Static-Dynamically Attentive Variational Network for Dialogue Generation

Jinxin Chang*, Ruifang He*,†, Longbiao Wang†, Ruifang Wang,
Xiangyu Zhao, and Ting Yang

Tianjin Key Laboratory of Cognitive Computing and Application,
College of Intelligence and Computing, Tianjin University, Tianjin, China
{changjinxin,rfhe,longbiao_wang}@tju.edu.cn
{wrf276224255,zhaoxiangyu009,16622898776}@163.com

Abstract. Dialogue generation is a challenging task of giving a humanlike response according to the dialogue context. Recently, most studies treat it as a translation task and employ a sequence-to-sequence with attention framework to produce the responses. This framework is suitable for the translation since it can align semantics. However, in the dialogue task, it could not properly model the real response generation process. Actually, people usually produce a response based on the anticipatory core thought, which does not need the strict semantic alignment. Therefore, we propose a static-dynamically attentive variational network (SdavNet) to imitate the process of human thinking before response. Specifically, we design a novel static-dynamic attention that contains a static attention to globally distill a leitmotiv based on a learned latent variable, and a dynamic attention to assist the decoder in expanding the leitmotiv to a meaningful and diverse response from a local perspective. Meanwhile, the von Mises-Fisher (vMF) distribution is introduced to improve the KL-vanishing problem existed in the variational network. Experimental results on two large datasets demonstrate our model's potential in dialogue generation.

Keywords: Dialogue generation · Neural networks
Neural variational model · Static-dynamic attention mechanism
von Mises-Fisher distribution

1 Introduction

Intelligent chatbots aim to interact naturally and consistently with human regarding to a wide range of topics in open domains [1]. A widely adopted approach to build a chatbot is to train a sequence-to-sequence (Seq2Seq) with attention model [2–4] which is first introduced in neural machine translation (NTM) task [5, 6]. Specifically, before Seq2Seq producing the word at each time step, the attention mechanism will calculate a probability distribution over the input words, which tells the model what to focus on to generate the next word. In spite of its current success in NMT, the Seq2Seq with attention framework tends to favor the "safe" responses in dialogue generation task, which are uninformative and meaningless like "me too" or "I don't know" [7, 8].

There are several possible reasons for this problem and one of them is that the dialogue task is intrisically different from the translation task in terms of sentence (response or target language) generation process. In case of translation, since there exists a 1-to-1 relationship between every word in a language and its translation in another language as shown in Figure 1, conventional attention mechanism is suitable for this task by aligning the context at each time step. However, in the conversation scenario, the sentence expressed by human not only has diverse word and phrase representations, but also can convey a clear core thought which is semantically relevant and coherent with the context history. It is mainly attributed to the way of human thinking where we often first come up with a leitmotiv for the response based on the dialogue context, and then reorganize it into the meaningful and diverse response. Yet this translation-based framework could not suitably imitate this process.

Previous methods addressing the "safe response" problem can be broadly divided into two categories: (1) the first category attempts to optimize the Seq2Seq architecture itself, including improving objective function with mutual information [8], highlighting the important parts of the context by a hierarchical attention mechanism [17], introducing the latent variables as global context structure [10–12], etc. (2) the second category encourages the specific responses by introducing the external resources, such as topic words [13] and knowledge base [14]. These methods seem to be great successes in dialogue generation, yet they still have insufficiency in generalization of core thoughts of responses.

In this work, inspired by the way of human thinking, we propose a novel **S**tatic-**D**ynamically **A**ttentive **V**ariational **N**etwork (**SdavNet**) to improve the quality of generated responses. The key idea of SdavNet is to first distill a leitmotiv vector for the predicted response and then expand it to a high-quality response, which

* Equal contribution.
†Corresponding author.

Fig. 1. A translation example from NIST Chinese-English translation tasks [9].

Fig. 2. The SdavNet framework during testing.

is mainly accomplished by the designed static-dynamic attention. In detail, our model contains three parts: a context understanding module, a leitmotiv distillation module and a leitmotiv-based response generation module. The understanding module first encodes the dialogue history into a history vector, based on which the distillation module strives to learn a vMF-distributed latent variable as the holistic structure information to assist the static attention to extract a useful leitmotiv vector. Afterwards the learned history vector, latent variable and leitmotiv vector are fed into the generation module to produce the diverse response with the help of the dynamic attention.

