SynthDST: Synthetic Data is All You Need for Few-Shot Dialog State Tracking

Atharva Kulkarni[1*], Bo-Hsiang Tseng[2], Joel Ruben Antony Moniz[2], Dhivya Piraviperumal[2], Hong Yu[2], Shruti Bhargava[2]

[1]Language Technologies Institute, Carnegie Mellon University,
[2]Apple Inc.

atharvak@cs.cmu.edu,
{bohsiang_tseng, joelrubenantony_moniz,
{dhivyaprp, hong_yu, shruti_bhargava}@apple.com

Abstract

In-context learning with Large Language Models (LLMs) has emerged as a promising avenue of research in Dialog State Tracking (DST). However, the best-performing in-context learning methods involve retrieving and adding similar examples to the prompt, requiring access to labeled training data. Procuring such training data for a wide range of domains and applications is time-consuming, expensive, and, at times, infeasible. While zero-shot learning requires no training data, it significantly lags behind the few-shot setup. Thus, '*Can we efficiently generate synthetic data for any dialogue schema to enable few-shot prompting?*' Addressing this question, we propose `SynthDST`, a data generation framework tailored for DST, utilizing LLMs. Our approach only requires the dialogue schema and a few hand-crafted dialogue templates to synthesize natural, coherent, and free-flowing dialogues with DST annotations. Few-shot learning using data from `SynthDST` results in $4-5\%$ improvement in Joint Goal Accuracy over the zero-shot baseline on MultiWOZ 2.1 and 2.4. Remarkably, our few-shot learning approach recovers nearly 98% of the performance compared to the few-shot setup using human-annotated training data[1].

1 Introduction

Dialogue State Tracking (DST) is an integral task in task-oriented dialogue systems that predicts the user intentions for each turn by mapping them to predefined slot-value pairs (Henderson, 2015). DST systems capture important information essential to model the downstream dialogue policy and help generate actionable responses (Jacqmin et al., 2022). Prior literature has typically framed DST as either a multi-class classification task (Henderson et al., 2014; Mrkšić et al., 2017; Wu et al., 2020; Chen et al., 2020) or a sequence-to-sequence learning task (Wu et al., 2019; Kim et al., 2020; Hosseini-Asl et al., 2020; Lee et al., 2021; Shin et al., 2022). With the rise of Large Language Models (LLMs), various techniques have been proposed to harness their emergent capabilities for dialogue state tracking (Hu et al., 2022; Chen et al., 2023a; Heck et al., 2023; King and Flanigan, 2023; Yang et al., 2023).

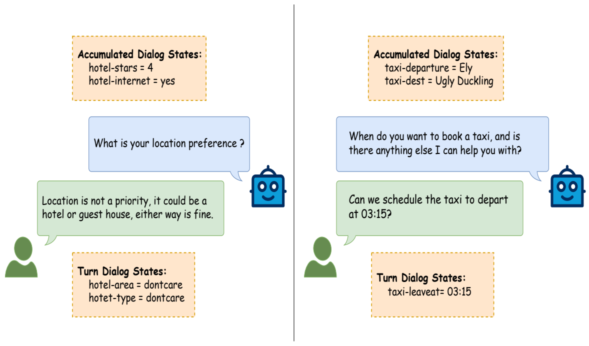

Figure 1: One of these is a dialog generated by `SynthDST`. Each dialog contains conversation history (as accumulated dialog states), system turn, user turn, and current turn's dialog states. Can you guess which dialog is synthetically generated by `SynthDST`?[2].

Most approaches for DST necessitate access to gold-standard human-annotated data for supervised fine-tuning (Wu et al., 2019; Shin et al., 2022) or retrieval-based in-context learning (Hu et al., 2022; King and Flanigan, 2023). This comes with four main drawbacks. First, curating fine-grained utterance-level annotated dialogue data in a *Wizard-of-Oz / human-to-human conversation* setup (e.g., MultiWOZ) is both time-consuming and expensive (Budzianowski et al., 2018). Second, many DST benchmarks contain incorrect annotations (Ye et al., 2022), which can hinder learning and may introduce

*Work done during internship at Apple Inc.
[1]Our synthetic data and code can be accessed at https://github.com/apple/ml-synthdst.

[2]The right example is synthetically generated.

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics Volume 1: Long Papers, pages 1988–2001

spurious biases (Qian et al., 2021). Third, nearly all DST datasets are confined to a limited number of domains. Training on these datasets limits the models' ability to generalize to unseen domains, thereby hampering their suitability for real-world deployment (Dingliwal et al., 2021). Fourth, real-world applications may need to regularly add new domains or modify existing schemas. However, iterating on data collections may pose a significant challenge (Jacqmin et al., 2022).

While zero-shot prompting of LLMs using only the dialogue schema provides a data-less approach for DST, it under-performs compared to the retrieval-based few-shot prompting that adds semantically similar training examples in the prompt (Hu et al., 2022; King and Flanigan, 2023). Given these challenges, one may wonder: *'How can we leverage LLMs' in-context learning capabilities when we do not have access to annotated training data?'* Or conversely, *'Can we efficiently generate synthetic data for any dialogue schema to enable few shot prompting?'* In this work, we aim to answer this.

We introduce `SynthDST`, an LLM-based approach for generating dialogues with dialog state annotations. `SynthDST` takes a dialogue schema as input and outputs four objects: the conversation state, the next system response, the next user response, and the updated conversation state. For this, it uses predefined intents and intent transitions (Table 1), along with hand-crafted templates (Tables 2, 3). The pipeline is detailed in Figure 2, and an example can be seen in Figure 1. It first programmatically generates raw data for the four output objects. Then, it transforms the raw intents into sentences with templates and further paraphrases them into natural language using LLMs. We evaluate `SynthDST` using the IC-DST framework (Hu et al., 2022) on MultiWOZ 2.1 (Eric et al., 2020) and 2.4 (Ye et al., 2022). Our results show a $4-5\%$ improvement over the zero-shot baseline on both datasets. Moreover, few-shot learning with `SynthDST` data achieves approximately 98% and 95% of the performance when using training data for MultiWOZ 2.1 and 2.4, respectively. In summary, our contributions are two-fold:

- We propose `SynthDST`, a scalable domain agnostic framework for generating synthetic dialogue data with dialog state annotations.
- We empirically demonstrate that retrieval-based few-shot prompting with `SynthDST`'s synthetic data surpasses both the zero-shot and random few-shot learning baselines. Moreover, it reaches close to the few-shot prompting performance with human-annotated training data.

