tbody>
</table>

**GOLD**

it’s martin scorsese . you liked his film , the departed , didn ’ t you ?

<table>
<thead>
<tr>
<th>Model</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq2Seq</td>
<td>yes , i did . i found it very disappointing .</td>
</tr>
<tr>
<td>iVAE _3_1</td>
<td>the cake and is very popular .</td>
</tr>
<tr>
<td>PLATO</td>
<td>i’m not sure , but i think he might be the director of gone with the wind .</td>
</tr>
<tr>
<td>DialogVED</td>
<td>it’ s the director of a foreign film called ‘ gone with the wind ’ .</td>
</tr>
<tr>
<td>DialogVED-Greedy</td>
<td>it’ s directed by a korean director , named zhuang lingo .</td>
</tr>
<tr>
<td>DialogVED-Sampling</td>
<td>it’s directed by a korean director , named zhuang lingo .</td>
</tr>
</tbody>
</table>

**Table 8: Case analysis of response generation on PersonaChat**

<table>
<thead>
<tr>
<th>Persona Profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>i love trying barbeque restaurants .</td>
</tr>
<tr>
<td>i am an administrative assistant .</td>
</tr>
<tr>
<td>i am looking forward to retiring in five years .</td>
</tr>
<tr>
<td>i like ot play racquetball .</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Context</th>
<th>[P2] hi , do you like bbq ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[P1] hello yes i love bbq</td>
<td></td>
</tr>
<tr>
<td>[P2] i love restaurants with bbq , they are a lot of fun .</td>
<td></td>
</tr>
<tr>
<td>[P1] my poor toes are so webbed</td>
<td></td>
</tr>
<tr>
<td>[P2] really ? afterwards , i go and play racquetball to loose some calories .</td>
<td></td>
</tr>
<tr>
<td>[P1] cool i like to paint</td>
<td></td>
</tr>
<tr>
<td>[P2] what do you do as a profession ? i work as administrative assistant</td>
<td></td>
</tr>
<tr>
<td>[P1] i dig in people trash for things to build</td>
<td></td>
</tr>
<tr>
<td>[P2] sounds unique but that is more of a hobby instead of a profession</td>
<td></td>
</tr>
<tr>
<td>[P1] true . i paint for a living</td>
<td></td>
</tr>
</tbody>
</table>

**GOLD**
good thing i am going to retire in 5 years , no more working for this guy !

<table>
<thead>
<tr>
<th>Model</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq2Seq</td>
<td>that sounds like a lot of work . do you have a job ?</td>
</tr>
<tr>
<td>LIC</td>
<td>yeah it is a lot of fun</td>
</tr>
<tr>
<td>PLATO</td>
<td>that is cool . i am looking forward to retiring in 5 years</td>
</tr>
<tr>
<td>DialogVED</td>
<td>i am looking forward to retiring in five years</td>
</tr>
<tr>
<td>DialogVED-Greedy</td>
<td>i am looking forward to retiring in five years</td>
</tr>
<tr>
<td>DialogVED-Sampling</td>
<td>i am looking forward to retiring in five years</td>
</tr>
</tbody>
</table>

**Table 8: Case analysis of response generation on PersonaChat**
<table>
<thead>
<tr>
<th>Video Caption</th>
<th>a man closes his window, then he sneezes twice before taking a drink, then he opens up a bag and digs through it looking for something before walking out of the room.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Summary</td>
<td>a man closes the window, goes to the table and goes through the items in a bag, takes a drink from the green cup and leaves the room.</td>
</tr>
</tbody>
</table>
| Context | [P1] what is the guy doing at the window?  
[P2] the guy is closing the window  
[P1] what does he do after that? |
| GOLD | stands at the table and takes things out of bag |
| **Model** | **Response** |
| Baseline | he picks up a book from the table |
| PLATO | he goes to the table and takes a drink from a green cup |
| DialogVED | he goes to the table and goes through the items in a bag before taking a drink |
| DialogVED-Greedy | he sneezes twice before taking a drink |

Table 9: Case analysis of response generation on DSTC7-AVSD