Stabilized In-Context Learning with Pre-trained Language Models for Few Shot Dialogue State Tracking

Derek Chen, Kun Qian, Zhou Yu
Dialogue NLP Lab
Columbia University
{dc3761, kq2157, zy2461}@columbia.edu

Abstract

Prompt-based methods with large pre-trained language models (PLMs) have shown impressive unaided performance across many NLP tasks. These models improve even further with the addition of a few labeled in-context exemplars to guide output generation. However, for more complex tasks such as dialogue state tracking (DST), designing prompts that reliably convey the desired intent is nontrivial, leading to unstable results. Furthermore, building in-context exemplars for dialogue tasks is difficult because conversational contexts are long while model input lengths are relatively short.

To overcome these issues we first adapt a meta-learning scheme to the dialogue domain which stabilizes the ability of the model to perform well under various prompts. We additionally design a novel training method to improve upon vanilla retrieval mechanisms to find ideal in-context examples. Finally, we introduce a saliency model to limit dialogue text length, allowing us to include more exemplars per query. In effect, we are able to achieve highly competitive results for few-shot DST on MultiWOZ.

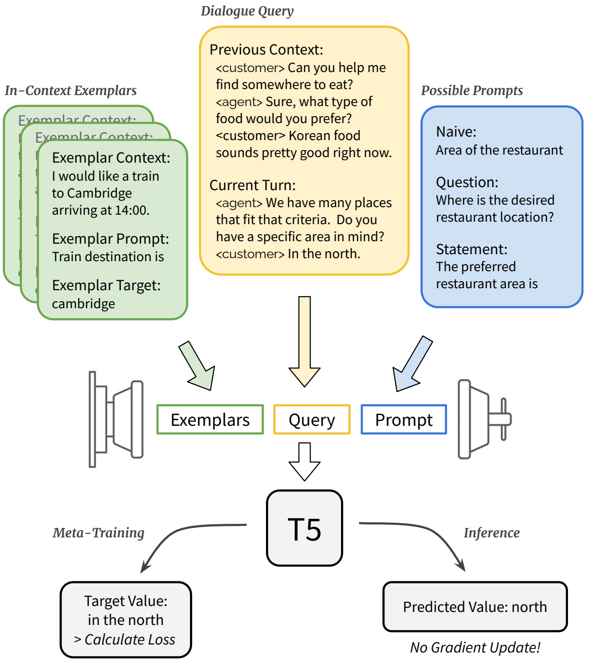

Figure 1: Our system squeezes multiple in-context exemplars, dialogue query with conversational context, and a full prompt into the finite input length of a large PLM to successfully perform few-shot dialogue state tracking, without any need for task-specific training.

1 Introduction

Tremendous gains have been made on dialogue state tracking (DST) using large pre-trained language models (PLMs) (Hosseini-Asl et al., 2020; Peng et al., 2021), Fine-tuning such systems though require significant amounts of data, which in turn require substantial effort to collect. Recently, prompting has emerged as a technique for achieving strong performance in a less resource intensive manner (Schick and Schütze, 2021; Liu et al., 2021). Even better performance is possible with in-context exemplars providing a pattern for the model to follow (Brown et al., 2020). Ideally, we should be able to apply these concepts to complex tasks like DST, but results so far have been limited (Madotto et al., 2021).

One reason for the lack of progress comes from the difficulty of hand-crafting prompts (patterns) and targets (verbalizers), which are highly sensitive to exact phrasing (Lester et al., 2021a). While manually designed prompts have been found to be brittle and unstable (Gu et al., 2021), automatically designed prompts (Gao et al., 2021a) cannot be easily applied to DST since many slots are non-enumerable (Rastogi et al., 2020). A second major hurdle is around dialogue sequence lengths, which are often much longer than those for other tasks (Quan and Xiong, 2020; Kottur et al., 2021) preventing the inclusion of many exemplars for guidance. Full conversations consist of long histories going back many turns, such that the context itself (sans prompt) is already capable of filling

Findings of the Association for Computational Linguistics: EACL 2023, pages 1551–1564
May 2-6, 2023 ©2023 Association for Computational Linguistics

a model's entire input length. Since state tracking requires carrying over previous dialogue states, naively truncating prior context effectively equates to random guessing (Heck et al., 2020; Kim et al., 2020). A third issue is selecting the exemplars themselves. Prior work recommends choosing a representative example from each class (Gao et al., 2021a), but this is not possible in many cases since most domain-slot-value label combinations simply do not appear in the dataset. Moving to the few-shot scenario further exacerbates this sparsity.

Separately, recall that our main goal is to do well in *few-shot* DST because we purposefully operate in a practical, low-resource data setting. Correspondingly, we aim to achieve good results with a similar low-resource model setting where training should be possible on a single publicly-available commodity server. This precludes the usage of gigantic models such as GPT-3, which are prohibitively expensive to train and bear high economic and environmental costs for inference alone (Strubell et al., 2019; Bender et al., 2021).

We directly tackle each of the three aforementioned issues to achieve state-of-the-art performance on MultiWOZ when restricted to models under 100 billion parameters. To minimize prompt issues, we introduce a meta in-context learning (ICL) framework to stabilize training and reduce variance in prompt performance. To deal with long dialogues, we are inspired by summarizaton work to condense dialogue histories and filter out non-salient sentences. Our third contribution is designing a novel loss function to train a retrieval model that selects ideal exemplars for priming our downstream model. Our analysis and ablations show that all components help improve our state tracking performance. Finally, we show that unlike other models which only work on specialized LMs, our proposed methods work on any sort of LM, and can be improved with additional training.

2 Related Works

2.1 Few-Shot Dialog State Tracking

Nearly all recent works on dialogue state tracking leverage large pre-trained LMs to achieve good performance (Heck et al., 2020; Kim et al., 2020; Peng et al., 2021). These methods require fine-tuning on large amounts of annotated data, whereas we hope to do well with minimal data.

