Learning Personalized End-to-End Goal-Oriented Dialog

Liangchen Luo,[†,*] **Wenhao Huang,**[‡] **Qi Zeng,**[†] **Zaiqing Nie,**[§] **Xu Sun**[†]

[†]MOE Key Lab of Computational Linguistics, School of EECS, Peking University, Beijing, China
[‡]Shanghai Discovering Investment, Shanghai, China [§]Alibaba AI Labs, Beijing, China
[†]{luolc,pkuzengqi,xusun}@pku.edu.cn
[‡]huangwh@discoveringgroup.com [§]zaiqing.nzq@alibaba-inc.com

Abstract

Most existing works on dialog systems only consider conversation content while neglecting the personality of the user the bot is interacting with, which begets several unsolved issues. In this paper, we present a personalized end-to-end model in an attempt to leverage personalization in goal-oriented dialogs. We first introduce a PROFILE MODEL which encodes user profiles into distributed embeddings and refers to conversation history from other similar users. Then a PREFERENCE MODEL captures user preferences over knowledge base entities to handle the ambiguity in user requests. The two models are combined into the PERSONALIZED MEMN2N. Experiments show that the proposed model achieves qualitative performance improvements over state-of-the-art methods. As for human evaluation, it also outperforms other approaches in terms of task completion rate and user satisfaction.

1 Introduction

There has been growing research interest in training dialog systems with end-to-end models (Vinyals and Le 2015; Sordoni et al. 2015; Sukhbaatar et al. 2015) in recent years. These models are directly trained on past dialogs, without assumptions on the domain or dialog state structure (Bordes, Boureau, and Weston 2017). One of their limitations is that they select responses only according to the content of the conversation and are thus incapable of adapting to users with different personalities. Specifically, common issues with such content-based models include: (i) the inability to **adjust language style** flexibly (Herzig et al. 2017); (ii) the lack of a **dynamic conversation policy** based on the interlocutor's profile (Joshi, Mi, and Faltings 2017); and (iii) the incapability of **handling ambiguities** in user requests.

Figure 1 illustrates these problems with an example. The conversation happens in a restaurant reservation scenario. First, the responses from the content-based model are plain and boring, and not able to adjust appellations and language styles like the personalized model. Second, in the recommendation phase, the content-based model can only provide candidates in a random order, while a personalized model

---

can change recommendation policy dynamically, and in this case, match the user dietary. Third, the word "*contact*" can be interpreted into "*phone*" or "*social media*" contact information in the knowledge base. Instead of choosing one randomly, the personalized model handles this ambiguity based on the learned fact that young people prefer social media account while the elders prefer phone number.
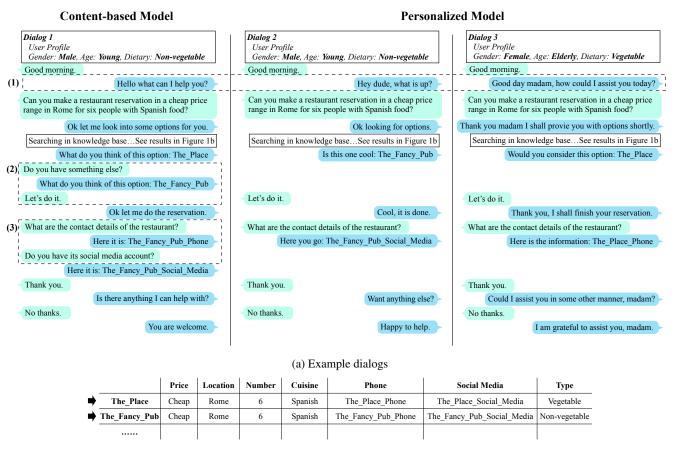
Psychologists have proven that during a dialog humans tend to adapt to their interlocutor to facilitate understanding, which enhances conversational efficiency (Brown 1965; 1987; Kroger and Wood 1992). To improve agent intelligence, we may polish our model to learn such human behaviors in conversations. A big challenge in building personalized dialog systems is how to utilize the user profile and generate personalized responses correspondingly. To overcome it, existing works (Qian et al. 2017; Herzig et al. 2017) often conduct extra procedures to incorporate personalization in training, such as intermediate supervision and pre-training of user profiles, which are complex and time-consuming. In contrast, our work is totally end-to-end.

In this paper, we propose a **PROFILE MODEL** and a **PREFERENCE MODEL** to leverage user profiles and preferences. The PROFILE MODEL learns user personalities with distributed profile representation, and uses a global memory to store conversation context from other users with similar profiles. In this way, it can choose a proper language style and change recommendation policy based on the user profile. To address the problem of ambiguity, the PREFERENCE MODEL learns user preferences among ambiguous candidates by building a connection between the user profile and the knowledge base. Since these two models are both under the MEMN2N framework and make contributions to personalization in different aspects, we combine them into the **PERSONALIZED MEMN2N**.

