Soloist: Building Task Bots at Scale with Transfer Learning and Machine Teaching

Baolin Peng, Chunyuan Li, Jinchao Li
Shahin Shayandeh, Lars Liden, Jianfeng Gao

Microsoft Research, Redmond, United States
{bapeng,chunyl,jincli,shahins,lars.liden,jfgao}@microsoft.com

Abstract

We present a new method, Soloist,[1] that uses transfer learning and machine teaching to build task bots at scale. We parameterize classical modular task-oriented dialog systems using a Transformer-based auto-regressive language model, which subsumes different dialog modules into a single neural model. We pre-train, on heterogeneous dialog corpora, a task-grounded response generation 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 new tasks with a handful of task-specific dialogs via machine teaching, where training samples are generated by human teachers interacting with the system. Experiments show that (i) Soloist creates new state-of-the-art on well-studied task-oriented dialog benchmarks, including CamRest676 and MultiWOZ; (ii) in the few-shot fine-tuning settings, Soloist significantly outperforms existing methods; and (iii) the use of machine teaching substantially reduces the labeling cost of fine-tuning. The pre-trained models and codes are available at https://aka.ms/soloist.

1 Introduction

The increasing use of personal assistants and messaging applications has spurred interest in building task-oriented dialog systems (or task bots) 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 (e.g., Zhou et al., 2020; Adiwardana et al., 2020; Roller et al., 2020b; Gao et al., 2020; Peng et al., 2020a). 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 modular pipeline, which has four modules and executes sequentially (Young et al., 2013; Gao et al., 2019a), 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 users' 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, including Google's Dialog Flow,[2] Microsoft's Power Virtual Agents (PVA),[3] Facebook's Wit.ai,[4] Amazon's Lex,[5] and IBM's Watson Assistant.[6] They are designed mainly to help develop systems *manually*, namely, 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., tens of thousands of bots for different tasks) can be prohibitively expensive.

[1] TaSk-Oriented DiaLOg wIth a Single pre-Trained model. In this paper, Soloist refers to both the proposed bot building method and the dialog model or system developed using the method.

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

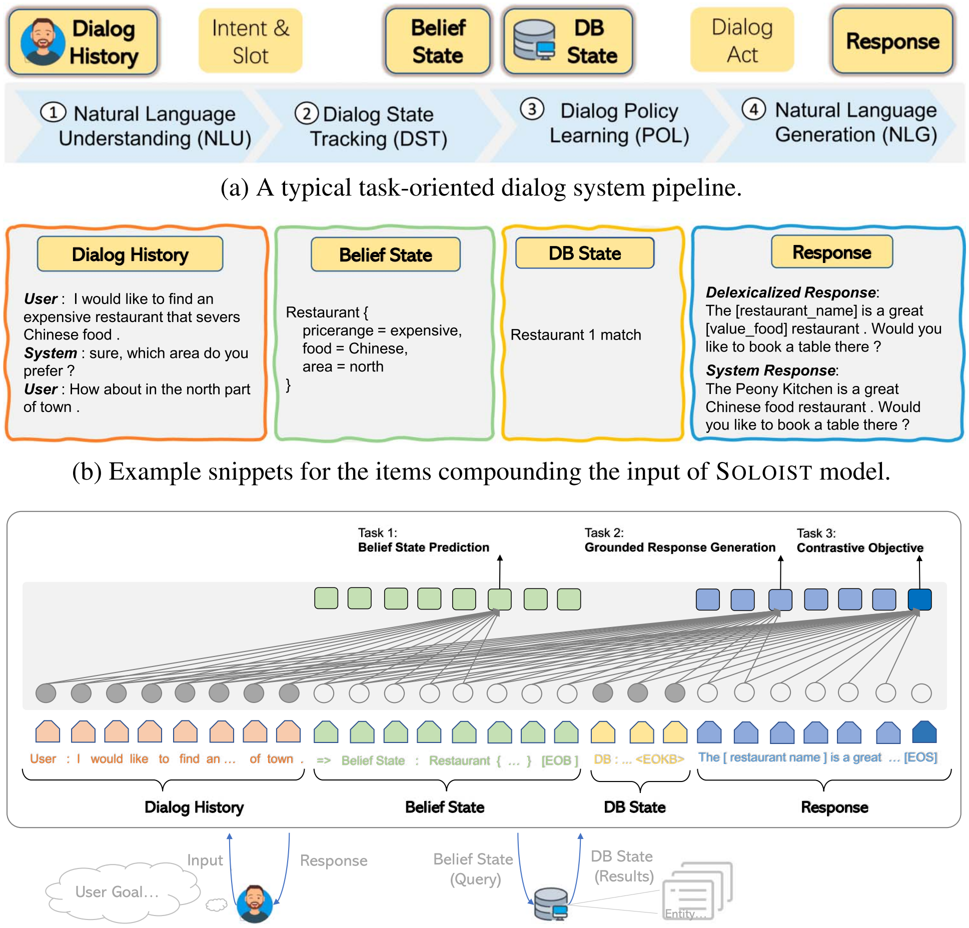
[3] https://powervirtualagents.microsoft.com/.

[4] https://wit.ai/.

[5] https://aws.amazon.com/lex/.

[6] https://www.ibm.com/watson/.

Transactions of the Association for Computational Linguistics, vol. 9, pp. 807–824, 2021. https://doi.org/10.1162/tacl_a_00399
Action Editor: James Henderson. Submission batch: 7/2020; Revision batch: 1/2021; Published 8/2021.


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)
Dialog History
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 }
DB State
Restaurant 1 match
Response
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 ?
Task 1: Belief State Prediction
Task 2: Grounded Response Generation
Task 3: Contrastive Objective
User : I would like to find an … of town .
=> Belief State : Restaurant { … } [EOB]
DB : ... <EOKB>
The [restaurant name] is a great … [EOS]
Dialog History
Belief State
DB State
Response
User Goal…
Input
Response
Belief State (Query)
DB State (Results)
Entity…

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

(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 parameterize 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.