The main contributions of this paper are listed as follows:

- Propose a novel static-dynamically attentive variational network to generate the meaningful and diverse responses by imitating the way of human thinking in the conversation scenario.
- Introduce a vMF distribution to slove the KL-vanishing problem during optimizing the variational network, to our knowledge, our model is the first work to use vMF distribution in the dialogue generation task.
- The experimental results on two large conversation datasets validate the effectiveness of our model.

2 Related Work

Since the availability of large conversation corpora on the Internet recently, building a data-driven chatbot has become a research hotspot. There exist both retrieval-based methods [15, 16] and generation-based methods [2–4] for open domain chatbots. The generation-based methods have the advantages of producing the more flexible and extensible responses. A prevailing practice is to employ a Seq2Seq with attention model [2–4] like the NMT task, however, it tends to generate the dull and uninformative responses.

To tackle this problem, one line of research has focused on improving Seq2Seq model itself. For example, Li et al. [8] introduces the Maximum Mutual Information (MMI) as a term in loss function to penalize the safe responses. Besides this, Xing et al. [17] proposes a hierarchical attention mechanism to highlight the important parts of the context at different levels, while some other models [10–12] are proposed in order to introduce the latent variables into the generation process. On the other hand, many attempts have also been made to augment the input with richer external information. For example, Li et at. [18] encodes persona information into the distributed embeddings to control the speaker consistency. Xing et al. [13] extracts a set of topics based on Latent Dirichlet Allocation (LDA) to encourage more topic coherent responses, and Liu et al. [14] designs the facts matching and entity diffusion to enrich responses.

Although the previous methods improve the response quality to some extent, they neglect the generalization of the core thoughts of responses, which is a crucial step to maintain semantic consistency with the context history. In this work, we design a static-dynamic attention mechanism that assists the SdavNet model in first distilling a leitmotiv vector for response and then expanding it to an unabridged and fluent sentence. Notably, different from the existing dual attention architecture that contains two dynamic attentions [13, 17], our designed attention consists of a static attention and a dynamic attention, which respectively generalizes the core thought of the response globally and improves the diversity for each word locally. Furthermore, the effectiveness of vMF distribution has also been proven in many tasks, such as sentence editing [19] and document modeling [20].

3 The Proposed Method

3.1 Task Description and Model Overview

Given a dialogue context $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$ where $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, ..., x_{i,N_i})$ is an utterance with N_i words, our task is to generate a response $\boldsymbol{y} = (y_1, y_2, ..., y_m)$ that is semantically relevant and diverse. Motivated by the way of human thinking, we propose a static-dynamically attentive variational network (SdavNet) to first distill a leitmotiv and then expand it into a response. Essentially, the goal is to estimate the conditional probability:

$$P(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}) = P(\boldsymbol{z}|\boldsymbol{x})P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}) \tag{1}$$

where the latent variable $\boldsymbol{z}$ is utilized to capture the structure information of the context. $P(\boldsymbol{z}|\boldsymbol{x})$, parameterzed as the prior network, stands for the true prior distribution of $\boldsymbol{z}$. And $P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}) = \prod_{i=1}^{m} p_{vocab}(y_i)$ denotes the probability of generating $\boldsymbol{y}$, which is implemented by a generation module in our model.

Figure 2 gives the architecture of SdavNet, which can be divided into three modules. The first module is context understanding module that encodes the context in a hierarchical setting. To obtain a suitable core thought for the response, we devise the leitmotiv distillation module to learn a vMF-based latent variable $\boldsymbol{z}$, which is then fed into the static attention to produce a leitmotiv vector. Finally, the leitmotiv-based response generation module is designed to generate the high-quality responses conditioned on the outputs of the above two modules.

