arXiv:2005.05298v3 [cs.CL] 22 Jun 2020

SOLOIST: Few-shot Task-Oriented Dialog with A Single Pre-trained Auto-regressive Model

Baolin Peng, Chunyuan Li, Jinchao Li
Shahin Shayandeh, Lars Liden, Jianfeng Gao
Microsoft Research, Redmond
`{bapeng,chunyl,jincli,shahins,laliden,jfgao}@microsoft.com`

Abstract

This paper presents a new method SOLOIST[1], which uses transfer learning to efficiently build task-oriented dialog systems at scale. We parameterize a dialog system using a Transformer-based auto-regressive language model, which subsumes different dialog modules (*e.g.,* state tracker, dialog policy, response generator) into a single neural model. We pre-train, on large heterogeneous dialog corpora, a large-scale Transformer model which can generate dialog responses grounded in user goals and real-world knowledge for task completion. The pre-trained model can be efficiently adapted to accomplish a new task with a handful of task-specific dialogs via machine teaching. Our experiments demonstrate that (*i*) SOLOIST creates new state-of-the-art results on two well-known benchmarks, CamRest and MultiWOZ, (*ii*) in the few-shot learning setting, the dialog systems developed by SOLOIST significantly outperform those developed by existing methods, and (*iii*) the use of machine teaching substantially reduces the labeling cost. We will release our code and pre-trained models for reproducible research.

1 Introduction

The increasing use of personal assistants and messaging applications has spurred interest in building task-oriented dialog systems that can communicate with users through natural language to accomplish a wide range of tasks, such as restaurant booking, weather query, flight booking, IT helpdesk, *etc.* The wide variety of tasks and domains has created the need for a flexible task-oriented dialog development platform that can support many different use cases, while remaining straightforward for developers to use and maintain.

A typical task-oriented dialog system uses a modularized pipeline, that has four modules and executes sequentially, as shown in Figure 1(a). A natural language understanding (**NLU**) module identifies user intents and extracts associated information such as slots and their values from userâĂŹs input. A dialog state tracker (**DST**) infers the *belief state* (or user goal) from *dialog history*. The belief state is often used to query a task-specific database (DB) to obtain the *DB state*, such as the number of entities that match the user goal. The dialog state and DB state are then passed to a dialog policy (**POL**) to select the next system action. A natural language generation (**NLG**) module converts the action to a natural language *response*.

Most popular commercial tools for dialog development employ the modular systems above, including Google's Dialog Flow[2], Microsoft's LUIS[3] and Bot Framework[4], Facebook's Wit.ai[5], Amazon's Lex[6], and IBM's Watson Assistant[7]. They are designed mainly to help develop systems *manually*, *i.e.,* writing code, crafting rules and templates. Unfortunately, even with these tools, building dialog systems remains a label-intensive, time-consuming task, requiring rich domain knowledge, reasonable coding skill, and expert experience. The cost of building dialog systems at scale (*i.e.,* hundreds of bots for different tasks) can be prohibitively expensive.

Due to the recent advances in neural approaches to conversational AI (Gao et al., 2019), researchers

---

[1] TASK-ORIENTED DIALOG WITH A SINGLE PRE-TRAINED MODEL

[2] `https://dialogflow.com/`

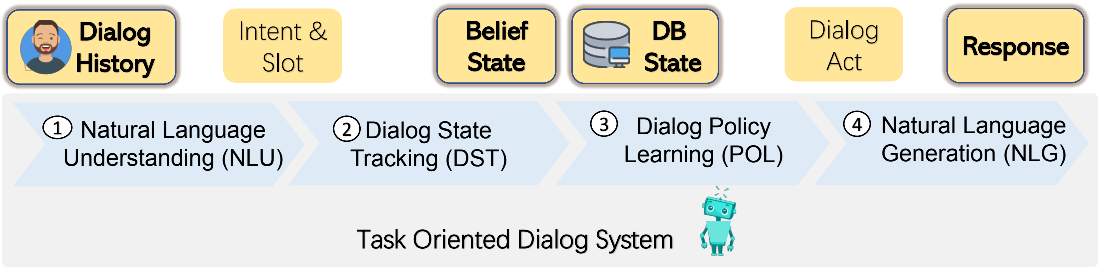
[3] `https://www.luis.ai/`

[4] `https://github.com/microsoft/BotFramework-Composer`

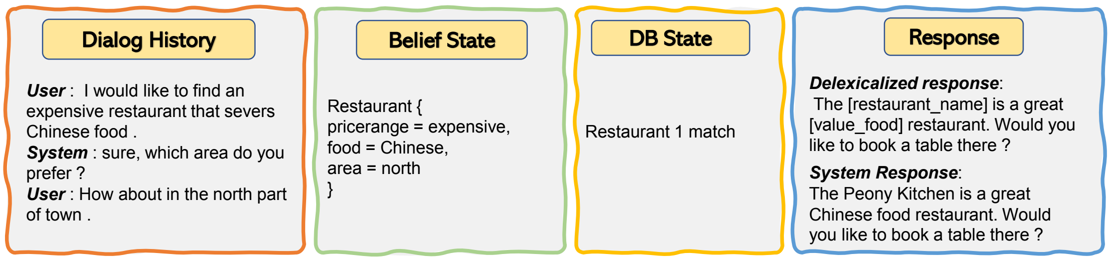
[5] `https://wit.ai/`

[6] `https://aws.amazon.com/lex/`

[7] `https://www.ibm.com/watson/`


Dialog History
Intent & Slot
Belief State
DB State
Dialog Act
Response
① Natural Language Understanding (NLU)
② Dialog State Tracking (DST)
③ Dialog Policy Learning (POL)
④ Natural Language Generation (NLG)
Task Oriented Dialog System

(a) A typical task-oriented dialog system pipeline.



User : I would like to find an expensive restaurant that severs Chinese food .
System : sure, which area do you prefer ?
User : How about in the north part of town .

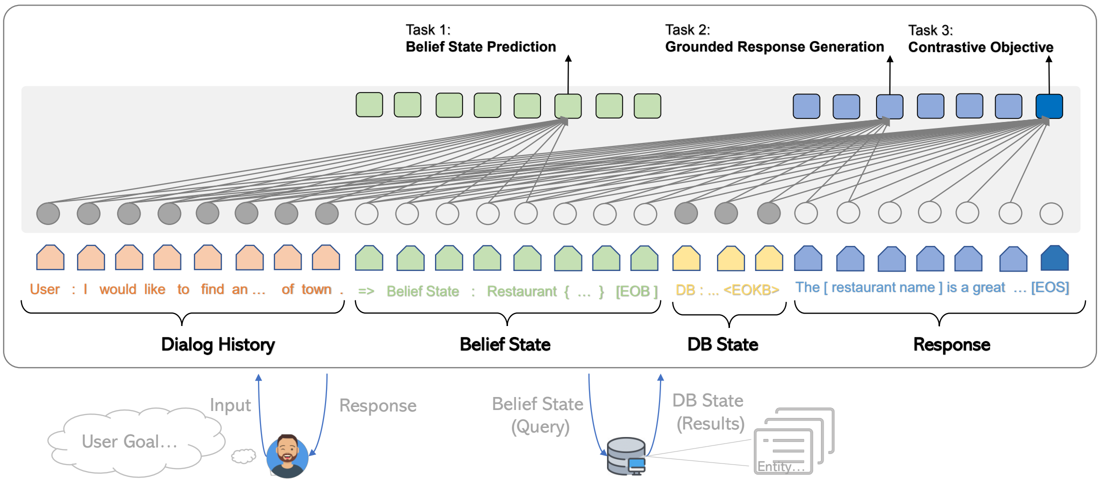
Restaurant {
pricerange = expensive,
food = Chinese,
area = north
}

Restaurant 1 match

Delexicalized response:
The [restaurant_name] is a great [value_food] restaurant. Would you like to book a table there ?

System Response:
The Peony Kitchen is a great Chinese food restaurant. Would you like to book a table there ?

(b) Example snippets for the items compounding the input of SOLOIST model.