2 Related Work

2.1 Synthetic Data Generation for Dialog

The advent of large language models has brought about a significant transformation in synthetic data generation. *LAD* (Mehri et al., 2022) generates linguistically diverse synthetic dialogues by imposing structural constraints on prompts for intent prediction, slot filling, and next action prediction. *RSODD* (Bae et al., 2022) adopts a human-in-the-loop approach to craft role-specific open-domain dialogues. Specifically, it takes role specification and examples designed by dialogue developers to generate artificial conversations, followed by human editing. On similar lines, Chen et al. (2023b) introduced *PLACES*, a framework utilizing topic information, background context, and expert-written conversations as in-context examples for synthetic dialogue generation. *Synergy* (Peng et al., 2021) adopts a different approach by modifying simulated dialogue sketches, each comprising multi-turn dialogue actions and belief states. A natural language generation module transforms these actions into natural language. Lastly, *DIALOGIC* (Li et al., 2022) presents a controllable dialogue simulation method that generates DST-annotated dialogues using a seed corpus.

In summary, the NLP community has shown a growing interest in synthetic data generation for dialogue applications. However, frameworks like *RSODD*, *PLACES*, and *Synergy* demand a level of human supervision and lack slot-value annotations, rendering them unsuitable for DST. While *DIALOGIC* generates synthetic data with dialog state annotations, it has limited coverage of dialogue acts, needs human intervention for annotation correction, and necessitates a seed corpus. Addressing these challenges, `SynthDST` provides more control over the generated dialogues by grounding them in dialogue states. Moreover, `SynthDST` does not require human intervention in filtering or editing the synthetic data, facilitating greater scalability.

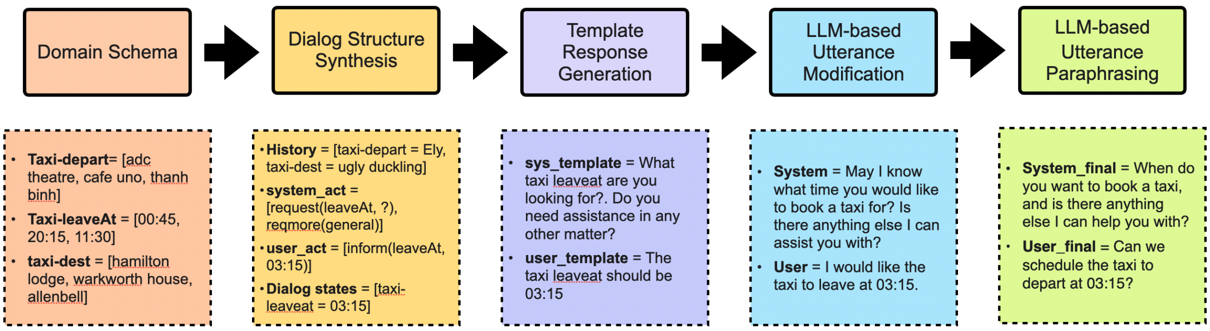

Figure 2: Overall pipeline of SynthDST for synthetic dialog generation

2.2 Zero and Few-Shot Learning for DST

Significant research is dedicated to dialogue state tracking with in-context learning. Lin et al. (2021) proposed a zero-shot cross domain DST method by prompting the T5 model (Raffel et al., 2020) with slot descriptions. Madotto et al. (2021) assessed different language models for DST through prompt-based few-shot learning. Other approaches, such as UnifiedSKG (Xie et al., 2022) and InstructDial (Gupta et al., 2022), introduce multi-tasked and instruction-tuned variants of T5 and BART, which exhibit strong zero-shot DST performance. IC-DST (Hu et al., 2022) frames DST as a text-to-SQL problem using the codex version of GPT-3. RefPyDST framework (King and Flanigan, 2023) formulates DST as a Python programming task, retrieves more diverse in-context examples, and introduces a novel re-weighting method during decoding. Heck et al. (2023) provide empirical evidence that ChatGPT can yield competitive results in DST without any complex prompting. More recently, a dual prompting strategy was proposed by Yang et al. (2023), decomposing DST into slot and value generation tasks. Compared to the above research, our work complements the zero-shot prompting techniques to harness the capabilities of LLMs without collecting human-annotated data.

3 Methodology

Figure 2 outlines the SynthDST's data generation pipeline. It utilizes just the dialogue schema and a set of handcrafted templates to generate fluent dialogues with dialog state annotations. Specifically, each dialog generated using SynthDST comprises a quartet of dialogue history, system turn, user turn, and the current turn's dialog states. SynthDST employs a three-step approach for generating dialogue data. We explain each step below.

3.1 Dialogue Structure Synthesis

Abstract Dialogue Model. The effectiveness of SynthDST in generating meaningful dialogues relies on its strategic selection of system and user dialogue acts. A dialogue act, represented by intent and its associated slot-value pairs, indicates the specific communicative action of a user and the system (Core and Allen, 1997; Traum, 1999; Budzianowski et al., 2018). Selecting valid dialogue acts for each system and user turn is non-trivial, as random pairing may yield incoherent and illogical dialogues. To address this issue, we adopt an approach similar to that of Campagna et al. (2020) by creating an abstract dialogue model. We define it as a set of system-user dialogue acts along with their valid transitions. Table 1 depicts the system-user intents and their valid transitions used in the abstract dialogue model. Our meticulously curated list of system-user intent transitions is independent of any dialogue domains and datasets, following a natural dialogue progression. Hence, it proffers greater generalizability and scalability.

Synthesizing Dialogue Structure from Abstract Dialogue Model. For synthesizing a sample, we begin by selecting a system-user intent pair from Table 1. Following previous works (Campagna et al., 2020; Hu et al., 2022), we represented dialogue history as the accumulated dialogue state. The dialogue history is constructed by randomly selecting slot-value pairs from the dialogue schema[3] following the chosen system intent. The system and user dialogue acts are sampled based on the dialogue history and the selected system-user intent. Lastly, we sample

[3]Dialogue schema presents a structured representation of the valid slots and values across different domains

System Intent	User Intent
start	inform
inform	inform, update, reqmore, confirm, book
nooffer	update, recheck, end
select	pick, update, reqmore
recommend	select, update, reqmore
request	inform
booking-request	inform
booking-inform	book, nobook, update, reqmore, inform
offerbooked	new_domain, confirm, end
booking-book	new_domain, confirm, end
booking-nobook	new_domain, recheck, end

Table 1: Coherent system-user intents. For each system intent, we define the list of user intents that can indicate a natural next turn flow.

dialogue Act	Template
recommend (d, s, v)	I would suggest the <d> with <s> <v>
offerbooked (d, s, v)	Booked <d> for <v> <s>
request (d, s, v)	What is your preferred <d> <v> ?

