MEMD: A Diversity-Promoting Learning Framework for Short-Text Conversation

Meng Zou **Xihan Li** **Haokun Liu** **Zhihong Deng***
Key Laboratory of Machine Perception (Ministry of Education)
School of Electronics Engineering and Computer Science
Peking University, Beijing 100871, China
{mengzou, xihanli, 1300012769, zhdeng}@pku.edu.cn

Abstract

Neural encoder-decoder models have been widely applied to conversational response generation, which is a research hot spot in recent years. However, conventional neural encoder-decoder models tend to generate commonplace responses like *"I don't know"* regardless of what the input is. In this paper, we analyze this problem from a new perspective: latent vectors. Based on it, we propose an easy-to-extend learning framework named MEMD (Multi-Encoder to Multi-Decoder), in which an auxiliary encoder and an auxiliary decoder are introduced to provide necessary training guidance without resorting to extra data or complicating network's inner structure. Experimental results demonstrate that our method effectively improve the quality of generated responses according to automatic metrics and human evaluations, yielding more diverse and smooth replies.

1 Introduction

Human-computer conversation is attracting particular attention recently. Research in this field falls into two categories: the retrieval-based method (Ji et al., 2014; Yan et al., 2017; Wu et al., 2017) and the generative method (Shang et al., 2015; Serban et al., 2016). While the retrieval-based method can guarantee completeness of output sentences, it fails to customized for particular posts from users. By contrast, the generative method may produce sentences with grammatical errors, however, it shows great promise in flexibility, which gives rise to a research hot spot.

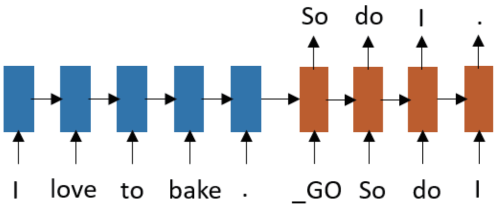


Figure 1: Conventional encoder-decoder model. Blue part represents the encoder, while red part represents the decoder.

Of the generative method, neural encoder-decoder based model (Sutskever et al., 2014) has become the mainstream (Figure 1). (Shang et al., 2015) firstly formalized the generation of response as a decoding process based on the latent representation of the input text, and both encoding and decoding are realized with recurrent neural networks (RNN). Such formalization is widely adopted by later work. However, the conventional encoder-decoder model's performance is far from satisfactory, for it tends to generate meaningless and generic responses like *"I don't know"*.

To tackle the issue of response diversity, lots of models have been proposed, and they can be broadly divided into three categories: (1) introducing external priori knowledge into the procedure of encoding/decoding (Mou et al., 2016; Xing et al., 2017), which usually requires pretreatment on extra large

*Zhihong Deng is the corresponding author.

Proceedings of the 27th International Conference on Computational Linguistics, pages 1281–1291
Santa Fe, New Mexico, USA, August 20-26, 2018.

data. (2) complicating network's structure to enhance model's capacity for encoding/decoding (Zhou et al., 2017; Serban et al., 2017a; Serban et al., 2017b; Zhao et al., 2017; Clark and Cao, 2017; Shen et al., 2017). (3) directly modifying the objective function to penalize generation probability of trivial responses (Li et al., 2016). However, these models merely discuss the latent vector, which is the vital link connecting the encoder and the decoder together.

Different from previous work, we analyse the problem of general response generation from the perspective of latent vectors. We notice that in the basic encoder-decoder model, what the decoder needs to generate a response is only a hidden vector. In other words, for the same decoder, latent vectors directly decide what will be generated. If latent vectors are clustered, the decoder is likely to generate similar responses. If latent vectors are dispersed, the decoder is likely to generate diverse responses. Based on the conjecture that dispersion of latent vectors has positive correlation with diversity of generated responses, encoder-decoder model's poor performance can be discussed in the two following situations:

- Suppose that the decoder is capable to generate different sentences given different latent vectors, model's unsatisfying performance of generating meaningless sentences should be imputed to the encoder—it tends to map whatever inputs to similar vectors. In this situation, to promote diversity in generation, the encoder should be encouraged to generate different latent vectors given different inputs.

