A Unified Dialogue User Simulator for Few-shot Data Augmentation

Dazhen Wan[†], **Zheng Zhang**[†], **Qi Zhu**[†], **Lizi Liao**[‡], **Minlie Huang**[†]
[†]CoAI Group, DCST, IAI, BNRIST, Tsinghua University
[‡]Singapore Management University
wandz19@tsinghua.org.cn, {zhangz.goal,zhuq96}@gmail.com,
lzliao@smu.edu.sg, aihuang@tsinghua.edu.cn

Abstract

Pre-trained language models have shown superior performance in task-oriented dialogues. However, existing datasets are on limited scales, which cannot support large-scale pre-training. Fortunately, various data augmentation methods have been developed to augment large-scale task-oriented dialogue corpora. However, they heavily rely on annotated data in the target domain, which require a tremendous amount of data collection and human labeling work. In this paper, we build a unified dialogue user simulation model by pre-training on several publicly available datasets. The model can then be tuned on a target domain with few-shot data. The experiments on a target dataset across multiple domains show that our proposed model brings remarkable performance increases through data augmentation.

Goal: You want to book a taxi. The taxi should go to anatolia and should leave after 22:00. The taxi should depart from kymmoy. Make sure you get car type and contact number.
User: i need a taxi to |anatolia|>. **System:** what time would you like to leave? **User:** i would like to leave after |22:00|>. **System:** there are |9|> options available to you. where would you like to leave from? **User:** i 'd like to leave from |kymmoy|>. **System:** the fare estimate is |9.78|>. do you want to book this taxi now? **User:** no, i would like the contact number if possible. **System:** the contact number is |07356725299|> **User:** thank you very much for your help. **System:** you are welcome. do you need anything else today

Table 1: A dialogue generated by our model with corresponding natural language user goal. The "|*anatolia*|>" with special tag indicates that the word is a value.

1 Introduction

Building a dialogue system typically requires seeing many dialogue samples to fully explore the state-action space and language variations (Pietquin et al., 2011; Gašić et al., 2012). However, collecting a large multi-turn dialogue dataset for a specific domain with crowd-sourced workers is time-consuming and labor-intensive. Dialogue user simulation, a technique that builds an agent to mimic the behaviors of real human users, is an economical alternative, providing unlimited user-system interactions at a low cost (Schatzmann et al., 2007a; Asri et al., 2016; Kreyssig et al., 2018; Gür et al., 2018; Shah et al., 2018).

Generally speaking, there are two ways to use a user simulator to alleviate the data scarcity problem: 1) The simulator serves as an interactive environment to train a system agent directly using reinforcement learning; 2) The simulator is paired with a system agent to generate synthetic dialogues that can be used in dialogue model training. Although the former is popular in traditional dialogue policy learning, in this paper we focus on the latter that facilitates more tasks, such as dialogue state tracking.

Early works on user simulation mainly focus on the user policy module, which perceives and responses in dialogue act level. However, these user simulators, either rule-based (Schatzmann et al., 2007a), statistical-based (Schatzmann et al., 2007b; Schatzmann and Young, 2009), or neural-based (Asri et al., 2016; Gür et al., 2018), require fine-grained dialogue act design/annotation and ignore the noise introduced by semantic parsing and language generation. Therefore, Crook and Marin (2017) proposed an RNN-based natural language level user simulator that generates a user utterance according to the context directly. However, such a data-driven approach requires enough data in the target domain, which is less effective in the low-resource scenario.

Recently proposed large-scale pre-trained language models (PLMs) (Devlin et al., 2019; Radford et al., 2019) provide a solution to few-shot user modeling, since they have achieved great suc-

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 3788–3799
December 7-11, 2022 ©2022 Association for Computational Linguistics

cess through transfer learning on a wide range of NLP tasks, including few-shot dialogue modeling (Madotto et al., 2020). Moreover, the performance can be further improved by pre-training on a large dialogue corpus before fine-tuning (Peng et al., 2020; Wu et al., 2020; Su et al., 2022). As for dialogue simulation, Mohapatra et al. (2021) used GPT-2 (Radford et al., 2019) to model user and system separately and generated dialogues through self-chat. The user generates an utterance according to the user goal in natural language and dialogue context, while the system first generates a query to retrieve entities from a database and then generates a delexicalized response. Kim et al. (2021) proposed to directly generate a whole dialogue with a single BART model (Lewis et al., 2020) given the user goal and candidate entities from the database. However, both methods require domain-specific database interaction, which prevents the simulators from using large scale dialogue corpora for domain-adaptive pre-training.

