Personalizing a Dialogue System with Transfer Reinforcement Learning

Kaixiang Mo[†], Yu Zhang[†], Shuangyin Li[†], Jiajun Li[‡], Qiang Yang[†]
Department of Computer Science and Engineering
Hong Kong University of Science and Technology, Hong Kong, China
[†]{kxmo, zhangyu, shuangyinli, qyang}@cse.ust.hk [‡]{jiajun.li}@alumni.ust.hk

Abstract

It is difficult to train a personalized task-oriented dialogue system because the data collected from each individual is often insufficient. Personalized dialogue systems trained on a small dataset is likely to overfit and make it difficult to adapt to different user needs. One way to solve this problem is to consider a collection of multiple users as a source domain and an individual user as a target domain, and to perform transfer learning from the source domain to the target domain. By following this idea, we propose a PErsonalized Task-oriented diALogue (PETAL) system, a transfer reinforcement learning framework based on POMDP, to construct a personalized dialogue system. The PETAL system first learns common dialogue knowledge from the source domain and then adapts this knowledge to the target domain. The proposed PETAL system can avoid the negative transfer problem by considering differences between the source and target users in a personalized Q-function. Experimental results on a real-world coffee-shopping data and simulation data show that the proposed PETAL system can learn optimal policies for different users, and thus effectively improve the dialogue quality under the personalized setting.

Introduction

Dialogue systems can be classified into two classes: open domain dialogue systems (Ritter, Cherry, and Dolan 2011; Galley et al. 2015; Serban et al. 2015; Li et al. 2016b; Mou et al. 2016) and task-oriented dialogue systems (Levin, Pieraccini, and Eckert 1997; Young et al. 2013; Wen et al. 2015; Wen et al. 2016; Williams and Zweig 2016). Open domain dialogue systems do not limit the dialogue topic to a specific domain, and typically do not have a clear dialogue goal. Task-oriented dialogue systems aim to solve a specific task via dialogues. In this paper, we focus on the task-oriented dialogue systems which aim to assist users to finish a task such as ordering a cup of coffee.

Personalized task-oriented dialogue systems aim to help a user complete a dialogue task better and faster than non-personalized dialogue systems. Personalized dialogue systems can learn about preferences and habits of a user during interactions with the user, and then utilize this personalized information to speed up the conversation process. Personalized dialogue systems could be categorized into rule-based dialogue systems (Thompson, Goker, and Langley 2004; Kim et al. 2014; Bang et al. 2015) and learning-based dialogue systems (Casanueva et al. 2015; Genevay and Laroche 2016). In rule-based personalized dialogue systems, the dialogue state, system speech-act and user speech-act are predefined by developers, hence it is difficult to reuse this system when the dialogue state and the speech-act are hard to define manually. Learning-based personalized dialogue systems could learn states and actions from training data without requiring explicit rules designed by developers.

However, it is difficult to train a personalized task-oriented dialogue system because the data collected from each individual is often insufficient. A personalized dialogue system trained on a small dataset is likely to fail on unseen but common dialogues due to the overfitting. One solution is to consider a collection of multiple users as a source domain and an individual user as a target domain, and transfer common dialogue structure and policy from the source domain to the target domain, which is defined as transfer reinforcement learning. Unlike the traditional transfer learning which does not consider states, the transfer reinforcement learning aims to transfer knowledge between two reinforcement learning tasks. When transferring dialogue knowledge, the challenge lies in the difference between the source and target domains. Some works (Casanueva et al. 2015; Genevay and Laroche 2016) have been proposed to transfer dialogue knowledge among similar users, but they did not model the difference among users, which might harm the performance in the target domain.

In this paper, we propose a PErsonalized Task-oriented diALogue (PETAL) system, which is a transfer reinforcement learning framework based on the POMDP for learning a personalized dialogue system. The PETAL system first learns common dialogue knowledge from the source domain and then adapts this knowledge to the target user. To achieve this goal, the PETAL system models personalized policies with a personalized Q-function defined as the expected cumulative general reward plus the expected cumulative personal reward. The personalized Q-function can model differences between the source and target users and thus can avoid the negative transfer problem brought by the differences. Experimental results on a real-world coffee-ordering dataset and simulation data show that the proposed PETAL system can choose optimal actions for different users and thus can effectively

improve the dialogue quality under the personalized setting.

Our contributions are three-fold. Firstly, we tackle the problem of learning common dialogue knowledge from the source domain and adapting to the target user in a personalized dialogue system. In multi-turn dialogue systems, learning optimal responses in different situations is a non-trivial problem. One naive policy is to always choose previously seen sentences, but it is not necessarily optimal. For example, in the online coffee ordering task, such naive policy could incur many logical mistakes such as asking repeated questions and confirming the order before the user finishes ordering. Secondly, we propose a transfer reinforcement learning framework based on the POMDP to model the preferences of different users. Unlike existing methods, the proposed PETAL system does not require a manually-defined ground-truth state space and it can model the personalized future expected reward. Finally, we demonstrate the effectiveness of the PETAL system on a real-world dataset as well as simulation data.

