

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

Anonymous EACL submission

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 (Weizenbaum, 1966), 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

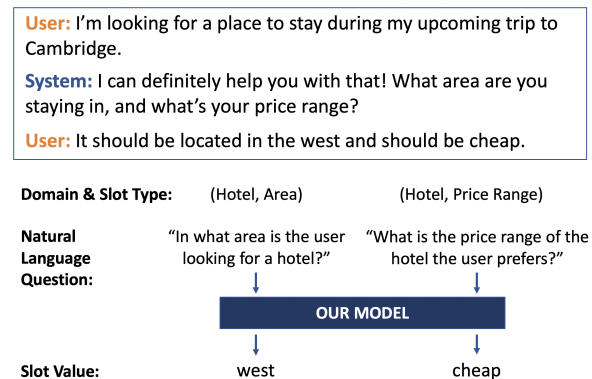


Figure 1: Based on a dialog history, a natural language questions are provided to our model to query a user's requirements and preferences (dialog state).

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.

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 10.40% 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).

Problem Statement We consider a conversation with T turns of user u_t and system utterances y_t : $C = \{y_1, u_1, \dots, y_T, u_T\}$. 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 $C_t = \{y_1, u_1, \dots, y_t, u_t\}$.

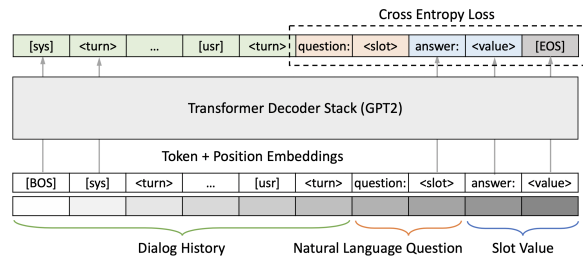


Figure 2: Our model performs DST via generative question-answering. Natural language questions for dialog slots allow our model to generalize to new slot types through its understanding of general language.

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 $v_{s,t} = \{w_1, w_2, \dots, w_k\}, w_i \in \mathcal{W}$. At inference time we pose a natural language question $s = \{w_1, \dots, w_n\}$ 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.

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

| | JGA (%) | # Params |
|----------------------|--------------|----------|
| DistilGPT2 LM | 36.35 | 82 M |
| DistilGPT2 CLM no PT | 39.34 | |
| DistilGPT2 CLM | 49.55 | |
| +Question | <u>50.83</u> | |
| 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.

slots as natural language questions, 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 BART (Lewis et al., 2020)—a Transformer encoder-decoder pre-trained using a denoising auto-encoder objective—and GPT2 (Radford et al., 2019)—a Transformer decoder pre-trained with next-token prediction. We achieved 1% better supervised DST performance with GPT2 in half the training time. We thus based our model architecture on a Transformer decoder (Vaswani et al., 2017) language model, accommodating a full conversation in context alongside a question and answer. This architecture allows us to leverage pre-trained language models like GPT2 and the 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

| | 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.

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; Hosseini-Asl et al., 2020):

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

This allows for flexibility in learned representations

| 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 | |
| <u>GPT2-m CLMQ</u> | 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.

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, 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.

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

| Model | Type | JGA | Extra Supervision |
|--------------------|------|--------------|-------------------------|
| DSTQA | C+S | 51.17 | Knowledge Graph |
| DS-DST | C+S | 51.21 | |
| <u>GPT2-m CLMQ</u> | G | 52.58 | |
| SOM-DST | G | 53.68 | Previous Dialog State |
| SST | C | 55.23 | Schema |
| TripPy | S | 55.30 | Previous Dialog Actions |
| <u>SimpleToD</u> | G | 55.72 | |

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.

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 carry-over (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 methods 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

| | Rest. | Hot. | Attr. | Train | Taxi |
|---------------|--------------|--------------|--------------|--------------|--------------|
| TRADE | 12.59 | 14.20 | 20.06 | 22.39 | 59.21 |
| MA-DST | 13.56 | 16.28 | 22.46 | <u>22.76</u> | 59.27 |
| SUMBT | <u>16.50</u> | <u>19.80</u> | <u>22.60</u> | 22.50 | <u>59.50</u> |
| 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**.

