A Cooperative Memory Network for Personalized Task-oriented Dialogue Systems with Incomplete User Profiles

Pei, J.; Ren, P.; de Rijke, M.

DOI
10.1145/3442381.3449843

Publication date
2021

Document Version
Final published version

Published in
The Web Conference 2021

License
CC BY

A Cooperative Memory Network for Personalized Task-oriented Dialogue Systems with Incomplete User Profiles

Jiahuan Pei
University of Amsterdam
Amsterdam, The Netherlands
j.pei@uva.nl

Pengjie Ren*
Shandong University
Qingdao, China
renpengjie@sdu.edu.cn

Maarten de Rijke
University of Amsterdam & Ahold Delhaize
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

There is increasing interest in developing personalized Task-oriented Dialogue Systems (TDSs). Previous work on personalized TDSs often assumes that complete user profiles are available for most or even all users. This is unrealistic because (1) not everyone is willing to expose their profiles due to privacy concerns; and (2) rich user profiles may involve a large number of attributes (e.g., gender, age, tastes, ...). In this paper, we study personalized TDSs without assuming that user profiles are complete. We propose a Cooperative Memory Network (CoMemNN) that has a novel mechanism to gradually enrich user profiles as dialogues progress and to simultaneously improve response selection based on the enriched profiles. CoMemNN consists of two core modules: User Profile Enrichment (UPE) and Dialogue Response Selection (DRS). The former enriches incomplete user profiles by utilizing collaborative information from neighbor users as well as current dialogues. The latter uses the enriched profiles to update the current user query so as to encode more useful information, based on which a personalized response to a user request is selected.

We conduct extensive experiments on the personalized bAbI dialogue benchmark datasets. We find that CoMemNN is able to enrich user profiles effectively, which results in an improvement of 3.06% in terms of response selection accuracy compared to state-of-the-art methods. We also test the robustness of CoMemNN against incompleteness of user profiles by randomly discarding attribute values from user profiles. Even when discarding 50% of the attribute values, CoMemNN is able to match the performance of the best performing baseline without discarding user profiles, showing the robustness of CoMemNN.

CCS CONCEPTS

• **Computing methodologies** → **Discourse, dialogue and pragmatics**; • **Information systems** → **Personalization**.

KEYWORDS

Dialogue systems, personalization, neural networks, collaborative agents

*Corresponding author

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449843

ACM Reference Format:
Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. 2021. A Cooperative Memory Network for Personalized Task-oriented Dialogue Systems with Incomplete User Profiles. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442381.3449843

1 INTRODUCTION

The use of Task-oriented Dialogue Systems (TDSs) is becoming increasingly widespread. Unlike Open-ended Dialogue Systems (ODSs) [12, 48], TDSs are meant to help users achieve specific goals during multiple-turn dialogues [3]. Applications include booking restaurants, planning trips, grocery shopping, customer service [e.g., 2, 21, 25, 26, 39, 45].

Considerable progress has been made in improving the performance of TDSs [e.g., 2, 7, 14, 15, 17, 27, 42]. Human-human dialogues reflect diverse personalized preferences in terms of, e.g., modes of expression habits [6, 46], individual needs and related to specific goals [9, 19, 23]. Recent work has begun to explore how to improve the user experience by personalizing TDSs in similar ways. Several personalized TDS models have been proposed and have achieved good performance [9, 19, 47]. Personalized TDS models use user profiles in order to be able to capture, and optimize for, users' personal preferences. Those user profiles may not always be available or complete. While profiles may be obtained by asking users to fill in personal profiles with all predefined attributes [9, 19, 47], more often than not, they are *incomplete* and have missing values for some of the attributes of interest: (1) not all users are willing to expose their profiles due to privacy concerns [37]; Tigunova et al. [36] have shown that users rarely reveal their personal information in dialogues explicitly; and (2) user profiles may involve many attributes (such as, e.g., gender, age, tastes), which makes it hard to collect values for all of them. For example, even if we know a user's favorite food is "fish and chips," this does not mean the user does not like "hamburgers."

In this paper, we study the problem of personalized TDSs with *incomplete user profiles*. This problem comes with two key challenges: (1) how to infer missing attribute values of incomplete user profiles; and (2) how to use enriched profiles so as to enhance personalized TDSs. There have been previous attempts to extract user profiles from open-ended dialogues [11, 13, 35, 36, 41] but to the best of our knowledge the problem of inferring and using missing attribute values has not been studied yet in the context of TDSs.

We address the problem of personalized TDSs with *incomplete user profiles* by proposing an end-to-end *Cooperative Memory Network* (CoMemNN) in which profiles and dialogues are used to mutually improve each other. See Figure 1 for an intuitive sketch.

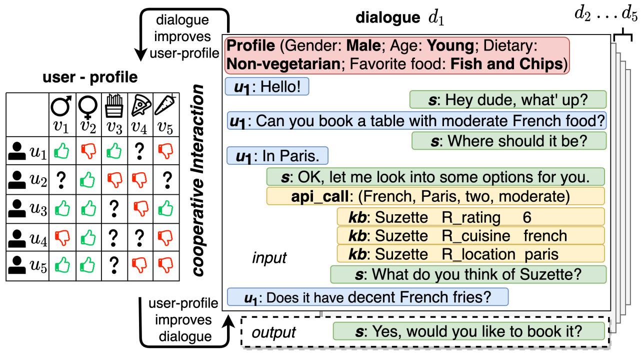

Figure 1: Cooperative interaction between user profiles and dialogues.

The intuition behind Cooperative Memory Network (CoMemNN) is that user profiles can be gradually improved (i.e., missing values can be added) by leveraging useful information from each dialogue turn, and, simultaneously, the performance of Dialogue Response Selection (DRS) can be improved based on enriched profiles for later turns. For example, when user u_1 produces the utterance "Does it have 'decent' french fries?" and the user reveals his like of "french fries," the attribute 'favorite food' in his user profile can be enriched with the value of "french fries." In addition, we want to consider collaborative information from similar users, assuming that similar users have similar preferences as reflected in their user profiles. For example, a young male non-vegetarian who is a big fan of "pizza" might also love "fish and chips" if there are several users with similar profiles stating "fish and chips" as their favorite food. In turn, knowledge of these preferences can affect the choice of the response selected by a TDS in case there are multiple candidate responses. In other words, users with similar profiles may expect the same or a similar response given a certain dialogue context [19]. CoMemNN operationalizes these intuitions with two key modules: User Profile Enrichment (UPE) and Dialogue Response Selection (DRS). The former enriches incomplete user profiles by utilizing useful information from the current dialogue as well as collaborative information from similar users. The latter uses the enriched profiles to update the query representing all requested information, based on which a personalized response is selected to reply to user requests.

To verify the effectiveness of CoMemNN, we conduct extensive experiments on the personalized bAbI dialogue (PbAbI) benchmark dataset, which comes in two flavors, a small version which has 1,000 dialogues, and a large version, which has 12,000 dialogues. First, we find that CoMemNN improves over the best baseline by 3.06%/2.80% on the small/large dataset, respectively, when using all available user profiles. Second, to assess the performance of CoMemNN in the presence of incomplete user profiles, we randomly discard values of attributes with varying probabilities and find that even when it discards 50% of the attribute values, the performance of CoMemNN matches the performance of the best performing baseline without discarding user profiles. In contrast, the best performing baseline decreases 2.12%/1.97% in performance on the small/large dataset with the same amount of discarded values.

The main contributions of this paper are as follows:

- We consider the task personalized TDSs with incomplete user profiles, which has not been investigated so far, to the best of our knowledge.
- We devise a CoMemNN model with dedicated modules to gradually enrich user profiles as a dialogue progresses and to improve response selection based on enriched profiles at the same time.
- We carry out extensive experiments to show the robustness of CoMemNN in the presence of incomplete user profiles.