Related Works

Personalized dialogue systems could be categorized into rule-based dialogue systems and learning-based dialogue systems. For rule-based personalized dialogue systems, Thompson et al. (Thompson, Goker, and Langley 2004) propose an interactive system where users can choose a place via an interactive conversational process and the system could learn user preferences to improve future conversations. Personalization frameworks proposed in (Kim et al. 2014; Bang et al. 2015) extract and utilize user-related facts (triples), and then generate responses by applying predefined templates to these facts. Different from rule-based personalized dialogue systems, learning-based personalized dialogue systems can learn states and actions from training data without requiring explicit rules. Li et al. (Li et al. 2016a) and Zhang et al. (Zhang et al. 2017) study the personalized response generation problem in single-turn open-domain dialogue system, which does not model the multi-turn dialogue context or the dialogue state. Casanueva et al. (Casanueva et al. 2015) propose to initialize personalized dialogue systems for a target user with data from similar users in the source domain to improve the performance for the target user. This work requires a predefined user similarity metric to select similar source users, but when the selected similar users are different from the target user, the performance of the target user will degrade. Genevay and Laroche (Genevay and Laroche 2016) propose to select and transfer an optimized policy from source users to a target user by using a multi-armed stochastic bandit algorithm which does not require a predefined user similarity measure. However, this method has a high complexity since for each target user, it requires n^2 bandit selection operations where n is the number of source users. Moreover, similar to (Casanueva et al. 2015), the differences between selected source users and the target user will deteriorate the performance. Different from these works, the proposed method does not assume the predefined dialogue states and system speech-acts required by the rule-based systems, and it explicitly models dialogue states and the differences between users.

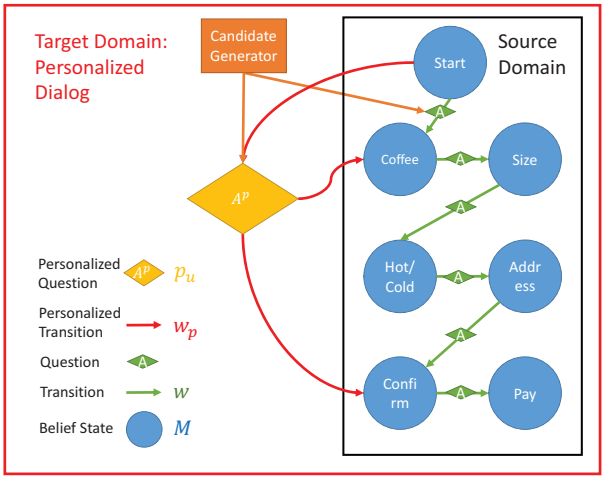

Figure 1: The flowchart of the proposed PETAL system on the coffee-ordering task.

Transfer learning (Taylor and Stone 2009; Pan and Yang 2010; Tan et al. 2014; Tan et al. 2015; Wei, Zheng, and Yang 2016) has been applied to other tasks in dialogue systems. Gasic et al. (Gašic et al. 2013) use transfer learning to extend a dialogue system to include a previously unseen concept. Gasic et al. (Gasic et al. 2014) propose an incremental scheme to adapt an existing dialogue management system to an extended domain. These two works transfer parameters in the policy of the source domain as a prior to the target domain. However, these two models do not deal with multiple source domains and they do not have explicit personalized mechanisms for different users. As a consequence, the negative transfer might occur when the differences between users are large. In contrast, the proposed method has an explicit personalization mechanism and can alleviate negative transfer.

In argumentation agents, there are some works (Hiraoka et al. 2014; Rosenfeld and Kraus 2016b; Rosenfeld and Kraus 2016a) which study personalized dialogue systems. However, these works, which aim to influence users' goal, have different motivations from ours and their formulations are totally different from ours. For example, Rosenfeld and Kraus (Rosenfeld and Kraus 2016b) propose an POMDP-based agent named SPA. The state space of the POMDP in the SPA is based on manually extracted arguments, while the proposed model can learn the state space without human annotation.

PETAL: Transfer Reinforcement Learning Framework for Personalized Dialogue Management

In this section, we introduce the proposed PETAL system. Figure 1 shows the flowchart of the PETAL system on a coffee-ordering task. Here we use PETAL to denote both the proposed framework and the proposed algorithm.

Matrices are denoted in bold capital case, row vectors are in the bold lower case, and scalars are in lower case. The text in the dialogues, denoted in curlicue, is represented

by the bag-of-words assumption. Each of the bag-of-words representations is a vector in which each entry has a binary value.

Problem Setting

In a multi-turn dialogue system, the feedback is usually delayed, so it is more natural to formulate the problem with reinforcement learning. Since the current state of the dialogue is not observable and the ground-truth dialogue states are assumed to be unknown, we formulate the dialogue as a POMDP, which is defined as 7-tuple $\{S, A, O, P, R, Z, \gamma\}$, where S denotes hidden unobservable states, A denotes the replies of the agent, O denotes users' utterances, P is the state transition probability function, R is the reward function, Z is the observation function, and $\gamma \in [0, 1]$ is the discounted factor. In the i-th turn of a dialogue with a user u, S_i^u is the hidden conversation state, $\mathcal{O}_i^u$ is the user utterance, $\mathcal{A}_i^u$ is the reply of the agent, and r_i^u is the reward. In the i-th turn, we only observe $\mathcal{O}_i^u$, $\mathcal{A}_i^u$ and r_i^u. We define $\mathbf{b}_i^u$ as the belief state vector, which represents the probability distribution of unobserved S_i^u. Unlike previous works, we do not assume that the underlying ground-truth state space S is provided. Instead we propose to learn a function to map the dialogue history $\mathcal{H}_i^u = \{\{\mathcal{O}_k^u, \mathcal{A}_k^u\}_{k=0}^{i-1}, \mathcal{O}_i^u\}$ to a compact belief state vector $\mathbf{b}_i^u$.

The inputs for this problem include

1. Abundant dialogue data $\{\{\mathcal{O}_i^{u_s}, \mathcal{A}_i^{u_s}\}_{i=0}^T\}$ of source customers $\{u_s\}$.
2. A few dialogue data $\{\{\mathcal{O}_i^{u_t}, \mathcal{A}_i^{u_t}\}_{i=0}^T\}$ of the target customer u_t.