Our experiments on a goal-oriented dialog corpus, the personalized bAbI dialog dataset, show that leveraging personal information can significantly improve the performance of dialog systems. The PERSONALIZED MEMN2N outperforms current state-of-the-art methods with over 7% improvement in terms of per-response accuracy. A test with real human users also illustrates that the proposed model leads to better outcomes, including higher task completion rate and user satisfaction.

Content-based Model   **Personalized Model**

Dialog 1	Dialog 2	Dialog 3
User Profile	*User Profile*	*User Profile*
Gender: **Male**, *Age:* **Young**, *Dietary:* **Non-vegetable**	*Gender:* **Male**, *Age:* **Young**, *Dietary:* **Non-vegetable**	*Gender:* **Female**, *Age:* **Elderly**, *Dietary:* **Vegetable**

Dialog 1

Good morning.

(1) Hello what can I help you?

Can you make a restaurant reservation in a cheap price range in Rome for six people with Spanish food?

Ok let me look into some options for you.

Searching in knowledge base…See results in Figure 1b

What do you think of this option: The_Place

(2) Do you have something else?

What do you think of this option: The_Fancy_Pub

Let's do it.

Ok let me do the reservation.

(3) What are the contact details of the restaurant?

Here it is: The_Fancy_Pub_Phone

Do you have its social media account?

Here it is: The_Fancy_Pub_Social_Media

Thank you.

Is there anything I can help with?

No thanks.

You are welcome.

Dialog 2

Good morning.

Hey dude, what is up?

Can you make a restaurant reservation in a cheap price range in Rome for six people with Spanish food?

Ok looking for options.

Searching in knowledge base…See results in Figure 1b

Is this one cool: The_Fancy_Pub

Let's do it.

Cool, it is done.

What are the contact details of the restaurant?

Here you go: The_Fancy_Pub_Social_Media

Thank you.

Want anything else?

No thanks.

Happy to help.

Dialog 3

Good morning.

Good day madam, how could I assist you today?

Can you make a restaurant reservation in a cheap price range in Rome for six people with Spanish food?

Thank you madam I shall provie you with options shortly.

Searching in knowledge base…See results in Figure 1b

Would you consider this option: The_Place

Let's do it.

Thank you, I shall finish your reservation.

What are the contact details of the restaurant?

Here is the information: The_Place_Phone

Thank you.

Could I assist you in some other manner, madam?

No thanks.

I am grateful to assist you, madam.

(a) Example dialogs

	Price	Location	Number	Cuisine	Phone	Social Media	Type
➡ **The_Place**	Cheap	Rome	6	Spanish	The_Place_Phone	The_Place_Social_Media	Vegetable
➡ **The_Fancy_Pub**	Cheap	Rome	6	Spanish	The_Fancy_Pub_Phone	The_Fancy_Pub_Social_Media	Non-vegetable
......							

(b) Searched results

Figure 1: Examples to show the common issues with content-based models. We can see that the content-based model (1) is incapable of adjusting appellations and language styles, (2) fails to provide the best candidate, and (3) fails to choose the correct answer when facing ambiguities. (a) Three dialogs are chosen from the personalized bAbI dialog dataset. Personalized and content-based responses are generated by the PERSONALIZED MEMN2N and a standard memory network, respectively. (b) Examples of valid candidates from a knowledge base that match the user request.

2 Related Work

End-to-end neural approaches to building dialog systems have attracted increasing research interest. It is well accepted that conversation agents include goal-oriented dialog systems and non goal-oriented (*chit-chat*) bots.

Generative recurrent models like SEQ2SEQ have showed promising performance in non goal-oriented *chit-chat* (Ritter, Cherry, and Dolan 2011; Lowe et al. 2015; Luo et al. 2018). More recently, retrieval-based models using a memory network framework have shown their potential in goal-oriented systems (Sukhbaatar et al. 2015; Bordes, Boureau, and Weston 2017). Although steady progress has been made, there are still issues to be addressed: most existing models are content-based, which are not aware of the interlocutor profile, and thus are not capable of adapting to different kinds of users. Considerable research efforts have been devoted so far to make conversational agents smarter by incorporating user profile.

Personalized Chit-Chat The first attempt to model persona is Li et al. (2016), which proposes an approach to as-

sign specific personality and conversation style to agents based on learned persona embeddings. Luan et al. (2017) describe an interesting approach that uses multi-task learning with personalized text data. There are some researchers attempting to introduce personalized information to dialogs by transfer learning (Yang et al. 2017; Zhang et al. 2017).

Since there is usually no explicit personalized information in conversation context, existing models (Qian et al. 2017; Herzig et al. 2017) often require extra procedures to incorporate personalization in training. Qian et al. (2017) add intermediate supervision to learn when to employ the user profile. Herzig et al. (2017) pre-train the user profile with external service. This work, in contrast, is totally end-to-end.