- Suppose that the encoder is capable to generate different latent vectors with specific and concrete semantics given different inputs, model's unsatisfying performance of generating meaningless sentences should be imputed to the decoder—it is insensitive to latent vectors and tends to generate similar sentences given whatever latent vectors. In this situation, to promote diversity in generation, decoder's capacity for generating sentences should be enhanced.

During training, however, since there is no constraint on the latent vector's specificity, neither encoder's capacity for latent vector generation nor decoder's capacity for sentence generation can be guaranteed, which leads to poor performance in conversation generation.

For the above motivation, we propose a learning framework named MEMD (Multi-Encoder to Multi-Decoder). In proposed framework, an auxiliary encoder, which aims at guiding the major encoder to generate diverse latent vectors, and an auxiliary decoder, which aims at providing latent vectors with specific and concrete semantics for the major decoder to "practice" decoding, are introduced. During training, parameters of these two encoders and two decoders are updated. While in test, only the major encoder and the major decoder are employed.

In summary, our contributions are as follows:

- We present a new angle to tackle the problem of response diversity—the latent vectors generated by the encoder.

- We propose an easy-to-extend learning framework: MEMD, which introduces necessary training guidance for both encoder and decoder without resorting to extra data or complicating inner structure of networks.

- Experimental results demonstrate that MEMD effectively improves the quality of generated responses according to automatic metrics and human evaluations, yielding more diverse and smooth replies.

2 Technical Background

2.1 Gated Recurrent Unit (GRU)

GRU (Cho et al., 2014) is a special kind of RNN, which is widely used for learning long-term dependencies. It is defined as follows: given a sequence of inputs $(w_1, w_2, \ldots, w_N)$, GRU iterates each timestep with an update gate z_n and a reset gate r_n. Let h_n denote the vector of hidden layer computed by GRU at time n, σ denote the sigmoid function and $\odot$ denote the element-wise product. The vector representation of hidden layer for each timestep n is given by:

$$\begin{aligned} z_n &= \sigma(W_{zw}w_n + W_{zh}h_{n-1}) && (1)\\ r_n &= \sigma(W_{rw}w_n + W_{rh}h_{n-1}) && (2)\\ \widetilde{h}_n &= tanh(W_{hw}w_n + W_{hh}(r_t \odot h_{t-1})) && (3)\\ h_n &= (1-z_n)h_{n-1} + z_n\widetilde{h}_n && (4) \end{aligned}$$

where W_{*w} is the transformation matrix from the input to GRU states, W_{*h} is the recurrent transformation matrix between the recurrent states h_n.

2.2 Encoder-decoder Models

In an encoder-decoder model, given a source sequence message $X = (x_1, x_2, \ldots, x_M)$ and a target sequence response $Y = (y_1, y_2, \ldots, y_N)$, the model would maximizes the generation probability of Y conditioned on X. While the encoder reads X word by word and represents it as a latent vector h_X through a recurrent neural network (RNN), the decoder estimates the generation probability of Y with h_X as initial state. The objective function of the model is as follows:

$$p(y_1, \ldots, y_N | x_1, \ldots, x_M) = p(y_1|X) \prod_{t=2}^{N} p(y_t|y_1, \ldots, y_{t-1}, X), \tag{5}$$

The latent vector h_X is calculated by

$$\begin{aligned} h_t &= f(x_t, h_{t-1}) && (6)\\ h_X &= h_M && (7) \end{aligned}$$

where h_t is the hidden state at time t and f is a non-linear transformation which can be a gated recurrent unit (GRU). The decoder is a standard RNN language model except the addition of the context vector c. The probability distribution p_t of candidate words at each timestep t is calculated as

$$\begin{aligned} s_0 &= h_X && (8)\\ s_t &= f(y_{t-1}, s_{t-1}) && (9)\\ p_t &= softmax(s_t) && (10) \end{aligned}$$

where s_t is the hidden state of the decoder RNN at timestep t.