2 RELATED WORK

In this section, we briefly present an overview of related work on personalized Open-ended Dialogue Systems (ODSs) and personalized Task-oriented Dialogue Systems (TDSs).

2.1 Personalized Open-ended Dialogue Systems

Previous studies on personalized ODSs mainly fuse unstructured persona information [22, 48]. Li et al. [12] first attempt to incorporate a persona into the Seq2Seq framework [34] to generate personalized responses. Ficler and Goldberg [6] apply an RNN language model conditioned on a persona to control response generation with linguistic style. Zhang et al. [48] find that selection models based on Memory Networks [33] are more promising than recurrent generation models based on Seq2Seq [34]. Mazare et al. [22] develop a response selection model based on MemNN and model persona to improve the performance of an ODS. Song et al. [32] explore how to generate diverse personalized responses using a variational autoencoder conditioned on a persona memory. Liu et al. [16] make use of persona interaction between two interlocutors. Xu et al. [43] further exploit topical information to extend persona.

Prior attempts to address data sparsity problems in order to enhance personalized ODSs have considered pretraining [8, 51], sketch generation and filling [30], multiple-stage decoding [31], multi-task learning [18], transfer learning [40, 44, 49], and meta-learning [20]. Only few studies have explored structured user profiles for ODSs [28, 50, 52].

Most of the methods listed above focus on unstructured persona information while we target structured user profiles. Importantly, they focus on ODSs, so they cannot be applied to TDSs directly.

2.2 Personalized Task-oriented Dialogue Systems

Unlike ODSs, personalized TDSs have not been investigated extensively so far. Joshi et al. [9] release the first and, so far, only benchmark dataset for personalized TDSs, to the best of our knowledge. They propose a memory network based model, MemNN, to encode user profiles and conduct personalized response selection. They also propose an extension of MemNN, Split MemNN, which splits a memory into a profile memory followed by a dialogue memory. Zhang et al. [47] introduce Retrieval MemNN by incorporating a retrieval module into memory network, which enhances the performance by retrieving the relevant responses from other users. Luo et al. [19] present Personalized MemNN which learns distributed embeddings for user profiles, dialogue history, and the dialogue history from users with the same gender and age, and shows better performance by using the idea user bias towards Knowledge Base (KB) entries over candidate responses. Mo et al. [23] introduce

Table 1: Summary of main notation used in the paper.

X_t^u	User utterance at turn t.
X_t^s	System response at turn t.
D_t	Dialogue history at turn t.
$\mathbf{h}_t$	Hidden representation of D_t.
u	A user profile in the form of $\{(a_i, v_i)\}_{i=1}^m$, v_i is a candidate value of i-th attribute a_i.
$\mathbf{p}$	One-hot representation of u.
$\mathbf{q}_t$	A query representation that represents the user's current request at turn t.
$\mathbf{M}_t^P$	Profile memory that contains user profile presentations of u and his/her neighbors at turn t.
$\mathbf{M}_t^D$	Dialogue memory that contains dialogue history presentation of u and his/her neighbors at turn t.

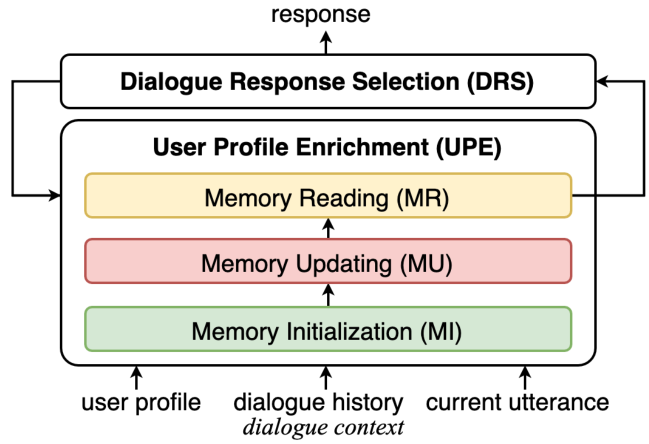

Figure 2: An overview of the CoMemNN architecture, which consists of two cooperative modules: UPE and DRS.

a transfer reinforce learning paradigm to alleviate data scarcity, which uses a collection of multiple users as a source domain and an individual user as a target domain.

The methods above all assume that complete user profiles can be obtained by urging users to fill in all blanks in user profiles, which is unrealistic in practice. Thus, it remains unexplored how the methods above perform when incomplete user profiles are provided, and whether we can bridge the gap in performance if their performance is negatively affected. An alternative is to first infer missing user profiles, e.g., by mining query logs or previous conversations [11, 13, 35, 36], and then apply the model with the above methods. But to do so, we need to train a model to infer missing user profiles asynchronously. Besides, it will likely bring cumulative errors to downstream TDS tasks. Instead, we propose to enrich user profiles *and* achieve a TDS simultaneously with an end-to-end model.

3 METHOD

3.1 Task

In this work, we follow previous studies and model a personalized TDS as a response selection task, which selects a response from predefined candidates given a dialogue context [5, 9, 19, 27, 29, 38, 47]. Table 1 summarizes the main notation used in this paper.

Given a dialogue context (u, D_t, X_t^u) at the t-th dialogue turn, our goal is to select an appropriate response $y_t = X_t^s$ from candidate responses $Y = \{X_j^s\}|_{j=1}^{|Y|}$. Here, u is the user profile, which consists of m attribute-value pairs $\{(a_i, v_i)\}_{i=1}^m$, where a_i is the i-th attribute and v_i is a candidate value of a_i. For example, in Fig. 1, the user profile is denoted as {(Gender, Male), (Age, Young), (Dietary, Non-vegetarian), (Favorite food, Fish and Chips)}. $D_t = X_{1:t-1}$ is the dialogue history. Similar to [9, 19, 47], D_t is represented as a sequence of words that are aggregated from historical utterances $[X_1^u, X_1^s, \ldots, X_{t-1}^u, X_{t-1}^s]$, alternating between the user u or system s. X_t^u denotes the current user utterance, representing the user's current request.

3.2 Overview of CoMemNN

An overview of the proposed architecture, CoMemNN, is shown in Fig. 2. A key aspect of the architecture is that it aims to capture all useful information from the given dialogue context (u, D_t, X_t^u), based on which we learn a query representation $\mathbf{q}_t$ to represent the user's current request. $\mathbf{q}_t$ is usually initialized with the current user utterance X_t^u [9, 19, 47]. Then, $\mathbf{q}_t$ is updated by the User Profile Enrichment (UPE) module by incorporating dialogue and personal information from dialogues and user profiles, respectively. Specifically, UPE captures the interaction between user profiles and dialogues with three submodules: Memory Initialization (MI), Memory Updating (MU) and Memory Reading (MR). MI searches neighbors of the current user to initialize the profile memory $\mathbf{M}_t^P$, which contains profiles from both the current user and his/her neighbors. MI also initializes the dialogue memory $\mathbf{M}_t^D$ with the dialogue history of both the current user and his/her neighbors, each of which is represented by addressing dialogue historical utterance representations with $\mathbf{q}_t$. MU updates the profile memory $\mathbf{M}_t^P$ and the dialogue memory $\mathbf{M}_t^D$ by considering their interaction, after which the user profiles are enriched by inferring missing values based on the dialogue and personal information from the current user and his/her neighbors. Afterwards, MR updates the query representation $\mathbf{q}_t$ by reading from the enriched profile memory as well as dialogue memory. Finally, the Dialogue Response Selection (DRS) module uses the updated query to match candidate responses so as to select an appropriate response. Next, we introduce each of the modules MI, MU and MR, one by one.