3.2 Context Understanding

To better understand the dialogue context, we employ a hierarchical encoder to obtain the hidden representations of the input, which contains a utterance encoder and a context encoder. The utterance encoder is implemented by bidirectional GRU [6], including a forward network f_{GRU}^{fw} and a backward network f_{GRU}^{bw}. It takes in words sequence of $\boldsymbol{x}_i$ and computes the utterance vector h_i as:

$$[h_{i,1}^{fw}, h_{i,2}^{fw}, ..., h_{i,N_i}^{fw}] = f_{GRU}^{fw}(\boldsymbol{x}_i) \tag{2}$$

$$[h_{i,1}^{bw}, h_{i,2}^{bw}, ..., h_{i,N_i}^{bw}] = f_{GRU}^{bw}(\boldsymbol{x}_i) \tag{3}$$

$$h_{i,j} = [h_{i,j}^{fw}, h_{i,N_i-j+1}^{bw}] \tag{4}$$

$$h_i = h_{i,N_i} \tag{5}$$

where $h_{i,j}^{fw}$ and $h_{i,j}^{bw}$ denote the hidden states of forward GRU and backward GRU respectively for the j-th word in $\boldsymbol{x}_i$. $h_{i,j}$ stands for the state vector of the j-th word in $\boldsymbol{x}_i$. h_i represents the summary of the utterance $\boldsymbol{x}_i$. Similarly, we can also obtain the utterance vector h_y of the response $\boldsymbol{y}$.

After the $\boldsymbol{x}_n$ is processed by the utterance encoder, the context encoder, an unidirectional GRU network f_{GRU}^{uni}, reads $h_1, h_2, ..., h_n$ in their order and calculates the hidden vectors as:

$$[l_1, l_2, ..., l_n] = f_{GRU}^{uni}(h_1, h_2, ..., h_n) \tag{6}$$

where the last hidden vector l_n is treated as the history vector representing the whole dialogue context history.

3.3 Leitmotiv Distillation

This module is designed to extract and distill a suitable leitmotiv that represents the core thought of the response. It consists of two parts: a vMF-based recognition/prior network component and a static attention component.

3.3.1 vMF-based Recongniton/Prior Network

On top of the context encoder, we introduce a recognition network $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})$ and a prior network $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ to sample latent variables during training and testing respectively. To slove the KL-vanishing problem in the neural variational model, we employ vMF as the distribution of latent variables. Thus we set the recogniton network $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}) \sim vMF(\mu_{re}, \kappa_{re})$ and the prior network $p_\theta(\boldsymbol{z}|\boldsymbol{x}) \sim vMF(\mu_{prior}, \kappa_{prior})$. Next, we will describe the vMF distribution, introduce the KL divergence and give a sampling technique for it.

vMF Distribution The von Mises-Fisher (vMF) distribution is defined in a hyperspherical latent space, parameterized by a direction vector $\mu \in \mathbb{R}^d$ with $\|\mu\| = 1$ indicating the mean direction and a concentration parameter $\kappa \in \mathbb{R}_{\geq 0}$. The PDF of the vMF distribution for a unit vector $\boldsymbol{z} \in \mathbb{R}^d$ is defined as:

$$f_m(\boldsymbol{z}; \mu, \kappa) = \mathbb{C}_d(\kappa) \exp(\kappa \mu^T \boldsymbol{z}) \tag{7}$$

$$\mathbb{C}_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)} \tag{8}$$

where $\mathbb{C}_d(\kappa)$ is the normalization constant, and I_ρ stands for the modified Bessel function of the first kind at order ρ.

In our model, we set $\kappa_{re}, \kappa_{prior}$ as a constant and compute μ_{re}, μ_{prior} as:

$$\tilde{\mu_{re}} = f_{re}([l_n, h_y]) \tag{9}$$

$$\mu_{re} = \tilde{\mu_{re}} / \|\tilde{\mu_{re}}\| \tag{10}$$

$$\tilde{\mu_{prior}} = f_{prior}(l_n) \tag{11}$$

$$\mu_{prior} = \tilde{\mu_{prior}} / \|\tilde{\mu_{prior}}\| \tag{12}$$

where f_{re} and f_{prior} are two linear transformations, and $\|\cdot\|$ denotes 2-norm to ensure the normalization.

