End-to-end Task-oriented Dialog Policy Learning based on Pre-trained Language Model

Anonymous ACL submission

Abstract

This paper presents our approach to dialog policy learning (DPL), which aims to determine the next system's action based on the current dialog state maintained by a dialog state tracking module. Different from previous stage-wise DPL, we propose an end-to-end DPL system to avoid error accumulation between the dialogue turns. The DPL system is deployed from two perspectives. Firstly, we consider turn-level DPL that selects the best dialog action from a predefined action set. Specifically, we proposed a dialog action-oriented BERT (DA-BERT), which integrates a new pre-training procedure named masked last action task (MLA) that encourages BERT to be dialog-aware and distill action-specific features. Secondly, we propose a word-level DPL that directly generates the dialog action. We creatively model DPL as a sequence generation model conditioned on the dialog action structure. Then GPT-2 equipped with an action structure parser module (termed as DA-GPT-2) is applied to learn the word level DPL. The effectiveness and different characteristics of the proposed models are demonstrated with the in-domain tasks and domain adaptation tasks on MultiWOZ with both simulator evaluation and human evaluation.

1 Introduction

Task-oriented dialogs that can serve users on certain tasks have increasingly attracted research efforts. Dialog policy optimization is one of the most critical tasks of dialog modeling. Recently, it has shown great potentials for using reinforcement learning (RL) based methods to formulate dialog policy learning (Li et al., 2017b; Peng et al., 2017; Lipton et al., 2016; Peng et al., 2018a; Takanobu et al., 2019; Wang et al., 2020; Li et al., 2020c).

Among these methods, dialog state tracking (DST), comprising of all information required to determine the response, is an indispensable module. However, DST inevitably accumulates errors from each module of the system. Therefore, in this paper, we establish an end-to-end DPL model without the help of DST. It takes the input as the historical dialog actions.

Meanwhile, many efforts have been made to generate the final natural language response (Bordes et al., 2016; Williams et al., 2017; Zhao et al., 2019). However, most of the previous studies treat the DPL task as either a single label classification task or a multi-label prediction task (Li et al., 2020b) based on turn-level action from pre-defined action sets, which is typically insufficient for complicated tasks. Can we get rid of this customized action list for more flexible dialog responses?

Recent pre-trained Language Models (LMs) which gather knowledge from the massive plain text show great potential for addressing the aforementioned challenges. However, due to the pre-training task and the corpus, the pre-trained LMs are task-agnostic, and cannot distinguish the characteristic of DPL when transferring knowledge. Therefore, we proposed dialog-aware pre-trained LMs, DA-BERT, and DA-GPT-2 for efficient end-to-end PDL from two perspectives of turn-level policy and word-level policy, respectively. Specifically, we proposed the **D**ialog **A**ction-oriented BERT termed as **DA-BERT**, in which a dialog act aware pre-training task based on a corpus composed of the historical annotated dialog action sequences are designed to encourage BERT to distill the act-specific features. Specifically, rather than predicting randomly masked words in the input (MLM task) and classifying whether the sentences are continuous or not (NSP task) (Devlin et al., 2018), DA-BERT is pre-trained by predicting the masked last acts in the input action sequences (termed as MLA task). Moreover, to generate more flexible dialog actions, we model dialog policy as a sequence generation problem (Sutskever et al., 2014) based on GPT-2, which takes word-level actions and is optimized with RE-

INFORCE (Williams, 1992). GPT-2 works well when pre-trained on sufficient target domain corpus, however, suffers from a poor performance without enough demonstration. To address the instabilities that arise from huge action spaces and inefficient exploration, we proposed a **D**ialog **A**ct Structure-based GPT-2, termed as **DA-GPT-2**. DA-GPT-2 is equipped with a structure parser module that draws the structural information of dialog actions to generate understandable actions with good structure. Our experiments show that DA-BERT and DA-GPT-2 achieve the best performance in turn-level DPL and word-level DPL, respectively.

To the best of our knowledge, this is the first work that strives to end-to-end DPL. Our main contributions are three-fold:

- We design the DA-BERT equipped with a new pre-training task MLA to make dialog policy learning better efficiency and transferability.
- We formulate dialog policy learning as a sequence generation problem and solve the problem by the proposed DA-GPT-2 based on a new optimization mechanism.
- We validate the effectiveness and analyze the different characteristics of the proposed models in a multi-domain task on a simulator.

2 Related Work

Dialog Policy Learning Reinforcement learning methods have been widely applied to optimize dialog policies (Young et al., 2013; Su et al., 2016, 2017; Williams et al., 2017; Peng et al., 2017, 2018a,b; Lipton et al., 2018; Li et al., 2020a; Lee et al., 2019b). Towards mitigating inefficient sampling, a lot of progress is being made in demonstration based methods on perspectives from reward designing(Brys et al., 2015; Hester et al., 2018; Li et al., 2020c), policy shaping (Cederborg et al., 2015; Griffith et al., 2013), or both (Wang et al., 2020). Different from previous methods that cast dialog policy learning as a single label classification problem, (Li et al., 2020b) proposed a sequential decision model to generate the joint action from atomic action templates (Zhu et al., 2020). (Jhunjhunwala et al., 2020) introduces a method to generate the dialog actions by ranking, filtering, and picking the top candidate sequences. However, the generation is based on fixed templated input utterances set and required a human trainer to correct the output.

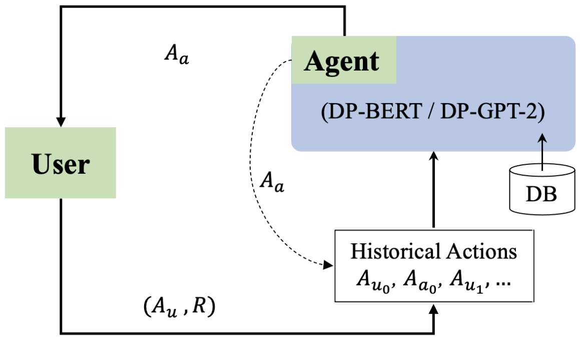


Figure 1: Illustration of the BERT/GPT-2 for dialog policy learning.

