Enhancing Task Bot Engagement with Synthesized Open-Domain Dialog

Anonymous ACL submission

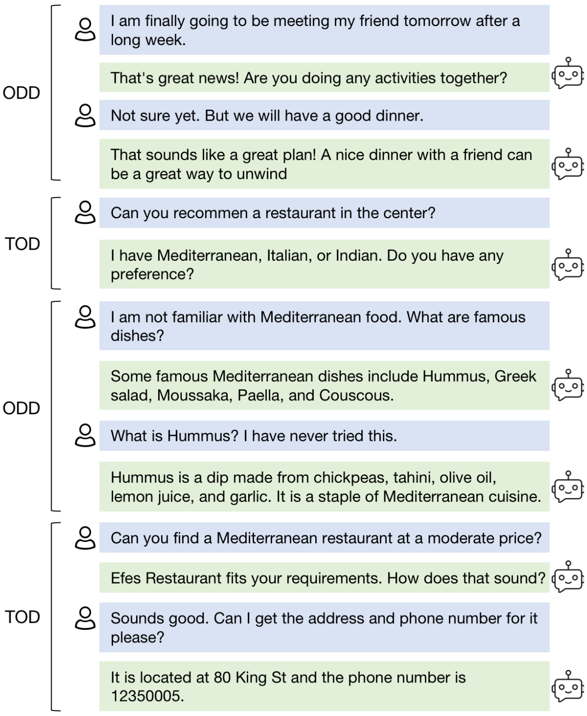
Abstract

The construction of dialog systems for various types of conversations, such as task-oriented dialog (TOD) and open-domain dialog (ODD), has been an active area of research. In order to more closely mimic human-like conversations that often involve the fusion of different dialog modes, it is important to develop systems that can effectively handle both TOD and ODD and access different knowledge sources. In this work, we present a new automatic framework to enrich TODs with synthesized ODDs. We also introduce the PivotBot model, which is capable of handling both TOD and ODD modes and can access different knowledge sources to generate informative responses. Evaluation results indicate the superior ability of the proposed model to switch smoothly between TOD and ODD tasks.

1 Introduction

Task-oriented dialog (TOD) systems and open-domain dialog (ODD) systems are two active areas of Conversational AI study (Gao et al., 2018; Ni et al., 2022). However, most of the existing studies model TOD and ODD systems separately, leading to a gap between the capabilities of these systems and natural human conversations. In real-world conversations, different dialog modes are often fused, as shown in Figure 1. The conversation may start with casual chats and then move towards task-related requests. Along the way, the user may express interest in entities mentioned in the conversation, such as Mediterranean food in the given example, leading to a brief ODD regarding the entity of interest. The user then returns to task completion, keeping the requests in mind while maintaining a casual conversation.

To address the challenge of training dialog models to handle both TOD and ODD modes, previous research has suggested training models on mixture of TOD and ODD datasets (Zhao et al., 2022) or enriching existing TOD datasets by combining chitchat with TOD system responses (Sun et al., 2021; Chen et al., 2022) or adding ODD to the beginning or end of a TOD (Young et al., 2022). However, these approaches have limitations, including limited information in chitchat augmentation and a lack of explicit distinction between dialog modes. Additionally, creating new datasets through human annotation is time-consuming and expensive. While Chiu et al. (2022) have introduced a framework for automatically generating dialogs that transition from ODD to TOD, this method may not be suitable for various mode transitions and cannot simulate informative system utterances with external knowledge.



I am finally going to be meeting my friend tomorrow after a long week.

That's great news! Are you doing any activities together?

Not sure yet. But we will have a good dinner.

That sounds like a great plan! A nice dinner with a friend can be a great way to unwind

Can you recommen a restaurant in the center?

I have Mediterranean, Italian, or Indian. Do you have any preference?

I am not familiar with Mediterranean food. What are famous dishes?

Some famous Mediterranean dishes include Hummus, Greek salad, Moussaka, Paella, and Couscous.

What is Hummus? I have never tried this.

Hummus is a dip made from chickpeas, tahini, olive oil, lemon juice, and garlic. It is a staple of Mediterranean cuisine.

Can you find a Mediterranean restaurant at a moderate price?

Efes Restaurant fits your requirements. How does that sound?

Sounds good. Can I get the address and phone number for it please?

It is located at 80 King St and the phone number is 12350005.

Figure 1: An example dialog that contains multiple transitions between different dialog modes.

In this work, we introduce a framework to automatically enrich TODs with synthesized ODDs. Our approach assumes that users lead conversations with explicit intentions, and that the system's objective is not only to fulfill users' requests but also to generate engaging responses on open-domain topics using external knowledge. We also consider general settings with more flexible dialog mode switches.

This paper makes the following contributions: (i) We propose a general framework for automatically enriching a TOD with knowledge-grounded ODDs and construct the MultiWOZChat dataset using this framework. (ii) We design a unified model, PivotBot, that performs both TOD and ODD tasks by predicting the appropriate dialog mode and accessing knowledge sources for response generation. (iii) We show experimental results that demonstrate the effectiveness of PivotBot in conducting seamless conversations of both types.

2 Proposed Framework

Figure 2 shows the proposed framework for automatically synthesizing one or more knowledge-grounded ODDs to a given TOD. The framework consists of three stages: (1) ODD initialization (2) ODD simulation, and (3) ODD to TOD transition. We define the following notations:

- Denote TOD by $D = \{\boldsymbol{u}_1^{d_1}, \boldsymbol{s}_1^{d_1}, ..., \boldsymbol{u}_{n_1}^{d_1}, \boldsymbol{s}_{n_1}^{d_1}, ..., \boldsymbol{u}_{n_1+n_2}^{d_2}, \boldsymbol{s}_{n_1+n_2}^{d_2}, ..., \boldsymbol{u}_n^{d_N}, \boldsymbol{s}_n^{d_N}\}$,[1] where N is the number of domains in the dialog, $\boldsymbol{u}_i^{d_j}$ and $\boldsymbol{s}_i^{d_j}$ are user and system utterances at turn i in domain j, n_i is the number of turns in domain d_i, and n is the total number of turns in D.
- Denote synthesized ODD by $D' = \{\boldsymbol{u}'_1, \boldsymbol{s}'_1, ..., \boldsymbol{u}'_{n'}, \boldsymbol{s}'_n\}$, where n' is the number of turns in the ODD, $\boldsymbol{u}'_t$ and $\boldsymbol{s}'_t$ represent user and system utterances at turn t, respectively.

Detailed implementation of each module can be found in Appendix A.

2.1 ODD Initialization

Given a TOD D, we initialize the synthesized ODD D' in two ways. If the ODD serves as the preface to the TOD, it is initialized by a randomly sampled user persona. If the ODD is inserted into the TOD as interludes and generated based on the TOD history, we leverage an existing chatbot to simulate a user utterance that can be inserted at a potential location. We then utilize this simulated user utterance to detect whether the user intends to have an open-domain conversation. The off-the-shelf BlenderBot model (Roller et al., 2021) is used as the chatbot in the implementation. These two initialization methods are employed across diverse simulation settings (Section 2.4).