KL Divergence Different from previous work [19, 20], without $vMF(\cdot, 0)$ as the prior, the KL divergence needs to be recalculated as:

$$\begin{aligned}\mathcal{L}_1 &= KL(q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})\|p_\theta(\boldsymbol{z}|\boldsymbol{x})) \\ &= KL(vMF(\mu_{re},\kappa_{re})\|vMF(\mu_{prior},\kappa_{prior})) \\ &= (d/2-1)\log\frac{\kappa_{re}}{\kappa_{prior}} - \kappa_{prior}\mu_{prior}\mu_{re}^{-1}\frac{I_{d/2}\kappa_{re}}{I_{d/2-1}\kappa_{re}} \\ &\quad + \log\frac{I_{d/2-1}\kappa_{prior}}{I_{d/2-1}\kappa_{re}} + \kappa_{re}\frac{I_{d/2}\kappa_{re}}{I_{d/2-1}\kappa_{re}}\end{aligned} \tag{13}$$

Sampling Technique for vMF We utilize the rejection sampling scheme [21] to sample a value $w \in [-1, 1]$, then derive a unit vector tangent υ at the mean vector μ on the surface of the hypersphere. Based on these, our latent variable $\boldsymbol{z}$ can be given by $\boldsymbol{z} = w\mu + \upsilon\sqrt{1-w^2}$.

3.3.2 Static Attention for Globally Leitmotiv Distilling

We design the static attention component to obtain a stable core thought for the response. Different from the traditional attention mechanism, the static attention in leitmotiv distillation module depends on both the hidden vectors of the context level encoder and the sampled global structure information $\boldsymbol{z}$, and only operates once when generating a response. Specifically, it calculates a weight vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n)$ for $\{l_i\}_{i=1}^n$, which is computed as:

$$e_i = v_{sta}^T \tanh(W_l l_i + W_z \boldsymbol{z} + b_{sta}) \tag{14}$$

$$\boldsymbol{\alpha} = softmax(\boldsymbol{e}) \tag{15}$$

where v_{sta}, W_l, W_z and b_{sta} are learnable neural network parameters. This weight vector can find the important utterances which make more contributions to the distilled core thought. Then a leitmotiv vector that represents the core thought of the response can be obtained by:

$$c_{sta} = \textstyle\sum_i \alpha_i h_i \tag{16}$$

3.4 Leitmotiv-based Response Generation

The leitmotiv-based response generation module builds on a deocder with a dynamic attention, which is responsible to generate semantically relevant and diverse responses. At the begining of generation, the decoder receives the history vector l_n, global structure information $\boldsymbol{z}$ and leitmotiv vector c_{sta} as the initial hidden state. Thereafter, at each step t, the decoder processes the word embedding of the previous word (while training, this is y_{t-1} in the ground truth; at test time it is the previous word emitted by the decoder), and produces a decoder state s_t which is used to determine the context vector c_{dyn}^t.

3.4.1 Dynamic Attention for Locally Grasping the Diversity

We employ the attention proposed by Bahdanau et al. [6], but apply it on the output of the utterance encoder to obtain the context vector c_{dyn}^t at the step t:

$$d_i^t = v_{dyn}^T \tanh(W_{h'} h_i' + W_s s_t + b_{dyn}) \tag{17}$$

$$\boldsymbol{\beta}^t = softmax(\boldsymbol{d}^t) \tag{18}$$

$$c_{dyn}^t = \textstyle\sum_i \beta_i^t h_i' \tag{19}$$

where h_i' stands for the hidden vector of the i-th word in whole context; $\boldsymbol{\beta}^t$ is the dynamic attention distribution at the time step t; $v_{dyn}, W_{h'}, W_s$ and b_{dyn} are learnable parameters. The learned context vector c_{dyn}^t can tell the decoder what to focus on when generating the next word y_t.

3.4.2 Leitmotiv-driven Decoding

After obtaining the leitmotiv vector c_{sta} and the context vector c_{dyn}^t, we concatenate them with the decoder state s_t, then they are used to calculate the vocabulary distribution p_{vocab} through two linear layers:

$$p_{vocab} = softmax(V'(V[s_t, c_{sta}, c_{dyn}^t] + b) + b') \tag{20}$$

where V, V', b and b are learnable parameters. p_{vocab} stands for the probability distribution over the whole vocabulary, and the probability to generate word y_t is $p_{vocab}(y_t)$.