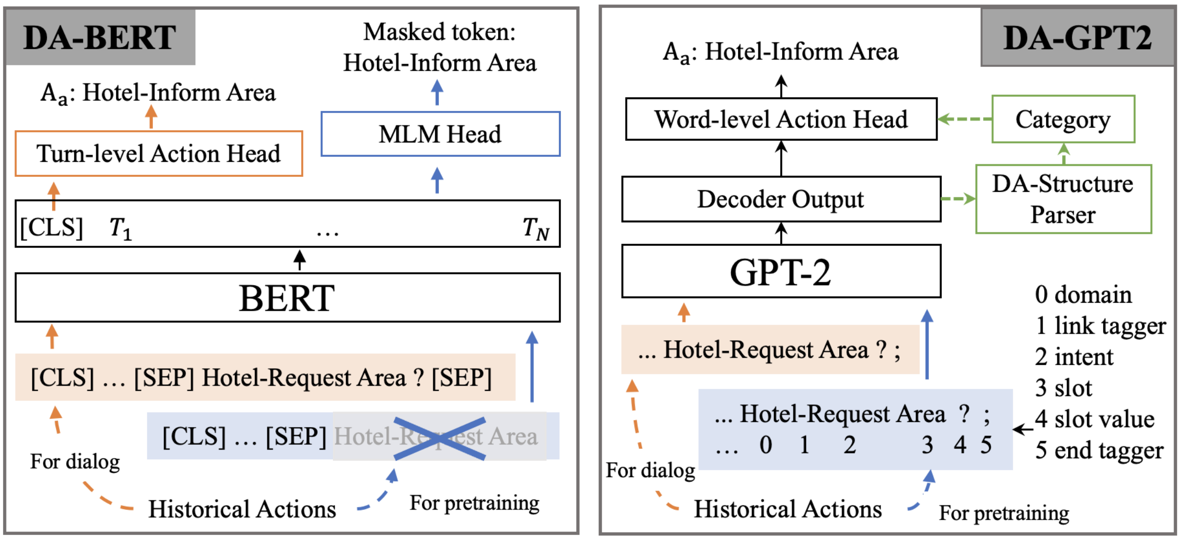
Pre-trained Language Models for Dialog Several recent studies have focused on Pre-trained Language Models for dialog, including BERT based dialog state tracking (Gulyaev et al., 2020; Chao and Lane, 2019), where BERT is applied as a context encoder and GPT-2 based dialog generation (Peng et al., 2020; Yang et al., 2020; Olabiyi and Mueller, 2019; Ham et al., 2020; Wolf et al., 2019), where GPT-2 is integrated as a response decoder. Unlike these works, we focus on investigating BERT and GPT-2 based dialog policy optimized with reinforcement learning.

Sequence Generation as Reinforcement Learning Our work is also related to recent efforts to integrate the Seq2Seq and reinforcement learning paradigms (Rennie et al., 2017; Li et al., 2017a; Keneshloo et al., 2019), where advantages of both are integrated. Our focus is on how to adapt the sequence generation model to dialog policy learning.

3 Approach

We cast the dialog policy learning problem as a Markov Decision Process and optimize the policy with deep reinforcement learning approaches. RL usually involves an interactive process (as shown in Figure 1), during which the dialog agent's behavior should choose actions that tend to increase the long-turn sum of rewards given from the user. It can learn to do this over time, by systematic trials and errors until reaches the optimal. In our setting, the dialog agent is encoded with the proposed DA-BERT or DA-GPT-2, which perceive the state and determine the next action A_a. These two models make valuable contributions to RL-based DPL.

We build the end-to-end DPL models from two perspectives. We first consider BERT-based DPL



(a) DP-BERT

(b) DP-GPT-2

Figure 2: The architecture of **D**ialog **A**ction-oriented BERT (**DA-BERT**) and the dialog action sequence generation model conditioned on **D**ialog **act** structure based on GPT-2 (**DA-GPT-2**). In this example, **DA-BERT** generates turn-level dialog action A_a based on historical actions, while **DA-GPT-2** generates word-level action based on decoder output from GPT-2 and category from structure parser.

on turn-level dialog actions, which are pre-defined as one or several concatenations of tuples containing a domain name, an intent type, and slot names, e.g. 'hotel-inform-price'. We also study word-level DPL takes a word as an action. GPT-2 is applied as the backbone to conduct the word-level policy to generate the dialog action word by word.

3.1 BERT for Turn-level DPL

We apply Deep Q-learning (Mnih et al., 2015) to optimize dialog policy for turn-level dialog action. $Q_\theta(s, a)$, approximating the state-action value function parameterized θ, is implemented based on DA-BERT as illustrated in Figure 2(a). In each turn, perceiving the state that consists of historical action sequences, DA-BERT determines the dialog action a with the generated value function $Q_\theta(\cdot|s)$. Historical action sequences are tokenized started from $[CLS]$, followed by the tokenized actions separated and ended with $[SEP]$. Then BERT's bidirectional Transformer encoder gets the final hidden states denoted $[t_0..t_n] = BERT([e_0..e_n])$ (n is the current sequence length, e_i is the embedding of the input token). The contextualized sentence-level representation t_0, is passed to an MLP module named Turn-level Action Classifier $\boldsymbol{T}$ to generate:

$$Q_\theta(s, a) = \boldsymbol{T}_a(BERT(Embed(s))) \quad (1)$$

where $Embed$ is the embedding modules of BERT, $\boldsymbol{T}_a$ denoted the a_{th} output unit of $\boldsymbol{T}$.