With the recent advances in neural approaches to conversational AI (Gao et al., 2019a), researchers have been 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., 2019b; Zhu et al., 2020), and Conversation Learner (Shukla et al., 2020) are made to allow the use of data-driven approaches based on machine learning and machine teaching to develop dialog modules. End-to-end trainable dialog systems have also been studied (e.g., Wen et al., 2017; Zhao and Eskenazi, 2016; Li et al., 2017; Williams et al., 2017; Lei et al., 2018; Gao et al., 2019a; Zhang et al., 2020b). 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 real-world applications.

In this paper, we propose a novel method of building task bots at scale, 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: Big language models pre-trained on large amounts of raw text (e.g., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and UniLM (Dong et al., 2019)) can be effectively fine-tuned for a wide range of NLP tasks with few in-domain labels. Recently, these pre-trained language models have also been employed to develop dialog modules such as NLU and DST (Henderson et al., 2020; Coope et al., 2020; Wu et al., 2020a). The proposed SOLOIST uses a similar pre-training-and-fine-tuning framework for building end-to-end dialog systems. We parameterize a task bot using a Transformer-based auto-regressive language model, which subsumes different dialog modules (i.e., NLU, DST, POL, and NLG) into a single neural model. Task bot building proceeds in two stages: (i) In the pre-training stage, initialized using GPT-2 (Radford et al., 2019), we train a Transformer-based, task-grounded, response generation model using large heterogeneous dialog corpora. The model learns the primary task completion skills such as DST and POL, and can generate dialog responses *grounded* in user goals and real-world knowledge for task completion. (ii) In the fine-tuning stage, we adapt the pre-trained Soloist model to complete a specific (new) task using a handful of task-specific dialogs via machine teaching, where training samples are generated by human teachers interacting with the system (Zhu, 2015; Shukla et al., 2020).

We show through a comprehensive empirical study that Soloist is an effective method of building task bots at scale by successfully transferring two capabilities from the pre-trained model to a new task bot: (i) the capability of NLU and NLG learned on raw text, and (ii) the capability of grounding system responses in user goals and real-world knowledge for task completion, learned on the out-domain dialog corpora.

Soloist achieves state-of-the-art performance on two well-studied task-oriented dialog benchmarks, lifting the combined score by 10 points in automatic evaluation, and the success rate by 20 points in human evaluation. In the few-shot fine-tuning settings, Soloist adapts to the 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 method for developing task bots at scale. Instead of collecting, labeling data, and building one bot per task, we can pre-train a task-grounded response 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 pipeline that produces a sequence, through concatenating the input-output pair 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 a dialog system (NLU, DST, POL, NLG) can be formulated as a *single* auto-regressive model.

GPT-2 (Radford et al., 2019) is a state-of-the-art (SoTA) auto-regressive language model trained on large amounts of open Web text data. Although after being fine-tuned using conversational data, GPT-2 can respond to users with realistic and coherent continuations about any topic of their choosing (Zhang et al., 2020c), 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. Nevertheless, unlike GPT-2, Soloist is pre-trained to generate responses grounded in user goals and real-world knowledge for task completion. While GPT-2 is a language model for text prediction, Soloist is a stateful decision-making model for task completion, with the capabilities of tracking dialog states, selecting best system actions, and so on. Thus, Soloist is pre-trained using task-oriented dialog sessions annotated with grounding information, i.e., user goals, dialog belief states, DB states, and system responses. 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 dialog history up to the current dialog turn, $\boldsymbol{b}$ is the dialog 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 generated using some automatic post-processing. Each item in $\boldsymbol{x}$ is by itself a sequence of tokens, as illustrated by the 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 publicly available heterogeneous dialog corpora with labels of belief states and DB states. The pre-trained model can be fine-tuned to any new task to generate responses grounded in task-specific user goals and a 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 neural model parameterized by $\boldsymbol{\theta}$ to characterize the sequence generation probability $p_{\boldsymbol{\theta}}(\boldsymbol{x})$. We use 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, we can further factorize them in the left-to-right auto-regressive manner, respectively.

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

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

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

Task 2: Grounded Response Generation. A delexicalized response of length T_r, $\boldsymbol{r} = [r_1, \cdots, r_{T_r}]$, is generated by our model token-by-token from left to right, grounded in dialog history $\boldsymbol{c}$, belief state $\boldsymbol{b}$ and DB state $\boldsymbol{s}$. The corresponding training objective is defined as

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

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

Task 3: Contrastive Objective. A contrastive objective is employed to promote the matched items (positive samples $\boldsymbol{x}$) while driving down the mismatched items (negative samples $\boldsymbol{x}'$). The negative samples are generated 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 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 in the sequence are matched ($y = 1$) or mismatched ($y = 0$). The contrastive object is cross-entropy defined as:

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

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

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

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

Figure 1(c) illustrates the model architecture and learning objectives. The model is auto-regressive in a left-to-right manner, with each of the three training tasks labeled on its corresponding output (i.e., sub-sequence separated by a special token).

Implementation Details. Each dialog turn in training data is processed to form a sequence of tokens consisting of four items $(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{r})$. For example, the dialog turn of Figure 1 (b) is represented 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, tokenized using byte pair encodings (Sennrich et al., 2016), can be readily used for multi-task training, as shown in Figure 1(c). The implementation of Soloist is based on Huggingface PyTorch Transformer (Wolf et al., 2020). The task-grounded pre-training of Soloist uses the public 117M-parameter GPT-2 as initialization.