Table 1. Statistical information of Cornell and Ubuntu.

Corpus	Train	Valid	Test	Avg. Utterances	Avg. Words	Vocab	Coverage
Cornell	135155	1393	1193	3.93	16.7	10000	98.18%
Ubuntu	448833	19584	18920	4.94	23.67	20000	99.12%

The loss function of the decoder is given by:

$$\begin{aligned}\mathcal{L}_2 &= -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\log p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})] \\ &= -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\sum_{i=1}^{m} \log p_{vocab}(y_i)]\end{aligned} \tag{21}$$

Therefore, the overall loss function $\mathcal{L}$ of SdavNet is:

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 \tag{22}$$

4 Experiments

4.1 Experiment Configuration

Datasets We conduct the experiments mainly on *Cornell Movie Dialogs Corpus*[1] [22] and *Ubuntu Dialogue Corpus*[2] [23]. The Cornell dataset contains more than 80,000 imagined movie dialogues, in which we remove the ungrammatical characters and divide the original dialogues into consecutive 2-10 utterances. The Ubuntu dataset contains about 500,000 dialogues collected from the Ubuntu Internet Relayed Chat channel, each of which starts with a Ubuntu-related technical problem and follows by the corresponding responses about solutions.

In the above two datasets, the last utterance in a conversation is regarded as the response and the remaining ones are the input context. The detailed statistical information is shown in Table 1.

Implementation Our model is implemented based on Tensorflow and trained on an Nvidia GPU card. We set word embeddings to size of 200 and initialize them randomly. The hidden sizes of all GRUs in encoders and decoder are 500. The latent variable $\boldsymbol{z}$ sampled from vMF has a size of 100. Furthermore, we employ the Adam optimizer [24] to tune our model with a learning rate of 0.001. The batch size is set to 30. We select the best models based on the early-stop strategy [25] on the validation set and report their performances on the test set.

Baselines We compare the SdavNet with several state-of-the-art models: S2SA [4], HRED [7], VHRED [10] and HVMN [12].

4.2 Evaluation Metrics

We analyze the model's performance in terms of the following metrics:

Embedding Based Metrics We employ three embedding based metrics (Average, Greedy, Extreme)[3] [26], which can measure the semantic relevance between generated responses and ground truths.

Distinct-1 & Distinct-2 Following Li et al. [8], we calculate the numbers of distinct unigrams and bigrams in the generated responses, then divide total unigrams and bigrams respectively. The higher value indicates that the responses are more diverse.

Human Evaluation In addition to the automatic mertics, we also recruit three human annotators to judge the quality of the generated responses. We show them 300 contexts with responses generated by each model. Each response is rated according to the following criteria: **1**: the response is not grammatical or semantically relevant; **2**: the response is grammatical and weakly related, but might not be informative enough (e.g., "I don't know"); **3**: the response is not only grammatical and semantically relevant, but also informative and interesting. The agreements among annotators are calculated with Fleiss' kappa [27].

4.3 Evaluation Results

Automatic Evaluation Results Table 2 shows all metric-based evaluation results of our model and baselines on two datasets. From the results, we can observe that: (1) HVMN can obtain the best performance among the

[1] The dataset is available at `https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html`.

[2] We use the same train-validation-test split as in [12].

[3] We use the embeddings trained on Google News Corpus: `https://code.google.com/archive/p/word2vec/`.

Table 2. Comparisons between our SdavNet and the baselines under the automatic evaluation.

Model	Average	Greedy	Extreme	Distinct-1	Distinct-2
Cornell					
S2SA	0.1902	0.1439	0.1305	0.46%	1.33%
HRED	0.4586	0.3215	0.2681	1.30%	3.29%
VHRED	0.4630	0.3234	0.2734	1.41%	4.37%
HVMN	0.5063	0.3711	0.3033	2.15%	9.70%
SdavNet	**0.5972**	**0.5161**	**0.3615**	**3.52%**	**19.64%**
Ubuntu					
S2SA	0.2156	0.1688	0.1265	0.47%	2.60%
HRED	0.5415	0.4117	0.31933	0.84%	4.09%
VHRED	0.5341	0.4027	0.3062	1.51%	7.72%
HVMN	0.5584	0.4229	0.3220	1.93%	10.05%
SdavNet	**0.5985**	**0.4388**	**0.3225**	**2.29%**	**13.21%**

Table 3. Human evaluation results on Cornell dataset.

