Zero-shot Generalization in Dialog State Tracking through Generative Question Answering

Shuyang Li,[1, 2] **Jin Cao,**[1] **Mukund Sridhar,**[1] **Henghui Zhu,**[3] **Shang-Wen Li,**[3]
Wael Hamza,[1] **Julian McAuley**[2]
[1]Amazon Alexa AI, [2]UC San Diego, [3]AWS AI
{shl008, jmcauley}@eng.ucsd.edu
{jincao, harakere, henghui, shangwel, waelhamz}@amazon.com

Abstract

Dialog State Tracking (DST), an integral part of modern dialog systems, aims to track user preferences and constraints (slots) in task-oriented dialogs. In real-world settings with constantly changing services, DST systems must generalize to new domains and unseen slot types. Existing methods for DST do not generalize well to new slot names and many require known ontologies of slot types and values for inference. We introduce a novel ontology-free framework that supports natural language queries for unseen constraints and slots in multi-domain task-oriented dialogs. Our approach is based on generative question-answering using a conditional language model pre-trained on substantive English sentences. Our model improves joint goal accuracy in zero-shot domain adaptation settings by up to 9% (absolute) over the previous state-of-the-art on the MultiWOZ 2.1 dataset.

1 Introduction

Dialog agents are gaining increasing prominence in daily life. These systems aim to assist users via natural language conversations, taking the form of digital assistants who help accomplish everyday tasks by interfacing with connected devices and services. A key component to understanding and enabling these task-oriented dialogs is Dialog State Tracking (DST): extracting user intent and goals from conversations via filling in belief slots (Lemon et al., 2006; Wang and Lemon, 2013). Assistive and recommendation use-cases for dialog agents in production settings are particularly challenging due to constantly changing services and applications with which they interface.

Traditional DST systems have achieved high accuracy when presented with a known ontology of slot types and valid values (Chen et al., 2020). In a real-world setting, however, a DST model must generalize to new slot *values* (e.g. new entities that are not present at training time) and new slot *types* (e.g. requirements regarding a new application). Recent work has sought to address these issues by posing DST as a reading comprehension or question answering (QA) task (Gao et al., 2019)—such models predict each slot value independently at any given turn and can theoretically be queried for new slots at inference time.

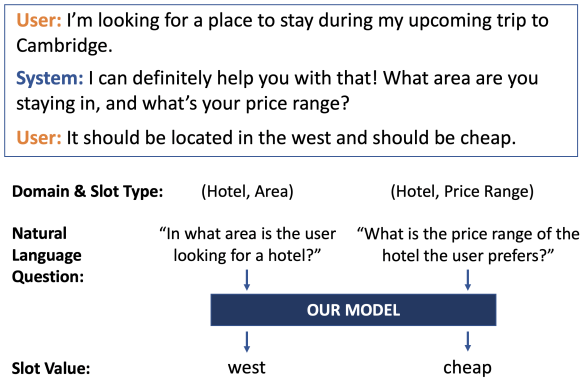

Figure 1: Based on a dialog history, a natural language questions are provided to our model to query a user's requirements and preferences (dialog state).

Some approaches toward DST as QA learn embedding vectors for each slot and/or domain word (Wu et al., 2019), but this is not robust to unseen slots whose specific names (e.g. 'Internet Access') may be totally unlike those in the training set. Gao et al. (2020) attempt to remedy this by posing a natural language question for each slot, but their hybrid span-extraction and classification-based system nonetheless requires access to the full ontology for unknown domains. We present an ontology-free model using natural language questions to represent slots that builds on conditional language modeling techniques—taking advantage of the rise of powerful generative language models (Radford et al., 2019)—to tackle DST as a *generative* QA

task. Our model can generalize to unseen domains, slot types, and values, and allows developers to query for arbitrary user requirements via simple questions. To summarize our main contributions:

- We propose an ontology-free conditional language modeling framework for dialog state tracking via generative question answering, achieving state-of-the-art performance in zero-shot domain adaptation settings for DST on MultiWOZ 2.1 (Eric et al., 2020) across all domains with average per-domain gains of 5.9% joint accuracy over previous best methods;
- We demonstrate performance competitive with state-of-the-art methods in a fully supervised setting;
- We show that our approach can be easily adapted to predict slot carry-over and transfer knowledge from a larger, more diverse dataset (Kim et al., 2019), improving zero-shot DST performance across all domains to 11% joint accuracy over the state-of-the-art.