Name	#Dialog	#Utterance	Avg. Turn	#Domain
task-grounded pre-training:				
Schema	22,825	463,284	20.3	17
Taskmaster	13,215	303,066	22.9	6
fine-tuning:				
MultiWOZ2.0	10,420	71,410	6.9	7
CamRest676	676	2,744	4.1	1
Banking77	–	25,716	–	21
Restaurant-8k	–	8,198	–	1

Table 1: Dialog corpora. The datasets in the upper block are used for task-grounded pre-training, and the datasets in the lower block are for fine-tuning.

Adam (Kingma and Ba, 2014) with weight decay is used for pre-training. Table 1 shows the dialog corpora (Kim et al., 2019; Rastogi et al., 2020; Byrne et al., 2019) used for task-grounded pre-training. To ensure there is no overlap between pre-training and fine-tuning datasets, we exclude the data akin to MultiWOZ (Budzianowski et al., 2018), CamRest676 (Wen et al., 2017), Banking77 (Casanueva et al., 2020), Restaurant-8k (Coope et al., 2020).

2.3 Fine-Tuning and Machine Teaching

When deploying Soloist to a new task, we collect task-specific $\boldsymbol{x}$ in the same format as that used for pre-training as (1). When $\boldsymbol{x}$ 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 task-specific data is often unavailable, or noisy/incomplete beforehand. One may deploy the dialog system and acquire high-quality task-specific labels (e.g., belief state and system response) for each dialog turn using machine teaching. 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 system's performance (Simard et al., 2017; Zhu, 2015; Williams and Liden, 2017; Shukla et al., 2020).

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, as illustrated in Figure 2. The corrected task-specific dialog turns are used to fine-tune the model.

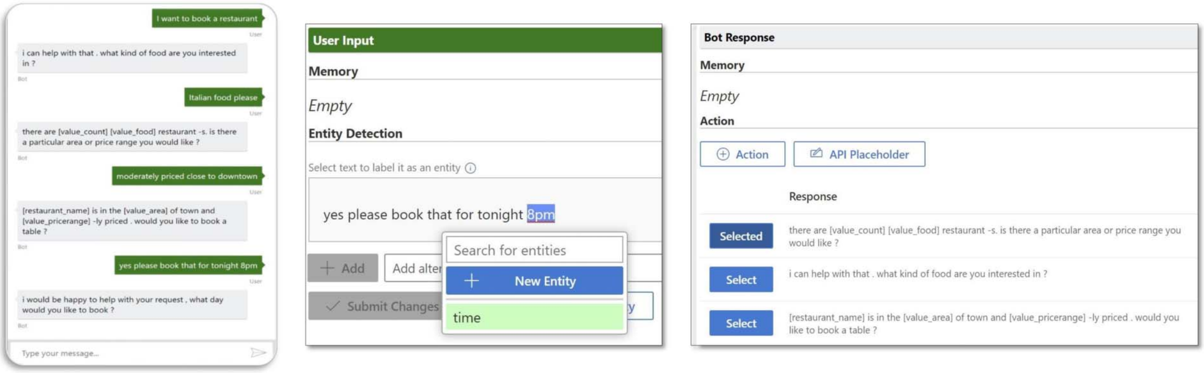
Implementation Details. To adapt a pre-trained Soloist to a new task in our experiments, we always fine-tune Soloist using a small amount of pre-collected task-specific dialogs, and then continue to fine-tune it via machine teaching, as detailed in Section 3.3. Training examples are truncated to ensure a maximal length of 512. The pre-trained models are fine-tuned with a mini-batch of 6 on 8 Nvidia V100 until no progress is observed on validation data 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. For the machine teaching experiment,pre-trained models are fine-tuned with SGD on a single Nvidia V100.

3 Experiments

This section evaluates the proposed Soloist to answer three questions: `Q1`: How does Soloist perform on standard benchmarks compared to SoTA methods? `Q2`: Does Soloist meet the goal of effectively generalizing to new domains in the few-shot fine-tuning setting? `Q3`: how effective machine teaching is for fine-tuning? Note that we employ the conventional fine-tuning method *without* machine teaching for a fair comparison when studying `Q1` and `Q2`.

3.1 Experimental Setup

Dialog Datasets for Fine-Tuning. We validate the end-to-end dialog system performance of Soloist on two well-studied datasets. (*i*) CamRest676 (Wen et al., 2017) is a single-domain task-oriented dialog corpus. 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



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 ?

Type your message...

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
Selected	there are [value_count] [value_food] restaurant -s. is there a particular area or price range you would like ?
Select	i can help with that . what kind of food are you interested in ?
Select	[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).

names such as restaurant name, phone number, and postcode. (*ii*) MultiWOZ dataset (Budzianowski et al., 2018) is a multi-domain task-oriented dialog dataset. It contains 8438/1000/1000 for training/validation/testing, respectively. Each dialog session contains 1 to 3 domains, such as 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. The first two metrics relate to the dialogue task completion—whether the system has provided an appropriate entity (`Inform`) and then answered all the requested attributes (`Success`). `BLEU` evaluates how natural the generated responses are compared to that generated by human agents. A combined score (`Combined`) is also reported using $\texttt{Combined} = (\texttt{Inform} + \texttt{Success}) \times 0.5 + \texttt{BLEU}$ as an overall quality measure.

Baselines. We compare Soloist with several strong baselines, which hold SoTA on the CamRest676 or MultiWOZ datasets. (*i*) Multi-Action Data Augmentation (DAMD) (Zhang et al., 2020b) is a 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*) GPT fine-tuning (Budzianowski and Vulić, 2019) is fine-tuned on GPT-2 to generate responses based on the dialog state and history. (*iv*) ARDM (Wu et al., 2019b) 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 a BERT-based dialog policy and graph structure dialog act representations.