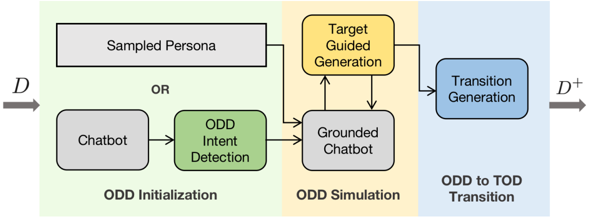


Figure 2: Framework for enriching a given TOD D with ODD. The framework consists of three phases: ODD initialization, ODD simulation, and ODD-to-TOD transition. Rounded and sharp-corner boxes represent models and variables, respectively. The gray color indicates that the model is off-the-shelf. The output is the augmented dialog D^+.

ODD Intent Detection To determine the appropriate time to include an ODD during task completion, we focus on detecting the user's intent to divert the conversation from the task and discuss context-related topics. Given a user utterance $\boldsymbol{u} = \{u_1, ..., u_n\}$, where u_i is the i-th token in the utterance, the ODD intent detection model aims to predict whether the utterance is in a TOD setting or ODD setting. The model is trained by minimizing cross-entropy loss:

$$\mathcal{L}(\hat{I}, I) = \sum_{i=1}^{N} -(\mathbb{1}(\hat{I}_i = I_i)\log(p_{\boldsymbol{\theta}}(I_i)) + (1 - \mathbb{1}(\hat{I}_i = I_i))\log(1 - p_{\boldsymbol{\theta}}(I_i)), \quad (1)$$

where N is number of training examples, $\hat{I}_i$ and I_i are predicted and ground truth intent of the i-th training example, $\boldsymbol{\theta}$ is the parameters of the model.

2.2 ODD Simulation

After initializing the ODD, we use a knowledge-grounded chatbot to mimic a system with access to external knowledge and a target-guided generation model to simulate a user. In practice, we adopt the BlenderBot 2.0 model (Xu et al., 2022; Komeili et al., 2022) and BlenderBot model to simulate system and user utterances, respectively. The ODD is considered complete if a goal $\boldsymbol{g}$ extracted from the subsequent TOD snippet is mentioned in a simulated user utterance.

Target-guided Generation To simulate the human user in the given TOD, we train a target-guided

[1] For settings we do not care about domains in TOD, D can be simplified to $\{\boldsymbol{u}_1, \boldsymbol{s}_1, ..., \boldsymbol{u}_n, \boldsymbol{s}_n\}$.

generation model that is designed to generate utterances based on the dialogue history and mention a preset target at the end of the ODD. The target-guided generation model is expected to generate a user utterance $\boldsymbol{u}'$ at turn $t+1$ based on a pre-determined target $\boldsymbol{g}$ and dialog context $\boldsymbol{c}$ up to turn t.[2] The target is extracted from the initial user utterance of the subsequent TOD part. Given pre-determined ODD goal $\boldsymbol{g} = \{g_1, ..., g_{N_g}\}$ and context $\boldsymbol{c}$, where g_i is the i-th token in the goal, the training objective is defined as

$$\begin{aligned}\mathcal{L}_U &= \log p(\boldsymbol{u}'_{t+1} \mid \boldsymbol{g}, \boldsymbol{c}) \\ &= \sum_{i=1}^{N_u} \log p_{\boldsymbol{\theta}}(u'_{t+1,i} \mid u'_{t+1,<i}, \boldsymbol{g}, \boldsymbol{c}), \qquad (2)\end{aligned}$$

where $\boldsymbol{\theta}$ is the set of trainable parameters in the model, N_u is the target length of predicted user utterance, and $u_{t+1,<i}$ represents tokens before index i of predicted user utterance at turn $t+1$.

2.3 ODD to TOD Transition

Finally, we generate a transition from the simulated ODD to the subsequent TOD to make the dialog more natural. The goal of transition generation is to predict a system utterance that can smoothly connect the last user utterance in the ODD with the initial user utterance in the following TOD. The training objective is

$$\begin{aligned}\mathcal{L}_T &= \log p(\boldsymbol{s}'_t \mid \boldsymbol{u}'_t, \boldsymbol{u}_{t+1}) \\ &= \sum_{i=1}^{N_s} \log p_{\boldsymbol{\theta}}(s'_{t,i} \mid s'_{t,<i}, \boldsymbol{u}'_t, \boldsymbol{u}_{t+1}), \qquad (3)\end{aligned}$$

where $\boldsymbol{u}'_t$ is the last user utterance in generated ODD, $\boldsymbol{u}_{t+1}$ is the first user utterance in the following TOD, $\boldsymbol{s}'_t$ is the transition system utterance.

2.4 Simulation Settings

Inspired by previous research that aims to make dialogs more natural and engaging by adding context to a given dialog (Young et al., 2022) or inserting topic transition turns (Sevegnani et al., 2021), we consider three simulation settings: prepending an ODD to a TOD, inserting an ODD as domain transition turns, and allowing ODDs to occur at any point during task completion. The illustration of three settings is shown in Figure 3.

Setting 1: Prepending ODD to TOD (`INITIAL`) We prepend an ODD to a TOD to generate dialogs with one mode switch from ODD to TOD. We assume that users initiate the conversation by having a quick ODD and then move forward to task completion. Assuming users start with a quick ODD and then move to task completion, we initialize the ODD with a persona from a manually created persona set and use a keyword from the initial user utterance in the subsequent TOD as the goal for the synthesized ODD. Once the target is mentioned in a user utterance, the ODD simulation stops. The transition generation model is then used to connect the synthesized ODD and TOD.

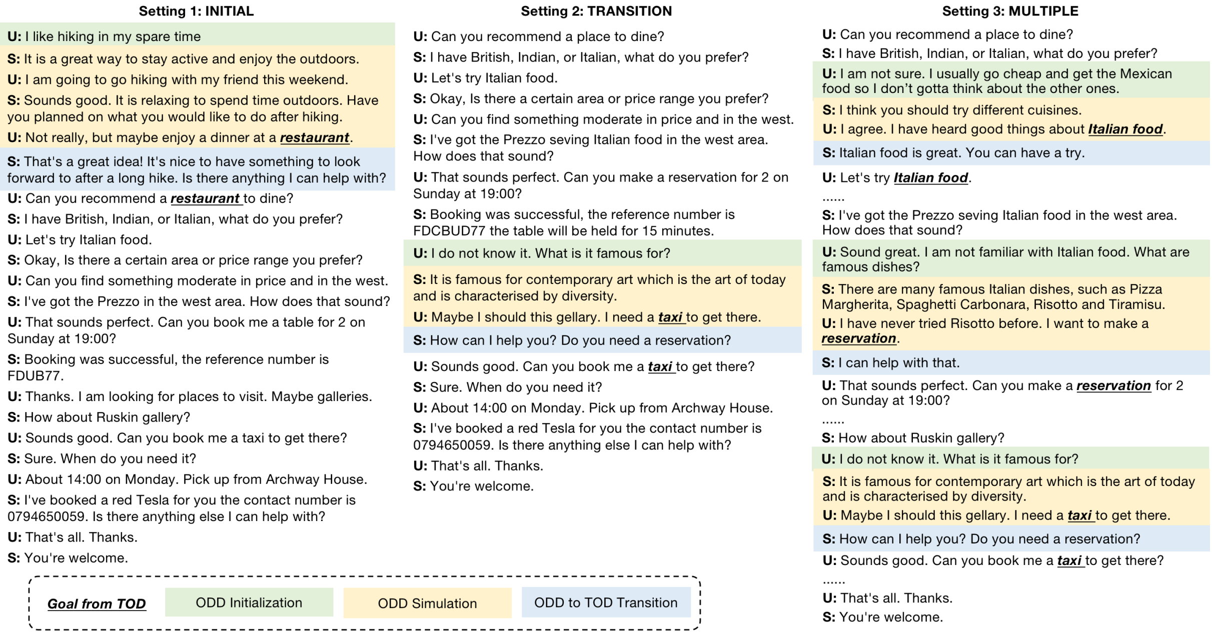
Setting 2: Inserting ODD for Domain Transition in TOD (`TRANSITION`) To make domain transitions in TODs more natural, we insert an ODD as transition turns. Suppose a TOD D contains N domains, where $N \geq 2$. We initialize an ODD using a chatbot after completing the conversation in domain i, and use intent detection model to select an utterance indicating ODD intent. The target of the ODD snippet is extracted from the first user utterance in domain $i+1$. The simulation and transition generation are similar to the previous setting. In the implementation, we only add an ODD to transition from the first domain to the second domain, and use the BlenderBot model for ODD initialization. The final dialogs contain two mode switches.

