Learning Personalized End-to-End task-Oriented Dialogue Generation

Bowen Zhang[1], Xiaofei Xu[1], Xutao Li[2], Yunming Ye[2(✉)], Xiaojun Chen[3], and Lianjie Sun[2]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[2] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China
[3] College of Computer Science and Software, Shenzhen University, Shenzhen, China
yeyunming@hit.edu.cn

Abstract. Building personalized task-oriented dialogue system is an important but challenging task. Significant success has been achieved by selecting the responses from the pre-defined template. However, preparing massive response template is time-consuming and human-labor intensive. In this paper, we propose an end-to-end framework based on the memory networks for responses generation in the personalized task-oriented dialog system. The static attention mechanism is used to encode the user-conversation relationship to form a global vector representation, and the dynamic attention mechanism is used to obtain import local information during the decoding phase. In addition, we propose a gating mechanism to incorporate user information into the network to enhance the personalized ability of the response. Experiments on the benchmark dataset show that our model achieves better performance than the strong baseline methods in personalized task-oriented dialogue generation.

Keywords: Dialogue Generation · Task-oriented Dialogue System · Personalized Response

1 Introduction

Task-oriented dialogue systems have become increasingly important in a variety of applications, such as reservation systems or navigation inquiry systems [1]. Earlier efforts in task-oriented dialogue systems are composed of pipeline structures (e.g., language understanding, dialogue management and language generation), where each module is designed separately and heavily relies on handcrafted rules [2, 3]. Inspired by the recent success of sequence-to-sequence(seq2seq) encoder-decoder model in language generation, the end-to-end dialogue systems, which input the dialogue history and directly output system responses, have shown promising results based on recurrent neural networks(RNN)[4] and memory networks [5, 6].

Although the encoder-decoder networks have made great success in task-oriented dialogue systems, the methods only generate responses based on the dialogue history, and cannot accommodate users with different personalities [7]. Therefore the response of the system is dull and fails to adjust the strategy of the conversation according to the personalized information.

The personalized task-oriented dialogue system is designed to generate responses that are more user-friendly and to help users complete conversations faster than non-personalized conversation systems [8]. In general, the personalized dialogue system can extract the requirement of the user during multi-turn interactions and then utilize personalized information to speed up the interaction process. Arguably, personalization drives the task-oriented dialogue system closer to the user's actual information needs [9]. Significant improvements have been achieved in the personalized system by using deep memory network with copy mechanism [10]. Joshi et al.[7] and Luo et al. [11] utilize the memory network to encode user information and conversation history to construct an end-to-end personalized task-oriented dialogue model. Compared with the RNN encoder, the memory network can effectively store long-term conversation history. Despite the effectiveness of the above methods, the personalized dialogue system remains considerable challenges for several reasons: (1) The performance of the previous methods is based on the selection of the numerous manual predefined responses template, which is essentially a multi-label classification problem and heavily relies on hand-crafted features [5]. (2) For the previous method with the copy mechanism, the only information sent to the decoder is the global hidden state of the encoder [12]. However, Bahdanau et al. [13] reveal that the performance of text generation decreases rapidly as the length of the input sentence grows, if only the global hidden vector is utilized.

To alleviate the aforementioned challenges, in this paper, we designed an end-to-end memory network with a static and dynamic attention mechanism that can generate personalized responses, instead of selecting from predefined templates. The proposed method also works in an encoder-decoder framework. The encoder is a memory network and trainable user profile embeddings are utilized as a query to form the global hidden state of dialogue. The way to form dialogue representation is named as static attention mechanism. The decoder is composed of an RNN and a memory network, accounting for generating personalized responses. The RNN part will generate a dynamic query to the memory network, and the memory network part utilizes the dynamic query with a carefully designed gating strategy to form a local representation, which will be the input of RNN in the next time stamp. The way to produce a local representation is termed as dynamic attention mechanism. The contributions of the paper are summarized as follows:

1. We propose a novel framework for personalized task-oriented dialogue generation scenario. (1) In the encoding phase, the static attention mechanism can learn the relationship between dialogue and user information to adequately represent the global representation of the dialogue history and knowledge

base. (2) The dynamic attention mechanism can trace the history of dialogue and important local features in the response generation stage.
2. As personalized task-oriented dialoguing needs meet two important objectives: (1) the responses are personalized so that people feel user-friendly; (2) the responses must solve the user requirement. To nicely integrate the two objectives, we propose a gating strategy in the decoder stage.
3. Extensive experiments are carried out on the personalized bAbI dialog dataset and the results demonstrate the superiority of the proposed model over state-of-the-art competitors.