2 Approach

We follow Gao et al. (2019) in treating Dialog State Tracking as a reading comprehension problem: at each turn of dialog, our model reads the dialog history and answers a fixed set of queries about user requirements and preferences (slots), with predictions aggregated to form the belief state. In our framework (Figure 1), we query for a given slot (e.g. Hotel Price Range) by asking a natural language question (Gao et al., 2020)—"What is the price range of the hotel the user prefers?". As our model's predictive ability is based on its general understanding of language and task-oriented conversation, we support zero-shot inference without the need to re-train the model or extend a formal ontology. For example, if a model has not been trained on data from the hotel domain, when presented with a hotel booking conversation we may nonetheless ask it a question like "In what area is the user looking for a hotel?" and received a prediction for that unseen requirement (Hotel Area).

While we conduct our experiments on English-language DST datasets, our approach is applicable to state tracking in any language, provided a conversation history is available.

Problem Statement We consider a conversation with T turns of user u_t and system utterances y_t:

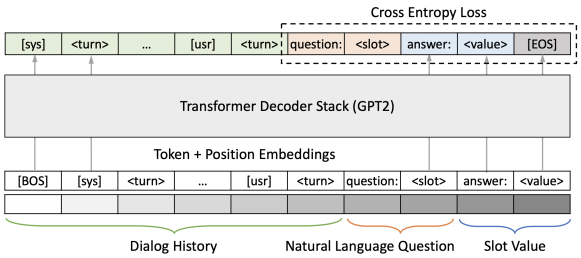

Figure 2: Our model performs DST via generative question-answering. Natural language questions for dialog slots allow our model to generalize to new slot types through its understanding of general language.

$C = \{y_1, u_1, \ldots y_T, u_T\}$. The belief state B_t at turn t comprises many tuples of slots $s \in S$ and their associated values $v_{s,t} \in V_s$, extracted from the conversation history $C_t = \{y_1, u_1, \ldots, y_t, u_t\}$. The set of possible values V_s can be arbitrarily large (e.g. possible hotel names), so we represent these values as sequences of vocabulary tokens $v_{s,t} = \{w_1, w_2, \ldots, w_k\}, w_i \in \mathcal{W}$. At inference time we pose a natural language question $s = \{w_1, \ldots, w_n\}$ and our model predicts an answer (slot value $v_{s,t}$) based on its understanding of the dialog history C_t. To predict the belief state B_t, our model independently answers $|S|$ different questions (Figure 1). In zero-shot DST, the system must predict values for slots outside of the initial ontology—these slot queries correspond to arbitrary natural language questions s' about entities and relationships in the conversation C_t.

Generalizing to New Domains and Slots Dialog State Tracking systems in real-world settings must scale to new users and services, accommodating new slot values (e.g. a new movie release) as well as new domains and slot types (e.g. a service update, or a new connected API). Existing methods require the developer to either write a complete ontology of slots and allowed values or modify their model architecture to add slot-specific prediction heads (Chen et al., 2020). Span-based approaches (Zhang et al., 2019; Zhou and Small, 2019) can correctly predict values that appear verbatim in a conversation but fail when a user paraphrases or mis-phrases a value. They also fall back to treating open-valued slots as classification problems (Zhang et al., 2019; Gao et al., 2020). We approach DST as an ontology-free generative question answering task, as generative methods (Wu et al., 2019; Kumar et al., 2020) have shown promise in few-shot and supervised DST settings.

	JGA (%)	# Params
DistilGPT2 LM	36.35	82 M
DistilGPT2 CLM no PT	39.34	
DistilGPT2 CLM	49.55	
+Question (CLMQ)	50.83	
GPT2 CLMQ	51.02	124 M
GPT2-medium CLMQ	**52.58**	355 M

Table 1: Ablation study of our framework, reporting supervised JGA on the MultiWOZ 2.1 test set.

	MultiWOZ	DSTC8
Train	7,906	16,142
Validation	1,000	2,482
Test	1,000	4,201
Domains	5	19
Slots	30	124
Open	9	59
Numeric	5	12
Temporal	5	10
Categorical	11	43

Table 2: Dataset statistics for MultiWOZ 2.1 and DSTC8: number of dialogs in each split, number of domains, and slots with slot category breakdowns.

While some approaches toward DST as QA learn a set of embeddings for each slot and/or domain (Gao et al., 2019; Wu et al., 2019; Kumar et al., 2020), this is not robust to unseen slots. We encode slots as natural language questions—manually formulating one question per slot—allowing us to share a pre-trained encoder for both dialog context and slot to leverage shared linguistic knowledge (Gao et al., 2020). Thus, our model is also agnostic to ontologies and can answer arbitrary English questions about the dialog history. We treat DST via QA as a conditional language modeling task, and train our model to predict the conditional likelihood of question (slot s) and answer (value $v_{s,t}$) tokens given a dialog context C_t at a given turn t:

$$P(v_{s,t}, s|C_t) = P(v_{s,t}|s, C_t) * P(s|C_t)$$

At inference time, the model is given the dialog context alongside a question—$[C_t; s]$—and asked to predict the value $v_{s,t}$ for that slot.

