Adversarial Learning of Task-Oriented Neural Dialog Models

Bing Liu
Carnegie Mellon University
Electrical and Computer Engineering
liubing@cmu.edu

Ian Lane
Carnegie Mellon University
Electrical and Computer Engineering
Language Technologies Institute
lane@cmu.edu

Abstract

In this work, we propose an adversarial learning method for reward estimation in reinforcement learning (RL) based task-oriented dialog models. Most of the current RL based task-oriented dialog systems require the access to a reward signal from either user feedback or user ratings. Such user ratings, however, may not always be consistent or available in practice. Furthermore, online dialog policy learning with RL typically requires a large number of queries to users, suffering from sample efficiency problem. To address these challenges, we propose an adversarial learning method to learn dialog rewards directly from dialog samples. Such rewards are further used to optimize the dialog policy with policy gradient based RL. In the evaluation in a restaurant search domain, we show that the proposed adversarial dialog learning method achieves advanced dialog success rate comparing to strong baseline methods. We further discuss the covariate shift problem in online adversarial dialog learning and show how we can address that with partial access to user feedback.

1 Introduction

Task-oriented dialog systems are designed to assist user in completing daily tasks, such as making reservations and providing customer support. Comparing to chit-chat systems that are usually modeled with single-turn context-response pairs (Li et al., 2016; Serban et al., 2016), task-oriented dialog systems (Young et al., 2013; Williams et al., 2017) involve retrieving information from external resources and reasoning over multiple dialog turns. This makes it especially im-

portant for a system to be able to learn interactively from users.

Recent efforts on task-oriented dialog systems focus on learning dialog models from a data-driven approach using human-human or human-machine conversations. Williams et al. (2017) designed a hybrid supervised and reinforcement learning end-to-end dialog agent. Dhingra et al. (2017) proposed an RL based model for information access that can learn online via user interactions. Such systems assume the model has access to a reward signal at the end of a dialog, either in the form of a binary user feedback or a continuous user score. A challenge with such learning systems is that user feedback may be inconsistent (Su et al., 2016) and may not always be available in practice. Further more, online dialog policy learning with RL usually suffers from sample efficiency issue (Su et al., 2017), which requires an agent to make a large number of feedback queries to users.

To reduce the high demand for user feedback in online policy learning, solutions have been proposed to design or to learn a reward function that can be used to generate a reward in approximation to a user feedback. Designing a good reward function is not easy (Walker et al., 1997) as it typically requires strong domain knowledge. El Asri et al. (2014) proposed a learning based reward function that is trained with task completion transfer learning. Su et al. (2016) proposed an online active learning method for reward estimation using Gaussian process classification. These methods still require annotations of dialog ratings by users, and thus may also suffer from the rating consistency and learning efficiency issues.

To address the above discussed challenges, we investigate the effectiveness of learning dialog rewards directly from dialog samples. Inspired by the success of adversarial training in computer vi-

sion (Denton et al., 2015) and natural language generation (Li et al., 2017a), we propose an adversarial learning method for task-oriented dialog systems. We jointly train two models, a generator that interacts with the environment to produce task-oriented dialogs, and a discriminator that marks a dialog sample as being successful or not. The generator is a neural network based task-oriented dialog agent. The environment that the dialog agent interacts with is the user. Quality of a dialog produced by the agent and the user is measured by the likelihood that it fools the discriminator to believe that the dialog is a successful one conducted by a human agent. We treat dialog agent optimization as a reinforcement learning problem. The output from the discriminator serves as a reward to the dialog agent, pushing it towards completing a task in a way that is indistinguishable from how a human agent completes it.

In this work, we discuss how the adversarial learning reward function compares to designed reward functions in learning a good dialog policy. Our experimental results in a restaurant search domain show that dialog agents that are optimized with the proposed adversarial learning method achieve advanced task success rate comparing to strong baseline methods. We discuss the impact of the size of annotated dialog samples to the effectiveness of dialog adversarial learning. We further discuss the covariate shift issue in interactive adversarial learning and show how we can address that with partial access to user feedback.