(c) The proposed SOLOIST model architecture and training objectives.

Figure 1: Illustration of a traditional modular task-oriented dialog system, an example for the model input, and the proposed model. The SOLOIST solution utilizes a single neural auto-regressive model in (c) to paramterize the sequential dialog pipeline in (a), with input sequence represented in (b). Different from GPT-2, the SOLOIST model learns to ground response generation in user goals and database/knowledge.

are developing data-driven methods and neural models for either individual dialog modules or end-to-end systems. For example, recent attempts such as RASA (Bocklisch et al., 2017), ConvLab (Lee et al., 2019; Zhu et al., 2020), and Conversation Learner (Shukla et al., 2020) have been made to allow the use of data-driven approaches based on machine learning or machine teaching for the development of dialog modules. End-to-end trainable dialog systems have been studied in (Wen et al., 2016; Zhao and Eskenazi, 2016; Li et al., 2017; Williams et al., 2017; Lei et al., 2018; Gao et al., 2019; Zhang et al., 2019a). Although these methods have achieved promising results, they require large amounts of task-specific labeled data for training, which are rarely available for new tasks.

In this paper we propose a novel task-oriented bot building paradigm, SOLOIST, which significantly eases the workflow of training and deploying dialog systems for new tasks, compared to existing tools and methods. Our approach is inspired by the recent success of applying transfer learning to natural language processing (NLP) tasks through the use of pre-trained models (*e.g.,* BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and UniLM (Dong et al., 2019)): It has been shown that a large-scale language model pre-trained on raw text can be effectively fine-tuned to a wide range of NLP tasks with few in-domain labels. The proposed SOLOIST is based on a similar pre-training-and-fine-tuning framework. We parameterize a dialog system using a Transformer-based auto-regressive language model (Vaswani et al., 2017), which subsumes different dialog modules (NLU, DST, POL and NLG) into a single neural model. Task-oriented bot building proceeds in two stages: (i) In the pre-training stage, initialized using GPT-2, we train a large-scale Transformer model of generating dialog responses that are *grounded* in dialog history, user goals and real-world knowledge for task completion using large heterogeneous dialog corpora. (ii) In the fine-tuning stage, we fine-tune the pre-trained model to complete a new task using a a handful of task-specific dialogs via machine teaching (Zhu, 2015; Shukla et al., 2020).

We show through a comprehensive empirical study that SOLOIST is an effective solution to building task-oriented bots at scale which successfully transfers, from the pre-trained model to a new task-specific bot, two capabilities: (i) the capability of understanding and generating natural language sentences, learnt by GPT-2 on raw text, and (ii) the capability of grounding text responses in user goals and real-world knowledge for task completion, learned from out-domain dialog corpora.

SOLOIST achieves new state-of-the-art results on two standard benchmarks, lifting the combined score by 10 points, and improving the success rate by 20 points on human evaluation. In the few-shot settings, SOLOIST adapts to new domain much more effectively than competing methods, achieving a reasonable success rate using less than 50 dialogs. The promising results demonstrate the potential of the new paradigm for developing task-oriented dialog bots. Instead of collecting, labeling data, and building a bot per task, we can pre-train a *universal, grounded* language generation model, and adapt it to new tasks via transfer learning and machine teaching.

2 SOLOIST

2.1 An Auto-regressive Model for Dialog

The modular dialog system in Figure 1 constitutes a data processing procedure that naturally produces a sequence, through concatenating input/output of each module along the generation process. Each consecutive pair in this sequence plays the role of annotated data for the corresponding module. Ideally, when the entire sequence is available, the data generation process of dialog system (NLU, DST, POL, NLG) can be formulated as a *single* auto-regressive model, and the full sequence can be learned in a self-supervised manner.

GPT-2 (Radford et al., 2019) is one of the most known auto-regressive language models. GPT-2 learns the language granularity from large amounts of open Web text data, and after fine-tuned using conversational data (Zhang et al., 2019b) can respond users with realistic and coherent continuations about a topic of their choosing. However, the generated responses are not useful for completing any specific task due to the lack of grounding. SOLOIST inherits GPT-2's capability of producing human-like responses. But, unlike GPT-2, SOLOIST learns to ground the generation process in user goals and real-world knowledge so that the generated responses are useful for completing tasks. Note that SOLOIST is a general framework for grounded language generation, where prescribed control codes in wildly different but related domains are used to pre-train a generic guided language model, which can quickly adapt to a new domain through fine-tuning with a few task-specific examples. Specifically for task-oriented task systems, we consider to pre-train on conversational data with grounding information, *i.e.,* belief states and DB states. More specifically, each dialog turn in our training data is represented as:

$$\boldsymbol{x} = (\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{r}), \qquad (1)$$

where $\boldsymbol{s}$ is the entire dialog history up to the current dialog turn, $\boldsymbol{b}$ is the belief state acquired from human annotation, $\boldsymbol{c}$ is the DB state automatically retrieved from a database using $\boldsymbol{b}$, and $\boldsymbol{r}$ is the delexicalized dialog response, from which the system response in natural language can be easily obtained with some automatic post-processing. Each item in $\boldsymbol{x}$ is by itself a sequence of tokens, as illustrated by examples in Figure 1(b). Thus, it is natural to treat the concatenation of them as a long sequence for model training as shown in Figure 1(c). We pre-train the SOLOIST model using heterogeneous dialog corpora with labels of belief states and DB states, which are publicly available. The pre-trained model can be fine-tuned to any new task to generate responses grounded in task-specific user goals and database.

2.2 Task-Grounded Pre-training

Given training data of N samples $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$, our goal is to build a statistical model parameterized by $\boldsymbol{\theta}$ to characterize $p_{\boldsymbol{\theta}}(\boldsymbol{x})$. We consider a multi-task objective for learning $\boldsymbol{\theta}$, where each task is a self-supervised learning task.

To leverage the sequential structure of a task-oriented dialog system, the joint probability $p(\boldsymbol{x})$ can be factorized in the auto-regressive manner as:

$$p(\boldsymbol{x}) = p(\boldsymbol{r}, \boldsymbol{c}, \boldsymbol{b}, \boldsymbol{s}) \tag{2}$$

$$= \underbrace{p(\boldsymbol{r}|\boldsymbol{c}, \boldsymbol{b}, \boldsymbol{s})}_{\text{Grounded Response Generation}} \; \underbrace{p(\boldsymbol{b}|\boldsymbol{s})}_{\text{Belief Prediction}} \; p(\boldsymbol{s}). \tag{3}$$

where the factorization from (2) to (3) is based on the fact that $p(\boldsymbol{c}|\boldsymbol{b}, \boldsymbol{s}) = p(\boldsymbol{c}|\boldsymbol{b}) = 1$, because the DB state $\boldsymbol{c}$ is obtained using a deterministic database-lookup process given a belief state $\boldsymbol{b}$ (*e.g.,* via an API call). Note that (3) decomposes the joint distribution modeling problem into two sub-problems: belief state prediction $p(\boldsymbol{b}|\boldsymbol{s})$ and grounded response generation $p(\boldsymbol{r}|\boldsymbol{c}, \boldsymbol{b}, \boldsymbol{s})$. Since $\boldsymbol{b}$ and $\boldsymbol{r}$ are sequences as well, we may further factorize them in the left-to-right auto-regressive manner, respectively.

Task 1: Belief Prediction. For the belief state sequence of length T_b, we define the objective of predicting the belief token as:

$$\mathcal{L}_{\text{B}} = \log p(\boldsymbol{b}|\boldsymbol{s}) = \sum_{t=1}^{T_b} \log p_{\boldsymbol{\theta}}(b_t | b_{<t}, \boldsymbol{s}), \tag{4}$$

where $b_{<t}$ indicates all tokens before t.