3 Model Architecture

For our conditional language model, we compared two common architectures: 1) an encoder-decoder model (Sutskever et al., 2014) with a bi-directional encoder; and 2) a purely auto-regressive decoder-only model. We conducted preliminary experiments using both a Transformer (Vaswani et al., 2017) encoder-decoder language model pre-trained using a de-noising auto-encoder objective (Lewis et al., 2020), as well as a Transformer decoder pre-trained with next-token prediction on English web pages. We achieved 1% better supervised DST performance with the decoder-only model in half the training time. Our model architecture thus comprises a Transformer decoder language model that allows us to leverage pre-trained language models like GPT2 (Radford et al., 2019) and common-sense world knowledge accrued through pre-training (Petroni et al., 2019).

We use a BPE (Sennrich et al., 2016) tokenizer to convert input text into a sequence of tokens. These are embedded in $\mathbb{R}^h$ and added to an $\mathbb{R}^h$ sinusoidal positional embedding. This input embedding is processed by l Transformer layers with hidden dimensionality h, each of which applies multi-headed attention with k heads followed by a feed-forward layer with a softmax nonlinearity. The final output hidden states are then projected into our vocabulary space of 50,257 sub-word tokens. We initialize our model weights with DistilGPT2 (Sanh et al., 2019), GPT2 (Radford et al., 2019), or GPT2-medium with $h = 768, 768, 1024$, $l = 6, 12, 24$, and $k = 12, 12, 16$ respectively.

As seen in Figure 2, our input sequence consists of a concatenation of dialog context C_t, slot query s, and slot value $v_{s,t}$: $[C_t; s; v_{s,t}]$. We pre-pend each utterance with a speaker token `[usr]` or `[sys]` for a user or system speaker to allow our model to identify additional context about each utterance. We pre-pend the slot query and value with `question:` and `answer:` respectively to distinguish slot queries from user-posed questions in the conversation. At training time, we calculate a cross-entropy loss similar to encoder-decoder models by maximizing the log likelihood of the slot query and value conditioned on the dialog context:

$$P(s, v_{s,t}|C_t) = \prod_{i}^{n} P(x_i|x_{<i}, C_t)$$

where $n = |[s; v_{s,t}]|$. We find through ablation experiments on our architecture that this loss computation method out-performs a naïve language-modeling approach that maximizes log likelihood of the full concatenated sequence $[C_t; s; v_{s,t}]$ via the factorized joint distribution (Peng et al., 2020;

Model	Type	JGA	NLQ
TRADE (Wu et al., 2019)	G	45.60	
SUMBT (Lee et al., 2019)*	C	46.70	
STARC (Gao et al., 2020)*	C+S	49.48	Y
MA-DST (Kumar et al., 2020)	G	51.88	
GPT2-m CLMQ	G	**52.58**	Y

Table 3: Supervised DST performance on MultiWOZ 2.1 of our model (underlined) compared to prior methods capable of zero-shot inference. Models using natural language questions (NLQ) are marked. **Requires access to slot-value ontologies at inference time.*

Model	Type	JGA	Extra Supervision
DSTQA	C+S	51.17	Knowledge Graph
DS-DST	C+S	51.21	
GPT2-m CLMQ	G	52.58	
SOM-DST	G	53.68	Previous Dialog State
SST	C	55.23	Schema
TripPy	S	55.30	Previous Dialog Actions
SimpleToD	G	**55.72**	Actions (Training)

Table 4: Supervised DST performance on MultiWOZ 2.1 of our model (underlined) against state-of-the-art DST methods incapable of zero-shot inference.

Hosseini-Asl et al., 2020):

$$P(x) = \prod_{i}^{n} P(x_i|x_{<i})$$

This allows for flexibility in learned representations for dialog context while regularizing slot query hidden states.

4 Data

We perform our experiments on **MultiWOZ** (Budzianowski et al., 2018), which contains over 10K single- and multi-domain task-oriented dialogs written by crowd-workers. We use the 2.1 version, with corrected and standardized annotations from Eric et al. (2020). We follow Wu et al. (2019) in lower-casing all dialogs and removing dialogs from training-only domains (Police and Hospital). The final dataset contains 9,906 conversations from 5 domains (Restaurant, Hotel, Attraction, Train, Taxi) covering 30 domain-slot pairs. Each dialog contains an average of 7 user and system turns.