3.3 Memory Initialization (MI)

Profile Memory Initialization. To model user-profile relations, we initialize the profile memory as: $\mathbf{M}_t^P = [\Psi(u_1), \ldots, \Psi(u_k)] \in \mathbb{R}^{k \times d}$, where u_1 is the Current Profile (CP) from the current user. The others are Neighbor Profiles (NPs) from neighbor users. For each user profile, the i-th attribute can be represented as an one-hot vector $\tilde{\mathbf{p}}_i \in \mathbb{R}^{C(p_i)}$, where there are $C(a_i)$ candidate values for p_i. Then, each user profile can be initialized as an one-hot vector $\mathbf{p} = \text{Concat}(\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_m) \in \mathbb{R}^n (n = \sum_{i=1}^m (C(p_i))$, which is the concatenation of one-hot representations of attributes. k is the number

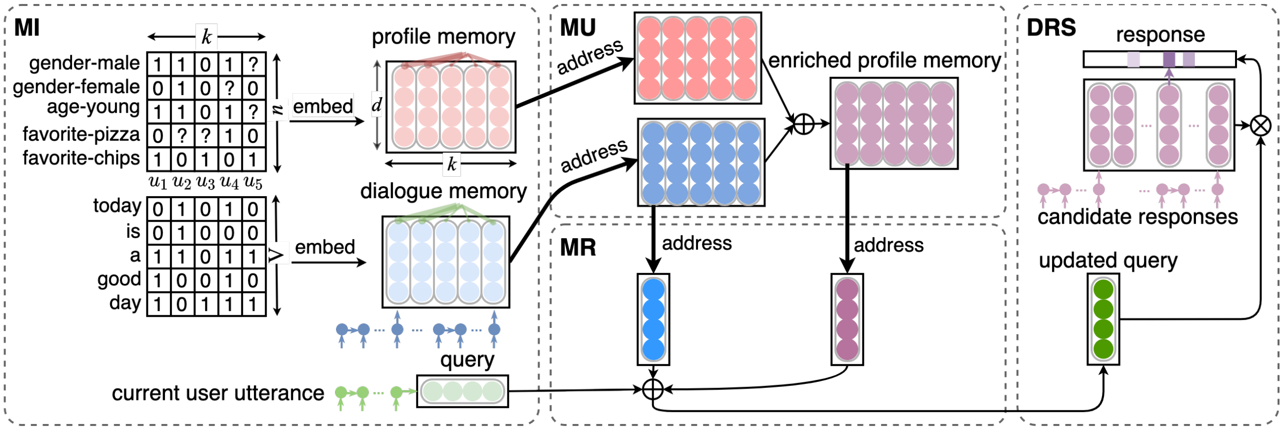

Figure 3: An overview of the dynamic pipeline of the CoMemNN model. The UPE modules captures the interaction between user profiles and dialogues by three submodules: MI, MU and MR. The DRS module and the UPE module cooperate so as to select better responses. Section 3 contains a walkthrough of the model.

of users, d is the embedding dimension, and Ψ is a linear transformation function. Given any user profile u, we find his/her $(k-1)$ nearest neighbors based on dot product similarity.

Dialogue Memory Initialization. To model user-dialogue relations, we initialize a dialogue memory $\mathbf{M}_t^D = [\mathbf{h}_t^1, \ldots, \mathbf{h}_t^k] \in \mathbb{R}^{k \times d}$, where $\mathbf{h}_t^1$ is the representation of the Current Dialogue (CD) from the current user. The others are the Neighbor Dialogues (NDs) from neighbor users. For each user, the dialogue history can be computed as:

$$\begin{aligned} \mathbf{h}_t &= \sum_{i=1}^{2(t-1)} \lambda_t^i \mathbf{H}_t^i \in \mathbb{R}^d \\ \lambda_t^i &= (\tilde{\mathbf{q}}_t)^T \cdot \mathbf{H}_t^i \in \mathbb{R}^1, \end{aligned} \tag{1}$$

where we use the updated query $\tilde{\mathbf{q}}_t$ to address the aggregated dialogue history $\mathbf{H}_t$, the addressing weight λ_t^i is computed by the dot product of query $\tilde{\mathbf{q}}_t$ and the i-th utterance representation $\mathbf{H}_t^i$.

Following [4, 19], we represent each utterance as a bag-of-words using the embedding matrix $\mathbf{E} \in \mathbb{R}^{d \times V}$, where d is the embedding dimension, V is the vocabulary size, $\Phi(\cdot)$ maps the utterance to a bag of dimension V. At the beginning of turn t, the updated query $\tilde{\mathbf{q}}_t$ is initialized as:

$$\tilde{\mathbf{q}}_t = \mathbf{E}\Phi(X_t^u) \in \mathbb{R}^d. \tag{2}$$

Similarly, the aggregated dialogue history $\mathbf{H}_t$ of the current user u_1 can be embedded as:

$$\begin{aligned} &\mathbf{H}_t = \\ &[\mathbf{E}\Phi(X_1^u), \mathbf{E}\Phi(X_1^s), \ldots, \mathbf{E}\Phi(X_{t-1}^u), \mathbf{E}\Phi(X_{t-1}^s)] \in \mathbb{R}^{2(t-1) \times d}. \end{aligned} \tag{3}$$

3.4 Memory Updating (MU)

Dialogue Memory Updating. To obtain an intermediate dialogue memory $\tilde{\mathbf{M}}_t^D$, we update the i-th dialogue memory slot $\tilde{\mathbf{M}}_t^D[:, i]$ using the newest updated query $\tilde{\mathbf{q}}_t$ to address initial dialogue memory $\mathbf{M}_t^D$ as:

$$\begin{aligned} \tilde{\mathbf{M}}_t^D[:, i] &= \sum_{j=1}^{k} \beta_t^j \mathbf{M}_t^D[:, j] \in \mathbb{R}^d \\ \beta_t^j &= (\tilde{\mathbf{q}}_t)^T \cdot \mathbf{M}_t^D[:, j] \in \mathbb{R}^1. \end{aligned} \tag{4}$$

Next, the initial dialogue memory is updated by assigning $\mathbf{M}_t^D = \tilde{\mathbf{M}}_t^D$. As the dialogue evolves, the profile memory will gradually improve the dialogue memory because $\tilde{\mathbf{q}}_t$ contains information from the previous profile memory, so addressing with $\tilde{\mathbf{q}}_t$ links profile-dialogue relations to the dialogue memory.

Profile Memory Updating. Similarly, we can obtain an intermediate profile memory $\tilde{\mathbf{M}}_t^P$ with the following steps:

$$\begin{aligned} \tilde{\mathbf{M}}_t^P[:, i] &= \sum_{j=1}^{k} \alpha_t^j \mathbf{M}_t^P[:, j] \in \mathbb{R}^d \\ \alpha_t^j &= (\mathbf{M}_t^P[:, i])^T \cdot \mathbf{M}_t^P[:, j] \in \mathbb{R}^1. \end{aligned} \tag{5}$$

Next, the profile memory slot $\mathbf{M}_t^P[:, i]$ is updated by a function $\Gamma(\cdot)$ using the intermediate profile memory slot $\tilde{\mathbf{M}}_t^P[:, i]$ and the newest updated dialogue memory slot $\tilde{\mathbf{M}}_t^D[:, i]$:

$$\mathbf{M}_t^P[:, i] = \Gamma(\tilde{\mathbf{M}}_t^P[:, i], \tilde{\mathbf{M}}_t^D[:, i]) \in \mathbb{R}^d, \tag{6}$$

where $\Gamma(\cdot)$ is a mapping function that is implemented by a Multiple Layer Perceptron (MLP) in this work. In this process, the dialogue memory helps to improve the profile memory because $\Gamma(\cdot)$ links dialogue-profile relations to the profile memory.