Model	score distribution			Kappa
	1	2	3	
S2SA	48.77%	44.01%	7.22%	0.44
HRED	45.55%	39.57%	14.88%	0.43
VHRED	41.44%	33.68%	24.88%	0.46
HVMN	39.11%	30.45%	30.44%	0.45
SdavNet	**36.44%**	**25.12%**	**38.44%**	0.40

baselines, whereas our model outperforms all baselines, which verifies the effects of the static-dynamic attention and vMF distribution. (2) In terms of three embedding based metrics, VHRED performs better than HRED on Cornell, while HRED has a better performance than VHRED on Ubuntu, which indicates that the latent variables have a more obvious effect on the open-domain datasets.

Human Evaluation Results Table 3 shows the human evaluation results on Cornell data, in which the score distribution values represent the percentages of responses belonging to each category. From the results, we can observe that: (1) SdavNet generates much more informative and interesting responses (i.e., 38.44% labeled as '3') and much less general responses (i.e., 25.12% labeled as '2') than all the baselines. The results verify the effectiveness of our model again. (2) The higher kappa scores demonstrate that the annotators come to a fair agreement. Meanwhile, the t-test we conduct indicates that the improvements of SdavNet are statistically significant (p-value$<$0.01).

4.4 Discussions

Model Ablation To examine the effectiveness of the static and dynamic attentions, we conduct two individual experiments by removing one of the attentions each time. From the results presented in Table 4, we have the following observations: (1) Removing the static (denoted as SdavNet$_{dyn}$) or dynamic (denoted as SdavNet$_{sta}$) attention makes the performances drop, which indicates that both the static attention and the dynamic attention are indispensable to improve SdavNet's performance; (2) Interestingly, SdavNet$_{dyn}$ obtains the highest distinct-1 and distinct-2 values but the lowest embedding based metric values among these three models. The results demonstrate that the dynamic attention can dominate the decoder to generate different words by aligning the context, but the generated responses are semantically irrelevant due to the lack of the guidance of core thoughts generalized by the static attention.

The Effect of vMF on KL The vMF-based KL term in VAE is a constant [19, 20], but the one in CVAE (Eq.(13)) depends on the learned μ_{prior} and μ_{re}, thus we conduct a set of experiments with different model setups (SdavNet with Gaussian or vMF distribution) to verify the effectiveness of vMF distribution in sloving the KL-vanishing problem. Figure 3 visualizes the evolution of the KL cost during training. We can see that for the Gaussian case, the KL value decreases from 68.33 to 15.75 and never recovers. On the contrary, the vMF-based KL cost is relatively stable around 41, which experimentally demonstrates that the variable μ_{prior} and μ_{re} have little effect on KL. Therefore, the vMF is a better selection to slove the KL-vanishing in the CVAE framework.

Table 4. Performances of model ablation on Cornell dataset.

Model	Average	Greedy	Extreme	Distinct-1	Distinct-2
SdavNet$_{dyn}$	0.3358	0.2116	0.2004	**10.82%**	**40.69%**
SdavNet$_{sta}$	0.5519	0.5138	0.3470	2.08%	10.33%
SdavNet	**0.5972**	**0.5161**	**0.3615**	3.52%	19.64%

Fig. 3. The value of the KL divergence during training with different setups on Cornell.

5 Conclusion

In this work, we propose a static-dynamically attentive variational network (SdavNet) for dialogue generation. Different from the traditional models, the proposed model first distills a leitmotiv that represents the core thought of the response, and then expands it to a complete response. As for the KL-vanishing problem in the variational models, we introduce the vMF distribution to improve the training procedure. Experimental results show that our model outperforms the competitive baseline models significantly, meanwhile generates semantically relevant and informative responses.