2 Related Work

Task-Oriented Dialog Learning Popular approaches in learning task-oriented dialog systems include modeling the task as a partially observable Markov Decision Process (POMDP) (Young et al., 2013). Reinforcement learning can be applied in the POMDP framework to learn dialog policy online by interacting with users (Gašić et al., 2013). Recent efforts have been made in designing end-to-end solutions (Williams and Zweig, 2016; Liu and Lane, 2017a; Li et al., 2017b; Liu et al., 2018) for task-oriented dialogs. Wen et al. (2017) designed a supervised training end-to-end neural dialog model with modularly connected components. Bordes and Weston (2017) proposed a neural dialog model using end-to-end memory networks. These models are trained offline using fixed dialog corpora, and thus it is unknown how well

the model performance generalizes to online user interactions. Williams et al. (2017) proposed a hybrid code network for task-oriented dialog that can be trained with supervised and reinforcement learning. Dhingra et al. (2017) proposed an RL dialog agent for information access. Such models are trained against rule-based user simulators. A dialog reward from the user simulator is expected at the end of each turn or each dialog.

Dialog Reward Modeling Dialog reward estimation is an essential step for policy optimization in task-oriented dialogs. Walker et al. (1997) proposed PARADISE framework in which user satisfaction is estimated using a number of dialog features such as number of turns and elapsed time. Yang et al. (2012) proposed a collaborative filtering based method in estimating user satisfaction in dialogs. Su et al. (2015) studied using convolutional neural networks in rating dialog success. Su et al. (2016) further proposed an online active learning method based on Gaussian process for dialog reward learning. These methods still require various levels of annotations of dialog ratings by users, either offline or online. On the other side of the spectrum, Paek and Pieraccini (2008) proposed inferring a reward directly from dialog corpora with inverse reinforcement learning (IRL) (Ng et al., 2000). However, most of the IRL algorithms are very expensive to run (Ho and Ermon, 2016), requiring reinforcement learning in an inner loop. This hinders IRL based dialog reward estimation methods to scale to complex dialog scenarios.

Adversarial Networks Generative adversarial networks (GANs) (Goodfellow et al., 2014) have recently been successfully applied in computer vision and natural language generation (Li et al., 2017a). The network training process is framed as a game, in which people train a generator whose job is to generate samples to fool a discriminator. The job of a discriminator is to distinguish samples produced by the generator from the real ones. The generator and the discriminator are jointly trained until convergence. GANs were firstly applied in image generation and recently used in language tasks. Li et al. (2017a) proposed conducting adversarial learning for response generation in open-domain dialogs. Yang et al. (2017) proposed using adversarial learning in neural machine translation. The use of adversarial learning in task-oriented dialogs has not been well

studied. Peng et al. (2018) recently explored using adversarial loss as an extra critic in addition to the main reward function based on task completion. This method still requires prior knowledge of a user's goal, which can be hard to collect in practice, in defining the completion of a task. Our proposed method uses adversarial reward as the only source of reward signal for policy optimization in addressing this challenge.

3 Adversarial Learning for Task-Oriented Dialogs

In this section, we describe the proposed adversarial learning method for policy optimization in task-oriented neural dialog models. Our objective is to learn a dialog agent (i.e. the generator, G) that is able to effectively communicate with a user over a multi-turn conversation to complete a task. This can be framed as a sequential decision making problem, in which the agent generates a best action to take at every dialog turn given the dialog context. The action can be in the form of either a dialog act (Henderson et al., 2013) or a natural language utterance. We study on dialog act level in this work. Let U_k and A_k represent the user input and agent outputs (i.e. the agent act a_k and the slot-value predictions) at turn k. Given the current user input U_k, the agent estimates the user's goal and select a best action a_k to take conditioning on the dialog history.

In addition, we want to learn a reward function (i.e. the discriminator, D) that is able to provide guidance to the agent for learning a better policy. We expect the reward function to give a higher reward to the agent if the conversation it had with the user is closer to how a human agent completes the task. Output of the reward function is the probability of a given dialog being successfully completed. We train the reward function by forcing it to distinguish successful dialogs and dialogs conducted by the machine agent. At the same time, we also update the dialog agent parameters with policy gradient based reinforcement learning using the reward produced by the reward function. We keep updating the dialog agent and the reward function until the discriminator can no longer distinguish dialogs from a human agent and from a machine agent. In the subsequent sections, we describe in detail the design of our dialog agent and reward function, and the proposed adversarial dialog learning method.
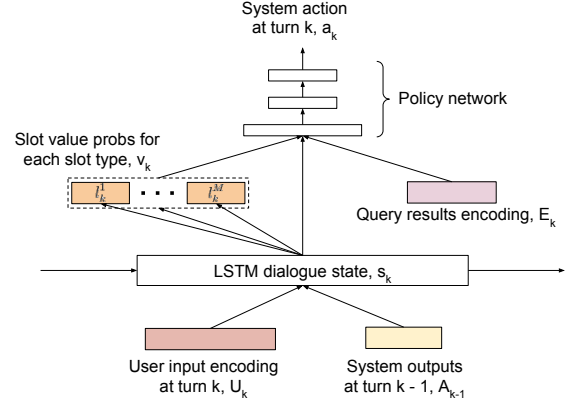