Task 2: Grounded Response Generation. To generate a response of length T_r, $\boldsymbol{r} = [r_1, \cdots, r_{T_r}]$, our model generate every token x_t grounded in the previous word tokens and task-oriented information $[\boldsymbol{c}, \boldsymbol{b}, \boldsymbol{s}]$:

$$\mathcal{L}_{\text{R}} = \log p(\boldsymbol{r}|\boldsymbol{c}, \boldsymbol{b}, \boldsymbol{s}) \tag{5}$$

$$= \sum_{t=1}^{T_r} \log p_{\boldsymbol{\theta}}(r_t | r_{<t}, \boldsymbol{c}, \boldsymbol{b}, \boldsymbol{s}).$$

Task 3: Contrastive Objective. We consider a contrastive objective to promote the matched items (positive samples $\boldsymbol{x}$), while driving down mismatches (negative samples $\boldsymbol{x}'$). Specifically, we sample a set of negative samples from sequence $\boldsymbol{x}$ by replacing some items in $\boldsymbol{x}$ with probability 50% with different items randomly sampled from the dataset $\mathcal{D}$. Since the the special token `[EOS]` attends all tokens in the sequence, the output feature on `[EOS]` is the fused representation of all items, we apply a binary classifier on top of the feature to predict whether the items of the sequence are matched ($y = 1$) or mismatched ($y = 0$):

$$\mathcal{L}_{\text{C}} = y \log(p_{\boldsymbol{\theta}}(\boldsymbol{x})) + (1-y) \log(1 - p_{\boldsymbol{\theta}}(\boldsymbol{x}')) \tag{6}$$

We consider three types of negative samples $\boldsymbol{x}'$, each of which is chosen with probability 1/3: (*i*) *negative belief*, where only the belief item is replaced (*ii*) *negative response*, where only the response item is replaced (*iii*) *negative belief + response*, where both the belief and response items are replaced.

Full Pre-training Objective. Learning $\boldsymbol{\theta}$ is performed via maximizing the log-likelihood (MLE) over the entire training dataset $\mathcal{D}$, using a joint objective that combines (4), (5) and (6):

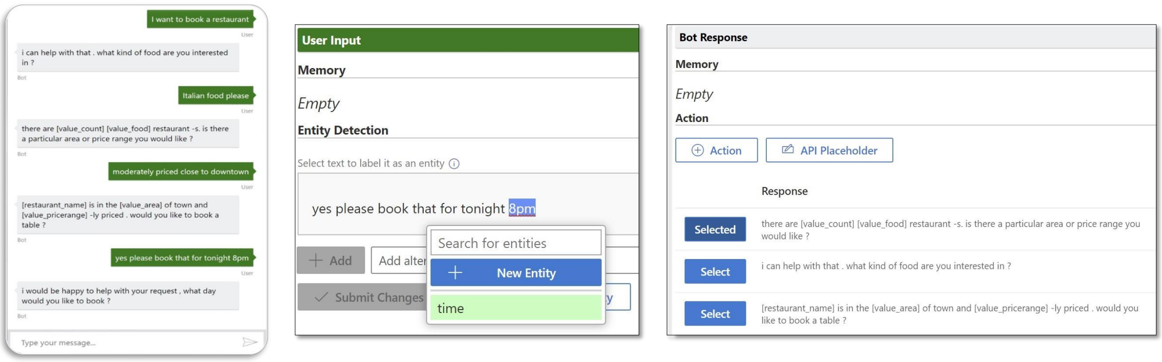
$$\mathcal{L}_{\boldsymbol{\theta}}(\mathcal{D}) = \sum_{n=1}^{|\mathcal{D}|} (\mathcal{L}_{\text{B}}(\boldsymbol{x}_n) + \mathcal{L}_{\text{R}}(\boldsymbol{x}_n) + \mathcal{L}_{\text{C}}(\boldsymbol{x}_n)) \tag{7}$$

In Figure 1(c), we illustrate the model architecture and learning objectives. The model is fully auto-regressive in a left-to-right manner, each objective appearing on a given sub-sequence and a special segment token.

Implementation Details We process each dialog turn $(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{r})$ in training data into a sequence of tokens. For instance, the processed sequence of the examples shown in Figure 1(b) is as follows, where different items are rendered in different colors.

> User : I would like to find an expensive restaurant that severs Chinese food. System : sure, which area do you prefer ? User : How about in the north part of town . => Belief State : Restaurant { pricerange = expensive, food = Chinese, area = north } < `EOB` > DB : Restaurant 1 match < `EOKB` > The [restaurant_name] is a great [value_food] restaurant. Would you like to book a table there ? < `EOS` >

This sequence can be directly fed into an auto-regressive model for training, as shown in Figure 1(c). Our implementation is based on Huggingface Pytorch Transformer (Wolf et al., 2019a). SOLOIST uses GPT-2 with 117M parameters as the initial checkpoint while SOLOIST$_{\text{L}}$ uses GPT-2 meidum with 337M parameters as the initial checkpoint for pre-training, and byte pair encodings (Sennrich et al., 2015) for the tokenization. We pre-train


I want to book a restaurant

i can help with that . what kind of food are you interested in ?

Italian food please

there are [value_count] [value_food] restaurant -s. is there a particular area or price range you would like ?

moderately priced close to downtown

[restaurant_name] is in the [value_area] of town and [value_pricerange] -ly priced . would you like to book a table ?

yes please book that for tonight 8pm

i would be happy to help with your request , what day would you like to book ?

User Input

Memory

Empty

Entity Detection

Select text to label it as an entity

yes please book that for tonight 8pm

Search for entities

time

Bot Response

Memory

Empty

Action

Response

there are [value_count] [value_food] restaurant -s. is there a particular area or price range you would like ?

i can help with that . what kind of food are you interested in ?

[restaurant_name] is in the [value_area] of town and [value_pricerange] -ly priced . would you like to book a table ?

(a) Human-bot dialog (b) Belief State Edit (c) Response Edit

Figure 2: Illustration of the machine teaching process using conversion learner. The human-bot conversion log in (a) can be edited via correcting its belief state in (b), and selecting/inserting a more appropriate response in (c).

Name	#Dialog	#Utterance	Avg. Turn	#Domain
Schema	22,825	463,284	20.3	17
Taskmaster	13,215	303,066	22.9	6
MultiWOZ	10,420	71,410	6.9	7
MSR-E2E	10,087	74,686	7.4	3
SMD	3,031	15,928	5.3	3
Frames	1,369	19,986	14.6	3
WOZ	1,200	5,012	4.2	1
CamRest676	676	2,744	4.1	1

Table 1: Statistics of dialog corpora (Wu et al., 2020a)

the grounded response generation model on dialog corpora (Kim et al., 2019; Rastogi et al., 2019; Byrne et al., 2019; Eric and Manning, 2017; Mrkšić et al., 2016; Asri et al., 2017) as shown in Table 1. To ensure non-overlapping of datasets used for pre-training and fine-tuning, we exclude the data akin to MultiWOZ and CamRest. The results reported in this paper are based on pre-trained models on Schema and Taskmaster dataset. We will also release models pre-trained on all the corpora. We use Adam (Kingma and Ba, 2014) with weight decay to pre-train the model for 100k steps.

2.3 Few-shot Fine-tuning

When deploying SOLOIST to a new task, we collect task-specific $\boldsymbol{x}$ in the same format as that in the pre-training stage as (1). When annotated log data is available, the conventional fine-tuning procedure is utilized: we use the same multi-task objective of (7) to update $\boldsymbol{\theta}$ to adapt the model to complete the new task using labeled task-specific dialogs.

In real applications, annotated log data is often unavailable, or noisy/incomplete beforehand. One may deploy the model, and acquire high-quality task-specific labels (*e.g.,* belief state and system response) for each dialog turn using machine teaching (Simard et al., 2017; Zhu, 2015; Williams and Liden, 2017; Shukla et al., 2020). Machine teaching is an active learning paradigm that focuses on leveraging the knowledge and expertise of domain experts as "teachers". This paradigm puts a strong emphasis on tools and techniques that enable teachers - particularly non-data scientists and non-machine-learning experts - to visualize data, find potential problems, and provide corrections or additional training inputs in order to improve the systemâĂŹs performance.