We also experiment with augmenting our training dataset in zero-shot settings with observations drawn from the DSTC8 (Kim et al., 2019) dataset, [1] which contains 16,152 dialogs from 45 domains. DSTC8 was created via template-based dialog models provided with service APIs, and then edited by crowd-workers (Shah et al., 2018). We normalize domains and slots corresponding to the same domain (e.g. `Bus_1, Bus_2`) for a total of 19 domains and 124 slot types in DSTC8. We further manually annotate each dataset with slot value types: open-valued (e.g. Hotel Name), numeric (e.g. Restaurant Guests), temporal (e.g. Taxi LeaveAt), and categorical (e.g. Attraction Type). Dataset statistics are shown in Table 2.

[1] https://github.com/google-research-datasets/dstc8-schema-guided-dialogue

5 Experiments

We measure DST performance via Joint Goal Accuracy (JGA): the proportion of turns with all belief slots predicted correctly, including those not present. In Section 5.1, we evaluate our model on fully *supervised* DST, in which all domains and slots are known at training time. In Section 5.2, we investigate *zero-shot* domain adaptation in which the model is evaluated on conversations from an unseen domain with previously unseen slots. We then explore how our framework seamlessly accommodates teaching a model to predict slot carryover (Section 5.3) and transfer learning with significantly more diverse domains and slot types (Section 5.4). To measure zero-shot JGA, we follow Campagna et al. (2020) and only consider slots specific to the held-out domain. We focus our analysis on the zero-shot setting, as our goal is to build DST systems that can easily and effectively generalize to new domains and services. We train all models to convergence with a maximum of 10 epochs on Nvidia V100 GPUs, using the Lamb optimizer (You et al., 2020) with a base learning rate of 2e-5. All predictions are made using greedy decoding.

5.1 Supervised DST

We first evaluate on the commonly benchmarked supervised DST task to demonstrate performance competitive with state-of-the-art. In this setting we compare our approach against prior methods capable of zero-shot inference in Table 3—TRADE, STARC, SUMBT, and MA-DST—and those incapable of doing so in Table 4, including DSTQA (Zhou and Small, 2019), DS-DST (Zhang et al., 2019), SOM-DST (Kim et al., 2020), SST (Chen et al., 2020), TripPy (Heck et al., 2020), and SimpleToD (Hosseini-Asl et al., 2020). Our model outperforms all prior models that support zero-shot generalization and is competitive with meth-

	Rest.	Hot.	Attr.	Train	Taxi
TRADE	12.59	14.20	20.06	22.39	59.21
MA-DST	13.56	16.28	22.46	22.76	59.27
SUMBT	16.50	19.80	22.60	22.50	59.50
Ours (GPT2)	21.05	18.54	23.67	24.34	59.10
Ours (GPT2-m)	**26.17**	**24.41**	**31.31**	**29.07**	**59.61**

Table 5: Zero-shot domain adaptation JGA (%) on MultiWOZ 2.1 test set for recent works and our models with question loss, on the (Rest)aurant, (Hot)el, (Attr)action, Train, and Taxi domains. Previous state-of-the-art results are underlined, with new best **bolded**.

ods that focus solely on supervised DST—most of which require extra supervision at training and inference time, including dialog actions and prior dialog states. We distinguish models by their prediction type as (C)lassification-, (S)pan extraction-, and (G)eneration-based methods.

As seen in Table 1, our formulation of DST as a generative QA task benefits significantly from the usage of a conditional decoder-style model. A standard auto-regressive language modeling formulation (**LM**) with loss computed over the entire input sequence achieves 13% lower JGA compared to computing cross entropy loss only over slot value tokens (**CLM**). Pre-training is also crucial—we see a 10-point drop in JGA when randomly initializing model weights (**no PT**) compared to initializing from pre-trained DistilGPT2 weights. We also compare two other sizes of our models: GPT2-based—comparable in size to SUMBT's (Lee et al., 2019) 112M parameters—and GPT2-medium-based—comparable in size to STARC's (Gao et al., 2020) 355M parameters. We find that scaling the size of our model results in modest improvements in supervised JGA. We hypothesize that extending our loss to cover both slot query and value tokens (**+Question/CLMQ**) helps regularize the hidden representations of question tokens, and we achieve a 1.3% improvement in JGA.

5.2 Zero-Shot DST

Our primary focus lies in the zero-shot domain adaptation setting, where conversations and target slots at inference time come from unseen domains. We use a leave-one-out setup, training our models on four domains from MultiWOZ and evaluating on the held-out domain. Our model must understand a wide variety of possible questions about unseen conversations to generalize well. We compare our model against strong baseline models for zero-shot DST: TRADE, SUMBT, and MA-DST; Table 5 contains results from our models alongside baseline results reported by Kumar et al. (2020) and Campagna et al. (2020). These models represent slots as domain-slot tuples: TRADE learns a separate embedding for each domain and word in slot names, while SUMBT and MA-DST encode domain-slot tuples via BERT (Devlin et al., 2019) and an RNN encoder, respectively.