Setting 3: Inserting Multiple Chitchats to Enrich TODs (`MULTIPLE`) In this more flexible setting, users can initiate conversations with requests and engage in small talk throughout the dialogue. The approach for generating ODDs is the same as in the `TRANSITION` setting, with the difference that we attempt to insert an ODD after each system utterance $\boldsymbol{s}_i$. This allows for multiple mode switches in the final dialogue.

2.5 MultiWOZChat Dataset

We construct MultiWOZChat dataset using the new framework to automatically enrich TODs from the MultiWOZ 2.1 dataset (Eric et al., 2020a). Table 1 summarizes basic statistics of the new dataset, which consists of 500, 198, and 1100 dialogs for the training, validation, and test sets respectively. In the `INITIAL` setting, the average length of a prepended ODD is three turns, and the mean utterance length is 16.18 tokens. In the `TRANSITION` setting, the average length of a transition ODD is shorter than three turns. In the `MULTIPLE` setting, the average number of ODDs inserted into a

[2] We conducted pilot experiments using formulations that included keyword prediction, but found not significant performance improvement. Thus, we decided to use the simplest formulation without turn-level keyword transitions.



Setting 1: INITIAL

U: I like hiking in my spare time
S: It is a great way to stay active and enjoy the outdoors.
U: I am going to go hiking with my friend this weekend.
S: Sounds good. It is relaxing to spend time outdoors. Have you planned on what you would like to do after hiking.
U: Not really, but maybe enjoy a dinner at a ***restaurant***.
S: That's a great idea! It's nice to have something to look forward to after a long hike. Is there anything I can help with?
U: Can you recommend a ***restaurant*** to dine?
S: I have British, Indian, or Italian, what do you prefer?
U: Let's try Italian food.
S: Okay, Is there a certain area or price range you prefer?
U: Can you find something moderate in price and in the west.
S: I've got the Prezzo in the west area. How does that sound?
U: That sounds perfect. Can you book me a table for 2 on Sunday at 19:00?
S: Booking was successful, the reference number is FDUB77.
U: Thanks. I am looking for places to visit. Maybe galleries.
S: How about Ruskin gallery?
U: Sounds good. Can you book me a taxi to get there?
S: Sure. When do you need it?
U: About 14:00 on Monday. Pick up from Archway House.
S: I've booked a red Tesla for you the contact number is 0794650059. Is there anything else I can help with?
U: That's all. Thanks.
S: You're welcome.

Setting 2: TRANSITION

U: Can you recommend a place to dine?
S: I have British, Indian, or Italian, what do you prefer?
U: Let's try Italian food.
S: Okay, Is there a certain area or price range you prefer?
U: Can you find something moderate in price and in the west.
S: I've got the Prezzo seving Italian food in the west area. How does that sound?
U: That sounds perfect. Can you make a reservation for 2 on Sunday at 19:00?
S: Booking was successful, the reference number is FDCBUD77 the table will be held for 15 minutes.
U: I do not know it. What is it famous for?
S: It is famous for contemporary art which is the art of today and is characterised by diversity.
U: Maybe I should this gellary. I need a ***taxi*** to get there.
S: How can I help you? Do you need a reservation?
U: Sounds good. Can you book me a ***taxi*** to get there?
S: Sure. When do you need it?
U: About 14:00 on Monday. Pick up from Archway House.
S: I've booked a red Tesla for you the contact number is 0794650059. Is there anything else I can help with?
U: That's all. Thanks.
S: You're welcome.

Setting 3: MULTIPLE

U: Can you recommend a place to dine?
S: I have British, Indian, or Italian, what do you prefer?
U: I am not sure. I usually go cheap and get the Mexican food so I don't gotta think about the other ones.
S: I think you should try different cuisines.
U: I agree. I have heard good things about ***Italian food***.
S: Italian food is great. You can have a try.
U: Let's try ***Italian food***.
......
S: I've got the Prezzo seving Italian food in the west area. How does that sound?
U: Sound great. I am not familiar with Italian food. What are famous dishes?
S: There are many famous Italian dishes, such as Pizza Margherita, Spaghetti Carbonara, Risotto and Tiramisu.
U: I have never tried Risotto before. I want to make a ***reservation***.
S: I can help with that.
U: That sounds perfect. Can you make a ***reservation*** for 2 on Sunday at 19:00?
......
S: How about Ruskin gallery?
U: I do not know it. What is it famous for?
S: It is famous for contemporary art which is the art of today and is characterised by diversity.
U: Maybe I should this gellary. I need a ***taxi*** to get there.
S: How can I help you? Do you need a reservation?
U: Sounds good. Can you book me a ***taxi*** to get there?
......
U: That's all. Thanks.
S: You're welcome.

Figure 3: Illustration of three simulation settings. Given a TOD between a user (U) and a system bot (S), we consider three settings to synthesize ODD(s) to the TOD.

TOD is four, and each ODD snippet has an average length of two turns. In the TRANSITION and MULTIPLE settings, the ODD durations are shorter, as they occur during task completion, and we do not want the conversation to be distracted from the task completion.

Setting	Split	Avg. mode switch	Total ODD turn	Total TOD turn	Avg. ODD turn	Avg. TOD turn	Avg. ODD length	Avg. TOD length
INITIAL	Train	1	1524	4086	3.05	8.17	16.18	18.07
	Dev		565	1599	2.85	8.08	15.90	18.30
	Test		3248	9031	2.95	8.21	15.99	18.17
TRANSITION	Train	2	1301	4086	2.60	8.17	18.22	18.07
	Dev		510	1599	2.58	8.08	18.26	18.30
	Test		2923	9031	2.66	8.21	18.21	18.17
MULTIPLE	Train	4.96	4356	4086	8.71	8.17	17.80	18.07
	Dev	4.90	1599	1599	8.47	8.08	17.61	18.30
	Test	5.11	9995	9031	9.87	8.21	17.82	18.17

Table 1: Statistics of simulated dialogs in different settings. Training/validation/test set contains 500/198/1100 dialogs.

3 Methodology

3.1 Problem Formulation

The full task consists of three processes: state prediction, knowledge retrieval, and knowledge-grounded response generation. We use off-the-shelf models for knowledge retrieval, which can be a database lookup or a search engine,[3] and do not consider it as a subtask. The full task is then divided into two subtasks: state prediction and knowledge-grounded response generation. In the t-th turn of a dialog, the model predicts the state $\boldsymbol{s}$ based on the dialog history $\boldsymbol{h} = \{\boldsymbol{u}_{t-k}, \boldsymbol{r}_{t-k}, ..., \boldsymbol{u}_t\}$, where k is the size of the history window, $\boldsymbol{u}_i$ and $\boldsymbol{r}_i$ represent the user utterance and system response at the i-th turn, respectively. The state indicates the appropriate dialog mode and the query to obtain knowledge $\boldsymbol{k}$. The model then generates a response $\boldsymbol{r}$ based on the dialog history, predicted state and knowledge.