2 Related Work

End-to-end neural network methods to establish a personalized dialogue system has attracted a lot of research interest, which is widely accepted as being divided into task-oriented and non-task-oriented systems [9].

The seq2seq approach is very effective for building a personalized dialogue system. Many research works focus to make dialogue agents smarter by using user profiles. Li et al. [15] first proposed a persona-based model for dealing with user consistency in neural response generation. Speaker models are used to capture user characteristics such as background information and speaking style. The dyadic speaker addressee model captures the properties of the interaction between two interlocutors. Subsequently, research interest in personalized dialogue grew rapidly. Luan et al. [16] extend the user personalization model to multi-task learning. Yang et al. [9] proposed a method of using deep reinforcement learning to achieve user-specific conversation, which can generate object-coherence, informative and grammatical responses. Herzig et al. [10] proposed a response generation model that allows agents to respond to information about personality traits. Zhang et al. [19] use the key-value memory network to store the context information of conversations and users profile to implement personalized Dialogue Agents. These methods essentially pay more attention to personalization and user consistency. These methods can be divided into non-task-oriented dialogue(Chit-Chat) system, which the goal is to generate personalized responses based on user-specific information and to ensure consistency of user information during the conversation.

For personalized task-oriented dialogue systems, Joshi et al. [7] first proposed a personalized-BAbi dataset that is more user-friendly than traditional BAbi datasets and can speed up the dialogue process based on the user information(with recommendation ability). Among them, they proposed split-memory network, which uses two memory networks to separately model the conversation history and user information, and then concatenate them as input to the decoder. The network is effective, but simply concatenate the user vector with the global content vector sometimes it pays more attention to the personalized response and ignores the specific goals. Luo et al. [11] later improved this model, which can capture user preferences over knowledge base entities to handle the ambiguity in user requests. However, both methods are based on the response selection in the templates, which is essentially a multi-task classification problem. However, designing a template requires a lot of manual work, which is time-consuming and greatly reduces scalability.

3 Method

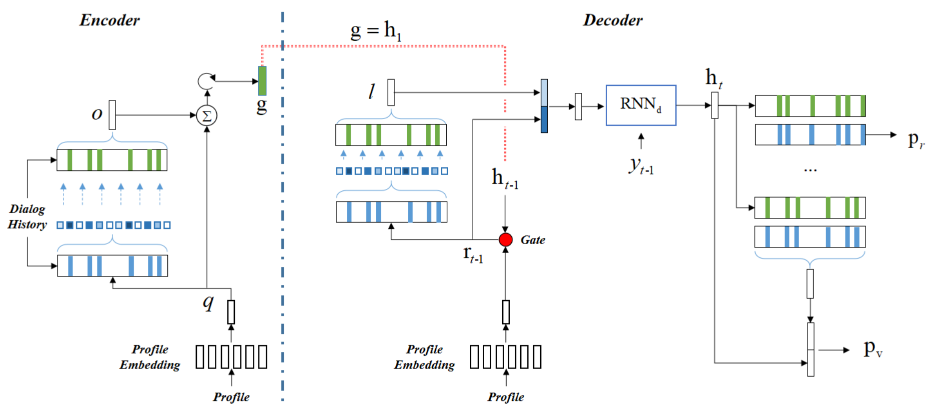

Fig. 1. The overall architecture of our model.