The expected output is

1. A policy π_{u_t} for target user.

The Framework

In order to solve the problem, we aim to find a policy π_{u_t} for the target user, which could choose an appropriate action $\mathcal{A}_i^{u_t}$ at the i-th turn based on current dialogue history $\mathcal{H}_i^{u_t}$, to maximize the cumulative reward defined as $\pi_{u_t} = \arg\max_\pi \mathbb{E}\left[\sum_{k=0}^\infty \gamma^k r_{t+k+1}^{u_t}\right]$.

To model belief states, we introduce a state projection matrix $\mathbf{M}$ to map the dialogue history $\mathcal{H}_i^u$ to the belief state $\mathbf{b}_i^u$, i.e., $\mathbf{b}_i^u = f(\mathcal{H}_i^u; \mathbf{M})$.

The Q-function is defined as the expected cumulative reward according to policy π_u by starting from belief state $\mathbf{b}_i^u$ and taking action $\mathcal{A}_i^u$ as

$$Q^{\pi_u}(\mathcal{H}_i^u, \mathcal{A}_i^u) = E_\pi\left[\sum_{k=0}^\infty \gamma^k r_{t+k+1}^u | \mathcal{H}_i^u, \mathcal{A}_i^u\right].$$

We choose value-based approaches because there is usually a small number of training data in the target domain, while policy-based approaches, which generate responses word by word, require a lot of training data.

In order to build a personalized dialogue system for the target user, we need to learn a personalized Q-function $Q^{\pi_{u_t}}$ for this user. However, since the training data $\{\{\mathcal{O}_i^{u_t}, \mathcal{A}_i^{u_t}\}_i^T\}$ for the target user u_t is very limited, we can hardly estimate the personalized Q-function $Q^{\pi_{u_t}}$ accurately. In order to learn an accurate $Q^{\pi_{u_t}}$, we can transfer common dialogue knowledge from the source domain, which has a lot of data from many other users $\{\{\mathcal{O}_i^{u_s}, \mathcal{A}_i^{u_s}\}_{i=0}^T\}$. However, different users may have different preferences, hence directly using the data from source users would bring negative effects. We propose to model the personalized Q-function as a general Q-function Q_g plus a personal one Q_p:

$$Q^{\pi_u}(\mathcal{H}_i^u, \mathcal{A}_i^u) = Q_g(\mathcal{H}_i^u, \mathcal{A}_i^u; \mathbf{w}) + Q_p(\mathcal{H}_i^u, \mathcal{A}_i^u; \mathbf{p}_u, w_p)$$
$$\approx \mathbb{E}_{\pi_u}\left[\sum_{k=0}^\infty \gamma^k r_{t+k+1}^{u,g} | \mathcal{H}_i^u, \mathcal{A}_i^u\right] + \mathbb{E}_{\pi_u}\left[\sum_{k=0}^\infty \gamma^k r_{t+k+1}^{u,p} | \mathcal{H}_i^u, \mathcal{A}_i^u\right],$$

where $r_t^{u,g}$ and $r_t^{u,p}$ denotes the general and personal rewards for user u at time t respectively, the general Q-function $Q_g(\mathcal{H}_i^u, \mathcal{A}_i^u; \mathbf{w})$ captures the expected reward related to the general dialogue policy for all users, $\mathbf{w}$ is the set of parameters for the general Q-function and contains a large amount of parameters such that it requires a lot of training data, and the personal Q-function $Q_p(\mathcal{H}_i^u, \mathcal{A}_i^u; \mathbf{p}_u, w_p)$ captures the expected reward related to the preference of each user.

The proposed framework is based on transfer reinforcement learning. $\mathbf{M}$, $\mathbf{w}$ and w_p are shared across different users, which could be trained on the source domain and then transferred to the target domain. These parameters contain common dialogue knowledge, which is independent of users' preferences. Moreover, $\mathbf{p}_u$, which is user-specific, capture the preferences of different users.

Parametric Forms for Personalized Q-function

In this section, we introduce parametric forms for $f(\mathcal{H}_i^u; \mathbf{M})$, $Q_g(\mathcal{H}_i^u, \mathcal{A}_i^u; \mathbf{w})$ and $Q_p(\mathcal{H}_i^u, \mathcal{A}_i^u; \mathbf{p}_u, w_p)$ in the personalized Q-function.

Dialogue states are defined as follows. All utterances and replies will be projected into state vectors with a state projection matrix $\mathbf{M}$, where $\mathbf{M}$ is initialized with the word2vec method and will be updated in the learning process. $\mathbf{b}_i^u = f(\mathcal{H}_i^u; \mathbf{M})$ maps the dialogue history, $\mathcal{H}_i^u = \{\{\mathcal{O}_k^u, \mathcal{A}_k^u\}_{k=0}^{i-1}, \mathcal{O}_i^u\}$, to a belief state vector. The belief state vector $\mathbf{b}_i^u$ is defined as $\mathbf{b}_i^u = \left[\mathbf{o}_{i-1}^{h,u}, \mathbf{o}_i^u, \mathbf{a}_{i-2}^{h,u}, \mathbf{a}_{i-1}^u\right]$, where $\xi = 0.8$ is the memory factor to discount historical state vectors at each time step, $\mathbf{o}_i^{h,u} = \sum_{k=0}^i \xi^{i-k} \mathbf{o}_k^u$, $\mathbf{o}_i^u = \mathcal{O}_i^u \mathbf{M}$, $\mathbf{a}_i^{h,u} = \sum_{k=0}^i \xi^{i-k} \mathbf{a}_k^u$, and $\mathbf{a}_{i-1}^u = \mathcal{A}_{i-1}^u \mathbf{M}$. Based on these definitions, we can see that $\mathbf{o}_i^{h,u}$ represents all previous user utterances, $\mathbf{o}_i^u$ represents the current user utterance, $\mathbf{a}_i^{h,u}$ represents all previous agent replies, and $\mathbf{a}_{i-1}^u$ represents the last agent reply.