Figure 1: Design of the task-oriented neural dialog agent.

3.1 Neural Dialog Agent

The generator is a neural network based task-oriented dialog agent. The model architecture is shown in Figure 1. The agent uses an LSTM recurrent neural network to model the sequence of turns in a dialog. At each turn, the agent takes a best system action conditioning on the current dialog state. A continuous form dialog state is maintained in the LSTM state s_k. At each dialog turn k, user input U_k and previous system output A_{k-1} are firstly encoded to continuous representations. The user input can either in the form of a dialog act or a natural language utterance. We use dialog act form user input in our experiment. The dialog act representation is obtained by concatenating the embeddings of the act and the slot-value pairs. If natural language form of input is used, we can encode the sequence of words using a bidirectional RNN and take the concatenation of the last forward and backward states as the utterance representation, similar to (Yang et al., 2016) and (Liu and Lane, 2017a). With the user input U_k and agent input A_{k-1}, the dialog state s_k is updated from the previous state s_{k-1} by:

$$s_k = \text{LSTM}_\text{G}(s_{k-1}, [U_k, A_{k-1}]) \quad (1)$$

Belief Tracking Belief tracking maintains the state of a conversation, such as a user's goals, by accumulating evidence along the sequence of dialog turns. A user's goal is represented by a list of slot-value pairs. The belief tracker updates its estimation of the user's goal by maintaining a probability distribution $P(l_k^m)$ over candidate values for each of the tracked goal slot type $m \in M$. With the current dialog state s_k, the probability over

candidate values for each of the tracked goal slot is calculated by:

$$P(l_k^m \mid \mathbf{U}_{\leq k}, \mathbf{A}_{<k}) = \text{SlotDist}_m(s_k) \quad (2)$$

where SlotDist_m is a single hidden layer MLP with softmax activation over slot type $m \in M$.

Dialog Policy We model the agent's policy with a deep neural network. Following the policy, the agent selects the next action in response to the user's input based on the current dialog state. In addition, information retrieved from external resources may also affects the agent's next action. Therefore, inputs to our policy module are the current dialog state s_k, the probability distribution of estimated user goal slot values v_k, and the encoding of the information retrieved from external sources E_k. Here instead of encoding the actual query results, we encode a summary of the retrieved items (i.e. count and availability of the returned items). Based on these inputs, the policy network produces a probability distribution over the next system actions:

$$P(a_k \mid U_{\leq k}, A_{<k}, E_{\leq k}) = \text{PolicyNet}(s_k, v_k, E_k) \quad (3)$$

where PolicyNet is a single hidden layer MLP with softmax activation over all system actions.

3.2 Dialog Reward Estimator

The discriminator model is a binary classifier that takes in a dialog with a sequence of turns and outputs a label indicating whether the dialog is a successful one or not. The logistic function returns a probability of the input dialog being successful. The discriminator model design is as shown in Figure 2. We use a bidirectional LSTM to encode the sequence of turns. At each dialog turn k, input to the discriminator model is the concatenation of (1) encoding of the user input U_k, (2) encoding of the query result summary E_k, and (3) encoding of agent output A_k. The discriminator LSTM output at each step k, h_k, is a concatenation of the forward LSTM output $\overrightarrow{h_k}$ and the backward LSTM output $\overleftarrow{h_k}$: $h_k = [\overrightarrow{h_k}, \overleftarrow{h_k}]$.