Based on DA-BERT, the dialog policy is trained with ϵ-greedy exploration that selects a random action with probability ϵ, or adopts a greedy policy $a = argmax_{a'} Q_\theta(s, a')$. In each iteration, $Q_\theta(s, a)$ is updated by minimizing the following square loss with stochastic gradient descent:

$$\begin{aligned} \mathcal{L}_\theta &= \mathbb{E}_{(s,a,r,s')\sim D}[(y_i - Q_\theta(s,a))^2] \\ y_i &= r + \gamma \max_{a'} Q'_\theta(s', a') \end{aligned} \quad (2)$$

where $\gamma \in [0, 1]$ is a discount factor, D is the experience replay buffer with collected transition tuples (s, a, r, s'), and $Q'(\cdot)$ is the target value function, which is only periodically updated.

3.1.1 Dialog Action-oriented Pre-training

Vanilla BERT is degraded when applied to dialog policy due to the generality of pre-training tasks and corpus. The NSP task encourages BERT to model the relationship between sentences, which may benefit natural language inference, however, biased dialog policy learning due to the inconsistency between success and continuity of sentences, e.g. discontinuous sentences can form a successful dialog. Also, the MLM task allows the word representation to fuse the left and right context, while the dialog agent is only allowed to access the left one. Considering that the ability to reason the next dialog action plays a key role for dialog policy, we replace the MLM and NSP task with a novel pre-training task: predicting masked last dialog action (MLA). MLA is based on a dialog action-oriented pre-training corpus, each piece of which is a dialog session composed of the annotated historical action sequences, for example, "*[CLS] Police-Inform Name [SEP] Police-Inform Phone Addr Post [SEP] general-thank none [SEP]*", (denoted as **sentence A**). Then we randomly cut between two consecutive actions of a session, and select the first half with masked last act as input. For example, we cut **sentence A** between the 2_{nd} and the 3_{rd} action, and mask the last act to get the input: "*[CLS] Police-Inform Name [SEP] [MASK]..[MASK]*". The label for the masked tokens is "*Police - Inform Phone Addr Post*".

The goal of MLA is to minimize the cross-entropy loss with input tokens $w_0, w_1, .., w_n$:

$$\mathcal{L}^{mla} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=n-k+1}^{n}\log \boldsymbol{p}(w_j^i|w_{0:j-1,j+1:n}^i) \tag{3}$$

where $w_{0:j-1,j+1:n}^i = w_0^i..w_{j-1}^i, w_{j+1}^i..w_n^i$, $\boldsymbol{p}$ is the language modeling head for predicting masked tokens. $w_j^i \in \{0..v-1\}$ is the label for the masked token, v is vocabulary size of BERT. m is the number of dialog sessions. n and k is the length of input and masked action sequence, respectively.

3.2 GPT-2 for Word-level DPL

For more expressive dialog actions, we follow the OpenAI GPT-2 (Radford et al., 2019) to model dialog policy as a sequence generation problem and optimize the policy with REINFORCE (Williams, 1992). Similar to DA-BERT, we first concatenate the current historical action sequence as a state, in which each action is ended with an end-of-text token '.'. Suppose the tokenized state is $s_t = [x_0..x_n]$ with length n, and the tokenized expected response is $\mathbb{X}_t = [x_{n+1}..x_{n+l}]$ with length l. The word-level dialog policy can be written as the product of a series of conditional probabilities:

$$\mathcal{P}_\varphi(\mathbb{X}_t|s_t) = \prod_{i=n+1}^{n+l}\mathcal{P}_\varphi(x_i|x_{n+1:i-1}, s_t) \tag{4}$$

where $x_{n+1:i-1} = x_{n+1}..x_{i-1}$, while φ is the parameters of the GPT-2 based policy network. Acting as an agent, GPT-2 predicts the next word and updates its internal "state" (modules of GPT-2). Upon generating the end-of-sequence token '.', the agent observes a "reward" from a user, that is, for instance, a -1 for each turn and a significant positive or negative reward indicating the status of the dialog at the end of a session. The goal of training is to minimize the negative expected reward:

$$\mathcal{L}_\varphi = -\mathbb{E}_{\mathbb{X}_t\sim\mathcal{P}_\varphi}[\sum_{t=0}^{T} r(\mathbb{X}_t)] \tag{5}$$

where $\mathbb{X}_t$ is a dialog action sequence of turn t. Practically, the expected gradient can be approximated by using a single Monte-Carlo sample $\mathbb{X} = (\mathbb{X}_0, ..., \mathbb{X}_T)$ in a dialog session with Max turn T from $\mathcal{P}_\varphi$, for each session example:

$$\nabla_\varphi\mathcal{L}_\varphi \approx -\sum_{t=0}^{T} r(\mathbb{X}_t)\nabla_\varphi \log \mathcal{P}_\varphi(\mathbb{X}_t|s_t)$$
$$= -\sum_{t=0}^{T} r(\mathbb{X}_t)\nabla_\varphi \sum_{i=n^t+1}^{n^t+l^t}\log \mathcal{P}_\varphi(x_i^t|x_{n^t+1:i-1}^t, s_t) \tag{6}$$

where n^t and l^t are the length of the current input sequence and output action sequence at turning t.

Based on the word-level dialog policy, the generated dialog action sequence is decoded by Action Decoder for final output. Action Decoder is designed to identify the domain, intent, and slot from the action sequence for GPT-2, and fill in slot value based on a database. Both BERT and GPT-2 based dialog action require action decoder to fill in slot value for final output. An action generated from GPT-2 is a sequence containing words related to domain, intent, and slot. We use a tagger "-" to indicate the linking of domain and intent, Action Decoder identifies the left word and right word of "-" as domain and intent respectively. The word behind intent and before the next domain is detected as slots.

3.2.1 DA-GPT-2

The biggest challenge of GPT-2 based dialog policy is the huge action space, which leads to many ineffective explorations. The huge action space not only reduces the learning efficiency but also may trap the RL agent into a local minimum. Besides, GPT-2 based policy model is unstable for it is prone to produce actions that cannot be decoded. Different from another sequence, the dialog action sequence is characterized by its special structure, which is reflected in that every word in the action sequence has its corresponding unique category, such as the domain name, the intent type, and the slot name.