Our GPT2-medium based model achieves state-of-the-art zero-shot performance on all five domains, and by a significant (5-10%) margin on the Restaurant, Hotel, Attraction, and Train domains. While increased model size modestly impacts supervised DST performance (Table 1), larger models perform significantly better in a zero-shot setting with average absolute gains of 4.8% and relative gains of 22% in JGA across domains. Such improvements are consistent with findings from Brown et al. (2020) that up-sizing language models improves zero-shot performance across various tasks and Petroni et al. (2019), who observe that larger pre-trained models can retain more commonsense and world knowledge from their pre-training corpus—which may help our model understand queries for unseen domains and slots.

Effect of Natural Language Questions Prior work that frames DST as QA typically represents the slot query as a concatenation (tuple) of domain and slot name. Zhang et al. (2019) explore the impact of three different slot representations—domain-slot tuples, short slot descriptions, and full questions—on a hybrid classification-extraction model for DST, and find little difference in performance. However, we find that full questions work much better than domain-slot tuples for our generative framework, especially in zero-shot DST. We hypothesize that natural language questions—structurally similar to dialog utterances and pre-training sentences—allow our model to best leverage its linguistic knowledge with minimal friction when jointly encoding the dialog history, slot query, and slot value.

Wu et al. (2019) find that zero-shot generalization in models that represent slots as tuples is primarily due to shared slot names between domains (e.g. Taxi and Train 'leaveAt'). In a real-world setting a newly added dialog service is unlikely to share slot names verbatim with existing services. To fairly compare tuples and natural language questions under our framework, we per-

USER	My friend told me about Carolina Bed and Breakfast. Do you know anything about it?
SYS	It's a 4 star guesthouse. What would you like to know about it?
USER	Can you give me the postcode? And, do they have internet?
SYS	The postcode is cb13nx; they have internet.
USER	Thanks. Any boat attractions in the west?
SYS	**Nothing in west. Closest boat is the Cambridge Punter in centre. Too far?**
USER	**Yes, it is. How about a museum?**

Error Modality	Slot	Gold	Prediction	Open	Numeric	Temporal	Categorical
Spurious	(Attraction, Name)	`n/a`	cambridge punter	8.4 %	22.3 %	47.7 %	16.0 %
Ignored	(Hotel, Internet)	yes	`n/a`	65.3 %	53.5 %	19.9 %	76.8 %
Wrong Value	(Attraction, Type)	museum	boat	26.3 %	24.2 %	32.4 %	7.2 %

Table 6: Example of different classes of DST errors, and the proportion of errors they make up across the four slot categories for all five domains in a zero-shot setting. Latest (target) turn is **bolded**.

form zero-shot experiments using each representation. For tuple-based questions, our model takes as slot query a synonym of the slot name (e.g. Taxi 'leaveAt' → 'Pick Up Time') instead of a full question (e.g. 'What time does the user want the taxi to pick them up¿). Full question models achieved 6% higher per-domain JGA compared to slot-tuple models, supporting the notion that slot-tuple models memorize slot names rather than understanding their meaning and thus do not generalize well in real-world settings. Using full questions, our model (Table 5) achieves state-of-the-art performance in zero-shot settings.

Error Modalities To analyze our model, we follow Gao et al. (2020) and categorize DST errors in three modalities: 1) the model predicts a *spurious* value for an irrelevant slot; 2) the model *ignores* a relevant slot; and 3) the model correctly infers the presence of a slot but predicts a *wrong value*. Table 6 shows examples of each type of error for a sample conversation, and what proportion of errors they make up in each slot category for our **GPT2-m CLMQ** model in a zero-shot setting. Temporal slots are least likely to be *ignored* by our model, as verbatim `HH:MM` values are easily identifiable in a conversation. However, it is difficult to distinguish between closely related unseen temporal slots like 'leaveAt' and 'arriveBy'. Values for categorical, numeric, and open-valued slots on the other hand can comprise common (non-slot) phrases used in conversation, and thus it is easy for our model to ignore such slot references.

We also examine the source of dialog slots: *users* explicitly express the majority (79.5%) of slot values, while a minority are either derived via user reactions to *system* suggestions (9.7%) or *implicitly* valued (10.8%)—not present verbatim in a conversation. However, our errors are distributed evenly between *user*, *system*, and *implicit* sourced slots—suggesting that it is challenging for our model to track dialog states that are updated reactively via user feedback. We thus see a future opportunity to improve DST models by emphasizing multi-hop reasoning and common-sense inference.