Once obtaining the discriminator LSTM state outputs $\{h_1, \ldots, h_K\}$, we experiment with four different methods in combining these state outputs to generated the final dialog representation d for the binary classifier:

BiLSTM-last Produce the final dialog representation d by concatenating the last LSTM state
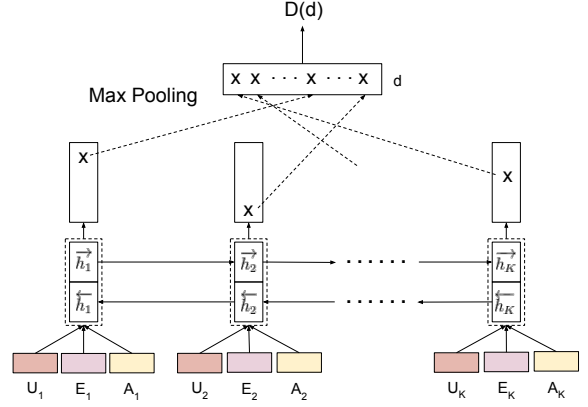

Figure 2: Design of the dialog reward estimator: Bidirectional LSTM with max pooling.

outputs from the forward and backward directions: $d = [\overrightarrow{h_K}, \overleftarrow{h_1}]$

BiLSTM-max Max-pooling. Produce the final dialog representation d by selecting the maximum value over each dimension of the LSTM state outputs.

BiLSTM-avg Average-pooling. Produce the final dialog representation d by taking the average value over each dimension of the LSTM state outputs.

BiLSTM-attn Attention-pooling. Produce the final dialog representation d by taking the weighted sum of the LSTM state outputs. The weights are calculated with attention mechanism:

$$d = \sum_{k=1}^{K} \alpha_k h_k \quad (4)$$

and

$$\alpha_k = \frac{\exp(e_k)}{\sum_{t=1}^{K} \exp(e_t)}, \quad e_k = g(h_k) \quad (5)$$

g a feed-forward neural network with a single output node. Finally, the discriminator produces a value indicating the likelihood the input dialog being a successful one:

$$D(d) = \sigma(W_o d + b_o) \quad (6)$$

where W_o and b_o are the weights and bias in the discriminator output layer. σ is a logistic function.

3.3 Adversarial Model Training

Once we obtain a dialog sample initiated by the agent and a dialog reward from the reward function, we optimize the dialog agent using REINFORCE (Williams, 1992) with the given reward.

The reward $D(d)$ is only received at the end of a dialog, i.e. $r_K = D(d)$. We discount this final reward with a discount factor $\gamma \in [0, 1)$ to assign a reward R_k to each dialog turn. The objective function can thus be written as $J_k(\theta_G) = \mathbb{E}_{\theta_G}[R_k] = \mathbb{E}_{\theta_G}\left[\sum_{t=k}^K \gamma^{t-k} r_t - V(s_k)\right]$, with $r_k = D(d)$ for $k = K$ and $r_k = 0$ for $k < K$. $V(s_k)$ is the state value function which serves as a baseline value. The state value function is a feed-forward neural network with a single-node value output. We optimize the generator parameter θ_G to maximize $J_k(\theta_G)$. With likelihood ratio gradient estimator, the gradient of $J_k(\theta_G)$ can be derived with:

$$
\begin{aligned}
\nabla_{\theta_G} J_k(\theta_G) &= \nabla_{\theta_G} \mathbb{E}_{\theta_G}[R_k] \\
&= \sum_{a_k \in \mathcal{A}} G(a_k|\cdot) \nabla_{\theta_G} \log G(a_k|\cdot) R_k \\
&= \mathbb{E}_{\theta_G}[\nabla_{\theta_G} \log G(a_k|\cdot) R_k]
\end{aligned}
\tag{7}
$$

where $G(a_k|\cdot) = G(a_k|s_k, v_k, E_k; \theta_G)$. The expression above gives us an unbiased gradient estimator. We sample agent action a_k following a softmax policy at each dialog turn and compute the policy gradient. At the same time, we update the discriminator parameter θ_D to maximize the probability of assigning the correct labels to the successful dialog from human demonstration and the dialog conducted by the machine agent:

$$
\nabla_{\theta_D}\Big[\mathbb{E}_{d \sim \theta_{demo}}[log(D(d))] + \\
\mathbb{E}_{d \sim \theta_G}[log(1 - D(d))]\Big]
\tag{8}
$$

We continue to update both the dialog agent and the reward function via dialog simulation or real user interaction until convergence.

