Transferable Dialogue Systems and User Simulators

Bo-Hsiang Tseng[†], **Yinpei Dai**[‡], **Florian Kreyssig**[†], **Bill Byrne**[†]
[†]Engineering Department, University of Cambridge, UK
[‡]Alibaba Group
{bht26,flk24,wjb31}@cam.ac.uk
yinpei.dyp@alibaba-inc.com

Abstract

One of the difficulties in training dialogue systems is the lack of training data. We explore the possibility of creating dialogue data through the interaction between a dialogue system and a user simulator. Our goal is to develop a modelling framework that can incorporate new dialogue scenarios through self-play between the two agents. In this framework, we first pre-train the two agents on a collection of source domain dialogues, which equips the agents to converse with each other via natural language. With further fine-tuning on a small amount of target domain data, the agents continue to interact with the aim of improving their behaviors using reinforcement learning with structured reward functions. In experiments on the MultiWOZ dataset, two practical transfer learning problems are investigated: 1) domain adaptation and 2) single-to-multiple domain transfer. We demonstrate that the proposed framework is highly effective in bootstrapping the performance of the two agents in transfer learning. We also show that our method leads to improvements in dialogue system performance on complete datasets.

1 Introduction

This work aims to develop a modelling framework in which dialogue systems (DSs) converse with user simulators (USs) about complex topics using natural language. Although the idea of joint learning of two such agents has been proposed before, this paper is the first to successfully train both agents on complex multi-domain human-human dialogues and to demonstrate a capacity for transfer learning to low-resource scenarios without requiring re-redesign or re-training of the models.

One of the challenges in task-oriented dialogue modelling is to obtain adequate and relevant training data. A practical approach in moving to a new domain is via transfer learning, where pre-training on a general domain with rich data is first performed and then fine-tuning the model on the target domain. End-to-end DS (Wen et al., 2017; Li et al., 2017; Dhingra et al., 2017) are particularly suitable for transfer learning, in that such models are optimised as a single system. By comparison, pipe-lined based DSs with multiple individual components (Young et al., 2013) require fine-tuning of each component system. These separate steps can be done independently, but it becomes difficult to ensure optimality of the overall system.

A similar problem arises in the data-driven US as commonly used in interaction with the DS. Though many USs have been proposed and been widely studied, they usually operate at the level of semantic representation (Kreyssig et al., 2018; El Asri et al., 2016). These models can capture user intent, but are otherwise somewhat artificial as user simulators in that they do not consume and produce natural language. As discussed above for DSs, the end-to-end architecture for the US also offers simplicity in transfer learning across domains.

There are also potential advantages to continued joint training of the DS and the US. If a user model is less than perfectly optimised after supervised learning over a fixed training corpus, further learning through interaction between the two agents offers the US the opportunity to refine its behavior. Prior work has shown benefits from this approach to dialogue policy learning, with a higher success rate at dialogue level (Liu and Lane, 2017b; Papangelis et al., 2019; Takanobu et al., 2020), but there has not been previous work that addresses multi-domain end-to-end dialogue modelling for both agents. Takanobu et al. (2020) address refinement of the dialogue policy alone at the semantic level, but do not address end-to-end system architectures. Liu and Lane (2017b); Papangelis et al. (2019) address single-domain dialogues (Henderson et al., 2014), but not the more realistic and

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 152–166
August 1–6, 2021. ©2021 Association for Computational Linguistics

complex multi-domain dialogues.

This paper proposes a novel learning framework for developing dialogue systems that performs **J**oint **O**ptimisation with a **U**ser **S**imula**T**or (**JOUST**).[1] Through the pre-training on complex multi-domain datasets, two agents are able to interact using natural language, and further create more diverse and rich dialogues. Using reinforcement learning (RL) to optimise both agents enables them to depart from known strategies learned from a fixed limited corpus, to explore new, potentially better policies. Importantly, the end-to-end designs in the framework makes it easier for transfer learning of two agents from one domain to another. We also investigate and compare two reward designs within this framework: 1) the common choice of task success at dialogue level; 2) a fine-grained reward that operates at turn level. Results on MultiWOZ dataset (Budzianowski et al., 2018) show that our method is effective in boosting the performance of the DS in complicated multi-domain conversation. To further test our method in more realistic scenarios, we design specific experiments on two low-resource setups that address different aspects of data sparsity. Our contributions can be summarised as follows:

- Novel contributions in joint optimisation of a fully text-to-text dialogue system with a matched user simulator on complex, multi-domain human-human dialogues.
- Extensive experiments, including exploring different types of reward, showing that our framework with a learnable US boost overall performance and reach new state-of-the-art performance on MultiWOZ.
- Demonstration that our framework is effective in two transfer learning tasks of practical benefit in low-resources scenarios with in-depth analysis of the source of improvements.

2 Pre-training the Dialogue System and User Simulator

In our joint learning framework, we first pre-train the DS and US using supervised learning so that two models are able to interact via natural language. This section presents the architectures of two agents, illustrated in Fig. 1, and the objectives used for supervised learning.

2.1 Dialogue system

Dialogue state tracking (DST) The first task of a DS is to process the dialogue history in order to maintain the belief state which records essential information of the dialogue. A DST model is utilized to predict the set of slot-value pairs which constitute the constraints of the entity for which the user is looking for, *e.g.* {`hotel_area=north`, `hotel_name=gonville_hotel`}.

The DST model used here is an encoder-decoder model with attention mechanism (Bahdanau et al., 2015). The set of slot-value pairs is formulated as a slot sequence together with a value sequence. For the t^{th} dialogue turn, the DST model first encodes the dialogue context and the most recent user utterance x^{us}_{t-1} using a bi-directional LSTM (Graves et al., 2005) to obtain hidden states $H^{enc}_t = \{h^{enc}_1, ..., h^{enc}_j, ...\}$. At the i^{th} decoding step of turn t, the previous decoder hidden state h^{dec}_{i-1} is used to attend over H^{enc}_t to obtain the attention vector a_i. The decoder takes a_i, h^{dec}_{i-1} and the embedding of the slot token predicted at $i-1$ to produce the current hidden state h^{dec}_i. The h^{dec}_i is then passed through separate affine transforms followed by the softmax function to predict a slot token and value for step i. The final belief state is the aggregation of predicted slot-value pairs of all decoding steps.

Database Query Based on the updated belief state, the system searches the database and retrieves the matched entities. In addition, a one-hot vector of size 3 characterises the result of every query.

Context Encoding To capture the dialogue flow, a hierarchical LSTM (Serban et al., 2016) encodes the dialogue context from turn to turn throughout the dialogue. At each turn t, the most recent user utterance x^{us}_{t-1} is encoded by an LSTM-based sentence encoder to obtain a sentence embedding e^{us}_t and hidden states H^{us}_t. Another LSTM is used as the context encoder, which encodes e^{us}_t as well as the output of the context encoder on the user side c^{us}_{t-1} from the previous turn (see Fig. 1). The context encoder produces the next dialogue context state c^{ds}_t for the downstream dialogue manager.