In this work, we propose an end-to-end user simulator that only takes user goal and dialogue context as input. In order to alleviate the data scarcity problem in the target domain, we pre-train the simulator on a collection of public dialogue datasets of different domains and annotation schema. We also pre-train a context-to-response model without database interaction on similar datasets as the system agent. Note that one could replace this simple model with any system agent such as PPTOD (Su et al., 2022), a powerful end-to-end model in few-shot scenario. Then we fine-tune both agents on the data of target domain and generate synthetic dialogues through self-chat, as shown in Table 1. To enhance the consistency between the user goal and dialogue as well as produce dialogue state for synthetic dialogues, we mark the values in utterances with special tokens for training. Although system response generation is not grounded on database, the value can be delexicalized or replaced for specific tasks.

We conduct both automatic and human evaluation on the quality of synthetic dialogues and show that our method can generate more goal-consistent dialogues than baselines with limited data in the target domain. We also show that using the synthetic dialogues can better improve few-shot dialogue state tracking than baselines.

Our contributions are summarized as follows:

- We collect open-sourced task-oriented dialogue datasets and pre-train a unified user simulator that can chat following the natural language user goal.
- Compared to previous data-driven methods, our simulator can be quickly adapted to a target domain with a few training examples, which tremendously alleviates the data collection cost.
- Experimental results demonstrate that our proposed simulator generates goal-consistent dialogues and improves the performance on dialogue state tracking task in the few-shot learning setting.

2 Related Works

Two lines of works are relevant to this study: neural user simulation and pre-trained language model. We briefly discuss their connections and differences to our work.

2.1 User Simulation

Early user simulation methods are primarily rule-based, such as agenda-based user simulator (ABUS) (Schatzmann et al., 2007a,b; Schatzmann and Young, 2009; Li et al., 2016b). The rules in these methods are highly domain-dependent, which can not be easily generalized to new domains. Meanwhile, the rule-based approaches lead to poor robustness. Statistical methods are then proposed to alleviate these problems through a data-driven approach. Asri et al. (2016) proposed an LSTM-based sequence-to-sequence model that takes the dialogue act-level history as input and generates the next user action. However, this method still requires language understanding and language generation components to accomplish a complete simulator. NUS (Kreyssig et al., 2018) is then proposed, which takes act-level history as input and directly generates natural language response, averting an extra language generation module. One drawback of such a method is that the language generation and policy planning functions are coupled into a single neural network. In such a way, each task's performance will affect the other's performance. Gür et al. (2018) proposed HUS, a hierarchical model with word-level and turn-level encoder-decoder framework. In HUS, a sentence is encoded by a word-level encoder to get its representation. The sentence representations are then fed into the turn-level encoder to get the high-level context representations.

Then based on the context representation, a decoder generates the user response word by word.

There are also some works that train the simulator and system together. Liu and Lane (2017) proposed an iterative policy learning method in which the system policy and the simulator policy are trained together with the same reward. Nevertheless, the assumption of a shared reward is only valid in cooperative dialogue tasks. In a competitive dialogue, such as negotiation, the rewards are apparently different. Takanobu et al. (2020) design role-aware rewards for different persons in the dialogue, which inspire them to achieve their own specific goals.

Recently, data augmentation methods have been proposed for task-oriented dialogue, which plays the same role for RL-based user simulation. Kim et al. (2021) proposed NeuralWOZ, a model-based user simulator with a BART-based (Lewis et al., 2020) Collector and a RoBERTa-based (Liu et al., 2019) Labeler to generate and annotate new dialogue corpus. Mohapatra et al. (2021) designed different types of prompts and used GPT-2 (Radford et al., 2019) for system and user simulators to generate dialogue corpus. However, both works require target domain data or data with a similar schema to build the simulator. In contrast, our proposed base model no longer depends on target data and can be launched with minimal target data.

2.2 Pre-trained Language Models

Another line of related work is pre-trained language models. Through self-supervised learning on large-scale unlabeled general text, pre-trained language models (PLMs) achieved SoTA performance on various language understanding and generation tasks. According to the model architecture, PLMs can be roughly classified into two categories: bi-directional and uni-directional. The bi-directional PLMs, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), are usually trained using Transformers with bi-direction attentions through token prediction, such as masked language model (MLM). These models are often applied for classification tasks, such as language understanding and question answering (QA). While the uni-directional PLMs, including GPT(Radford et al., 2018), GPT-2(Radford et al., 2019) and GPT-3 (Brown et al., 2020), are trained using a Transformer decoder with uni-direction attention to maximize the generation likelihood of conditional language generation.