4 Experiments

4.1 Dataset

We use data from the second Dialog State Tracking Challenge (DSTC2) (Henderson et al., 2014) in the restaurant search domain for our model training and evaluation. We add entity information to each dialog sample in the original DSTC2 dataset. This makes entity information a part of the model training process, enabling the agent to handle entities during interactive evaluation with users. Different from the agent action definition used in DSTC2, actions in our system are produced by concatenating the act

---

Algorithm 1 Adversarial Learning for Task-Oriented Dialog

1: **Required:** dialog corpus S_{demo}, user simulator U, generator G, discriminator D
2: Pretrain a dialog agent (i.e. the generator) G on dialog corpora S_{demo} with MLE
3: Simulate dialogs S_{simu} between U and G
4: Sample successful dialogs $S_{(+)}$ and random dialogs $S_{(-)}$ from $\{S_{demo}, S_{simu}\}$
5: Pretrain a reward function (i.e. the discriminator) D with $S_{(+)}$ and $S_{(-)}$ ▷ eq 8
6: **for** number of training iterations **do**
7:   **for** G-steps **do**
8:    Simulate dialogs S_b between U and G
9:    Compute reward r for each dialog in S_b with D ▷ eq 6
10:    Update G with reward r ▷ eq 7
11:   **end for**
12:   **for** D-steps **do**
13:    Sample dialogs $S_{(b+)}$ from $S_{(+)}$
14:    Update D with $S_{(b+)}$ and S_b (with S_b as negative examples)   ▷ eq 8
15:   **end for**
16: **end for**

---

and slot types in the original dialog act output (e.g. "$confirm(food = italian)$" maps to "$confirm\_food$"). The slot values (e.g. $italian$) are captured in the belief tracking outputs. Table 1 shows the statistics of the dataset used in our experiments.

# of train/dev/test dialogs	1612/506/ 1117
# of dialog turns in average	7.88
# of slot value options	
Area	5
Food	91
Price range	3

Table 1: Statistics of DSTC2 dataset.

4.2 Training Settings

We use a user simulator for our interactive training and evaluation with adversarial learning. Instead of using a rule-based user simulator as in many prior work (Zhao and Eskenazi, 2016; Peng et al., 2017), in our study we use a model-based simulator trained on DSTC2 dataset. We follow the design and training procedures of (Liu and Lane, 2017b) in building the model-based simulator. The stochastic policy used in the simula-

tor introduces additional diversity in user behavior during dialog simulation.

Before performing interactive adversarial learning with RL, we pretrain the dialog agent and the discriminative reward function with offline supervised learning on DSTC2 dataset. We find this being helpful in enabling the adversarial policy learning to start with a good initialization. The dialog agent is pretrained to minimize the cross-entropy losses on agent action and slot value predictions. Once we obtain a supervised training dialog agent, we simulate dialogs between the agent and the user simulator. These simulated dialogs together with the dialogs in DSTC2 dataset are then used to pretrain the discriminative reward function. We sample 500 successful dialogs as positive examples, and 500 random dialogs as negative examples in pretraining the discriminator. During dialog simulation, a dialog is marked as successful if the agent's belief tracking outputs fully match the informable (Henderson et al., 2013) user goal slot values, and all user requested slots are fulfilled. This is the same evaluation criteria as used in (Wen et al., 2017) and (Liu and Lane, 2017b). It is important to note that such dialog success signal is usually not available during real user interactions, unless we explicitly ask users to provide this feedback.

During supervised pretraining, for the dialog agent we use LSTM with a state size of 150. Hidden layer size for the policy network MLP is set as 100. For the discriminator model, a state size of 200 is used for the bidirectional LSTM. We perform mini-batch training with batch size of 32 using Adam optimization method (Kingma and Ba, 2014) with initial learning rate of 1e-3. Dropout ($p = 0.5$) is applied during model training to prevent the model from over-fitting. Gradient clipping threshold is set to 5.

During interactive learning with adversarial RL, we set the maximum allowed number of dialog turns as 20. A simulation is force to terminated after 20 dialog turns. We update the model with every mini-batch of 25 samples. Dialog rewards are calculated by the discriminative reward function. Reward discount factor γ is set as 0.95. These rewards are used to update the agent model via policy gradient. At the same time, this mini-batch of simulated dialogs are used as negative examples to update the discriminator.