Policy The dialogue manager determines the system dialogue act based on the current state of the dialogue. The system dialogue act is treated as a sequence of tokens in order to handle cases in

[1] The code is released at `https://github.com/andy194673/joust`.

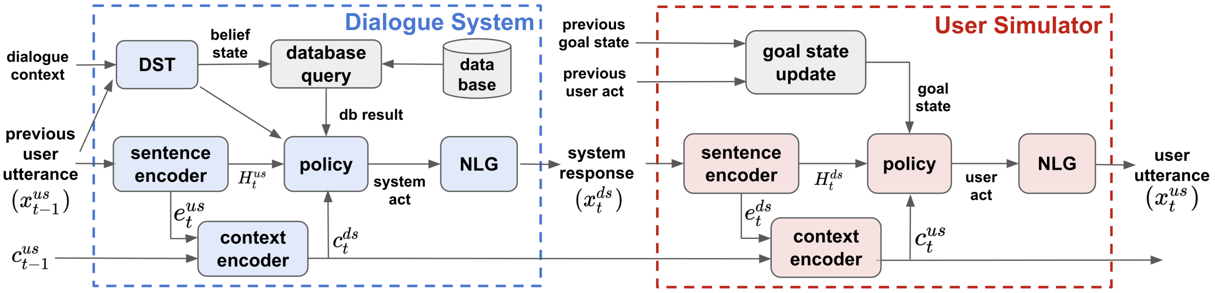

Figure 1: Overall architecture of the proposed framework, where the dialogue system (DS) and user simulator (US) discourse with each other. t denotes dialogue turn index. The context encoder is shared between the two agents.

which multiple system actions exist in the same turn.The problem is therefore formulated as a sequence generation task using an LSTM. At each decoding step, the inputs to the policy decoder are: 1) the embedding of the act token predicted at the previous step; 2) the previous hidden state; 3) the attention vector obtained by attending over the hidden states of the user utterance H_t^{us} using 2) as query; 4) the database retrieval vector; 5) the summarized belief state, which is a binary vector where each entry corresponds to a domain-slot pair. The output space contains all possible act tokens. For better modeling of the dialogue flow, the initialization of the hidden state is set to the context state c_t^{ds} obtained by the context encoder.

Natural language generation (NLG) The final task of the DS is to generate the system response, based on the predicted system dialogue act. To generate the word sequence another LSTM is used as the NLG model. At each decoding step, the previous hidden state serves as a query to attend over the hidden states of the policy decoder. The resulting attention vector and the embedding of the previous output word are the inputs to an LSTM whose output is the word sequence with *delexicalized* tokens. These *delexicalized* tokens will be replaced by retrieval results to form the final utterance.

2.2 User Simulator

As in the DS, the proposed US has a dialogue manager, an NLG model and a dialogue context encoder. However, in place of a DST to maintain the belief state, the US maintains an internal goal state to track progress towards satisfying the user goals.

Goal State The goal state is modelled as a binary vector that summarises the dialogue goal. Each entry of the vector corresponds to a domain-slot pair in the ontology. At the beginning of a dialogue, goal state entries are turned on for all slots that make up the goal. At each dialogue turn, the goal state is updated based on the previous user dialogue act. If a slot appears in the previous dialogue act, either as information from the user or as a request by the US, the corresponding entry is turned off.

Context encoding, Policy & NLG in the US These steps follow their implementations in the DS. For context encoding in the US, a sentence encoder first encodes the system response using an LSTM to obtain hidden states H_t^{ds} and sentence embedding e_t^{ds}. The context encoder takes e_t^{ds} and DS context state c_t^{ds} as inputs to produce the dialogue context state c_t^{us} which is passed to the DS at the next turn.

Also as in the DS, the policy and the NLG model of the US are based on LSTMs. The input to the policy are goal state, hidden states of the sentence encoder H_t^{ds} and context state c_t^{us}, to produce the user dialogue act, represented as in the DS as a sequence of tokens. The NLG model takes the hidden states of policy decoder as input to generate the user utterance, which is then *lexicalised* by replacing *delexicalised* tokens using the user goal.

2.3 Supervised Learning

For each dialogue turn, the ground truth dialogue acts and the output word sequences are used as supervision for both the DS and the US. The losses of the policy and the NLG model are the cross-entropy losses of the predicted sequence probability p and the ground-truth y:

$$\begin{aligned} L_{pol}^{*} &= \sum_{i=1}^{|A|} -y_{a,i}^{*} \log p_{a,i}^{*} \\ L_{nlg}^{*} &= \sum_{i=1}^{|W|} -y_{w,i}^{*} \log p_{w,i}^{*} \end{aligned} \qquad (1)$$

In the above, $*$ can be either ds or us, referring either to the DS or the US: e.g. $p^{ds}_{a,i}$ is the probability of the system act token at the i^{th} decoding step in a given turn. The ground-truth y contains both word sequences and act sequences with W and A as their lengths.

The DST annotations are also used as supervision for the DS. The loss of the DST model is defined as the sum of the cross-entropy losses for slot and value:

$$L^{ds}_{dst} = \sum_{i=1}^{|SV|} -y^{ds}_{s,i} \log p^{ds}_{s,i} - y^{ds}_{v,i} \log p^{ds}_{v,i} \qquad (2)$$

where $|SV|$ is the number of slot-value pairs in a turn; i is the decoding step index. $p^{ds}_{s,i}$ and $p^{ds}_{v,i}$ are the predictions of slot and value at the i^{th} step.

The overall losses for the DS and the US are:

$$\begin{aligned} L^{ds}(\theta^{ds}) &= L^{ds}_{dst} + L^{ds}_{pol} + L^{ds}_{nlg} \\ L^{us}(\theta^{us}) &= L^{us}_{pol} + L^{us}_{nlg} \end{aligned} \qquad (3)$$

where θ^{ds} and θ^{us} are the parameters of DS and US, respectively. The two agents are updated jointly to minimize the sum of the losses ($L^{ds} + L^{us}$). The success rate of the generated dialogues is used as the stopping criterion for supervised learning.

3 RL Optimisation of the Dialogue System and User Simulator

After the DS and US models are pre-trained from the corpus using supervised learning, they are fine-tuned using reinforcement learning (RL) based on the dialogues generated during their interactions. Two reward designs are presented after which the optimisation strategy is given.

3.1 Dialogue-Level Reward

Following common practice (El Asri et al., 2014; Su et al., 2017; Casanueva et al., 2018; Zhao et al., 2019), the success of the simulated dialogues is used as the reward, which can only be observed at the end of the dialogue. A small penalty is given at each turn to discourage lengthy dialogues. When updating the US jointly with the DS during interaction using RL, the reward is shared between two agents.