3.2 End-to-End Evaluation

CamRest676. Table 2 shows the result and lists annotations used by different models. Soloist achieves the best scores in all the metrics. ARDM performs similarly to Soloist in terms of `Success` and `BLEU`. However, ARDM cannot track dialog states and requires a separately trained state tracker to accomplish tasks. GPT-2 fine-tuned with task-specific data works reasonably well but lags behind Soloist by a large margin. Sequicity, which uses a jointly trained model with belief state and policy annotations, underperforms Soloist. This result also shows that, compared to other end-to-end models, Soloist not only achieves better performance but requires lower labeling cost for fine-tuning due to the use of task-grounded pre-training.

MultiWOZ. The result is shown in Table 3. Soloist achieves the best performance in terms of `Inform`, `Success`, and `Combined`, lifting the previous SoTA by a significant margin (e.g., about 10 points improvement in `Combined` over DAMD). Soloist also outperforms the method of Ham et al. (2020), where GPT-2 is fine-tuned and applied for end-to-end dialog modeling. Compared to the

Model	Annotations		Evaluation Metrics			
	Belief 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 fine-tuning (Budzianowski and Vulić, 2019)			–	86.20	19.20	–
ARDM[1] (Wu et al., 2019b)			–	87.10	25.20	–
Soloist	✓		**94.70**	**87.10**	**25.50**	**116.40**

[1] ARDM is not fully E2E, as it requires a rule-based dialog state tracker.

Table 2: End-to-End evaluation on CamRest676. Results of existing methods are from Wu et al. (2019b).

Model	Annotations		Evaluation Metrics			
	Belief 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., 2019b)	✓	✓	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., 2020b)	✓	✓	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 setting.

Table 3: End-to-end evaluation on MultiWOZ.

classical modular dialog systems such as DAMD, Soloist uses 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.

Domain	Attra.	Train	Hotel	Rest.	CamRest676
#Train	50	50	50	50	20
#Valid	50	50	50	50	136
#Test	100	200	200	200	136

Table 4: Data statistics for domains used in few-shot evaluation. Attra. denotes Attraction domain and Rest. means Restaurant.

Model	CamRest676		
	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$_L$	**88.05**	**84.79**	18.88

Table 5: End-to-end evaluation on CamRest676 in the few-shot fine-tuning setting.

3.3 Few-Shot Evaluation

It is desirable for task bots to effectively generalize to new tasks with few task-specific training samples. Thus, the few-shot fine-tuning setting is a more realistic setting for evaluating dialog systems. Unfortunately, the existing task-oriented dialog benchmarks typically contain for each task hundreds to thousands of dialogs. Therefore, we re-organize CamRest676 and MultiWOZ to simulate the few-shot fine-tuning setting for end-to-end evaluation.[7] We sample from the MultiWOZ dataset the dialog tasks that contain only one domain. Attraction, Train, Hotel, and Restaurant domains are used. We do not use the domains of Police, Taxi, and Hospital, as they do not require explicitly tracking dialog states for task completion. For each domain, we randomly sample 50 dialog sessions for training and validation and 200 dialog sessions for testing. The only exception is the Attraction domain, which has 100 sessions for testing. For CamRest676, we randomly sample 20 sessions. Details are shown in Table 4.

Table 5 and 6 report the end-to-end performance in the few-shot fine-tuning settings on CamRest676 and MultiWOZ, respectively. On all the domains, Soloist obtains substantially better performance in all the metrics. Removing task-grounded pre-training significantly hurts the performance of Soloist, although Soloist

[7] 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., 2020b)	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
$\text{Soloist}_{\text{L}}$	**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 6: End-to-end evaluation on MultiWOZ in the few-shot fine-tuning setting.

Model	1%			5%			10%			20%		
	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑
DAMD (Zhang et al., 2020b)	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 7: End-to-end evaluation on MultiWOZ with varying sizes of task-specific training data for fine-tuning.

Model	Attraction			Train			Hotel			Restaurant		
	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑	Inform ↑	Success ↑	BLEU ↑
Soloist	45.00	19.00	7.67	67.68	58.08	7.13	33.50	**22.50**	**8.70**	50.50	10.00	8.61
Soloist +Extra	63.00	41.00	11.08	65.15	57.58	**9.74**	41.50	19.00	7.96	44.50	27.00	9.77
Soloist +Teach	**78.00**	**45.00**	**11.90**	**68.18**	**63.64**	9.45	**46.50**	**22.50**	7.68	**53.00**	**32.00**	**9.81**

Table 8: Machine teaching results. Soloist is trained with 10 examples for each domain. Soloist+Teach indicates continual training with 5 dialogs recommended by CL with human teacher corrections. Soloist+Extra indicates continual training using 5 randomly sampled dialogs with full annotations.

without task-grounded pre-training still consistently outperforms DAMD in all the domains. Soloist without task-grounded pre-training is conceptually similar to Ham et al. (2020), but is architecturally simpler and needs fewer annotations. The result verifies the importance of task-grounded pre-training on annotated dialog corpora, allowing Soloist to learn how to track dialog and database states to accomplish a task. To study the impact of using larger model size, we build a large version of Soloist, $\text{Soloist}_{\text{L}}$, which is task-grounded pre-trained on the same data but using GPT-2_{medium} with 345M parameters as initialization. $\text{Soloist}_{\text{L}}$ consistently outperforms Soloist by a large margin. It indicates that a larger model is a better few-shot learner, exhibiting stronger generalization ability with limited in-domain data. We leave it to future work to significantly scale up Soloist.

We conduct experiments to fine-tune Soloist by varying the percentage of task-specific training samples, ranging from 1% (80 examples) to 20% (1600 examples), on the MultiWOZ dataset. As shown in Table 7, Soloist consistently outperforms DAMD for a wide range of dataset sizes, and the improvement is more substantial when smaller numbers of in-domain examples are used for fine-tuning.