Few-shot learning can be achieved in many ways, with transfer learning probably being the most popular, where knowledge is transferred from one domain to another (Wu et al., 2019; Campagna et al., 2020). Data augmentation also supports few-shot learning by generating additional training examples from the few-shot data (Yin et al., 2020; Summerville et al., 2020; Mi et al., 2021). Clustering techniques like prototypical networks have also shown prior success (Snell et al., 2017).

2.2 Meta In-context Learning with Prompting

This work leans on the few-shot techniques of meta-learning (Finn et al., 2017) and prompting with large PLMs (Madotto et al., 2021). Meta-learning allows you to get away with only a few examples at test time by pre-training a model to learn how to learn (Nichol et al., 2018). More recent methods which circumvent the need to calculate second-order gradients (Nichol and Schulman, 2018) have been successfully applied to the task of DST (Dingliwal et al., 2021), but still require fine-tuning on the query set.

Using prompts as natural language instructions have been found to work well on a wide variety of NLP tasks, including dialogue state tracking (Yang et al., 2022). Prompts can be brittle though, so prompt engineering has become its own complex task with numerous ideas on finding discrete prompts (Gao et al., 2021a) or tuning soft prompts, such as through adapters (Xu et al., 2022), prefix tuning (Li and Liang, 2021), or prompt tuning (Lester et al., 2021b). Others have even altered the prompt structure into code in order to fit the capabilities of the network (Lee et al., 2021). Inspired by the success of meta in-context learning on classification tasks (Min et al., 2021; Chen et al., 2022), our work aims to side-step the prompt design issue altogether. Concretely, our method applies meta-learning to teach a model to recognize arbitrary instructions, thereby eliminating the need to rely on domain expertise to craft an optimal prompt.

2.3 Exemplar Retrieval

Lastly, our work is related to retrieval with dense vectors to find good exemplars for in-context learning (Liu et al., 2022). Using dense vectors for similarity search have been applied to dialogue in the past, but mainly in the context of open-domain chat (Adolphs et al., 2021; Komeili et al., 2022) or knowledge-base retrieval (Eric et al., 2017). Lee et al. (2021) is concurrent work which leverages embeddings to search for exemplars in dialogue.

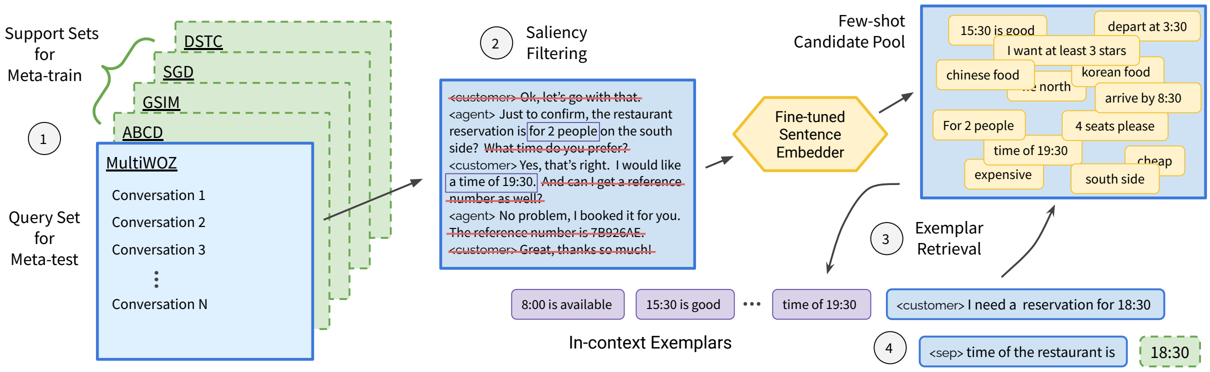
Support Sets for Meta-train
DSTC
SGD
GSIM
ABCD
1
MultiWOZ
Conversation 1
Conversation 2
Conversation 3
⋮
Conversation N
Query Set for Meta-test
2
Saliency Filtering
~~<customer> Ok, let's go with that.~~
<agent> Just to confirm, the restaurant reservation is for 2 people on the south side? ~~What time do you prefer?~~
<customer> Yes, that's right. I would like a time of 19:30. ~~And can I get a reference number as well?~~
<agent> No problem, I booked it for you. ~~The reference number is 7B926AE.~~
~~<customer> Great, thanks so much!~~
Fine-tuned Sentence Embedder
Few-shot Candidate Pool
15:30 is good
depart at 3:30
I want at least 3 stars
chinese food
korean food
north
arrive by 8:30
For 2 people
4 seats please
time of 19:30
cheap
expensive
south side
3
Exemplar Retrieval
8:00 is available
15:30 is good
•••
time of 19:30
<customer> I need a reservation for 18:30
In-context Exemplars
4
<sep> time of the restaurant is
18:30

Figure 2: Our method SM2 includes (1) meta-learning with various support sets, (2) saliency filtering to remove irrelevant utterances and (3) improved exemplar retrieval from a few-shot candidate pool. Exemplars are full utterances with dialogue context, which we display as short phrases for illustrative purposes only. They are concatenated and fed into the model for prediction in Step 4. Items in green boxes, including the target value, are only available during meta-training. Purple items are raw text, while yellow ones represent their embedding vectors.

3 Our Method

This section describes our proposal of a **S**tabilized dialogue state tracker, which leverages **M**eta in-context learning, dialogue **S**ummarization and a novel **M**ulti-part training loss for fine-tuning a retrieval model, which we refer to as **SM2** for short.