PLMs are also applied on open-domain and task-oriented dialog system. For open-domain chit-chat models, there are DialoGPT(Zhang et al., 2020), Meena(Adiwardana et al., 2020), Blender(Roller et al., 2021), Plato(Bao et al., 2020), and EVA(Zhou et al., 2021). These models are built upon massive amount of dialogue corpus with conditional generation objective. For PLMs in task-oriented dialogues, the researchers are more concerned about understanding tasks. Wu et al. (2020) proposed TOD-BERT, which adopts the original BERT model and obtains strong performance on several sub-tasks. There are also works applying uni-directional generative models on task-oriented dialogues to build end-to-end models, such as SOLOIST(Peng et al., 2021) and SimpleToD(Hosseini-Asl et al., 2020).

Recently, some works have been using prompt engineering for few-shot learning in dialogue systems. Lin et al. (2021) designed prompts for multichoice and extractive QA to link the task of QA and DST and then used only QA data to build a DST model for zero-shot learning. Lee et al. (2021) proposed to use schema-driven prompting and natural language descriptions to boost the performance of dialogue state tracking. While in PPTOD(Su et al., 2022), the model is trained on four different sub-tasks using unified prompts to boost the performance of each other.

3 Method

3.1 Overview

This section describes the details of pre-training data preparation and how we train the user simulator. As illustrated in Figure 1, we first pre-train the simulator and system agent using a collection of different task-oriented dialogue datasets that are firstly transformed into a unified format. Then we use MultiWOZ(Eric et al., 2020), a large-scale publicly available multi-domain task-oriented dialogue dataset, as the target domain and tune the models in few-shot settings. After that, the user simulator interacts with the system agent to generate dialogues based on user goals. To obtain the dialogue states of synthetic dialogues, we design an automatic annotation procedure that aligns values expressed in the context with corresponding slots in the user goal.

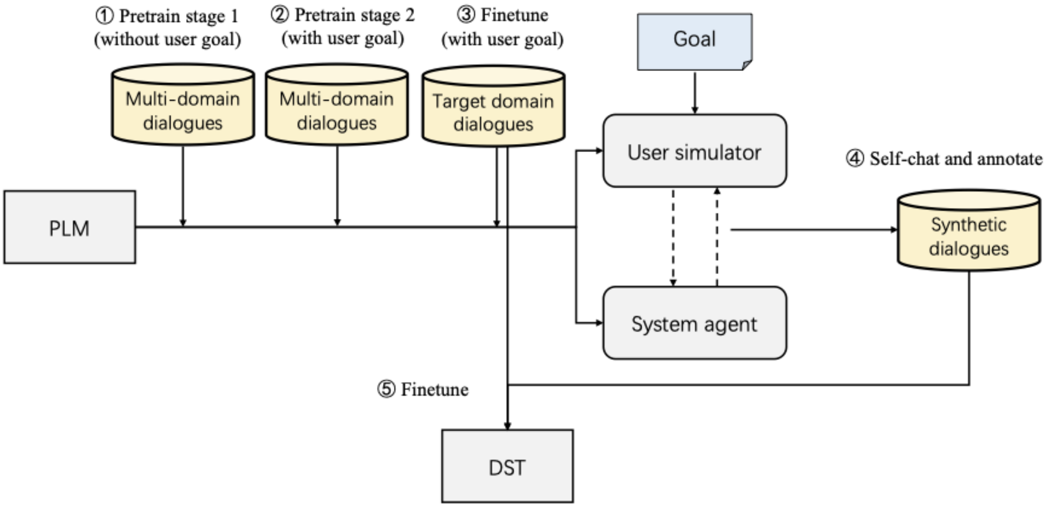

Figure 1: Illustration of our augmentation method. 1) Pretrain user simulator and system agent on multi-domain dialogues without goal labels. 2) Pretrain user simulator and system agent on multi-domain dialgoues with paraphrased user goals. 3) Finetune user simulator and system agent on target domain dialogues. 4) The user simulator interacts with the system agent to generate dialogues, which are annotated automatically. 5) The synthetic dialogues and the original target domain dialogues are used to finetune the downstream DST model.

3.2 User Simulator

Our user simulator uses T5 (Raffel et al., 2020) as the backbone model. T5 is a pre-trained text-to-text transformer model with encoder-decoder architecture. The input to the model is the concatenation of task description, user goal, and dialogue context:

- **Task description** I is a natural language description of the task, indicating the domains of the dialogue to generate. It is auto-generated by filling the template "*Below is the conversation between a user and a system about* ______" with domain names, which is inspired by Zheng et al. (2022).
- **User goal** G is a natural language instruction telling the user simulator the constraints of the target entity and which attributes of the entity are of interest. Formally, there are two kinds of slots in the user goal: 1) informable slots whose values should be expressed by the user, and 2) requestable slots that require the user to get the value through conversation.
- **Dialogue context** $C_t = (u_1, s_1, u_2, ..., s_t)$ consists of utterances up to t-th turn, where u_i and s_i are utterances from user and system respectively.