3.4 Machine Teaching Results

The machine teaching module of Conversational Learner (CL) (Shukla et al., 2020) allows human teachers (dialog authors) to select and visualize dialogs, find potential problems, and provide corrections or additional training samples to improve the bot's performance. We use CL to evaluate the effectiveness of machine teaching for task bot fine-tuning. In our experiment, we first sample 10 dialogs from each domain to fine-tune Soloist as described in Section 3.3. The result is presented in the first row of Table 8. We then deploy the model to interact with human users via CL. The row of Soloist+Teach shows the result of machine teaching, where a human teacher has manually corrected 5 dialogs, which are recommended by CL using a ranking heuristic based on perplexity. The corrections are utilized to continually fine-tune the deployed system.

Table 8 shows that Soloist+Teach consistently improves `Combined` by a large margin compared with that without human teaching. Soloist+Extra is used as an ablation baseline, where 5 randomly selected dialogs with full annotations from experts are added as extra examples to fine-tune the model. It shows lower performance than machine teaching. Figure 3 demonstrates the

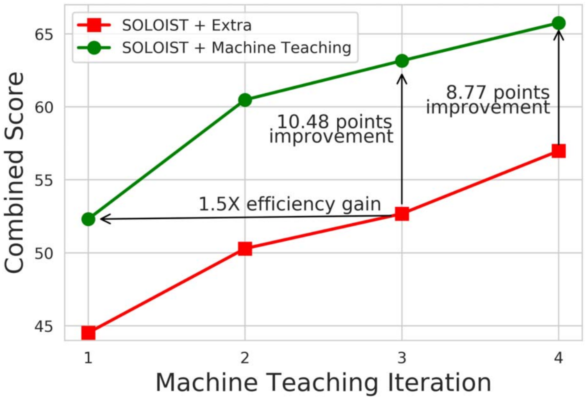


Figure 3: Machine teaching performance of different iterations in `Restaurant` domain. Machine teaching with CL achieves near 1.5X efficiency gain (i.e., the 1st iteration used 15 dialogs while the 3rd iteration has 25 dialogs) and boosts performance by 10 points compared with that without teaching.

performance of Soloist in `Restaurant` by repeating the above machine teaching process in multiple iterations. We observe that in the second iteration of machine teaching Soloist+Teach improves `Combined` by more than 8 points while Soloist+Extra achieves 5 points higher. The result demonstrates the effectiveness of our two-step fine-tuning scheme to deploy Soloist for a new task (domain). In terms of machine teaching cost, taking the `restaurant` domain as an example, we assume that one slot-value pair of belief state correction counts as one edit and a response correction counts as 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 6×.

3.5 Component-Wise Evaluation

This section evaluates Soloist on two NLU tasks (i.e., intent classification and slot filling), the DST task and the response generation task. We show that although Soloist is an end-to-end dialog model, it also performs well on these component tasks.

Intent Classification The task is to classify a user utterance into one of several pre-defined classes (intents). We follow the experiment setting of Casanueva et al. (2020). The last hidden state of Soloist is used as the sequence representation for classification. Several baseline methods are used for comparison. BERT-fixed and BERT-tuned are fine-tuned on BERT, with BERT parameters fixed and updated during fine-tuning, respectively. A linear classifier with a softmax layer is added on top of BERT for classification. Universal Sentence Encoder and ConveRT are sentence encoders tailored for modeling sentence pairs, and are trained for optimizing the conversational response selection task. The results in Table 9 show that Soloist is comparable with SoTA intent classification models. Soloist is the best performer when the full dataset is used for fine-tuning but its performance deteriorates more quickly than USE+ConveRT when fewer samples are used for fine-tuning. It is interesting to investigate whether incorporating intent classification tasks in task-grounded pre-training can boost Soloist's performance. We leave it to future work.

Model	`Banking77`		
	10	30	Full
BERT-Fixed	67.55	80.07	87.19
BERT-Tuned	83.42	90.03	93.66
USE	84.23	89.74	92.81
ConveRT	83.32	89.37	93.01
USE+ConveRT	**85.19**	**90.57**	93.36
Soloist	78.7	89.28	**93.80**

Table 9: Intent classification accuracy scores (5 runs average) on Banking77 with varying number of training examples (10, 30 examples for each intent, and full training examples. The baseline results are cited from Casanueva et al. (2020).

Slot Filling. We follow the experiment setting of Coope et al. (2020) and formulate slot filling as a turn-based span extraction problem. The results in Table 10 show that Soloist performs significantly better than the SoTA method Span-ConveRT, a variant of ConveRT designed explicitly for slot filling. The gap is wider when fewer examples are used for training. For example, when 64 samples are used for training, Soloist outperforms Span-ConveRT by 20 points in F1 score.

Dialog State Tracking. We compare the dialog state tracking capability of Soloist with several strong baselines on MultiWOZ 2.0 and 2.1. The results in Table 11 show that Soloist achieves the best performance on MultiWOZ2.1 and similar performance to DST-Picklist (Zhang et al., 2020a), which requires pre-defined task ontology to guide state tracking. In comparison with Simple-TOD (Hosseini-Asl et al., 2020) that is based on GPT-2,

Fraction	Soloist	Span-ConveRT	V-CNN-CRF	Span-BERT
1 (8198)	**0.98**	0.96	0.94	0.93
1/2 (4099)	**0.95**	0.94	0.92	0.91
1/4 (2049)	**0.93**	0.91	0.89	0.88
1/8 (1024)	**0.89**	**0.89**	0.85	0.85
1/16 (512)	**0.84**	0.81	0.74	0.77
1/32 (256)	**0.79**	0.64	0.57	0.54
1/64 (128)	**0.74**	0.58	0.37	0.42
1/128 (64)	**0.61**	0.41	0.26	0.30

Table 10: Average F1 scores across all slots for Restaurant-8K with varying training set fractions. Numbers in parentheses represent training set sizes. The baseline results are quoted from Coope et al. (2020).

Soloist obtains 1.13% higher joint goal accuracy. We attribute the gain to the task-grounded pre-training that equips Soloist with task completion skills including dialog state tracking.