Consequently, the decision-making process of an action sequence can be decomposed into two phases: determining the category of the next word and selecting the category-specific word. Motivated by the above observation we cast our problem in a hierarchical framework, as shown in Figure 2(b). We make the structure parser responsible for the category-level decision, and the word-level classifier determines the concrete word. The structure parser learns a hidden parameter $\mathbf{z}$ as the distribution $\mathcal{P}_\tau(z_i|s_t, x_{0:i-1})$ over word categories conditioned on the previous output tokens and the current state. We consider 5 categories of the words, $z_i \in \{0, 1, 2, 3, 4\}$ corresponding to the domain name, the intent type, the slot name, the link tagger "-", and the end token ".", respectively. While the word-level policy is the distribution of the output tokens. More specifically, the probability of a word-level action is the joint probability of the generated sequence conditioned on the current state and the category distribution:

$$\mathcal{P}_{\tau,\varphi}(\mathbb{X}_t|\mathbf{z_t}, s_t) = \prod_{i=n+1}^{n+l} \mathcal{P}_{\tau,\varphi}(x_i|x_{n+1:i-1}, \mathbf{z_i}, s_t) \tag{7}$$

where $\mathbf{z_i}$ is the category distribution for x_i, n and l is the length of the state and generated action sequence, respectively.

Dialog Action Structure Loss To encourage generating the related categories to guide word decision, structure parser is trained using the following cross-entropy loss:

$$\mathcal{L}^s_\tau = -\frac{1}{n} \sum_{i=n+1}^{n+l} \log \mathcal{P}_\tau(z_i|s_t, x_{0:i-1}) \tag{8}$$

where z_i is the expected category of x_i.

Word Loss The GPT2-based RL agent is responsible for generating dialog action sequence word by word. Besides the structure, to give the valid action sequence that can be decoded by Action Decoder, the agent should learn the accurate distribution above words for each category. To achieve that, the agent train to minimize the following word loss:

$$\mathcal{L}^w_{\tau,\varphi} = -\frac{1}{n} \sum_{i=n+1}^{n+l} \log \mathcal{P}_{\tau,\varphi}(x_i|x_{0:i-1}, \mathbf{z_i}, s_t) \tag{9}$$

We use a separate training scheme to optimize DA-GPT-2 based on REINFORCE. In each iteration, we update policy network $\mathcal{P}_{\tau,\varphi}$ with loss:

$$\mathcal{L}_{\varphi,\tau} = -\mathbb{E}_{\mathbb{X}_t \sim \mathcal{P}_{\varphi,\tau}}[\sum_{t=0}^{T} r(\mathbb{X}_t)] \tag{10}$$

For faster convergence, $\mathcal{L}^s_\tau$ and $\mathcal{L}^w_{\tau,\varphi}$ are only calculated and backward propagated for successful dialog.

3.2.2 Dialog Action Structure Pre-training

GPT-2 is pre-trained on extremely massive text data OpenWebText (Radford et al., 2019). It has demonstrated superior performance in characterizing data distribution and knowledge of the human language. To enable the guidance of categories for more accurate dialog actions, we propose to continuously pre-train GPT-2 on a large amount of annotated dialog action sequences with corresponding word categories. We first pre-process the dialog actions A into a sequence $\mathbf{A}_i$ along with the label $\mathbf{S}_i$ containing the category of each word using the following format: ($\boldsymbol{A_i}$: domain-intent slot$_1$..slot$_n$, .. $=$ $\boldsymbol{S_i}$: 0 1 2 3 3..4, ..). Here we set the category label of domain, '-', intent, slot, and '.' as $0,\ 1,\ 2,\ 3,\ 4,$ respectively. Meanwhile, we set GPT-2 with the structure parser as our backbone language model, concatenate the sequentialized dialog action A_i with its category labels S_i, and fed them into the language model. Finally, the model is trained to minimize the loss of predicting the next word and the related category.

4 Experiments and Results

We evaluate the proposed dialog policy models with a user simulator setup on MultiWoz (Budzianowski et al., 2018). Additionally, to assess the generalization capability of our approaches, we conduct domain adaptation experiments. Finally, human evaluation results are reported. The experiments do not involve the NLG part because they are held at the dialog-action level, i.e., RL agent interactives with user simulator with dialog actions. Notably, our models can be equipped with any NLG models.

4.1 Dataset

We continuously pre-train the proposed models on MultiWoz (Budzianowski et al., 2018), a large-scale fully annotated corpus of human-human conversations. Each dialog of MultiWoz is rich in annotations of dialog actions of user and system utterances. All models are only optimized on MultiWoz, which contains 9 domains, 13 intents, and 28 slots. The total size of the pre-training corpus of MultiWoz is 8434. More details of the dataset and their processing procedure are in Appendix A.

4.2 Baseline Agents

We compare the performance of the proposed DA-BERT and DA-GPT-2 with vanilla BERT, vanilla GPT-2, and several variants. Note that, our work is the first attempt to study end-to-end DPL, therefore, we do not compare the stage-wise methods (except DQN).

- **DQN** agent is trained with a deep Q-Network.
- **BERT** agent is equipped with BERT as encoder that replacing MLP in DQN.
- **DA-BERT**$_{\texttt{MWoz}}$ is our proposed agent that is pre-trained with MLA task as described in Section 3.1.1 on MultiWoz dataset.
- **GPT-2** agent is initialized with official GPT-2's pre-trained weights and optimized with policy loss $\mathcal{L}_\varphi$ as equ. 5.
- **DA-GPT-2**$_{\texttt{MWoz}}$ is our proposed agent that is based on GPT-2 and equipped with structure parser 2(b). It is pre-trained with word loss $\mathcal{L}^w_{\tau,\varphi}$ as equ. 9 and action structure loss $\mathcal{L}^s_\tau$ as equ. 8 on MultiWoz, and then optimized on $\mathcal{L}^w_{\tau,\varphi}$, $\mathcal{L}^s_\tau$, and policy loss $\mathcal{L}_\varphi$ as equ. 5.