3.5 Memory Reading (MR)

Dialogue Memory Reading. Since the first memory slot corresponds to the current user, we compute $\mathbf{m}_t^D$ by hard addressing and use it to update the query $\tilde{\mathbf{q}}_t$ as follows:

$$\begin{aligned} \tilde{\mathbf{q}}_t &= \tilde{\mathbf{q}}_t + \mathbf{m}_t^D \in \mathbb{R}^d \\ \mathbf{m}_t^D &= \tilde{\mathbf{M}}_t^D[:, 1] \in \mathbb{R}^d. \end{aligned} \tag{7}$$

Profile Memory Reading. Similarly, we obtain $\mathbf{m}_t^P$ by hard addressing and use it to update the query $\tilde{\mathbf{q}}_t$ as follows:

$$
\begin{aligned}
\tilde{\mathbf{q}}_t &= \tilde{\mathbf{q}}_t + \mathbf{m}_t^P \in \mathbb{R}^d \\
\mathbf{m}_t^P &= \mathbf{M}_t^P[:, 1] \in \mathbb{R}^d.
\end{aligned}
\tag{8}
$$

3.6 Dialogue Response Selection

We use the latest updated query $\tilde{\mathbf{q}}_t$ to match with candidate dialogue responses and the predicted response distribution is computed as follows:

$$
\begin{aligned}
\tilde{\mathbf{y}}_t &= \text{Softmax}(\tilde{\mathbf{q}}_t^T \mathbf{r}_1 + \mathbf{b}_1, \ldots, \tilde{\mathbf{q}}_t^T \mathbf{r}_{|Y|} + \mathbf{b}_{|Y|}) \in \mathbb{R}^{|Y|} \\
\mathbf{b}_j &= \begin{cases} \mathbf{f}_i \in \mathbb{R}^1 & \text{if } \mathbf{r}_j \text{ mentions } i\text{-th attribute of a KB entry} \\ 0 & \text{otherwise} \end{cases} \\
\mathbf{f} &= \text{ReLU}(\mathbf{F}\mathbf{p}_1) \in \mathbb{R}^{kb},
\end{aligned}
\tag{9}
$$

where $\mathbf{r}_j$ is the representation of the j-th candidate response, $|Y|$ is the number of all candidate responses. We follow Luo et al. [19] to model the user bias towards KB entries over the j-th candidate response by a term $\mathbf{b}_j$, where the dimension kb is the number of attributes of a KB entry. $\mathbf{p}_1 \in \mathbb{R}^n$ is the one-hot representation of the current user profile. $\mathbf{F} \in \mathbb{R}^{kb \times n}$ maps user profiles into a KB entry.

3.7 Learning of CoMemNN

Multiple-hop reading or updating has been shown to help improve performance of MemNN by reading or updating the memory multiple times [9, 19, 33]. To enhance CoMemNN, we devise a learning algorithm to update the query and memories with multiple hops, and further differentiate the specific losses of the UPE and DRS modules. The learning procedure is shown in Algorithm 1. First, MI searches neighbors $\{u_2, \ldots, u_k\}$ of the current user u_1 to initialize the profile memory $\mathbf{M}_t^P$ and dialogue memory $\mathbf{M}_t^D$. Second, MU and MR are conducted *HopN* times, and for each time: MU updates the dialogue memory $\mathbf{M}_t^D$ and the profile memory $\mathbf{M}_t^P$ by considering their cooperative interaction. After that, MR updates the query representation $\mathbf{q}_t$ by reading from the enriched dialogue memory followed by profile memory. Last, the Dialogue Response Selection (DRS) module uses the newest updated query $\tilde{\mathbf{q}}_t$ to match candidate responses so as to predict a response distribution $\tilde{\mathbf{y}}_t$.

To evaluate the performance of DRS and UPE, we define two mapping functions to get prediction labels:

- Argmax($\cdot$): it outputs the index y_t with the highest probability in a predicted response distribution $\tilde{\mathbf{y}}_t$;
- PiecewiseArgmax($\cdot$): it generates a 1-0 vector from the predicted enriched profile $\mathbf{m}_t^P$, where $\tilde{\mathbf{p}}_t^1[i] = 1$ only if $\mathbf{m}_t^P[i]$ achieves the highest probability among the values that belong to the same attribute.

To optimize DRS, we use a standard cross-entropy loss between the prediction $\tilde{\mathbf{y}}$ and the one-hot encoded true label $\mathbf{y}$:

$$
\mathcal{L}_{\text{DRS}}(\theta) = -\frac{1}{N_1} \sum_{i=1}^{N_1} \sum_{j=1}^{|Y|} \mathbf{y}_j \log \tilde{\mathbf{y}}_j, \tag{10}
$$

where θ are all parameters in the model and N_1 is the number of training samples.

Algorithm 1: Multiple hop CoMemNN.

Input: turn t, user u_1, profile $\mathbf{p}_1$, dialogue history $\mathbf{H}_t$, query $\mathbf{q}_t$, response candidates $\{\mathbf{r}_1, \ldots, \mathbf{r}_{|Y|}\}$, max hop *HopN*, $(k-1)$ neighbors
Output: A index $\mathbf{y}_t$ of next response; An one-hot vector $\tilde{\mathbf{p}}_t^1$ presenting the enriched profile.

$\{u_2, \ldots, u_k\} \leftarrow \text{Search}(\mathbf{p}_1, k-1)$; ▷ MI
$\mathbf{M}_t^P \leftarrow [\mathbf{p}_1, \ldots, \mathbf{p}_k]$;
$\mathbf{M}_t^D \leftarrow [\mathbf{h}_t^1, \ldots, \mathbf{h}_t^k]; \mathbf{h}_t^i \leftarrow (\tilde{q}_t, \mathbf{H}_t^i), i \in [1, k]; \tilde{\mathbf{q}}_t \leftarrow \mathbf{q}_t$;
while *hop* $\leq$ *HopN* **do**
  $\tilde{\mathbf{M}}_t^D \leftarrow \mathbf{M}_t^D; \tilde{\mathbf{M}}_t^P \leftarrow \mathbf{M}_t^P$; ▷ MU
  $\mathbf{M}_t^D \leftarrow \tilde{\mathbf{M}}_t^D$;
  $\mathbf{M}_t^P \leftarrow \Gamma(\tilde{\mathbf{M}}_t^P, \tilde{\mathbf{M}}_t^D)$;
  $\mathbf{m}_t^D \leftarrow \mathbf{M}_t^D; \tilde{\mathbf{q}}_t \leftarrow \tilde{\mathbf{q}}_t + \mathbf{m}_t^D$; ▷ MR
  $\mathbf{m}_t^P \leftarrow \mathbf{M}_t^P; \tilde{\mathbf{q}} \leftarrow \tilde{\mathbf{q}}_t + \mathbf{m}_t^P$;
end
$\tilde{\mathbf{y}}_t \leftarrow \text{Softmax}(\tilde{\mathbf{q}}_t^T \mathbf{r}_1 + \mathbf{b}_1, \ldots, \tilde{\mathbf{q}}_t^T \mathbf{r}_{|Y|} + \mathbf{b}_{|Y|})$; ▷ DRS
$y_t \leftarrow \text{Argmax}_j(\tilde{\mathbf{y}}_t)$;
$\tilde{\mathbf{p}}_t^1 \leftarrow \text{PiecewiseArgmax}(\mathbf{m}_t^P)$

To control the learning of UPE, we introduce the element-wise mean squared loss between the sampled profile $\mathbf{p} = \{p_1, \ldots, p_{N_2}\}$ and its corresponding enriched profile $\tilde{\mathbf{p}} = \{\tilde{p}_1, \ldots, \tilde{p}_{N_2}\}$:

$$
\mathcal{L}_{\text{UPE}}(\theta) = -\frac{1}{N_2} \sum_{i=1}^{N_2} (p_i - \tilde{p}_i), \tag{11}
$$

where θ are all parameters in the model and N_2 is the number of sampled values.