Acknowledgments. We thank the anonymous reviewers for their valuable feedback. Our work is supported by National Natural Science Foundation of China (61976154), Tianjin Natural Science Foundation (18JCYBJC15500), National Natural Science Foundation of China (61771333) and Tianjin Municipal Science and Technology Project (18ZXZNGX00330).

References

1. Perez-Marin, D.: Conversational agents and natural language interaction: Techniques and effective practices. IGI Global (2011)
2. Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1577–1586 (2015)
3. Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.Y., Gao, J., Dolan, B.: A neural network approach to context-sensitive generation of conversational responses. In: Proceedings of NAACL-HLT, pp. 196–205 (2015)
4. Vinyals, O., V. Le, Q.: A neural conversational model. In: Proceedings of the International Conference on Machine Learning, Deep Learning Workshop (2015)
5. Sutskever, I., Vinyals, O., V Le, Q.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27 (NIPS), pp. 3104–3112 (2014)
6. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2014)
7. Serban, I.V., Sordoni, A., Bengio, Y., Courville, A.C., Pineau, J.: Building endto-end dialogue systems using generative hierarchical neural network models. In: AAAI-16, pp. 3776–3784 (2016)
8. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. In: Proceedings of NAACL-HLT, pp. 110–119 (2016)
9. Kuang, S., Xiong, D., Luo, W., Zhou, G.: Modeling coherence for neural machine translation with dynamic and topic caches. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 596–606 (2018)
10. Serban, I., Sordoni, A., Lowe, R.J., Charlin, L., Pineau, J., Courville, A.C., Bengio, Y.: A hierarchical latent variable encoder-decoder model for generating dialogues. In: AAAI-17 (2017)
11. Zhao, T., Zhao, R., Eskenazi, M.: 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), pp. 654–664 (2017)
12. Chen, H., Ren, Z., Tang, J., Zhao, Y.E., Yin, D.: Hierarchical variational memory network for dialogue generation. In: Proceedings of the 2018 World Wide Web Conference (WWW) (2018)
13. Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., Ma, W.Y.: Topic aware neural response generation. In: AAAI-17, pp. 3351–3357 (2017)
14. Liu, S., Chen, H., Ren, Z., Feng, Y., Liu, Q., Yin, D.: Knowledge diffusion for neural dialogue generation. In: ACL, pp. 1–10 (2018)
15. Wu, Y., Wu, W., Xing, C., Zhou, M., Li, Z.: Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In: ACL, pp. 496–505 (2017)
16. Yan, R., Zhao, D., E, W.: Joint learning of response ranking and next utterance suggestion in human-computer conversation system. In: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 685–694. ACM (2017)
17. Xing, C., Wu, W., Wu, Y., Zhou, M., Ma, W.Y.: Hierarchical recurrent attention network for response generation. In: AAAI-18 (2018)

18. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A persona-based neural conversation model. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 994–1003 (2016)
19. Guu, K., Hashimoto, T.B., Oren, Y., Liang, P.: Generating sentences by editing prototypes. Transactions of the Association for Computational Linguistics (TACL). 6, 437–450 (2018)
20. Xu, J., Durrett, G.: Spherical latent spaces for stable variational autoencoders. In: Proceedings of the 2018 Conference on EMNLP, pp. 4503–4513 (2018)
21. Wood, A.: Simulation of the von mises fisher distribution. Communications in Statistics Simulation and Computation. 23(1), 157–164 (1994)
22. Danescu-Niculescu-Mizil, C., Lee, L.: Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In: Proceedings of the 2nd Workshop on (CMCL '11), pp. 76–87 (2011)
23. Lowe, R., Pow, N., Serban, I.V., Pineau, J.: The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pp. 285–294 (2015)
24. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
25. Caruana, R., Lawrence, S., Giles, C.L.: Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In: Advances in (NIPS), pp. 402–408 (2000)
26. Liu, C.W., Lowe, R., Serban, I.V., Noseworthy, M., Charlin, L., Pineau, J.: 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 EMNLP, pp. 2122–2132 (2016)
27. Fleiss, J.L., Cohen, J.: The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. Educational and psychological measurement. 33(3), 613–619 (1973)