3.2 Turn-Level Reward

While the dialogue-level reward is straight-forward, it only considers the final task success rate of the dialogues and neglects the quality of the individual turns. For complex multi-domain dialogues there is a risk that this will make it difficult for the system to learn the relationship between actions and rewards. We thus propose a turn-level reward function that encapsulates the desired behavioural features of fundamental dialogue tasks. The rewards are designed separately for the US and the DS according to their characteristics.

DS Reward A good DS should learn to refine the search by requesting needs from the user and providing the correct entities, with their attributes, that the user wishes to know. Therefore at the current turn a positive reward is assigned to DS if: 1) it requests slots that it has not requested before; 2) it successfully provides an entity; or 3) is answers correctly all additional attributes requested by the user. Otherwise, a negative reward is given.

US Reward A good US should not repeatedly give the same information or request attributes that have already been provided by the DS. Therefore, a positive reward is assigned to the US if: 1) it provides new information about slots; 2) it asks new attributes about a certain entity, or 3) it replies correctly to a request from the DS. Otherwise a penalty is given.

3.3 Optimization

We apply the *Policy Gradient Theorem* (Sutton et al., 2000) to the space of (user/system) dialogue acts. In the t^{th} dialogue turn, the reward r^{ds}_t or r^{us}_t is assigned to the two agents at final last step of their generated act sequence. The return for the action at the i^{th} step is $R^*_i = \gamma^{|A^*|-i} r^*_t$, where $*$ denotes ds or us, and $|A^*|$ is the length of the act sequence of each agent. $\gamma \in [0, 1]$ is a discounting factor. The policy gradient of each turn can then be written as:

$$\nabla_{\theta^*} J^*(\theta^*) = \sum_{i}^{|A^*|} R^*_i \nabla_{\theta^*} \log p^*_{a,i} \qquad (4)$$

where $p^*_{a,i}$ is the probability of the act token at the i^{th} step in the predicted dialogue act sequence. The two agents are updated using Eqn. (4) at each turn within the entire simulated dialogue.

4 Experiments

Dataset The MultiWOZ 2.0 dataset (Budzianowski et al., 2018) is used for all experiments. It contains 10.4k dialogues with an average of 13.6 turns. Each dialogue can span up to three domains. Compared to previous benchmark corpora such as DSTC2 (Williams et al., 2016) or WOZ2.0 (Wen et al., 2017), MultiWOZ is more challenging because 1) its rich ontology contains 39 slots across 7 domains; 2) the DS can take multiple actions in a single turn; 3) the complex dialogue flow makes it difficult to hand-craft a rule-based DS or an agenda-based US. Lee et al. (2019) provided the user act labels.

Training Details The positive and negative RL rewards of Sec. 3 are tuned in the range [-5, 5] based on the dev set. The user goals employed for interaction during RL are taken from the training data without synthesizing new goals. Further training details can be found in Appendix A.1.

Evaluation Metrics The proposed model is evaluated in terms of the inform rate (Info), the success rate (Succ), and BLEU.[2] The inform rate measures whether the DS provides the correct entity matching the user goal, while the success rate further requires the system to answer all user questions correctly. Following (Mehri et al., 2019), the combined performance (Comb) is also reported, calculated as $0.5 * (\text{Info} + \text{Succ}) + \text{BLEU}$.

4.1 Interaction Quality

First, it is examined whether the proposed learning framework improves the discourse between dialogue system and user simulator. Several variants of our model are examined: 1) two agents are pre-trained using supervised learning, serving as baseline; 2) RL is used to fine-tune only the DS (RL-DS) or both agents (RL-Joint). In each RL case, we can either use rewards at the dialogue level (dial-R, Sec. 3.1) or rewards at the turn-level (turn-R, Sec. 3.2). The two trained agents interact based on 1k user goals from the test corpus, with the generated dialogues being evaluated using the metrics above.

From Table 1, we can see that the application of RL in our framework improves the success rate by more than 10% (b-e vs. a). This indicates that the DS learns through interaction with the learned US, and the designed rewards, to be better at completing the task successfully. Moreover, the joint optimisation of both the US and the DS provides dialogues with higher success rate than only optimising the DS (c&e vs. b&d). It shows that the behaviour of the US is realistic enough and diverse enough to interact with the DS, and its behavior can be improved together during RL optimisation. Finally, by comparing two reward designs, the fine-grained rewards at the turn level seem to be more effective towards guiding two agents' interaction (b&c vs. d&e), which is reasonable since they reflect more than simple success rate in terms of the nature of the tasks. Some real, generated dialogues through the interactions are provided in Appendix A.6; we note that after RL, both agents respond to requests more correctly and also learn not to repeat the same information, leading to a more successful and smooth interaction without loops in the dialogue. The corresponding error analysis of each of the agents is provided later in Sec. 4.4.1.

Model	Info	Succ
(a) Supervised Learning	69.77	58.02
(b) RL-DS w/ dial-R	81.38	70.67
(c) RL-Joint w/ dial-R	82.83	71.57
(d) RL-DS w/ turn-R	85.62	70.34
(e) RL-Joint w/ turn-R	**86.49**	**73.04**

Table 1: Quality for dialogues generated by two agents in JOUST using the test corpus user goals. BLEU is not reported since no reference sentences are available for these interactions.

Model	Info	Succ	BLEU	Comb
HRED-TS (Peng et al., 2019)	70.0	58.0	17.5	81.5
DAMD (Zhang et al., 2019)	76.3	60.4	16.6	85.0
SimpleTOD*(Hosseini-Asl et al., 2020)	84.4	70.1	15.0	92.3
SOLOIST* (Peng et al., 2020)	**85.5**	72.9	16.5	95.7
MinTL-BART* (Lin et al., 2020a)	84.9	**74.9**	**17.9**	**97.8**
JOUST Supervised Learning	77.4	66.7	17.4	89.5
JOUST RL-Joint w/ dial-R	80.6	69.4	17.5	92.5
JOUST RL-Joint w/ turn-R	**83.2**	**73.5**	**17.6**	**96.0**

Table 2: Empirical comparison with state-of-the-art dialogue systems using the predicted belief state. * indicates leveraging of pre-trained transfomer-based models.

4.2 Benchmark Results

We conduct experiments on the official test set for comparison to existing end-to-end DSs. The trained DS is used to interact with the fixed test corpus following the same setup of Budzianowski et al. (2018). Results are reported using a predicted belief state (Table 2) and using an oracle belief state (Table 3). In general, we can observe similar performance trends as in Sec. 4.1 with RL optimization

[2] For a fair comparison to previously proposed models, the same evaluation script provided by the MultiWOZ organizers `https://github.com/budzianowski/multiwoz` is used and the official data split for train/dev/test is followed.