In order to model the correlations between entries in $\mathbf{a}_i^u$ and $\mathbf{b}_i^u$, the general Q-function $Q_g(\mathcal{H}_i^u, \mathcal{A}_i^u; \mathbf{w})$ is defined as

$$Q_g(\mathcal{H}_i^u, \mathcal{A}_i^u; \mathbf{w}) = \mathbf{a}_i^u \mathbf{W} (\mathbf{b}_i^u)^T,$$

where superscript ^{T} denotes the transpose of a vector or matrix, $\mathbf{W} \in \mathbb{R}^{d \times 4d}$ is a parameter matrix to be learned. Based on the properties of the Kronecker product and operator $\mathrm{vec}(\cdot)$ which transforms a matrix to a vector in a columnwise manner, we can rewrite $Q_g(\mathcal{H}_i^u, \mathcal{A}_i^u; \mathbf{w})$ as a linear

function on $\mathbf{w} = \text{vec}(\mathbf{W})^T \in \mathbb{R}^{4d^2}$: $Q_g(\mathcal{H}_i^u, \mathcal{A}_i^u; \mathbf{w}) = (\mathbf{b}_i^u \otimes \mathbf{a}_i^u)\mathbf{w}^T$, where $\mathbf{b}_i^u \otimes \mathbf{a}_i^u$ is the Kronecker product of $\mathbf{b}_i^u$ and $\mathbf{a}_i^u$. In multi-turn dialogue systems, there should be different optimal actions in different belief states. The rationale to use the Kronecker product is that the general Q-function should depend on the combination of belief state $\mathbf{b}_i^u$ and action $\mathbf{a}_i^u$, but not independently on $\mathbf{b}_i^u$ and $\mathbf{a}_i^u$.

The personal Q-function learns personalized preferences for each user to avoid the negative effect brought by transferring biased dialogue knowledge across users with different preferences. We denote by C_j the set of all possible choices in the j-th choice set we want to collect and by $\{c_{ij}^u\}_{j=1}^m$ the choices presented in the i-th agent response $\mathcal{A}_i^u$, where m is the total number of order choices, hence c_{ij}^u is an exact choice in C_j. For example, in the coffee-ordering task, $C_1 = \{$Latte, Cappuccino,$\ldots\}$ could be the type of coffees and c_{i1}^u could be any coffee in C_1. On the user side, c_{ij}^u is just the choice of user u for the j-th choice set in the i-th dialogue turn. For example, c_{i1}^u could be "latte" and c_{i2}^u could be "iced". Based on an assumption that different choice sets are independent of each other, for the j-th choice set, the probability of a user u to choose c_{ij}^u follows a categorical distribution $\mathcal{C}(c_{ij}^u; \mathbf{p}_{uj}) = p_{u,j,c_{ij}^u}$ where $|C_j|$ denotes the cardinality of a set, $\mathbf{p}_{uj} \in \mathbb{R}^{|C_j|}$, and $p_{u,j,k}$ denotes the k-th entry in $\mathbf{p}_{uj}$. Hence the personal Q-function for user u is formulated as

$$Q_p(\mathcal{H}_i^u, \mathcal{A}_i^u; \mathbf{p}_u, w_p) = w_p \sum_{j=1}^{m} \mathcal{C}(c_{ij}^u; \mathbf{p}_{uj}) \delta(C_j, \mathcal{H}_i^u),$$

where the personal preference $\mathbf{p}_u = \{\mathbf{p}_{uj}\}_{j=1}^m$ for user u is learned from the training data of that user, $\delta(C_j, \mathcal{H}_i^u)$ equals 1 if the user has not yet made a choice about C_j in the dialogue history $\mathcal{H}_i^u$ and 0 otherwise. $\delta(C_j, \mathcal{H}_i^u)$ implies whether the system will receive a personal reward in the rest of the dialogue, as the Q-function models the cumulative future reward. Here w_p controls the importance of the personalized reward and it is learned from data. When w_p is close to zero, the personalized Q-function will depend on the general dialogue policy. Note that $\sum_{j=1}^m \mathcal{C}(c_{ij}^u|\mathbf{p}_{uj})\delta(C_j, \mathcal{H}_i^u)$ is 0 if we know nothing about the user or $\mathcal{A}_i^u$ does not show any personal preference of user u. Because the vocabulary of choices is much smaller than the whole vocabulary, we can estimate the personal preference parameters $\mathbf{p}_u$ with a few dialogue data $\{\{\mathcal{O}_i^{u_t}, \mathcal{A}_i^{u_t}\}_i^T\}$ of the target user.

By combining the general and personal Q-functions, the personalized Q-function can finally be defined as

$$Q^{\pi_u}(\mathcal{H}_i^u, \mathcal{A}_i^u) = (\mathbf{b}_i^u \otimes \mathbf{a}_i^u)\mathbf{w}^T + w_p \sum_{j=1}^{m} \mathcal{C}(c_{ij}^u|\mathbf{p}_{uj})\delta(C_j, \mathcal{H}_i^u).$$

Here $\mathbf{M}, \mathbf{w}, w_p$ are shared across different users, which could be trained on the source domains and then transferred to the target domain.

Reward

The total reward is the sum of general reward and personal reward, which can be defined as follows:

1. A personal reward $r^{u,p}$ of 0.3 will be received when the user confirms the suggestion of the agent, and a negative reward of -0.2 will be received if the user declines the suggestion by the agent. This is related to the personal information of the user. For example, the user could confirm the address suggested by the agent.
2. A general reward $r^{u,g}$ of 0.1 will be received when the user provides the information about each c_j.
3. A general reward $r^{u,g}$ of 1.0 will be received when the user proceeds with payment.
4. A general reward $r^{u,g}$ of -0.05 will be received by the agent for each dialogue turn to encourage shorter dialogue, -0.2 will be received by the agent if it generates non-logical responses such as asking repeated questions.