2.3 Attention Mechanism

The traditional sequence-to-sequence model assumes that each word is generated from the same context vector. However, in practice, different words in Y might be related to different words or phrases in X. In order to solve this problem, attention mechanism (Bahdanau et al., 2015; Cho et al., 2014) is introduced into this model. With attention, the context vector c_i corresponded to each y_i in Y is a weighted average of all hidden states of the encoder. Formally, c_i is defined as

$$\begin{aligned} c_i &= \sum_{j=1}^{M} \alpha_{ij} h_j && (11)\\ \alpha_{ij} &= \frac{exp(e_{ij})}{\sum_{k=1}^{M} exp(e_{ik})} && (12)\\ e_{ij} &= \eta(s_{i-1}, h_j) && (13) \end{aligned}$$

where η is a multi-layer perceptron (MLP).

3 Proposed Framework

The basic idea of MEMD is simple: we want the encoder to be able to generate different latent vectors given different inputs and the decoder to be able to generate sentence given a latent vector. We prevent the encoder from generating similar vectors by requiring that posts themselves should be reconstructed from the latent vectors, since if vectors are similar, they cannot be interpreted into diverse sentences by the same decoder. And for decoder, we strengthen its decoding ability by requiring it to reconstruct responses given vectors that really encode responses, since latent vectors' effectiveness cannot be guarantee when the encoder takes as input not responses but posts.

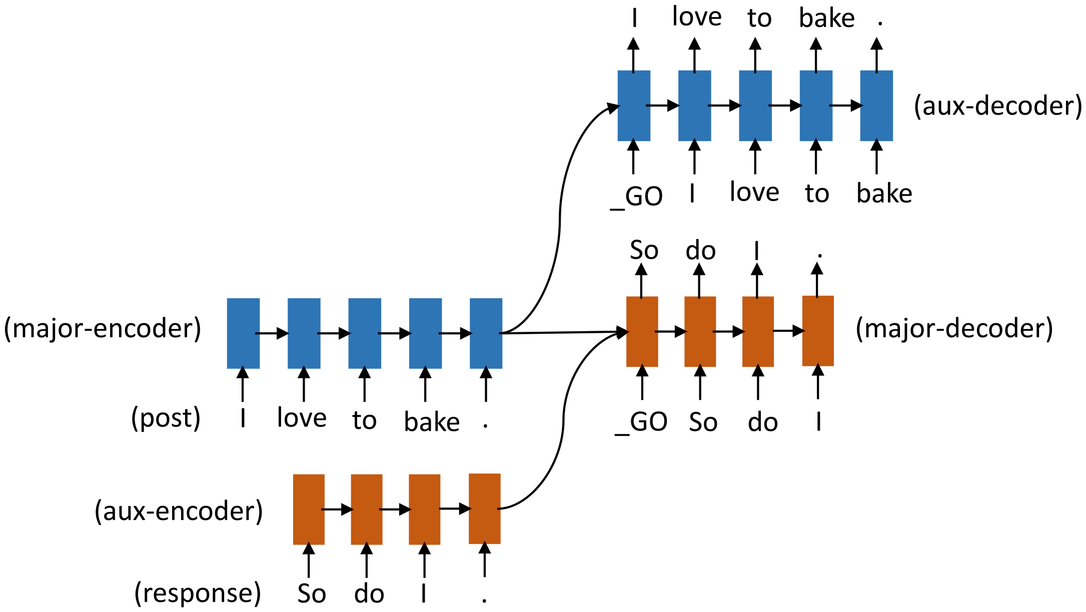


Figure 2: Architecture of MEMD. This framework contains four main components: 1) a major encoder, 2) a major decoder, 3) an auxiliary encoder, and 4) an auxiliary decoder. Three training paths are constructed: 1) from major-encoder to aux-decoder, 2) from major-encoder to major-decoder, and 3) from aux-encoder to major-decoder. Blue parts' inputs/targets are posts, while red parts' inputs/targets are responses.