3.2 PivotBot

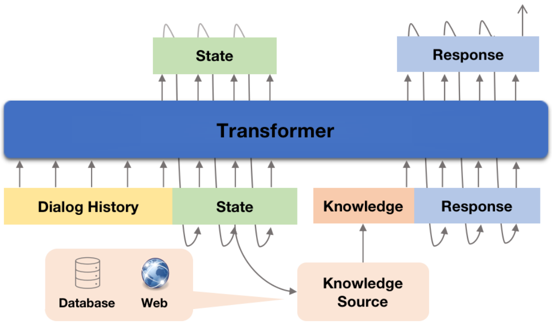


Figure 4: Overall architecture of the PivotBot model

We construct a unified model, PivotBot, as shown in Figure 4. PivotBot first predicts a state indicating the appropriate dialog mode and query to obtain knowledge based on the dialog history. The knowledge acquisition is completed by off-the-shelf models based on the prediction. Finally, the model per-

[3] In the implementation, we adopted the Bing search engine.

forms grounded generation to generate a response.

State Prediction State $\boldsymbol{s}$ tracks a user's goal throughout a dialog. In particular, a state $\boldsymbol{s}$ is in the form `m:q`, where `m` represents the dialog mode, and `q` stands for the query to acquire knowledge from a knowledge source. We consider two dialog modes: TOD modeling and knowledge-grounded ODD. If the model predicts performing TOD modeling, a database state is obtained from the predefined database using the predicted belief state. If the state indicates the dialog mode is ODD, external knowledge can be retrieved from the Web using the predicted search query. If the search query is empty, it implies that external knowledge is not needed for response generation, and the retrieved knowledge is also empty. Given dialog history $\boldsymbol{h}$, the training objective of state prediction can be formulated as

$$\mathcal{L}_S = \log p(\boldsymbol{s} \mid \boldsymbol{h}) = \sum_{i=1}^{N_t} \log p_{\boldsymbol{\theta}}(s_i \mid s_{<i}, \boldsymbol{h}), \quad (4)$$

where $\boldsymbol{\theta}a$ represents trainable parameters in the model, N_t is the target length of predicted state sequence, and $s_{<i}$ denotes tokens before index i.

Grounded Generation System response $\boldsymbol{r} = \{r_1, r_2, ..., r_{N_r}\}$ with length N_r is generated grounded on dialog history $\boldsymbol{h}$, predicted state $\boldsymbol{s}$ and retrieved knowledge $\boldsymbol{k}$. The training objective is defined as

$$\begin{aligned}\mathcal{L}_R &= \log p(\boldsymbol{r} \mid \boldsymbol{h}, \boldsymbol{s}, \boldsymbol{k}) \\ &= \sum_{i=1}^{N_r} \log p_{\boldsymbol{\theta}}(r_i \mid r_{<i}, \boldsymbol{h}, \boldsymbol{s}, \boldsymbol{k}). \quad (5)\end{aligned}$$

Training Objective of Full Task A training example consists of four components: dialog history $\boldsymbol{h}$, state $\boldsymbol{s}$, retrieved knowledge $\boldsymbol{k}$, and (delexicalized) dialog response $\boldsymbol{r}$. The overall training objective is

$$\mathcal{L}_{\boldsymbol{\theta}}(\mathcal{D}) = \sum_{i=1}^{N_D} (\mathcal{L}_S(\boldsymbol{x_i}) + \mathcal{L}_R(\boldsymbol{x_i})), \quad (6)$$

where $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N_D}$ is the training dataset containing N_D training examples.

4 Experiments

4.1 Experimental Setup

We train models using 100, 200, and 500 dialogs and evaluate them on the entire test set. Our primary focus is evaluating the models trained in the few-shot setting, as this approach more closely reflects real-world scenarios.

Baselines Previous studies either do not distinguish different dialog modes or only focus on social chats without external knowledge. However, our task requires models to switch between ODD and TOD modes and choose the appropriate knowledge source. To ensure a fair comparison, we train two baselines for our problem setting instead of comparing with models designed for different settings.

- TaskBot serves as a baseline and is only capable of performing TOD with access to a database, which is trained solely on TOD turns in the MultiWOZChat dataset.
- ChatBot is a baseline model that can only perform ODD, which is trained on ODD turns in the MultiWOZChat dataset.

The baselines and PivotBot are implemented using HuggingFace T5-base and GODEL(Peng et al., 2022). Further details of implementations can be found in Appendix A.

Implementation The models are implemented using HuggingFace T5-base and GODEL (Peng et al., 2022). Training examples are truncated or padded to a length of 512. To ensure input strings contain dialog history and retrieved knowledge, the history is truncated on the left with a max length of 256 and consists of five utterances with a history window size of 2. AdamW optimizer (Loshchilov and Hutter, 2019) with a constant learning rate of 0.001 is used for training with a mini-batch size of 8 on a Tesla P100 for up to 15 epochs or until no validation loss decrease is observed. Each setting is evaluated eight times with random seeds.

Evaluation Metrics We evaluate the performance of the models in three settings: (1) standard TOD completion (Budzianowski et al., 2018; Eric et al., 2020b; Nekvinda and Dušek, 2021), (2) ODD response generation, and (3) the full task involving both TOD and ODD.

We evaluate TOD completion using four metrics: (1) `BLEU` (Papineni et al., 2002) measures the fluency of the generated responses; (2) `Success` indicates if all requested attributes are answered; (3) `Inform` measures whether the correct entity is provided (e.g., restaurant address); (4) `Combine` score is an overall measure calculated as as (`Inform`+`Success`) × 0.5 + `BLEU`.

We evaluate ODD using three metrics: (1) `Accuracy` measures the model's ability to predict the correct dialog mode, which can be calculated by comparing the predicted dialog mode with the ground truth mode; (2) Success Rate assesses the model's performance in state prediction at the dialog level, and measures the model's potential for success in the ODD task. It can be calculated by dividing the number of dialogs in which the model correctly predicts the dialog mode for all ODD turns by the total number of dialogs with ODD turns; (3) BLEU measures the naturalness of the model's responses.

We evaluate the model's performance on the full task using BLEU, Inform, Success, and Combine score. BLEU score is computed for all responses in the dialogs, while Inform and Success metrics are limited to dialogs that succeed in both TOD modeling and ODD tasks. The potential success of the ODD task is used as an indicator, and Inform and Success are computed for dialogs where the dialog mode predictions for all ODD turns are accurate.

Human Evaluation Setup We conducted two-phase human evaluation. In the first stage, we hired Amazon Mechanical Turk workers to interact with three models: TaskBot with T5 as the backbone (T5-TaskBot), PivotBot with T5 as the backbone (T5-PivotBot), and PivotBot with GODEL as the backbone (GODEL-PivotBot). The workers were provided with information-seeking goals from the MultiWOZ 2.1 dataset and were allowed to chat freely with the models to complete the goals. After each conversation, workers rated the appropriateness and engagingness of the model's responses on a 5-point Likert scale and indicated if all requests were completed.