For those datasets without user goal annotation, we use user dialogue acts to deduce user goals automatically, as shown in Table 2. **First**, we merge all user dialogue acts from the dialogue to get a structured goal, which is represented as a list of (domain, intent, slot, value) tuples. **Next**, a textual goal is generated using natural language templates that are manually designed according to the intent. However, such a goal follows simple patterns and is of low language diversity, which potentially limits the model generalization ability. To alleviate this problem, we **further** paraphrase each sentence in a goal with an online paraphrasing tool[1]. For example, given a dialogue act $(Restaurant, Inform, Cuisine, American)$, the template-generated description is "*The cuisine of food served in the restaurant is american*", while the paraphrased one could be "*The restaurant's cuisine is American.*"

Given the task description I, user goal G, and dialogue context C_{t-1} as the input of T5 encoder, the user simulator θ is trained to decode a goal-grounded response $u_t = (w_1^t, w_2^t, ..., w_{|u_t|}^t)$ auto-regressively:

$$p_\theta(u_t|I, G, C_{t-1}) = \prod_{i=1}^{|u_t|} p_\theta(w_i^t|w_{<i}^t, I, G, C_{t-1}).$$

The loss function on a single dialogue of T turns is therefore calculated as:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log p_\theta(u_t|I, G, C_{t-1}). \quad (1)$$

[1] https://quillbot.com/

User dialogue acts: [(restaurant_1, inform, city, san jose), (restaurant_1, inform, cuisine, american), (restaurant_1, request, street_address, ?), (restaurant_1, request, phone_number, ?), (restaurant_1, inform, price_range, moderate)]
Template goal: You are looking for a restaurant to dine. The city in which the restaurant is located is san jose. The cuisine of food served in the restaurant is american. You want to know the address of the restaurant. You want to know the phone number of the restaurant. The price range for the restaurant is moderate.
Paraphrased goal: You are looking for a restaurant to dine. The city in which the restaurant is located is san jose. The restaurant's cuisine is American. You want to know the address of the restaurant. You're looking for the restaurant's phone number. The restaurant has a reasonable price range.

Table 2: An example of user dialogue acts, templated goal and paraphrased goal.

Dataset	Domains	# Dialogues
DSTC2 (2014)	restaurant	3.2K
AirD (2018)	flight	40K
CamRest (2017)	restaurant	676
rnnlg (2016)	restaurant, hotel	2K
KVReT (2017)	car assistant	3K
msre2e (2018)	restaurant, movie, car	10K
m2m (2018)	restaurant, movie	3K
Frames (2017)	flight, hotel	1.4K
schema (2020)	20 domains	23K
Taskmaster (2019)	13 domains	13K
WoW (2019)	knowledge dialogue	22K
MultiWOZ (2020)	5 domains[2]	10K

Table 3: The datasets for domain adaptive pre-training and target task finetuning.

3.3 System Agent

To generate synthetic dialogues through self-chat, we also train a vanilla system agent ϕ. The only difference between the system agent and user simulator is that the system does not take user goal as input. The loss function is defined as:

$$\mathcal{L}(\phi) = -\sum_{t=1}^{T} \log p_\phi(s_t|I, C_{t-1}, u_t). \quad (2)$$

3.4 Domain Adaptive Pre-training

We perform domain adaptive pre-training (DAPT) for both user and system models before fine-tuning (FT) on the target domain. To enhance the knowledge transfer between DAPT and FT, we select 12 representative open-sourced dialogue datasets covering a variety of domains, as shown in Table 3. The middle group of datasets has user goal or user dialogue acts annotation (for WoW we regard all grounded knowledge sentences of user as the user goal), while the Top group of datasets has neither. Therefore, we first train the user simulator without the user goal on the top group datasets and then continue to train the simulator on the middle group of datasets. We jointly trained the system agent on both groups of datasets except WoW since learning to generate a response without knowledge is meaningless in WoW dialogue. Besides, we use special tokens to wrap the values from dialogue acts once they appear in the utterance, as shown in Table 1.

We use the same backbone model (t5-base-lm-adapt[3], 247.5M parameters) and hyper-parameters to train both user and system models. We use Adafactor optimizer (Shazeer and Stern, 2018) with learning-rate=1e-5, batch-size=256, epoch=10, and linear learning rate schedule without warming up.