3.1 Model Architecture

Suppose that short-text conversation consists of a post and a response, we denote a post of length M as $X = [x_1, \ldots, x_M]$ and a response of length N as $Y = [\_GO, y_1, \ldots, y_N]$ where $\_GO$ is a special symbol indicating the begin of a response. Each training example is a (post, response) pair, namely, (X, Y).

Figure 2 illustrates the architecture of MEMD, in which there are two encoders—major encoder (major-encoder) and auxiliary encoder (aux-encoder), and two decoders—major decoder (major-decoder) and auxiliary decoder (aux-decoder). These four components constitute three training paths: path 1 is from major-encoder to aux-decoder, path 2 is from major-encoder to major-decoder, and path 3 is from aux-encoder to major-decoder. Note that the major-encoder produces only one latent vector when fed one post, and the major-decoder receives only one latent vector per decoding procedure. In other words, the major-encoder is shared between path 1 and path 2, and the major-decoder is shared between the path 2 and path 3.

The major-encoder takes as input a post X, and returns its hidden state vector at the last step, i.e., h_M, as output. The aux-encoder has the same structure as the major-encoder, but takes as input the response Y. The output of the aux-encoder is also it's hidden state vector at the last step, i.e., $\hat{h}_N$. The major-decoder takes as input h_M or $\hat{h}_N$, and it's generation target is Y. When it takes h_M as input, it constitutes a conventional encoder-decoder model with the major-encoder. When it takes $\hat{h}_N$ as input, it constitutes an auto-encoder with the aux-encoder. The aux-decoder has same structure as the major-decoder. It takes as input h_M, and its generation target is X.

3.2 Training Procedure

We first present the training objective along each path, and then give the whole training algorithm.

We denote the parameters of the major-encoder as θ_{ME}, and the parameters of aux-decoder as θ_{AD}. The training objective of path 1 (from the major-encoder to the aux-decoder) is:

$$\min_{\theta_{ME},\theta_{AD}} L_{ME-AD} = -\log p_1(X|X) \tag{14}$$

The parameters of the major-decoder are denoted as θ_{MD}. The training objective of path 2 (from the major-encoder to the major-decoder) is:

$$\min_{\theta_{ME},\theta_{MD}} L_{ME-MD} = -\log p_2(Y|X) \tag{15}$$

The parameters of the aux-encoder are denoted as θ_{AE}. The training objective of path 3 (from the aux-encoder to the major-decoder) is:

$$\min_{\theta_{AE},\theta_{MD}} L_{AE-MD} = -\log p_3(Y|Y) \tag{16}$$

The overall training objective is:

$$\min_{\theta_{ME},\theta_{AE},\theta_{MD},\theta_{AD}} L = L_{ME-AD} + L_{ME-MD} + L_{AE-MD} \tag{17}$$

In actual implementation, for the sake of flexibility and extendibility, we don't directly optimize Eq.(17) but interleave the optimization of Eq.(14), Eq.(15) and Eq.(16) at each iteration, which is inspired by the alternating training approach (Dong et al., 2015). Algorithm 1 summarizes the training procedure.

Algorithm 1 MEMD for short-text conversation

```
Input: Training data $\{(X, Y)_n\}$
Output: major-encoder, aux-encoder, major-decoder, aux-decoder
1: Initialize $\theta_{ME}, \theta_{AE}, \theta_{MD}, \theta_{AD}$
2: repeat
3:     Train through path 1 by Eq.(14)
4:     Train through path 2 by Eq.(15)
5:     Train through path 3 by Eq.(16)
6: until convergence
```

3.3 Discussion

Our proposed framework MEMD is seemingly similar to the many-to-many setting in multi-task sequence-sequence learning (Luong et al., 2015). However, there are obvious distinctions between MEMD and multi-task sequence-sequence learning. MEMD aims at introducing constrains for encoder and decoder from the perspective of latent vectors, and does not require extra data for training. These introduced constrains are designed based on the characteristics of conversational response generation task. The many-to-many setting in multi-task sequence-sequence learning, however, aims at improving the generalization performance of the central task by resorting to training data of other related tasks.