We proceed fine-tuning using Conversation Learner (Shukla et al., 2020), a machine teaching tool, in the following steps: (*i*) Dialog authors deploy the pre-trained SOLOIST model for a specific task. (*ii*) Users (or human subjects recruited for system fine-tuning) interact with the system and generate human-bot dialog logs. (*iii*) Dialog authors revise a dozen of training samples by selecting representative failed dialogs from the logs, correcting their belief and/or responses so that the system can complete these dialogs successfully. We illustrate the dialog editing process using Conversation Learner in Figure 2. Readers may refer to Shukla et al. (2020) for details. The corrected task-specific dialog turns are used to fine-tune the model. It is shown that machine teaching is a more effective approach to improving deployed dialog systems by providing on-the-spot corrections.

Implementation Details Instead of using machine teaching from scratch, we assume that a few task-specific data are available for fine-tuning. Details are presented in Sec. 3.3. Training examples are truncated to ensure max length 512. The model is trained with a mini-batch of 6 on 8 Nvidia V100 until observing no significant progress on validation loss or up to 10 epochs. Nucleus sampling (Holtzman et al., 2019) is used for decoding, where the sampling top-p ranges from 0.2 to 0.5 for all our models. The best setup of hyper-parameters is selected through grid-search on the validation set. Due to high latency of SOLOIST$_{\text{L}}$ decoding, we only report its results in few-shot evaluation setting. Unless otherwise stated, the results of SOLOIST are obtained using models with 117M paramters.

3 Experiments

In this section, we evaluate the proposed SOLOIST to answer three research questions: `Q1`: How does SOLOIST perform on standard benchmarks compared to SoTA? `Q2`: Does SOLOIST meet the goal of effectively generalizing to new domains in the few-shot learning setting? `Q3`: Is machine teaching a more efficient approach to fine-tuning when applied? Note that we employed the conventional fine-tuning scheme *without* machine teaching for fair comparison when studying `Q1` and `Q2`.

3.1 Experimental Setup

Datasets for Fine-tuning We validate the proposed SOLOIST on two public datasets. (*i*) CamRest676 is a single-domain task-oriented dialog corpus collected by Wen et al. (2016). It contains 408/136/136 dialogs for training/validation/testing, respectively. Following Lei et al. (2018), we delexicalize each token that occurs in the ontology with its slot names such as restaurant name, phone number, and postcode. (*ii*) MultiWOZ dataset (Budzianowski et al., 2018) is a large-scale human-human multi-domain task-oriented dialog dataset. It contains 8438/1000/1000 for training/validation/testing, respectively. Each dialog session in the corpus contains 1 to 3 domains, including Attraction, Hotel, Hospital, Police, Restaurant, Train, and Taxi. MultiWOZ is inherently challenging due to its multi-domain setting and diverse language styles.

Automatic Evaluation Metrics. Following Budzianowski et al. (2018), `Inform`, `Success`, and `BLEU` scores are reported. `Inform` measures if the system provides a correct entity (inform rate). `Success` measures the exact matching of answering all the requested information (success rate) and if the answered information matches users' goal . `BLEU` evaluates how natural the generated utterance is compared with human readers. A combined score (`Combined`) is also reported using $\texttt{Combined} = (\texttt{Inform} + \texttt{Success}) \times 0.5 + \texttt{BLEU}$ as an overall quality measure, as suggested in Budzianowski et al. (2018).

Baselines. We compare with several strong baseline methods, which hold state-of-the-arts on CamRest676 or MultiWOZ datasets. (*i*) Multi-Action Data Augmentation (DAMD) (Zhang et al., 2019a) is a state-of-the-art modular system, where each dialog module is implemented using a neural network, and the whole system is trained in an end-to-end manner. (*ii*) Sequicity (Lei et al., 2018) is similar to DAMD except that it does not use multi-action data augmentation. (*iii*) A GPT-2 model that is fine-tuned on dialog data. The model is not grounded, and needs to work with a separate dialog state tracker for task completion. (*iv*) ARDM (Wu et al., 2019) utilizes GPT-2 as the pre-trained model to learn to generate role-aware responses given dialog context. The model has to work with a separate dialog state tracker for task completion. (*v*) HDSA (Chen et al., 2019) is a modular dialog system which generates responses using BERT-based dialog policy and graph structure dialog act representations.

3.2 Comparing to SOTA systems

CamRest676 Table 2 shows the results using generated belief states on CamRest676. The annotations unitized by the models are also listed. SOLOIST achieves the best scores over all the metrics. ARDM performs similarly to SOLOIST in terms of Success and BLEU score. However, ARDM is infeasible to track dialog states and requires a separately trained state tracker to accomplish tasks. GPT-2 fine-tuned with task-specific data works reasonably good but legs behind SOLOIST by a large margin. Sequicity (Lei et al., 2018) is a jointly trained model with belief state and policy annotation, and under-performs SOLOIST. This result suggests that in simple tasks like CamRest676, SOLOIST is able to achieve user goals with only belief state annotations and maintains good fluency due to the benefit from task-grounded pre-training.

MultiWOZ Context-to-Response We first consider the context-to-response generation task (Wen

Model	Annotations		Evaluation Metrics			
	Belif State	Policy	Inform ↑	Success ↑	BLEU ↑	Combined ↑
Sequicity (Lei et al., 2018)	✓	✓	92.30	85.30	21.40	110.20
Sequicity (w/o RL)	✓	✓	94.00	83.40	23.40	112.10
GPT-2 finetuned			-	86.20	19.20	-
ARDM (Wu et al., 2019)			-	87.10	25.20	-
SOLOIST	✓		**94.70**	**87.10**	**25.50**	**116.40**

Table 2: End-to-End Evaluation on CamRest676.

Model	Annotations		Evaluation Metrics			
	Belif State	Policy	Inform ↑	Success ↑	BLEU ↑	Combined ↑
Baseline (Budzianowski et al., 2018)	✓		71.29	60.94	18.80	84.93
TokenMoE (Pei et al., 2019)	✓		75.30	59.70	16.81	84.31
GPT-2 (Budzianowski and Vulić, 2019)	✓		70.96	61.36	19.05	85.21
HDSA (Chen et al., 2019)	✓	✓	82.90	68.90	23.60	99.50
Structured Fusion (Mehri et al., 2019)	✓	✓	82.70	72.10	16.34	93.74
LaRL (Zhao et al., 2019)	✓		82.80	79.20	12.80	93.80
ARDM (Wu et al., 2019)			87.40	72.80	20.60	100.70
DAMD (Zhang et al., 2019a)	✓	✓	89.20	77.90	18.60	102.15
SOLOIST	✓		**89.60**	**79.30**	18.03	**102.49**

Table 3: Context-to-response evaluation on MultiWOZ.

Model	Annotations		Evaluation Metrics			
	Belif State	Policy	Inform ↑	Success ↑	BLEU ↑	Combined ↑
Sequicity (Lei et al., 2018)	✓	✓	66.41	45.32	15.54	71.41
HRED-TS (Peng et al., 2019)	✓	✓	70.00	58.00	17.50	81.50
Structured Fusion (Mehri et al., 2019)	✓	✓	73.80	58.60	16.90	83.10
DSTC8 Track 1 Winner [1] (Ham et al., 2020)	✓	✓	73.00	62.40	16.00	83.50
DAMD (Zhang et al., 2019a)	✓	✓	76.40	60.40	16.60	85.00
SOLOIST	✓		**85.50**	**72.90**	**16.54**	**95.74**

[1] The result of DSTC8 Track 1 Winner is produced by adapting their code to our current setting.

Table 4: End-to-end evaluation on MultiWOZ.

et al., 2016), where the ground truth belief states and database search results are given, based on which responses are predicted. The results are shown in Table 3. The proposed SOLOIST achieves the best performance in terms of Inform and Success scores but performs slightly worse in terms of BLEU score. The overall combined score of SOLOIST is comparable with the current SoTA method DADM (Zhang et al., 2019a). However, DAMD leverages the labels of dialog act on both the user and system sides, which demands significantly higher labeling efforts than SOLOIST. HDSA achieves the best number on BLEU. Compared to HDSA, SOLOIST is much simpler and able to perform better in terms of combined score. SOLOIST performs better than ARDM on combined score. It is worth mentioning that ARDM does not consider dialog state tracking and thus requires an extra dialog state tracker to accomplish a certain task. These results reveal that SOLOIST is able to learn dialog policy accurately and generate natural language responses in the multi-domain scenario.