As depicted in Figure 1, our model is a variant of the Mem2Seq proposed in Madotto et al. [5]. Additionally, we propose a static multi-hop attention and dynamic attention to improve the performance of personalized task-oriented dialogue systems. To better understand our approach, in subsection 3.1, we first give the problem definition. Then, we expound our framework step by step in subsection 3.2. Finally, the training objective of the algorithm is given in subsection 3.3.

3.1 Problem Definition

We use $X = [x_1, x_2, \ldots, x_n]$ to denote the concatenation of multi-turn dialogue history with the current utterance of user, where n is the length of X. Similarly, we define the knowledge base tuples as $K = [k_1, k_2, \ldots, k_l]$. Each dialogue has a set of user-specific information and we concatenate them as $U = [u_1, \ldots, u_\ell]$, where u_i is the i-th feature in user profile and ℓ is the number of features. Specially, we further define $D = [K, X, \$]$ as a concatenation of two sets and \$ used as a sentinel. The goal of our model is to generate a response sequence $Y = [y_1, \ldots, y_m]$ when given D and U, where m is the length of Y.

3.2 Framework Structure

Our model uses a multi-hop attention-based Mem2Seq structure with copy mechanism as the backbone of seq2seq. It consists of two components: Memory Encoder and the Memory Decoder networks. The memory encoder network encodes the dialogue history and user information into a vector and sends them to the memory decoder, which then generates a response.

Memory Encoder Network Inspired by [20], memories consist of a set of trainable embedding matrices $\{C_1^u, C_1^r, \ldots, C_{t-1}^r, C_t^u\}$, where each C_j maps tokens from D to a embedding vector and u, r donate for user or agent utterances. It is well known that query as a reading head is very important in reading memory and obtaining global content information. Thus we expect to fuse personalized information in the query so that the user information can be effectively merged into the global representation vector and the reading pointer. Therefore the **static attention mechanism** is proposed. Specifically, we donate M as the embedding represents for U, and query vector $\mathbf{q}$ is the average vector of M. Note that the dimension of $\mathbf{q}$ is the same C_j. Thus we can calculate the attention weights probability at hop h by:

$$p^h = softmax(\mathbf{q}^h \times C_i^h) \tag{1}$$

where $softmax(z_i) = e^{z_i} / \sum_j e^{z_j}$, and $\times$ represents the multiplication of vectors with each corresponding vector in a matrix. Then, the model reads out the memory by the weighted sum over C^h,

$$\mathbf{o}^h = \sum_i p_i^h C_i^h \tag{2}$$

in the next hop, the query is updated by using $\mathbf{q}^{h+1} = \mathbf{q}^h + \mathbf{o}^h$. Finally, the model obtains the global vector representation $\mathbf{g}$ by concatenating the last hop of $\mathbf{o}$ and $\mathbf{q}$:

$$\mathbf{g} = \mathbf{o} \oplus \mathbf{q} \tag{3}$$

where $\oplus$ is the concatenation operator. Note that $\mathbf{g}$ is the input for the first decoding step.

Memory Decoder Network Since the memory network stores dialogue history and knowledge base, the memory size is often very large. Using only one global context vector does not apply to response generation. Therefore, we propose **a dynamic attention mechanism** in which each generated token is obtained by important features in memories. In the decoder phase, we use the gated recurrent unit network (GRU) [21] to dynamically generate each query, which is then used as the pointer for reading the memory to select the tokens that need to be generated or copied. The response in personalized dialogue systems aims to achieve a personalized response while completing the goals. Thus we introduced a **gate mechanism** that allows the decoder to focus on user information for personalized responses, while focusing on contextual information when addressing requirements.

Specifically, for decoding y_t, the first step of the decoder tends to use $\mathbf{q}^1$ and hidden last state $\mathbf{h}_{t-1}$ to generate the dynamic attention vector $\mathbf{r}_{t-1}$ though the gate,

$$\begin{aligned} p_g &= \sigma(W_1\mathbf{q}^1 + \mathbf{b}_1) \\ \mathbf{r}_{t-1} &= \mathbf{h}_{t-1} + p_g \times \mathbf{q}^1 \end{aligned} \tag{4}$$

where W_1 and b_1 are trainable parameters. Subsequently, at each token generation stage, we used $\mathbf{r}_{t-1}$ to make a **dynamic attention** for the memories C to obtain vector representation $\boldsymbol{l}$.

