

Generating Multiple Diverse Responses for Short-Text Conversation

Jun Gao,^{1*} Wei Bi,^{2†} Xiaojiang Liu,² Junhui Li,¹ Shuming Shi²

¹School of Computer Science and Technology, Soochow University, Suzhou, China
imgaojun@gmail.com, lijunhui@suda.edu.cn

²Tencent AI Lab, Shenzhen, China
{victoriabi, kieranliu, shumingshi}@tencent.com

Abstract

Neural generative models have become popular and achieved promising performance on short-text conversation tasks. They are generally trained to build a 1-to-1 mapping from the input post to its output response. However, a given post is often associated with multiple replies simultaneously in real applications. Previous research on this task mainly focuses on improving the relevance and informativeness of the top one generated response for each post. Very few works study generating multiple accurate and diverse responses for the same post. In this paper, we propose a novel response generation model, which considers a set of responses jointly and generates multiple diverse responses simultaneously. A reinforcement learning algorithm is designed to solve our model. Experiments on two short-text conversation tasks validate that the multiple responses generated by our model obtain higher quality and larger diversity compared with various state-of-the-art generative models.

Introduction

Endowing the machine with the ability to converse with humans using natural language is one of the fundamental challenges in artificial intelligence (Turing 1950). In particular, conversation models in open domains have received increasing attention due to its wide applications including chatbots, virtual personal assistants and etc. With the vast amount of conversation data available on web services, generative models have shown their great potential. Especially, the Sequence-to-sequence (Seq2seq) framework (Vinyals and Le 2015; Sordoni et al. 2015), which learns the mapping from the input to its corresponding output directly, has achieved promising performance. Following the most conventional setting for generative short-text conversation models (Shang, Lu, and Li 2015; Li, Monroe, and Jurafsky 2016; Mo et al. 2016; Shen et al. 2017), we consider the single round chat conversation with no context information, i.e. an input post from a user and the output response given by the machine.

Despite the popularity of the Seq2seq models, various problems occur when they are applied for short-text conver-

sation. One important issue is that they try to model the input post and the output response as a 1-to-1 mapping. Whereas, a post is often associated with multiple responses simultaneously in real applications, which is a 1-to-n mapping. For example, for a user’s tweet on Twitter, there are often multiple replies obtained from the other users.

In order to handle a set of replies, a simple method is to pair each of them with the input as multiple training instances and then train the Seq2seq models. Previous results show that this method tends to generate generic responses (Li et al. 2016a; 2016b). Moreover, only one target response can be generated in this method. To obtain multiple responses as in the real applications, many approaches resort to beam search or enhanced beam search for diverse answers (Li, Monroe, and Jurafsky 2016; Shao et al. 2017). Recently, Zhou *et al.* (2017; 2018) introduced multiple latent embeddings as language responding mechanisms into the Seq2seq framework. By choosing different responding mechanisms, multiple output responses can be generated. Zhao *et al.* (2017) adopted the conditional variational auto-encoder (CVAE) and generated multiple responses by drawing samples from the prior distribution. Yet, these methods still decomposed multiple responses into independent ones, and did not model the fact of multiple responses.

In this paper, we propose a novel response generation model for short-text conversation, which considers multiple responses jointly. We introduce a latent variable distribution for each post and sample different values from the learned latent distribution to generate multiple diverse responses. Our novelty lies in two aspects. First, we model the post with its various responses as a bag of instances and infer the distribution of the latent variable by considering multiple responses jointly. Second, unlike in CVAE, we adopt interpretable latent variables, of which different samplings should be capable to capture the divergence between different responses. Here, we set the latent variable as a word in the vocabulary. Each sampling of the latent variable is then to select a specific word, which drives the generated response to be more meaningful around this word.

Since open-domain conversations cover a wide range of topics and areas, we require a large vocabulary to adequately include the possible latent words in all responses. Thus, it is impractical to compute the exact gradient from the fully factorized probabilities over all latent variables and apply the

*Work done when Jun Gao was interning at Tencent AI Lab.

†Corresponding author

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

standard back-propagation training procedure. We propose to treat each latent word as an action, and use a reinforcement learning(RL) algorithm to solve the large latent space issue. We first use the latent word inference network to estimate the distribution over all words for an input post. Based on this probability distribution, we sample multiple latent words, which are then combined as parts of inputs in the generation network. The generation network will return the reward of each sampled latent words by considering all generated responses jointly. Specifically, we design a sampling scheme to select multiple diverse latent words for our RL algorithm to converge effectively.

To summarize, our contributions are threefold:

1. We propose a response generation model which directly considers an input post with its multiple responses jointly. Our model could automatically generate multiple diverse responses for a single post.
2. We consider latent words as actions and propose to use a reinforcement learning algorithm to solve our model.
3. Experimental results show that our model outperforms existing state-of-the art generative methods in generating multiple high-quality and diverse responses. All our code and datasets are available are <https://ai.tencent.com/ailab/nlp/dialogue.html>.