Table 2: Selected system template responses

the current belief state values, considering the dialogue history and the user dialogue act. This approach guarantees the generation of coherent and contextually appropriate dialogue structure.

3.2 Template Response Generation

Prompting LLMs to generate free-flowing dialogues from the raw conversation structure offers limited control over its characteristics. As highlighted by Li et al. (2022) and Chen et al. (2023b), unconstrained generation based solely on the dialogue history or topic can produce erroneous dialogues, often necessitating human review and correction. On the other hand, prompting an LLM to modify a skeletal dialogue offers better control. Thus, drawing inspiration from Rastogi et al. (2020) and Kale and Rastogi (2020), we adopt the template-guided approach that enables fine-grained control over dialogue content. While Kale and Rastogi (2020) primarily offer templates for system response generation, their method entails crafting separate templates for each domain, dialogue act, and slot triplet. This results in more than 200 templates, demanding extensive human effort. Furthermore, these templates are not domain- and slot-agnostic, demanding effort with each new domain and schema modification. Building upon their methodology, we introduce domain-agnostic templates for both system and user responses.

Given a quartet of domain, dialogue act, slot, and value, respectively, we map it to a template that depends only on the dialogue act. Each template contains designated placeholder tokens for domain, slots, and values, which are substituted during template generation. This guarantees that the generated dialogues are grounded in the provided belief states. We utilize templates for just 22 dialogue acts (11 each for system and user), thus considerably reducing human efforts. We generate between 2 and 4 templates per dialogue act to encourage diversity. Also, our templates are domain-agnostic and can be scaled to newer domains without additional effort. Tables 2 and 3 illustrate some of our system and user templates, respectively.

dialogue Act	Template
inform (d, s, v)	The <d> <s> should be <v>
nobook (d, s, v)	No, don't book the <d> for <v> <s>
reqmore (d, s, v)	What is the <d>'s <s> ?

Table 3: Selected user template responses

3.3 LLM-based Template Modification

While the templates offer natural language descriptions for both system and user responses, they lack linguistic and conversational variations. Additionally, as these templates are designed to be domain and slot-value agnostic, they may contain certain grammatical and fluency errors. As a result, transforming these template-based responses into more naturalistic and free-flowing language can lead to contextually appropriate dialogues. Following previous research efforts in synthetic data generation (Mehri et al., 2022; Xiang et al., 2022; Li et al., 2022; Chen et al., 2023b), we employ GPT-3.5 (Brown et al., 2020) for converting the template responses to free-flowing dialogues. For this, we explore three distinct prompting strategies, detailed as follows.

We initially experimented with '*dialogue-level prompting*', instructing the LLM to modify the entire two-turn dialogue. This approach led to a hallucination of slot-value pairs and the generation of disfluent dialogues as the LLM often merged or interchanged information between user and system utterances. We also encountered instances where one of the system-user responses was skipped, generating a single utterance. We then explore a '*multi-step prompting*' approach, which employs a sequential prompting process. First, we prompt the LLM to refine the system template and then modify the user template independently by providing the modified system response. While this addresses the issue of skipped utterances, it still suffers from information blending between system and user responses, resulting in incorrect slot-value annotations and dialogues.

To overcome these drawbacks, we opt for '*utterance-level prompting*'. In this method, we refine the system and user template independently. This approach results in succinct responses strongly anchored in the template structure and consistent with the slot values. Importantly, it avoids the issue of information merging between system and user turns. We use this as our final prompting strategy. The prompt used is as follows:

> *Following is a template <user/system> response for a conversation between a <domain> chatbot and a user. Paraphrase the template by making it more fluent, engaging, polite, and coherent. Also, correct grammatical mistakes. Reorder the sentences if necessary.*
> *Strictly generate the response in the form of a JSON object {'<user/system>_paraphrased': ''} with correct formatting (including curly brackets). Do not return anything else apart from the JSON object.*
> *'<user/system>_template': '<template>'*

While the utterance modification using the above prompting scheme results in naturalistic conversations, we find that their stylistic diversity remains limited. Thus, to make the dataset more diverse, we use '*paraphrase prompting*', to generate different paraphrased dialogue variants. Similar to *utterance-level prompting*, we independently paraphrase both system and user responses to create the final dialogue. Our selection of prompts for paraphrasing is randomized from the following set:

> *Rephrase the sentences while retaining the original meaning.*
> *Use synonyms or related words to express the sentences with the same meaning.*
> *Use conversational language and paraphrase the following sentences.*
> *Generate a crisp and to the point single sentence from the given sentences using conversational language.*

4 Experimental Setup

4.1 Synthetic Data Generation

Using `SynthDST`, we create two types of synthetic corpora. In line with prior DST works (Wu et al., 2019; Hu et al., 2022), we generate data equivalent to 1%, 5%, and 10% of the training data size, ensuring a fair comparison with regard to number of samples in the retrieval bank. Each set contains 1) 50% of conversations featuring new slot-value pairs, 2) 15% of conversations with no new belief states introduced 3) 10% each of conversation starters and terminators, 4) 10% of conversations updating existing slots with new values, and 5) 5% involving the repetition or deletion of prior slot-value information. This ensures that the data bank follows a realistic distribution of conversations while encompassing diverse dialogue flows. Moreover, such careful data curation is known to stabilize ICL performance (Chang and Jia, 2023). Additionally, we generate sampling invariant versions of synthetic datasets, denoted as $\text{unique}_{\text{all}}$ and $\text{unique}_{\text{all5x}}$. The $\text{unique}_{\text{all}}$ dataset includes all valid unique dialogue flows, while$\text{unique}_{\text{all5x}}$ includes five instances of each unique dialogue flow. This results in a dataset of about 7k and 25k dialogues, respectively. Detailed information regarding these datasets can be found in Appendix A.1.