A common approach to leveraging personality in these works is using a conditional language model as the response decoder (Ficler and Goldberg 2017; Li et al. 2016). This can help assign personality or language style to chit-chat bots, but it is useless in goal-oriented dialog systems. Instead of assigning personality to agents (Li et al. 2016; Luan et al. 2017; Qian et al. 2017), our model pays more

attention to the user persona and aims to make agents more adaptive to different kinds of interlocutors.

Personalized Goal-Oriented Dialog As most previous works (Li et al. 2016; Liu et al. 2018; Qian et al. 2017) focus on chit-chat, the combination of personalization and goal-oriented dialog remains unexplored. Recently a new dataset has been released that enriches research resources for personalization in chit-chat (Zhang et al. 2018). However, no open dataset allows researchers to train goal-oriented dialog with personalized information, until the personalized bAbI dialog corpus released by Joshi, Mi, and Faltings (2017).

Our work is in the vein of the memory network models for goal-oriented dialog from Sukhbaatar et al. (2015) and Bordes, Boureau, and Weston (2017). We enrich these models by incorporating the profile vector and using conversation context from users with similar attributes as global memory.

3 End-to-End Memory Network

Since we construct our model based on the MEMN2N by Bordes, Boureau, and Weston (2017), we first briefly recall its structure to facilitate the delivery of our models.

The MEMN2N consists of two components: context memory and next response prediction. As the model conducts a conversation with the user, utterance (from the user) and response (from the model) are in turn appended to the memory. At any given time step t there are $c_1^u, \cdots c_t^u$ user utterances and $c_1^r, \cdots c_{t-1}^r$ model responses. The aim at time t is to retrieve the next response c_t^r.

Memory Representation Following Dodge et al. (2015), we represent each utterance as a bag-of-words using the embedding matrix A, and the context memory m is represented as a vector of utterances as:

$$m = (A\Phi(c_1^u), A\Phi(c_1^r), A\Phi(c_2^u), A\Phi(c_2^r), \\ \cdots, A\Phi(c_{t-1}^u), A\Phi(c_{t-1}^r)) \quad (1)$$

where $\Phi(\cdot)$ maps the utterance to a bag of dimension V (the vocabulary size), and A is a $d \times V$ matrix in which d is the embedding dimension.

So far, information of which speaker spoke an utterance, and at what time during the conversation, are not included in the contents of memory. We therefore encode those pieces of information in the mapping Φ by extending the vocabulary to contain $T = 1000$ extra "time features" which encode the index i of an utterance into the bag-of-words, and two more features (#u, #r) encoding whether the speaker is the user or the bot.

The last user utterance c_t^u is encoded into $q = A\Phi(c_t^u)$, which also denotes the initial query at time t, using the same matrix A.

Memory Operation The model first reads the memory to find relevant parts of the previous conversation for responses selection. The match between q and the memory slots is computed by taking the inner product followed by a softmax: $\alpha_i = \text{Softmax}(q^\top m_i)$, which yields a vector of attention weights. Subsequently, the output vector is constructed by $o = R \sum_i \alpha_i m_i$ where R is a $d \times d$ square matrix. In a multi-layer MEMN2N framework, the query is then updated with $q_2 = q + o$. Therefore, the memory can

be iteratively reread to look for additional pertinent information using the updated query q_2 instead of q, and in general using q_k on iteration k, with a fixed number of iterations N (termed N hops).

Let $r_i = W\Phi(y_i)$, where $W \in \mathbb{R}^{d \times V}$ is another word embedding matrix, and y is a (large) set of candidate responses which includes all possible bot utterances and API calls. The final predicted response distribution is then defined as:

$$\hat{r} = \text{Softmax}(q_{N+1}^\top r_1, \cdots, q_{N+1}^\top r_C) \quad (2)$$

where there are C candidate responses in y.

4 Personalized Dialog System

We first propose two personalized models. The PROFILE MODEL introduces the personality of the interlocutor explicitly (using profile embedding) and implicitly (using global memory). The PREFERENCE MODEL models user preferences over knowledge base entities.

The two models are independent to each other and we also explore their combination as the PERSONALIZED MEMN2N. Figure 2 shows the structure of combined model. The different components are labeled with dashed boxes separately.

4.1 Notation

The user profile representation is defined as follows. Each interlocutor has a user profile represented by n attributes $\{(k_i, v_i)\}_{i=1}^n$, where k_i and v_i denote the key and value of the i-th attribute, respectively. Take the user in the first dialog in Figure 1 as an example, the representation should be $\{(\text{Gender}, \text{Male}), (\text{Age}, \text{Young}), (\text{Dietary}, \text{Non-vegetable})\}$. The i-th profile attribute is represented as a one-hot vector $a_i \in \mathbb{R}^{d_i}$, where there are d_i possible values for key k_i. We define the user profile $\hat{a} \in \mathbb{R}^{d^{(p)}}$ as the concatenation of one-hot representations of attributes: $\hat{a} = \text{Concat}(a_1, \cdots, a_n)$, where $d^{(p)} = \sum_i^n d_i$. The notations of the memory network are the same as introduced in Section 3.