Finally, the final loss is a linear combination:

$$
\mathcal{L}(\theta) = \mu \mathcal{L}_{\text{DRS}}(\theta) + (1 - \mu) \mathcal{L}_{\text{UPE}}(\theta), \tag{12}
$$

where μ is a hyper-parameter to balance the relative importance of the constituent losses.

4 EXPERIMENTAL SETUP

4.1 Research questions

We seek to answer the following questions in our experiments: (Q1) How well does CoMemNN perform? Does it significantly and continuously outperform state-of-the-art methods? (Q2) What are the effects of different components in CoMemNN? (Q3) Do different profile attributes contribute differently? and (Q4) How well does CoMemNN perform in terms of robustness?

4.2 Dataset and evaluation

We use the personalized bAbI dialogue (PbAbI) dataset [9] for our experiments; this is an extension of the bAbI dialogue (bAbI) dataset that incorporates personalization [2]. To the best of our knowledge, this is the only available open dataset for personalized TDSs. There are two versions: a large version with around 12,000 dialogues and a small version with 1,000 dialogues. These two datasets share the same vocabulary with 14,819 tokens and candidate response set with 43,863 responses. It defines four user profile attributes (gender, age, dietary preference, and favorite food) and composes corresponding attribute-value pairs to a user profile. Each conversation is provided with all of the above user profile attributes, e.g., {(Gender, Male), (Age, Young), (Dietary, Non-vegetarian), (Favorite: Fish and Chips)}. But this does not mean the given user profile is complete because the user may also like "Paella", although "Fish and Chips" is his/her favorite food. To simulate incomplete profiles with various degrees of incompleteness, we randomly discard attribute values from a user profile with probabilities of [0%, 10%, 30%, 50%, 70%, 90%, 100%] and obtain 7 alternative datasets, respectively.

We evaluate the performance of the full dialogue task using the following two metrics [9]:

- *Response Selection Accuracy* (RSA): the fraction of correct responses out of all candidate responses [9, 19]; and
- *Profile Enrichment Accuracy* (PEA): we define this metric as the fraction of correct profile values out of all discarded profile values.

We use a paired t-test to measure statistical significance ($p < 0.01$) of relative improvements.

To compare model stability, we propose a statistic σ, namely *stability coefficient*, which is defined as the standard deviation of a list of performance results. Formally, given a list of evaluation values $[z_1, \ldots, z_{N+1}]$, either RSA or PEA scores, σ is computed as follows:

$$\sigma(\mathbf{z}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{z}_i - \bar{\mathbf{z}})^2} \qquad (13)$$
$$\mathbf{z} = [z_2 - z_1, \ldots, z_{N+1} - z_N],$$

where $\bar{z}$ is the mean of the values in performance difference list $\mathbf{z}$.

4.3 Baselines

We compare with all the methods that have reported results on the PbAbI dataset [9].

- **Memory Network (MemNN)**. It regards the profile information as the first user utterance ahead of each dialogue and achieves personalization by modeling dialogue context using the standard MemNN model [1].
- **Split Memory Network (SMemNN)**. It splits memory into a profile memory and a dialogue memory. The former encodes user profile attributes as separate entries and the latter operates the same as the MemNN. The element-wise sum of both memories are used for final decision [9].
- **Retrieval Memory Network (RMemNN)**. It features an encoder-encoder memory network with a retrieval module that employs the user utterances and user profiles to collect relevant information from similar users' conversations [47].
- **Personalized Memory Network (PMemNN)**. It uses MemNN to model the current user profile, the current dialogue history, as well as the dialogue history of all users with the same gender and age. It also models user bias towards different KB entries [19].
- **Neighbor-based Personalized Memory Network (NPMemNN)**. Our implementation of PMemNN is based on Pytorch. Unlike PMemNN, we use the dialogue history from the nearest $(k-1)$ neighbors instead of all users with the same gender and age.

4.4 Implementation details

We follow the experimental settings detailed in [19]. The embedding size of word/profile is 128. The size of memory is 250. The mini-batch size is 64. The maximum number of training epoch is 250, and the number of hops is 3 (see Algorithm 1). The K-Nearest Neighbors (KNN) algorithm is implemented based on faiss[1] with the inner product measurement and the number of collaborative users $k = 100$. We implement NPMemNN and CoMemNN in PyTorch.[2] And the code of the other models is taken from the original papers. We use Adam [10] as our optimization algorithm with learning rate of 0.01 and initialize the learnable parameters with the Xavier initializer. We also apply gradient clipping [24] with range $[-10, 10]$ during training. We use $l2$ regularization to alleviate overfitting, the weight of which is set to 10^{-5}. We treat the importance of losses of DRS and UPE equally, i.e., $\mu = 0.5$. The code is available online.[3]

5 RESULTS (Q1)

5.1 Results without discarding user profiles

We show the overall response selection performance of all methods in Table 2.

Table 2: Overall performance in terms of the RSA metric. Bold face indicates leading results. Significant improvements over NPMemNN are marked with * (paired t-test, $p <$ 0.01).

	Small set (%)	Large set (%)
MemNN [9]	77.74	85.10
SMemNN [9]	78.10	87.28
RMemNN [47]	83.94	87.33
PMemNN [19]	88.07	95.33
NPMemNN	87.91	97.49
CoMemNN	**91.13***	**98.13***

First, CoMemNN outperforms all baselines on both the small and large datasets by a large margin. It significantly outperforms the best baseline PMemNN by 3.06% on the small dataset and 2.80% on the large dataset. The improvements demonstrate the effectiveness of CoMemNN. We believe the main reason is that the proposed cooperative mechanism is able to enrich the incomplete profiles gradually as dialogues progress and the enriched profiles improve help to response selection simultaneously. We will analyze this in more depth in the next session.

Second, the performance of NPMemNN is comparable to that of PMemNN on the small dataset and achieves 2.16% higher RSA on the large dataset. Recall that NPMemNN is our implementation of PMemNN using Pytorch; the only difference is the KNN algorithm used for neighbor searching, so the result shows that our new neighbor searching method is more effective. Since our CoMemNN is built upon NPMemNN, for the remaining experiments, we will use NPMemNN for further comparison and analysis.

Third, the results on the small and large datasets mostly show consistent trends. For the remaining analysis experiments in the

[1] https://github.com/facebookresearch/faiss
[2] https://pytorch.org/
[3] https://github.com/Jiahuan-Pei/CoMemNN

next section (Section 6), we will report results on the small dataset only. The findings on the large dataset are qualitatively similar.

5.2 Results with different profile discard ratios

We compare CoMemNN and NPMemNN under different profile discard ratios. The results are shown in Table 3.

Table 3: Comparison of CoMemNN and NPMemNN in terms of the RSA metric w.r.t. different profile discard ratios. Bold face indicates leading results. Significant improvements over NPMemNN are marked with * (paired t-test, $p < 0.01$). The values of Diff. are computed by absolute difference of RSA (%) between CoMemNN and NPMemNN.