In the second step, for generatie y_t, $\mathbf{h}_{t-1}$, $\boldsymbol{l}$ and y_{t-1} are sent to the GRU to generate the new $\mathbf{h}_t$:

$$\mathbf{v} = softmax(\mathbf{r}_{t-1} \times \mathbf{C}_i) \tag{5}$$

$$\boldsymbol{l} = \sum_i \mathbf{v}_i \mathbf{C}_i \tag{6}$$

$$\alpha = \sigma(W_2(\boldsymbol{l} \oplus \mathbf{r}^{t-1}) + b_2) \tag{7}$$

$$\mathbf{h}^t = GRU(M(y^{t-1}), \alpha) \tag{8}$$

where $M(y^{t-1})$ donates the embedding vector of y^{t-1}, and W_2, b_2 are trainable parameters. Next, we send the generated $\mathbf{h}^t$ as the query to the memory network in the decoder. On the one hand, it produces a probability distribution all over dialogue history and knowledge, and on the other hand it can generate a distribution based on the word corpus. Thus, we can implement the generation and copying of tokens. Specifically, inspired by [5], we take the multiplication probability of the first hop in the memory network as the pointer distribution p_r. Next, we can acquire $\mathbf{o}^d$ as the content vector in the decoder, which similar to Eq.2. The probability of generating tokens p_v are obtained by passing the content vector $\mathbf{o}^d$ and hidden state $\mathbf{h}^j$ through a fully connected layer,

$$p_v = softmax(W_3(\mathbf{h}^j \oplus \mathbf{o} + b_3)) \tag{9}$$

where W_3 and b_3 are trainable parameters.

3.3 Training Objective

The training objective of our method consists of two parts, which are the standard cross-entropy loss functions:

$$\zeta = \sum_{t=1}^{m} p(y_i)\log(p_v(\hat{y}_i)), \quad J = \sum_{t=1}^{m} p(y_i)\log(p_r(\hat{y}_i)) \tag{10}$$

where $p(y_i)$ and $p(\hat{y}_i)$ are the actual word distribution and the generative word distribution for the i-th word of the response. Overall, the final objective function is minimized by:

$$L = \zeta + J \tag{11}$$

4 Experimental Setup

4.1 Experimental Data

In this study, we conduct extensive experiments on the personalized bAbI dialogue corpus [7] to illustrate the effectiveness of our method. This is a multi-turn dialog corpus with personalized interactions that extends from the bAbI dialogue dataset [3]. It designs five separate tasks for the restaurant reservation task. We give a brief introduction to each task.

Personalization Task 1: Issuing API calls. The agent must ask questions to fill the missing fields of the user request and then generate the API-call correctly.

Personalization Task 2: Updating API calls. The agent must change the API call accordingly based on changes in user requirements.

Personalization Task 3: Displaying Options. Based on the user's request, the agent uses the API call to query the knowledge base and add the correct entity to the response. The robot must recommend the restaurant to the user based on the user profile to accomplish this task.

Personalization Task 4: Providing extra information. The user asks for information about the restaurant and based on his multiple needs, the robot must learn to retrieve the correct knowledge base entity from history and customize it to the user.

Personalization Task 5: Conducting full dialogues. This is a complete dialogue combining all aspects of tasks 1-4.

The personalized bAbI dialogue corpus contains two sets. The full data set contains 6000 dialogues, and the small data set contains 1000 dialogues.

4.2 Model Configurations

We give the implementation details of the model as follows: In all the experiments, for equivalent the size between query and memory cells, we set the same RNN hidden size and memory size between [64, 512]. the drop rate we set in the range [0.1-0.5], and use the random mask in memory network as the same setting in [12]. We choose h=$\{1,3,6\}$ hop to encode and decode the memory network, and use greedy search during the response generation. Other weight parameters are initialized by randomly sampling the values from the uniform distribution U(-0.01, 0.01). We initial other weight parameters by random sampling from a uniform distribution U (-0.01, 0.01). The model is trained using the Adam optimization algorithm with a batch size of 8 and a decay rate of [0.2-0.9] [22].