5.3 Predicting Carried Over Slots

Long-range dependencies and slot values carried over from early turns are particularly important to model for accurate DST in long conversations (Kumar et al., 2020). We observe this in the zero-shot setting: our model is able to predict all slots accurately for 61% of conversation first-turns, dropping to 46% after one turn, and 5.7% after seven turns (the average conversation duration). We implement an oracle module to discard predictions when a dialog state does not need updating, obtaining an upper bound for DST improvements due to carry-over prediction. With this oracle, we see an average 5-point improvement in JGA across domain, indicating that carry-over prediction can greatly benefit our model. State-of-the-art models for fully supervised DST often rely on explicitly processing previous dialog states—via slot-value graphs (Zhou and Small, 2019; Chen et al., 2020) or as a separate input to the model at each turn (Heck et al., 2020; Kim et al., 2020). In our framework we can target slot carry-over by training a model to predict a `carried over` token in place of the true slot value whenever a slot value does not need updating at the current turn (**+ Carryover**). At inference time, we replace predicted carry-over tokens with the slot's last predicted value.

Our carry-over implementation improved JGA for all domains (Table 7) by an average of 3.14%, and improved JGA across all context lengths—with the largest improvements (+7%) at the sec-

	Rest.	Hotel	Attr.	Train	Taxi
Previous SOTA	16.50	19.80	22.60	22.76	59.50
GPT2 CLMQ	21.05	18.54	23.67	24.34	59.10
+ Carryover	24.00	19.91	28.45	30.75	59.29
+ DSTC8	24.65	22.94	34.30	38.55	59.68
GPT2-m CLMQ	26.17	24.41	31.31	29.07	59.68
+ CO, DSTC8	**27.69**	**24.88**	**42.39**	**41.05**	**60.32**

Table 7: Zero-shot JGA on MultiWOZ 2.1 test set with carry-over prediction and transfer learning.

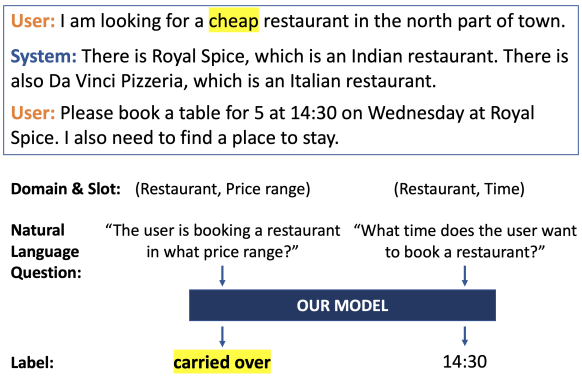

Figure 3: Our model can be trained to predict the presence of slot carry-over by replacing slot values from previous turns with the `carried over` token.

ond and third turn of a conversation (Figure 4). The `carried over` token allows our model to hedge against low confidence slots, falling back to predictions from previous turns where the target slot may be directly mentioned. This helps reduce the *wrong value* error rate by an average of 31% across each domain. Our model can also propagate null values with carry-over, reducing spurious predictions by an average of 36% across domains. However, we also observe our carry-over model propagating 78% of its errors from previous turns, suggesting that further improvements can result via accurately predicting slot updates.

5.4 Transfer Learning for Generalization

Our framework is ontology-agnostic and thus easily supports transfer learning without modifying the architecture by simply writing natural language questions for additional slots. Gao et al. (2020) found that intermediate fine-tuning of RoBERTa-Large (Liu et al., 2019) on passage-based QA tasks (Fisch et al., 2019) improved zero-shot DST performance. In preliminary experiments, we found no significant impact from intermediate fine-tuning on the SQuAD v2.0 (Rajpurkar et al., 2018) passage-based QA dataset. However, we observe significant improvements when training with joint, non-curriculum learning (McCann et al., 2018; Raffel et al., 2019)—augmenting our training data with an equal number of examples sampled from DSTC8, taking care to remove data from the held-out domain in both MultiWOZ and DSTC8.

Our framework allows for easy joint optimization with carry-over and transfer learning: by training new models on MultiWOZ 2.1 augmented with DSTC8 (**+ DSTC8**) we gain a further average 3.5-point improvement in per-domain JGA (Table 7). On average, our model makes 29% fewer *spurious* errors, and 6.9% fewer errors in open-valued slots, suggesting that our model scales well with additional training data with semantically distinct slot types and values. Our model also makes 9.7% fewer errors on categorical slots and 63% fewer mistakes where it assigns the value of one categorical slot to another, despite being unable to observe the set of possible categorical options—suggesting that exposure to more diverse categorical slots allows our model to better understand and distinguish between such slots. While temporal slots comprise only 17% of MultiWOZ and 10% of DSTC8 slots, these additional examples seem to help our model better disambiguate temporal references and make 32% fewer errors in such slots.