Implementation Details We adopt BERT$_{base}$ (uncased) and DistilGPT-2 (Sanh et al., 2019) with default hyperparameters in Huggingface Transformers (Wolf et al., 2020) as the backbone language model. Turn-level Action Classifier for DA-BERT is a linear layer with 400 output units corresponding to 400 action candidates. Word-level Action Classifier for DA-GPT-2 is the sum of two linear layers of the language modeling head (Wolf et al., 2020) and structure parser (with 5 output units for 5-word categories). We set the discount factor as $\gamma = 0.9$. We apply the rule-based agent from ConvLab (Lee et al., 2019a) for warm_start. The warm_start epoch for BERT and GPT2 based agents are 1000 and 50, respectively. More details of implementation are shown in Appendix B.

4.3 User Simulator

We leverage a public available agenda-based user simulator (Zhu et al., 2020) for our experiment setup on MultiWoz (Budzianowski et al., 2018). During training, the simulator initializes with a user goal and takes system acts as input and outputs user acts with reward. The reward is set as -1 for each turn to encourage short turns and a positive reward $(2 \cdot T)$ for successful dialog or a negative reward of $-T$ for failed one, where T (set as 40) is the maximum number of turns in each dialog. A dialog is considered successful only if the agent helps the user simulator accomplish the goal and satisfies all the user's search constraints.

Table 1: The performance of different agents. Succ. denotes the final success rate, Turn and Reward are the average turn and the average reward of the whole training process, respectively.

Model	Succ.↑	Turn↓	Reward↑
DQN	0.01	19.51	-53.66
BERT	0.64	14.75	-15.47
BERT$_{\texttt{MWoz}}$	0.72	12.14	14.21
DA-BERT$_{\texttt{MWoz}}$	**0.84**	10.21	27.35
GPT-2	0.30	17.45	-23.13
GPT-2$_{\texttt{MWoz}}$	0.77	8.15	35.29
DA-GPT-2$_{\texttt{MWoz}}$	0.78	**7.71**	**37.12**

4.4 Simulator Evaluation

All agents are evaluated with the success rate (Succ.) at the end of the training, average turn (Turn), average reward (Reward).

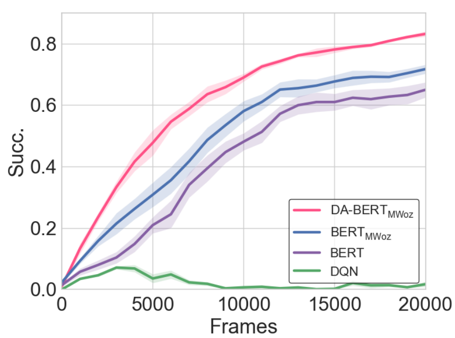


Figure 3: Comparisons on DA-BERT.

Main Results. The main simulation results are shown in Table. 1, Figure 3, and Figure 4. The results indicate that the proposed DA-GPT-2$_{\texttt{MWoz}}$ learns much faster, while DA-BERT$_{\texttt{MWoz}}$ achieves a better convergence in in-domain evaluation. The consequence is not surprising since DA-BERT$_{\texttt{MWoz}}$ selects the action from human-defined action sets, however, DA-GPT-2$_{\texttt{MWoz}}$ needs to generate its answers, which suffers from more uncertainty.

DA-BERT$_{\texttt{MWoz}}$, pre-trained with the mask last act task (MLA) on the MultiWoz corpus achieves the best Succ. (on average 0.84) with the highest learning efficiency in BERT-based models. The performance of DA-BERT$_{\texttt{MWoz}}$ reveals that our MLA pre-training task can not only encode the characteristics of dialog policy for efficiency improvement but also show better transfer abilities because dropping it BERT$_{\texttt{MWoz}}$ degrades the performance of DA-BERT$_{\texttt{MWoz}}$. Additionally, BERT is consistently the worst in BERT-based models, which is not surprising since it is only initialized with official BERT's pre-trained weights without in-domain pre-training. The generality of pre-training corpus and task, domain awareness, and knowledge transferability of BERT are poor. Furthermore, without any pre-training, DQN is consistently the worst.

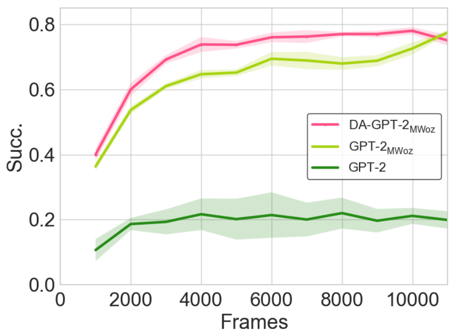


Figure 4: Comparisons on DA-GPT-2.

Besides, DA-GPT-2$_{\texttt{MWoz}}$, pre-trained on the MultiWoz corpus and optimized with both structure and word loss achieves the best Succ. (on average 0.78) with the highest learning efficiency among GPT-2 based models. DA-GPT-2$_{\texttt{MWoz}}$ learns faster and performs significantly better than GPT-2$_{\texttt{MWoz}}$ with a clear margin, which indicates the good performance of dialog action structure-based optimization and pre-training mechanism.

Finally, the comparison results of Turn and Reward are illustrated in Table. 1. It depicts that DA-GPT-2$_{\texttt{MWoz}}$ achieves the shortest average turn and highest average reward, which is consistent with the learning curves in Figure 3 and Figure 4.