4.2 Profile Model

Our first model is the PROFILE MODEL, which aims to integrate personalized information into the query and ranking part of the MEMN2N. The model consists of two different components: profile embedding and global memory.

Profile Embedding In the MEMN2N, the query q plays a key role in both reading memory and choosing the response, while it contains no information about the user. We expect to add a personalized information term to q at each iteration of the query. Then, the model can be aware of the user profile in the steps of searching relevant utterances in the memory and selecting the final response from the candidates. We thus obtain a distributed profile representation $p \in \mathbb{R}^d$ by applying a linear transformation with the one-hot user profile: $p = P\hat{a}$, where $P \in \mathbb{R}^{d \times d^{(p)}}$. Note that this distributed profile representation shares the same embedding dimension d with the bag-of-words. The query update equation can be changed as:

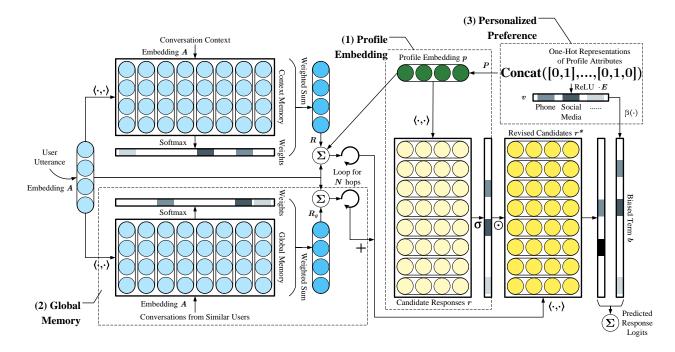$$q_{i+1} = q_i + o_i + p, \quad (3)$$

Figure 2: PERSONALIZED MEMN2N architecture. The incoming user utterance is embedded into a query vector. The model first reads the memory (at top-left) to find relevant history and produce attention weights. Then it generates an output vector by taking the weighted sum followed by a linear transformation. Part (1) is **Profile Embedding**: the profile vector p is added to the query at each iteration, and is also used to revise the candidate responses r. Part (2) is **Global Memory**: this component (at bottom-left) has an identical structure as the original MEMN2N, but it contains history utterances from other similar users. Part (3) is **Personalized Preference**: the bias term is obtained based on the user preference and added to the prediction logits.

where q_i and o_i are the query and output at the i-th hop, respectively.

Also, the likelihood of a candidate being selected should be affected directly by the user profile, no matter what the query is. Therefore, we obtain tendency weights by computing the inner product between p and candidates followed by a sigmoid, and revise the candidates accordingly:

$$r_i^* = \sigma(p^\top r_i) \cdot r_i, \tag{4}$$

where σ is a sigmoid. The prediction $\hat{r}$ is then computed by Equation (2) using r^* instead of r.

Global Memory Users with similar profiles may expect the same or a similar response for a certain request. Therefore, instead of using the profile directly, we also implicitly integrate personalized information of an interlocutor by utilizing the conversation history from similar users as a global memory. The definition of similarity varies with task domains. In this paper, we regard those with the same profile as similar users.

As shown in Figure 2, the global memory component has an identical structure as the original MEMN2N. The difference is that the contents in the memory are history utterances from other similar users, instead of the current conversation. Similarly, we construct the attention weights, output vector,

and iteration equation by

$$\boldsymbol{\alpha}_i^{(g)} = \text{Softmax}(\boldsymbol{q}^\top \boldsymbol{m}_i^{(g)}) \tag{5}$$

$$\boldsymbol{o}^{(g)} = \boldsymbol{R_g} \sum_i \boldsymbol{\alpha}_i^{(g)} \boldsymbol{m}_i^{(g)} \tag{6}$$

$$\boldsymbol{q}_{i+1}^{(g)} = \boldsymbol{q}_i^{(g)} + \boldsymbol{o}_i^{(g)}, \tag{7}$$

where $\boldsymbol{m}^{(g)}$ denotes the global memory, $\boldsymbol{\alpha}^{(g)}$ is the attention weight over the global memory, $\boldsymbol{R_g}$ is a $d \times d$ square matrix, $\boldsymbol{o}^{(g)}$ is the intermediate output vector and $\boldsymbol{q}_{i+1}^{(g)}$ is the result at the i-th iteration. Lastly, we use $\boldsymbol{q}^+ = \boldsymbol{q} + \boldsymbol{q}^{(g)}$ instead of $\boldsymbol{q}$ to make the following computation.

4.3 Preference Model

The PROFILE MODEL has not yet solved the challenge of handling the ambiguity among KB entities, such as the choice between "*phone*" and "*social media*" in Figure 1. The ambiguity refers to the user preference when more than one valid entities are available for a specific request. We propose inferring such preference by taking the relation between user profile and knowledge base into account.