4 Experiment

4.1 Data Augmentation

To generate synthetic dialogues in a target domain, we first fine-tune both user and system agents on a few samples in that domain. We use MultiWOZ 2.1 (Eric et al., 2021) as the target dataset and tune the models with the same hyper-parameters except a smaller epoch=5. To examine whether the simulator is endowed with the general conversational ability and can quickly be adapted to new scenarios with limited data, we set the available dialogues to 5%/10% of the original training set of the target dataset. Then we sample user goals from the remaining training set and generate dialogues through self-chat. We set the decoding hyper-parameters of both model to: top_k = 50, top_p = 0.9, temperature = 1.

4.2 State Annotation

For some task-oriented dialogue tasks, such as dialogue state tracking and end-to-end modeling, annotated dialogue state labels are required. However, the above data augmentation framework does not give dialogue state labels. Since the simulator can mark the value with special tokens, we extract the values from user utterances and match them with corresponding slots in the user goal. When there

[2]In fact, MultiWOZ has 7 domains in toal. We don't use the "hospital" domain and "police" domain, because there are no dialogues from these two domains in validation and testing sets.

[3]https://huggingface.co/google/t5-base-lm-adapt

are multiple values for a single slot, the correspondence is determined with the help of transformer attention scores. Then we accumulate the expressed values until the current turn as the dialogue state annotation.

Dataset	Goal	Context	Unknown
train	85.24%	0.52%	14.24%
dev	86.68%	0.59%	12.73%
5% Ours	94.60%	3.34%	2.05%
10% Ours	94.21%	3.71%	2.08%

Table 4: The source of slot values present in the original train/dev set and the generated dialogues. The percentage means how many values can be found from the two sources. 5% Ours means dialogues generated by our user simulator and system agent, which are finetuned on 5% training data.

Since the user simulator is required to generate responses conditioned on task description and goals, the slot values present in the generated dialogues should be consistent with the grounded user goal. We checkout where the generated values come from to examine whether the augmented dialogues faithfully follow the instructions. As shown in Table 4, the dialogues in the original MultiWOZ data have many unknown slot values which are not found in their goals or context. However, using those data to train our user simulator, the resulting synthesized data's values share more common values with the goal and context. It means our proposed simulator model can capture the connections between grounded user goal and dialogues, making the generated responses more consistent with goal and context.

4.3 Baselines

To validate the effectiveness of our proposed method, we compare with two state-of-the-art methods on synthesized data augmentation. More details about these two different methods are illustrated as below:

- **NeuralWOZ** (NW) (Kim et al., 2021) is a data augmentation method that synthesizes labeled dialogues with a Collector and a Labeler. The Collector takes Goal Instructions and API Call Results as inputs and generates a dialogue, and the Labeler annotates the dialogue state by formulating it as a multi-choice problem. The Collector model is built upon BART (Lewis et al., 2020) (406M parameters) and the Labeler model is based on RoBerta (Liu et al., 2019).

- **Simulated Chats** (SC) (Mohapatra et al., 2021) proposed a dialogue generation framework that mimics the data collection process employed by crowd workers. It also involves a user and agent bot, which takes instructions and KB results as input to generate dialogues. A Longformer model is involved to make evaluation and selection of the generated results. Both models are trained upon GPT-2 (Radford et al., 2019) backbone model (124M parameters).

4.4 Data Statistics

To better examine the quality of augmented data, we statistically analyzed some metrics of generated dialogues, as shown in Table 5. The averaged turn number indicates the session length of each dialogue. Response length indicates the averaged token numbers within the responses. Goal recall measures how many informable goals in the user goal show up in the generated dialogues. Given a dialog with T turns and the user goal G, the language model loss is:

$$\mathcal{L}(\psi) = -\frac{1}{|C_T|}\sum_{t=1}^{T}\log p_{\psi}(C_T|G),$$

where ψ is a T5-small[4] fine-tuned on the full training set of the target domain that generates the whole dialogue from user goal. Compared to baseline methods, the session length of our generated dialogues is longer, which means our proposed simulator model can conduct more extended interactions with the system. Compared to NW and SC, the goal recall of our model is much better. It means that our simulator does learn to generate dialogues following the task descriptions and goals by capturing the goal-dialogue connections within the training data. As for the language model loss, our model still achieves a better score, indicating our generated dialogues are more similar to the dialogues of target dataset.