4.5 Ablation Study

Effectiveness of DA-GPT-2 Components To illustrate the true source of gains of the proposed DA-GPT-2, we design an ablative setting. What can be depicted from the comparison results in Figure 5 include: 1) A combination of action structure loss and word loss is advantageous because

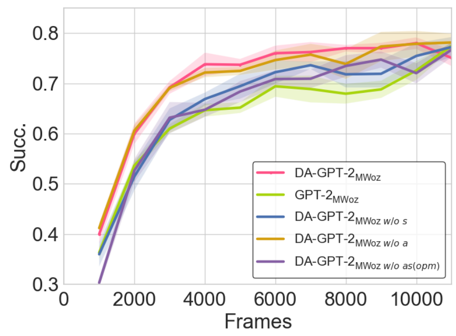


Figure 5: Comparisons on the variants of the DA-GPT-2.

removing one of them ("$w/o\ s$" or "$w/o\ a$") impairs DA-GPT-2's performance; 2) Action structure loss or word loss is also effective, indicated by the superior performance of ("$w/o\ s$" or "$w/o\ a$") compared to using only policy loss for optimization (GPT-2$_{\texttt{MWoz}}$); 3) Even if action structure loss and word loss are used in the pre-training stage but not in the in-domain training stage ("$w/o\ as(opm)$"), it can also improve the performance to some extent compared with (GPT-2$_{\texttt{MWoz}}$).

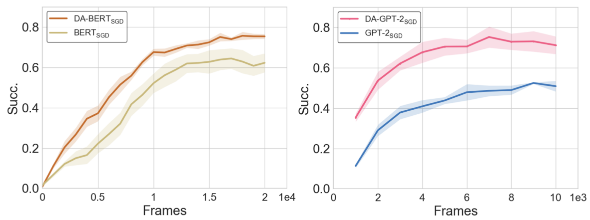


Figure 6: Comparisons of agents pre-trained on SGD corpus.

Effect of Pre-training Corpus We further test the effect of different pre-training corpus on the performance. Another corpus, Schema-Guided dialog (SGD) (Rastogi et al., 2019) is applied. It consists of over 20k annotated conversations between a human and a virtual assistant of 16 domains. More details of SGD is in Appendix A. The models are pre-trained on SGD and optimized on MultiWoz to investigate the influence of pre-training corpus. Some bullet names are explained as follows.

- DA-BERT$_{\texttt{SGD}}$ is a variant of DA-BERT$_{\texttt{MWoz}}$ which is pre-trained on SGD and trained on MultiWoz.

- DA-GPT-2$_{\texttt{SGD}}$ is a variant of DA-GPT-2$_{\texttt{MWoz}}$ which is pre-trained on SGD and optimized on MultiWoz.

The core conclusion indicated from Figure 7 is that DA-BERT and DA-GPT-2 are robust to different pre-training corpus. Firstly, MLA is beneficial for BERT DPL models even with pre-trained on different corpus because removing it $\text{BERT}_{\texttt{SGD}}$ degrades the performance of $\text{DA-BERT}_{\texttt{SGD}}$. Besides, the proposed dialog action structure parser does better in extracting the knowledge of dialog action sequence especially the structure information that is invariant over domains. As a consequence, $\text{DA-GPT-2}_{\texttt{SGD}}$ outperforming $\text{GPT-2}_{\texttt{SGD}}$.

4.6 Domain Adaptation

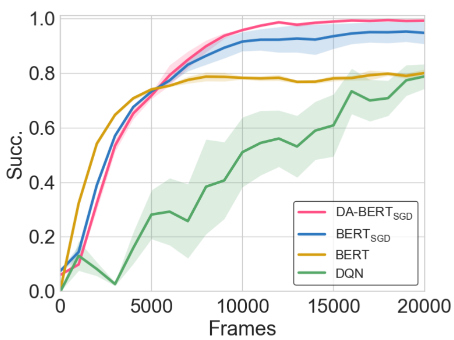


Figure 7: Comparisons on BERT based agents of domain adaptation.

To assess the ability to new task adaptation, we compare the agents that continually learn a new domain Restaurant, starting from being well trained on the other six domains (i.e. Train, Hotel, Hospital, Taxi, Police, Attraction). Figure 7 and Figure 8 show the performances of new task adaptation for turn level DPL and word-level DPL, respectively.

Firstly, though both $\text{DA-BERT}_{\texttt{SGD}}$ and $\text{BERT}_{\texttt{SGD}}$ are pre-trained on SGD additionally, $\text{BERT}_{\texttt{SGD}}$ still lags behind $\text{DA-BERT}_{\texttt{SGD}}$, showing that pre-trained with MLA task is more effective than MLM and NSP for adaptation to new domain. Meanwhile, BERT performs worse than $\text{BERT}_{\texttt{SGD}}$, which is no surprise since $\text{BERT}_{\texttt{SGD}}$'s gain from SGD. Moreover, DQN's adaptation ability is consistently the worst. However, pre-training (on the six domains) also benefits DQN to obtain a better learning efficiency.

Meanwhile, the results in Figure 8 confirm that DA-GPT-2 pre-trained and optimized with action structure loss and word loss is capable of quickly adapting to the new environment compared from $\text{DA-GPT-2}_{\texttt{SGD}}$ and GPT-2.

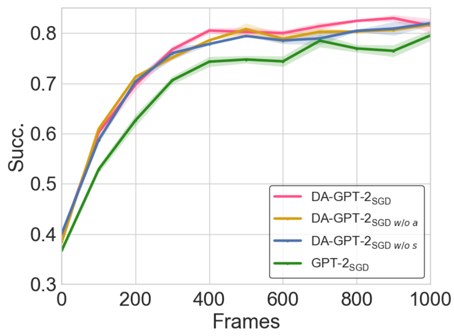


Figure 8: Comparisons on GPT-2 based agents of domain adaptation.