4.2 In-Context Learning Model

Our experiments are based on the IC-DST framework introduced by Hu et al. (2022). It reformulates DST as a text-to-SQL problem, using a tabular description of the ontology followed by relevant in-context examples in the LLM prompt. The IC-DST framework leverages the *text-davinci-codex* version (Chen et al., 2021) of OpenAI's GPT-3 model (Brown et al., 2020). It uses the cumulative dialogue state to represent conversation history. This design choice enhances efficiency, incorporates more in-context examples, and performs effectively in the presence of domain shifts. Additionally, IC-DST introduces a novel similarity score to retrieve better in-context examples. We encourage readers to refer to Hu et al. (2022) for a comprehensive understanding of the IC-DST framework.

We introduce specific modifications to the IC-DST framework, reducing its complexities and making it suitable for current versions of GPT

Percentage	Method	MultiWoZ 2.1						
		Attraction	**Hotel**	**Restaurant**	**Taxi**	**Train**	**JGA$_\mathrm{D}$**	**JGA$_\mathrm{A}$**
–	Zero Shot	$71.8_{0.0}$	$45.3_{0.2}$	$63.1_{0.8}$	$72.7_{0.6}$	$61.5_{0.8}$	$62.9_{0.3}$	$39.9_{0.3}$
	Few Shot$_\mathrm{random}$	$74.4_{0.2}$	$48.8_{2.9}$	$60.9_{5.3}$	$74.0_{0.7}$	$60.3_{2.1}$	$63.7_{1.9}$	$40.3_{2.4}$
–	Few Shot$_{\mathrm{unique}_\mathrm{all}}$	$72.0_{0.4}$	$51.2_{1.8}$	$65.3_{0.7}$	$75.5_{0.7}$	$69.0_{1.0}$	$66.6_{0.3}$	$45.3_{0.5}$
	Few Shot$_{\mathrm{unique}_{\mathrm{all}5\mathrm{x}}}$	$72.3_{0.6}$	$51.6_{0.7}$	$65.9_{1.3}$	$74.6_{1.0}$	$69.0_{0.6}$	$66.7_{0.1}$	$45.0_{0.1}$
1%	Few Shot$_\mathrm{SynthDST}$	$72.6_{0.4}$	$51.9_{0.4}$	$66.9_{0.6}$	$75.1_{0.1}$	$\mathbf{68.7_{1.6}}$	$67.1_{0.3}$	$\mathbf{45.8_{0.3}}$
	Few Shot$_\mathrm{train}$	$73.9_{0.4}$	$52.4_{0.6}$	$67.3_{1.0}$	$76.6_{0.5}$	$66.0_{0.9}$	$67.2_{0.3}$	$45.0_{0.4}$
5%	Few Shot$_\mathrm{SynthDST}$	$71.0_{0.9}$	$52.1_{1.3}$	$65.9_{1.5}$	$76.3_{0.5}$	$\mathbf{68.4_{0.3}}$	$66.7_{0.6}$	$44.9_{0.8}$
	Few Shot$_\mathrm{train}$	$74.3_{1.0}$	$54.2_{0.7}$	$69.0_{1.6}$	$78.6_{1.1}$	$66.7_{0.9}$	$68.6_{0.8}$	$46.2_{1.1}$
10%	Few Shot$^\dagger{}_\mathrm{SynthDST}$	$71.2_{0.9}$	$51.5_{0.6}$	$67.2_{1.5}$	$76.3_{0.4}$	$\mathbf{69.0_{0.3}}$	$67.1_{0.2}$	$45.4_{0.6}$
	Few Shot$_\mathrm{train}$	$74.2_{0.2}$	$53.8_{0.4}$	$69.1_{1.3}$	$78.3_{1.5}$	$66.4_{0.9}$	$68.3_{0.5}$	$46.1_{0.8}$
100%	Few Shot$^\dagger{}_\mathrm{train}$	$74.0_{0.1}$	$51.9_{0.3}$	$69.0_{0.4}$	$79.6_{0.4}$	$70.4_{0.8}$	$69.0_{0.0}$	$46.0_{0.1}$
$\Delta_{\mathrm{SynthDST}^\dagger-\mathrm{zeroshot}}$		$\downarrow 0.6$	$\uparrow 6.2$	$\uparrow 4.1$	$\uparrow 3.5$	$\uparrow 7.5$	$\uparrow 4.2$	$\uparrow 5.5$
$\Delta_{\mathrm{SynthDST}^\dagger-\mathrm{random}}$		$\downarrow 3.2$	$\uparrow 2.7$	$\uparrow 6.3$	$\uparrow 2.3$	$\uparrow 8.7$	$\uparrow 3.4$	$\uparrow 5.1$
$\Delta_{\mathrm{SynthDST}^\dagger/\mathrm{train}^\dagger}$		96.2	99.2	97.4	95.8	98.0	97.4	98.7
Percentage	**Method**	**MultiWoZ 2.4**						
		Attraction	**Hotel**	**Restaurant**	**Taxi**	**Train**	**JGA$_\mathrm{D}$**	**JGA$_\mathrm{A}$**
–	Zero Shot	$78.2_{0.2}$	$52.1_{0.1}$	$67.2_{0.7}$	$72.6_{0.5}$	$66.1_{0.1}$	$67.2_{0.2}$	$45.6_{0.3}$
	Few Shot$_\mathrm{random}$	$81.3_{0.6}$	$51.6_{4.3}$	$63.2_{6.1}$	$73.6_{0.4}$	$63.2_{2.5}$	$66.6_{1.9}$	$44.2_{2.6}$
–	Few Shot$_{\mathrm{unique}_\mathrm{all}}$	$79.1_{0.8}$	$56.8_{1.5}$	$66.9_{1.4}$	$76.4_{0.4}$	$73.2_{0.4}$	$70.5_{0.6}$	$50.4_{1.0}$
	Few Shot$_{\mathrm{unique}_{\mathrm{all}5\mathrm{x}}}$	$78.7_{0.2}$	$57.4_{0.9}$	$67.6_{0.5}$	$74.8_{0.1}$	$73.8_{0.8}$	$70.4_{0.2}$	$50.4_{0.4}$
1%	Few Shot$_\mathrm{SynthDST}$	$79.4_{0.5}$	$57.2_{0.8}$	$69.2_{0.3}$	$76.2_{0.4}$	$\mathbf{72.5_{1.6}}$	$70.9_{0.5}$	$51.0_{1.1}$
	Few Shot$_\mathrm{train}$	$81.4_{0.3}$	$58.8_{2.4}$	$72.1_{2.3}$	$77.1_{0.2}$	$70.2_{2.2}$	$71.9_{0.5}$	$52.1_{1.0}$
5%	Few Shot$_\mathrm{SynthDST}$	$79.1_{1.4}$	$56.8_{1.1}$	$69.8_{1.6}$	$77.1_{0.9}$	$\mathbf{72.4_{2.0}}$	$71.1_{1.2}$	$50.4_{1.8}$
	Few Shot$_\mathrm{train}$	$81.3_{0.6}$	$60.0_{0.4}$	$74.5_{0.9}$	$78.5_{0.6}$	$72.4_{1.9}$	$73.4_{0.5}$	$54.2_{1.0}$
10%	Few Shot$^\dagger{}_\mathrm{SynthDST}$	$77.9_{0.5}$	$57.6_{0.3}$	$69.9_{0.5}$	$77.1_{0.6}$	$\mathbf{73.2_{0.8}}$	$71.1_{0.2}$	$50.9_{0.3}$
	Few Shot$_\mathrm{train}$	$82.1_{0.9}$	$60.6_{0.8}$	$75.0_{0.8}$	$79.1_{0.9}$	$71.3_{1.2}$	$73.6_{0.3}$	$53.8_{0.9}$
100%	Few Shot$^\dagger{}_\mathrm{train}$	$84.0_{0.4}$	$60.0_{0.4}$	$75.9_{0.4}$	$81.3_{0.2}$	$74.7_{0.4}$	$75.2_{0.1}$	$55.2_{0.2}$
$\Delta_{\mathrm{SynthDST}^\dagger-\mathrm{zeroshot}}$		$\downarrow 0.3$	$\uparrow 5.5$	$\uparrow 2.7$	$\uparrow 4.5$	$\uparrow 7.1$	$\uparrow 3.9$	$\uparrow 5.3$
$\Delta_{\mathrm{SynthDST}^\dagger-\mathrm{random}}$		$\downarrow 3.4$	$\uparrow 6.0$	$\uparrow 6.7$	$\uparrow 3.5$	$\uparrow 10.1$	$\uparrow 4.5$	$\uparrow 6.7$
$\Delta_{\mathrm{SynthDST}^\dagger/\mathrm{train}^\dagger}$		92.7	96.0	92.1	94.8	98.0	94.5	92.2