Note that the personal reward could not be distinguished from the general reward during the training process. The reward function is designed to encourage the agent to actively make personalized suggestions when the agent can make correct guesses with probability (denoted as p) higher than 0.6. The expected reward is $0.3p - 0.2(1-p)$ for making a suggestion and 0.1 for asking a general question. When $p > 0.6$, $0.3p - 0.2(1-p)$ is larger than 0.1.

Loss Function and Parameter Learning

There are in total four sets of parameters to be learned. We denote all the parameters by $\Theta = \{\mathbf{M}, \mathbf{w}, w_p, \{\mathbf{p}_u\}\}$. When dealing with real-world data, the training set consists of $(\mathcal{H}_i^u, \mathcal{A}_i^u, r_i^u)$, which records optimal actions provided by human, and hence the loss function is defined as follows:

$$\mathcal{L}(\Theta) = \mathbb{E}\left[\left(r_i^u + \max_{\mathcal{A}'_{i+1}} \gamma Q(\mathcal{H}_{i+1}^u, \mathcal{A}'_{i+1}|\Theta) - Q(\mathcal{H}_i^u, \mathcal{A}_i^u|\Theta)\right)^2\right],$$

In the on-policy training with a user simulator, the loss function is defined as

$$\mathcal{L}(\Theta) = \mathbb{E}\left[\left(r_i^u + \gamma Q(\mathcal{H}_{i+1}^u, \mathcal{A}_{i+1}^u|\Theta) - Q(\mathcal{H}_i^u, \mathcal{A}_i^u|\Theta)\right)^2\right].$$

where r_i^u is the reward obtained at time step i and $\mathcal{H}_{i+1}^u$ is the update dialogue history at time step $i+1$.

We use the value iteration method (Bellman 1957) to learn both the general and personal Q-functions. We adopt an online stochastic gradient descent algorithm (Bottou 2010) with a learning rate 0.0001 to optimize our model. Specifically, we use the State-Action-Reward-State-Action (SARSA) algorithm. In the on-policy training with the simulation, the model has a decreasing probability $\eta = 0.2e^{-\frac{\beta}{1000}}$ of choosing a random reply in the candidate set so as to ensure the sufficient exploration, where β is the number of training dialogues seen by the algorithm.

Algorithm and Complexity

The detailed PETAL algorithm is shown in Algorithm 1. We train our model for each user in the source domain. $\mathbf{M}$, $\mathbf{w}$ and w_p are shared by all users and there is a separate $\mathbf{p}_u$ for each user in the source domain. We transfer $\mathbf{M}$, $\mathbf{w}$ and w_p to the target domain by using them to initialize the corresponding variables in the target domain, and then train them as well as $\mathbf{p}_u$ for each target user with limited training data. In noisy

Algorithm 1 The PETAL Algorithm

Input: $\mathcal{D}^s$,$\mathcal{D}^t$
Output: $\Theta = \{\mathbf{M}, \mathbf{w}, w_p \{\mathbf{p}_u\}\}$
procedure TRANSFER REINFORCEMENT LEARNING ALGORITHM($\mathcal{D}^s$,$\mathcal{D}^t$)
    $\{\mathbf{M}, \mathbf{w}, w_p\} \leftarrow$ TRAIN-SOURCE-MODEL($\mathcal{D}^s$)
    $\{\mathbf{M}, \mathbf{w}, w_p, \{\mathbf{p}_u\}\} \leftarrow$TRANSFER($\mathcal{D}^t$,$\mathbf{M}$,$\mathbf{w}$,$w_p$)
function TRAIN-SOURCE-MODEL($\mathcal{D}^s$)
    for $\{\mathcal{O}_i^u, \mathcal{A}_i^u\}$ in $\mathcal{D}^s$ **do**
        for $(\mathcal{H}_i^u, \mathcal{A}_i^u, r_i^u, \mathcal{H}_{i+1}^u, \mathcal{A}_{i+1}^u)$ in $\{\mathcal{O}_i^u, \mathcal{A}_i^u\}$ **do**
            $\Theta_{t+1} \leftarrow \Theta_t + \alpha \Delta_\Theta \mathcal{L}(\Theta_t)$
    return $\{\mathbf{M}, \mathbf{w}, w_p\}$
function TRANSFER($\mathcal{D}^t$, $\mathbf{M}$, $\mathbf{w}$, w_p)
    for $\{\{\mathcal{O}_i^u, \mathcal{A}_i^u\}_i^T\}$ in $\mathcal{D}^t$ **do**
        for $(\mathcal{H}_i^u, \mathcal{A}_i^u, r_i^u, \mathcal{H}_{i+1}^u, \mathcal{A}_{i+1}^u)$ in $\{\mathcal{O}_i^u, \mathcal{A}_i^u\}$ **do**
            $\Theta_{t+1} \leftarrow \Theta_t + \alpha \Delta_\Theta \mathcal{L}(\Theta_t)$
    return $\{\mathbf{M}, \mathbf{w}, w_p, \{\mathbf{p}_u\}\}$

real data, the dialogue states in the source and the target are not completely identical and it is empirically better to tune all parameters on the target domain. Since the source and target users might have different preferences, $\mathbf{p}_u$ learned in source domain is not very useful in the target domain. The personal preference of each target user will be learned separately in each $\mathbf{p}_u$. Without modelling $\mathbf{p}_u$ for each user, different preferences of the source and target users might interfere with each other and thus cause negative transfer.