MultiWOZ End-to-End We now consider a more pragmatic evaluation setting of studying a model's end-to-end learning performance, where the generated belief states are used for database search and response generation. The results are shown in Table 4. SOLOIST achieves the best performance in terms of inform and success rates,

Domain	`Attra.`	`Train`	`Hotel`	`Rest.`	`CamRest`
#Train	50	50	50	50	20
#Valid	50	50	50	50	136
#Test	100	200	200	200	136

Table 5: Data statistics for domains used in few-shot evaluation. `Attra.` denotes attraction domain and `Rest.` means restaurant domain.

Model	`CamRest`		
	`Inform` ↑	`Success` ↑	`BLEU` ↑
Sequicity (Lei et al., 2018)	60.61	66.11	11.15
SOLOIST w/o pre-training	73.88	72.22	13.11
SOLOIST	85.82	84.22	19.18
SOLOIST$_{\mathrm{L}}$	**88.05**	**84.79**	**18.88**

Table 6: End-to-end evaluation on CamRest in a few-shot learning setting.

and combined score, lifting the previous SOTA by DAMD by a significant margin (*e.g.,* about 10 points improvement on the combined score). Our method also outperforms the method of Ham et al. (2020), where GPT-2 is fine-tuned and applied to end-to-end dialog. Compared with the classical modular dialog systems or the jointly trained model DAMD, it is worth noting that SOLOIST has a much simpler architecture and requires much lower labeling effort. For example, SOLOIST requires only the belief states, while DAMD requires additional annotations for task definition (*i.e.,* defining the intents, slots, and the corresponding value ranges) and dialog acts.

3.3 Few-shot Evaluation

It is desired that dialog systems can effectively generalize with a few training examples. We argue that the few-shot learning setting is a more realistic scenario for dialog modeling. Unfortunately, the existing corpus typically contains hundreds to thousands of dialogs depending on the complexity of dialog tasks. As such, we re-organize CamRest676 and MultiWOZ to simulate the few-shot learning setting for the end-to-end dialog modeling[8]. We sample from MultiWOZ dialogs the dialog tasks that only contain one domain. Attraction, Train, Hotel, and Restaurant domains are used. We ignore Police, Taxi, Hospital, as these tasks do not require an explicit tracking state to accomplish the task. For each task (or domain), we randomly sample 50 dialog sessions for training and validation and 200 dialog sessions for testing, except the Attraction domain that only has 100 sessions for testing. For CamRest, we only randomly sample 20 sessions from the original CamRest676 since this dataset is relatively small. Details are shown in Table 5.

Table 6 and 7 report the end-to-end performance in the few-shot learning settings on CamRest and MultiWoz, respectively. In all the tasks, SOLOIST and SOLOIST$_{\mathrm{L}}$ show substantially better performance on all the metrics. Removing pre-training on dialog corpora downgrades the performance of SOLOIST, but still consistently outperforms DAMD in all the domains. Removing pre-training, SOLOIST is conceptually similar to (Ham et al., 2020), but is architecturally simpler and needs less annotations. This also verifies the importance of grounded pre-training on annotated dialog corpora, allowing SOLOIST to learn how to track dialog and database states to accomplish a task.

We conduct experiments to fine-tune SOLOIST by varying percentages of training data ranging from 1% (80 examples) to 20% (1600 examples) on the MultiWOZ dataset. As shown in Table 8, SOLOIST consistently outperforms DAMD for a wide range of dataset sizes, and the improvement is more substantial when smaller numbers of in-domain labels are used for fine-tuning.

3.4 Machine Teaching Results

We leverage the user interface (UI) of Conversation Learner (Shukla et al., 2020) for dialog authors (human teachers) to correct wrong or inadequate responses, and evaluate the potential of SOLOIST being continually improved after the deployment. Table 9 shows the results. We firstly use 20 dialog sessions from restaurant domain to perform the few-shot fine-tuning step of SOLOIST. Its evaluation result is listed in the first row of table 9. We then deploy it to interact with users. The row of SOLOIST + Teach shows the result of machine teaching where a human teacher is involved to manually correct 5 failed dialog sessions, which are then utilized to continually fine-tune the deployed system. We observe that SOLOIST + Teach improves the combined score by 10% compared to that without human teaching. The results demonstrate the effectiveness of our two-step fine-tuning scheme to deploy SOLOIST for a new task. SOLOIST + *Extra* is used as an ablation baseline, where 5 randomly selected dialog sessions are added as extra examples

---

[8] We will release the re-organized datasets.

Model	Attraction			Train			Hotel			Restaurant		
	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑
DAMD (Zhang et al., 2019a)	70.00	15.00	6.90	75.00	39.50	6.20	62.50	20.50	7.60	68.00	19.50	10.50
SOLOIST w/o pre-training	65.66	46.97	5.85	59.00	44.00	7.07	62.50	40.00	7.70	75.50	44.50	11.00
SOLOIST	86.00	65.00	12.90	80.81	64.65	9.96	74.50	43.50	8.12	81.00	55.50	12.80
SOLOIST$_\text{L}$	**86.00**	**68.00**	**14.60**	**81.31**	**74.24**	**11.90**	**75.00**	**51.50**	**10.09**	**84.00**	**62.50**	**13.17**

Table 7: End-to-end evaluation on MultiWOZ in a few-shot learning setting.

Model	1%			5%			10%			20%		
	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑
DAMD (Zhang et al., 2019a)	34.40	9.10	8.10	52.50	31.80	11.60	55.30	30.30	13.00	62.60	44.10	14.90
SOLOIST w/o pre-training	46.10	24.40	10.39	63.40	38.70	11.19	64.90	44.50	13.57	70.10	52.20	14.72
SOLOIST	**58.40**	**35.30**	**10.58**	**69.30**	**52.30**	**11.80**	**69.90**	**51.90**	**14.60**	**74.00**	**60.10**	**15.24**

Table 8: End-to-end Evaluation on MultiWOZ with varying sizes of training data.

Model	Inform ↑	Success ↑	BLEU ↑	Combined ↑
SOLOIST	72.09	44.19	9.30	67.44
SOLOIST + Extra	**79.07**	45.35	10.00	72.01
SOLOIST + Teach	77.91	**58.14**	**12.00**	**79.67**

Table 9: Machine teaching results.

Model	Success ↑	Under. ↑	Appr. ↑	Turns ↓
SOLOIST	**91.67**	**4.29**	**4.43**	18.97
DSTC8 Track 1 Winner	68.32	4.15	4.29	19.51
DSTC8 2nd Place	65.81	3.54	3.63	15.48
DSTC8 3rd Place	65.09	3.54	3.84	**13.88**
DSTC8 Baseline	56.45	3.10	3.56	17.54

Table 10: Human evaluation results. The remaining results except SOLOIST are quoted from (Li et al., 2020b). Under. is short for language understanding score and Appr. stands for appropriateness score.

to fine-tune the model. It shows lower performance than machine teaching. Assume that one slot-value pair of belief state correction counts one edit and a response correction counts ten edits. The total numbers of edits for SOLOIST + *Teach* and SOLOIST + *Extra* are 61 and 396, respectively, suggesting that machine teaching reduces the labeling cost by 6x.