Model	Info	Succ	BLEU	Comb
SimpleTOD*(Hosseini-Asl et al., 2020)	88.9	67.1	16.9	94.9
MoGNet (Pei et al., 2020)	85.3	73.3	20.1	99.4
ARDM* (Wu et al., 2019)	87.4	72.8	20.6	100.7
DAMD (Zhang et al., 2019)	89.2	77.9	18.6	102.2
SOLOIST* (Peng et al., 2020)	89.6	**79.3**	18.3	102.5
PARG (Gao et al., 2020)	91.1	78.9	18.8	103.8
MarCo* (Wang et al., 2020)	**92.3**	78.6	**20.0**	**105.5**
JOUST Supervised Learning	88.5	79.4	18.3	102.3
JOUST RL-Joint w/ dial-R	93.9	85.7	16.9	106.7
JOUST RL-Joint w/ turn-R	**94.7**	**86.7**	**18.7**	**109.4**

Table 3: Empirical comparison with state-of-the-art dialogue systems using oracle belief state. * indicates leveraging of pre-trained transfomer-based models.

of our model. Joint learning of two agents using RL with the fine-grained rewards reaches the best combined score and success rate. This implies that the exploration of more dialogue states and actions in the simulated interactions reinforces the behaviors that lead to higher success rate, and that these generalise well to unfamiliar states encountered in the test corpus.

Our best RL model produces competitive results in Table 2 when using predicted belief state, and can further outperform the previous work in Table 3 when using oracle belief state. Note that we do not leverage the powerful pre-trained transformer-based models like SOLOIST or MinTL-BART model. We found that with RL optimisation, our LSTM-based models can still perform competitively. In terms of DS model structure, the most similar work would be the DAMD model. The performance gain found in comparing "JOUST Supervised Learning" to DAMD is partially due to the better performance of our DST model.[3]

We also conduct experiments using only 50% of the training data for supervised learning to verify the efficacy of the proposed method under different amounts of data. As shown in Table 4, it is observed that our method also improves the model upon supervised learning when trained with less data and the improvements are consistent with the complete data scenario.

4.3 Transfer Learning

In this section, we demonstrate the capability of transfer learning of the proposed framework under two low-resource setups: Domain Adaptation and Single-to-Multiple Domain Transfer. Two fine-tuning methods are adopted: the straightforward fine-tuning without any constraints (Naive) and elastic weight consolidation (EWC) (Kirkpatrick et al., 2017). We show that the proposed RL can be further applied to both methods and produces significantly improved results. Here we experiment the best RL variants using turn-level rewards (same as (e) in Table 1).

Model	Info.	Succ.	BLEU	Comb.
Belief State = Predicted				
Supervised Learning	70.37	55.43	17.29	80.19
RL-Joint w/ turn-R	**74.83**	**60.60**	**17.41**	**85.12**
Belief State = Oracle				
Supervised Learning	89.67	74.5	16.96	99.04
RL-Joint w/ turn-R	**94.27**	**81.47**	**17.20**	**105.06**

Table 4: Results of JOUST using 50% training data in supervised learning.

Domain Adaptation In these experiments, each of five domains is selected as the target domain. Taking the *hotel* domain for example, 300 dialogues[4] involving the hotel domain are sampled from the training corpus as adaptation data. The rest of the dialogues, not involving the hotel domain, form the source data. Both the DS and the US are first trained on the source data (Source), and then fine-tuned on the limited data of the target domain (Naive, EWC). Afterwards, the pair of agents is trained in interaction using the proposed RL training regime (+RL).

Results in the form of the combined score are given in Table 5 (corresponding success rates are provided in Appendix A.5). As expected, models pre-trained on source domains obtain low combined scores on target domains. Fine-tuning using Naive or EWC method significantly bootstraps the systems, where the regularization in EWC benefits more for the low-resource training. By applying our proposed framework to the two sets of fine-tuned models, the performance can be further improved by 7-10% in averaged numbers, with both predicted and oracle belief states. This indicates that through the interaction with the US, the DS is not constrained by having seen only a very limited amount of target domain data, and that it can learn effectively from the simulated dialogues using the simple reward structure (the RL learning curve is presented in Sec. 4.4.3). With a better initialization points such as EWC models, the models can learn from a higher quality interaction and produce better results (EWC+RL vs Naive+RL). On aver-

[3] In correspondence, the DAMD authors report a DST model with joint accuracy of ca. 35%, while ours is 45%.

[4] For each domain, 300 dialogues accounts for 10% of all target-domain data. Refer to Appendix A.2 for data statistics.

Model	Restaurant	Hotel	Attraction	Train	Taxi	Avg.
Belief State = Predicted						
Source	21.1	28.6	25.2	59.6	48.7	36.6
Naive	46.7	56.2	66.1	68.5	66.3	60.8
EWC	56.7	58.2	71.6	69.3	78.7	66.9
Naive+RL	57.0	66.8	72.5	**72.3**	75.4	68.8
EWC+RL	**64.6**	**67.8**	**75.8**	71.6	**87.6**	**73.5**
Belief State = Oracle						
Source	33.2	40.1	34.3	70.7	55.4	46.7
Naive	85.6	84.2	77.9	96.7	93.4	87.5
EWC	84.1	85.1	89.8	101.7	97.5	91.6
Naive+RL	**97.6**	99.2	88.5	104.0	103.4	98.5
EWC+RL	97.5	**100.7**	**96.0**	**104.9**	**106.3**	**101.1**

Table 5: Combined scores in domain adaptation. 300 dialogues are used for each target domain adaptation.

age, the final performance obtained by EWC+RL model doubles that of Source model, which demonstrates the efficacy of the proposed method in domain adaptation.

Single-to-Multiple Domain Transfer Another transfer learning scenario is investigated where only limited multi-domain data is accessible but sufficient single-domain dialogues are available. This setup is based on a practical fact that single-domain dialogues are often easier to collect than multi-domain ones. All single-domain dialogues in the training set form the source data. For each target multi-domain combination, 100 dialogues[5] are sampled as adaptation data. As before, the DS and the US are first pre-trained on the source data and then fine-tuned on the adaptation data. Afterwards, two agents improve themselves through interaction. The models are tested using the multi-domain dialogues of the test corpus.

Results in the form of the combined score are given in Table 6 (refer to Appendix A.5 for success rates). Although the Source models capture individual domains, they cannot manage the complex flow of multi-domain dialogues and hence produce poor combined scores, with worst results on combinations of three domains. Fine-tuning improves performance significantly, as the systems learn to transition between domains in the multi-domain dialogue flow. Finally, applying our RL optimization further increases the performance by 6-9% on average. This indicates that the dialogue agents can learn more complicated policies through exploring more dialogue states and actions while interacting with user simulator. We analyse the sources of improvements in the following section.