Related Work

The Seq2seq framework has been widely used for conversational response generation (Vinyals and Le 2015; Sordoni et al. 2015; Shang, Lu, and Li 2015). Such models learn the mapping from an input \mathbf{x} to one output \mathbf{y} by maximizing the pairwise probability of $p(\mathbf{y}|\mathbf{x})$. During testing, these models only target for one response. In order to obtain multiple responses, beam search can be used. However, the resulting multiple sequences are often very similar. Many approaches have been proposed to re-rank diverse meaningful answers into higher positions. For example, Li *et al.* (2016) proposed a simple fast decoding algorithm to directly encourage response diversity in the scoring function used in beam search. Shao *et al.* (2017) heuristically re-ranked the responses segment by segment to inject diversity earlier in the decoding process. These methods only modified the decoding steps and still often generated responses using different words but with similar semantics.

A few works have explored different factors that decide the generation of diverse responses. For task-oriented dialogue systems, Wen *et al.* (2017) proposed to model a latent space to represent intentions. Zhao *et al.* (2017) adopted the CVAE for learning discourse-level diversity for dialog models. They further incorporated a distribution over potential dialogue acts for better discourse-level diversity. Unlike them, our work targets for the single round open-domain conversation task, which assumes no discourse level information. Also, it is observed that in CVAE with a fixed Gaussian prior, the learned conditional posteriors tend to collapse to a single mode, yielding little diversity in the generated results (Wang, Schwing, and Lazebnik 2017). In our model, if we choose two latent words with far different meanings, the generated responses should be different.

The most relevant work to ours is from Zhou *et al.* (2017; 2018). They introduced latent embeddings as the language responding mechanisms into the Seq2seq model. By choosing different responding mechanisms, multiple responses can be generated. However, their model decomposed the joint probability of multiple responses into a product of individual probabilities: $p(\{\mathbf{y}\}|\mathbf{x}) = \prod_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$. Despite that the latent embedding has no interpretable meaning, they have to set a very small number of mechanisms in order to optimize the fully factorized probabilities over the latent embeddings in a standard back-propagation training procedure. Our model assumes a much larger latent space with interpretability. Thus, it is impractical to compute the exact gradient, which we will show in later section. We then propose a reinforcement learning algorithm to train our model.

We notice that Yao *et al.* (2017) proposed to generate an informative reply based on a cue word. In our model, we utilize a latent word to guide the response generation, which shares certain similarities. Their results also validates our model assumption that it should be effective to integrate auxiliary word-level information for generating high-quality responses. However, they simply computed the cue word for each post using the point-wise mutual information (PMI) from the word concurrence statistics, and focused on designing a better decoding cell to deal with the cue word. Our work considers another problem, which is to infer multiple diverse latent words from a parametric model to generate multiple responses with satisfying quality and diversity.

Most of the related methods mentioned above will be compared in our experiments and our results show that our proposed method outperforms them in terms of generating multiple relevant and diverse responses.

Models

Problem Formulation and Model Overview

We are given training samples $\{(\mathbf{x}, \{\mathbf{y}\})\}$, with \mathbf{x} denoting the input post and $\{\mathbf{y}\}$ its set of output responses. We assume an unobserved latent variable z for each input \mathbf{x} . Before introducing our model, we first discuss how to construct a proper latent space \mathbf{Z} in short-text conversation tasks:

- For any $z \in \mathbf{Z}$, it should be interpretable, which can show its relevance to the input \mathbf{x} as well as the responses $\{\mathbf{y}\}$. Thus, we are easy to identify whether a good unobservable z is assigned for a sample $(\mathbf{x}, \{\mathbf{y}\})$.
- Each $z \in \mathbf{Z}$ should be capable of capturing the sentence-level discrepancy such that given distinct \mathbf{x} 's, the inferred z 's can be different and given distinct z 's, the generated responses can be different.

To meet these two criteria, we set the latent space \mathbf{Z} as the vocabulary used in each task and each z corresponds to one of its words. In terms of the first criterion, we can directly check the relevance from the plain text of z , \mathbf{x} and $\{\mathbf{y}\}$. For the second one, the vocabulary defines a large enough latent space for a given \mathbf{x} to select out multiple distinct and diverse z 's to generate multiple responses.

We observe that it is often the case that not all responses relate to one single latent word z . For example, with a query

“I feel boring in the weekends. How do you spend your time?”, it has two responses “I play guitar sometimes” and “My favorite is camping and you can join me next time.”. Both “guitar” and “camping” are valid latent words of the query, but each of them is relevant to only one of the responses. Thus, we get inspired from multi-instance learning (Zhou 2004), in which a positive bag of instances contains at least one positive instance. Here, a query with its multiple responses is considered as a bag of instances. That is, if a given \mathbf{x} is assigned with a latent word z , its bag of instances contains at least one response \mathbf{y} relevant to this z . In turn, if multiple responses are relevant to different z 's, multiple z 's can be inferred with relatively large probabilities simultaneously for a single \mathbf{x} . Our training objective is to minimize the average risk of generating the bag of responses $\{\mathbf{y}\}$ over the probability distribution of z given \mathbf{x} , which is defined as:

$$J(\Theta) = \mathcal{L}(\{\mathbf{y}\}|\mathbf{x}) = \mathbb{E}_{p(z|\mathbf{x})}[\mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z)], \quad (1)$$

where $p(z|\mathbf{x})$ is the probability of a latent z given \mathbf{x} , and $\mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z)$ is the loss of generating the bag of responses $\{\mathbf{y}\}$ given both \mathbf{x} and z . In our model, we use deep neural networks to parameterize $p(z|\mathbf{x})$ and $\mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z)$, denoted as \mathbf{W} and \mathbf{G} respectively. $\Theta = \{\mathbf{W}, \mathbf{G}\}$ contains all model parameters. Note that unlike previous works (Zhou et al. 2017; 2018), we do not decompose $\{\mathbf{y}\}$ into individual \mathbf{y} 's and infer the latent z for each of them separately.