3.5 Human Evaluation

Corpus-based evaluation may not adequately represent the complexity of dialog systems in deployment. Therefore, we conduct human evaluations to assess the quality of SOLOIST interacting with real users. Following the same evaluation protocol in DSTC8 track 1 challenge (Kim et al., 2019), we host the best model on MultiWOZ in the backend as bot services and crowdsource the work to Amazon Mechanical Turk. For each dialog session, we present Turks a goal with instructions. Then Turks are required to converse with SOLOIST to achieve the goal and judge the overall dialog at the end of a session in terms of the following three metrics. (*i*) *Success/Fail* evaluates task completion. (*ii*) *Language Understanding Score* ranging from 1 (bad) to 5 (good) indicates the extent to which the system understands user inputs. (*iii*) *Response Appropriateness Score* scaling from 1 (bad) to 5 (good) denotes whether the response is appropriate and human-like. Turks are further required to briefly describe a justification of giving a specific rating. In total, 120 dialog sessions are gathered for analysis.

Table 10 shows the human assessment results on MultiWOZ. SOLOIST achieves substantially better performance than the DSTC8 Track 1 Winner over all the metrics. These results are consistent with the automatic evaluation. However, it is interesting to see that the gap of success rate is more than 20 percent on human evaluation, much larger than the 10 percent gap on automatic evaluation. We attribute this to the fact that the language usage from different Turks varies considerably and is noisy, containing redundant or deficient words and typos etc. By virtue of large-scale pre-training on homogeneous dialog corpora, SOLOIST is able to handle language variations and tolerate noisy inputs to accomplish a task. It is worth mentioning that in quite a few cases Turks comment that they feel like talking to a real person.

3.6 Ablation Analysis

To study the effect of different schemes to construct negative samples in the pre-training objectives, we conduct an ablation analysis on MultiWOZ. The results are reported in Table 11. We see that removing belief state only from negative samples substan-

Model	Inform ↑	Success ↑	BLEU ↑	Combined ↑
Full objective	85.50	72.90	16.54	95.74
- belief	81.50	69.30	16.82	92.22
- belief & response	82.50	67.30	16.28	91.18

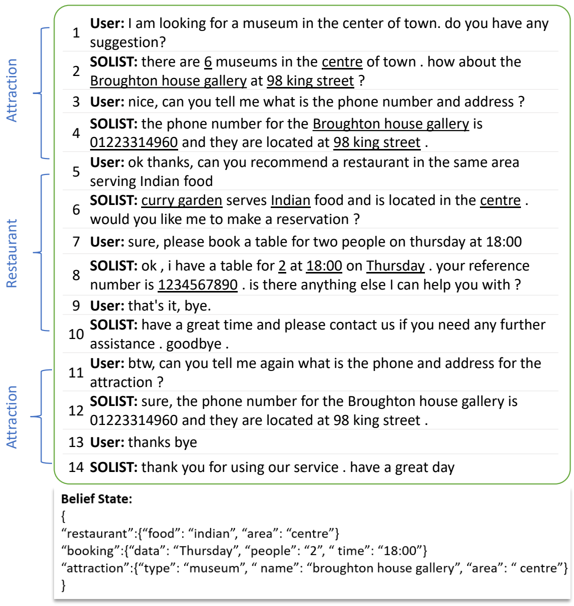
Table 11: Ablation study on different negative samples in the contrastive objective on MultiWOZ in the end-to-end evaluation setup; The 2nd and 3rd row indicate removing individual belief only and individual belief & response, respectively.

tially degrades the inform and success score. We further remove the mismatched responses, and only use the mismatched belief state and response pairs to constructive negative samples. This decreases the success rate and BLEU score. The results show the importance of constructing various types of negative samples in designing an effective contrastive objective.

Interactive Example. Figure 3 depicts an interactive example using SOLOIST as the agent to communicate with a user, who wants to first get the phone number and address of an attraction in the center of town, then book a table for two people on Thursday at 18:00 at an indian restaurant in ther center area. The belief state at the final turn and database entries that match the belief state are shown in the figure. We see that SOLOIST accurately tracks dialog states, responsibly converses with the user, and successfully accomplishes the task. For example, at the 11-th turn, the user asks again the information about the attraction domain, SOLOIST can switch to the attraction domain smoothly and respond appropriately. It is noteworthy that though the user does not mention the area for the restaurant but instead states in the same area in 5-th turn, SOLOIST could still capture the center area for the restaurant revealing strong coreference ability.

4 Related Work

Pre-trained Language Models. Recent advances on self-supervised learning have witnessed the blooming of large-scale pre-trained language models (Devlin et al., 2019; Radford et al., 2019), which achieved state-of-the-art performance on a variety of language understanding and generation tasks. The closest line of research to ours are GPT-2 (Radford et al., 2019), and its variants to ground language generation on the prescribed control codes such as CTRL (Keskar et al., 2019) and Grover (Zellers et al., 2019), or latent variables such as Optimus (Li et al., 2020a).

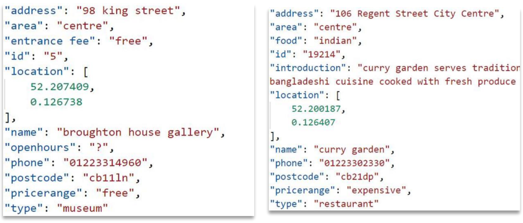


1 **User:** I am looking for a museum in the center of town. do you have any suggestion?
2 **SOLIST:** there are 6 museums in the centre of town . how about the Broughton house gallery at 98 king street ?
3 **User:** nice, can you tell me what is the phone number and address ?
4 **SOLIST:** the phone number for the Broughton house gallery is 01223314960 and they are located at 98 king street .
5 **User:** ok thanks, can you recommend a restaurant in the same area serving Indian food
6 **SOLIST:** curry garden serves Indian food and is located in the centre . would you like me to make a reservation ?
7 **User:** sure, please book a table for two people on thursday at 18:00
8 **SOLIST:** ok , i have a table for 2 at 18:00 on Thursday . your reference number is 1234567890 . is there anything else I can help you with ?
9 **User:** that's it, bye.
10 **SOLIST:** have a great time and please contact us if you need any further assistance . goodbye .
11 **User:** btw, can you tell me again what is the phone and address for the attraction ?
12 **SOLIST:** sure, the phone number for the Broughton house gallery is 01223314960 and they are located at 98 king street .
13 **User:** thanks bye
14 **SOLIST:** thank you for using our service . have a great day

Belief State:

```
{
“restaurant”:{“food”: “indian”, “area”: “centre”}
“booking”:{“data”: “Thursday”, “people”: “2”, “ time”: “18:00”}
“attraction”:{“type”: “museum”, “ name”: “broughton house gallery”, “area”: “ centre”}
}
```

(a) Belief state



```
"address": "98 king street",
"area": "centre",
"entrance fee": "free",
"id": "5",
"location": [
    52.207409,
    0.126738
],
"name": "broughton house gallery",
"openhours": "?",
"phone": "01223314960",
"postcode": "cb11ln",
"pricerange": "free",
"type": "museum"
```

```
"address": "106 Regent Street City Centre",
"area": "centre",
"food": "indian",
"id": "19214",
"introduction": "curry garden serves tradition
bangladeshi cuisine cooked with fresh produce
"location": [
    52.200187,
    0.126407
],
"name": "curry garden",
"phone": "01223302330",
"postcode": "cb21dp",
"pricerange": "expensive",
"type": "restaurant"
```

(b) Attraction DB entry (c) Restaurant DB entry

Figure 3: An interactive example.

Specifically in dialog domains, several latest works have adapted the pre-trained models to the task-oriented and chit-chat dialog systems. For chit-chat dialog systems, DialoGPT (Zhang et al., 2019b; Wolf et al., 2019b) and CGRG (Wu et al., 2020b) extended GPT-2 (Radford et al., 2019) to ground on dialog response generation settings. Plato (Bao et al., 2019) is a pre-trained discrete latent variable model for response generation. Meena (Adiwardana et al., 2020) and BST (Roller et al., 2020) pre-train extremely large models and have demonstrated expressive results on the social chit-chat conversation. For task-oriented dialogs, BERT-ToD (Wu et al., 2020a) adapts the pre-trained BERT (Devlin et al., 2019) to achieve super performance on four dialog subtasks. SC-GPT (Peng et al., 2020) is a pre-trained model for the NLG module that converts a dialog act into a response in natural language. The proposed SOLOIST generalize the idea to the entire dialog pipeline.