Context-to-Response. In this task, systems need to generate responses given the ground-truth belief state and DB search result (Wen et al., 2017). The results on MultiWOZ 2.0 are shown in Table 12. Soloist achieves the best performance in terms of Inform and Success but performs slightly worse in BLEU. The Combined score of Soloist is comparable with the current SoTA method DAMD. However, DAMD uses the labels of dialog act on both the user and system sides, which demands significantly higher labeling efforts than Soloist for model training. HDSA achieves the best BLEU score. Compared with HDSA, Soloist is much simpler and able to perform better in terms of Combined. Soloist outperforms ARDM in Combined. It is worth mentioning that ARDM cannot perform dialog state tracking and requires an extra dialog state tracker to accomplish tasks. These results show that Soloist can learn dialog policies accurately and generate natural language responses in the multi-domain scenario.

3.6 Human Evaluation Results

We conduct human evaluation to assess the quality of Soloist interacting with human users. Following the evaluation protocol in the DSTC8 track 1 challenge (Kim et al., 2019), we host the best performed Soloist on the validation set in MultiWOZ domain in the back-end 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 experience at the end of a session using four metrics. (*i*) Success evaluates task completion. (*ii*) Under. (language understanding score) ranging from 1 (bad) to 5 (good) indicates the extent to which the system understands user inputs. (*ii*) Appr. (response appropriateness score) scaling from 1 (bad) to 5 (good) denotes whether the response is appropriate and human-like. (*iv*) Turns is the average number of turns in a dialog overall successful dialog sessions. Turks are further required to write down a justification of giving a specific rating. In total, 120 dialog sessions are gathered for analysis.

Model	Joint Goal Accuracy ↑	
	MWoz2.0	MWoz2.1
MDBT (Ramadan et al., 2018)	15.57	–
GLAD (Zhong et al., 2018)	35.57	–
GCE (Nouri and Hosseini-Asl, 2018)	36.27	–
FJST (Eric et al., 2020)	40.20	38.00
HyST (Goel et al., 2019)	44.24	–
SUMBT (Lee et al., 2019a)	46.65	–
TOD-BERT (Wu et al., 2020a)	–	48.00
Neural Reading (Gao et al., 2019b)	41.10	–
TRADE (Wu et al., 2019a)	48.62	45.60
COMER (Ren et al., 2019)	48.79	–
NADST (Le et al., 2020)	50.52	49.04
DSTQA (Zhou and Small, 2019)	51.44	51.17
SOM-DST (Kim et al., 2020)	51.38	52.57
DST-Picklist (Zhang et al., 2020a)	**53.30**	–
MinTL (Lin et al., 2020)	52.10	53.62
SST (Chen et al., 2020)	51.17	55.23
Tripy (Heck et al., 2020)	–	55.29
Simple-TOD (Hosseini-Asl et al., 2020)	–	55.72
Soloist	53.20	**56.85**

Table 11: Dialog state tracking results on MultiWOZ 2.0 and 2.1.

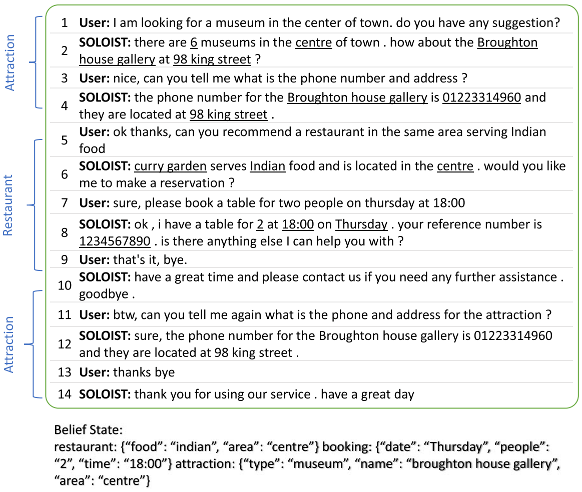
Table 13 shows the human assessment results on MultiWOZ. The results are consistent with the automatic evaluation. Soloist achieves substantially better performance than other systems over all the metrics. Moreover, Soloist outperforms the DSTC8 Track 1 Winner by a much larger margin in Success (+20 points) in human evaluation than that in automatic evaluation (+10 points in Table 3). We attribute this to the fact that Turks use more diverse language to interact with the target bots in interactive human evaluation than that in the pre-collected MultiWOZ dataset and the use of heterogeneous dialog data for task-grounded pre-training makes Soloist a more robust task bot than the others. In many test cases against Soloist, Turks comment that they feel like they are talking to a real person.

Model	Annotations		Evaluation Metrics			
	Belief 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 fine-tuning (Budzianowski and Vulic, 2019)	✓		70.96	61.36	19.05	85.21
Structured Fusion (Mehri et al., 2019b)	✓	✓	82.70	72.10	16.34	93.74
LaRL (Zhao et al., 2019)	✓		82.80	79.20	12.80	93.80
MD-Sequicity (Zhang et al., 2020b)	✓	✓	86.60	71.60	16.68	95.90
HDSA (Chen et al., 2019)	✓	✓	82.90	68.90	**23.60**	99.50
ARDM (Wu et al., 2019b)			87.40	72.80	20.60	100.70
DAMD (Zhang et al., 2020b)	✓	✓	89.20	77.90	18.60	102.15
SOLOIST	✓		**89.60**	**79.30**	18.03	**102.49**

Table 12: Context-to-response evaluation on MultiWOZ.

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 13: Human evaluation results. The results except SOLOIST are quoted from Li et al. (2020b).


Attraction
Restaurant
Attraction

1 **User:** I am looking for a museum in the center of town. do you have any suggestion?

2 **SOLOIST:** 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 **SOLOIST:** 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 **SOLOIST:** 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 **SOLOIST:** 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 **SOLOIST:** 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 **SOLOIST:** sure, the phone number for the Broughton house gallery is 01223314960 and they are located at 98 king street .