Assuming we have a knowledge base that describes the details of several items, where each row denotes an item and each column denotes one of their corresponding properties. The entity $e_{i,j}$ at row i and column j is the value of the j-th property of item i.

The PREFERENCE MODEL operates as follows.

Given a user profile and a knowledge base with K columns, we predict the user's preference on different columns. We first model the user preference $v \in \mathbb{R}^K$ as:

$$v = \text{ReLU}(E\hat{a}) \qquad (8)$$

where $E \in \mathbb{R}^{K \times d^{(p)}}$. Note that we assume the bot cannot provide more than one option in a single response, so a candidate can only contains one entity at most. The probability of choosing a candidate response should be affected by this preference if the response mentions one of the KB entities.

We add a bias term $b = \beta(v, r, m) \in \mathbb{R}^C$ to revise the logits in Equation (2). The bias for k-th candidate b_k is constructed as the following steps. If the k-th candidate contains no entity, then $b_k = 0$; if the candidate contains an entity $e_{i,j}$, which belongs to item i, then $b_k = \lambda(i, j)$, where given the current conversation context ctx,

$$\lambda(i, j) = \begin{cases} v_j, & \text{if item } i \text{ is mentioned in } ctx; \\ 0, & \text{otherwise.} \end{cases} \qquad (9)$$

For example, the candidate "*Here is the information: The_Place_Phone*" contains a KB entity "*The_Place_Phone*" which belongs to restaurant "*The_Place*" and column "*Phone*". If "*The_Place*" has been mentioned in the conversation, the bias term for this response should be v_{Phone}.

We update the Equation (2) to

$$\hat{r} = \text{Softmax}(q_{N+1}^\top r_1 + b_1, \cdots, q_{N+1}^\top r_C + b_C). \quad (10)$$

4.4 Combined Model

As discussed previously, the PROFILE MODEL and the PREFERENCE MODEL make contributions to personalization in different aspects. The PROFILE MODEL enables the MEMN2N to change the response policy based on the user profile, but fails to establish a clear connection between the user and the knowledge base. On the other hand, the PREFERENCE MODEL bridges this gap by learning the user preferences over the KB entities.

To take advantages of both models, we construct a general PERSONALIZED MEMN2N model by combining them together, as shown in Algorithm 1. All these models are trained to minimize a standard cross-entropy loss between $\hat{r}$ and the true label r_{true}.

5 Experiments

5.1 Dataset

The personalized bAbI dialog dataset (Joshi, Mi, and Faltings 2017) is a multi-turn dialog corpus extended from the bAbI dialog dataset (Bordes, Boureau, and Weston 2017). It introduces an additional user profile associated with each dialog and updates the utterances and KB entities to integrate personalized style. Five separate tasks in a restaurant reservation scenario are introduced along with the dataset. Here we briefly introduce them for better understanding of our experiments. More details on the dataset can be found in the work by Joshi, Mi, and Faltings (2017).

Algorithm 1 Response Prediction by PERSONALIZED MEMN2N

Input: User utterance q, Context memory m, global memory $m^{(g)}$, candidates r and user profile $\hat{a}$
Output: The index y of the next response

1: **procedure** PREDICT($q, m, m^{(g)}, r, \hat{a}$)
2: $\quad p \leftarrow P\hat{a}$ $\qquad\qquad\qquad$ ▷ Profile embedding
3: $\quad q^{(g)} \leftarrow q$
4: $\quad$ **for** N hops **do**
5: $\quad\quad \alpha \leftarrow \text{Softmax}(q^\top m)$
6: $\quad\quad q \leftarrow q + p + R\sum_i \alpha_i m_i$
7: $\quad\quad \alpha^{(g)} \leftarrow \text{Softmax}((q^{(g)})^\top m^{(g)})$
8: $\quad\quad q^{(g)} \leftarrow q^{(g)} + R_g \sum_i \alpha_i^{(g)} m_i^{(g)}$
9: $\quad$ **end for**
10: $\quad v = \text{ReLU}(E\hat{a})$
11: $\quad b = \beta(v, r, m)$ $\qquad\qquad\qquad$ ▷ Bias term
12: $\quad q^+ = q + q^{(g)}$ $\qquad\qquad\qquad$ ▷ Final query
13: $\quad r^* = \sigma(p^\top r) \odot r$ $\qquad\qquad$ ▷ Revised candidates
14: $\quad \hat{r}_i \leftarrow \text{Softmax}((q^+)^\top r_i^* + b_i)$
15: $\quad y \leftarrow \arg\max_i \hat{r}_i$
16: **end procedure**

Task 1: Issuing API Calls Users make queries that contain several blanks to fill in. The bot must ask proper questions to fill the missing fields and make the correct API calls.

Task 2: Updating API Calls Users may update their request and the bot must change the API call accordingly.