Table 4: Comparison of per-domain Joint Goal Accuracy (JGA$_\mathrm{D}$) and all-domain Joint Goal Accuracy (JGA$_\mathrm{A}$) on MultiWoZ 2.1 and 2.4 using zero-shot, random few-shot, and retrieval-based few shot prompting with different percentages of synthetic and training data.

models. Firstly, due to the deprecation of the *text-davinci-codex*, we experiment with *gpt-3.5-turbo*, a newer chat model that exhibits similar coding capabilities. Secondly, the IC-DST framework uses explicit fine-tuning of the retriever on the training data. This process needs compute resources and time and presupposes access to training data. Consequently, we have adopted an off-the-shelf solution in the form of Sentence-BERT (Reimers and Gurevych, 2019), specifically the *all-mpnet-base-v2* model (Song et al., 2020). We keep the rest of the formulations unchanged.

4.3 Dataset

MultiWOZ 2.1 (Eric et al., 2020) is a multi-domain human-to-human dialogue dataset that contains over 10K dialogues across 8 domains. This is the updated version of the original MultiWOZ 2.0 dataset (Budzianowski et al., 2018). MultiWOZ 2.1 is a widely used benchmark for DST and in dialogue systems research.

MultiWOZ 2.4 (Ye et al., 2022) builds on top of the 2.1 version and makes substantial changes to the validation and test sets. MultiWOZ 2.4 can be viewed as a cleaner version of MultiWOZ 2.1 that better reflects model performance.

4.4 Evaluation Metrics

We employ the conventional Joint Goal Accuracy (JGA) as our evaluation metric. This metric considers a prediction correct when all slots-values match the ground truth. We report the *All-Domain Joint Goal Accuracy (JGA$_\mathrm{A}$)* for the overall performance and the *Per-domain Joint Goal Accuracy (JGA$_\mathrm{D}$)* for domain-level performance (Wu et al., 2019; Hu et al., 2022).

5 Results and Discussion

Table 4 presents our results for MultiWOZ 2.1 and 2.4. The zero-shot setting is the only baseline that does not rely on any human-annotated data, similar to our approach. We also report on

Method	Attraction	Hotel	Restaurant	Taxi	Train	JGA_D	JGA_A
Few Shot_T	$73.3_{0.7}$	$49.7_{0.2}$	$63.6_{1.4}$	$75.9_{0.5}$	$66.7_{0.6}$	$65.8_{0.3}$	$43.8_{0.2}$
Few Shot_LLM	$72.6_{0.4}$	$51.9_{0.4}$	$66.9_{0.6}$	$75.1_{0.1}$	$68.7_{1.6}$	$67.1_{0.3}$	$45.8_{0.3}$

Table 5: Ablation study of SynthDST. Few Shot_T and Few Shot_LLM refer to template and LLM-modified data, respectively.

a random setting, where we randomly add 2 examples per domain (resulting in 10 examples) from the training data to form a static set of in-context examples. Additionally, we assess the performance of synthetic and training data at different percentages as explained in section 4.1. For all setups, the average performance over 3 runs is reported.

Synthetic Data Consistently Beats Zero-Shot. Few-shot prompting using data generated from SynthDST and the zero-shot setups illustrate scenarios where no training data is used. This is particularly relevant to practical settings where obtaining human-labeled data can be prohibitively expensive in terms of cost and human effort. From table 4, observe that few-shot using data from SynthDST leads to substantial gains over zero-shot. Specifically, we observe about 4% and 5% improvements for JGA_D and JGA_A, respectively, across both the datasets. Moreover, it gives notably high gains on the two worst performing domains, about 6% on *hotel* and 7% on *train*. In summary, synthetic data may provide a good solution when no training data is available.

Retrieval-Based ICL with Synthetic Data Outperforms ICL with Random Training Examples. In some scenarios, ML practitioners may have access to a limited number of in-domain examples. Therefore, using a few static examples for few-shot learning is a relevant baseline. Table 4 reveals that utilizing randomly selected in-domain examples leads to similar or worse performance than the zero-shot setting. Notably, the performance drops significantly on *restaurant* and *train* domains. This observation aligns with previous findings (Liu et al., 2022), highlighting the high variance in results and emphasizing that random example selection is not an effective choice for ICL. SynthDST offers improvements of approximately $5-6\%$ on the JGA_D and JGA_A for both MultiWOZ versions across most domains. Interestingly, we notice substantial gains in the *attraction* domain. We conjecture that these gains can be attributed to the distribution in the test split. We discuss more on this in Appendix A.2.