13 **User:** thanks bye

14 **SOLOIST:** thank you for using our service . have a great day

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

Figure 4: An interactive example.

Figure 4 depicts a dialog example where a user interacts with SOLOIST to complete a multi-domain task. The user starts the conversation by asking for a recommendation of a museum in the center of town. SOLOIST identifies the user intent, and provides a recommendation based on the search result from an attraction DB. Then, the user wants to book a table in a restaurant in the same area. We can see that through the conversation, SOLOIST develops belief state, which can be viewed as the system's understanding of what the user needs and what is available in the DB. Based on the belief state and DB state, SOLOIST picks the next action, either asking for clarification or providing the user with information being requested. This example also demonstrates that SOLOIST is able to deal with some NLU challenges displayed often in human conversations, such as co-reference resolution. For example, SOLOIST understands that the ‘‘same area’’ at Turn 5 refers to ‘‘centre of town’’, and then identifies a proper entity from the restaurant booking DB to make the reservation.

4 Related Work

Dialog Systems. Dialog systems are typically grouped into two categories, task-oriented systems and social chatbots (e.g., Chen et al., 2017; Gao et al., 2019a; Roller et al., 2020a; Zhou et al., 2020). Recently many variants have been developed to extend the scope of dialog systems, including empathetic dialog systems (Ma et al., 2020; Zhou et al., 2018), chatbots for sentiment analysis (Li et al., 2020c), dialog systems with commonsense knowledge (Young et al., 2018; Shuster et al., 2020), or using audio features (Young et al., 2020). In this paper, we focus on end-to-end dialog models for task-oriented systems.

Pre-Trained Language Models. Recent advances on self-supervised learning have witnessed the blooming of large-scale pre-trained language models (e.g., Devlin et al., 2019; Radford et al., 2019; Dong et al., 2019), which achieve SoTA performance on a variety of language understanding and generation tasks. The closest to SOLOIST are GPT-2 (Radford et al., 2019) and its variants that ground language generation in 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).

Recently, pre-trained language models have been adopted to develop task-oriented and chit-chat dialog systems. To name a few examples of chit-chat dialog systems: DialoGPT (Zhang et al., 2020c), TransferTransfo (Wolf et al., 2019) and CGRG (Wu et al., 2020b) adapt GPT-2 using human conversational data for response generation. Plato (Bao et al., 2020) pre-trains a discrete latent variable model for response generation. Meena (Adiwardana et al., 2020) and BST (Roller et al., 2020b) pre-train large models on conversational data and have demonstrated expressive performance in generating social chit-chat dialogs. For task-oriented dialogs, Mehri et al. (2019a) explores different pre-training methods for dialog context representation learning. TOD-BERT (Wu et al., 2020a) adapts the pre-trained BERT to achieve strong performance on four dialog sub-tasks. ConveRT (Henderson et al., 2020) pre-trains a model on Reddit data for intent classification and response selection. Span-ConveRT (Coope et al., 2020) extends the framework to entity extraction. SC-GPT (Peng et al., 2020b) uses a pre-trained language model to convert a dialog act to a natural language response. All these works use the pre-training and fine-tuning framework. However, they follow the modular architecture of task bots, and the pre-trained models are used for improving individual dialog modules such as NLU and DST. Soloist generalizes these methods to the entire dialog pipeline, building an end-to-end dialog system.

End-to-End Trainable Dialog Systems. The end-to-end dialog systems based on neural models have been studied in Wen et al. (2017); Li et al. (2017); Lei et al. (2018); Xu et al. (2019). Although these methods have achieved promising results, they are designed for specific domains, rendering difficulties in generalizing to multi-domains such as MultiWOZ. Dialog models that can handle multi-domain tasks are studied in (Pei et al., 2019; Budzianowski and Vulić, 2019; Mehri et al., 2019b; Zhao et al., 2019; Wu et al., 2019b; Zhang et al., 2020b; Peng et al., 2017). However, these works require large amounts of in-domain labels to achieve good performance. In contrast, Soloist can effectively adapt to a new task in the few-shot fine-tuning settings.

The most related work to ours is Ham et al. (2020), which is the first attempt to fine-tune GPT-2 to build end-to-end dialog models. Hosseini-Asl et al. (2020) take a similar approach, and is a concurrent work of Soloist. However, Soloist differs from these two methods in two major aspects. The first is the use of task-grounded pre-training that allows Soloist to learn primary task completion skills, such as tracking dialog states and select system actions. These skills can be easily reused and adapted (e.g., via few-shot fine-tuning) to solve new dialog tasks, leading to a much higher task success rate, as reported in Section 3. The second is that the annotation cost required for training Soloist is much lower than that of Ham et al. (2020) or Hosseini-Asl et al. 2020. Training Soloist requires only belief states as labels. But training of Ham et al. (2020) and Hosseini-Asl et al. (2020) requires labeling each dialog turn with dialog acts. In addition, while Soloist is end-to-end trainable, the other two models are not and need heuristic rules to handle different database search conditions.

5 Conclusion

Soloist is a method of building task bots at scale with transfer learning and machine teaching. Unlike GPT-2, Soloist is pre-trained in a task-grounded manner. So, it can generate responses grounded in user goals and real-world knowledge for task completion. Experiments show that Soloist creates new SoTA on two popular task-oriented dialog benchmarks, and that Soloist outperforms existing methods by a large margin in the few-shot fine-tuning settings where only a limited number of task labels are available for fine-tuning.

We hope that Soloist can inspire dialog researchers and developers to comprehensively explore the new paradigm for building task bots based on task-grounded pre-training and fine-tuning via machine teaching, and improving the recipe we present in this paper, namely, formulating task-oriented dialog as a single auto-regressive language model, pre-training a task-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.