Task 3: Displaying Options Given a user request, the KB is queried and the returning facts are added to the dialog history. The bot is supposed to sort the options based on how much users like the restaurant. The bot must be conscious of the user profile and change the sorting strategy accordingly to accomplish this task.

Task 4: Providing Information Users ask for some information about a restaurant, and more than one answer may meet the requirement (i.e., contact with-respect-to social media account and phone number). The bot must infer which answer the user prefers based on the user profile.

Task 5: Full Dialog This task conducts full dialog combining all the aspects of Tasks 1 to 4.

The difficulties of personalization in these tasks are not incremental. In Tasks 1 and 2, the bot is only required to select responses with appropriate meaning and language style. In Tasks 3 and 4, the knowledge base is supposed to be searched, which makes personalization harder. In these two tasks, apart from capturing shallow personalized features in the utterances such as language style, the bot also has to learn different searching or sorting strategies for different user profiles. In Task 5 we expect an average performance (utterance-wise) since it combines the other four tasks.

There are two variations of dataset provided for each task: a full set with around 6000 dialogs and a small set with only 1000 dialogs to create realistic learning conditions. We get the dataset released on ParlAI.[1]

Models	T1: Issuing API Calls	T2: Updating API Calls	T3: Displaying Options	T4: Providing Information	T5: Full Dialog
1. Supervised Embeddings	84.37	12.07	9.21	4.76	51.60
2. MemN2N	99.83 (98.87)	**99.99** (99.93)	58.94 (58.71)	57.17 (57.17)	85.10 (77.74)
3. Split MemN2N	85.66 (82.44)	93.42 (91.27)	68.60 (68.56)	57.17 (57.11)	87.28 (78.10)
4. Profile Embedding	**99.96 (99.98)**	99.96 (99.94)	71.00 (70.95)	57.18 (57.18)	93.83 (81.32)
5. Global Memory	99.76 (98.96)	99.93 (99.74)	71.01 (71.11)	57.18 (57.18)	91.70 (81.43)
6. Profile Model	99.93 (99.96)	99.94 (99.94)	71.12 (70.78)	57.18 (57.18)	93.91 (82.57)
7. Preference Model	99.80 (99.95)	99.97 (**99.97**)	68.90 (68.34)	81.38 (80.30)	94.97 (86.56)
8. Personalized MemN2N	99.91 (99.93)	99.94 (99.95)	**71.43 (71.52)**	**81.56 (80.79)**	**95.33 (88.07)**

Table 1: Evaluation results of the PRESONALIZED MEMN2N on the personalized bAbI dialog dataset. Rows 1 to 3 are baseline models. Rows 4 to 6 are the PROFILE MODEL with profile embedding, global memory and both of them, respectively. In each cell, the first number represents the per-response accuracy on the full set, and the number in parenthesis represents the accuracy on a smaller set with 1000 dialogs.

5.2 Baselines

We consider the following baselines:

Supervised Embedding Model: a strong baseline for both chit-chat and goal-oriented dialog (Dodge et al. 2015; Bordes, Boureau, and Weston 2017).

Memory Network: the MEMN2N by Bordes, Boureau, and Weston (2017), which has been described in detail in Section 3. We add the profile information as an utterance said by the user at the beginning of each dialog. In this way the standard MEMN2N may capture the user persona to some extent.

Split Memory Network: the model proposed by Joshi, Mi, and Faltings (2017) that splits the memory into two parts: profile attributes and conversation history. The various attributes are stored as separate entries in the profile memory before the dialog starts, and the conversation memory operates the same as the MEMN2N.

5.3 Experiment Settings

The parameters are updated by Nesterov accelerated gradient algorithm (Nesterov 1983) and initialized by Xavier initializer. We try different combinations of hyperparameters and find the best settings as follows. The learning rate is 0.001, and the parameter of momentum γ is 0.9. Gradients are clipped to avoid gradient explosion with a threshold of 10. We employ early-stopping as a regularization strategy. Models are trained in mini-batches with a batch size of 64. The dimensionality of word/profile embeddings is 128. We set the maximum context memory and global memory size (i.e. number of utterances) as 250 and 1000, separately. We pad zeros if the number of utterances in a memory is less than 250 or 1000, otherwise we keep the last 250 utterances for the context memory, or randomly choose 1000 valid utterances for the global memory.

5.4 Results

Following Joshi, Mi, and Faltings (2017), we report per-response accuracy across all models and tasks on the per-

sonalized bAbI dataset in Table 1. The per-response accuracy counts the percentage of correctly chosen candidates.

Profile Model Rows 4 to 6 of Table 1 show the evaluation results of the PROFILE MODEL.

As reported in Joshi, Mi, and Faltings (2017), their personalized dialogs model might be too complex for some simple tasks (such as Tasks 1 and 2, which do not rely on KB facts) and tends to overfit the training data. It is reflected in the failure of the split memory model on Tasks 1 and 2. Although it outperforms the standard MEMN2N in some complicated tasks, the latter one is good enough to capture the profile information given in a simple raw text format, and defeats the split memory model in simpler tasks.