4 Experiments

4.1 Dataset

We carry out experiment on an open-domain dialogue dataset: STC-weibo corpus developed by (Shang et al., 2015). STC-weibo corpus consists of millions of post-response pairs crawled from Weibo[1], which is popular Twitter-like microblogging service in China and has length limit of 140 Chinese characters on both posts and responses. We filter post-response pairs that include "alink" which represents a hyperlink, since we find that sentences are low-quality when "alink" appears. Besides, each post corresponds to 28 different responses at average. To minimize noise, we selected the response that contains the maximum number of frequent bigram in the whole corpus. After data cleaning, we finally get 199384 post-response pairs, and conduct the train/dev/test split of 197424/1000/960.

4.2 Baselines

We use the following models that needn't resort to extra data as our baselines for fair comparison:

Enc-Dec: the standard encoder-decoder model.

Enc-Dec-A: the standard encoder-decoder model with attention.

MMI: the best performing model in (Li et al., 2016).

For each baseline, there is a corresponding version of MEMD, whose major-encoder and major-decoder are the same to the baseline's encoder and decoder respectively. In other words, the baseline and its corresponding MEMD are structurally identical in test. Under this controlled setting we can validate the effectiveness of the proposed learning framework.

4.3 Implementation Details

We implement models in TensorFlow[2] and train them using Adam. The encoder is implemented as bidirectional GRU, and the decoder is implemented as multi-layer GRU (3 layers in Enc-Dec and Enc-Dec-A, 2 layers in MMI). The dimensions of hidden state are set to be 512 in Enc-Dec and Enc-Dec-A, and 256 in MMI. We use 100-dimension word embedding, and keep the size of vocabulary to be 60000. The word embedding is pretrained on the training set and updated during training. We set the learning rate to be 2×10^{-3} for path 2 and 3×10^{-4} for path 1 and path 3. And the batch size is set to be 48. We test the model on development data every 1000 mini-batches. When the model's performance on object function doesn't improve within 4 successive tests on development data, we view it convergent and stop training.

4.4 Evaluation metrics

Distinct-1 & distinct-2: Follow (Xing et al., 2017), we counted numbers of distinct unigrams and bigrams in the generated responses, and divide them by the total number of unigram and bigram respectively. The higher these two metrics are, the more informative and diverse the generated responses are.

Distinct-B & distinct-S: To measure the diversity of sentence pattern, we counted the number of distinct four words at the beginning of sentences, and divide them by the total number of generated sentences. We denote this metric as distinct-B. Moreover, we count the number of distinct sentence, and also calculate the ratio of distinct sentence to the total number of generated sentences. We denote this metric as distinct-S. The higher these two metrics are , the more diverse the generated responses are.

Sentence-level BLEU: Inspired by metrics used for evaluating machine translation, we use BLEU (Chen and Cherry, 2014) to evaluate the responses generated by different models.

Human annotation: Since automatic metrics may not consistently agree with human perception (Stent et al., 2005), we conduct human evaluation on 50 randomly sampled generated sentences. Three labelers with rich Weibo experience were invited to do evaluation. Responses generated by different models were pooled and randomly shuffled for each labeler. we adopt the criteria used in (Xing et al., 2017):

+2:The response is not only relevant and natural, but also informative and interesting.

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

[2] https://www.tensorflow.org/

+1: The response can be used as a reply to the message, but it is too universal like “Yes, I see”, “Me too” and “I don’ t know”.

0: The response cannot be used as a reply to the message. It is either semantically irrelevant or disfluent (e.g., with grammatical errors).