4.3 Results and Analysis

In this section, we show and discuss our empirical evaluation results. We first compare dialog agent trained using the proposed adversarial reward to those using human designed reward and using oracle reward. We then discuss the impact of discriminator model design and model pretraining on the adversarial learning performance. Last but not least, we discuss the potential issue of covariate shift during interactive adversarial learning and show how we address that with partial access to user feedback.

4.3.1 Comparison to Other Reward Types

We first compare the performance of dialog agent using adversarial reward to those using designed reward and oracle reward on dialog success rate. Designed reward refers to reward function that is designed by humans with domain knowledge. In our experiment, based on the dialog success criteria defined in section 4.2, we *design* the following reward function for RL policy learning:

- +1 for each informable slot that is correctly estimated by the agent at the end of a dialog.

- If ALL informable slots are tracked correctly, +1 for each requestable slot successfully handled by the agent.

In addition to the comparison to human designed reward, we further compare to the case of using oracle reward during agent policy optimization. Using oracle reward refers to having access to the final dialog success status. We apply a reward of +1 for a successful dialog, and a reward of 0 for a failed dialog. Performance of the agent using oracle reward serves as an upper-bound for those using other types of reward. For the learning with adversarial rewards, we use BiLSTM-max as the discriminator model. During RL training, we normalize the rewards produced by different reward functions.

Figure 3 show the RL learning curves for models trained using different reward functions. The dialog success rate at each evaluation point is calculated by averaging over the success status of 1000 dialog simulations at that point. The pretrain baseline in the figure refers to the supervised pretraining model. This model does not get updated during interactive learning, and thus the curve stays flat during the RL training cycle. As
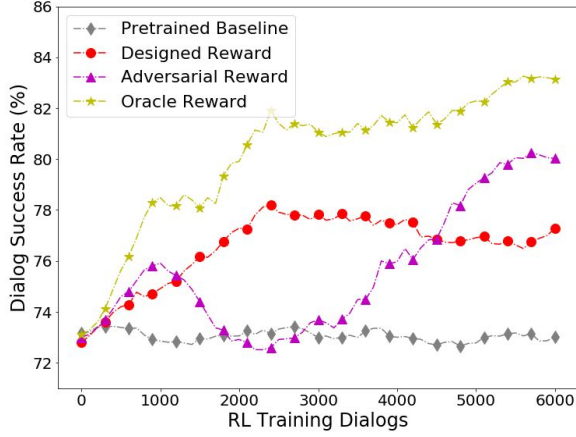
Figure 3: RL policy optimization performance comparing with adversarial reward, designed reward, and oracle reward.

shown in these curves, all the three types of reward functions lead to improved dialog success rate along the interactive learning process. The agent trained with designed reward falls behind the agent trained with oracle reward by a large margin. This shows that the reward designed with domain knowledge may not fully align with the final evaluation metric. Designing a reward function that can provide an agent enough supervision signal and also well aligns the final system objective is not a trivial task (Popov et al., 2017). In practice, it is often difficult to exactly specify what we expect an agent to do, and we usually end up with simple and imperfect measures. In our experiment, agent using adversarial reward achieves a 7.4% improvement on dialog success rate over the supervised pretraining baseline at the end of 6000 interactive dialog learning episodes, outperforming that using the designed reward (4.2%). This shows the advantage of performing adversarial training in learning directly from expert demonstrations and in addressing the challenge of designing a proper reward function. Another important point we observe in our experiments is that RL agents trained with adversarial reward, although enjoy higher performance in the end, suffer from larger variance and instability on model performance during the RL training process, comparing to agents using human designed reward. This is because during RL training the agent interfaces with a moving target, rather than a fixed objective measure as in the case of using the designed reward or oracle reward. The model performance gradually becomes stabilized when both the dialog agent and the reward model are close to convergence.

4.3.2 Impact of Discriminator Model Design

We study the impact of different discriminator model designs on the adversarial learning performance. We compare the four pooling methods described in section 3.2 in producing the final dialog representation. Table 2 shows the offline evaluation results on 1000 simulated test dialog samples. Among the four pooling methods, max-pooling on bidirectional LSTM outputs achieves the best classification accuracy in our experiment. Max-pooling also assigns the highest probability to successful dialogs in the test set comparing to other pooling methods. Attention-pooling based LSTM model achieves the lowest performance across all the three offline evaluation metrics in our study. This is probably due to the limited number of training samples we used in pretraining the discriminator. Learning good attentions usually requires more data samples and the model may thus overfit the small training set. We observe similar trends during interactive learning evaluation that the attention-based discriminator leads to divergence of policy optimization more often than the other three pooling methods. Max-pooling discriminator gives the most stable performance during our interactive RL training.