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 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 train our models on four domains from MultiWOZ and evaluate on a 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 common-sense 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 perform zero-shot experiments using each representation, replacing slot names with synonyms (e.g. Taxi ‘leaveAt’ → ‘Pick Up Time’) at inference time for tuple-based slot queries to simulate a truly unseen domain. 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*

| | | | | | | | | | |
|-----|----------------|--|--------|------------------|--------|---------|----------|-------------|-----|
| 500 | USER | My friend told me about Carolina Bed and Breakfast. Do you know anything about it? | | | | | | 550 | |
| 501 | SYS | It's a 4 star guesthouse. What would you like to know about it? | | | | | | 551 | |
| 502 | USER | Can you give me the postcode? And, do they have internet? | | | | | | 552 | |
| 503 | SYS | The postcode is cb13nx; they have internet. | | | | | | 553 | |
| 504 | USER | Thanks. Any boat attractions in the west? | | | | | | 554 | |
| 505 | SYS | Nothing in west. Closest boat is the Cambridge Punter in centre. Too far? | | | | | | 555 | |
| 506 | USER | Yes, it is. How about a museum? | | | | | | 556 | |
| 507 | Error Modality | Slot | Gold | Prediction | Open | Numeric | Temporal | Categorical | 557 |
| 508 | Spurious | (Attraction, Name) | n/a | cambridge punter | 8.4 % | 22.3 % | 47.7 % | 16.0 % | 558 |
| 509 | Ignored | (Hotel, Internet) | yes | n/a | 65.3 % | 53.5 % | 19.9 % | 76.8 % | 559 |
| 510 | Wrong Value | (Attraction, Type) | museum | boat | 26.3 % | 24.2 % | 32.4 % | 7.2 % | 560 |

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**.

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 second 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

| | 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 | <u>26.29</u> | <u>22.94</u> | <u>38.31</u> | <u>38.00</u> | 59.16 |
| 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.

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 (comprising 19 domains and 124 unique slots), taking care to remove data from the held-out domain in both MultiWOZ and DSTC8.

Our augmented GPT2 CLMQ model sees significant improvements in JGA across domains (Table 7). We also observe an average reduction of *ignored* errors by 8.5% across domains, and 8.4% 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 18% fewer errors on categorical slots and 55% 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 allow our model to better disambiguate temporal references, reducing errors in such slots by an average of 26% across domains.

We find that our framework allows for easy joint optimization with transfer learning and carry-over: when applying both to our largest model, we observe further improvements in zero-shot JGA for all domains—averaging 4.5 points better than GPT2-m CLMQ, for an average gain of 10.4% JGA over

User: I am looking for a **cheap** restaurant in the north part of town.
System: There is Royal Spice, which is an Indian restaurant. There is also Da Vinci Pizzeria, which is an Italian restaurant.
User: Please book a table for 5 at 14:30 on Wednesday at Royal Spice. I also need to find a place to stay.

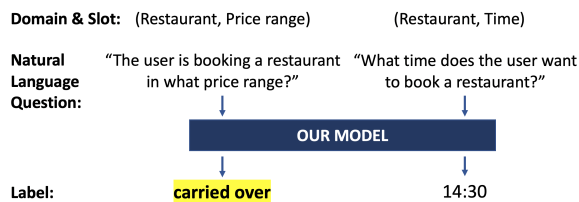


Figure 3: Our model can be trained to predict the presence of slot carry-over by replacing slot values from previous turns with the `carried over` token.

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 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

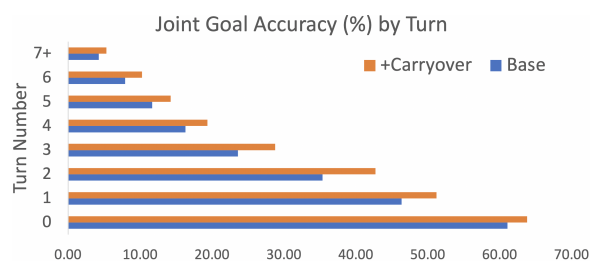


Figure 4: Per-turn JGA on MultiWOZ Test set for GPT2-CLMQ with and without carry-over prediction.

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.

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 super-

vised 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.

References

- 800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
- 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 Geischauser, 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 Sangwoo 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. [MA-DST: 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*.
- 850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899