To overcome such a challenge, we avoid using excessively complex structures to model the personality. Instead, we only represent the profile as an embedding vector or implicitly. As expected, both profile embedding and global memory approach accomplish Tasks 1 and 2 with a very high accuracy and also notably outperform the baselines in Task 3, which requires utilizing KB facts along with the profile information. Also, the performance of combining the two components together, as shown in row 6, is slightly better than using them independently. The result suggests that we can take advantages of using profile information in an explicit and implicit way in the meantime.

Preference Model Since the PROFILE MODEL does not build a clear connection between the user and the knowledge base, as discussed in Section 4, it may not solve ambiguities among the KB columns. The experiment results are consistent with this inference: the performance of the PROFILE MODEL on Task 4, which requires user request disambiguation, is particularly close to the baselines.

Row 7 shows the evaluation results of the PREFERENCE MODEL, which is proposed to handle the above mentioned challenge. The model achieves significant improvements on Task 4 by introducing the bias term derived from the learned user preference.

Besides, the restaurant sorting challenge in Task 3 depends on the properties of a restaurant to some extent. In-

tuitively, different properties of the restaurants are weighted differently, and the user preference over the KB columns can be considered as scoring weights which is useful for task-solving. As a result, the model also improves the performance in Task 3 compared to the standard MEMN2N.

Personalized MemN2N We test the performance of the combined PERSONALIZED MEMN2N as well. As we have analyzed in Section 4, the PROFILE MODEL and the PREFERENCE MODEL make contributions to personalization in different aspects and their combination has the potential to take advantages of both models. Experiment results confirm our hypothesis that the combined model achieves the best performance with over 7% (and 9% on small sets) improvement over the best baseline for the full dialog task (Task 5).

6 Analysis

As the proposed PERSONALIZED MEMN2N achieves better performance than previous approaches, we conduct an analysis to gain further insight on how the integration of profile and preference helps the response retrieval.

6.1 Analysis of Profile Embeddings

Since we use the learned profile embeddings to obtain tendency weights for candidates selection, as is illustrated in Equation (4), we expect to observe larger weights on candidates that correctly match the profile. For instance, given a profile "*Gender: Male, Age: Young*", we can generate a weight for each response candidate. Due to the fact that candidates are collected from dialogs with different users, they can be divided based on the user profile. Those candidates in the group of *young male* should have larger weights than others.
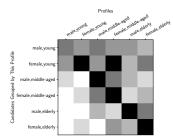

Figure 3: Confusion matrix for profiles and generated tendency weights. Darker cell means larger weight value.

We group the candidates by their corresponding user profile. For each profile, we generate tendency weights and collect the average value for each group. Figure 3 visualizes the results by a confusion matrix. The weights on the diagonal are significantly larger than others, which demonstrates the contribution of profile embeddings in candidate selection.

6.2 Analysis of Global Memory

To better illustrate how much the global memory impacts the performance of the proposed model, we conduct a control experiment. Specifically, we build a model with the same global memory component as described in Section 4.2, but

Models	T5: Full Dialog
Global Memory (similar users)	91.70 (81.43)
Global Memory (random users)	87.17 (78.02)

Table 2: Evaluation results of the control experiment on Task 5: Full Dialog.
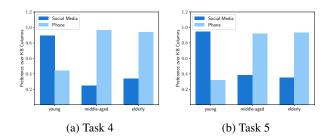

(a) Task 4   (b) Task 5

Figure 4: The preference arguments learned from Task 4: Providing Information, and Task 5: Full Dialog. The preference score is computed by an L2 normalization from v.

the utterances in the memory are from randomly chosen users rather than similar users. We report the results of the control experiment on Task 5 in Table 2. The numbers indicate that the global memory does help improve the performance.

6.3 Analysis of Preference

Remember that we use a preference vector v to represent the user's preference over the columns in the knowledge base. Therefore, we investigate the learned arguments grouped by profile attributes. As seen in Figure 4, the model successfully learns the fact that young people prefer social media as their contact information, while middle-aged and elderly people prefer phone number. The result shows great potential and advantage of end-to-end models. They are capable of learning meaningful intermediate arguments while being much simpler than existing reinforcement learning methods and pipeline models for the task of personalization in dialogs.

6.4 Human Evaluation

To demonstrate the effectiveness of the personalization approach over standard models more convincingly, we build an interactive system based on the proposed model and baselines, and conduct a human evaluation. Since it is impractical to find testers with all profiles we need, we randomly build 20 profiles with different genders, ages and preferences, and ask three judges to act as the given roles. They talk to the system and score the conversations in terms of task completion rate and satisfaction. Task completion rate stands for how much the system accomplish the users' goal. Satisfaction refers to whether the responses are appropriate to the user profile. The scores are averaged and range from 0 to 1 (0 is the worst and 1 is perfect). We find that PERSONALIZED MEMN2N wins the MEMN2N baseline with 27.6% and 14.3% higher in terms of task completion rate and satisfaction, respectively, with $p < 0.03$.