Table 6 demonstrates the Distinct-N (Li et al., 2016a) scores of the generated dialogues. Higher Distinct scores mean more diverse utterances. NeuralWOZ usually achieves the highest distinct scores under different settings. We further examine the

[4] https://huggingface.co/t5-small

Dataset	# Turn	Response Length	Goal Recall	LM Loss
train	12.93	11.78 / 16.17 / 13.81	0.8051	1.577
dev	13.74	11.98 / 16.36 / 14.01	0.8599	1.525
5% NW	13.19	12.34 / 13.09 / 12.69	0.1785	3.143
5% SC	11.59	9.27 / 14.46 / 11.86	0.1926	1.778
5% Ours w/o pretrain	11.75	11.29 / 18.46 / 14.87	0.6334	1.883
5% Ours	15.17	11.05 / 13.80 / 12.43	**0.8506**	**1.689**
10% NW	13.25	10.53 / 15.19 / 12.67	0.2227	2.768
10% SC	12.26	9.97 / 14.48 / 12.21	0.2468	1.673
10% Ours w/o pretrain	12.92	11.30 / 17.58 / 14.44	0.7419	1.595
10% Ours	15.03	10.67 / 13.98 / 12.32	**0.8364**	**1.574**

Table 5: Statistics of the synthesized dialogues, including averaged turn number, response length, goal recall, and language model loss. The three values of response length respectively mean the length of user response, agent response, and average length. Besides the augmented data, statistics of the original train/dev set are also reported.

Dataset	Distinct-1	Distinct-2	Distinct-3	Distinct-4
train	0.006 / 0.018 / 0.011	0.065 / 0.099 / 0.072	0.191 / 0.251 / 0.207	0.346 / 0.418 / 0.373
dev	0.021 / 0.041 / 0.025	0.153 / 0.207 / 0.158	0.353 / 0.426 / 0.368	0.544 / 0.608 / 0.564
5% NW	0.082 / 0.092 / 0.072	0.291 / 0.325 / 0.263	0.522 / 0.578 / 0.494	0.703 / 0.762 / 0.685
5% SC	0.004 / 0.010 / 0.006	0.023 / 0.048 / 0.035	0.059 / 0.112 / 0.089	0.109 / 0.185 / 0.155
5% Ours w/o pretrain	0.020 / 0.048 / 0.031	0.104 / 0.183 / 0.136	0.245 / 0.366 / 0.302	0.400 / 0.529 / 0.468
5% Ours	0.017 / 0.048 / 0.028	0.096 / 0.176 / 0.123	0.235 / 0.351 / 0.281	0.389 / 0.519 / 0.449
10% NW	0.053 / 0.089 / 0.062	0.211 / 0.293 / 0.225	0.412 / 0.522 / 0.438	0.594 / 0.699 / 0.625
10% SC	0.004 / 0.010 / 0.006	0.021 / 0.043 / 0.031	0.059 / 0.101 / 0.080	0.107 / 0.174 / 0.145
10% Ours w/o pretrain	0.017 / 0.040 / 0.026	0.094 / 0.159 / 0.118	0.221 / 0.320 / 0.266	0.366 / 0.476 / 0.423
10% Ours	0.016 / 0.043 / 0.026	0.094 / 0.161 / 0.116	0.225 / 0.325 / 0.265	0.375 / 0.489 / 0.428

Table 6: Distinct-N scores of the original train/dev set and the augmented data generated by NeuralWOZ(NW), Simulated Chats(SC), and our proposed method. Ours w/o pretrain means the original T5 model is fine-tuned on the target data without pretraining on the 11 datasets. The three values implied the Distinct score of user response, agent response, and averaged score. The closest values of synthesized data to the scores of original development data are underlined.

Method	5%-shot	10%-shot
None	27.6	37.1
NeuralWOZ	35.2	40.4
Simulated Chats	34.5	40.8
Ours w/o pretrain	34.6	40.6
Ours	**37.5**	**41.2**

Table 7: Joint goal accuracy of the T5 DST models trained on different data. 'None' refers to the setting where only the 5% and 10% original dataset is used during training. For each method, we report the best result on different amounts of augmented data.

difference between synthesized and original values. As we can see, our proposed model can achieve closest values in 5% setting, which means its augmented dialogues are similar to the original data.

4.5 Dialogue State Tracking

We use the same few-shot data along with the synthetic data to train a dialogue state tracking model to examine the quality of the synthesized dialogues generated by different augmentation methods. As for the DST model, we use a T5-small model as the backbone model and finetune it on the augmented data. It converts the DST task into an encoder-decoder framework, by taking tokenized dialogue context as input and generate the response token sequence. The input is formated as "user: u_1 system: s_1 ... user: u_t" and the output is like "[$domain_t^1$]([$slot_t^{1,1}$] [$value_t^{1,1}$], ...); ...; [$domain_t^m$] ([$slot_t^{m,1}$] [$value_t^{m,1}$],...,[$slot_t^{m,n}$] [$value_t^{m,n}$])". Here u_t and s_t indicate the user and system utterance in the t-th turn, while ($domain_t^i$, $slot_t^{i,j}$, $value_t^{i,j}$) refers to a candidate slot value tuple in the t-th turn.