Model	Prediction Accuracy	Success Prob.	Fail Prob.
BiLSTM-last	0.674	0.580	0.275
BiLSTM-max	**0.706**	**0.588**	0.272
BiLSTM-avg	0.688	0.561	**0.268**
BiLSTM-attn	0.652	0.541	0.285

Table 2: Performance of different discriminator model design, on prediction accuracy and probabilities assigned to successful and failed dialogs.

4.3.3 Impact of Annotated Dialogs for Discriminator Training

Annotating dialogs for model training requires additional human efforts. We investigate the impact of the size of the annotated dialog samples on discriminator model training. The amount of annotated dialogs required for learning a good discriminator depends mainly on the complexity of a task. Given the rather simple nature of the slot filling based DSTC2 restaurant search task, we experiment with annotating 100 to 1000 discriminator training samples. We use BiLSTM-max discriminator model in these experiments. The adversarial
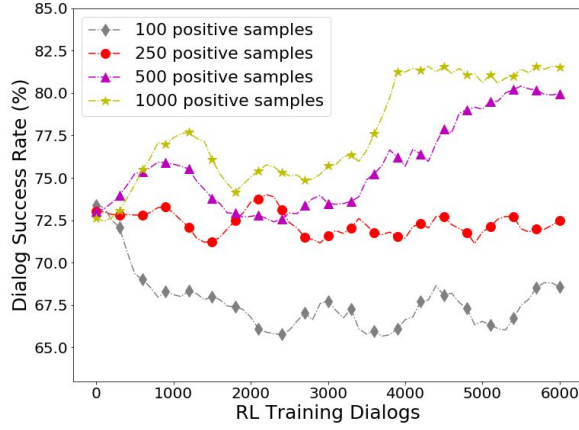
Figure 4: Impact of discriminator training sample size on RL dialog learning performance.

Figure 5: Addressing covariate shift in online adversarial dialog learning with partial access to user feedback.

RL training curves with different levels of discriminator training samples are shown in Figure 4. As these results illustrate, with 100 annotated dialogs as positive samples for discriminator training, the discriminator is not able to produce dialog rewards that are useful in learning a good policy. Learning with 250 positive samples does not lead to concrete improvement on dialog success rate neither. With the growing number of annotated samples, the dialog agent becomes more likely to learn a better policy, resulting in higher dialog success rate at the end of the interactive learning sessions.

4.3.4 Partial Access to User Feedback

A potential issue with RL based interactive adversarial learning is the covariate shift (Ross and Bagnell, 2010; Ho and Ermon, 2016) problem. Part of the positive examples for discriminator training are generated based on the supervised pretraining dialog policy before the interactive learning stage. During interactive RL training, the agent's policy gets updated. The newly generated dialog samples based on the updated policy may be equally good comparing to the initial set of positive dialogs, but they may look very different. In this case, the discriminator is likely to give these dialogs low rewards as the pattern presented in these dialogs is different to what the discriminator is initially trained on. The agent will thus be discouraged to produce such type of successful dialogs in the future with these negative rewards. To address such covariate shift issue, we design a DAgger (Ross et al., 2011) style imitation learning method to the dialog adversarial learning. We assume that during interactive learning with users, occasionally we can receive feedback from users
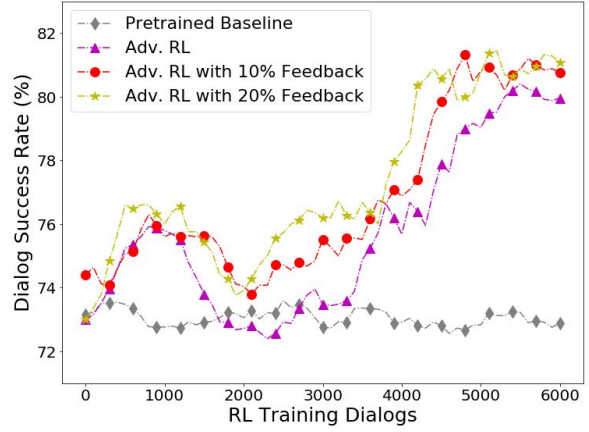
indicating the quality of the conversation they had with the agent. We then add those dialogs with good feedback as additional training samples to the pool of positive dialogs used in discriminator model training. With this, the discriminator can learn to assign high rewards to such good dialogs in the future. In our empirical evaluation, we experiment with the agent receiving positive feedback 10% and 20% of the time during its interaction with users. The experimental results are shown in Figure 5. As illustrated in these curves, the proposed DAgger style learning method can effectively improve the dialog adversarial learning with RL, leading to higher dialog success rate.