4.5 Results and Analysis

We investigate three strategies to initialize parameters of the major-encoder and the major-decoder. From Table 1 we can find that the way of initialization have a great influence on model's performance. When major-encoder and the major-decoder are pretained through path 1 and path 3 (advance-1&3), which in fact constitute two independent auto-encoders, the model tends to use the same words, and lacks variety on sentence pattern according to distinct-1, distinct-2, distinct-B and distinct-S. What's more, results of human annotation suggest that generated sentences are low-quality. When no pretraining is adopted, the model's diversity and generation quality are improved. When pretraining is conducted on path 2 (advance-2), which means the major-encoder and the major-decoder are initialized with a convergent Enc-Dec, the model gets significant improvement on diversity and generation quality. We notice that BLEU scores are somewhat incompatible to human annotation, here we put more priority on human annotation, and use BLEU scores as reference.

pretrain strategy	distinct-1	distinct-2	distinct-B	distinct-S	BLEU	+2	+1	0
advance-1&3	102/.010	308/.038	69/.072	388/.404	**0.569**	16%	54%	30%
no pretrain	172/.017	478/.059	154/.160	456/.475	0.536	26%	52%	22%
advance-2	**882/.086**	**2294/.276**	**547/.570**	**910/.948**	0.559	**40%**	36%	24%

Table 1: Results of MEMD on evaluation metrics. The first four columns are in the format of "the total number/proportion". Before employing algorithm 1, three strategies are adopted to initialize major-encoder's and major-decoder's parameters: 1) train them on path 1 and path 3 in advance. 2)no pretrain. 3)train them on path 2 in advance.

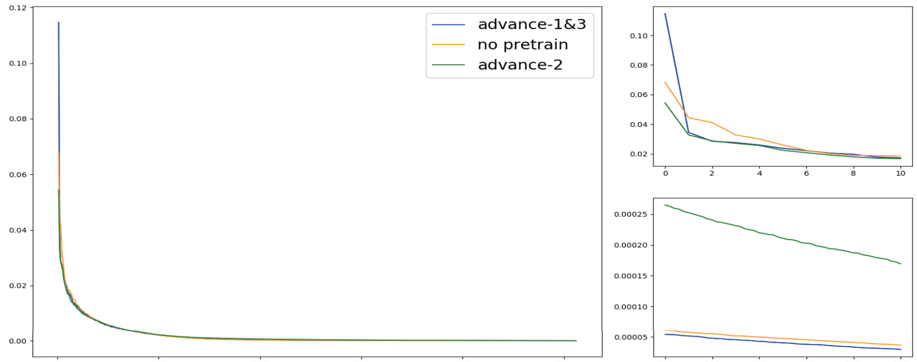


Figure 3: The left subfigure shows models' all 512 sorted PV under different pretraining strategies. The upper-right subfigure shows PV in the interval of 1 to 10, and the lower-right subfigure shows PV in the interval of 300 to 350.

Besides of evaluation metrics, we carry out analysis on latent vectors' distribution under different pretraining strategies. Since latent vectors under different pretraining strategies aren't in the same semantic space, direct comparison is infeasible. To tackle this problem, we utilize PCA (Principal Component Analysis) to get variances along each axes, which are represented by eigenvalues of covariance matrix subtracted the mean of each latent vectors. By dividing the eigenvalue by the sum of all eigenvalues, we get the percentage of variance (PV) explained by corresponding axis. If the variance of PV is large, which indicates variances are mainly distributed along a few axes, latent vectors are clustered. If the variance of

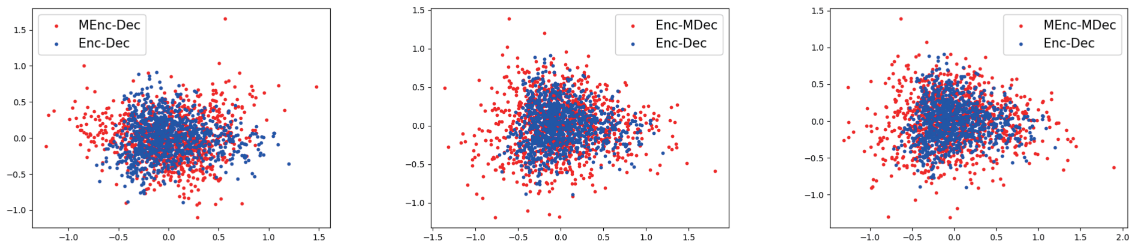
PV is small, which indicates they may have high variances along many axes, latent vectors are scattered. We use the variance of PV to indicate the dispersion of latent vectors. Models' sorted PV along all 512 axes are showed in Figure 3.