3.1 Preliminaries

The goal of dialogue state tracking (DST) is to extract key information from the conversation as a means of understanding the customer's intentions in each dialogue turn. More formally, given the dialogue history $H = \{C_1, A_1, C_2, A_2, \ldots, C_t\}$ composed of a series of utterances between a customer C_i and an agent A_i, the model should predict the cumulative dialogue state up to current t-th turn. This state is represented as a set of *(domain, slot, value)* tuples, which our system produces by iterating over valid domain-slot pairs and then aggregating all non-null, predicted values for the given turn. A few-shot setup only allows access to K% of the available labeled data, with k=[1,5,10] for our experiments, where samples are randomly selected from the full labeled dataset. While we compare to models *trained* on k-shot data, our system actually goes a step further since our eventual model receives *no gradient signal* from the task-specific data and instead relies solely on in-context learning to perform inference.

3.2 Stabilized Meta-learning

The intuition behind prompting is that large PLMs understand instructions when written in natural language (Brown et al., 2020). Thus, we write natural language patterns in an attempt to elicit the dialogue state from the model. However, as previously discussed, minor tweaks in prompt text may cause extreme changes in generated output, leading to highly unstable results (Gu et al., 2021).

Recent works on Meta-ICL (Min et al., 2021; Chen et al., 2022) have shown promise in stabilizing the variance of prompts such that crafting the perfect prompt is no longer necessary, and instead, any reasonable natural language prompt will suffice. Classic meta-learning leverages abundant labeled data from support sets to adapt a model to quickly learn a limited-data target task, denoted as the query set. Finn et al. (2017) proposes MAML that simulates the inner adaptation step during meta-training by conducting a temporary one-step update before computing the loss. Afterwards, a costly second-order gradient is calculated in the outer loop to train the model for faster future adaptations. To get around the expensive loss calculation, variants such as FOMAML have since been developed (Nichol et al., 2018; Nichol and Schulman, 2018). Meta-ICL ingeniously avoids this calculation by replacing the inner adaptation step with in-context learning, which does not require computing gradients! More specifically, in-context learning refers to the use of exemplars to guide the model towards exhibiting ideal behavior. Critically, these exemplars are included as part of the standard model input and thus do not require gradient updates to provide a useful boost.

Following the idea of Meta-ICL, we consider each dataset as a single task and treat MultiWOZ as the held out target task. Specifically, all support datasets are transformed into the DST format for meta-training, where the in-context inner loop consists of support set training examples. Although the model does not learn about the query set in meta-training, it *is* familiarizing itself with complex DST prompts during that time, allowing it to quickly adapt to the target task in meta-testing. Furthermore, since the prompt meaning is learned during meta-training, theoretically any prompt can be used to instruct the model, including prompts constructed from random tokens (See Table 2).

3.3 Dialogue Compression

Condensing the dialogue context not only fits more exemplars into the model input sequence, but also helps the model focus on more relevant text for predicting dialogue states. We introduce two general ideas under the umbrella of compressing long dialogues into shorter input sequences.

Context Summarization As the task name implies, DST requires tracking dialogue states over long periods of time, including slot-values that were carried over from the start of the conversation. Indeed, initial experiments validated a monotonic decrease in joint goal accuracy as each marginal utterance was removed. Therefore, as an alternative to simply removing prior utterances, we propose summarizing the dialogue history instead. The summary of all prior turns is represented as the predicted dialogue state up to that point, which is represented as a series of (domain, slot, value) tuples. We tried further limiting the input length by only including state tuples directly related to the current slot prediction, but surprisingly found that this formulation of the summary fared worse.

Saliency Filtering Many sentences within a conversation do not contain valuable information, such as "Thanks, that is all I need today." or "Good bye". In order to filter away these lines, the first instinct is to train a large model, but our situation only has access to a few labeled examples, so to keep things simple, we instead gather a small handful of heuristics to identify non-salient utterances. For example, lines that discuss a "reference number" or are excessively terse are targeted for removal. We verify the performance of our heuristics on the limited few-shot examples, where we heavily weight the model's recall of salient utterances over its precision. We take a very conservative approach since accidentally dropping a single relevant sentence can cause a severe penalty in joint goal accuracy.

3.4 Multi-part Retrieval Training

Exemplars are the only guiding signal when dealing with in-context learning, so selecting quality cases is of utmost importance. To do so, we fine-tune the sentence embedder used during retrieval by taking advantage of the limited, few-shot data available.

Exemplar Retrieval Exemplars are retrieved based on their proximity to the query example. Concretely, we first encode all available exemplars into a shared embedding space using a SBERT embedder (Reimers and Gurevych, 2019) where the raw text fed into the embedder is the exemplar's dialogue history. For each incoming query, we encode the instance in the same manner, and then compare their embeddings to rank the closest exemplars in the few-shot candidate pool (Step 3 in Figure 2). Finally, we keep pulling exemplars from the top of the stack to feed into the model until the entire context length of 512 is at capacity. Since the exemplar embeddings are pre-computed, looking for similar exemplars during inference is a very quick operation.

Embedder Fine-tuning To improve the performance of our retrieval model, we explore two categories of training techniques. Inspired by the rise of contrastive learning (Hadsell et al., 2006) as a pre-training method for NLP tasks (Gao et al., 2021b; Karpukhin et al., 2020), we first study a CONTRASTIVE loss which brings positive examples closer together while pushing negative examples further apart. In our case, exemplars sharing the same domain and slot are positive (Y=0) while all others are negative (Y=1). The loss becomes:

$$\begin{aligned}\text{Loss}(i,j) = \; & \frac{1-Y}{2}[dist(z_i, z_j)]^2 \; + \\ & \frac{Y}{2}\{max(0, m - dist(z_i, z_j)))\}^2\end{aligned}$$

where z_i represents the embedding vector for utterance i while m is a margin, set to 1. We explored various distance functions (e.g. euclidean) and found that distance based on cosine similarity worked best:

$$dist(z_i \cdot z_j) = 1 - \frac{z_i \cdot z_j}{|z_i| \cdot |z_j|}$$

Since we retrieve exemplars based on cosine score, we can directly optimize for this as second technique with a MEAN-SQUARED ERROR loss. More specifically, the positive pair is assigned a target score of 1 when the two examples share the same domain and slot and 0 otherwise, mirroring the setup of the contrastive loss. The model's predicted cosine score is then compared against this target to calculate an averaged L2-loss. We generate κ pairs for each of N exemplars, and train our ranker with:

$$L(i,j) = \frac{1}{NK}\sum_{i=1}^{N}\sum_{j=1}^{K} ||\text{Target}(i,j) - \text{Pred}(i,j)||^2$$