4.3 Baseline Methods

To fully validate the performance of the model, we compared several strong baselines in the task-oriented dialogue generation.

- **MemNN** [7]: This method proposes to use memory network to encode the content and user profiles, in which employs two network structures: (1)**MemNN-org** the user profile concatenate in the dialogue memories of the encoding stage. (2)**Mem2Seq-split** uses a split memory network to store the user information and concatenate the hidden vectors as the final output of the encoder. However, These methods generate the response by selecting the templates.
- **PMemN2N** [11]: This method essentially similar to the basic framework as MemNN, but it combines the dialogue style information of the same user attribute in the encoder, which enhances the model personalization ability.
- **Mem2Seq** [5]: It is an end-to-end differentiable model, which the encoder is the memory network and the decoder uses RNN to generate query and memory network to generate response tokens. Followed by [7], we further employ three models: **Mem2Seq-org**, **Mem2Seq-split** and **Mem2Seq-att** which uses the embedding vector as the query of memory.
- **GLMP**[12]: This model is a variant of Mem2Seq, including global and local encoder to share external knowledge. We add the user information in memory cells, which the same as **MemNN-org**.
- **Seq2Seq-att** [24]: This model is the basic seq2seq method that combines the attention and pointer mechanisms. This method is widely used in text generation tasks.

4.4 Evaluation Metrics

Per-response/dialogue Accuracy: Per-response is based on each turn of responses, while Per-dialogue is based on an entire multi-turn dialogue. It is correct only if the generated and actual responses are identical, which also can be considered a task completion rate. Since Bordes [7] and Luo [11] employ their models by selecting the response from predefined candidates, directly using this metric for evaluation is more challenging for our model. Therefore, we also use the **BLEU score** which commonly used in the tasked-oriented dialogue generation task [25] to verify the performance of our network.

5 Experimental Results

Table 1 shows the per-response results of the full and the small datasets respectively. Methods 1-3 are based on template selection, and 4-8 are existing start-of-the-art task-oriented dialogue generation models. Since the problem of the generation methods is far more challenging than the template selection methods, the two types of problems cannot be directly compared. Despite this, for tasks 1-4, our approach yielded the best results comparison for both generation and selection methods. One can find that our method is far superior to other comparison methods in tasks 3 and 4. For example, for task 3, our approach improves 10.27% and 11.55% compared to the most advanced template selection and generation methods on full and small datasets respectively. For the generation methods, our model gains 1.27% (0.82%) improvement for task 4 over

Table 1. Evaluation Results of Per-response Accuracy.

		Task 1	Task 2	Task 3	Task 4	Task 5	BLEU
	SMALL SET						
1	MemNN-org	98.87	99.93	58.71	57.17	77.74	-
2	MemNN-split	82.44	91.27	68.56	57.11	78.1	-
3	PMemN2N	99.93	99.95	71.52	80.79	88.07	-
4	Seq2Seq-att	98.21	95.74	70.13	78.82	76.15	84.99
5	Mem2Seq-org	98.54	97.83	70.31	89.73	80.22	91.99
6	Mem2Seq-split	98.53	97.92	71.25	90.11	80.38	92.67
7	Mem2Seq-att	99.67	99.89	72.99	91.07	82.91	94.24
8	GLMP	99.27	99.69	72.25	88.97	80.73	92.62
9	Ours	**99.99**	**100**	**77.38**	**92.34**	**83.89**	**96.23**
	FULL SET						
1	MemNN-org	99.83	99.99	58.94	57.17	85.10	-
2	MemNN-split	85.66	93.42	68.60	57.17	87.28	-
3	PMemN2N	99.91	99.94	71.43	81.56	95.33	-
4	Seq2Seq-att	99.42	98.82	71.78	87.73	80.41	89.23
5	Mem2Seq-org	99.88	99.87	72.13	89.91	82.19	94.23
6	Mem2Seq-split	99.92	99.90	73.64	89.80	82.38	94.11
7	Mem2Seq-att	99.96	99.98	74.18	91.01	85.39	96.20
8	GLMP	99.45	99.77	74.56	90.97	86.20	94.91
9	Ours	**100**	**100**	**78.94**	**91.83**	**87.26**	**97.98**