To examine the few-shot learning ability of our model, we conduct few-shot learning experiments by finetuning the model on 5% and 10% original data with the augmented data. Note that those small sets of original data are both utilized to train the simulator and the DST model. Table 7 shows the best performance across different sizes of augmented dialogues ranging from 100 to 5000. Joint goal accuracy means the slot and value prediction accuracy within each turn. Based on these results, we can see that the augmented dialogues can dramatically improve DST performances. Compared to methods that use only the small amount original data for model training, the joint goal accuracies of four augmented methods achieves a relative increase of 28% in 5%-shot setting. When more original data is available, this performance growth dropped to 10% in 10%-shot setting.

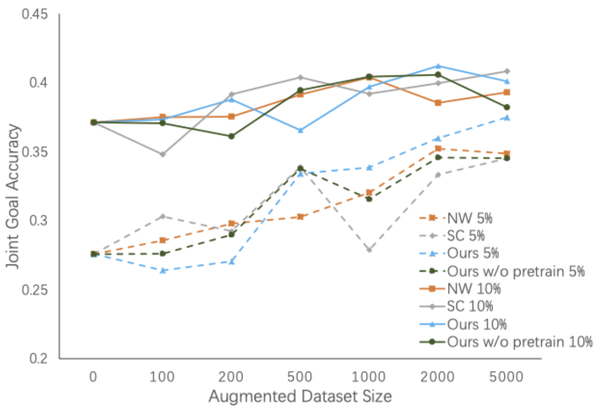

Figure 2: The joint goal accuracy result of the T5 DST models trained on different amounts of original and augmented data.

We further analyze how the amount of augmented data affects DST performance. Intuitively, more augmented data leads to better performance if it is of high quality. We conduct experiments on both 5%-shot and 10%-shot settings. As shown in Figure 2, when the quantity of augmented data is increased, sustained growth in goal accuracy can be obtained. In particular, our method achieves the highest increase among the four baseline methods. When more original data is used for training, the growth becomes flattened. If we remove the pre-training stage (Ours w/o pretrain), the performance increment declines. It indicates that the pre-training on the commonly used task-oriented dialogue datasets does endow our model with more generalization abilities on conversational skills.

	NW	**SC**	**Ours**
Consistency	3.8	3.4	4.2
Grammar	4.0	4.1	4.4
Fluency	4.1	3.8	3.9

Table 8: Human evaluation scores on the augmented data of NeuralWOZ, Simulated Chats and our method.

4.6 Human Evaluation

To further access the performance difference between our proposed model and the baseline methods, we also conduct human evaluation through the Amazon Mechanical Turk (AMT) platform. Following the scheme used in Mohapatra et al. (2021), we use three popular metrics, including Consistency, Grammar, and Fluency. Generally speaking, Consistency measures whether the dialogue is relevant to the grounded task description and user goal. Grammar indicates whether the generated dialogues are grammatical or not. Fluency measures whether the responses of the user and system are coherent with the dialogue context. The crowd workers are asked to give a score for each metric on the Likert scale (1-5).

As demonstrated in Table 8, the augmented dialogues of our proposed model are of high quality and achieve highest scores on Consistency and Grammar. As for the Fluency metric, the three methods perform similarly. The good performance on Consistency and Grammar further illustrates that our proposed model could better model the dependencies between user goal and dialogues.

5 Conclusion

In this paper, we develop a unified dialogue simulation base model. We collect a collection of representative open-sourced task-oriented dialogue datasets and transform them into a unified format, including task description, user goal, and dialogue context. Through domain adaptive pre-training on those datasets, our proposed model successfully learns the essential conversation ability and skills of a user and can be easily adapted to a downstream task by using only a small amount of target data. To examine the performance of our model, we use it to generate synthesized dialogues and conduct dialogue state tracking experiments. The results demonstrate that our proposed model brings remarkable performance increases through data augmentation. We believe our proposed method can foster future research on dialogue simulation and data augmentation on low-resource task-oriented dialogue tasks.

Acknowledgements

This paper was supported by the National Key Research and Development Program of China (No. 2021ZD0113304), the National Science Foundation for Distinguished Young Scholars (with No. 62125604), and the NSFC projects (Key project with No. 61936010 and regular project with No. 61876096). This work was also supported by the Guoqiang Institute of Tsinghua University, with Grant No. 2019GQG1 and 2020GQG0005, and sponsored by Tsinghua-Toyota Joint Research Fund.