Model	H+T	R+T	A+T	A+H+X	H+R+X	A+R+X	Avg.
Belief State = Predicted							
Source	46.0	55.4	34.3	22.0	26.6	19.9	34.0
Naive	57.2	69.2	65.0	40.3	36.0	42.8	51.7
EWC	57.4	72.1	66.1	43.7	39.0	45.0	53.9
Naive+RL	63.2	74.4	**68.4**	**47.4**	42.7	**48.7**	57.5
EWC+RL	**64.7**	**77.6**	67.6	46.6	**43.2**	48.5	**58.0**
Belief State = Oracle							
Source	82.3	93.3	76.2	36.8	55.4	42.4	64.4
Naive	88.8	98.4	85.9	72.2	79.8	76.7	83.6
EWC	95.5	96.9	89.6	70.0	81.5	79.6	85.5
Naive+RL	99.7	**104.3**	92.0	80.6	**97.2**	**89.3**	93.9
EWC+RL	**100.2**	103.0	**93.9**	**82.6**	95.0	89.2	**94.0**

Table 6: Combined scores in single-to-multiple domain transfer where 100 dialogues on each target scenario are used for adaptation. R, H, A, T, X represent Restaurant, Hotel, Attraction, Train, Taxi domain.

Model	Dialogue System		User Simulator	
	Miss Ent.	Wrong Ans.	Rep. Att.	Miss Ans.
Naive	17.59	36.99	10.12	47.27
Naive+RL	**2.73**	**9.54**	**1.47**	**32.60**

Table 7: Error analysis (%) of the US and the DS agents averaged over 5 adaptation domains. Lower is better.

4.4 Analysis

4.4.1 Error Analysis

We first examine the behavior of the US and the DS to understand the improved success rate in transfer learning. The models are those of Table 5 and are examined after fine-tuning using Naive method (Naive) and then after reinforcement learning (Naive+RL). For the DS, the rates of missing entities (Miss Ent.) and of wrong answers (Wrong Ans.) are reported. For the US, rates of repetitions of attributes (Rep. Att.) and of missing answers (Miss Ans.) are reported. The results shown in Table 7 are averaged over the five adaptation domains[6]. We see that with RL optimisation the errors made by the two agents are reduced significantly. Notably, the user model learns not to repeat the information already provided and attempts to answer more of the questions from the dialogue agent. These are the behaviors the reward structure of Sec. 3.2 are intended to encourage, and they lead to more successful interactions in policy learning.

4.4.2 Exploration of States and Actions

We now investigate whether our framework encourages exploration through increased interaction in transfer learning. We report the number of unique belief states in the training corpus and in the dialogues generated during RL interaction, as well as the unique action sequences per state that each

[5] There are 6 types of domain combinations in MultiWOZ, as shown in Table 6. For each multi-domain combination, 100 dialogues accounts for 11% of its multi-domain data.

[6] Results for each domain can be found in Appendix A.3.

	Domain adaptation		Single-to-Multiple	
	states	actions	states	actions
Corpus	614	3.34	223	3.61
Interact.	**1425**	**6.22**	399	**15.33**

Table 8: Number of unique dialogue states and average dialogue actions per state in the training corpus and in the RL interactions in two transfer learning setups.

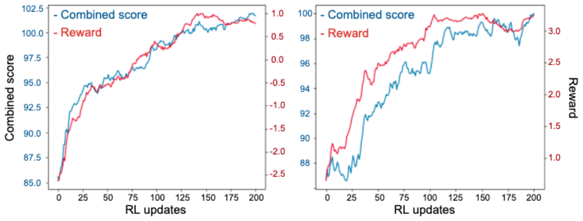

Figure 2: Learning curves observed on the dev set during RL optimization. Two domain adaptation cases are presented, with restaurant (left) and hotel (right) as target domain respectively.

agent predicts.

As shown in Table 8, the DS encounters more states in interaction with the US and also takes more unique actions in reinforcement learning relative to what it sees in supervised learning. In this way the DS considers additional strategies during the simulated training dialogues, with the opportunity to reach better performance even with only limited supervised data. Detailed results for each adaptation case are provided in Appendix A.4.

4.4.3 RL Learning Curve

Here we show that the designed reward structure is indeed a useful objective for training. Figure 2 shows learning curves of the model performance and the received (turn-level) rewards during RL training. The two examples are from the domain adaptation experiments in Sec. 4.3, where restaurant (left) and hotel (right) are the target domain. We can see that both the reward value and model performance are consistently improved during RL, and their high correlation verifies the efficacy of the proposed reward design for training task-oriented dialogue systems.

4.5 Human Evaluation

The human assessment of dialogue quality is performed to confirm the improvements of the proposed methods. 400 dialogues, generated by the two trained agents, are evaluated by 14 human assessors. Each assessor is shown a comparison of two dialogues where one dialogue is generated by the models using supervised learning (SL) and another is generated by the models after RL optimization. Note that here we are evaluating the performance gain during interactions between two agents (Sec. 4.1), instead of the gain in benchmark results by interacting with the static corpus (Sec. 4.2). This is why the baseline is our SL model instead of the existing state-of-the-art systems.

Win Ratio (%)	SL	RL
DS Success	26.0	**74.0**
US Human-like	29.5	**70.5**
Dialogue Flow	21.0	**79.0**

Table 9: Human assessment of the system quality under supervised learning and reinforcement learning.

The assessor offers judgement regarding:

- Which dialogue system completes the task more successfully (**DS Success**)?
- Which user simulator behaves more like a real human user (**US Human-like**)?
- Which dialogue is more natural, fluent and efficient (**Dialogue Flow**)?

The results with relative win ratio, shown in Table 9, are consistent with the automatic evaluation. With the proposed RL optimisation, the DS is more successful in dialogue completion. More importantly, joint optimisation of the US is found to produce more human-like behavior. The improvement under the two agents leads to a more natural and efficient dialogue flow.

5 Related Work

In the emerging field of end-to-end DSs, in which all components of a system are trained jointly (Liu and Lane, 2017a; Wen et al., 2017; Lei et al., 2018). RL methods have been used effectively to optimize end-to-end DSs in (Dhingra et al., 2017; Liu et al., 2017; Zhao et al., 2019), although using rule-based USs or a fixed corpus for interaction. Recent works utilise powerful transformers such as GPT-2 (Peng et al., 2020; Hosseini-Asl et al., 2020) or T5 (Lin et al., 2020b) for dialogue modeling and reach state-of-the-art performance; however, the area of having a user simulator involved during training is unexplored. By comparison, this work uses a learned US as the environment for RL. The two agents we propose are able to generate abundant high-quality dialog examples and they can be extended easily to unseen domains. By utilizing an interactive environment instead of a fixed corpus, more dialogue strategies are explored and more dialogue states are visited.