The number of parameters in our model is around $d^2 + dv$, where v is the total vocabulary size and d is the dimension of the state vector. In our experiment where $v = 1,500$ and $d = 50$, the number of parameters in the general Q-function is about $85k$ and that for the personal Q-function is about 100 for each user, hence the parameters in the personal Q-function could be learned accurately with the limited data in the target domain.

Experiments

In this section, we experimentally verify the effectiveness of the proposed PETAL model by conducting experiments on a real-world dataset and a simulation dataset.

Baselines

We compare the proposed PETAL model with six baseline algorithms which are listed as follows:

1. NoneTL: The dialogue system is trained only with the data from target users.
2. Sim (Casanueva et al. 2015): The dialogue system is trained with the data from both target user and the most similar user in the source domain.
3. Bandit (Genevay and Laroche 2016): For each target user, the most useful source user is identified by a bandit algorithm.
4. PriorSim (Gašic et al. 2013): For each target user, the policy from the most similar user in the source domain is used as a prior.

Table 1: Statistics of the datasets

	Source Domain		Target Domain	
Dataset	Users	Dialogues	Users	Dialogues
Real Data	52	1,859	20	329
Simulation	11	176,000	5	100

5. PriorAll (Gašic et al. 2013): For each target user, the dialogue policy trained on all the users in the source domain is used as a prior.
6. All: The policy is pretrained on all source users' data without the personalized Q-function part.

In order to avoid possible performance difference caused by different base models, we implemented the above baseline methods on top of the same base model defined in the general Q-function part.

Experiments on Real-World Data

In this section, we evaluate our model on a real-world dataset. This dataset, which is collected between July 2015 and April 2016 from an O2O coffee ordering service in a major instant message platform in China, contains 2,185 coffee dialogues between 72 consumers and coffee makers. The users order coffee by providing the coffee type, the temperature, the cup size and the delivery address, hence there are 4 order choices. We select 52 users with more than 23 dialogues as the source domain. Each of the remaining 20 users is used separately as a target domain. In total, there are 1,859 coffee dialogues in the source domain and 329 coffee dialogues in the target domain. 221 earlier dialogues in the target domain are used as the training set and the remaining 108 dialogues form the test set. The statistics of this dataset is shown in Table 1. Note that the popular DSTC datasets do not have personalized preferences and thus could not be used in this paper.

Each user in the target domain is regarded as a target user. We transfer knowledge from the whole source domain to each target user in the target domain. Firstly, we train the source model in the source domain. Then, we transfer the model to the target domain and train on the training set of the target domain. Finally, we test the model on the testing set of target domain.

Table 2: A case study on the real-world dataset. The last column shows candidate responses, where the ground-truth response is marked with *. The first and second columns show predicted rewards of "All" and "PETAL" on these candidates.

User utterance :		I want a cup of coffee.
All	PETAL	Response Candidates
0.86	1.36	* Same as before? Tall hot americano and deliver to Central Conservatory of Music?
0.99	0.92	All right, deliver to No.1199 Beiyuan Road, Chaoyang District, Beijing?
0.72	0.69	What's your address?

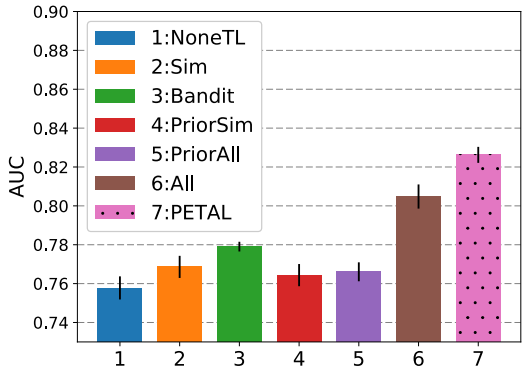

Figure 2: Average AUC score on the the real-world dataset (the higher the better).

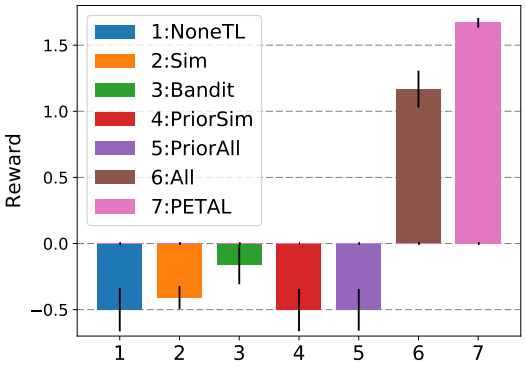

Figure 3: Average Reward on the simulation dataset (the higher the better).

Evaluation Metrics For each turn of the testing conversation, a model will rank the ground-truth reply $\mathcal{A}_i^u$ among 10 randomly chosen agent replies. The label assigned to $\mathcal{A}_i^u$ is 1 and those for randomly chosen agent replies are 0. By following (Williams and Zweig 2016), we calculate the AUC score for each turn in a conversation and the performance of an algorithm is measured by the average AUC score of each dialogue for every user in the test set. Since AUC is the area under the ROC curve which measures the probability that a ground-truth reply will be ranked on top of random replies, if a model ranks the ground-truth reply higher, then this model is better.

Results In Figure 2(a), we report the mean and standard deviation of averaged AUC scores with 5 different random seeds, which are used to randomly sample agent replies as candidates. The performance of "NoneTL", "PriorSim" and "PriorAll" are worse than "All" because fitting only target domain data can cause the overfitting. Transferring data from similar users (i.e., "Sim") is not as good as transferring data from all source users (i.e., "All"), because common knowledge has to be learned from more data. The proposed "PETAL" method performs the best because it learns common knowledge from all users and avoids the negative transfer caused by different preferences among source and target users, which indicates that the proposed personalized model fits dialogues better and demonstrates the effectiveness of PETAL on this real-world dataset.