Mem2Seq-att (the best competitor) on the small (full) dataset. consequently, the improvement for recommending restaurants (task 3) and providing relevant information (task 4) according to the user information can prove that our approach can effectively utilize user information to achieve personalized responses and accomplish user goals. Task 5 is the synthesis of tasks 1-4, which is more complicated to evaluate. Therefore, we also give the BLEU evaluation, which commonly used in the dialogue generation methods to prove the effectiveness of our model. Compared to generating problems, our method achieves the highest score in both accuracy and BLEU evaluation. For example, our method obtains 87.26% in per-response accuracy and 97.98% in BLEU on the full dataset, which in general, much higher than those of other baselines.

Table 2. Evaluation Results of Per-dialogue Accuracy.

	Task 1	Task 2	Task 3	Task 4	Task 5
Seq2Seq-att	87.2	97.0	3.7	66.7	1.2
Mem2Seq-org	97.1	97.9	6.7	70.5	2.6
Mem2Seq-split	98.3	97.6	7.4	69.9	3.3
Mem2Seq-att	99.3	99.9	8.4	70.9	5.2
Ours	100	100	8.7	71.6	5.6

To further investigate the performance of the proposed method, following [5], we employ per-dialogue accuracy compare with baselines on the small dataset. As we can see from Table 2, our method achieves best per-dialogue accuracy. Note that the Seq2Seq-att model performs poorly on per-dialogue evaluation compared to the methods of the memory-based network(rows 2-4), especially on tasks 3 and 5. This is due to the weak ability of Seq2Seq-att for knowledge base query, and it is inefficient for encoding long dialogue history based on the RNN approach. The mechanism of the memory network can effectively query the knowledge and represent the dialogue history.

5.1 Ablation Study

Table 3. Ablation Study.

	Task 1	Task 2	Task 3	Task 4	Task 5
Ours	100	100	8.7	71.6	5.6
w/s	99.9	99.9	5.9	70.9	3.7
w/d	99.4	99.9	8.1	71.2	5.3
w/g	99.9	99.9	8.5	71.3	5.4

In order to investigate the effects of each part, we perform the ablation test on the small dataset that discarding the static attention mechanism (denoted as w/s), the dynamic attention mechanism (denoted as w/d) and the user information gate mechanism (denoted as w/g). Note that for the method without static attention mechanism, we randomly initialize the query of the memory encoder and store the user information in memory cells.

We summaries the per-dialogue results in Table 3. From the results, we can observe that all the proposed components have a significant impact on our model. After discarding the two attention mechanisms, the performance of the model declined significantly, especially the static method. This is our expectation because the static attention captures the context of inter-relation between user and dialogue while coding the context, while the dynamic attention can help to obtain information about important local contexts for decoding. In addition, the user-guided gating mechanism also helps to improve the effectiveness of the model. In summary, the best performance of all experiments can be achieved by combining all factors.

6 Conclusion and Future Work

In this paper, we introduce a novel end-to-end personalization model in task-oriented dialog generation. Experimental results on a benchmark dataset and further analysis indicated that our method considers and alleviates to some extent the aforementioned challenges.

In the future, we plan to extend the personalized task-oriented dialogue system to cross-domain task, which can reduce labor costs and closer to actual needs.

7 Acknowledgement

This research was supported in part by NSFC under Grant Nos. No.U1836107, 61572158 and 61602132.