From Figure 3 we can clearly observe that the blue line has the largest value among three lines at the very beginning, then it drops down most sharply, and keeps small value later. This indicates that its most part of variances are occur on only a few axes, which means the latent vectors are clustered in the semantic space. The green line has the smallest value among three lines at the beginning and drops down most smoothly. This indicates its variances are distributed more evenly on all axes, so the latent vectors are more scattered in the whole semantic space. The PV variances of advance-1&3, no pretrain, and advance-2 are 4.251×10^{-5}, 3.142×10^{-5}, and 2.138×10^{-5} respectively, indicating that latent vectors of them are more scattered accordingly. The sorted PV distribution and variances of PV echo the results from Table 1.

To validate the effectiveness of the aux-encoder and the aux-encoder in the proposed MEMD, we designed two variations: 1) MEnc-Dec, which only has aux-encoder. 2)Enc-MDec, which only has aux-decoder. Together with MEMD, we get three models and train them with pretraining method of advance-2. The evaluation results are shown in Table 2. Both MEnc-Dec and Enc-MDec get higher scores on distinct-1, distinct-2, distinct-B, distinct-S, and human annotation comparing with Enc-Dec, which can prove that both aux-encoder and the aux-encoder do help promote diversity in response generation. The effectiveness of aux-decoder and aux-encoder support the two situations discussed in introduction. When compare Enc-MDec with MEnc-Dec, we find the latter brings greater promotion to the baseline on human annotation, which indicates that the aux-encoder plays important role in digesting the major decoder's ability to generate smooth sentences. When both aux-encoder and aux-decoder are adopted, MEMD's performance is further improved and gets best results on distinct-1, distinct-2, distinct-B， distinct-S, and human annotation.

	distinct-1	distinct-2	distinct-B	distinct-S	BLEU	+2	+1	0
Enc-Dec	148/.016	412/.055	145/.151	383/.399	0.555	18%	46%	36%
MEnc-Dec	796/.079	2125/.259	512/.533	870/.906	**0.568**	32%	42%	26%
Enc-MDec	730/.068	2133/.241	536/.558	877/.914	0.554	22%	50%	28%
MEMD	**882/.086**	**2294/.276**	**547/.570**	**910/.948**	0.559	**40%**	36%	24%

Table 2: Results of MEMD's variations on evaluation metrics. The first four columns are in the format of "the total number/proportion". Enc-Dec is the baseline, MEnc-Dec represents Enc-Dec with aux-encoder, and Enc-MDec represents Enc-Dec with aux-decoder.



(a)

(b)

(c)

Figure 4: Visualization of all latent vectors generated by Enc-Dec, Menc-Dec, Enc-MDec and MEnc-MDec in test.

We visualize two sets of all 960 latent vectors which are generated by Enc-Dec, MEnc-Dec, Dec-MDec and MEMD respectively in Figure 4. Note that the major-encoders and the major-decoders of MEnc-Dec, Dec-MDec and MEMD are initialized using Enc-Dec's parameters. We can clearly observe that after training under the MEMD framework, latent vectors become much scattered. Taking Figure 4 and Table 2 together, we find that dispersion of latent vectors has positive correlation with diversity of generated responses, which supports our conjecture.

	distinct-1	distinct-2	distinct-B	distinct-S	BLEU	+2	+1	0
Enc-Dec	148/.016	412/.055	145/.151	383/.399	0.555	18%	46%	36%
MEMD	**882/.086**	**2294/.276**	547/.570	**910/.948**	0.559	40%	36%	24%
Enc-Dec-A	310/.032	683/.088	162/.168	447/.466	**0.571**	20%	42%	38%
MEMD-A	822/.074	2157/.236	**550/.573**	873/.909	0.562	42%	36%	22%
MMI	345/.056	661/.157	317/.330	500/.521	0.440	32%	48%	20 %
MEMD-M	478/.073	820/.178	344/.358	572/.596	0.463	**46%**	32%	22%

Table 3: Results of baselines and their corresponding MEMD on evaluation metrics. The first four columns are in the format of "the total number/proportion".