There have been various approaches to building USs. In the research literature of USs, one line of research is rule-based simulation such as the agenda-based user simulator (ABUS) (Schatzmann and Young, 2009; Li et al., 2016). The ABUS's structure is such that it has to be re-designed for different tasks, which presents challenges in shifting to new scenarios. Another line of work is data-driven modelling. El Asri et al. (2016) modelled user simulation as a seq2seq task, where the output is a sequence of user dialogue acts the level of semantics. Gur et al. (2018) proposed a variational hierarchical seq2seq framework to introduce more diversity in generating the user dialogue act. Kreyssig et al. (2018) introduced the Neural User Simulator (NUS), a seq2seq model that learns the user behaviour entirely from a corpus, generates natural language instead of dialogue acts and possesses an explicit goal representation. The NUS outperformed the ABUS on several metrics. Kreyssig (2018) also compared the NUS and ABUS to a combination of the ABUS with an NLG component. However, none of these prior works are suitable for modelling complex, multi-domain dialogues in an end-to-end fashion. By contrast, the user model proposed here consumes and generates text and so can be directly employed to interact with the DS, communicating via natural language.

The literature on joint optimization of the DS and the US is line of research most relevant to our work. Takanobu et al. (2020) proposed a hybrid value network using MARL (Lowe et al., 2017) with role-aware reward decomposition used in optimising the dialogue manager. However, their model requires separate NLU/NLG models to interact via natural language, which hinders its application in the transfer learning to new domains. Liu and Lane (2017b); Papangelis et al. (2019) learn both the DS and the US in a (partially) end-to-end manner. However, their systems are designed for the single-domain dataset (DSTC2) and cannot handle the complexity of multi-domain dialogues: 1) their models can only predict one dialogue act per turn, which is not sophisticated enough for modelling multiple concurrent dialogue acts; 2) the simple DST components cannot achieve satisfactory performance in the multi-domain setup; 3) the user goal change is not modelled along the dialogue proceeds, which we found in our experiments very important for learning complex behaviors of user simulators. Relative to these three publications, this paper focuses on joint training of two fully end-to-end agents that are able to participate in complex multi-domain dialogues. More importantly, it is shown that the proposed framework is highly effective for transfer learning, which is a novel contribution relative to previous work.

6 Conclusion and Future Work

We propose a novel joint learning framework of training both the DS and the US for complex multi-domain dialogues. Under the low-resource scenarios, the two agents can generate more dialogue data through interacting with each other and their behaviors can be significantly improved using RL through this self-play strategy. Two types of reward are investigated and the turn-level reward benefits more due to its fine-grained structure. Experiments shows that our framework outperforms previously published results on the MultiWOZ dataset. In two transfer learning setups, our method can further improves the well-performed EWC models and bootstraps the final performance largely. Future work will focus on improving the two agents' underlying capability with the powerful transformer-based models.

Acknowledgements

Bo-Hsiang Tseng is supported by Cambridge Trust and the Ministry of Education, Taiwan. Florian Kreyssig is funded by an EPSRC Doctoral Training Partnership Award. This work has been performed using resources provided by the Cambridge Tier-2 system operated by the University of Cambridge Research Computing Service (http://www.hpc.cam.ac.uk) funded by EPSRC Tier-2 capital grant EP/P020259/1.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Iñigo Casanueva, Paweł Budzianowski, Stefan Ultes, Florian Kreyssig, Bo-Hsiang Tseng, Yen-Chen Wu, and Milica Gašić. 2018. Feudal dialogue management with jointly learned feature extractors. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*.

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2017. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–495, Vancouver, Canada. Association for Computational Linguistics.

Layla El Asri, Jing He, and Kaheer Suleman. 2016. A sequence-to-sequence model for user simulation in spoken dialogue systems. In *Proceedings of the 17th Annual Conference of the International Speech Communication Association*, San Francisco.

Layla El Asri, Romain Laroche, and Olivier Pietquin. 2014. Task completion transfer learning for reward inference. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation. *ACL*.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks*, pages 799–804. Springer.

Izzeddin Gur, Dilek Z. Hakkani-Tür, Gökhan Tür, and Pararth Shah. 2018. User modeling for task oriented dialogues. *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 900–906.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Florian Kreyssig. 2018. Deep learning for user simulation in a dialogue system. Master's thesis, University of Cambridge, June.

Florian Kreyssig, Iñigo Casanueva, Paweł Budzianowski, and Milica Gašić. 2018. Neural user simulation for corpus-based policy optimisation of spoken dialogue systems. In *Proc. SIGDIAL*, Melbourne.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Zheng Zhang, Yaoqin Zhang, Xiang Li, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, and Jianfeng Gao. 2019. ConvLab: Multi-domain end-to-end dialog system platform. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 64–69, Florence, Italy. Association for Computational Linguistics.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Melbourne, Australia. Association for Computational Linguistics.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 733–743, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Xiujun Li, Zachary Chase Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016. A user simulator for task-completion dialogues. *ArXiv*, abs/1612.05688.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020a. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online. Association for Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020b. Mintl: Minimalist transfer learning for task-oriented dialogue systems. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 3391–3405.

Bing Liu and Ian Lane. 2017a. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. In *Interspeech 2017*.

Bing Liu and Ian Lane. 2017b. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489.

Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2017. End-to-end optimization of task-oriented dialogue model with deep reinforcement learning. In *NIPS Workshop on Conversational AI*.

Ryan Lowe, YI WU, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6379–6390. Curran Associates, Inc.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 165–177, Stockholm, Sweden. Association for Computational Linguistics.

Alexandros Papangelis, Yi-Chia Wang, Piero Molino, and Gokhan Tur. 2019. Collaborative multi-agent dialogue model training via reinforcement learning. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 92–102, Stockholm, Sweden. Association for Computational Linguistics.

Jiahuan Pei, Pengjie Ren, Christof Monz, and Maarten de Rijke. 2020. Retrospective and prospective mixture-of-generators for task-oriented dialogue response generation. *ECAI*.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

Shuke Peng, Xinjing Huang, Zehao Lin, Feng Ji, Haiqing Chen, and Yin Zhang. 2019. Teacher-student framework enhanced multi-domain dialogue generation. *arXiv preprint arXiv:1908.07137*.

Jost Schatzmann and Steve J. Young. 2009. The hidden agenda user simulation model. *IEEE Transactions on Audio, Speech, and Language Processing*, 17:733–747.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3776–3783.

Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gašić, and Steve Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 147–157, Saarbrücken, Germany. Association for Computational Linguistics.

Richard S Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*.

Ryuichi Takanobu, Runze Liang, and Minlie Huang. 2020. Multi-agent task-oriented dialog policy learning with role-aware reward decomposition. In *ACL*.

Kai Wang, Junfeng Tian, Rui Wang, Xiaojun Quan, and Jianxing Yu. 2020. Multi-domain dialogue acts and response co-generation. *arXiv preprint arXiv:2004.12363*.

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*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Jason D. Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue Discourse*, 7:4–33.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2019. Alternating recurrent dialog model with large-scale pre-trained language models. *arXiv preprint arXiv:1910.03756*.

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

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2019. Task-oriented dialog systems that consider multiple appropriate responses under the same context. *AAAI*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1208–1218.