In the second stage, we conducted a static evaluation of the dialogs collected in the previous phase. Each worker was presented with a pair of dialogs, one produced by T5-TaskBot and the other by T5-PivotBot, or one produced by T5-PivotBot and the other by GODEL-PivotBot, and was asked to choose the better dialog based on the system performance. Then workers rated the appropriateness and engagingness of each system's utterances in the dialogs using a 5-point Likert scale.

4.2 Automatic Evaluation Results

We present the results for models trained in the few-shot setting using 100 training dialogs with the GODEL backbone.[4] For the full task evaluation, we only report the combined score. The evaluation results using 200/500 training dialogs are in Appendix B.

INITIAL Setting Evaluation Table 2 shows the evaluation results in the INITIAL setting. PivotBot significantly outperforms the baseline models in the full task evaluation, demonstrating the importance of incorporating different dialog modes. PivotBot also slightly outperforms TaskBot in the TOD task in terms of the Combined score. This suggests that the ability to handle both TOD and ODD tasks with appropriate dialog modes and knowledge sources is critical for PivotBot to excel in the full task. While ChatBot cannot provide requested attributes or entities, it performs better than other models in predicting the dialog mode in the ODD evaluation setting. Though PivotBot cannot beat ChatBot in the ODD evaluation, it achieves comparable results while generating more fluent responses and simultaneously handling task completion.

TRANSITION Setting Evaluation Table 3 contains evaluation results in the TRANSITION setting. PivotBot performs significantly better than baselines in the full task. TaskBot slightly outperforms PivotBot in the TOD modeling task. ChatBot still achieves the best performance in the ODD task. Though PivotBot cannot perform better than baselines in single task evaluation, it can obtain comparable results with the specialist baselines. The gap between ChatBot and PivotBot in success rate is more obvious, indicating that it is more challenging for the model to learn both dialog modes simultaneously and accurately predict the mode when the mode switches in dialogs become more complex.

MULTIPLE Setting Evaluation The evaluation results in the MULTIPLE setting are presented in Table 4. In the full task evaluation, PivotBot remains the best-performing model. The performance of TaskBot and PivotBot is comparable in the TOD task. However, in the ODD task evaluation, while PivotBot's turn-level prediction accuracy does not significantly decrease, the model is more likely to fail in the ODD task at the dialog level due to the increased number of ODD turns and more complex mode switches within a dialog.

Cross-Setting Evaluation Table 5 contains the Combined scores of PivotBot trained in each setting evaluated in all three settings, allowing us to examine the relationships among the different settings.

[4] We also evaluated the models using the T5-base backbone and found that models with the GODEL backbone outperform those based on T5-base, with statistically significant performance differences.

Model	Full Task	TOD Evaluation				ODD Evaluation		
	Combined	BLEU	Success	Inform	Combined	Accuracy	Success Rate	BLEU
TaskBot	12.26(0.43)	**15.00**(0.57)	37.43(4.02)	52.61(4.26)	60.01(4.41)	0.00(0.00)	0.00(0.00)	1.33(0.22)
ChatBot	7.98(0.25)	0.97(0.16)	0.60(0.00)	10.70(0.00)	6.62(0.16)	**99.93**(0.05)	**99.79**(0.16)	6.43(0.51)
PivotBot	**58.06**(5.15)	14.90(0.58)	**38.66**(5.22)	**53.55**(5.62)	**61.01**(5.48)	98.90(0.45)	97.35(0.98)	**6.82**(0.41)

Table 2: End-to-end evaluation in the `INITIAL` setting. Mean values and standard deviations (in parentheses) are reported

Model	Full Task	TOD Evaluation				ODD Evaluation		
	Combined	BLEU	Success	Inform	Combined	Accuracy	Success Rate	BLEU
TaskBot	12.37(0.36)	**15.02**(0.46)	**35.43**(4.14)	**50.56**(5.35)	**58.02**(5.03)	0.00(0.00)	0.00(0.00)	1.22(0.12)
ChatBot	7.88(0.13)	1.22(0.16)	0.60(0.00)	10.70(0.00)	6.87(0.16)	**100.00**(0.01)	**99.99**(0.03)	**5.35**(0.18)
PivotBot	**49.58**(7.13)	14.92(0.64)	33.49(6.13)	47.06(8.21)	55.19(7.56)	96.17(0.64)	90.00(1.72)	4.97(0.28)

Table 3: End-to-end evaluation in the `TRANSITION` setting. Mean values and standard deviations (in parentheses) are reported

Model	Full Task	TOD Evaluation				ODD Evaluation		
	Combined	BLEU	Success	Inform	Combined	Accuracy	Success Rate	BLEU
TaskBot	8.10(0.27)	**14.79**(0.48)	34.74(5.29)	**50.16**(6.69)	57.24(6.11)	0.00(0.00)	0.00(0.00)	0.93(0.07)
ChatBot	8.76(0.29)	1.14(0.09)	0.60(0.00)	10.70(0.00)	6.79(0.09)	**100.00**(0.00)	**100.00**(0.00)	**5.05**(0.48)
PivotBot	**42.43**(3.23)	14.77(0.65)	**35.75**(3.13)	49.76(4.32)	**57.52**(3.89)	96.66(0.28)	74.39(2.15)	4.97(0.42)

Table 4: End-to-end evaluation in the `MULTIPLE` setting. Mean values and standard deviations (in parentheses) are reported

Training Setting	Evaluation Setting		
	`INITIAL`	`TRANSITION`	`MULTIPLE`
`INITIAL`	**58.06**(5.15)	12.12(0.42)	8.54(0.38)
`TRANSITION`	22.80(10.99)	49.58(7.13)	22.26(2.23)
`MULTIPLE`	49.69(6.20)	**51.91**(4.28)	**42.43**(3.23)

Table 5: End-to-end cross setting evaluation results. Mean values and standard deviations (in parentheses) of the `Combined` score for PivotBot models trained in different settings are reported.

The model trained in the `INITIAL` setting performs best in that same evaluation setting. The model trained in the `TRANSITION` setting obtains comparable performance with the model in the `MULTIPLE` setting in the `TRANSITION` evaluation setting but struggles in the other two evaluation settings. The model trained in the `MULTIPLE` setting obtains the highest `Combined` scores in the other two evaluation settings, indicating its ability to generalize well to different settings.

4.3 Human Evaluation Results

In the first phase of human evaluation, we collected 200 dialogs for each model. To make the evaluation task more manageable for the workers, we only sampled information-seeking goals involving a single domain, which may also have made it easier for the models to fulfill all users' requests. The results are shown in Table 6. Consistent with the automatic evaluation, both TaskBot and PivotBot can complete users' requests, but PivotBot shows greater ability in generating appropriate and engaging responses. The GODEL backbone further enhances PivotBot's engagingness.

	T5-TaskBot	T5-PivotBot	GODEL-PivotBot
Success	0.99(0.10)	1.00(0.07)	**1.00**(0.00)
Appropriateness	4.10(1.11)	4.27(1.00)	**4.35**(0.01)
Engagingness	4.09(1.13)	4.31(0.88)	**4.44**(0.71)

Table 6: Results of the first phrase of human evaluation. Mean values and standard deviations (in parentheses) are reported. `Success` is measured in binary scale, while `Appropriate` and `Engagingness` are measured on a 5-point Likert scale.