Table 2: Human evaluation results on BERT and GPT-2 based agents. We use models at epoch 10000 for all agents. Succ. denotes success rate

Model	Succ.↑
DQN	0.00
BERT	0.38
$\text{BERT}_{\texttt{MWoz}}$	0.58
$\text{DA-BERT}_{\texttt{MWoz}}$	0.68
GPT-2	0.20
$\text{GPT-2}_{\texttt{MWoz}}$	0.78
$\text{DA-GPT-2}_{\texttt{MWoz}}$	0.76

4.7 Human Evaluation

We further conduct a human evaluation to validate the simulation results. We choose the agents trained with 10000 epochs. Before the test, all evaluators are instructed to interact with the agents to achieve their goals. In each session, a user is assigned a goal and a randomly selected agent. The user can terminate the dialog if they think the session is must fail. At the end of each session, the user is required to judge if the dialog is a success or a failure. We collect 50 conversations for each agent. The results are illustrated in Table. 2, which is consistent with the simulation results.

5 Conclusion

In this paper, we investigate large-scale pre-trained LMs for end-to-end DPL from turn-level and word-level. Firstly, We design a new pre-training task MLA and build the DA-BERT model to improve BERT-based dialog policy learning efficiency and transferability. Besides, we propose the DA-GPT-2 accompanied by a dialog action structure-aware pre-training method to increase the flexibility of action and the richness of expression. The evaluation results show the effectiveness and indicate the different application scenarios of the proposed

DA-BERT and DA-GPT-2.

References

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

Tim Brys, Anna Harutyunyan, Halit Bener Suay, Sonia Chernova, Matthew E Taylor, and Ann Nowé. 2015. Reinforcement learning from demonstration through shaping. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Thomas Cederborg, Ishaan Grover, Charles L Isbell, and Andrea L Thomaz. 2015. Policy shaping with human teachers. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Guan-Lin Chao and Ian Lane. 2019. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. *arXiv preprint arXiv:1907.03040*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. 2013. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in neural information processing systems*, pages 2625–2633.

Pavel Gulyaev, Eugenia Elistratova, Vasily Konovalov, Yuri Kuratov, Leonid Pugachev, and Mikhail Burtsev. 2020. Goal-oriented multi-task bert-based dialogue state tracker. *arXiv preprint arXiv:2002.02450*.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2. ACL.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. 2018. Deep q-learning from demonstrations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Megha Jhunjhunwala, Caleb Bryant, and Pararth Shah. 2020. Multi-action dialog policy learning with interactive human teaching. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 290–296.

Yaser Keneshloo, Tian Shi, Naren Ramakrishnan, and Chandan K Reddy. 2019. Deep reinforcement learning for sequence-to-sequence models. *IEEE Transactions on Neural Networks and Learning Systems*.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Xiang Li, Yaoqin Zhang, Zheng Zhang, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, et al. 2019a. Convlab: Multi-domain end-to-end dialog system platform. *arXiv preprint arXiv:1904.08637*.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Zheng Zhang, Yaoqin Zhang, Xiang Li, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, and Jianfeng Gao. 2019b. 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.

Jinchao Li, Baolin Peng, Sungjin Lee, Jianfeng Gao, Ryuichi Takanobu, Qi Zhu, Minlie Huang, Hannes Schulz, Adam Atkinson, and Mahmoud Adada. 2020a. Results of the multi-domain task-completion dialog challenge. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, Eighth Dialog System Technology Challenge Workshop*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

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

Ziming Li, Julia Kiseleva, and Maarten de Rijke. 2020b. Rethinking supervised learning and reinforcement learning in task-oriented dialogue systems. *arXiv preprint arXiv:2009.09781*.

Ziming Li, Sungjin Lee, Baolin Peng, Jinchao Li, Shahin Shayandeh, and Jianfeng Gao. 2020c. Guided dialog policy learning without adversarial learning in the loop. *arXiv preprint arXiv:2004.03267*.

Zachary Lipton, Xiujun Li, Jianfeng Gao, Lihong Li, Faisal Ahmed, and Li Deng. 2018. Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zachary C Lipton, Jianfeng Gao, Lihong Li, Xiujun Li, Faisal Ahmed, and Li Deng. 2016. Efficient exploration for dialog policy learning with deep bbq networks & replay buffer spiking. *CoRR abs/1608.05081*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

Oluwatobi Olabiyi and Erik T Mueller. 2019. Dlgnet: A transformer-based model for dialogue response generation. *arXiv preprint arXiv:1908.01841*.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Yun-Nung Chen, and Kam-Fai Wong. 2018a. Adversarial advantage actor-critic model for task-completion dialogue policy learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6149–6153. IEEE.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018b. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2182–2192.

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. *arXiv preprint arXiv:1704.03084*.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. *arXiv preprint arXiv:2002.12328*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. *arXiv preprint arXiv:1707.00130*.

Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Continuously learning neural dialogue management. *arXiv preprint arXiv:1606.02689*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27:3104–3112.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 100–110, Hong Kong, China. Association for Computational Linguistics.

Huimin Wang, Baolin Peng, and Kam-Fai Wong. 2020. Learning efficient dialogue policy from demonstrations through shaping. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6355–6365.

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. *arXiv preprint arXiv:1702.03274*.

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

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

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

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2020. Ubar: Towards fully end-to-end task-oriented dialog systems with gpt-2. *arXiv preprint arXiv:2012.03539*.

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, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. *arXiv preprint arXiv:1902.08858*.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. *arXiv preprint arXiv:2002.04793*.

Table 3: The data annotation schema.