SynthDST Competes Effectively with Training Data. Table 4 reports the performance on different percentage splits of training data. The results indicate that SynthDST consistently recovers over 95% and 92% of the training data performance on MultiWOZ 2.1 and 2.4 across all domains. Surprisingly, it even outperforms the 1% training data setup in MultiWOZ 2.1. Also, there are improvements of $1-3\%$ on the *train* domain for both versions. Moreover, it significantly reduces the performance gap, particularly in the *hotel* domain, which exhibited the poorest performance in the zero-shot setting.

Quality Trumps Quantity in Synthetic Data. In Section 4.1, we emphasize the importance of meticulously curating the ICL data pool for improved few-shot learning. From Table 4 it becomes evident that few-shot learning with unique_all and $\text{unique}_\text{all5x}$ data almost never surpasses the performance of the carefully curated data. Despite unique_all and $\text{unique}_\text{all5x}$ being approximately 14x and 47x times larger than the 1% data subset, respectively, it is clear that having a substantial representation of relevant examples is superior to having an equal representation of all examples. Moreover, less relevant examples can introduce noise and adversely affect predictions if the proportions of labels appearing in context differ greatly from the test instance (Zhao et al., 2021). Nevertheless, we still maintain a consistent improvement of over 5% compared to the zero-shot and random settings, underscoring the effectiveness of our synthetic data.

Template Data or LLM Modified Data? Table 5 presents an ablation study conducted on the 1% split of MultiWoZ 2.1. We observe that relying solely on template data yields improved performance in the *attraction* domain but significantly lower results in the *hotel*, *restaurant*, and *train* domains, resulting in an overall decrease in performance. Transitioning from templates to more naturalistic conversations leads to an approximate 2% improvement on JGA_D and JGA_A. There is also a noticeable improvement in the *restaurant*, *hotel*, and *train* domain. Comparing these findings with Table 4, we observe that relying solely on template data results in an improvement of nearly 4% in JGA_A. Therefore, even without LLMs, SynthDST offers

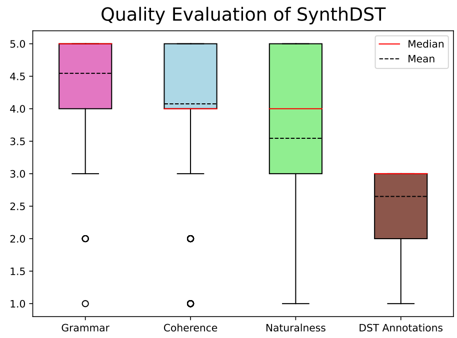

Figure 3: Box plot of Human evaluation scores.

significant gains over the zero-shot setting.

Synthetic Data Helps Unveil Annotation Bias. Inconsistent annotation has been a pervasive issue in DST datasets (Zang et al., 2020; Han et al., 2021; Ye et al., 2022). While MultiWoZ 2.4 presents a much cleaner version, our study uncovers a distinct concern unanswered previously: incongruities related to domain ontology. More precisely, our examination has revealed that the current annotations treat parking and internet slots labeled as 'yes' as synonymous with 'free.' However, these are two separate slot values in the schema and convey distinct meanings. To illustrate, when parking slots are marked as 'yes,' it generally indicates the availability of parking. Nevertheless, it does not necessarily imply that the parking is free; users might still be required to pay for parking despite the availability of slots.

6 Dataset Quality Analysis

Is the data generated by `SynthDST` of Good Quality? As `SynthDST` is a human-involvement-free synthetic data generation approach (except for template definition), assessing its quality is crucial. Consequently, we conducted a human evaluation on 200 dialogues from our 1% dataset split. Four evaluators, experienced in dialogue systems research, evaluate the data. Given the generated samples containing the dialogue history, average system utterance, average user utterance, and new dialogue state, the evaluators assessed the dialogues on four dimensions, namely, ***Grammar***, ***Coherence***, ***Naturalness***, and ***Annotations***. The annotations are rated from 1-3, whereas the others are graded on a 1-5 scale. The detailed scales are given in Appendix A.3.

In Figure 3, we present the results of our human evaluation. The majority of the dataset demonstrates high scores for ***Grammar***, indicating grammatical correctness and minimal mistakes. For ***Coherency***, both the mean and median scores exceed 4, signifying that the dialogues are mostly coherent and logically structured. While ***Naturalness*** exhibits slightly more variability, the mean, and median still surpass 4, indicating that most dialogues maintain a natural conversational flow resembling real-world conversations. Lastly, the ***Annotations*** scale attains a median of 3 and a mean > 2.5, suggesting that most of the annotations are correct.

Is `SynthDST` More Cost-Effective than Human Annotation? Creating the MultiWOZ dataset involved 1,249 workers and incurred a cost of approximately $\$30,000$, excluding post-processing expenses (Budzianowski et al., 2018; Li et al., 2022). In contrast, `SynthDST` significantly reduces both cost and time requirements. Specifically, `SynthDST` utilizes a total of 4 OpenAI API calls for each sample, 1 for modifying the system template into an utterance, 1 for modifying the user template into an utterance, then 1 for further paraphrasing the system utterance, and lastly for paraphrasing the user utterance. Table 6 presents the details of input-output tokens utilization and the total cost for each prompting step across different data splits. We see that `SynthDST` can generate an entire MultiWOZ-sized dataset ($\approx$ 55k dialogues) in just about $\$40$. Moreover, generating 1% equivalent data requires less than $\$1$ while maintaining the DST performance. Thus, `SynthDST` presents a cost-effective method to collect DST data.

7 Conclusion and Future Work

In this work, we present `SynthDST`, a synthetic data generation framework that leverages the dialogue schema to create coherent dialogues with DST annotations using a template-guided LLM-based approach. This framework enables the use of in-context learning for DST without human-annotated training data. Performance with `SynthDST` reaches close to the performance with training data on dialogue state tracking. This opens the possibility of supporting new domains without needing cumbersome and expensive training data collection. Moreover, it also reduces

Percentage	Utterance Modification				Utterance Paraphrasing				Cost
	Avg sys. inp. tok.	**Avg sys. out. tok.**	**Avg user inp. tok.**	**Avg user out. tok.**	**Avg sys. inp. tok.**	**Avg sys. out. tok.**	**Avg user inp. tok.**	**Avg user out. tok.**	**(USD)**
1% ($\approx$ 549 data)	120.46	28.93	114.02	25.63	41.09	30.15	37.98	26.90	\$0.38
5% ($\approx$ 2748 data)	119.54	27.95	114.27	25.78	40.23	29.52	37.83	26.46	\$1.88
10% ($\approx$ 5495 data)	119.95	28.23	114.14	25.91	40.37	29.41	38.06	26.54	\$3.78

Table 6: Cost Analysis of SynthDST in USD. Leveraging OpenAI's GPT-3.5-turbo, the expense is \$0.0010 per 1000 input tokens and \$0.0020 per 1000 output tokens. With these cost projections, generating a synthetic dataset equivalent in size to MultiWoZ ($\approx$ 55k examples) using SynthDST will cost less than \$40!

some annotation bias from these datasets.