References

1. Wen T H, Vandyke D, Mrksic N, et al. A network-based end-to-end trainable task-oriented dialogue system[J]. arXiv preprint arXiv:1604.04562, 2016.
2. Chen H, Liu X, Yin D, et al. A survey on dialogue systems: Recent advances and new frontiers[J]. ACM SIGKDD Explorations Newsletter, 2017, 19(2): 25-35.
3. Bordes A, Boureau Y L, Weston J. Learning end-to-end goal-oriented dialog[J]. arXiv preprint arXiv:1605.07683, 2016.
4. Lei, W., Jin, X., Kan, M. Y., Ren, Z., He, X., and Yin, D. (2018, July). Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1437-1447).
5. Madotto, Andrea, Chien-Sheng Wu, and Pascale Fung. "Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems." arXiv preprint arXiv:1804.08217 (2018).
6. Wu, C. S., Madotto, A., Winata, G., and Fung, P. "End-to-end recurrent entity network for entity-value independent goal-oriented dialog learning." Dialog System Technology Challenges Workshop, DSTC6. 2017.
7. Joshi, Chaitanya K., Fei Mi, and Boi Faltings. "Personalization in goal-oriented dialog." arXiv preprint arXiv:1706.07503 (2017).
8. Mo, K., Zhang, Y., Li, S., Li, J., and Yang, Q. Personalizing a dialogue system with transfer reinforcement learning[C]//Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
9. Yang, M., Qu, Q., Lei, K., Zhu, J., Zhao, Z., Chen, X., and Huang, J. Z "Investigating Deep Reinforcement Learning Techniques in Personalized Dialogue Generation." Proceedings of the 2018 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2018.
10. Herzig, J., Shmueli-Scheuer, M., Sandbank, T., and Konopnicki, D. "Neural response generation for customer service based on personality traits." Proceedings of the 10th International Conference on Natural Language Generation. 2017.
11. Luo, L., Huang, W., Zeng, Q., Nie, Z., and Sun, X. "Learning Personalized End-to-End Goal-Oriented Dialog." arXiv preprint arXiv:1811.04604 (2018).
12. Wu, Chien-Sheng, Richard Socher, and Caiming Xiong. "Global-to-local Memory Pointer Networks for Task-Oriented Dialogue." arXiv preprint arXiv:1901.04713 (2019).
13. Cho, K., Van Merrinboer, B., Bahdanau, D., and Bengio, Y. "On the Properties of Neural Machine Translation: EncoderDecoder Approaches." Syntax, Semantics and Structure in Statistical Translation (2014): 103.

14. Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
15. Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., and Dolan, B. "A Persona-Based Neural Conversation Model." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vol. 1. 2016.
16. Luan, Y., Brockett, C., Dolan, B., Gao, J., and Galley, M. Multi-Task Learning for Speaker-Role Adaptation in Neural Conversation Models[C]//Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2017: 605-614.
17. Yang, M., Tu, W., Qu, Q., Zhao, Z., Chen, X., and Zhu, J. "Personalized response generation by Dual-learning based domain adaptation." Neural Networks 103 (2018): 72-82.
18. Mo, K., Li, S., Zhang, Y., Li, J., and Yang, Q. Personalizing a dialogue system with transfer learning[J]. arXiv preprint arXiv:1610.02891, 2016.
19. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. Personalizing Dialogue Agents: I have a dog, do you have pets too?[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 2204-2213.
20. Sukhbaatar, Sainbayar, Jason Weston, and Rob Fergus. "End-to-end memory networks." Advances in neural information processing systems. 2015.
21. Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. "Empirical evaluation of gated recurrent neural networks on sequence modeling." NIPS 2014 Workshop on Deep Learning, December 2014. 2014.
22. Zeiler, Matthew D. "ADADELTA: an adaptive learning rate method." arXiv preprint arXiv:1212.5701 (2012).
23. Liu, Fei, and Julien Perez. "Gated end-to-end memory networks." Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. 2017.
24. Gulcehre, C., Ahn, S., Nallapati, R., Zhou, B., and Bengio, Y. "Pointing the Unknown Words." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vol. 1. 2016.
25. Eric, Mihail, and Christopher D. Manning. "A Copy-Augmented Sequence-to-Sequence Architecture Gives Good Performance on Task-Oriented Dialogue." EACL 2017 (2017): 468.