A Appendices

A.1 Training Details

Both the DS and the US are trained in an end-to-end fashion using the Adam optimizer. The sizes of the embedding and of the hidden layers are set to 300. During supervised training, the batch size is 100 and the learning rate is 0.001, while during RL, 10 is used as the batch size and 0.0001 as the learning rate for stability. We set the discounting factor γ to 1. The computing infrastructure used is Linux 4.4.0-138-generic x86_64 with the NVIDIA GPU GTX-1080. Average run time per model using 100% training data is around 6 hours. Model parameters is around 11M in total.

The turn-level rewards used for the best models in benchmark results are reported in Table 10 below. All rewards are tuned based on the combined score of the validation performance averaged over three seeds. As for dialogue-level rewards, a positive reward 1.0 will be given if a dialogue is successful.

Model	**Rewards on DS**			**Rewards on US**		
	r^{req}	r^{pro}	r^{ans}	r^{req}	r^{inf}	r^{ans}
RL-DS	0, -1	0, -5	2.5, -5	0, 0	0, 0	0, 0
RL-Joint	0, -1	0, -5	2.5, -5	0, -1	1, -1	1, -1

Table 10: The configuration of turn-level rewards for each best model in the reported benchmark results. Each reward has positive and negative values.

A.2 Details of Dataset

As noted in the paper, we follow the original split of the MultiWOZ dataset and the number of dialogues for train/dev/test split is 8420/1000/1000. Data statistics of the number of dialogues in the two transfer learning scenarios are provided in Tables 11 and 12.

Data	**Restaurant**	**Hotel**	**Attraction**	**Train**	**Taxi**
Train	300	300	300	300	300
Dev	438	415	400	484	206
Test	437	394	396	495	195

Table 11: Number of dialogues of the splits in each domain adaptation

Data	**H+T**	**R+T**	**A+T**	**A+H+X**	**H+R+X**	**A+R+X**
Train	100	100	100	100	100	100
Dev	149	157	148	110	100	131
Test	144	155	163	92	91	129

Table 12: Number of dialogues of the splits in each scenario in single-to-multi domain transfer learning.

A.3 Error Analysis

The error analysis of each domain adaptation cases are provided in Tables 13 and 14.

Model	**Restaurant**	**Hotel**	**Attraction**	**Train**	**Taxi**	**Average**
Missing provision rate (%)						
SL	14.84	24.74	29.08	8.61	10.68	17.59
RL	**1.45**	**2.33**	**7.83**	0.41	1.62	**2.73**
Missing answer rate (%)						
SL	44.39	61.52	32.33	11.95	34.78	36.99
RL	**5.64**	20.04	10.35	2.58	**9.09**	**9.54**

Table 13: Error analysis on dialogue system on each domain in terms of two behaviors. Lower the better.

Model	**Restaurant**	**Hotel**	**Attraction**	**Train**	**Taxi**	**Average**
Repeat inform rate (%)						
SL	8.45	9.15	16.81	9.69	6.48	10.12
RL	**1.84**	**1.28**	**0.70**	**1.47**	**2.05**	**1.47**
Missing answer rate (%)						
SL	27.84	56.41	64.68	35.41	52.00	47.27
RL	**24.44**	**40.74**	**22.95**	**26.44**	**48.42**	**32.60**

Table 14: Error analysis on user simulator on each domain in terms of two behaviors. Lower the better.

A.4 Exploration

The detailed numbers of explored dialogue states and the average of unique dialogue actions per state in each case of two transfer learning scenarios are provided in Tables 15 and 16.

Model	**Restaurant**	**Hotel**	**Attraction**	**Train**	**Taxi**	**Average**
Number of dialogue states						
SL	514	726	545	513	774	614
RL	**1458**	**1666**	1087	**601**	2313	**1425**
Average of dialogue actions per state						
SL	3.51	3.10	3.74	3.70	2.65	3.34
RL	4.92	**5.49**	**7.61**	**7.98**	5.10	**6.22**

Table 15: Number of dialogue states and average of dialogue actions per state in each domain adaptation case.

Model	**H+T**	**R+T**	**A+T**	**A+H+X**	**H+R+X**	**A+R+X**	**Average**
Number of dialogue states							
SL	250	184	118	263	352	172	223
RL	523	294	**208**	348	**636**	**383**	399
Average of dialogue actions							
SL	2.84	3.95	5.40	3.00	2.43	4.05	3.61
RL	6.98	**13.56**	**17.90**	21.03	**11.30**	**21.20**	**15.33**

Table 16: Number of dialogue states and average of dialogue actions per state in each single-to-multi domain case.

A.5 Transfer Learning

Here we provide the results in success rate in two transfer learning setups.

Model	Restaurant	Hotel	Attraction	Train	Taxi	Avg.
Belief State = Predicted						
Source	5.0	10.9	5.4	36.2	0.0	11.5
Naive	26.4	35.8	41.0	48.0	35.0	37.2
EWC	35.9	37.8	47.6	47.7	55.2	44.9
Naive+RL	36.8	46.0	46.2	**49.8**	41.4	44.1
EWC+RL	**42.3**	**47.7**	**51.9**	48.5	**63.9**	**50.8**
Belief State = Oracle						
Source	11.8	18.6	9.1	45.3	0.0	17.0
Naive	60.7	62.1	46.8	73.9	67.2	62.1
EWC	59.3	62.4	64.5	79.5	74.5	68.0
Naive+RL	73.1	76.2	58.5	82.6	81.4	74.4
EWC+RL	**73.3**	**79.3**	**70.5**	**82.8**	**84.6**	**78.1**

Table 17: Success rate in domain adaptation. 300 dialogues are used for each target domain adaptation.

Model	H+T	R+T	A+T	A+H+X	H+R+X	A+R+X	Avg.
Belief State = Predicted							
Source	30.8	40.7	15.1	5.1	8.8	4.9	17.6
Naive	40.7	48.8	37.8	16.3	15.8	19.9	29.9
EWC	41.0	50.5	42.3	19.6	17.2	20.4	31.8
Naive+RL	**47.2**	53.8	42.9	18.1	**22.0**	24.0	34.7
EWC+RL	45.8	**57.9**	**44.6**	**20.7**	20.2	**24.6**	**35.6**
Belief State = Oracle							
Source	60.4	74.6	41.3	13.4	28.2	19.9	39.6
Naive	67.1	76.6	54.4	48.2	53.9	46.3	57.7
EWC	74.5	78.1	60.3	43.8	57.9	50.4	60.8
Naive+RL	**79.6**	83.4	60.3	**57.3**	68.1	60.2	68.2
EWC+RL	79.2	**84.5**	**66.1**	55.4	**70.0**	**61.5**	**69.4**

Table 18: Success rate in single-to-multiple domain transfer where 100 dialogues on each target scenario are used for adaptation. R, H, A, T, X represent Restaurant, Hotel, Attraction, Train, Taxi domain.