The overall framework of our model is shown in Fig. 1. It consists of two key components: a latent word inference network for $p(z|\mathbf{x})$, from which multiple words can be sampled out to drive the generation of diverse responses; a response generation network for $\mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z)$, which generates the given multiple responses according to the input \mathbf{x} and sampled words. We iteratively update the two networks in that: the latent word inference network starts with an initial policy, i.e. $p(z|\mathbf{x})$, revise its strategy based on the reward and improve the accuracy of the inferred latent words; the generation network receives more accurate latent words as inputs to facilitate generating the given responses, and output the reward for each sampled z for updating the latent word inference network. In the following subsections, we describe these two networks, the RL algorithm for their joint training and prediction in detail.

Latent Word Inference Network

This network estimates the probability distribution $p(z|\mathbf{x})$. We first encode the input \mathbf{x} with a bidirectional GRU to obtain an input representation $\mathbf{h}_\mathbf{x}$ and then compute the probability of each latent word as follows:

$$p(z|\mathbf{x}) = \text{softmax}(\mathbf{W}_2 \cdot \tanh(\mathbf{W}_1 \mathbf{h}_\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2), \quad (2)$$

where parameters in the bidirectional GRU, \mathbf{W}_1 , \mathbf{b}_1 , \mathbf{W}_2 and \mathbf{b}_2 are jointly trained in our model.

Generation Network

We assume that $\mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z)$ can be estimated from individual loss $\ell(\mathbf{y}|\mathbf{x}, z)$'s for $\mathbf{y} \in \{\mathbf{y}\}$ through a simple parameter-free and differentiable function $f(\cdot)$:

$$\mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z) = f_{\mathbf{y} \in \{\mathbf{y}\}}(\ell(\mathbf{y}|\mathbf{x}, z)). \quad (3)$$

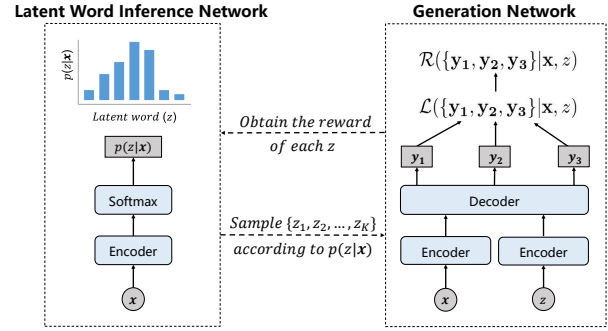


Figure 1: An overview of the proposed model. The given \mathbf{x} is associated with three output responses $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$. Each latent z is a word in the vocabulary.

We first introduce the detailed network structure to estimate $\ell(\mathbf{y}|\mathbf{x}, z)$, and then discuss proper choices of $f(\cdot)$.

With a sampled z , we first encode \mathbf{x} through another bidirectional GRU for an input representation $\mathbf{h}_\mathbf{x}^g$. At the same time, we obtain the representation \mathbf{h}_z^g of z using an embedding layer followed by a fully connected layer. Both representations are then leveraged to decode the output sequence \mathbf{y} . Here, our decoding function is constructed as follows. We compute the attention between \mathbf{h}_z^g and \mathbf{x} and construct the context vector \mathbf{c}_z :

$$a_j = \frac{\exp(\mathbf{h}_z^{gT} \mathbf{h}_\mathbf{x}^g(j))}{\sum_t \exp(\mathbf{h}_z^{gT} \mathbf{h}_\mathbf{x}^g(t))}, \quad \mathbf{c}_z = \sum_t a_t \mathbf{h}_\mathbf{x}^g(t), \quad (4)$$

where $\{\mathbf{h}_\mathbf{x}^g(t)\}$ is the series of hidden vectors obtained in the encoder for each word in \mathbf{x} . Intuitively, \mathbf{c}_z is a weighted sum of the input representations, which captures the information of \mathbf{x} relevant to the selected z .

We use GRU as the decoding cell with $\{\mathbf{h}_\mathbf{y}(t)\}$ denoting its series of hidden vectors. In each decoding step, we calculate the attention and the context vector $\mathbf{c}_\mathbf{y}(t)$ for $\mathbf{h}_\mathbf{y}(t)$, which can be obtained by replacing \mathbf{h}_z^g with $\mathbf{h}_\mathbf{y}(t)$ in Eq. 4. Then, we concatenate $\mathbf{h}_\mathbf{y}(t)$, \mathbf{c}_z and $\mathbf{c}_\mathbf{y}(t)$ together and forward this vector into a fully connected layer followed by a softmax output layer. $\ell(\mathbf{y}|\mathbf{x}, z)$ is set as the negative log likelihood of $p(\mathbf{y}|\mathbf{x}, z)$. Comparing with the decoding step in traditional Seq2seq models (Bahdanau, Cho, and Bengio 2015), we integrate more information from the input representations relevant to the selected word z to help predict the current output word $\mathbf{y}(t)$.