Multi-part Modification The standard method for selecting negatives has a few drawbacks since all negatives are treated the same. While this is necessary for unsupervised contrastive learning, our case deals with labeled exemplars. Even binary labels would provide a useful training signal, but we even have varying degrees of similarity. In particular, a positive example would be an exemplar that has a matching domain, slot and value. However, exemplars that contain a matching domain or slot still deserves partial consideration rather than being deemed a pure negative example. Consequently, we introduce a MULTI-CONTRASTIVE loss where the different elements of domain, slot and value are considered positive attributes, weighted with their respective lambdas. These coefficients were chosen by tuning on a held-out development set:

$$\text{Loss}(i,j) = \frac{\lambda_d + \lambda_s + \lambda_v}{4}[dist(z_i, z_j)]^2 + \frac{\lambda_n}{4}\{max(0, m - dist(z_i, z_j)))\}^2$$

where:

$$\lambda_d = 3, \quad \lambda_s = 7, \quad \lambda_v = 10$$
$$\lambda_n = 1.0, \quad margin = 1.0$$

For a final loss function, we also test a novel cosine similarity loss where the target label is modified to include multiple parts, MULTI-MSE. The target is altered such that a matching domain for each pair gets $\lambda_d = 0.3$, a matching slot receives another $\lambda_s = 0.3$ boost and matching values get an additional $\lambda_v = 0.4$, where the weights are derived by tuning on the dev set. The final target score is the cumulative sum of the three components - positive pairs sharing all elements get a full score of 1, negative pairs with no matching elements receive a 0, and most pairs lie somewhere in the middle.

$$\text{Target}(i,j) = \sum_e \lambda_e[\mathbb{1}\{e_i = e_j\}], \forall e \in \{d, s, v\}$$
$$\text{s.t.} \qquad \lambda_d + \lambda_s + \lambda_v = 1$$

Dataset	# Dialogs	# Domains	# Slots
MultiWOZ	8,438	7	24
SGD	16,142	16	214
GSIM	1,500	2	13
DSTC2	1,612	1	8
ABCD	8,034	30	231

Table 1: Statistics of involved task-oriented dialogue datasets. Note that the numbers reported are for the training portions for all datasets.

3.5 Model Input

The eventual sequence we feed into the model takes all of the above ideas into account. We start with a context summary represented as the predicted dialogue state, followed by the current turn which consists of two utterances. Each utterance includes a special `<agent>` or `<customer>` token for the respective speaker. Next, a separator token is added, along with a discrete prompt describing the domain and slot. Lastly, we prepend as many exemplars as we can fit into the model maximum token length, truncating from the beginning when necessary. This results in a final model input of:

$$[N\ exemplars][prev\_dialog\_state][agent\_utt]$$
$$[customer\_utt] < sep > [prompt][value]$$

Notably, the final `[value]` token is only present during meta-training, and belongs to the support datasets. This value is precisely what we hope to predict when testing the left out query set.

4 Experiments

This section outlines our training implementation details as well as key experiments.

4.1 Training Setup

We consider Schema Guided Dialogue (SGD) (Rastogi et al., 2020), DSTC2 (Henderson et al., 2014), Action-Based Conversations Dataset (ABCD) (Chen et al., 2021), and Google Simulated Chat (GSIM) (Shah et al., 2018) as support sets (listed in Table 1). We then use MultiWOZ 2.1 (Budzianowski et al., 2018; Eric et al., 2019) as a query set, as well as MultiWOZ 2.4 (Zang et al., 2020) which is the cleanest version of MultiWOZ at time of writing. All datasets have dialogue compression techniques applied and use the best performing embedder for exemplar retrieval.

For our training we use T5 (Raffel et al., 2020) with both the three and eleven billion parameters

Prompt Style	Prompt Example
Statement	“The destined location of the taxi is”
Question	“Where is the destination of the taxi ?”
Schema	“<domain> taxi - rent cheap cabs to avoid traffic <slot> destination - what place you want the taxi to take you”
Naive	“destination of the taxi is”
None	“taxi destination”
Random	“blue cobra”

Table 2: Examples for different prompt styles. Here we consider a domain of “taxi” and a slot of “destination”.

versions (T5-3b/T5-11b), where our best models are selected through early stopping on validation data. We set the learning rate as $3e-4$, employ an Adafactor (Shazeer and Stern, 2018) optimizer and cosine scheduler with warmup of 10,000 steps. Our best system uses an ensemble of exemplar embedders that were trained with of $\kappa = [20, 30, 40]$ and learning rate of $3e-5$. More details can be found in Appendix C.

4.2 Prompt Variations

Model training can be considered stable if different prompts produce similar outcomes. To test this, we collect six prompts based on common sense and prior work. As much as possible, we use prompts designed by others to avoid biasing the rankings.