End-to-end Trainable Dialog Systems. The end-to-end trainable networks for dialog systems have been studied in (Wen et al., 2016; Lei et al., 2018). Though these methods have achieved promising results, they were usually designed for a specific domain, rendering difficulties in generalizing to multi-domains such as the recent MultiWOZ dataset (Budzianowski et al., 2018) and ConvLab (Lee et al., 2019). To tackle this, several models were proposed to handle the multi-domain dialog response generation (Pei et al., 2019; Budzianowski and Vulić, 2019; Mehri et al., 2019; Zhao et al., 2019; Wu et al., 2019; Zhang et al., 2019a). However, these works need a significant number of in-domain training examples to achieve good performance, facing challenges in the few-shot learning settings. In contrast, our SOLOIST can easily generalize to multiple new domains with a few labelled examples.

To the best of our knowledge, the most related work to ours is (Ham et al., 2020), which was the first attempt to leverage GPT-2 to fine-tune on the new task-oriented dialogs task. Hosseini-Asl et al. (2020) is a concurrent work following this line of research. However, our work is different from (Ham et al., 2020; Hosseini-Asl et al., 2020) in two major aspects: (*i*) We first pre-train our model on a large number of out-of-domain task-oriented dialog turns to endow the model with task-grounded language generation ability, then fine-tune it on new domains. However, Ham et al. (2020) and Hosseini-Asl et al. (2020) directly fine-tuned GPT-2 on new domains, which shows inferior performance than SOLOIST. (*ii*) The models in (Ham et al., 2020; Hosseini-Asl et al., 2020) require more expensive annotation, and not truly end-to-end trainable. It needs heuristic rules to handle different database search conditions. Further, it separately formulates POL and NLG, which requires annotations on dialog acts. While our model requires to annotate only the belief state, showing lower annotation cost than existing methods. It is fully trainable thanks to the simplified but effective input representations.

5 Conclusion

In this paper, we have presented SOLOIST. Unlike GPT-2, SOLOIST grounds response generation in user goals and knowledge for task completion. Machine teaching is used to boost the fine-tuning performance. Experimental results on two benchmarks demonstrate that SOLOIST creates new state-of-the art performance. When a few labelled examples are available in new domains, SOLOIST outperforms existing methods by a large margin.

We hope that SOLOIST can inspire the community to comprehensively explore the new paradigm for building task-oriented dialog systems: formulating task-oriented dialog as a single auto-regressive model, pre-training a grounded response generation model on heterogeneous dialog corpora, and adapting the pre-trained model to new tasks through fine-tuning using a handful task-specific examples via machine teaching.

Acknowledgments

We are grateful to the entire Philly Team inside Microsoft for providing our computing platform.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A corpus for adding memory to goal-oriented dialogue systems. *arXiv preprint arXiv:1704.00057*.

Siqi Bao, Huang He, Fan Wang, and Hua Wu. 2019. Plato: Pre-trained dialogue generation model with discrete latent variable. *arXiv preprint arXiv:1910.07931*.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *CoRR*, abs/1712.05181.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's gpt-2–how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. *arXiv preprint arXiv:1907.05774*.

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*.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Andy Cedilnik, and Kyu-Young Kim. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. *arXiv preprint arXiv:1909.05358*.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

Mihail Eric and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. *arXiv preprint arXiv:1705.05414*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2019. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

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*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, et al. 2019. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Xiang Li, Yaoqin Zhang, Zheng Zhang, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, and Jianfeng Gao. 2019. Convlab: Multi-domain end-to-end dialog system platform. *CoRR*, abs/1904.08637.

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)*.

Chunyuan Li, Xiang Gao, Yuan Li, Xiujun Li, Baolin Peng, Yizhe Zhang, and Jianfeng Gao. 2020a. Optimus: Organizing sentences via pre-trained modeling of a latent space. *arXiv preprint arXiv:2004.04092*.

Jinchao Li, Baolin Peng, Sungjin Lee, Jianfeng Gao, Ryuichi Takanobu, Qi Zhu, Minlie Huang, Hannes Schulz, Adam Atkinson, and Mahmoud Adada. 2020b. 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*.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. *arXiv preprint arXiv:1907.10016*.

Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2016. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*.

Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. 2019. A modular task-oriented dialogue system using a neural mixture-of-experts. *arXiv preprint arXiv:1907.05346*.

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*.

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*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Swadheen Shukla, Lars Liden, Shahin Shayandeh, Eslam Kamal, Jinchao Li, Matt Mazzola, Thomas Park, Baolin Peng, and Jianfeng Gao. 2020. Conversation learner–a machine teaching tool for building dialog managers for task-oriented dialog systems. *arXiv preprint arXiv:2004.04305*.

Patrice Y. Simard, Saleema Amershi, David Maxwell Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. 2017. Machine teaching: A new paradigm for building machine learning systems. *CoRR*, abs/1707.06742.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

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*.

Jason D Williams and Lars Liden. 2017. Demonstration of interactive teaching for end-to-end dialog control with hybrid code networks. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 82–85.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019a. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020a. ToD-BERT: Pre-trained natural language understanding for task-oriented dialogues.

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*.

Zeqiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, and Bill Dolan. 2020b. A controllable model of grounded response generation. *arXiv preprint arXiv:2005.00613*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2019a. Task-oriented dialog systems that consider multiple appropriate responses under the same context. *arXiv preprint arXiv:1911.10484*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019b. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. *arXiv preprint arXiv:1606.02560*.

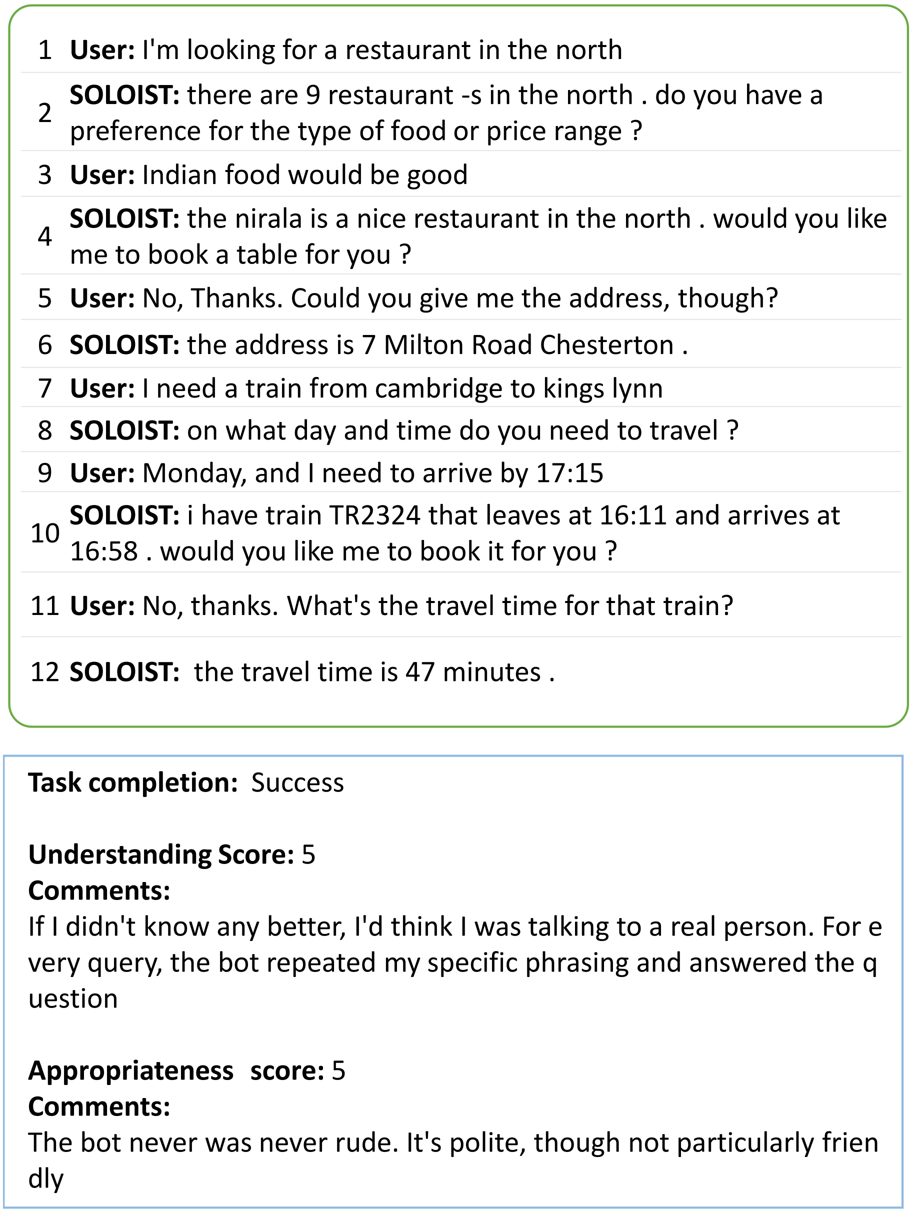
Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. *arXiv preprint arXiv:1902.08858*.