5 Conclusions

In this work, we investigate the effectiveness of applying adversarial training in learning task-oriented dialog models. The proposed method is an attempt towards addressing the rating consistency and learning efficiency issues in online dialog policy learning with user feedback. We show that with limited number of annotated dialogs, the proposed adversarial learning method can effectively learn a reward function and use that to guide policy optimization with policy gradient based reinforcement learning. In the experiment in a restaurant search domain, we show that the proposed adversarial learning method achieves advanced dialog success rate comparing to baseline methods using other forms of reward. We further discuss the covariate shift issue during interactive adversarial learning and show how we can address it with partial access to user feedback.

References

Antoine Bordes and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *International Conference on Learning Representations*.

Emily L Denton, Soumith Chintala, Rob Fergus, et al. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494.

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 ACL*.

Layla El Asri, Romain Laroche, and Olivier Pietquin. 2014. Task completion transfer learning for reward inference. *Proc of MLIS*.

Milica Gašić, Catherine Breslin, Matthew Henderson, Dongho Kim, Martin Szummer, Blaise Thomson, Pirros Tsiakoulis, and Steve Young. 2013. Online policy optimisation of bayesian spoken dialogue systems via human interaction. In *ICASSP*, pages 8367–8371. IEEE.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2013. Dialog state tracking challenge 2 & 3.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2014. The second dialog state tracking challenge. In *SIGDIAL*.

Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *ICLR*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proc. of ACL*.

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. In *Proceedings of ACL*.

Xiujun Li, Yun-Nung Chen, Lihong Li, and Jianfeng Gao. 2017b. End-to-end task-completion neural dialogue systems. *arXiv preprint arXiv:1703.01008*.

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

Bing Liu and Ian Lane. 2017b. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *Proceedings of 2017 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*.

Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *NAACL*.

Andrew Y Ng, Stuart J Russell, et al. 2000. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670.

Tim Paek and Roberto Pieraccini. 2008. Automating spoken dialogue management design using machine learning: An industry perspective. *Speech communication*, 50(8-9):716–729.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Yun-Nung Chen, and Kam-Fai Wong. 2018. Adversarial advantage actor-critic model for task-completion dialogue policy learning. In *ICASSP*.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2240.

Ivaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, and Martin Riedmiller. 2017. Data-efficient deep reinforcement learning for dexterous manipulation. *arXiv preprint arXiv:1704.03073*.

Stéphane Ross and Drew Bagnell. 2010. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635.

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 30th AAAI Conference on Artificial Intelligence (AAAI-16)*.

Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gasic, 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.

Pei-Hao Su, Milica Gašić, Nikola Mrkšić, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. On-line active reward learning for policy optimisation in spoken dialogue systems. In *Proceedings of ACL*.

Pei-Hao Su, David Vandyke, Milica Gasic, Dongho Kim, Nikola Mrksic, Tsung-Hsien Wen, and Steve Young. 2015. Learning from real users: Rating dialogue success with neural networks for reinforcement learning in spoken dialogue systems. In *Interspeech*.

Marilyn A Walker, Diane J Litman, Candace A Kamm, and Alicia Abella. 1997. Paradise: A framework for evaluating spoken dialogue agents. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 271–280. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, 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 *Proc. of EACL*.

Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *ACL*.

Jason D Williams and Geoffrey Zweig. 2016. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Zhaojun Yang, Gina-Anne Levow, and Helen Meng. 2012. Predicting user satisfaction in spoken dialog system evaluation with collaborative filtering. *IEEE Journal of Selected Topics in Signal Processing*, 6(8):971–981.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2017. Improving neural machine translation with conditional sequence generative adversarial nets. *arXiv preprint arXiv:1703.04887*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In *SIGDIAL*.