Next, we discuss how to design a proper function $f(\cdot)$ with $\ell(\mathbf{y}|\mathbf{x}, z)$'s estimated from the above network structure. Typically, $f(\cdot)$ is set to be the average function:

$$\mathcal{L}_{avg}(\{\mathbf{y}\}|\mathbf{x}, z) = \frac{1}{|\{\mathbf{y}\}|} \sum_{\mathbf{y} \in \{\mathbf{y}\}} \ell(\mathbf{y}|\mathbf{x}, z), \quad (5)$$

where $|\{\mathbf{y}\}|$ is the cardinality of $\{\mathbf{y}\}$. However \mathcal{L}_{avg} may not be proper in our setting. As discussed in Section , not all responses may relate to a single z . Thus, the loss terms $\ell(\mathbf{y}|\mathbf{x}, z)$'s for those \mathbf{y} 's relevant to z will be small, whereas

the remaining will be large. The overall loss $\mathcal{L}_{avg}(\{\mathbf{y}\}|\mathbf{x}, z)$ could still be large. Hence, we propose to use the minimum function as the $f(\cdot)$ function:

$$\mathcal{L}_{min}(\{\mathbf{y}\}|\mathbf{x}, z) = \min_{\mathbf{y} \in \{\mathbf{y}\}} \ell(\mathbf{y}|\mathbf{x}, z). \quad (6)$$

If the sampled z paired with \mathbf{x} can generate one of its responses very well, its overall loss value now will be small.

Joint Training with Reinforcement Learning

To update the networks, we generally compute the gradient of $J(\Theta)$ in Eq. 1 with respect to Θ :

$$\begin{aligned} \nabla_{\Theta} J(\Theta) &= \nabla_{\Theta} \sum_{z \in \mathbf{Z}} p(z|\mathbf{x}) \mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z) \\ &= \sum_{z \in \mathbf{Z}} [\mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z) \nabla_{\mathbf{W}} p(z|\mathbf{x}) + p(z|\mathbf{x}) \nabla_{\mathbf{G}} \mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z)] \\ &= E_{p(z|\mathbf{x}; \mathbf{W})} [\mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z) \nabla_{\mathbf{W}} \log p(z|\mathbf{x}) + \nabla_{\mathbf{G}} \mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z)]. \end{aligned}$$

A major difficulty faced by our model is the intractability in calculating the above gradient since the expectation involves a huge number of terms for a large latent space \mathbf{Z} . To address this problem, we propose to use reinforcement learning for optimization, which is shown in Algorithm 1.

In Step 1, we use existing keyword extraction tools to extract keywords for each post as the latent words to pre-train the latent word inference network. Then we use the top one inferred word from the latent word inference network as the input z to pre-train the generation network with the standard loss function Eq. 5 used in Seq2seq models. During training, the REINFORCE algorithm (Williams 1992) is used to train the latent word inference network, and standard back-propagation is adopted for updating the generation network.

Two key issues remains when we perform the joint training. First, when we perform K trials to sample multiple z 's, it may be difficult and time-consuming for the algorithm to explore the large latent space, if we directly use the sampling distribution in Eq. 2. Second, the loss function in Eq. 5/Eq. 6 is not bounded and thus will introduce a large variance in policy gradients, which affects the stability of the algorithm.

To address the first issue, we design a two-step sampling scheme, which can effectively limit the word sampling in a much smaller latent space. For the second issue, we propose to replace $\mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z)$ in the policy gradient with a bounded reward $\mathcal{R}(\{\mathbf{y}\}|\mathbf{z}, \mathbf{x})$.

Sampling multiple diverse latent words Firstly, we notice that each input post \mathbf{x} is often relevant to only a few z 's, since the information provided by a short text should be very limited. Thus, we maintain a small candidate set $\mathbf{Z}_{\mathbf{x}}$ for each \mathbf{x} , and only sample the latent z in $\mathbf{Z}_{\mathbf{x}}$ using their $p(z|\mathbf{x}; \mathbf{W})$. In our experiment, the candidate set $\mathbf{Z}_{\mathbf{x}}$ is constructed using outputs from existing keyword extraction methods on the input \mathbf{x} and its output sequences $\{\mathbf{y}\}$.¹

Second, it is desirable that K diverse z 's can be selected out for each \mathbf{x} . Thus we perform clustering on the candidate set $\mathbf{Z}_{\mathbf{x}}$ using their context vector \mathbf{c}_z 's in Eq. 4. For z 's in

¹We have tried using all words in \mathbf{Z} as $\mathbf{Z}_{\mathbf{x}}$ and the performance is much worse.