By applying both carry-over and transfer learning to our largest model, we observe further improvements in zero-shot JGA for all domains—averaging 5.1 points better than GPT2-m CLMQ, for an average gain of 11% JGA over previous state-of-the-art across domains (Table 7).

6 Qualitative Analysis

We manually reviewed 300 errors made by our GPT2-medium CLMQ model in the zero-shot setting—annotating 20 errors from each modality (spurious, ignored, wrong value) from each domain with the gold label quality and perceived cause of error totaling 300 annotated examples. As widely observed in recent DST work (Zhou and Small, 2019; Kumar et al., 2020), a significant proportion of DST errors on MultiWOZ are unavoidable—caused by annotation errors. While version 2.1 corrected some of these, annotation errors and inconsistencies remain responsible for 30% of sampled errors—in particular, in 10% of errors the original annotator did not record reactive preferences while in 5% of errors the original annotator did. These inconsistencies can hurt our model's ability to infer

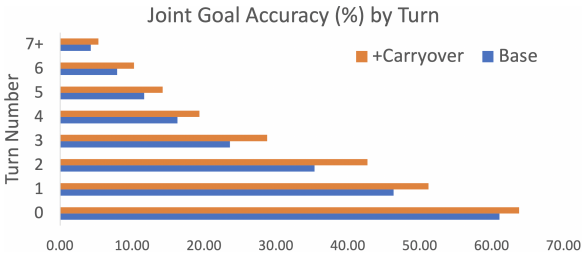

Figure 4: Per-turn JGA on MultiWOZ Test set for GPT2-CLMQ with and without carry-over prediction.

reactive and implied requirements and preferences.

We are also particularly interested in *slot transfers*—when our model mistakenly predicts one slot's value for a different slot, comprising 36% of our manually reviewed errors. In the Taxi and Hotel domains, our model transfers slots from the same domain over 75% of the time, with most swaps occurs between same-category slots (e.g. temporal slots like Taxi 'LeaveAt' and 'ArriveBy'). Slots in these domains are closely semantically related, with values that can fit any slot of that category (e.g. 13:10 vs. 15:15). While a human can easily infer that the earlier of two times must be departure and the later arrival, our model has no inherent understanding of temporal mechanics or numeracy (Wallace et al., 2019). In future work, we will explore learning such knowledge directly via hierarchical softmax output distributions to distinguish between output modalities (Spithourakis and Riedel, 2018), and fine-tuning our model with contrastive losses to learn to rank numerals and times (Hoffer and Ailon, 2015).

For Restaurant, Attraction, and Train, our model tends to swap slot values with those from other domains in the conversation. This is often due to semantically similar slots whose values, at first glance, may not be obviously identifiable as such (e.g. 'Bridge' or 'The Place'). Kumar et al. (2020) similarly observe a particularly high incidence of slot transfers between different-domain 'Name' slots. Other such slots include price ranges and numbers of guests. We have seen that data augmentation with DSTC8 can improve our model's ability to disambiguate such slots—this suggests that we could further improve our model by exposing it to in-domain, conversational reading comprehension data.

While no such dataset currently exists, in future work we aim to explore using question generation (Du et al., 2017) and paraphrasing (Tseng et al., 2014) models to perform in-domain data augmentation, creating reading comprehension questions for task-oriented dialogs that targeting entities and relations not covered by an ontology. We also wish to explore methods for generating general reading comprehension questions for out-of-domain conversations (Shakeri et al., 2020) to improve our model's domain adaptation ability.

7 Related Works

Modern dialog state tracking seeks to capture evolving user intents in a structured belief state (Thomson and Young, 2010). Traditional systems rely on hand-crafted features (Henderson et al., 2014) and classify slot values from a fixed ontology (Mrksic et al., 2017; Ramadan et al., 2018). Gao et al. (2019) and Zhou and Small (2019) fill some slots via spans extracted from dialog history, although they treat non-numeric slots as categorical. Generative methods (Xu and Hu, 2018; Wu et al., 2019) can predict arbitrary unseen values, with Hosseini-Asl et al. (2020) achieving state-of-the-art supervised DST performance in MultiWOZ 2.1 although they cannot predict unseen slots.

By posing DST as generative QA, our framework can leverage language models pre-trained on open-domain documents (Radford et al., 2019) to understand unfamiliar queries. Like Gao et al. (2020), we seek to answer natural language questions about each slot. We contrast our approach to zero-shot DST—which never has access to slots or dialog from the target domain—and that of Campagna et al. (2020), who expose their 'zero-shot' models to synthetic in-domain conversations that require access to the full ontology of the 'held-out' evaluation domain.