- 900 Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. *The natural language de-*
901 *cathlon: Multitask learning as question answering.*
902 *arXiv preprint arXiv:1806.08730.* 950
- 903 951
- 904 Nikola Mrksic, Diarmuid Ó Séaghdha, Tsung-Hsien
905 Wen, Blaise Thomson, and Steve J. Young. 2017.
906 *Neural belief tracker: Data-driven dialogue state*
907 *tracking.* In *ACL.* 952
- 908 Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayan-
909 deh, Lars Liden, and Jianfeng Gao. 2020. *SOLOIST:*
910 *few-shot task-oriented dialog with A single pre-*
911 *trained auto-regressive model.* *arXiv preprint*
912 *arXiv:2005.05298.* 953
- 913 Fabio Petroni, Tim Rocktäschel, Sebastian Riedel,
914 Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu,
915 and Alexander H. Miller. 2019. *Language models*
916 *as knowledge bases?* In *EMNLP.* 954
- 917 Alec Radford, Jeffrey Wu, Dario Amodei, Daniela
918 Amodei, Jack Clark, Miles Brundage, and Ilya
919 Sutskever. 2019. *Better language models and their*
920 *implications.* *OpenAI Blog.* 955
- 921 Colin Raffel, Noam Shazeer, Adam Roberts, Katherine
922 Lee, Sharan Narang, Michael Matena, Yanqi Zhou,
923 Wei Li, and Peter J. Liu. 2019. *Exploring the limits*
924 *of transfer learning with a unified text-to-text trans-*
925 *former.* *arXiv preprint arXiv:1910.10683.* 956
- 926 Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018.
927 *Know what you don't know: Unanswerable ques-*
928 *tions for squad.* In *ACL.* 957
- 929 Osman Ramadan, Pawel Budzianowski, and Milica Ga-
930 sic. 2018. *Large-scale multi-domain belief tracking*
931 *with knowledge sharing.* In *ACL.* 958
- 932 Victor Sanh, Lysandre Debut, Julien Chaumond, and
933 Thomas Wolf. 2019. *Distilbert, a distilled version*
934 *of BERT: smaller, faster, cheaper and lighter.* *arXiv*
935 *preprint arXiv:1910.01108.* 959
- 936 Rico Sennrich, Barry Haddow, and Alexandra Birch.
937 2016. *Neural machine translation of rare words with*
938 *subword units.* In *ACL.* 960
- 939 Pararth Shah, Dilek Hakkani-Tür, Gökhan Tür, Ab-
940 hinav Rastogi, Ankur Bapna, Neha Nayak, and
941 Larry P. Heck. 2018. *Building a conversational*
942 *agent overnight with dialogue self-play.* *arXiv*
943 *preprint arXiv:1801.04871.* 961
- 944 Georgios P. Spithourakis and Sebastian Riedel. 2018.
945 *Numeracy for language models: Evaluating and im-*
946 *proving their ability to predict numbers.* In *ACL.* 962
- 947 Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014.
948 *Sequence to sequence learning with neural networks.*
949 In *NeurIPS.* 963
- 950 Blaise Thomson and Steve J. Young. 2010. *Bayesian*
951 *update of dialogue state: A POMDP framework for*
952 *spoken dialogue systems.* *Comput. Speech Lang.* 964
- 953 Ya-Min Tseng, Yi-Ting Huang, Meng Chang Chen,
954 and Yeali S. Sun. 2014. *Generating comprehension*
955 *questions using paraphrase.* In *TAAI*, volume 8916,
956 pages 310–321. Springer. 957
- 958 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob
959 Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz
960 Kaiser, and Illia Polosukhin. 2017. *Attention is all*
961 *you need.* In *NeurIPS.* 958
- 962 Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh,
963 and Matt Gardner. 2019. *Do NLP models know*
964 *numbers? probing numeracy in embeddings.* In
965 *EMNLP.* 959
- 966 Alex Wang, Amanpreet Singh, Julian Michael, Felix
967 Hill, Omer Levy, and Samuel R. Bowman. 2019.
968 *GLUE: A multi-task benchmark and analysis plat-*
969 *form for natural language understanding.* In *ICLR.* 960
- 970 Bingning Wang, Kang Liu, and Jun Zhao. 2016. *Inner*
971 *attention based recurrent neural networks for answer*
972 *selection.* In *ACL.* 961
- 973 Zhuoran Wang and Oliver Lemon. 2013. *A simple*
974 *and generic belief tracking mechanism for the dia-*
975 *log state tracking challenge: On the believability of*
976 *observed information.* In *SIGDIAL.* 962
- 977 Joseph Weizenbaum. 1966. *ELIZA - a computer pro-*
978 *gram for the study of natural language communi-*
979 *cation between man and machine.* *Commun. ACM.* 963
- 980 Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-
981 Asl, Caiming Xiong, Richard Socher, and Pascale
982 Fung. 2019. *Transferable multi-domain state gener-*
983 *ator for task-oriented dialogue systems.* In *ACL.* 964
- 984 Puyang Xu and Qi Hu. 2018. *An end-to-end approach*
985 *for handling unknown slot values in dialogue state*
986 *tracking.* In *ACL.* 965
- 987 Yang You, Jing Li, Sashank J. Reddi, Jonathan Hseu,
988 Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song,
989 James Demmel, Kurt Keutzer, and Cho-Jui Hsieh.
990 2020. *Large batch optimization for deep learning:*
991 *Training BERT in 76 minutes.* In *ICLR.* 966
- 992 Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu,
993 Yao Wan, Philip S. Yu, Richard Socher, and Caim-
994 ing Xiong. 2019. *Find or classify? dual strategy for*
995 *slot-value predictions on multi-domain dialog state*
996 *tracking.* *arXiv preprint arXiv:1910.03544.* 967
- 997 Li Zhou and Kevin Small. 2019. *Multi-domain dia-*
998 *logue state tracking as dynamic knowledge graph en-*
999 *hanced question answering.* *NeurIPS Workshop on*
1000 *Conversational AI.* 968