Numerous potential avenues for future research emerge from our current work. While we experiment only with the MultiWOZ datasets, SynthDST can readily be extended to other corpora. While SynthDST predominantly relies on the close-sourced OpenAI GPT-3 model, it would be interesting to see how it performs with open-sourced LLMs. We encourage further research that validates its performance across diverse domains and models. Moreover, our approach does not incorporate safeguards to detect hallucinations in LLM-generated data, which is a direction for future investigations.

8 Limitations

We designed SynthDST as a domain-agnostic framework to enable scalability across different domains. However, this domain-agnostic approach comes with a trade-off – it struggles to capture inter-slot dependencies. For instance, when the slot *"attraction-type"* contains *"sports,"* it should ideally retrieve sports-related attractions for the *"attraction-name"* slot. Unfortunately, the current framework cannot accomplish this without compromising its domain-agnostic nature. Furthermore, SynthDST lacks a post-hoc human correction module, resulting in the retention of such potentially erroneous examples in the dataset. Nevertheless, such examples are few and far between, as indicated by the high human evaluation scores. Thus, it's important to emphasize that despite these challenges, SynthDST continues to deliver commendable performance.

9 Ethical Consideration

This work uses LLMs for synthetic data generation. It makes an effort to ensure grounded and consistent data is generated by the LLM, however there can still be hallucinations and/or inconsistencies in the predictions. It is highly recommended to implement further guardrails to use such data synthesis approaches in real world scenarios.

10 Acknowledgements

The authors would like to thank Jeffrey Nichols, Stephen Pulman, Barry Theobald, Nidhi Rajshree, and the anonymous reviewers for their help and feedback. Coo

References

Sanghwan Bae, Donghyun Kwak, Sungdong Kim, Donghoon Ham, Soyoung Kang, Sang-Woo Lee, and Woomyoung Park. 2022. Building a role specified open-domain dialogue system leveraging large-scale language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2128–2150, Seattle, United States. 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.

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.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 122–132, Online. Association for Computational Linguistics.

Ting-Yun Chang and Robin Jia. 2023. Data curation alone can stabilize in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8123–8144, Toronto, Canada. Association for Computational Linguistics.

Derek Chen, Kun Qian, and Zhou Yu. 2023a. Stabilized in-context learning with pre-trained language models for few shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1551–1564, Dubrovnik, Croatia. Association for Computational Linguistics.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7521–7528.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Maximillian Chen, Alexandros Papangelis, Chenyang Tao, Seokhwan Kim, Andy Rosenbaum, Yang Liu, Zhou Yu, and Dilek Hakkani-Tur. 2023b. PLACES: Prompting language models for social conversation synthesis. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 844–868, Dubrovnik, Croatia. Association for Computational Linguistics.

Mark G Core and James Allen. 1997. Coding dialogs with the damsl annotation scheme. In *AAAI fall symposium on communicative action in humans and machines*, volume 56, pages 28–35. Boston, MA.

Saket Dingliwal, Shuyang Gao, Sanchit Agarwal, Chien-Wei Lin, Tagyoung Chung, and Dilek Hakkani-Tur. 2021. Few shot dialogue state tracking using meta-learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1730–1739, Online. 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 Twelfth Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ting Han, Ximing Liu, Ryuichi Takanabu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. 2021. Multiwoz 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation. In *Natural Language Processing and Chinese Computing*, pages 206–218, Cham. Springer International Publishing.

Michael Heck, Nurul Lubis, Benjamin Ruppik, Renato Vukovic, Shutong Feng, Christian Geishauser, Hsien-chin Lin, Carel van Niekerk, and Milica Gasic. 2023. ChatGPT for zero-shot dialogue state tracking: A solution or an opportunity? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 936–950, Toronto, Canada. Association for Computational Linguistics.

Matthew Henderson. 2015. Machine learning for dialog state tracking: A review. In *Proceedings of The First International Workshop on Machine Learning in Spoken Language Processing*.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, 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.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2627–2643, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Léo Jacqmin, Lina M. Rojas Barahona, and Benoit Favre. 2022. "do you follow me?": A survey of recent approaches in dialogue state tracking. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 336–350, Edinburgh, UK. Association for Computational Linguistics.

Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6505–6520, Online. Association for Computational Linguistics.

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

Brendan King and Jeffrey Flanigan. 2023. Diverse retrieval-augmented in-context learning for dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5570–5585, Toronto, Canada. 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.

Zekun Li, Wenhu Chen, Shiyang Li, Hong Wang, Jing Qian, and Xifeng Yan. 2022. Controllable dialogue simulation with in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4330–4347, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021. Leveraging slot descriptions for zero-shot cross-domain dialogue StateTracking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5640–5648, Online. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Andrea Madotto, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2021. Few-shot bot: Prompt-based learning for dialogue systems. *arXiv preprint arXiv:2110.08118*.

Shikib Mehri, Yasemin Altun, and Maxine Eskenazi. 2022. LAD: Language models as data for zero-shot dialog. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 595–604, Edinburgh, UK. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Zhu Zhang, Jinchao Li, Chenguang Zhu, and Jianfeng Gao. 2021. Synergy: Building task bots at scale using symbolic knowledge and machine teaching. *arXiv preprint arXiv:2110.11514*.

Kun Qian, Ahmad Beirami, Zhouhan Lin, Ankita De, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. 2021. Annotation inconsistency and entity bias in MultiWOZ. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 326–337, Singapore and Online. Association for Computational Linguistics.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Jamin Shin, Hangyeol Yu, Hyeongdon Moon, Andrea Madotto, and Juneyoung Park. 2022. Dialogue summaries as dialogue states (DS2), template-guided summarization for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages

3824–3846, Dublin, Ireland. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

David R. Traum. 1999. *Speech Acts for Dialogue Agents*, pages 169–201. Springer Netherlands, Dordrecht.