7   Conclusion and Future Work

We introduce a novel end-to-end model for personalization in goal-oriented dialog. Experiment results on open datasets and further analysis show that the model is capable of overcoming some existing issues in dialog systems. The model improves the effectiveness of the bot responses with personalized information, and thus greatly outperforms state-of-the-art methods.

In future work, more representations of personalities apart from the profile attribute can be introduced into goal-oriented dialogs models. Besides, we may explore on learning profile representations for non-domain-specific tasks and consider KB with more complex format such as ontologies.

References

Bordes, A.; Boureau, Y.-L.; and Weston, J. 2017. Learning end-to-end goal-oriented dialog. In *the 5th sInternational Conference on Learning Representations (ICLR)*.

Brown, R. 1965. Social psychology.

Brown, R. 1987. Theory of politeness: An exemplary case. In *meeting of the Society of Experimental Social Psychologists, Charlottesville, VA*.

Dodge, J.; Gane, A.; Zhang, X.; Bordes, A.; Chopra, S.; Miller, A. H.; Szlam, A.; and Weston, J. 2015. Evaluating prerequisite qualities for learning end-to-end dialog systems. *CoRR* abs/1511.06931.

Ficler, J., and Goldberg, Y. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, 94–104. Copenhagen, Denmark: Association for Computational Linguistics.

Herzig, J.; Shmueli-Scheuer, M.; Sandbank, T.; and Konopnicki, D. 2017. Neural response generation for customer service based on personality traits. In *Proceedings of the 10th International Conference on Natural Language Generation*, 252–256. Santiago de Compostela, Spain: Association for Computational Linguistics.

Joshi, C. K.; Mi, F.; and Faltings, B. 2017. Personalization in goal-oriented dialog. *CoRR* abs/1706.07503.

Kroger, R. O., and Wood, L. A. 1992. Are the rules of address universal? iv: Comparison of chinese, korean, greek, and german usage. *Journal of cross-cultural psychology* 23(2):148–162.

Li, J.; Galley, M.; Brockett, C.; Spithourakis, G.; Gao, J.; and Dolan, B. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 994–1003. Berlin, Germany: Association for Computational Linguistics.

Liu, B.; Xu, Z.; Sun, C.; Wang, B.; Wang, X.; Wong, D. F.; and Zhang, M. 2018. Content-oriented user modeling for personalized response ranking in chatbots. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26(1):122–133.

Lowe, R.; Pow, N.; Serban, I.; and Pineau, J. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *CoRR* abs/1506.08909.

Luan, Y.; Brockett, C.; Dolan, B.; Gao, J.; and Galley, M. 2017. Multi-task learning for speaker-role adaptation in neural conversation models. *CoRR* abs/1710.07388.

Luo, L.; Xu, J.; Lin, J.; Zeng, Q.; and Sun, X. 2018. An Auto-Encoder Matching Model for Learning Utterance-Level Semantic Dependency in Dialogue Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 702–707. Brussels, Belgium: Association for Computational Linguistics.

Nesterov, Y. 1983. A method for unconstrained convex minimization problem with the rate of convergence o (1/k^ 2). In *Doklady AN USSR*, volume 269, 543–547.

Qian, Q.; Huang, M.; Zhao, H.; Xu, J.; and Zhu, X. 2017. Assigning personality/identity to a chatting machine for coherent conversation generation. *CoRR* abs/1706.02861.

Ritter, A.; Cherry, C.; and Dolan, W. B. 2011. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 583–593. Edinburgh, Scotland, UK.: Association for Computational Linguistics.

Sordoni, A.; Galley, M.; Auli, M.; Brockett, C.; Ji, Y.; Mitchell, M.; Nie, J.-Y.; Gao, J.; and Dolan, B. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 196–205. Denver, Colorado: Association for Computational Linguistics.

Sukhbaatar, S.; szlam, a.; Weston, J.; and Fergus, R. 2015. End-to-end memory networks. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc. 2440–2448.

Vinyals, O., and Le, Q. V. 2015. A neural conversational model. *CoRR* abs/1506.05869.

Yang, M.; Zhao, Z.; Zhao, W.; Chen, X.; Zhu, J.; Zhou, L.; and Cao, Z. 2017. Personalized response generation via domain adaptation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, 1021–1024. New York, NY, USA: ACM.

Zhang, W.; Liu, T.; Wang, Y.; and Zhu, Q. 2017. Neural personalized response generation as domain adaptation. *CoRR* abs/1701.02073.

Zhang, S.; Dinan, E.; Urbanek, J.; Szlam, A.; Kiela, D.; and Weston, J. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *CoRR* abs/1801.07243.