Discard Ratio	0%	10%	30%	50%	70%	90%	100%
NPMemNN	87.91	86.11	86.56	85.79	83.93	84.08	**84.83**
CoMemNN	**91.13***	**89.90***	**88.69***	**87.80***	**86.35***	**84.83***	82.85
Small Set/Diff.	3.22	3.79	2.13	2.01	2.42	0.75	−1.98
NPMemNN	97.49	97.01	96.05	95.52	95.40	90.96	90.50
CoMemNN	**98.13***	**97.94***	**97.68***	**97.53***	**96.98***	**96.63***	**92.73***
Large Set/Diff.	0.64	0.93	1.63	2.01	1.58	5.67	2.23

First, CoMemNN significantly outperforms NPMemNN on both the small and large datasets when the profile discard ratios range from 0% to 90%. Specifically, it gains an improvement of 0.75%–3.79% on the small dataset and 0.64%–5.67% on the large dataset, respectively. Without discarding profile attribute values, CoMemNN achieves 3.22% / 0.64% of improvement compared with NPMemNN. Unlike the raw profiles where each attribute has only one value, the enriched profile generated by CoMemNN is able to represent a distribution over all possible values, which can better capture users' preference. For example, a user may label "Fish and Chips" as his favorite food, but this does not mean he does not like "Paella." With the raw profile, this is not addressed.

Second, the performance of CoMemNN steadily decreases with the increase of the profile discard ratio, as is to be expected. This is reasonable as it becomes more and more challenging for CoMemNN to find back missing values of user profiles. Interestingly, the performance difference between CoMemNN and NPMemNN first increases and then decreases with the increase of the profile discard ratio. A possible reason is that CoMemNN is able to infer the missing values of user profiles effectively with lower profile discard ratios. However, the profile enrichment ability decreases due to the lack of too many profile values. This hypothesis can be verified by the results that the increase trend lasts longer on the large dataset. Because even with the same profile discard ratio, there are more values of user profiles left on the large dataset for CoMemNN to infer the missing ones. We note that NPMemNN outperforms CoMemNN when all user profiles are discarded on the small dataset. The reason is that UPE cannot enrich user profiles properly in this case, which results in a negative impact on DRS. But this is not the case on the large dataset where UPE can still enrich user profiles properly when the model can find enough personal information clues from more dialogue history.

Third, to answer **Q4**, we compute the statistic σ (Eq. 13) to compare the model stability. The σ values for CoMemNN and NPMemNN are 0.3357/1.0407 on the small dataset and 1.3479/1.4849 on the large dataset, respectively. Thus, NPMemNN has higher deviations, which shows that CoMemNN is more stable than NPMemNN with various profile discard ratios.

6 ANALYSIS

We analyze the performance of the following variants of CoMemNN:

- **CoMemNN**. The full model.
- **CoMemNN-PEL**. CoMemNN without Profile Enrichment Loss (PEL), defined in Eq. 11.
- **CoMemNN-PEL-UPE**. CoMemNN without PEL or UPE. This is exactly NPMemNN.
- **CoMemNN-NP**. CoMemNN without the Neighbor Profile (NP) as input for UPE.
- **CoMemNN-NP-CP**. CoMemNN without NP or the Current Profile (CP) as input for UPE.
- **CoMemNN-ND**. CoMemNN without the Neighbor Dialogue (ND) of dialogues as input for UPE.
- **CoMemNN-ND-CD**. CoMemNN without ND or the Current Dialogue (CD) of dialogues as input for UPE.
- **CoMemNN-ND-NP**. CoMemNN without ND or NP of dialogues as input for UPE.

6.1 Ablation study on PEA (Q2)

We study the Profile Enrichment Accuracy (PEA) performance of different variants in Table 4.

First, CoMemNN can effectively enrich user profiles by inferring the missing values. It is able to correctly predict more than 98.98% of missing values in user profiles under different profile discard ratios. We believe UPE benefits a lot from modeling the interaction between user profiles and dialogues. UPE is able to capture more personal information from dialogue history with dialogues gradually going on. The PEA scores are all very high, because the PbAbI dataset is simulated, which makes it relatively easy to predict missing attribute values of user profiles.

Second, we can see that each component of UPE generally has a positive effect on the performance since most PEA scores of most variants decrease. Specifically, CoMemNN-PEL decreases by 8.38%–14.20% compared with CoMemNN. This means that it is important to add the UPE loss (Eq. 11), rather than only optimizing the DRS loss (Eq. 10). We also show how the four components of UPE (i.e., NP, CP, ND, and CD as defined in Section 3.3) affect its performance. We find that: (1) CoMemNN-ND-NP continuously decreases 0.90%–2.32% with the increase of the profile discard ratio. This means that neighbor users play an important role. (2) CoMemNN-ND-CD (with 100% profile discard ratio) decreases dramatically, which is as expected, because CoMemNN cannot infer the missing values without any dialogue history and profiles. This also explains the increase of the corresponding RSA score in Table 5. (3) The decrease is mostly less than 2.32% except that the decrease of CoMemNN-ND-CD (with 100% profile discard ratio, i.e., no NP or CP as well) is 64.2%. This reveals that different information sources are complementary to each other. The performance will not be affected largely unless all the four inputs (i.e., NP, CP, ND, CD) are removed.

Table 4: Performance of UPE evaluated in terms of Profile Enrichment Accuracy (PEA). In each cell, the first number represents the PEA (%), and the number in parentheses shows the difference compared with CoMemNN. ↓ and || denote a decrease and no change compared to CoMemNN, respectively.

Discard Ratio	10%	30%	50%	70%	90%	100%
CoMemNN	99.99	99.93	99.82	99.83	99.38	98.98
CoMemNN-PEL	85.71 (↓14.28)	87.85 (↓12.08)	91.34 (↓8.48)	89.19 (↓10.64)	90.04 (↓9.34)	90.60 (↓8.38)
CoMemNN-NP	99.87 (↓0.12)	99.85 (↓0.08)	99.24 (↓0.58)	99.15 (↓0.68)	99.13 (↓0.25)	98.86 (↓0.12)
CoMemNN-NP-CP	98.89 (↓1.10)	99.09 (↓0.84)	99.16 (↓0.66)	99.20 (↓0.63)	99.14 (↓0.23)	98.92 (↓0.06)
CoMemNN-ND	99.72 (↓0.26)	99.87 (↓0.06)	99.80 (↓0.02)	99.46 (↓0.37)	98.72 (↓0.66)	97.23 (↓1.75)
CoMemNN-ND-CD	99.99 (||0.00)	99.86 (↓0.07)	99.68 (↓0.14)	99.69 (↓0.14)	99.19 (↓0.19)	34.78 (↓64.2)
CoMemNN-ND-NP	99.09 (↓0.90)	98.98 (↓0.95)	97.95 (↓1.87)	97.69 (↓2.14)	97.06 (↓2.32)	97.23 (↓1.75)

Table 5: Ablation study on DRS evaluated in terms Response Selection Accuracy (RSA). In each cell, the first number represents the RSA (%), and the number in parentheses shows the difference compared with CoMemNN. ↓ and ↑ denote decrease and increase, respectively. Underlining marks results that are ≥1.0% higher than those of CoMemNN.