	Multiwoz	SGD
Domain	Attraction, Hospital, Booking, Hotel, Restaurant, Taxi, Train, Police, general	Restaurant, Media, Event, Music, Movie, Flight, RideSharing, RentalCar, Bus, Hotel, Service, Home, Bank, Calendar, Weather, Travel
Intent	welcome, greet, bye, reqmore, Inform, Request, Book, OfferBooked, NoBook, Recommend, NoOffer, OfferBook, Select	InformIntent, Request, Inform, Offer, RequestAlts, InformCount, Select, Confirm, Affirm, NotifySuccess, ThankYou, bye, OfferIntent, AffirmIntent, Negate, reqmore, NotifyFailure, NegateIntent
Slot	Name, none, Area, Choice, Type, Price, Addr, Leave, Food, Phone, Stars, Day, Post, Arrive, Internet, Parking, Dest, Depart, Fee, Ref, Id, People, Time, Ticket, Stay, Car, Open, Department	intent, city, Depart, Dest, Food, Name, Car, Addr, Phone, Price, count, Time, Leave, Arrive, party_size, group_size, Day, has_live_music, serves_alcohol, title, subtitles, directed_by, Type, number_of_tickets, album, artist, playback_device, year, city_of_event, People, airlines, seating_class, number_stops, passengers, refundable, Fee, is_redeye, shared_ride, number_of_riders, approximate_ride_duration, transfers, travelers, Stars, has_laundry_service, offers_cosmetic_services, is_unisex, Area, number_of_baths, number_of_beds, rent, pets_allowed, furnished, balance, amount, number_of_rooms, pets_welcome, Stay, has_wifi, temperature, precipitation, humidity, wind, good_for_kids, free_entry

A Data Annotation Schema

Table. 3 lists all annotated dialog domains, intents, and slots of MultiWoz and SGD in detail. Because GPT-2 is case sensitive, we map some annotations of SGD with the same or related meanings but different cases from those of MultiWoz. The specific mapping rules are shown in Table. 4 with format: "original word: mapped word". The words not in the Table. 4 are not processed. The "x" in string "_x" in the box "domain" in Table. 4 stands for the number, such as 1 in "restaurants_1".

B Implementation and Parameters

We adopt BERT$_{base}$ (uncased) and DistilGPT-2 (**?**), a distilled version of GPT-2 (Radford et al., 2019) as the backbone language model, and use default hyperparameters for BERT and DistilGPT-2 in Huggingface Transformers (Wolf et al., 2020). We pre-train and optimize all models on one RTX 2080Ti GPU and GTX TITAN X. For BERT and GPT-2's pre-training, the batch size is 8, and the training epoch is 3. The learning rate for BERT and GPT-2 are 0.00003 and 0.0005, respectively. To reduce resource consumption, we leverage FP16 computation[1] to use 16-bit (mixed) precision (through NVIDIA apex) for all models. Turn-level Action Classifier of BERT-based policy network (DA-BERT) is a linear layer with 400 output units corresponding to 400 candidates of action. Word-level Action Classifier of DA-GPT-2 is the sum of two linear layers: the language modeling head of GPT2LMHeadModel of Huggingface Transformers (Wolf et al., 2020) and Structure Parser (with 768 input units and 5 output units corresponding to 5 categories of words). ϵ-greedy is utilized for policy exploration. We set the discount factor as $\gamma = 0.9$. The target Q-network is updated at the end of each epoch. To mitigate warm-up issues, We apply the rule-based agent of ConvLab (Lee et al., 2019a) to provide experiences at the beginning, the warm_start epoch for BERT-based agents are 1000, while for GPT2 based agent is 50.

[1] https://docs.nvidia.com/deeplearning/performance/mixed-precision-training/index.html

Table 4: The data annotation schema.

	SGD
Domain	Restaurants_x: Restaurant, Media_x: Media, Events_x: Event, Music_x: Music, Movies_x: Movie, Flights_x: Flight, RideSharing_x: RideSharing, RentalCars_x: RentalCar, Buses_x: Bus, Hotels_x: Hotel, Services_x: Service, Homes_x: Home, Banks_x: Bank, Calendar_x: Calendar, Weather_x: Weather, Travel_x: Travel
Intent	INFORM_INTENT: InformIntent, REQUEST: Request, INFORM: Inform, OFFER: Offer, REQUEST_ALTS: RequestAlts, INFORM_COUNT: InformCount, SELECT: Select, CONFIRM: Confirm, AFFIRM: Affirm, NOTIFY_SUCCESS: NotifySuccess, THANK_YOU: ThankYou, GOODBYE: bye, OFFER_INTENT: OfferIntent, AFFIRM_INTENT: AffirmIntent, NEGATE: Negate, REQ_MORE: reqmore, NOTIFY_FAILURE: NotifyFailure, NEGATE_INTENT: NegateIntent
Slot	origin_city: Depart, destination_city: Dest, pickup_city: Depart, cuisine: Food, restaurant_name: Name, event_name: Name, song_name: Name, movie_name: Name, theater_name: Name, car_name: Car, origin_station_name: Depart, destination_station_name: Dest, dentist_name: Name, stylist_name: Name, doctor_name: Name, property_name: Name, recipient_account_name: Name, hotel_name: Name, attraction_name: Name, street_address: Addr, venue_address: Addr, address: Addr, phone_number: Phone, price_range: Price, time: Time, show_time: Time, outbound_departure_time: Leave, outbound_arrival_time: Arrive, inbound_departure_time: Leave, inbound_arrival_time: Arrive, wait_time: Time, pickup_time: Leave, departure_time: Leave, leaving_time: Leave, appointment_time: Time, event_time: Time, available_start_time: Leave, available_end_time: Arrive, date: Day, show_date: Day, departure_date: Leave, return_date: Arrive, dropoff_date: Arrive, pickup_date: Leave, leaving_date: Leave, check_in_date: Leave, check_out_date: Arrive, appointment_date: Day, visit_date: Day, event_date: Day, genre: Type, venue: Addr, category: Type, event_location: Addr, address_of_location: Addr, location: Addr, pickup_location: Depart, to_location: Dest, from_location: Depart, subcategory: Type, number_of_seats: People, event_type: Type, show_type: Type, ride_type: Type, type: Type, car_type: Type, fare_type: Type, account_type: Type, recipient_account_type: Type, price: Price, total_price: Price, origin_airport: Depart, destination_airport: Dest, origin: Depart, destination: Dest, fare: Fee, ride_fare: Fee, to_station: Dest, from_station: Depart, where_to: Dest, number_of_adults: People, rating: Stars, average_rating: Stars, star_rating: Stars, area: Area, number_of_days: Stay, price_per_night: Price