A.6 Generated Dialogue Examples

Here we provides two dialogue pairs examples that are generated by the two agents' interaction using supervised learning (SL) and using RL respectively in the following pages. As seen in the Table 19 first dialogue, since the SL user model answers wrong to the system's request, the system keeps asking the unsolved question, leading to an unsuccessful dialogue with loop (highlighted in color). On the other hand, the interaction between RL models (second dialogue) is much smoother and efficient since both agents answer correctly to each other. This indicates the designed rewards are useful to improve the agents' behaviors. Similar trend can be observed in Table 20, where the user model keeps stating the same information about restaurant.

User Goal
Train: destination: cambridge; day: wednesday; arriveBy: 18:15; departure: kings lynn; request=[trainID, price, leaveAt] **Attraction**: name: club salsa; request=[address]
Dialogue generated by two agents trained using SL (Dialogue has a loop and fails as the provided train is incorrect due to the wrong departure)
00-usr: i am looking for a train that leaves on wednesday and arrives at cambridge .
01-sys: there are few trains that meet your criteria . what time would you like to leave ?
02-usr: i would like to arrive by 18:15 .
03-sys: there are few trains leaving london kings cross to cambridge on wednesday . what time would you like to leave ?
04-usr: i would like to arrive by 18:15 .
05-sys: there are few trains leaving london kings cross to cambridge . is there a particular time you would like to leave ?
06-usr: i would like to arrive by 18:15 .
07-sys: there are few trains leaving london kings cross to cambridge . is there a particular time you would like to leave ?
08-usr: i would like to arrive by 18:15 .
09-sys: there are few trains leaving london kings cross to cambridge . is there a particular time you would like to leave ?
10-usr: i would like to arrive by 18:15 .
11-sys: there are few trains leaving london kings cross to cambridge . what time would you like to leave ?
12-usr: i would like to arrive by 18:15 .
13-sys: there are few trains leaving london kings cross to cambridge . what time would you like to leave ?
14-usr: i would like to arrive by 18:15 .
15-sys: there are few trains leaving london kings cross to cambridge . what time would you like to leave ?
16-usr: i would like to arrive by 18:15 .
17-sys: there are few trains leaving london kings cross to cambridge . what time would you like to leave ?
18-usr: i would like to arrive by 18:15 .
19-sys: TR0269 arrives at 18:15 . would you like me to book it for you ?
20-usr: yes , please . i need the price and travel time .
21-sys: the train id is TR0269 , the price is 10 and the travel time is 50 .
22-usr: i am also looking for a particular attraction called club salsa .
23-sys: club salsa is a nightclub in the centre . the address is 1 station road . entrance is free . is there anything else i can help you with ?
24-usr: that is all i need . thank you for your help .
25-sys: thank you for using our system !
Dialogue generated by two agents trained using RL (Dialogue is smooth and successful)
00-usr: i am looking for a train from kings lynn on wednesday .
01-sys: what is your destination ?
02-usr: i need to arrive in cambridge by 18:15 .
03-sys: i have train TR0415 that leaves at 17:11 and arrives at 17:58 . would you like me to book it for you ?
04-usr: yes , please book 2 tickets .
05-sys: booking was successful , the total fee is 9.80 pounds payable at the station . reference number is fztwszhh . is there anything else i can help you with ?
06-usr: yes , i am also looking for a particular attraction called club salsa .
07-sys: club salsa is located at 1 station road , postcode cb12jb . is there anything else i can help you with ?
08-usr: no , that s all i need today . thank you .
09-sys: thank you for using our service . have a great day !

Table 19: Example 1: two dialogues generated by SL system and RL system respectively. The dialogue loop with repeated information is highlighted. Output utterances are lexicalised with corresponding slot values (belief state for system; user goal for user). If the system generates attributes that has not existing in the belief state, the values are randomly sampled from the ontology, e.g., the train departure "london kings cross" in the first dialogue.

User Goal
Restaurant: pricerange=expensive; area=west; request=[phone, postcode]
Train: destination: cambridge; day: friday; arriveBy: 21:30; departure: norwich; request=[price, leaveAt]
Dialogue generated by two agents trained using SL (Dialogue has a loop and fails as the involved restaurant is cheap instead of expensive)
00-usr: i am looking for a expensive restaurant in the west .
01-sys: there are few restaurant -s that meet your criteria . what type of food would you like ?
02-usr: i would like to book a table for 1 at 1 . can i get the phone number for thanh binh ?
03-sys: thanh binh phone number is 01223362456 .
04-usr: i am also looking for a restaurant called thanh binh .
05-sys: thanh binh is located at 17 Magdalene Street City Centre , phone number is 01223362456 .
06-usr: i am also looking for a restaurant called thanh binh .
07-sys: thanh binh is located at 17 Magdalene Street City Centre , phone number is 01223362456 .
08-usr: i am also looking for a restaurant called thanh binh .
09-sys: thanh binh is located at 17 Magdalene Street City Centre , phone number is 01223362456 .
10-usr: i am also looking for a restaurant called thanh binh .
11-sys: thanh binh is located at 17 Magdalene Street City Centre , phone number is 01223362456 .
12-usr: thank you . i am also looking for a restaurant called thanh binh .
13-sys: thanh binh is located at 17 Magdalene Street City Centre , phone number is 01223362456 .
14-usr: can you give me the postcode for thanh binh ?
15-sys: the postcode is cb21ab . is there anything else i can help you with ?
16-usr: i am also looking for a train from norwich to cambridge on friday .
17-sys: there are 1 trains that fit your criteria . what time would you like to leave or arrive by ?
18-usr: i need to arrive by 21:30 .
19-sys: None arrives at 21:30 . would you like me to book it for you ?
20-usr: no , but can you give me the departure time and departure time ?
21-sys: the departure time is 05:16 .
22-usr: thank you so much . that s all i need today .
23-sys: thank you for using our service . have a great day !
Dialogue generated by two agents trained using RL (Dialogue is smooth and successful)
00-usr: i am looking for a expensive restaurant in the west .
01-sys: tandoori palace is a expensive indian restaurant in the west .
02-usr: what is the phone number and postcode ?
03-sys: the phone number is 01223506055 and the postcode is cb43le .
04-usr: thank you . i also need a train from norwich to cambridge on friday .
05-sys: i can help you with that . what time would you like to leave or arrive by ?
06-usr: i need to arrive by 21:30 .
07-sys: TR8009 arrives at 21:30 .
08-usr: what is the travel time and price ?
09-sys: the travel time is 79 minutes and the price is 17.60 pounds .
10-usr: what is the departure time ?
11-sys: the train leaves at 18:16 .
12-usr: what is the train id ?
13-sys: the train id is TR8009 .
14-usr: thank you very much .
15-sys: thank you , goodbye .

Table 20: Example 2: two dialogues generated by SL system and RL system respectively. The dialogue loop with repeated information is highlighted. Output utterances are lexicalised with corresponding slot values (belief state for system; user goal for user). If the user generates attributes beyond the given user goal, the values are randomly sampled from the ontology, e.g., the restaurant name "thanh binh" in the first dialogue.