Discard Ratio	0%	10%	30%	50%	70%	90%	100%
CoMemNN	91.13	89.90	88.69	87.80	86.35	84.83	82.85
CoMemNN-PEL	90.84 (↓0.29)	90.29 (↑0.39)	89.07 (↑0.38)	87.18 (↓0.62)	85.42 (↓0.93)	80.54 (↓4.29)	81.23 (↓1.62)
CoMemNN-PEL-UPE	87.91 (↓3.22)	86.11 (↓3.79)	86.56 (↓2.13)	85.79 (↓2.01)	83.93 (↓2.42)	84.08 (↓0.75)	<u>84.83</u> (↑1.98)
CoMemNN-NP	91.06 (↓0.07)	<u>91.23</u> (↑1.33)	89.17 (↑0.48)	85.26 (↓2.54)	83.30 (↓3.05)	82.10 (↓2.73)	82.83 (↓0.02)
CoMemNN-NP-CP	86.60 (↓4.53)	86.10 (↓3.80)	84.56 (↓4.13)	83.53 (↓4.27)	82.48 (↓3.87)	81.95 (↓2.88)	81.35 (↓1.50)
CoMemNN-ND	90.91 (↓0.22)	87.33 (↓2.57)	89.06 (↑0.37)	87.49 (↓0.31)	86.59 (↑0.24)	85.38 (↑0.55)	<u>85.41</u> (↑2.56)
CoMemNN-ND-CD	87.70 (↓3.43)	90.44 (↑0.54)	85.79 (↓2.90)	84.90 (↓2.90)	83.56 (↓2.79)	82.57 (↓2.26)	<u>85.38</u> (↑2.53)
CoMemNN-ND-NP	90.04 (↓1.09)	91.08 (↑1.18)	89.23 (↑0.54)	87.38 (↓0.42)	85.76 (↓0.59)	85.46 (↑0.63)	<u>85.41</u> (↑2.56)

Lastly, we compute the stability coefficient σ (Eq. 13) of the variants in Table 4 which are 0.1867, 1.8781, 0.2236, 0.1402, 25.6845, 0.1867, 0.4182, respectively. This shows that all variants are robust in terms of the performance of UPE with small stability coefficient, except for CoMemNN-ND-CD.

6.2 Ablation study on RSA (Q2)

We investigate the RSA performance of different variants in Table 5.

First, the performance decreases generally by removing any component of UPE. In particular, CoMemNN-PEL has a greater effect on RSA when the profile discard ratios get larger. This is reasonable because the larger the profile discard ratio, the more space for improvement the proposed model has compared with NPMemNN. CoMemNN-PEL-UPE is inferior to CoMemNN-PEL generally, which means that the UPE module helps as it implicitly impact the DRS loss (Eq. 10). But this ability weakens when the profile discard ratio is larger than 90%.

Second, we observe that the four information sources (i.e., NP, CP, ND, CD) have different effects under different profile discard ratios. Particularly, the profiles of the current users and their neighbors generally contribute most to the RSA performance. We can see that CoMemNN-NP-CP drops 1.50%–4.53% under all profile discard ratios. The reason is that user profiles directly store personal information; it is easier to infer missing values from collaborative user profiles than from dialogues.

Third, we find that NP and ND are complementary to each other. CoMemNN-NP either has a massive drop (2.54%–3.05%) or small changes (≤0.48%) with the most profile discard ratios, except for one obvious rise (1.33%) under the 10% profile discard ratio. In contrast, CoMemNN-ND works fine under the 10% profile discard ratio, but it performs poorly for the rest. Thus, the performance of CoMemNN is influence strongly by a drop in attribute values unless we remove both NP and ND under 100% profile discard ratios.

Lastly, the dialogue history also contributes to the RSA performance in most cases. CoMemNN-ND-CD shows decrease (2.26%–3.43%) or a small change (0.54%) for most of cases, except for an obvious increase under the 100% profile discard ratio. We think that the reason is that some of the predicted profiles are not even in provided profiles, which leads to a very limited PEA score of 34.78% under the 100% profile discard ratio (see Table 4). But these predicted values happen to be useful for selecting an appropriate response in DRS.

6.3 Effect of multiple-hop mechanism (Q2)

We compare the RSA performance of CoMemNN and NPMemNN with different numbers of hops. The results are shown in Table 6.

We see that CoMemNN greatly outperforms NPMemNN by a large margin (1.96%–3.56%) with all number of hops. This further confirms the non-trivial improvement of CoMemNN. Besides, CoMemNN improves by 1.06% when the number of hop changes from 1 to 3 and slightly decreases with 4. This means that CoMemNN benefits from a multiple-hop mechanism.

Table 6: Analysis of the effect of hop number on DRS. Bold face indicates leading results. Significant improvements over NPMemNN are marked with * (paired t-test, $p < 0.01$). The values of Diff. are computed by absolute difference of RSA (%) between CoMemNN and NPMemNN.

#Hop	1	2	3	4
NPMemNN	88.11	87.22	87.91	87.61
CoMemNN	**90.07***	**90.78***	**91.13***	**90.77***
Diff.	1.96	3.56	3.22	3.16

6.4 Effect of different profile attributes (Q3)

We explore how the four types of profile attributes (i.e., gender, age, dietary preference, and favorite food) affect the RSA performance. The results are shown in Table 7.

Table 7: Analysis of profile attribute importance to DRS. Discard attribute table shows we discard all values of a specific attribute or a combination of two specific attributes. Retain attribute table shows we retain all values of a specific attribute and discard all values for the rest. Underline indicates the lower bound baseline that retains no attributes. Bold face indicates the upper bound baseline that retains all attributes.

Discarded attribute	none	gender	age	dietary	favorite	all
gender	/	*93.05*	*91.94*	88.86	*91.95*	/
age	/	/	*92.26*	89.37	91.04	/
dietary	/	/	/	86.74	86.42	/
favorite	/	/	/	/	90.25	/
Retained attribute	<u>82.85</u>	87.46	87.93	90.57	87.37	**91.13**

First, each attribute works well in isolation. Specifically, when we only retain the values of each single attribute, we obtain the results in the last row as 87.46%, 87.93%, 90.57%, 87.37% for gender, age, dietary, favorite, respectively. The attribute "dietary" contributes most followed by "age", "gender" and "favorite."

Second, different types of attributes depend on each other and influence the RSA performance differently. If we only remove the values of one attribute, we get the results on the diagonal: 93.05%, 92.26%, 86.74%, 90.25%, respectively. Removing "dietary" drops most followed by "favorite." Thus, "dietary" contributes more than the rest.

An exception is that the RSA performance increases when discarding "gender" and "age." We believe this is the effect of the neighbors. To show this, we further investigate the effect of "gender" and "age" without using neighbor information. The results are shown in Table 8.

We can see that removing "gender" and "age" decreases the performance in this case. Thus, the different effects of "gender" and "age" are due to the neighbors.

7 CONCLUSION

In this paper, we have studied personalized TDSs without assuming that we have complete user profiles. We have proposed Cooperative

Table 8: Analysis of profile attribute importance to DRS without the effect of neighbors. Bold face indicates the baseline of CoMemNN without neighbors. In each cell, the first number represents the RSA (%), and the number in parenthesis shows the difference values, and ↓ denotes decrease compared with the baseline.

	RSA (Diff.)
CoMemNN w/o neighbors	**90.34**
CoMemNN w/o neighbors - gender	88.25 (↓2.09)
CoMemNN w/o neighbors - age	85.62 (↓4.72)
CoMemNN w/o neighbors - gender - age	83.73 (↓6.61)

Memory Network (CoMemNN), which introduces a cooperative mechanism to enrich user profiles gradually as dialogues progress, and to improve response selection based on enriched profiles simultaneously. We also devise a learning algorithm to effectively learn CoMemNN with multiple hops.

Extensive experiments on the personalized bAbI dialogue (PbAbI) dataset demonstrate that CoMemNN significantly outperforms state-of-the-art baselines. Further analysis experiments confirm the effectiveness of CoMemNN by analyzing the performance and contribution of each component.

A limitation of our work is that we tested the performance of CoMemNN on the only open available personalized TDSs dataset PbAbI. We encourage the community to work on creating additional resources for this task.

As to future work, we hope to experiment on more datasets and investigate how the performance varies on different datasets and whether we can further improve the performance by leveraging non-personalized TDS datasets.

ACKNOWLEDGMENTS

This research was partially supported by the China Scholarship Council (CSC). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683* (2016).