References

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

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. Plato: Pre-trained dialogue generation model with discrete latent variable. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 85–96. https://doi.org/10.18653/v1/2020.acl-main.9

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. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22. https://doi.org/10.18653/v1/D19-5602

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026. https://doi.org/10.18653/v1/D18-1547

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. 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 4506–4517. https://doi.org/10.18653/v1/D19-1459

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45. https://doi.org/10.18653/v1/2020.nlp4convai-1.5

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35. https://doi.org/10.1145/3166054.3166058

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *AAAI*, pages 7521–7528. https://doi.org/10.1609/aaai.v34i05.6250

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. https://doi.org/10.18653/v1/P19-1360

Sam Coope, Tyler Farghly, Daniela Gerz, Ivan Vulic, and Matthew Henderson. 2020. Span-convert: Few-shot span extraction for dialog with pretrained conversational representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 107–121. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.11

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 422–428.

Jianfeng Gao, Michel Galley, and Lihong Li. 2019a. Neural approaches to conversational AI. *Foundations and Trends in Information Retrieval*, 13(2–3):127–298. https://doi.org/10.1561/1500000074

Jianfeng Gao, Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Heung-Yeung Shum. 2020. Robust conversational AI with grounded text generation. *CoRR*, abs/2009.03457.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019b. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273.

Rahul Goel, Shachi Paul, and Dilek Hakkani-Tür. 2019. Hyst: A hybrid approach for flexible and accurate dialogue state tracking. *Proceedings of Interspeech 2019*, pages 1458–1462. https://doi.org/10.21437/Interspeech.2019-1863

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44.

Matthew Henderson, Iñigo Casanueva, Nikola Mrksic, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulic. 2020. Convert: Efficient and accurate conversational representations from transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 2161–2174. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.196

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

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, Minlie Huang, Luis Lastras, Jonathan K. Kummerfeld, Walter S. Lasecki, Chiori Hori, Anoop Cherian, Tim K. Marks, Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, and Raghav Gupta. 2019. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582.

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

Hung Le, Richard Socher, and Steven C. H. Hoi. 2020. Non-autoregressive dialog state tracking. *arXiv preprint arXiv:2002.08024*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019a. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483.

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.

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, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020a. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699, Online. 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. 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*.

Wei Li, Wei Shao, Shaoxiong Ji, and Erik Cambria. 2020c. BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis. *arXiv preprint arXiv:2006.00492*.

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*.

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

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*.

Yukun Ma, Khanh Linh Nguyen, Frank Z. Xing, and Erik Cambria. 2020. A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70. `https://doi.org/10.1016/j.inffus.2020.06.011`

Shikib Mehri, Evgeniia Razumovskaia, Tiancheng Zhao, and Maxine Eskenazi. 2019a. Pretraining methods for dialog context representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3836–3845. `https://doi.org/10.18653/v1/P19-1373`

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019b. Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 165–177. `https://doi.org/10.18653/v1/W19-5921`

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking model. *arXiv preprint arXiv:1812.00899*.

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, Chunyuan Li, Zhu Zhang, Chenguang Zhu, Jinchao Li, and Jianfeng Gao. 2020a. RADDLE: An evaluation benchmark and analysis platform for robust task-oriented dialog systems. *CoRR*, abs/2012.14666.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2240. https://doi.org/10.18653/v1/D17-1237

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020b. Few-shot natural language generation for task-oriented dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 172–182, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.17

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.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437. https://doi.org/10.18653/v1/P18-2069

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696. https://doi.org/10.1609/aaai.v34i05.6394

Liliang Ren, Jianmo Ni, and Julian McAuley. 2019. Scalable and accurate dialogue state tracking via hierarchical sequence generation. 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 1876–1885.

Stephen Roller, Y-Lan Boureau, Jason Weston, Antoine Bordes, Emily Dinan, Angela Fan, David Gunning, Da Ju, Margaret Li, Spencer Poff, Pratik Ringshia, Kurt Shuster, Eric Michael Smith, Arthur Szlam, Jack Urbanek, and Mary Williamson. 2020a. Open-domain conversational agents: Current progress, open problems, and future directions. *CoRR*, abs/2006.12442.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 343–349. https://doi.org/10.18653/v1/2020.acl-demos.39

Kurt Shuster, Da Ju, Stephen Roller, Emily Dinan, Y-Lan Boureau, and Jason Weston. 2020. The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 2453–2470. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.222

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.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M. Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449.

Jason D. Williams, Kavosh Asadi Atui, and Geoffrey Zweig. 2017. Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 665–677. https://doi.org/10.18653/v1/P17-1062

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. https://doi.org/10.18653/v1/W17-5511

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-demos.6

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

Chien-Sheng Wu, Steven C. H. Hoi, Richard Socher, and Caiming Xiong. 2020a. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019a. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2019b. 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. *CoRR*, abs/2005.00613.

Haotian Xu, Haiyun Peng, Haoran Xie, Erik Cambria, Liuyang Zhou, and Weiguo Zheng. 2019. End-to-end latent-variable task-oriented dialogue system with exact log-likelihood optimization. *World Wide Web*, pages 1–14.

Steve J. Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of IEEE*, 101(5):1160–1179. https://doi.org/10.1109/JPROC.2012.2225812

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 4970–4977. AAAI Press.

Tom Young, Vlad Pandelea, Soujanya Poria, and Erik Cambria. 2020. Dialogue systems with audio context. *Neurocomputing*, 388:102–109. https://doi.org/10.1016/j.neucom.2019.12.126

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*.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, S. Yu Philip, Richard Socher, and Caiming Xiong. 2020a. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 154–167.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020b. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9604–9611.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020c. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2020.acl-demos.30`

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–10.

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

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467. `https://doi.org/10.18653/v1/P18-1135`

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93. `https://doi.org/10.1162/coli_a_00368`

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *arXiv preprint arXiv:1911.06192*.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 142–149, Online. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2020.acl-demos.19`

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*.