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. *CoRR*, abs/2002.04793.

Xiaojin Zhu. 2015. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

A Supplemental Material

More examples from human evaluation and its associated judges are show in Figure 4 and Figure 5.



1 **User:** I'm looking for a restaurant in the north

2 **SOLOIST:** there are 9 restaurant -s in the north . do you have a preference for the type of food or price range ?

3 **User:** Indian food would be good

4 **SOLOIST:** the nirala is a nice restaurant in the north . would you like me to book a table for you ?

5 **User:** No, Thanks. Could you give me the address, though?

6 **SOLOIST:** the address is 7 Milton Road Chesterton .

7 **User:** I need a train from cambridge to kings lynn

8 **SOLOIST:** on what day and time do you need to travel ?

9 **User:** Monday, and I need to arrive by 17:15

10 **SOLOIST:** i have train TR2324 that leaves at 16:11 and arrives at 16:58 . would you like me to book it for you ?

11 **User:** No, thanks. What's the travel time for that train?

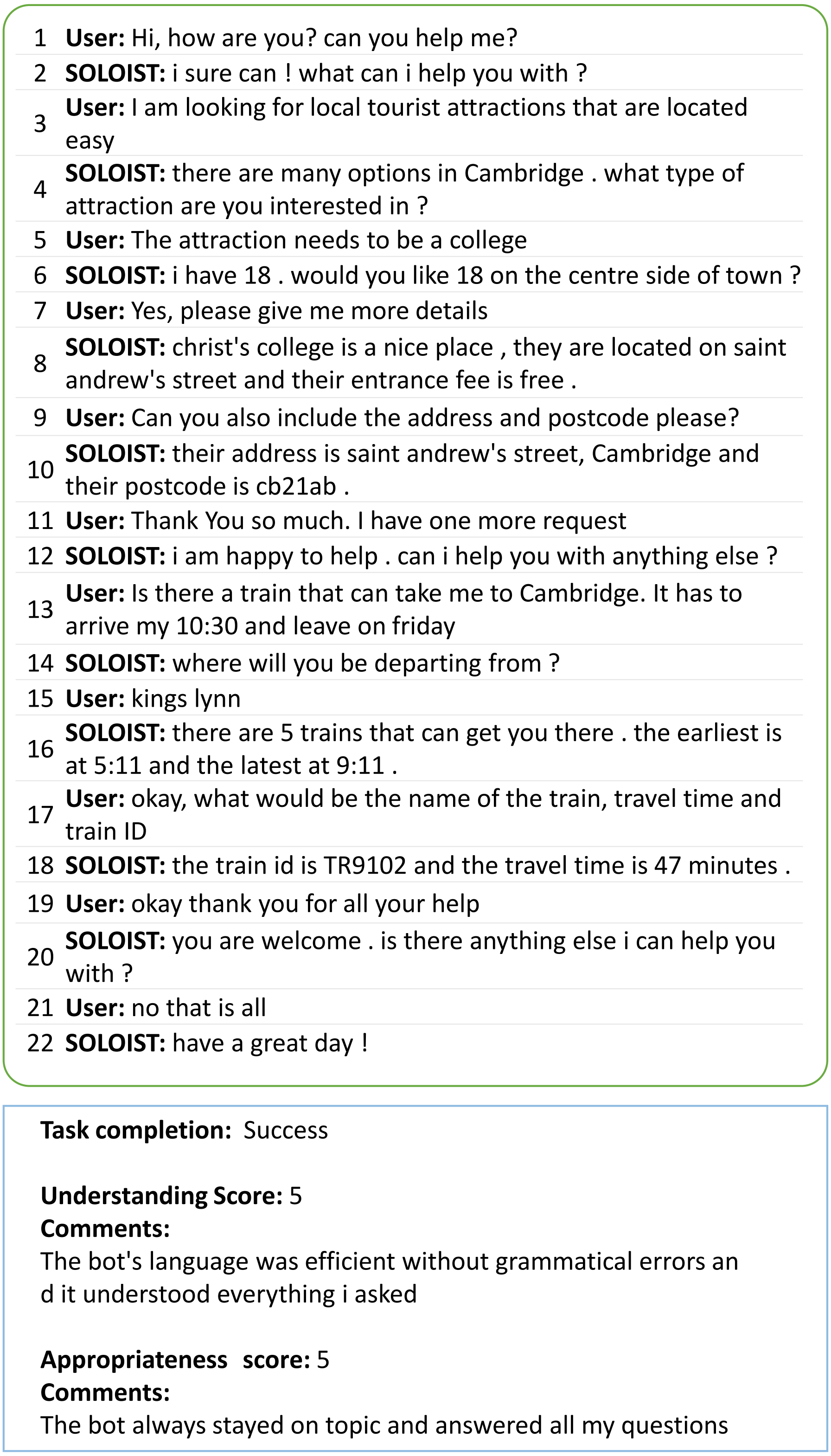
12 **SOLOIST:** the travel time is 47 minutes .

Task completion: Success

Understanding Score: 5
Comments:
If I didn't know any better, I'd think I was talking to a real person. For every query, the bot repeated my specific phrasing and answered the question

Appropriateness score: 5
Comments:
The bot never was never rude. It's polite, though not particularly friendly

Figure 4: An example from human evaluation and its associated judges from Mturks. The agent is quite responsive and answers every query from the user.



1 **User:** Hi, how are you? can you help me?

2 **SOLOIST:** i sure can ! what can i help you with ?

3 **User:** I am looking for local tourist attractions that are located easy

4 **SOLOIST:** there are many options in Cambridge . what type of attraction are you interested in ?

5 **User:** The attraction needs to be a college

6 **SOLOIST:** i have 18 . would you like 18 on the centre side of town ?

7 **User:** Yes, please give me more details

8 **SOLOIST:** christ's college is a nice place , they are located on saint andrew's street and their entrance fee is free .

9 **User:** Can you also include the address and postcode please?

10 **SOLOIST:** their address is saint andrew's street, Cambridge and their postcode is cb21ab .

11 **User:** Thank You so much. I have one more request

12 **SOLOIST:** i am happy to help . can i help you with anything else ?

13 **User:** Is there a train that can take me to Cambridge. It has to arrive my 10:30 and leave on friday

14 **SOLOIST:** where will you be departing from ?

15 **User:** kings lynn

16 **SOLOIST:** there are 5 trains that can get you there . the earliest is at 5:11 and the latest at 9:11 .

17 **User:** okay, what would be the name of the train, travel time and train ID

18 **SOLOIST:** the train id is TR9102 and the travel time is 47 minutes .

19 **User:** okay thank you for all your help

20 **SOLOIST:** you are welcome . is there anything else i can help you with ?

21 **User:** no that is all

22 **SOLOIST:** have a great day !

Task completion: Success

Understanding Score: 5
Comments:
The bot's language was efficient without grammatical errors and it understood everything i asked

Appropriateness score: 5
Comments:
The bot always stayed on topic and answered all my questions

Figure 5: An example from human evaluation and its associated judges from Mturks.