[2] Antoine Bordes and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *ICLR*.

[3] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter* 19, 2 (2017), 25–35.

[4] Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander Miller, Arthur Szlam, and Jason Weston. 2015. Evaluating prerequisite qualities for learning end-to-end dialog systems. *arXiv preprint arXiv:1511.06931* (2015).

[5] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *SIGDIAL*. 37–49.

[6] Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*. 94–104.

[7] Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. 2019. Training neural response selection for task-oriented dialogue systems. In *ACL*. 5392–5404.

[8] Jonathan Herzig, Michal Shmueli-Scheuer, Tommy Sandbank, and David Konopnicki. 2017. Neural response generation for customer service based on personality

traits. In *Proceedings of the 10th International Conference on Natural Language Generation*. 252–256.

[9] Chaitanya K Joshi, Fei Mi, and Boi Faltings. 2017. Personalization in goal-oriented dialog. In *NIPS Conversational AI Workshop*.

[10] Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

[11] Aaron W Li, Veronica Jiang, Steven Y Feng, Julia Sprague, Wei Zhou, and Jesse Hoey. 2020. ALOHA: Artificial learning of human attributes for dialogue agents. In *AAAI*. 8155–8163.

[12] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *ACL*. 994–1003.

[13] Xiang Li, Gokhan Tur, Dilek Hakkani-Tür, and Qi Li. 2014. Personal knowledge graph population from user utterances in conversational understanding. In *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 224–229.

[14] Zibo Lin, Deng Cai, Yan Wang, Xiaojiang Liu, Hai-Tao Zheng, and Shuming Shi. 2020. Grayscale data construction and multi-level ranking objective for dialogue response selection. *arXiv preprint arXiv:2004.02421* (2020).

[15] Fei Liu and Julien Perez. 2017. Gated end-to-end memory networks. In *EACL*. 1–10.

[16] Qian Liu, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. 2020. You impress me: Dialogue generation via mutual persona perception. *arXiv preprint arXiv:2004.05388* (2020).

[17] Yichao Lu, Manisha Srivastava, Jared Kramer, Heba Elfardy, Andrea Kahn, Song Wang, and Vikas Bhardwaj. 2019. Goal-oriented end-to-end conversational models with profile features in a real-world setting. In *NAACL-HLT*. 48–55.

[18] Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao, and Michel Galley. 2017. Multi-task learning for speaker-role adaptation in neural conversation models. In *IJCNLP*. 605–614.

[19] Liangchen Luo, Wenhao Huang, Qi Zeng, Zaiqing Nie, and Xu Sun. 2019. Learning personalized end-to-end goal-oriented dialog. In *AAAI*. 6794–6801.

[20] Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. Personalizing dialogue agents via meta-learning. In *ACL*. 5454–5459.

[21] Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *ACL*. 1468–1478.

[22] Pierre-Emmanuel Mazare, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *EMNLP*. 2775–2779.

[23] Kaixiang Mo, Yu Zhang, Shuangyin Li, Jiajun Li, and Qiang Yang. 2018. Personalizing a dialogue system with transfer reinforcement learning. In *AAAI*. 5317–5324.

[24] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *ICML*. 1310–1318.

[25] Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. 2019. A modular task-oriented dialogue system using a neural mixture-of-experts. In *SIGIR Workshop on Conversational Interaction Systems*.

[26] Jiahuan Pei, Pengjie Ren, Christof Monz, and Maarten de Rijke. 2020. Retrospective and prospective mixture-of-generators for task-oriented dialogue response generation. In *ECAI*. 2148–2155.

[27] Jiahuan Pei, Arent Stienstra, Julia Kiseleva, and Maarten de Rijke. 2019. SEntNet: Source-aware recurrent entity network for dialogue response selection. In *IJCAI Workshop SCAI*.

[28] Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Assigning personality/profile to a chatting machine for coherent conversation generation.. In *IJCAI*. 4279–4285.

[29] Janarthanan Rajendran, Jatin Ganhotra, Satinder Singh, and Lazaros Polymenakos. 2018. Learning end-to-end goal-oriented dialog with multiple answers. In *EMNLP*. 3834–3843.

[30] Michael Shum, Stephan Zheng, Wojciech Kryściński, Caiming Xiong, and Richard Socher. 2019. Sketch-Fill-AR: A persona-grounded chit-chat generation framework. *arXiv preprint arXiv:1910.13008* (2019).

[31] Haoyu Song, Yan Wang, Wei-Nan Zhang, Xiaojiang Liu, and Ting Liu. 2020. Generate, delete and rewrite: A three-stage framework for improving persona consistency of dialogue generation. *arXiv preprint arXiv:2004.07672* (2020).

[32] Haoyu Song, Wei-Nan Zhang, Yiming Cui, Dong Wang, and Ting Liu. 2019. Exploiting persona information for diverse generation of conversational responses. In *IJCAI*. 5190–5196.

[33] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*. 2440–2448.

[34] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*. 3104–3112.

[35] Anna Tigunova. 2020. Extracting personal information from conversations. In *Companion Proceedings of The Web Conference 2020*. 284–288.

[36] Anna Tigunova, Andrew Yates, Paramita Mirza, and Gerhard Weikum. 2019. Listening between the lines: learning personal attributes from conversations. In *The Web Conference*. ACM, 1818–1828.

[37] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *SIGKDD*. ACM, 1235–1244.

[38] Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*. 438–449.

[39] Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *ACL*. 665–677.

[40] Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149* (2019).

[41] Chien-Sheng Wu, Andrea Madotto, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020. Getting to know you: User attribute extraction from dialogues. In *Proceedings of the 12th Language Resources and Evaluation Conference*. 581–589.

[42] Chien-Sheng Wu, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2018. End-to-end dynamic query memory network for entity-value independent task-oriented dialog. In *IEEE-ICASSP*. IEEE, 6154–6158.

[43] Minghong Xu, Piji Li, Haoran Yang, Pengjie Ren, Zhaochun Ren, Zhumin Chen, and Jun Ma. 2020. A neural topical expansion framework for unstructured persona-oriented dialogue generation. In *ECAI*. –.

[44] Min Yang, Zhou Zhao, Wei Zhao, Xiaojun Chen, Jia Zhu, Lianqiang Zhou, and Zigang Cao. 2017. Personalized response generation via domain adaptation. In *SIGIR*. 1021–1024.

[45] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proc. IEEE* 101, 5 (2013), 1160–1179.

[46] Wenhuan Zeng, Abulikemu Abuduweili, Lei Li, and Pengcheng Yang. 2019. Automatic generation of personalized comment based on user profile. In *ACL: Student Research Workshop*. 229–235.

[47] Bowen Zhang, Xiaofei Xu, Xutao Li, Yunming Ye, Xiaojun Chen, and Zhongjie Wang. 2020. A memory network based end-to-end personalized task-oriented dialogue generation. *Knowledge-Based Systems* (2020), 106398.

[48] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too?. In *ACL*. 2204–2213.

[49] Wei-Nan Zhang, Qingfu Zhu, Yifa Wang, Yanyan Zhao, and Ting Liu. 2019. Neural personalized response generation as domain adaptation. *World Wide Web* 22, 4 (2019), 1427–1446.

[50] Yinhe Zheng, Guanyi Chen, Minlie Huang, Song Liu, and Xuan Zhu. 2019. Personalized dialogue generation with diversified traits. *arXiv preprint arXiv:1901.09672* (2019).

[51] Yinhe Zheng, Rongsheng Zhang, Minlie Huang, and Xiaoxi Mao. 2020. A Pre-training based personalized dialogue generation model with persona-sparse data. In *AAAI*. 9693–9700.

[52] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics* 46, 1 (2020), 53–93.