Since LMs supposedly operate on prompts as continuation of natural language, the (a) *Statement* prompt takes the form ‘The restaurant cuisine is <blank>’, where we hope the model completes the sentence with the correct slot-value. (b) A *Question* prompt reverses the meaning with ‘What is the restaurant cuisine?’ (c) *Schema* comes from (Lee et al., 2021) and MWOZ 2.2 descriptions, which aims to provide the model with the maximum amount of information. It includes a special token, name, and full description for both the domain and slot. (See Table 2) (d) *Naive* takes the opposite approach by simply following the format of “<slot> of the <domain> is <blank>”. (e) Taken even further, the *None* prompt does not use any natural language at all, instead opting to only include the domain and slot name for evaluation purposes. (f) Finally, we include a *Random* prompt which drops any notion of semantics by replacing the domain with a random color and the slot with a random animal. To empathize with the difficulty of hand-engineering a prompt, note that each option (except for random) seems reasonable, and it is hard to know a priori which one works best.

	MRR@10	NDCG@10	MAP@100
Default	16.7%	9.59%	1.81%
Contrastive	17.4%	10.6%	2.28%
Multi-contrast	17.1%	9.89%	1.90%
Mean Squared	25.1%	15.5%	3.31%
Multi-MSE	**26.8%**	**18.4%**	**5.24%**

Table 3: Results of fine-tuning the sentence embedder with various loss functions. Multi-part cosine is best.

As a baseline, we start with in-context learning without meta-training. We feed in the prompts directly and measure their variance as the standard deviation among scores. Then, we perform meta-learning with all prompts again and measure their results, where we expect that the variance among the scores has now decreased.

4.3 Filtering Threshold

In order to verify that our saliency model successfully removes irrelevant sentences, we employ two experts to annotate 50 dialogs, which is well below the allowed 1% of few-shot data. We then run the saliency model on this tiny evaluation set with different filtering thresholds, ranging from 0.1 to 0.9, with results illustrated in Figure 3. As the threshold increases, only sentences with high relevance are left, as evidenced by high precision and low recall. A maximum F1-score is reached at 0.6, but we would rather keep all relevant sentences at the expense of amassing a handful of irrelevant sentences than to risk missing important information. As a result, we choose 0.4 as the filtering threshold, which achieves a recall of 0.998 and acceptably high precision. Qualitative examples of irrelevant sentences that were removed can be found in section 5.4.

4.4 Retrieval Methods

We adapt SBERT (Reimers and Gurevych, 2019) to our DST task with four different objective functions: standard contrastive loss, multi-part contrastive loss, binary cosine similarity loss and multi-part cosine similarity loss. We test with number of pairs per exemplar in a range from 10 to 100 in increments of ten. We found $\kappa = 30$ to work best, which we use moving forward. As a control, we also include the default SBERT model without any further fine-tuning. We evaluate the results of training on the few-shot examples with Mean Recipricol Rank (MRR@10), Normalized Discounted Cumulative Gain (NDCG@10) and Maximum Average Precision (MAP@100) as our metrics.

Models	Parameter	MultiWOZ2.1			MultiWOZ2.4		
	Size	1%	5%	10%	1%	5%	10%
TRADE (Wu et al., 2019)	<1B	12.58	31.17	36.18	-	-	-
SGPDST (Lee et al., 2021)		32.11	43.14	**46.92**	-	-	-
DS2-BART (Shin et al., 2022)		28.25	37.71	40.29	30.55	42.53	41.73
DS2-T5 (Shin et al., 2022)		33.76	44.20	45.38	36.76	49.89	51.05
IC-DST GPT-Neo 2.7b (Hu et al., 2022)	<100B	16.70	26.90	31.65	17.36	29.62	34.38
IC-DST CodeGen 2.7b (Hu et al., 2022)		20.72	29.62	33.81	21.87	33.16	37.45
SM2-3b (Our Method)		38.06	39.94	39.85	37.59	49.22	50.33
- Saliency Filtering		36.11	38.26	38.63	-	-	-
- Context Summarization		37.02	37.83	37.80	-	-	-
- Embedder Fine-tuning		27.15	30.88	31.40	-	-	-
SM2-11b (Our Method)		**38.36**	**44.64**	46.02	**40.03**	**51.14**	**51.97**
IC-DST Codex-davinc 175b (Hu et al., 2022)	>100B	43.13	47.08	48.67	48.35	55.43	56.88

Table 4: DST performance using 1%, 5% and 10% of the training set. Naive prompt used for our method. Bolded numbers indicate highest performance on models under 100 billion parameters. Note that models <1B params fine-tune on task data. Ablation results are also included for dialogue compression and embedder training.

As is shown in Table 3, the multi-part cosine loss showcases the strongest ability to select meaningful exemplars. This shows the benefit of providing partial credit to all elements of the dialogue state. Surprisingly though, the multi-part contrastive loss underperformed. Preliminary error analysis revealed negative examples were successfully separated from positive examples, but the different positive examples were mixed together. We adopt the embedder trained with the MULTI-MSE for all remaining experiments.

5 Results and Analysis

The goal of this work is to achieve strong results on DST without worrying about tedious prompt-engineering. Consequently, we first analyze the ability of the best performing models and then discuss performance stability across different prompts.

5.1 Main Results

Table 4 shows that methods based on in-context learning clearly surpass those based on fine-tuning with few-shot data, as evidenced by the strong performance of SM2 as well as the concurrent work of IC-DST (Hu et al., 2022). In fact, our SM2-11b model is able to achieve the best joint goal accuracy on MultiWOZ 2.1 and 2.4 for most few-shot splits, when focused on models less than 100B parameters. Furthermore, when considering just models operating with in-context learning, SM2-3b greatly outperforms the IC-DST 2.7b models in the same order of magnitude. We note that our method is agnostic to model size, so it is certainly possible to combine them with systems larger than 100B params. Doing so would likely yield strong performance without sacrificing stability.