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.

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

Jiannan Xiang, Zhengzhong Liu, Yucheng Zhou, Eric Xing, and Zhiting Hu. 2022. ASDOT: Any-shot data-to-text generation with pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1886–1899, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuting Yang, Wenqiang Lei, Pei Huang, Juan Cao, Jintao Li, and Tat-Seng Chua. 2023. A dual prompt learning framework for few-shot dialogue state tracking. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 1468–1477, New York, NY, USA. Association for Computing Machinery.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2022. MultiWOZ 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 351–360, Edinburgh, UK. Association for Computational Linguistics.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

A Appendix

A.1 Synthetic Data Generation

Table 7 contains the domain distribution of the different splits of SynthDST. For the 1%, 5%, and 10% percentage data, we uniformly sample each domain data according to the sampling scheme explained in Section 4.1. For the $\mathrm{synthetic}_1$ and $\mathrm{synthetic}_5$ datasets, we observe an uneven distribution of domains. This disparity arises due to our emphasis on acquiring unique system-user dialogue act pairs. Since each domain has a distinct number of dialogue acts, the distribution becomes skewed.

Data	Attraction	Hotel	Restaurant	Taxi	Train	Total
1%	106	111	116	105	111	549
5%	547	553	553	548	547	2748
10%	1093	1112	1109	1086	1095	5495
$\mathrm{synthetic}_1$	526	2968	1843	795	1536	7668
$\mathrm{synthetic}_5$	2142	9223	6146	2856	5422	25789

Table 7: Synthetic data distribution across domain.

A.2 Extended Discussion

Impact of Test Distribution on the Results. Figure 4 depicts the coarse and fine-grained distribution of the different domains in the MultiWOZ test set. The coarse-grained distribution suggests a relatively balanced representation of all domains, except for the *taxi* domain, which is less prominent. However, when examining the fine-grained distribution, a different picture emerges. Since MultiWOZ comprises multiple domains within a single dialogue, some domains overlap. In this fine-grained analysis, it becomes evident that the *attraction* domain, when considered in isolation, is the most underrepresented sub-category. However, it frequently appears in tandem with other domains such as *train* and *restaurant*. Therefore, we hypothesize that an increase in the performance of *train* and *restaurant* results in a decrease in *attraction*. This hypothesis is substantiated by the results presented in Table 4. Specifically, the scores for the *attraction* domain demonstrate an increase, while the*train* and *restaurant* domains experience a decrease in performance (as evidenced by the Few $\mathrm{shot}_{\mathrm{random}}$). Similarly, the opposite is observed for Few $\mathrm{Shot}_{\mathrm{synthetic}}$.

Impact of Off-The-Shelf Retriever. Unlike other ICL approaches, we refrain from fine-tuning the retriever to mimic a no-training data scenario. As illustrated by the results in Table 4, the performance demonstrates little correlation with the expansion of the retrieval pool. Furthermore, there are instances where the performance actually decreases, notably in the 1% $\rightarrow$ 5% setup for synthetic data and the 5% $\rightarrow$ 10% setup for training data across both datasets. We postulate that this might be attributed to off-the-shelf retrievers occasionally retrieving irrelevant examples since they lack awareness of the semantics of the end-task data. In summary, our results attest that we can achieve good performance with a small data set with off-the-shelf retrievers.

A.3 Human Evaluation

Metric	Scale
Grammar	1 = Highly Incoherent or Unintelligible 2 = Poorly Constructed and Difficult to Understand 3 = Moderately Fluent, but Some Awkwardness 4 = Mostly Fluent and Easily Understandable 5 = Extremely Fluent and Natural
Coherence	1 = Responses Lack Logical Flow and Are Highly Disjointed 2 = Poor Logical Flow, and Responses Often Do Not Connect 3 = Responses Have Some Logical Flow but Lack Consistency 4 = Logical Flow Is Mostly Maintained with Few Disruptions 5 = Highly Coherent and Smooth Logical Flow
Naturalness	1 = Very Robotic and Unnatural, Clearly Generated 2 = Lack of Natural Language Patterns, Not Believable 3 = Moderately Natural, but Still Exhibits Robot-Like Phrasing 4 = Fairly Natural and Believable in a Conversational Context 5 = Extremely Natural and Difficult to Distinguish from Human Speech
Annotations	1 = Completely Incorrect 2 = Partially correct, covering most of the slot value pairs 3 = Exactly correct, covering all the possible slot value pairs

Table 8: Human Evaluation Scale.

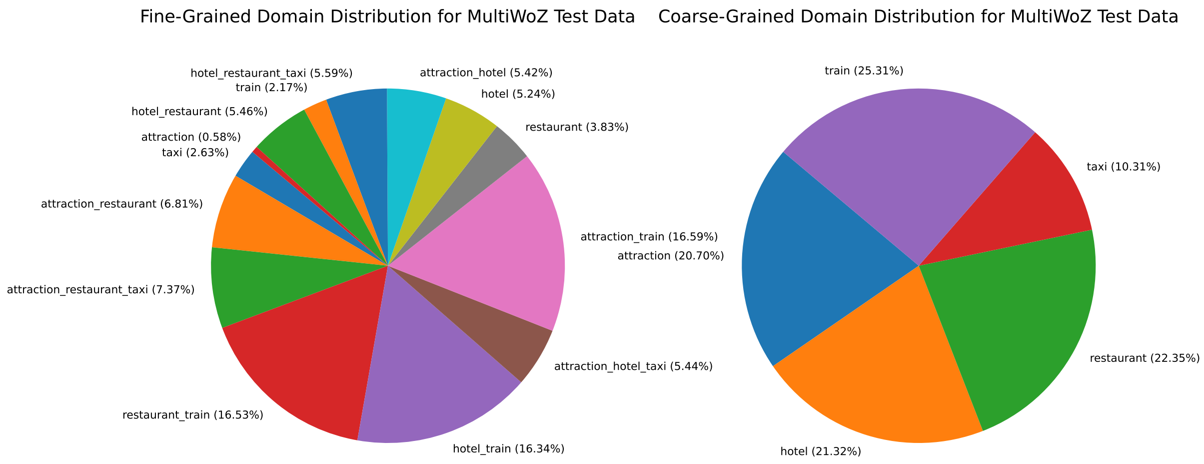

Figure 4: Domain distribution for MultiWoZ test data.