Algorithm 1: RL Training Algorithm

Input: $(\mathbf{x}, \{\mathbf{y}\}), \mathbf{Z}$

Output: Θ

- 1: $\Theta \leftarrow$ Pre-trained from baseline methods.
 - 2: **for** each $(\mathbf{x}, \{\mathbf{y}\})$ **do**
 - 3: **for** $k = 1, \dots, K$ **do**
 - 4: Sample a word z according to the proposed sampling scheme.
 - 5: Obtain the generated sentence $\hat{\mathbf{y}} = \text{argmin}_{\mathbf{y}^*} \ell(\mathbf{y}^*|\mathbf{x}, z)$.
 - 6: Compute the reward according to $\mathcal{R}(\{\mathbf{y}\}|\mathbf{x}, z)$.
 - 7: Update the latent word inference network via the policy gradient $\mathcal{R}(\{\mathbf{y}\}|\mathbf{x}, z) \nabla_{\mathbf{W}} \log p(\mathbf{z}|\mathbf{x})$.
 - 8: Update the generation network via the gradient $\nabla_{\mathbf{G}} \mathcal{L}(\{\mathbf{y}\}|\mathbf{x}, z)$.
 - 9: **end for**
 - 10: **end for**
-

different clusters, they tend to have different attentions on the input post. For each trial of sampling, we only select latent words from clusters, from which no words has been selected before.

Reward function design A few recent works have been proposed to learn a discriminator for a data-driven reward function (Li et al. 2017; 2018). However, these methods involve more model modules and complex parameter tuning. To keep our model simple, we make use of the generated sentence $\hat{\mathbf{y}}$ from the generation network (Step 5 in Algorithm 1), and compute the reward with respect to a single \mathbf{y} using the parameter-free F1 score between \mathbf{y} and $\hat{\mathbf{y}}$ which measures the average overlap by treating the two sequences as bags of tokens:

$$\mathcal{R}(\mathbf{y}|\mathbf{z}, \mathbf{x}) = \begin{cases} \text{F1}(\mathbf{y}, \hat{\mathbf{y}}), & \mathbf{y} \cap \hat{\mathbf{y}} \neq \emptyset, \\ -1, & \text{else.} \end{cases}$$

The reward with respect to $\{\mathbf{y}\}$ is computed as:

$$\mathcal{R}(\{\mathbf{y}\}|\mathbf{x}, z) = \begin{cases} \frac{1}{|\{\mathbf{y}\}|} \sum_{\mathbf{y}} \mathcal{R}(\mathbf{y}|\mathbf{x}, z), & \text{Eq. 5 is used,} \\ \max_{\mathbf{y}} \mathcal{R}(\mathbf{y}|\mathbf{x}, z), & \text{Eq. 6 is used.} \end{cases}$$

Experimental results show that our model with the use of this simple reward function works very well.

Generating Multiple Responses

With a trained model, we discuss how to generate K diverse responses for a testing input \mathbf{x} . We set the top 1000 z 's of $p(z|\mathbf{x})$ from the latent word inference network as the candidate set. Next, we perform clustering on the candidate set similarly as in training and select out the centroid z 's of the largest K clusters. Finally, each of the selected z 's is combined \mathbf{x} to generate the response.

Experiments

In this experiment, we first present our two datasets and introduce the compared methods. The training details including the keyword extraction, clustering and network configurations, are provided in Appendix. Next, we evaluate the

performance of the various compared methods. We further show some analyses to validate the effectiveness of our RL training algorithm.

Datasets

We perform experiments on two short-text conversation tasks in which each input post is generally associated with multiple output sequences. **Weibo**: We used the benchmark dataset (Shang, Lu, and Li 2015) and pre-processed it for high-quality data pairs. In total, we have above 4 million training pairs. **Twitter**: We crawled post-response pairs using the Twitter API. With several data cleaning steps, we have around 750 thousand training pairs. Both datasets have the vocabulary size 50,000. Details of the data pre-processing and statistics are in Appendix.

Compared Methods

Beam Search(BS): We use the vanilla Seq2seq model with attention employed in decoding (Bahdanau, Cho, and Bengio 2015). Standard beam search is applied in testing.

Diverse Beam Search(DBS) (Li, Monroe, and Jurafsky 2016): We use the same Seq2seq model for training. In testing, a modified ranking score is used in beam search to encourage diverse results. We set a fixed diversity rate as 0.5.

Maximum Mutual Information(MMI) (Li, Monroe, and Jurafsky 2016): We generate the N-best list using the same Seq2Seq model, and then rerank the results by this method.

Multiple-Mechanism(MultiMech) (Zhou et al. 2017) : It introduced latent embeddings for diverse response generation. Following their setting, we use 4 latent mechanisms. Then we select top 3 mechanisms and perform beam search to obtain one response for each top mechanism.

CVAE (Zhao, Zhao, and Eskenazi 2017): As stated in Section , it adopted CVAE for diverse response generation. We replace the dialogue acts used in their original model as the keyword used in our pre-trained step.

HGFU (Yao et al. 2017): This model used the PMI to extract a cue word for a post and then incorporate it for generating a more informative response. However, we find calculating the PMI is very time-consuming for a testing sample, which involves $O(\text{the sequence length} \times \text{vocabulary size} \times \text{the number of the training data})$ operations. Thus we change to apply the keyword used in our pre-trained step.