Case Study A case study is shown in Table 2, where we show three candidates. From the results, we can see that the proposed "PETAL" method ranks the ground-truth response in the first place based on the predicted reward given by the learned personalized Q-function but the "All" method without personalization ranks a wrong address higher, which demonstrates the effectiveness of the proposed method.

Experiments on Simulation Data

In this section, we compare our model with baseline models on the simulated coffee-ordering dialogue data. The simulated users order coffee by providing their coffee type, temperature, size and delivery address, and the agents reply by choosing from a set of predefined candidate responses without knowing the speech-act. We have 11 simulated users in the source domain, in which 10 users have their own coffee preferences while the rest one has no preference. The target domain has 5 users, which have different preferences from users in the source domain. A simulator is designed based on the real-world dataset used in the previous section. The simulator will order according to his preference with probability 0.8 and otherwise the simulator will order coffee randomly. The training set of each user in the target domain has 20 dialogues and the test set has 300 dialogues. The reward in the experiment is the same as the reward defined in the reward section. Firstly, we train the source model by interacting with the source users for a fixed iteration. Then we transfer the source model to target domain and train on the target domain user for 20 dialogues. Finally, we test the model on the target user for 300 dialogues.

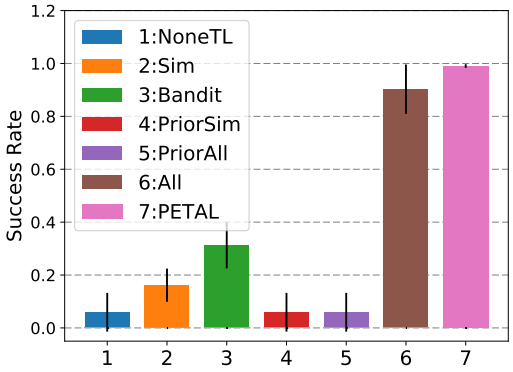

Figure 4: Success rate on the simulation dataset (the higher the better).

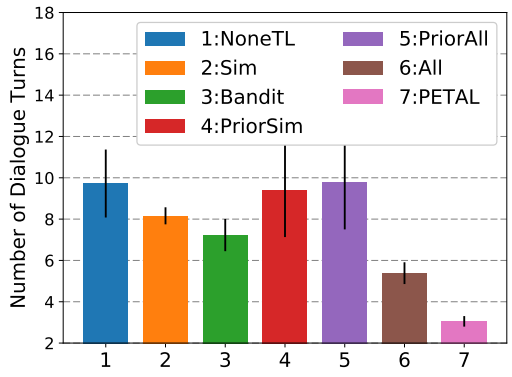

Figure 5: Average dialogue length on the simulation dataset (the lower the better).

Evaluation Metrics Each model will choose a reply from a set of candidates generated with templates and updated dialogue information at each turn, and the simulated users will react to the chosen reply accordingly. We use three evaluation metrics to measure the quality of the whole dialogue. Specifically, for each model, we report the mean and standard deviation of averaged reward (Genevay and Laroche 2016), averaged success rate (Casanueva et al. 2015) and averaged dialogue length over all possible target users in 5 trials with different random seeds.

Table 3: Personalized dialogue case, when the user wants everything as usual.

User:	I want a cup of coffee.
Agent:	Same as before? Tall hot macchiato and deliver to No.1199 Minsheng Road, Pudong District Shanghai?
User:	Yes.
Agent:	Please pay.
User:	Payment completed.

Table 4: Personalized dialogue case, when the user wants to try new options.

User:	I want a cup of coffee.
Agent:	Same as before? Tall hot macchiato and deliver to No.1199 Minsheng Road, Pudong District, Shanghai?
User:	I want iced mocha today.
Agent:	Sure, please pay.
User:	Payment completed.

Results The results are shown in Figure 3(a), Figure 4(a) and Figure 5(a). PETAL outperforms all baselines and obtains the highest average reward, the highest success rate and the lowest dialogue length, which implies that PETAL has found a better dialogue policy which can adapt its behaviour according to the preference of target users and again demonstrates the effectiveness of PETAL in a live environment.

Table 5: A non-personalized dialogue case

User:	I want a cup of coffee.
Agent:	Hot coffee?
User:	Yes.
Agent:	What would you like to drink?
User:	Hot macchiato, please.
Agent:	What's your address?
User:	No.1199 Minsheng Road, Pudong District, Shanghai.
Agent:	Alright. Tall, Grande, or Venti?
User:	Tall.
Agent:	Please pay.
User:	Payment completed.

Case Study We show a typical case for the simulation data from Table 3 to Table 5. The non-personalized dialogue system corresponding to the "All" model has to ask the users all the choices even for frequent users in Table 5, because there is no universal recommendation for all the frequent users with different preferences. However, PETAL has learned the target users' preferences in previous dialogues. As shown in Table 3, the response from the agent is specially tailored for the target user because personalized questions given by the PETAL method can guide the user to complete the coffee-ordering task faster than general questions, leading to shorter dialogue and higher averaged reward. If the user does not want everything as usual, which is shown in Table 4, PETAL can still react correctly due to the shared dialogue knowledge transferred from the source domain. These cases show that PETAL can choose different optimal actions for different users and effectively shorten the conversation.

Conclusion

In this paper, we tackle the problem of learning a personalized dialogue system. We propose the PETAL system, a transfer reinforcement learning framework based on the POMDP. The PETAL system first learns common dialogue knowledge from the source domain and then adapts this knowledge to the target user. We propose to model a personalized policy with a personalized Q-function, which can avoid the negative transfer problem brought by differences between the source users and the target user. As a future direction, we will investigate to transfer knowledge from heterogeneous domains such as knowledge graphs and images.