On that note, Table 5 shows that models trained with SM2 exhibit roughly a 2x reduction in variance over models trained under other regimes. While fine-tuning on certain prompts produces some of the highest scores we observe, other prompts yield some of the lowest, highlighting how hand-crafting prompts are wrought with danger. The instability is most pronounced for the random prompt, which meta-learning is able to smooth over. Also worth noting is that meta-learning from SM2 is able to stabilize prompt performance across multiple model types, including sequence-to-sequence (row 4) or auto-regressive LMs (row 5). This is in contrast to purely in-context models, such as those which were pre-trained on code and must always obey a rigid coding structure during inference.

5.2 Ablation Study

To evaluate the different contributions, we run three ablation experiments, each of which removes one of the key components of SM2. The results presented in Table 4 show that each change makes a noticeable impact. Without saliency filtering, model performance drops by a small, but consistent amount of roughly 1-2%. Disabling context summarization means truncating dialogue history to four utterances and precluding previous dialogue state, which causes an even bigger decrease in accuracy. Using the default SBERT embedder deals the most damage of all, leading to a nearly 10% drop. This suggests that exemplar selection is most critical for in-context learning methods.

Prompt Style	None	Naive	Schema	Statement	Question	Random	STDEV
Fine-Tune	35.3	39.2	38.7	41.1	39.3	24.7	6.02
In-Context	17.5	19.9	14.6	18.9	12.4	4.80	5.58
Pre-train	31.8	35.4	28.2	27.8	34.6	17.2	6.65
SM2 T5-3b	33.9	39.9	30.0	38.2	35.6	33.1	3.58
SM2 GPT-XL	9.70	8.70	8.50	11.4	8.90	1.20	3.53

Table 5: Joint goal accuracy over different prompt styles. Models trained with 5% of training data. The backbone model of Fine-tune and In-Context is T5-3b. Instability is measured as standard deviation of the accuracy scores.

The proposed ideas are also independently applicable to other NLP tasks. For example, compressing inputs to fit more exemplars into an model input sequence can be applied to dialogue generation with large LMs or even reading compression, which requires reasoning over long supporting paragraphs. A multi-part training mechanism can be applied to tasks that contain multiple elements, such as the premise, hypothesis and labels of NLI.

5.3 Additional Discussion

We now turn our attention to the impact of different training regimes, as shown in Table 5. Fine-tuning (row 1) serves as an oracle since it represents training directly on the data in the target domain. Unsurprisingly, SM2 reaches lower average results in comparison. In contrast, SM2 significantly outperforms in-context learning (row 2) since neither perform gradient updates, while SM2 includes a meta-learning stage. Finally, to disentangle the effects of pre-training and meta-ICL, we also compare against a baseline which does not perform in-context learning (row 3). Rather than learning the prompts, this baseline instead simply performs transfer learning from the source datasets to the target dataset. Such a setup does not work as well due to the domain shift from the source distribution to the target distribution.

Digging deeper, we notice that our method displays a meaningful jump in performance when going from 1% to 5% data, but not much when going to 10%. The increased amount of data fails to provide much marginal value since the exemplars being selected did not change much despite choosing from a larger candidate pool. Instead, the finite sequence length became the bottleneck on downstream accuracy.

The performance of the in-context methods are interesting in their own right. Statement prompt does best, while Random does worst, but despite having no training, is well above chance. This surprising result confirms other research on prompt analysis, which found that large PLMs sometimes perform *too well*, implying that the models are actually paying attention to superficial cues rather than truly understanding the text within a prompt (Webson and Pavlick, 2021; Kavumba et al., 2022).

5.4 Qualitative Analysis

The top half of Table 6 shows an utterance with "*domain=restaurant*" and "*slots=price range, food type*". Despite having minimal n-gram overlap with the example, the first exemplar E1 receives a high score by matching the same domain and slot of the target utterance. On the other hand, the second exemplar E2 discusses an entirely different topic, producing a low score. This demonstrates the effectiveness of the sentence embedder in distinguishing the value of these exemplars. The bottom half of Table 6 shows how the saliency model successfully conserves a large amount of token space. Short sentences and those void of any dialog state information are safe for removal. When all sentences in an utterance are filtered, then we also remove the associated speaker token. Despite our conservative thresholds, the majority of useless information is successfully trimmed out to allow the model to focus on the most pertinent areas instead.

6 Conclusion

In this paper, we presented a method of performing few-shot dialogue state tracking by leveraging large pre-trained LMs with prompts. Our technique does not require any gradient-based training for the target task and instead relies on in-context learning to guide model generation. To enable success in this low-resource setting, we stabilize training across prompts with Meta-ICL, apply saliency filtering and context summarization to reduce dialogue length, and fine-tune a sentence embedder with a custom loss objective to improve exemplar retrieval. These techniques combined allow us to reach state-of-the-art results on MultiWOZ when limited to models under 100 billion parameters.

<table>
<tr><th colspan="4">Exemplar Retrieval</th></tr>
<tr><th>Dialog ID</th><th>Target Utterance</th><th>Exemplar</th><th>Score</th></tr>
<tr><td rowspan="2">SSNG0074.json</td><td rowspan="2">I am looking for a restaurant in the **moderate price range** that serves **bistro type food**.</td><td>E1: I would love to help. any particular food you'd like? **no**, I'd just like for it to be in the east and **moderately priced**.</td><td>0.738</td></tr>
<tr><td>E2: Seventeen locations meet your criteria. Would you prefer a guesthouse or a hotel? A hotel is fine whichever you recommend.</td><td>-0.074</td></tr>
<tr><th colspan="4">Saliency Filtering</th></tr>
<tr><td>PMUL0287.json</td><td colspan="3"><Agent>: The phone number is 01223259988. <User>: ~~Perfect.~~ Can you help me with a reservation for 6 people at 14:30 this coming sunday? ~~And please make sure I have a confirmation number to use.~~ ~~<Agent>:our reservation is set!~~</td></tr>
<tr><td>PMUL1635.json</td><td colspan="3"><Agent>: What day will you be staying? <User>: Friday and Can you book it for me and get a reference number ?~~<Agent>:Booking was successful.~~ ~~Reference number is : BMUKPTG6.~~ ~~Can I help you with anything else today?~~ <User>: I am looking to book a train that is leaving from Cambridge to Bishops Stortford on Friday.</td></tr>
</table>

Table 6: Examples of how exemplar retrieval and saliency filtering operate. Same colored text represents matching domain and slots. The strikethrough of text means removal of the irrelevant sentence by the saliency model.