Ours: We explore our model using the pre-trained networks only (**Ours(Pretrain)**) and two variants utilizing the two loss functions in Eq. 5 and 6 in the generation network, denoted as **Ours(Avg)** and **Ours(Min)**.

Evaluation

Top 3 results of each method are collected for evaluation. For automatic evaluations, we report:

BLEU (Papineni et al. 2002): A widely used metric to measure the response relevance;

Distinct-1/2: Number of distinct unigrams/bigrams in the generated multiple responses scaled by the total number of generated unigrams/bigrams for each post, which can be considered an automatic metric for evaluating the diversity among the multiple responses generated for a single post.

Since automatic metrics for open-domain generative models may not be consistent with human perceptions (Liu et al. 2016), we also do human annotations to further evaluate the generated results. We randomly select 200 testing posts for each dataset and recruit 3 annotators from a professional labeling company to label the following two aspects for results of all compared methods:

Quality: Each post-response pair is evaluated with: bad (ungrammatical or irrelevant), normal (basically grammatical and relevant) or good (grammatical and highly relevant with specific meaning and information); responses on normal and good levels are treated as “acceptable”;

Diversity: Number of fundamentally distinct responses among the output responses of each method; thus the largest diversity score of a post is 3 in our setting.

Results on Weibo

Overall Performance Results of automatic evaluations and human annotations are shown in Table 1 and 2 respectively. In terms of BLEU, HGFU performs the best, while Ours(Min) achieves the second best on BLEU-2/3/4. However, BLEU only evaluates how much overlapped words the generated responses have towards the given references. In open-domain conversations, responses not close to the references may not necessarily be poor. Thus, the quality scores from the human annotations are more reliable. As shown, Ours(Min) obtains the highest acceptable ratio. More importantly, our model achieves a much higher good ratio, with approximately 195% improvement over HGFU. This indicates that our method can output more high-quality responses. In terms of distinct-1/2 and diversity, Ours(Min) performs the best. Especially from the human evaluation results, Ours(Min) obtains about 110% improvement over the Seq2seq baselines (BS, DBS and MMI), as well as 70% over the enhanced methods (MultiMech and HGFU). This validates that our method can output more diverse responses while maintaining each of them to be relevant to the post.

Comparing with Ours(Pretrain), Ours(Avg) cannot obtain consistent improvement on both quality and diversity. This shows that the loss in Eq. 6 is more suitable for our problem setting as discussed in Section 3.

We notice that the re-ranking methods (DBS and MMI) do not outperform BS considering both quality and diversity. As stated in introduction, these methods may have limited effectiveness by only encouraging responses with different words to be ranked into higher positions during testing. Also, the diversity of CVAE is much worse than the other methods. This is consistent with our discussion in Section that directly sampling from a fixed Gaussian prior tends to cause mode collapse and yields little response diversity.

Analysis and Case study We first evaluate the overall relevance between the latent words selected by three variants of our methods and their corresponding post/generated response. We use a simple method to obtain the sentence embeddings of the posts/ generated responses, which takes the average of all the word embeddings in the sentences (Adi et al. 2016; Zhao, Lee, and Eskenazi 2018). We use pre-trained word embeddings trained with the C-BOW model (Mikolov et al. 2013) on the Chinese Wikipedia corpus. Then we com-

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	distinct-1	distinct-2
BS	24.784	18.923	13.362	7.866	44.316	53.250
DBS	24.187	18.437	13.016	7.674	44.962	54.114
MMI	25.683	19.597	13.845	8.187	43.369	52.265
CVAE	26.575	18.290	12.302	6.510	38.158	47.272
MultiMech	24.436	18.952	13.644	8.403	41.321	49.592
HGFU	43.530	32.356	22.826	14.043	36.262	42.323
Ours(Pretrain)	31.135*	23.361	16.889	10.594	58.534*	76.023*
Ours(Avg)	30.435	23.187	16.511	10.086	43.618	51.769
Ours(Min)	31.067	23.523*	16.893*	10.945*	62.407	76.767

Table 1: Automatic evaluation results on Weibo. The best/second best results are bold/starred.

Input	2013年会即将开始, 还在琢磨给大家什么惊喜啊 The New Year's party 2013 is about to begin and I am still wondering what surprise to give to everybody	06月18日03时天津4.0级地震的震中图 18 June at 3:00 PM, the picture of epicenter of the magnitude 4.0 earthquake in Tianjin
MultiMech	我也想知道 I also want to know 恭喜恭喜! 恭喜恭喜! Congratulations! Congratulations! 恭喜恭喜! 恭喜恭喜! Congratulations! Congratulations!	我的天, 我的天! Oh, my god, my god! 我的天, 我的天! Oh, my god, my god! 我的天, 我的天! Oh, my god, my god!
Ours(Min)	有什么惊喜吗? ! [惊喜] What's the surprise?! [surprise] 新年快乐! 元旦快乐! [年会] Happy New Year! Happy New Year! [New Year's party] 哈哈, 新年快乐! 平安快乐! [感谢] Haha, happy New Year! Be safe and happy! [appreciate]	我的天, 太恐怖了! [地震] Oh, my god, that's terrible! [earthquake] 我的天津, 我的家乡。[天津] My Tianjin, my hometown. [Tianjin] 我的妈呀, 吓死我了[结束] Holy crap, I was scared to death [finish]

Figure 2: Examples showing the generated responses on Weibo. Words in the brackets are the selected latent words.