In the second phase, we conducted pairwise comparisons of the models' performance and present the results in Table 7. The relationships between the models are consistent with the first phase. Notably, there are fewer ties in the pairwise comparisons of dialog-level performance than in the evaluations of appropriateness and engagingness. This could be because pairwise comparisons provide evaluators with a clearer choice, while evaluating appropriateness and engagingness may be more subjective. Other factors, such as the length and quality of the dialogues, may also influence the evaluators' judgments in pairwise comparisons of overall performance, while a model's appropriate-

	T5-PivotBot vs. T5-TaskBot		
	Win	Tie	Loss
Overall	51.52*	17.68	30.81*
Appropriateness	50.51**	36.87	12.63**
Engagingness	50.51**	30.30	19.19**
	GODEL-PivotBot vs. T5-PivotBot		
	Win	Tie	Loss
Overall	44.72	23.62	31.66
Appropriateness	43.94**	43.22	13.07**
Engagingness	53.77**	34.17	12.06**

Table 7: Results of the second phrase of human evaluation. "Overall" stands for the dialog-level evaluation results. "Win" (or "Loss") refers to the percentage of cases where T5-PivotBot (in the upper section) and GODEL-PivotBot (in the lower section) wins (or loses). * / ** denote p-values of less than 0.05 / 0.01.

ness and engagingness may be judged solely on its own merits.

5 Related Work

Dialog Systems for Fused Task of ODD and TOD Several previous works have addressed the challenge of constructing dialog systems that can handle multiple dialog modes. Some work focused on constructing systems that independently model different dialog skills or training dialog models on mixture of TOD and ODD datasets to enable it to switch between conversation styles (Madotto et al., 2020; Lin et al., 2021) . Other approaches have involved constructing new datasets for mixed settings, by adding ODD utterances to system utterances in TODs (Zhao et al., 2017; Sun et al., 2021; Chen et al., 2022), or enriching TODs with human-annotated ODD snippets to include one mode transition within an augmented dialog (Young et al., 2022). To reduce the need for human involvement in dataset construction, Chiu et al. (2022) proposed a framework for automatically generating dialogs that transition from ODD to TOD with a simulated user and simulated salesperson, assuming that users do not explicitly state their intentions and that the system must detect and respond to these intentions.

Target-guided Generation for ODDs Some previous work (Xing et al., 2017; Lian et al., 2019; Ling et al., 2021) focused on guiding the conversation generation in a short-term, while others studied the multi-turn target-guided process of conversations. Tang et al. (2019) proposed the task of target-guided open-domain conversation where the model leads the conversation from a random initial topic to a target word. Qin et al. (2020) improved the previous work by constraining candidate keywords and augmenting responses with predicted keywords. Kishinami et al. (2022) modified the previous task setting and focused on evaluating the ability of a model to plan a target-oriented conversation. Researchers also considered actively leading a conversation to a target by incorporating knowledge graphs (Wu et al., 2019; Xu et al., 2020; Zhong et al., 2021).

6 Conclusion and Future Work

This paper introduces an easily-implemented and generalizable framework for enriching a TOD with ODDs in different settings. A unified model, PivotBot, with both TOD and ODD dialog modes is designed. The evaluation results demonstrate the effectiveness of the proposed model and the significance of integrating multiple dialog modes for generating appropriate and engaging responses.

Future work on the data simulation can involve integrating various external knowledge, such as knowledge graphs and personality traits, and exploring alternative guided generation methods to improve the consistency and controllability of the generated ODDs. The knowledge retrieval function used in the PivotBot model could be further optimized by training a more efficient retrieval and selection model. Additionally, creating a system with comprehensive capabilities, including recommendation and personalization, would enhance its suitability for real-world use.

7 Ethical Considerations

In creating the MultiWOZChat dataset, we used BlenderBot models with safety controls to simulate ODDs and MultiWOZ 2.1 for TODs to exclude harmful dialogs. However, existing chatbots may still use unsafe language, and pre-trained language models may have been exposed to text containing social bias or toxicity, potentially leading to offensive responses from the PivotBot model. Additionally, off-the-shelf chatbots may generate hallucinatory content, resulting in unreliable responses from the PivotBot model. Future work should focus on exploring more efficient safety measures and improving response accuracy.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 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, Brussels, Belgium. Association for Computational Linguistics.

Zhiyu Chen, Bing Liu, Seungwhan Moon, Chinnadhurai Sankar, Paul Crook, and William Yang Wang. 2022. KETOD: Knowledge-enriched task-oriented dialogue. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2581–2593, Seattle, United States. Association for Computational Linguistics.

Ssu Chiu, Maolin Li, Yen-Ting Lin, and Yun-Nung Chen. 2022. SalesBot: Transitioning from chit-chat to task-oriented dialogues. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6143–6158, Dublin, Ireland. 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. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander H. Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur D. Szlam, Iulian Serban, Ryan Lowe, Shrimai Prabhumoye, Alan W. Black, Alexander I. Rudnicky, Jason Williams, Joelle Pineau, Mikhail S. Burtsev, and Jason Weston. 2019a. The second conversational intelligence challenge (ConvAI2). *ArXiv*, abs/1902.00098.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019b. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020a. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020b. 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, Marseille, France. European Language Resources Association.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational AI. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 2–7, Melbourne, Australia. Association for Computational Linguistics.

Yosuke Kishinami, Reina Akama, Shiki Sato, Ryoko Tokuhisa, Jun Suzuki, and Kentaro Inui. 2022. Target-guided open-domain conversation planning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 660–668.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478.

Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to select knowledge for response generation in dialog systems. *arXiv preprint arXiv:1902.04911*.

Zhaojiang Lin, Andrea Madotto, Yejin Bang, and Pascale Fung. 2021. The Adapter-Bot: All-in-one controllable conversational model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 16081–16083.

Yanxiang Ling, Fei Cai, Xuejun Hu, Jun Liu, Wanyu Chen, and Honghui Chen. 2021. Context-controlled topic-aware neural response generation for open-domain dialog systems. *Inf. Process. Manag.*, 58:102392.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, Jamin Shin, and Pascale Fung. 2020. Attention over parameters for dialogue systems. *arXiv preprint arXiv:2001.01871*.

Tomáš Nekvinda and Ondřej Dušek. 2021. Shades of BLEU, flavours of success: The case of MultiWOZ. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 34–46, Online. Association for Computational Linguistics.

Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, and Erik Cambria. 2022. Recent advances in deep learning based dialogue systems: A systematic survey. *Artificial intelligence review*, pages 1–101.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Baolin Peng, Michel Galley, Pengcheng He, Chris Brockett, Lars Liden, Elnaz Nouri, Zhou Yu, Bill Dolan, and Jianfeng Gao. 2022. GODEL: Large-scale pre-training for goal-directed dialog. arXiv.

Jinghui Qin, Zheng Ye, Jianheng Tang, and Xiaodan Liang. 2020. Dynamic knowledge routing network for target-guided open-domain conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8657–8664.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Karin Sevegnani, David M. Howcroft, Ioannis Konstas, and Verena Rieser. 2021. OTTers: One-turn topic transitions for open-domain dialogue. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2492–2504, Online. Association for Computational Linguistics.

Kai Sun, Seungwhan Moon, Paul Crook, Stephen Roller, Becka Silvert, Bing Liu, Zhiguang Wang, Honglei Liu, Eunjoon Cho, and Claire Cardie. 2021. Adding chit-chat to enhance task-oriented dialogues. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1570–1583, Online. Association for Computational Linguistics.

Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric Xing, and Zhiting Hu. 2019. Target-guided open-domain conversation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5624–5634, Florence, Italy. Association for Computational Linguistics.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804, Florence, Italy. Association for Computational Linguistics.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Jing Xu, Arthur Szlam, and Jason Weston. 2022. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5180–5197, Dublin, Ireland. Association for Computational Linguistics.

Jun Xu, Haifeng Wang, Zhengyu Niu, Hua Wu, and Wanxiang Che. 2020. Knowledge graph grounded goal planning for open-domain conversation generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9338–9345.

Tom Young, Frank Xing, Vlad Pandelea, Jinjie Ni, and Erik Cambria. 2022. Fusing task-oriented and open-domain dialogues in conversational agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11622–11629.

Tiancheng Zhao, Allen Lu, Kyusong Lee, and Maxine Eskenazi. 2017. Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability. *arXiv preprint arXiv:1706.08476*.

Xinyan Zhao, Bin He, Yasheng Wang, Yitong Li, Fei Mi, Yajiao Liu, Xin Jiang, Qun Liu, and Huanhuan Chen. 2022. UniDS: A unified dialogue system for chit-chat and task-oriented dialogues. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 13–22, Dublin, Ireland. Association for Computational Linguistics.

Peixiang Zhong, Yong Liu, Hao Wang, and Chunyan Miao. 2021. Keyword-guided neural conversational model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14568–14576.

A Proposed framework

A.1 ODD Intent Detection

The detection model is implemented using HuggingFace BERT-base (Devlin et al., 2019) model and is trained on a combination of four datasets: MultiWOZ 2.1, ConvAI2(Dinan et al., 2019a), FusedChat (with pretended ODDs), and Wizard of Wikipedia (WoW) (Dinan et al., 2019b), with equal numbers of TOD and ODD turns for balance.

A.2 Target-guided Generation

MultiWOZ target candidate We consider values of 8 slots in the MultiWOZ 2.1 dataset as potential targets. These slots are `name`, `area`, `pricerange`, `type`, `departure`, `destination`, `department`, and `day`. The values can be represented as nouns, adjectives, or phrases.

Training We train the distilled BlenderBot on three datasets (FusedChat, WoW, ConvAI2) to generate diverse user utterances. We use a keyword extraction method (Tang et al., 2019) to set target for ODDs in WoW and ConvAI2, and extract a target from the initial user utterance of the TOD part for the prepended ODDs from FusedChat.

Inference We use the trained target-guided generation model to simulate the user in ODD and extract the goal g from the given TOD using the set of candidate targets from MultiWOZ 2.1.

A.3 Transition Generation

The implementation is based on the HuggingFace T5-base (Raffel et al., 2020) model. The training datasets are the same as Sec.A.2. A training example consists of user utterances at turn t and $t + 1$ and system response at turn t.

B Automatic Evaluation Results

`INITIAL` Setting Evaluation Table 8 and 12 show evaluation results in the `TRANSITION` setting. As the number of training dialogs increases, all models show improvement. ChatBot and PivotBot models improve in generating fluent ODD responses, while TaskBot focuses more on TOD modeling and fails to respond appropriately to ODDs.

`TRANSITION` Setting Evaluation Table 9 and 13 contain evaluation results in the `TRANSITION` setting. Performance improvements can be observed for all models with an increase in training dialogs. In addition, the response quality improves for both ChatBot and PivotBot, and PivotBot shows better ability to choose appropriate dialog modes.

# Training dialogs	Model	Full Task Evaluation BLEU	 Success	 Inform	 Combined
200	TaskBot	13.34(0.22)**	0.00(0.00)	0.00(0.00)	13.34(0.22)**
	ChatBot	2.56(0.14)**	0.60(0.00)	10.71(0.03)	8.22(0.15)**
	PivotBot	**14.53**(0.18)**	**40.66**(1.81)**	**52.74**(2.70)**	**61.23**(2.24)**
500	TaskBot	14.41(0.25)**	0.00(0.00)	0.00(0.00)	14.41(0.25)**
	ChatBot	2.92(0.09)**	0.60(0.00)	10.70(0.00)	8.57(0.09)**
	PivotBot	**15.76**(0.20)**	**42.45**(2.33)*	**53.79**(3.26)*	**63.88**(2.62)*

Table 8: End-to-end full task evaluation using GODEL as backbone in `INITIAL` setting. Statistically significant differences exist between GODEL-based and T5-based models (*p<0.05, **p<0.01).

# Training dialogs	Model	Full Task Evaluation BLEU	 Success	 Inform	 Combined
200	TaskBot	13.49(0.15)**	0.00(0.00)	0.00(0.00)	13.49(0.15)**
	ChatBot	2.42(0.14)**	0.60(0.00)	10.70(0.00)	8.08(0.14)**
	PivotBot	**14.27**(0.31)**	**32.75**(5.67)	**42.54**(7.26)	**51.92**(6.53)
500	TaskBot	14.49(0.26)**	0.00(0.00)	0.00(0.00)	14.49(0.26)**
	ChatBot	2.63(0.06)**	0.60(0.00)	10.70(0.00)	8.28(0.06)**
	PivotBot	**15.49**(0.37)**	**41.39**(1.73)**	**51.65**(2.30)*	**62.01**(2.11)**

Table 9: End-to-end full task evaluation using GODEL as backbone in `TRANSITION` setting. Statistically significant differences exist between GODEL-based and T5-based models (*p<0.05, **p<0.01).

`MULTIPLE` Setting Evaluation The evaluation results in the `MULTIPLE` setting, shown in Table 10 and 14, are consistent with the results in the previous settings. The PivotBot model improves its ability to make more accurate predictions with an increase in the number of training dialogs.

Cross-Setting Evaluation Table 11 and Table 15 present the cross-setting evaluation results. With more training dialogs, models show performance improvement in all evaluation settings. The model trained in the `MULTIPLE` setting demonstrates the ability to generalize well and obtains the highest (or comparable) scores in all settings.

# Training dialogs	Model	Full Task Evaluation BLEU	 Success	 Inform	 Combined
200	TaskBot	8.90(0.35)**	0.00(0.00)	0.00(0.00)	8.90(0.35)**
	ChatBot	3.72(0.22)**	0.60(0.00)	10.70(0.00)	9.37(0.22)**
	PivotBot	**11.43**(0.18)**	**29.10**(4.51)	**38.54**(4.83)	**45.25**(4.64)
500	TaskBot	9.8(0.18)**	0.00(0.00)	0.00(0.00)	9.80(0.18)**
	ChatBot	4.19(0.08)**	0.60(0.00)	10.70(0.00)	9.84(0.08)**
	PivotBot	**12.66**(0.12)**	**37.54**(4.09)**	**47.96**(5.43)*	**55.40**(4.65)**

Table 10: End-to-end full task evaluation using GODEL as backbone in `MULTIPLE` setting. Statistically significant differences exist between GODEL-based and T5-based models. (*p<0.05, **p<0.01).