We take inspiration from previous work that frames a wide selection of natural language understanding (NLU) tasks (Wang et al., 2019) as QA (McCann et al., 2018) and span extraction (Keskar et al., 2019). While question-answering can be posed as a span extraction task (Wang et al., 2016), generative approaches have proven successful in answering questions about complex passages (Fan et al., 2019). We use a language modeling approach, taking cues from Raffel et al. (2019) who demonstrate that a large language model trained on next-token prediction can learn to solve many different NLU tasks posed as text. Recent work has also shown that large pre-trained language models can generalize to new NLU tasks with few or no examples (Brown et al., 2020), and we leverage

this alongside world knowledge acquired during the pre-training process (Petroni et al., 2019) to build a DST model that is robust to new domains and slot-value ontologies.

8 Conclusion

This paper proposes a conditional language modeling approach to multi-domain DST posed as a generative question answering task. By leveraging natural language questions as state queries, our model can generalize to unseen domains, slots, and values via its understanding of language. Our model achieves state-of-the-art zero-shot results on the MultiWOZ 2.1 dataset with average per-domain absolute improvements of 5.9% joint accuracy. We also demonstrate that our framework is easily extensible to support transfer learning and learning slot carry-over. In the future, it is worth exploring mechanisms for our model to better understand relative temporal values and general reading comprehension questions from conversations in order to disambiguate semantically similar dialog slots.

Acknowledgments

We thank Ben Liu, Maryam Fazel-Zarandi, Anuj Goyal, and anonymous reviewers for providing valuable feedback on this work. Work was performed during first author's internship at Amazon Alexa AI. Findings and observations are of the authors only, and do not necessarily reflect the views of Amazon or UCSD.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. *arXiv preprint arXiv:2005.14165*.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica S. Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *ACL*.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *AAAI*.

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

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *ACL*.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Kumar Goyal, Peter Ku, and Dilek Hakkani-Tür. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *LREC*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: long form question answering. In *ACL*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *MRQA@EMNLP*.

Shuyang Gao, Sanchit Agarwal, Tagyoung Chung, Di Jin, and Dilek Hakkani-Tür. 2020. From machine reading comprehension to dialogue state tracking: Bridging the gap. *arXiv preprint arXiv:2004.05827*.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tür. 2019. Dialog state tracking: A neural reading comprehension approach. In *SIGDIAL*.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *SIGDIAL*.

Matthew Henderson, Blaise Thomson, and Steve J. Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *SIGDIAL*.

Elad Hoffer and Nir Ailon. 2015. Deep metric learning using triplet network. In *ICLR Workshop*.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Unifying question answering and text classification via span extraction. *arXiv preprint arXiv:1904.09286*.

Seokhwan Kim, Michel Galley, R. Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, Minlie Huang, Luis A. Lastras, Jonathan K. Kummerfeld, Walter S. Lasecki, Chiori Hori, Anoop Cherian, Tim K. Marks, Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, and Raghav Gupta. 2019. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *ACL*.

Adarsh Kumar, Peter Ku, Anuj Kumar Goyal, Angeliki Metallinou, and Dilek Hakkani-Tür. 2020. MADST: multi-attention-based scalable dialog state tracking. In *AAAI*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: slot-utterance matching for universal and scalable belief tracking. In *ACL*.

Oliver Lemon, Kallirroi Georgila, James Henderson, and Matthew N. Stuttle. 2006. An ISU dialogue system exhibiting reinforcement learning of dialogue policies: Generic slot-filling in the TALK in-car system. In *EACL*.

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 *ACL*.

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*.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Nikola Mrksic, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve J. Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *ACL*.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. SOLOIST: few-shot task-oriented dialog with A single pre-trained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *EMNLP*.

Alec Radford, Jeffrey Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever. 2019. Better language models and their implications. *OpenAI Blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*.

Osman Ramadan, Pawel Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *ACL*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

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

Siamak Shakeri, Cícero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. In *EMNLP*.

Georgios P. Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. In *ACL*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*.

Blaise Thomson and Steve J. Young. 2010. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Comput. Speech Lang.*

Ya-Min Tseng, Yi-Ting Huang, Meng Chang Chen, and Yeali S. Sun. 2014. Generating comprehension questions using paraphrase. In *TAAI*, volume 8916, pages 310–321. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In *EMNLP*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner attention based recurrent neural networks for answer selection. In *ACL*.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *SIGDIAL*.

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 *ACL*.

Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *ACL*.

Yang You, Jing Li, Sashank J. Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training BERT in 76 minutes. In *ICLR*.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S. Yu, Richard Socher, and Caiming Xiong. 2019. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *NeurIPS Workshop on Conversational AI*.