Acknowledgement

We thank the support of the National Grant Fundamental Research (973 Program) of China under Project 2014CB340304, Hong Kong CERG projects (16211214, 16209715 and 16244616), NSFC (61473087 and 61673202) and the Natural Science Foundation of Jiangsu Province (BK20141340).

References

[Bang et al. 2015] Bang, J.; Noh, H.; Kim, Y.; and Lee, G. G. 2015. Example-based chat-oriented dialogue system with personalized long-term memory. In *Proceedings of International Conference on Big Data and Smart Computing*, 238–243.

[Bellman 1957] Bellman, R. 1957. A Markovian decision process. Technical report, DTIC Document.

[Bottou 2010] Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of 19th International Conference on Computational Statistics*, 177–186.

[Casanueva et al. 2015] Casanueva, I.; Hain, T.; Christensen, H.; Marxer, R.; and Green, P. 2015. Knowledge transfer between speakers for personalised dialogue management. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

[Galley et al. 2015] Galley, M.; Brockett, C.; Sordoni, A.; Ji, Y.; Auli, M.; Quirk, C.; Mitchell, M.; Gao, J.; and Dolan, B. 2015. deltaBLEU: A discriminative metric for generation tasks with intrinsically diverse targets. *arXiv preprint arXiv:1506.06863*.

[Gašic et al. 2013] Gašic, M.; Breslin, C.; Henderson, M.; Kim, D.; Szummer, M.; Thomson, B.; Tsiakoulis, P.; and Young, S. 2013. POMDP-based dialogue manager adaptation to extended domains. In *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

[Gasic et al. 2014] Gasic, M.; Kim, D.; Tsiakoulis, P.; Breslin, C.; Henderson, M.; Szummer, M.; Thomson, B.; and Young, S. J. 2014. Incremental on-line adaptation of POMDP-based dialogue managers to extended domains. In *Proceedings of the 15th Annual Conference of the International Speech Communication Association*, 140–144.

[Genevay and Laroche 2016] Genevay, A., and Laroche, R. 2016. Transfer learning for user adaptation in spoken dialogue systems. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 975–983.

[Hiraoka et al. 2014] Hiraoka, T.; Neubig, G.; Sakti, S.; Toda, T.; and Nakamura, S. 2014. Reinforcement learning of cooperative persuasive dialogue policies using framing. In *Proceedings of the 25th International Conference on Computational Linguistics*, 1706–1717.

[Kim et al. 2014] Kim, Y.; Bang, J.; Choi, J.; Ryu, S.; Koo, S.; and Lee, G. G. 2014. Acquisition and use of long-term memory for personalized dialog systems. In *Proceedings of International Workshop on Multimodal Analyses Enabling Artificial Agents in Human-Machine Interaction*, 78–87.

[Levin, Pieraccini, and Eckert 1997] Levin, E.; Pieraccini, R.; and Eckert, W. 1997. Learning dialogue strategies within the Markov decision process framework. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, 72–79.

[Li et al. 2016a] Li, J.; Galley, M.; Brockett, C.; Spithourakis, G. P.; Gao, J.; and Dolan, B. 2016a. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

[Li et al. 2016b] Li, J.; Monroe, W.; Ritter, A.; and Jurafsky, D. 2016b. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

[Mou et al. 2016] Mou, L.; Song, Y.; Yan, R.; Li, G.; Zhang, L.; and Jin, Z. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. *arXiv preprint arXiv:1607.00970*.

[Pan and Yang 2010] Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10):1345–1359.

[Ritter, Cherry, and Dolan 2011] 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.

[Rosenfeld and Kraus 2016a] Rosenfeld, A., and Kraus, S. 2016a. Providing arguments in discussions on the basis of the prediction of human argumentative behavior. *ACM Transactions on Interactive Intelligent Systems* 6(4):30.

[Rosenfeld and Kraus 2016b] Rosenfeld, A., and Kraus, S. 2016b. Strategical argumentative agent for human persuasion. In *Proceedings of the 22nd European Conference on Artificial Intelligence*.

[Serban et al. 2015] Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A.; and Pineau, J. 2015. Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808*.

[Tan et al. 2014] Tan, B.; Zhong, E.; Xiang, E. W.; and Yang, Q. 2014. Multi-transfer: Transfer learning with multiple views and multiple sources. *Statistical Analysis and Data Mining* 7(4):282–293.

[Tan et al. 2015] Tan, B.; Song, Y.; Zhong, E.; and Yang, Q. 2015. Transitive transfer learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1155–1164.

[Taylor and Stone 2009] Taylor, M. E., and Stone, P. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10:1633–1685.

[Thompson, Goker, and Langley 2004] Thompson, C. A.; Goker, M. H.; and Langley, P. 2004. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research* 21:393–428.

[Wei, Zheng, and Yang 2016] Wei, Y.; Zheng, Y.; and Yang, Q. 2016. Transfer knowledge between cities. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1905–1914.

[Wen et al. 2015] Wen, T.-H.; Gasic, M.; Mrksic, N.; Su, P.-H.; Vandyke, D.; and Young, S. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

[Wen et al. 2016] Wen, T.-H.; Gasic, M.; Mrksic, N.; Rojas-Barahona, L. M.; Su, P.-H.; Ultes, S.; Vandyke, D.; and Young, S. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

[Williams and Zweig 2016] Williams, J. D., and Zweig, G. 2016. End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*.

[Young et al. 2013] Young, S.; Gašić, M.; Thomson, B.; and Williams, J. D. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE* 101(5):1160–1179.

[Zhang et al. 2017] Zhang, W.; Liu, T.; Wang, Y.; and Zhu, Q. 2017. Neural personalized response generation as domain adaptation. *arXiv preprint arXiv:1701.02073*.