Limitations

Our work has the following limitations: 1) Although our method can generate more goal-consistent dialogues, using these dialogues to augment few-shot DST shows limited improvement over baselines. 2) While our pre-trained user simulator can be used in many different ways, we only verify its effectiveness in dialog state tracking on MultiWOZ dataset. 3) We need to annotate the value spans in utterances. However, this can be done by a BIO tagging model pre-trained on task-oriented dialogues. 4) We do not explore using different system agents to generate dialogues, which may vastly affect the quality of synthetic dialogues.

References

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

Layla El Asri, Jing He, and Kaheer Suleman. 2016. A Sequence-to-Sequence Model for User Simulation in Spoken Dialogue Systems. In *Proc. Interspeech 2016*, pages 1151–1155.

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, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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 4516–4525, Hong Kong, China. Association for Computational Linguistics.

Paul Crook and Alex Marin. 2017. Sequence to sequence modeling for user simulation in dialog systems. In *INTERSPEECH*, pages 1706–1710.

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

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219, Saarbrücken, Germany. Association for Computational Linguistics.

Mihail Eric, Nicole Chartier, Behnam Hedayatnia, Karthik Gopalakrishnan, Pankaj Rajan, Yang Liu, and Dilek Hakkani-Tur. 2021. Multi-sentence knowledge selection in open-domain dialogue. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 76–86, Aberdeen, Scotland, UK. Association for Computational Linguistics.

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, Marseille, France. European Language Resources Association.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.

M Gašić, Matthew Henderson, Blaise Thomson, Pirros Tsiakoulis, and Steve Young. 2012. Policy optimisation of pomdp-based dialogue systems without state space compression. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 31–36. IEEE.

Izzeddin Gür, Dilek Hakkani-Tür, Gokhan Tür, and Pararth Shah. 2018. User modeling for task oriented dialogues. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 900–906. IEEE.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems*, volume 33, pages 20179–20191. Curran Associates, Inc.

Sungdong Kim, Minsuk Chang, and Sang-Woo Lee. 2021. NeuralWOZ: Learning to collect task-oriented dialogue via model-based simulation. 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 3704–3717, Online. Association for Computational Linguistics.

Florian Kreyssig, Iñigo Casanueva, Paweł Budzianowski, and Milica Gašić. 2018. Neural user simulation for corpus-based policy optimisation of spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 60–69, Melbourne, Australia. Association for Computational Linguistics.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue state tracking with a language model using schema-driven prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4937–4949, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016b. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.

Xiujun Li, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv preprint arXiv:1807.11125*.

Zhaojiang Lin, Bing Liu, Andrea Madotto, Seungwhan Moon, Zhenpeng Zhou, Paul Crook, Zhiguang Wang, Zhou Yu, Eunjoon Cho, Rajen Subba, and Pascale Fung. 2021. Zero-shot dialogue state tracking via cross-task transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7890–7900, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2017. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489. IEEE.

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*.

Andrea Madotto, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Language models as few-shot learner for task-oriented dialogue systems. *arXiv preprint arXiv:2008.06239*.

Biswesh Mohapatra, Gaurav Pandey, Danish Contractor, and Sachindra Joshi. 2021. Simulated chats for building dialog systems: Learning to generate conversations from instructions. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1190–1203, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824.

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

Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, and Hervé Frezza-Buet. 2011. Sample-efficient batch reinforcement learning for dialogue management optimization. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(3):1–21.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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

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

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8689–8696.

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.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007a. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152.

Jost Schatzmann, Blaise Thomson, and Steve Young. 2007b. Statistical user simulation with a hidden agenda. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 273–282.

Jost Schatzmann and Steve Young. 2009. The hidden agenda user simulation model. *IEEE transactions on audio, speech, and language processing*, 17(4):733–747.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676, Dublin, Ireland. Association for Computational Linguistics.

Ryuichi Takanobu, Runze Liang, and Minlie Huang. 2020. Multi-agent task-oriented dialog policy learning with role-aware reward decomposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 625–638, Online. Association for Computational Linguistics.

Wei Wei, Quoc Le, Andrew Dai, and Jia Li. 2018. AirDialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3844–3854, Brussels, Belgium. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference on North American Chapter of the Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics.

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

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. 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, Online. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. 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.

Chujie Zheng, Sahand Sabour, Jiaxin Wen, and Minlie Huang. 2022. Augesc: Large-scale data augmentation for emotional support conversation with pretrained language models.

Hao Zhou, Pei Ke, Zheng Zhang, Yuxian Gu, Yinhe Zheng, Chujie Zheng, Yida Wang, Chen Henry Wu, Hao Sun, Xiaocong Yang, et al. 2021. Eva: An open-domain chinese dialogue system with large-scale generative pre-training. *arXiv preprint arXiv:2108.01547*.