Method	Quality		Diversity
	%Acceptable	%Good	
BS	44.611	4.333	0.678
DBS	44.667	4.389	0.683
MMI	44.611	4.056	0.680
CVAE	29.833	1.888	0.380
MultiMech	60.388	5.000	0.806
HGFU	60.833	7.111	0.846
Ours(Pretrain)	46.722	9.611	1.148
Ours(Avg)	49.278	8.556	0.843
Ours(Min)	63.000	20.944	1.436

Table 2: Human evaluation results on Weibo.

pute the cosine similarity between the latent word and the sentence embedding, and normalize the score into [0,1]. Results are shown in Fig. 3(a). As can be seen, both Ours(Avg) and Ours(Min) can select out latent words that are more relevant to both the posts and responses than Ours(Pretrain). Meanwhile, the improvement of Ours(Min) is much larger than that of Ours(Avg).

Fig. 2 illustrates two example posts with their generated responses.² As can be seen, the responses with different mechanisms can be identical. The reason may be that the number of mechanisms is very small and each mechanism can only capture high-level responding characteristics shared by many input data. A given post may not be sensitive to different responding mechanisms so that the same

²Full results of all methods are shown in Appendix.

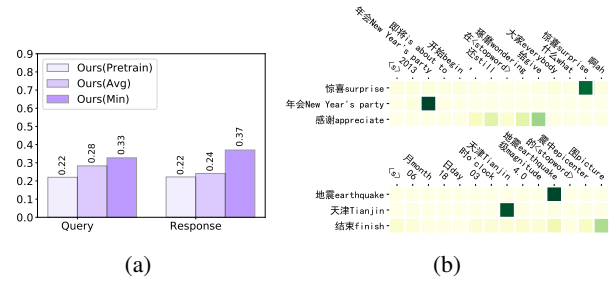


Figure 3: (a) Cosine similarity between the latent word and the query/response. (b) Visualization of the attention paid to the relevant parts of the input post in Fig. 2 for each of the selected words; a large version of this figure is shown in Appendix for better readability.

response will be generated. Whereas, the responses by our methods are driven by different latent words, which contain more specific data-dependent information. If the selected words have far different meanings, the generated responses are highly probable to be different.

We further examine the relationship between the latent words selected in our method and the generated results. Fig. 3(b) shows the attention vector of each selected word over its corresponding post. The first two latent words for both posts are highly aligned to one word in the post, which is often the case that this word appears in the post. Then our method probably generates specific responses about that

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	distinct-1	distinct-2
BS	39.595	34.331	27.727	22.857	51.867	63.951
DBS	37.004	32.075	25.836	21.198	53.703	66.226
MMI	44.306	38.727	31.760	26.661	48.678	60.135
CVAE	35.071	29.487	22.509	17.380	52.451	64.876
MultiMech	75.164	50.679	30.897	19.434	42.717	48.153
HGFU	28.145	18.214	12.963	9.488	5.786	6.609
Ours(Pretrain)	82.751	64.902	47.177	34.282	52.281	66.642
Ours(Avg)	82.886*	66.586*	48.288*	34.891*	55.371	67.210*
Ours(Min)	86.682	74.356	60.294	49.965	53.904*	67.301

Table 3: Automatic evaluation results on Twitter. The best/second best results are bold/starred.

Input	that hangover life is real	hi milana. i cant wait to watch you perform tonight. god bless
MultiMech	so true ! the real deal so true !	good luck ! i love you oh my god
Ours(Min)	you know what to say [real] i love the way you think! [love] i thought the same. [enjoy]	i love her ! she is so beautiful. [congratulations] i am going to see that song show. be safe, get her a ticket! [album] i am ready [shut]

Figure 4: Examples showing the generated responses on Twitter. Words in the brackets are the selected latent words.

Method	Quality		Diversity
	%Acceptable	%Good	
BS	28.055	3.778	0.657
DBS	29.000	3.722	0.697
MMI	31.667	4.667	0.678
CVAE	20.666	3.055	0.475
MultiMech	21.166	3.500	0.536
HGFU	14.000	2.722	0.245
Ours(Pretrain)	27.389	4.333	0.585
Ours(Avg)	30.667	4.444	0.570
Ours(Min)	42.000	7.944	0.713

Table 4: Human evaluation results on Twitter.

word. If the word is aligned with multiple words as the third one for both posts, the generated response is relevant to the these attended words but not the latent word itself. Overall, since we select words from different clusters, which are computed using the context vector c_z , our method tends to select words attended on different parts of the post and the generated responses are generally with diverse meanings.

Results on Twitter

Table 3 shows the automatic evaluation annotation results on Twitter. On all metrics, our proposed models outperform the other compared methods. The human evaluation results are provided in Table 4, and our method is still the best among all compared methods. Especially on the the Good ratio, Ours(Min) has an increase of 70% over MMI (the best re-ranking method) and 126% over MultiMech (the best diversity-promoting method). Generated responses of two posts are provided in Fig. 4. Full results of all methods are again provided in Appendix.