Moving forward, we plan to explore techniques that push model and data efficiency even further. Distillation and pruning can lead to much fewer model parameters, while numerous data augmentation techniques seem promising in maximizing the advantage of limited labeled data. Lastly, rather than meta-learning across different dialog domains, we also would like to explore meta-train model with different prompt styles. With the current framework, the prompt used in inference is required to be the same as the training. However, we might want to use flexible prompts in practice. Consequently, we could meta-train across different prompt styles to allow the model to quickly learn a new prompt style during inference.

7 Limitations

Our method is model-agnostic and can be combined with larger pre-trained model over 100 billion parameters for further improvement on DST task. However, due the budget limit, this is unlikely to be directly validated. Ironically, our method also has the limitation that it cannot be combined with smaller models since the emergent behavior of being to understand prompts only seems to occur with sufficiently large pre-trained models.

Separately, the proposed saliency filtering and the exemplar retrieval module are designed based on the dialog state tracking task, but not specifically for the MultiWOZ dataset. As a result, we planned to apply our framework to other task-oriented dialog datasets, e.g. SGD (Rastogi et al., 2020) to verify that our framework is generalizable, but have not done so yet due to time constraints. We also ran our experiments with a different model type in GPT-XL, but did not have a chance to properly tune the parameters, leading to low performance.

We would have liked to run our experiments with different random seeds. Considering the stability of our framework among different prompt styles, different random seeds should not cause high variance. However, we still need to run experiments to verify this assumption.

References

Leonard Adolphs, Kurt Shuster, Jack Urbanek, Arthur Szlam, and Jason Weston. 2021. Reason first, then respond: Modular generation for knowledge-infused dialogue. CoRR, abs/2111.05204.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

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, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff 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, abs/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 Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 5016–5026. Association for Computational Linguistics.

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

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3002–3017, Online. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of Proceedings of Machine Learning Research, pages 1597–1607. PMLR.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

Saket Dingliwal, Bill Gao, Sanchit Agarwal, Chien-Wei Lin, Tagyoung Chung, and Dilek Z. Hakkani-Tür. 2021. Few shot dialogue state tracking using meta-learning. In EACL.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. arXiv preprint arXiv:1907.01669.

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

Chelsea Finn, P. Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In ICML.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3816–3830, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. Simcse: Simple contrastive learning of sentence embeddings. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 6894–6910. Association for Computational Linguistics.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. PPT: pre-trained prompt tuning for few-shot learning. CoRR, abs/2109.04332.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2:1735–1742.

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 Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL), pages 263–272.

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.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. ArXiv, abs/2203.08568.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 6769–6781. Association for Computational Linguistics.

Pride Kavumba, Ryo Takahashi, and Yusuke Oda. 2022. Are prompt-based models clueless? In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2333–2352, Dublin, Ireland. Association for Computational Linguistics.

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

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

Satwik Kottur, Chinnadhurai Sankar, Zhou Yu, and Alborz Geramifard. 2021. DialogStitch: Synthetic deeper and multi-context task-oriented dialogs. In Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 21–26, Singapore and Online. Association for Computational Linguistics.

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

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021a. The power of scale for parameter-efficient prompt tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021b. The power of scale for parameter-efficient prompt tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4582–4597, Online. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In DEELIO.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ArXiv, abs/2107.13586.

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

Fei Mi, Wanhao Zhou, Lingjing Kong, Fengyu Cai, Minlie Huang, and Boi Faltings. 2021. Self-training improves pre-training for few-shot learning in task-oriented dialog systems. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 1887–1898. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. ArXiv, abs/2110.15943.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. ArXiv, abs/1803.02999.

Alex Nichol and John Schulman. 2018. Reptile: a scalable metalearning algorithm. arXiv: Learning.

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

Jun Quan and Deyi Xiong. 2020. Modeling long context for task-oriented dialogue state generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7119–7124, Online. Association for Computational Linguistics.

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

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 255–269, Online. Association for Computational Linguistics.

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

Noam M. Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. ArXiv, abs/1804.04235.

Jamin Shin, Hangyeol Yu, Hyeongdon Moon, Andrea Madotto, and Juneyoung Park. 2022. Dialogue summaries as dialogue states (DS2), template-guided summarization for few-shot dialogue state tracking. In Findings of the Association for Computational Linguistics: ACL 2022, pages 3824–3846, Dublin, Ireland. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. ArXiv, abs/1703.05175.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Adam Summerville, Jordan Hashemi, James Ryan, and William Ferguson. 2020. How to tame your data: Data augmentation for dialog state tracking. In Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pages 32–37, Online. Association for Computational Linguistics.

Albert Webson and Ellie Pavlick. 2021. Do prompt-based models really understand the meaning of their prompts? CoRR, abs/2109.01247.

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

Yan Xu, Etsuko Ishii, Samuel Cahyawijaya, Zihan Liu, Genta Indra Winata, Andrea Madotto, Dan Su, and Pascale Fung. 2022. Retrieval-free knowledge-grounded dialogue response generation with adapters. In Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering, pages 93–107, Dublin, Ireland. Association for Computational Linguistics.

Yuting Yang, Wenqiang Lei, Juan Cao, Jintao Li, and Tat-Seng Chua. 2022. Prompt learning for few-shot dialogue state tracking. ArXiv, abs/2201.05780.

Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dialog state tracking with reinforced data augmentation. In AAAI.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. Multiwoz 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. CoRR, abs/2007.12720.

A Loss Functions

Gao et al. (2021b) proposes a softmax-based contrastive loss:

$$L_i = -log\frac{e^{sim(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(\mathbf{h}_i,\mathbf{h}_j^+)/\tau}}$$

which is popular among NLP tasks. However, this loss function requires extremely large batch sizes to work well (Chen et al., 2020). This is especially difficult for us since we specifically target a low-resource setting with small GPU memory requirements. More critically, this softmax contrastive loss views all negatives as being the same. However, in the case of dialog state tracking, where dialog state is represented as (domain, slot, value), the matching is decided at three levels. For example, two dialogue examples can (and should) be considered a negative pair when they have different values for all three elements. In another case though, they might be considered a negative pair by not having matching "value", but still sharing the same "domain" and "slot". The softmax constrastive loss considers these two cases as the same, which is not ideal for the DST task. Therefore, we implement the for our experiments. The classic max-margin contrastive loss (Hadsell et al., 2006) is also unable to make a clear distinction for partial credit either, but should be able to when the loss is the sum of multiple elements. Therefore, we use the max-margin loss for our experiments.

B Filtering Results

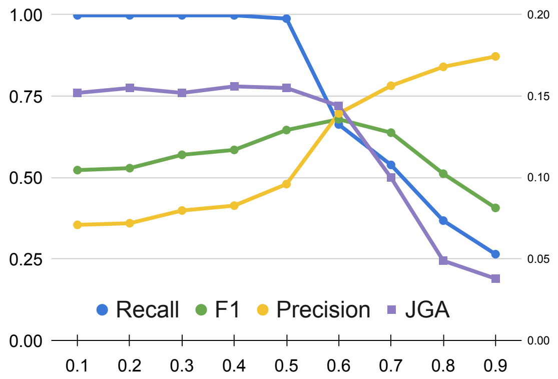

Figure 3: Graph of precision, recall and F1 when varying the acceptance threshold. Joint goal accuracy (JGA) correlates closely with recall due to the nature of DST.

C Other Implementation Details

In this section, we introduce more implementation details. For training, we search the learning rate within the interval [3e-5, 1e-4, 3e-4, 1e-3, 3e-3]. In order to deploy large pre-trained models like T5-3b and T5-11b, we first adjust the batch size. To achieve a balance between GPU memory consumption and batch performance, we alter the number of gradient accumulation steps to maintain a consistent effective batch size of 64 across runs. Furthermore, we also change everything into bitfloat 16 (BF16) and adopt AdaFactor as the optimizer to lower the number of parameters.

We additionally perform ensemble decoding for multiple times using different retrieval embedders. These sentence embedders are distinguished by being trained on different levels of kappa, where we end up choosing embedders trained with kappa of [20,30,40]. These values were selected since they were the models which had the best results as measured by MRR@10 and MAP@10. We run exemplar retrieval with these models and take the majority vote of the system.

In addition to adopting different prompts for our models, we also apply the concept of verbalizers (Schick and Schütze, 2021). More specifically, we use verbalizers to map natural sounding output to the more limited slot-values in the ontology. For example, given the prompt 'Whether the hotel offers wifi', we consider both 'True' (or 'False') and 'Yes' (or 'No') to be the same answer.

D Input Example

(See next page.)

Exemplar 0 (Truncated)	<pad> options available. Would you like to narrow it down by departure time or arrival time? <customer> I'd like to leave after 21:45, if possible. I won't need to book. I'll just need the arrival time, please? <sep> departure of the train is cambridge</s>
Exemplar 1	taxi destination kambar, taxi departure lovell lodge <agent> when would you like to arrive? <customer> It doesn't matter. I just want to leave there after 10:45 <sep> destination of the taxi is kambar</s>
Exemplar 2	taxi destination riverboat georgina, taxi departure archway house, hotel area north, hotel day thursday, hotel stay 5, hotel people 3, hotel stars 4, attraction name cambridge punter, attraction type boat <agent> what time would you like to leave or arrive by? <customer> I'd like to leave the hotel by 3:15 please. <sep> stars of the hotel is 4</s>
Exemplar 3	train day saturday, train destination cambridge, train departure ely <agent> sure, do you know what time you want to arrive? <customer> I want to arrive by 11:30. <sep> departure of the train is ely</s>
Exemplar 4	restaurant area centre, restaurant people 8, restaurant day thursday, restaurant time 14:00, restaurant food chinese, restaurant price range cheap, taxi destination charlie chan, taxi departure museum of classical archaeology, attraction name museum of classical archaeology <agent> When would you like the leave and arrive by? <customer> I don't mind what time we leave, but I need to arrive at the restaurant by 14:00. <sep> departure of the taxi is museum of classical archaeology</s>
Exemplar 5	restaurant area south, restaurant food asian oriental, restaurant name any, restaurant price range any, train arrive by none, train day wednesday, train destination cambridge, train departure london kings cross, train leave at none, attraction area east <agent> what time were you wanting to leave by or arrive by? <customer> I want to arrive by 12:15. <sep> arrive by of the train is 12:15</s>
Prev State	taxi destination pizza hut fen ditton
Dialog Context	<agent> What time do you want to leave and what time do you want to arrive by? <customer> I want to leave after 17:15.
Prompt	leave at of the taxi is</s>
Label	after 17:15

Table 7: A practical example used during inference which uses our fine-tuned sentence embedder for exemplar retrieval. To be easy to read, we separate each component, including exemplars, query sequence and prompt. Each exemplar contains previous states, dialog context, prompt and label, which corresponds to Sec. 3.5. The 0-th exemplar is truncated so that the entire sequence length can fit into the model.