To reveal the expansibility of the proposed learning framework, we apply MEMD to the other two baselines—Enc-Dec-A and MMI. From Table 3 we can see that MEMD-A has superior performance than Enc-Dec-A on distinct-1, distinct-2, distinct-B, distinct-S and human annotation, which indicates better diversity and quality of generated responses. MMI can produce smoother and more specific sentences than Enc-Dec and Enc-Dec-A, and MEMD-M gets better performance on distinct-1, distinct-2, distinct-B, distinct-S and human annotation than MMI. We also find that better performance on distinct-1, distinct-2, distinct-B, and distinct-S cannot guarantee better performance on human annotation, which can be explained as sentences contain varied words but have poor quality or are irrelevant to posts. Table 4 exhibits some examples. From Table 4 we can see that responses generated by Enc-Dec are general and lack pertinency to posts, while MEMD can catch some keywords in posts like "Hangzhou", "snow", and "talk". For Enc-Dec-A and MMI, they can sometimes generate natural responses, and based on them, MEMD-A and MEMD-M generate more informative and relevant responses.

Post	**Enc-Dec**	**MEMD**	**Enc-Dec-A**	**MEMD-A**	**MMI**	**MEMD-M**
杭州又下雪了，雪中西湖是最美的 It snows again in Hangzhou, and West Lake in the snow is the most beautiful.	这是哪里啊？我也想去 Where is this? I want to go, too.	杭州下雪了，我也在杭州 It snows in Hangzhou, and I am in Hangzhou, too.	我也想去杭州了。 I want to go to Hangzhou, too.	是杭州吗？杭州下雪了？ Is this Hangzhou? Is it snowing in Hangzhou?	... ...	西湖下雪了，西湖下雪了。 It snows in West Lake, it snows in West Lake.
已过零点，跟自己说一声：生日快乐。 The time has passed twelve o'clock at night, I talked to myself: happy birthday.	我也是 Me too.	你这是在说什么啊？ What are you talking about?	这是一个UNK This is a UNK	生日快乐，我的生日快乐 Happy birthday, happy birthday to me	生日快乐 Happy birthday	祝生日快乐，祝生日快乐 Happy birthday to you, happy birthday to you

Table 4: Case study.

5 Conclusion

In this paper, we view the generation diversity from a new angle: the latent vectors. Base on it, we propose an easy but effective learning framework named MEMD. In experiment, we found that the initialization method for the major-encoder and the major-decoder have great impact on the model's performance. Besides, we verify the effectiveness of the aux-encoder and the aux-decoder, and transfer the basic MEMD to MEMD-A and MEMD-M. We analyze the distribution of latent vectors, and find it consistent with evaluation metrics, which supports our conjecture that dispersion of latent vectors has positive correlation with diversity of generated responses.

Acknowledgements

This work is partially supported by the National High Technology Research and Development Program of China (Grant No. 2015AA015403) and the National Natural Science Foundation of China (Grant No. 61170091). We would also like to thank the anonymous reviewers for their helpful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 362--367.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724--1734.

Stephen Clark and Kris Cao. 2017. Latent variable dialogue models and their diversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 182--187.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1723--1732.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *CoRR*, abs/1408.6988.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110--119.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *CoRR*, abs/1511.06114.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 3349--3358.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 3776--3784.

Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron C. Courville. 2017a. Multiresolution recurrent neural networks: An application to dialogue response generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3288--3294.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017b. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3295--3301.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1577--1586.

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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 504--509.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *Computational Linguistics and Intelligent Text Processing, 6th International Conference, CICLing 2005, Mexico City, Mexico, February 13-19, 2005, Proceedings*, pages 341--351.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104--3112.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 496--505.

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 Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3351--3357.

Rui Yan, Dongyan Zhao, and Weinan E. 2017. Joint learning of response ranking and next utterance suggestion in human-computer conversation system. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 685--694.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 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 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 654--664.

Ganbin Zhou, Ping Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. 2017. Mechanism-aware neural machine for dialogue response generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3400--3407.