Analysis of RL Training

We vary the size of the latent space on Twitter to test if a large latent space is needed in our model. In Fig. 5(a), we

show the results of Ours(min) with the latent space using the top 1k, 10k and all words in the vocabulary. We find out that all metric scores decrease drastically with the use of a smaller latent space. Thus, it is critical to use a large latent space for our model to perform well.

In Fig. 5(b), we plot the average training reward of Ours(Min) obtained by each epoch on Twitter. The reward increases as the training algorithm iterates. It indicates our designed RL training algorithm is capable of revising its policy to obtain more accurate latent variable distributions to generate better responses.

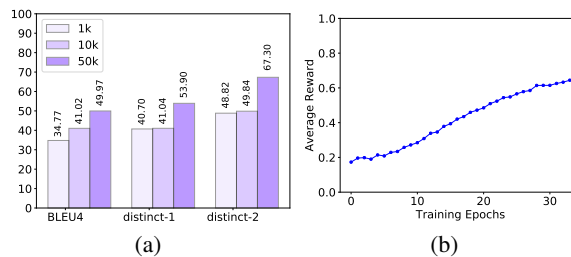


Figure 5: (a) BLEU and distinct-1/2 with the size of latent space as 1k, 10k and 50k. (b) Training rewards with different epochs.

Conclusion

We propose a new response generation model that learns the post with its set of responses jointly for short-text conversation. Our model can be optimized in a reinforcement learning algorithm, which can deal with the large latent space assumed in our model. By sampling multiple diverse latent words from the latent word inference network, the generation network can output different responses. We perform extensive experiments on two real datasets collected from

Weibo and Twitter. Results show that, our method can effectively increase both the quality and diversity of the multiple generated responses compared with existing baselines and several state-of-the-art generative methods.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No. 61876120).

References

- Adi, Y.; Kermany, E.; Belinkov, Y.; Lavi, O.; and Goldberg, Y. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.
- Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. *Proceedings of the 2015 International Conference on Learning Representations (ICLR)*.
- Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 110–119.
- Li, J.; Monroe, W.; Ritter, A.; Jurafsky, D.; Galley, M.; and Gao, J. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1192–1202.
- Li, J.; Monroe, W.; Shi, T.; Jean, S.; Ritter, A.; and Jurafsky, D. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2157–2169.
- Li, Z.; Jiang, X.; Shang, L.; and Li, H. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Li, J.; Monroe, W.; and Jurafsky, D. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.
- Liu, C.-W.; Lowe, R.; Serban, I.; Noseworthy, M.; Charlin, L.; and Pineau, J. 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*, 2122–2132.
- Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mo, K.; Li, S.; Zhang, Y.; Li, J.; and Yang, Q. 2016. Personalizing a dialogue system with transfer reinforcement learning. *arXiv preprint arXiv:1610.02891*.
- Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 311–318. Association for Computational Linguistics.
- Shang, L.; Lu, Z.; and Li, H. 2015. Neural responding machine for short-text conversation. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)* 1577–1586.
- Shao, L.; Gouws, S.; Britz, D.; Goldie, A.; Strophe, B.; and Kurzweil, R. 2017. Generating long and diverse responses with neural conversation models. *arXiv preprint arXiv:1701.03185*.
- Shen, X.; Su, H.; Li, Y.; Li, W.; Niu, S.; Zhao, Y.; Aizawa, A.; and Long, G. 2017. A conditional variational framework for dialog generation. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Sordani, A.; Galley, M.; Auli, M.; Brockett, C.; Ji, Y.; Mitchell, M.; Nie, J.-Y.; Gao, J.; and Dolan, B. 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-HLT)*, 196–205.
- Turing, A. 1950. Computing machinery and intelligence. *Mind*.
- Vinyals, O., and Le, Q. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.
- Wang, L.; Schwing, A.; and Lazebnik, S. 2017. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *Advances in Neural Information Processing Systems*, 5756–5766.
- Wen, T.-H.; Miao, Y.; Blunsom, P.; and Young, S. 2017. Latent intention dialogue models. In *International Conference on Machine Learning*, 3732–3741.
- Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8(3-4):229–256.
- Yao, L.; Zhang, Y.; Feng, Y.; Zhao, D.; and Yan, R. 2017. Towards implicit content-introducing for generative short-text conversation systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2190–2199.
- Zhao, T.; Lee, K.; and Eskenazi, M. 2018. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Zhao, T.; Zhao, R.; and Eskenazi, M. 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)*, 654–664.
- Zhou, G.; Luo, P.; Cao, R.; Lin, F.; Chen, B.; and He, Q. 2017. Mechanism-aware neural machine for dialogue response generation. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 3400–3407.
- Zhou, G.; Luo, P.; Xiao, Y.; Lin, F.; Chen, B.; and He, Q. 2018. Elastic responding machine for dialog generation with dynamically mechanism selecting. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*.
- Zhou, Z.-H. 2004. Multi-instance learning: A survey. *Department of Computer Science & Technology, Nanjing University, Tech. Rep.*