Evaluation setting	Training setting	# Training dialogs	Full Task Evaluation BLEU	 Success	 Inform	 Combined
`INITIAL`	`INITIAL`	500	**15.76**(0.20)	**42.45**(2.33)	**53.79**(3.26)	**63.88**(2.62)
	`TRANSITION`		15.24 (0.24)	31.65 (8.13)	39.93 (10.41)	51.03 (9.41)
	`MULTIPLE`		15.15 (0.20)	35.84 (4.08)	45.73 (5.64)	55.93 (4.78)
`TRANSITION`	`INITIAL`	500	14.17 (0.33)	1.52 (1.24)	2.11 (1.72)	15.99 (1.62)
	`TRANSITION`		**15.49** (0.37)	**41.39** (1.73)	**51.65** (2.30)	**62.01** (2.11)
	`MULTIPLE`		15.23 (0.18)	38.48 (4.11)	49.03 (5.57)	58.98 (4.74)
`MULTIPLE`	`INITIAL`	500	10.18 (0.20)	0.09(0.10)	0.19(0.20)	10.33(0.30)
	`TRANSITION`		11.82 (0.24)	20.86 (1.43)	27.28 (1.96)	35.89 (1.81)
	`MULTIPLE`		**12.66** (0.12)	**37.54** (4.09)	**47.96** (5.43)	**55.40**(4.65)

Table 11: End-to-end cross evaluation of the full task

# Training dialogs	Model	TOD Evaluation				ODD Evaluation		
		BLEU	Success	Inform	Combined	Accuracy	Success Rate	BLEU
200	TaskBot	16.36(0.32)**	36.93(5.46)*	48.19(7.15)	58.92(6.19)	0.00(0.00)	0.00(0.00)	1.25(0.24)**
	ChatBot	0.91(0.12)**	0.60(0.00)	10.71(0.00)	6.56(0.12)**	**99.97**(0.05)	**99.90**(0.15)	7.57(0.41)**
	PivotBot	**16.37**(0.25)**	**41.29**(1.69)**	**53.61**(2.59)**	**63.85**(2.16)**	99.21(0.50)*	98.00(1.21)*	**7.75**(0.19)**
500	TaskBot	**17.73**(0.34)**	39.95(3.22)	50.28(4.04)	62.85(3.54)	0.00(0.00)	0.00(0.00)	1.09(0.16)**
	ChatBot	0.83(0.12)**	0.60(0.00)	10.70(0.00)	6.48(0.12)**	**100.00**(0.00)	**100.00**(0.00)	**9.29**(0.18)**
	PivotBot	17.50(0.22)**	**42.69**(2.32)*	**54.11**(3.23)*	**65.90**(2.59)*	99.79(0.16)	99.42(0.41)	9.25(0.20)**

Table 12: End-to-end evaluation of single tasks in the INITIAL setting using GODEL as backbone. Almost all differences between GODEL-based models and T5-based models are statistically significant. (*p<0.05, **p<0.01).

# Training dialogs	Model	TOD Evaluation				ODD Evaluation		
		BLEU	Success	Inform	Combined	Accuracy	Success Rate	BLEU
200	TaskBot	**16.48**(0.22)**	**38.79**(5.58)**	**50.78**(7.64)	**61.26**(6.50)*	0.00(0.00)	0.00(0.00)	1.17(0.12)**
	ChatBot	1.19(0.13)**	0.60(0.00)	10.70(0.00)	6.84(0.13)**	**100.00**(0.00)	**100.00**(0.00)	**6.04**(0.25)**
	PivotBot	16.47(0.37)**	34.93(6.31)	45.56(8.24)	56.71(7.34)	97.38(0.50)**	93.22(1.26)**	5.71(0.20)**
500	TaskBot	**17.72**(0.33)**	42.46(2.44)**	**53.53**(2.99)**	65.71(2.74)**	0.00(0.00)	0.00(0.00)	1.00(0.11)**
	ChatBot	1.11(0.07)	0.60(0.00)	10.70(0.00)	6.76(0.07)**	**100.00**(0.00)	**100.00**(0.00)	**6.79**(0.25)**
	PivotBot	17.71(0.43)**	**42.69**(1.82)**	53.40(2.35)	**65.75**(2.16)*	98.65(0.12)**	96.67(0.31)**	6.75(0.14)**

Table 13: End-to-end evaluation of single tasks in the TRANSITION setting using GODEL as backbone. Almost all differences between GODEL-based models and T5-based models are statistically significant. (*p<0.05, **p<0.01).

# Training dialogs	Model	TOD Evaluation				ODD Evaluation		
		BLEU	Success	Inform	Combined	Accuracy	Success Rate	BLEU
200	TaskBot	**16.18**(0.31)**	**38.69**(6.25)	**50.63**(7.32)	**60.84**(6.69)	0.00(0.00)	0.00(0.00)	0.91(0.08)**
	ChatBot	1.14(0.04)**	0.60(0.00)	10.70(0.00)	6.79(0.04)**	**100.00**(0.00)	**100.00**(0.00)	**6.15**(0.40)**
	PivotBot	16.04(0.18)**	34.40(5.55)	45.04(5.63)	55.76(5.52)	98.22(0.41)**	85.37(3.07)**	5.91(0.37)**
500	TaskBot	**17.40**(0.23)	39.19(3.33)	49.83(3.67)	61.90(3.57)	0.00(0.00)	0.00(0.00)	0.90(0.07)**
	ChatBot	1.04(0.07)**	0.60(0.00)	10.70(0.00)	6.69(0.07)**	**100.00**(0.00)	**100.00**(0.00)	**7.17**(0.11)**
	PivotBot	17.26(0.24)*	**40.69**(3.66)**	**51.94**(4.99)*	**63.57**(4.12)**	99.05(0.38)	91.86(2.98)	7.12(0.12)**

Table 14: End-to-end evaluation of single tasks in the MULTIPLE setting using GODEL as backbone. Almost all differences between GODEL-based models and T5-based models are statistically significant. (*p<0.05, **p<0.01).

Evaluation setting	Training setting	# Training dialogs	TOD Evaluation				ODD Evaluation		
			BLEU	Success	Inform	Combined	Accuracy	Success Rate	BLEU
init ODD	INITIAL	500	17.50 (0.22)	**42.69**(2.32)	**54.11**(3.23)	**65.90**(2.59)	**99.79**(0.16)	**99.42**(0.41)	**9.25**(0.20)
	TRANSITION		**17.84**(0.43)	40.49(2.38)	51.30(3.06)	63.74(2.66)	91.65(8.24)	77.54(20.89)	4.66(0.24)
	MULTIPLE		17.44(0.26)	36.63(4.04)	46.73(5.45)	59.11(4.59)	99.30(1.20)	97.93(3.52)	5.41(0.27)
domain transition	INITIAL	500	17.08(0.37)	**43.41**(2.73)	**55.03**(4.11)	**66.30**(3.29)	35.67(14.32)	4.26(3.34)	2.33(0.33)
	TRANSITION		**17.71**(0.43)	42.69(1.82)	53.40(2.35)	65.75(2.16)	98.65(0.12)	96.57(0.31)	6.75(0.14)
	MULTIPLE		17.28(0.19)	38.83(4.11)	49.55(5.57)	61.47(4.76)	**99.58**(0.17)	**98.91**(0.43)	**7.22**(0.21)
multiple ODDs	INITIAL	500	16.44(0.30)	39.46(2.91)	51.50(3.62)	61.92(3.10)	31.28(14.11)	0.57(0.44)	2.21(0.30)
	TRANSITION		17.15(0.43)	38.80(1.13)	50.06(1.46)	61.58(1.39)	93.04(0.79)	53.91(4.02)	5.39(0.09)
	MULTIPLE		**17.26**(0.24)	**40.69**(3.66)	**51.94**(4.99)	**63.57**(4.12)	**99.05**(0.38)	**91.86**(2.98)	**7.12**(0.12)

Table 15: End-to-end cross evaluation of single tasks