Effective and Efficient Conversation Retrieval for Dialogue State Tracking with Implicit Text Summaries

Seanie Lee[†*] **Jianpeng Cheng**[‡] **Joris Driesen**[‡] **Alexandru Coca**[♣*] **Anders Johannsen**[‡]
[†]KAIST [‡]Apple [♣]University of Cambridge
[†]lsnfamily02@kaist.ac.kr
[‡]{jianpeng.cheng, joris_driesen, ajohannsen}@apple.com
[♣]ac2123@cam.ac.uk

Abstract

Few-shot dialogue state tracking (DST) with Large Language Models (LLM) relies on an effective and efficient conversation retriever to find similar in-context examples for prompt learning. Previous works use raw dialogue context as search keys and queries, and a retriever is fine-tuned with annotated dialogues to achieve superior performance. However, the approach is less suited for scaling to new domains or new annotation languages, where fine-tuning data is unavailable. To address this problem, we handle the task of conversation retrieval based on text summaries of the conversations. A LLM-based conversation summarizer is adopted for query and key generation, which enables effective maximum inner product search. To avoid the extra inference cost brought by LLM-based conversation summarization, we further distill a light-weight conversation encoder which produces query embeddings without decoding summaries for test conversations. We validate our retrieval approach on MultiWOZ datasets with GPT-Neo-2.7B and LLaMA-7B/30B. The experimental results show a significant improvement over relevant baselines in few-shot DST settings.

1 Introduction

Dialogue state tracking (DST) is one of the most crucial components in task-oriented dialogue systems. The goal of DST is to track users' intents, slots and values at every turn of a dialogue based on a predefined schema (Budzianowski et al., 2018). The challenge of training a supervised DST model lies in the cost of dialogue state annotations, which is not scalable to new schemas, domains or annotation languages. To address these challenges, recent works (Hu et al., 2022; Chen et al., 2023) adopt in-context learning with pre-trained large language models (LLM) for few-shot DST. In the few-shot setting, similar dialogue exemplars are retrieved based on the test sample and then these exemplars are added to the LLM prompt for target generation. This approach is attractive since no domain-specific fine-tuning is required for the LLM but it can still generalize to unseen domains.

One challenge in few-shot DST is how to retrieve salient conversation exemplars (e.g., in a set of 3 to 5) from the support set, which serves as demonstrations for the LLM. Ideally, a retrieved exemplar should carry both the same dialogue history and state change as the test sample. However, in a practical few-shot setting (e.g., with at most 100 annotated support examples), it is likely that no exemplar in the support set satisfies the above requirement. Consider a test example with two user turns:

```
user: book a flight to London Heathrow
system: where are you departing from
user: Amsterdam
```

It is possible that the closest exemplar we can get from the support set is:

```
user: I’m leaving Manchester by air
system: where are you flying to
user: To Paris
```

which neither matches the test dialogue state nor the state change. Nevertheless, we hope that LLM can generalize by learning from such exemplars with an identical user intent. The retrieval task gets harder when the conversation becomes lengthy with only partial history related to the current user input. For example, in another test dialog:

```
user: what’s the weather in London
system: sunny
user: book a flight to London Heathrow
system: where are you departuring from
user: Amsterdam
```

the user's current intent is identical to the earlier test sample, but it involves unrelated history. Still we want to match the test sample to a similar exemplar, which reflects the user's intent up to the current point of conversation. This retrieval cannot be easily accomplished with pre-trained dense

[*] Work done during an internship at Apple.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 96–111
June 16-21, 2024 ©2024 Association for Computational Linguistics

retrieval models based on word or sentence similarity. To optimize retrieval performance, previous works (Hu et al., 2022; Chen et al., 2023) fine-tune a dense retriever with "structurally similar" dialogue examples identified from dialogue state annotations with heuristics. Hu et al. (2022) additionally report that including dialogue state information in the retrieval key is helpful. However, the approach is not scalable to a practical few-shot setting (with fewer than 100 annotated support examples), as fine-tuning easily leads to overfitting and catastrophic forgetting (McCloskey and Cohen, 1989; Lee et al., 2022). It is also impractical to expect every domain owner to create their own fine-tuning data with well-engineered rules.

In this work, we propose a new solution for conversation retrieval starting with the introduction of a LLM-based conversation summarizer. For each exemplar to be indexed and also each test dialog, the summarizer produces a text summarizing *what the user wants at this point of the conversation*. In Section 2, we provide a discussion of this specific summarization choice and how it compares to dialogue state. The summaries are then used as condensed search keys and queries applicable to pre-trained dense retrievers with standard nearest neighbor search. We empirically show that in the few-shot setting, using summaries as retrieval keys and queries is more effective than using raw dialogues.

Notably the conversation summarization task described above can be easily handled by state-of-the-art LLMs via prompt learning, as we will show in an ablation study. However, the deployment of such a retrieval system also introduces extra model parameters and inference cost. Unlike search keys, which can be pre-built offline, a search query needs to be auto-regressively decoded for each test dialogue right during inference. To improve the efficiency of this conversation retriever, our second contribution in this work focuses on distilling a light-weight conversation encoder which embeds a raw dialogue directly into a vector space similar to the embedding of its summary. The light-weight conversation encoder enables efficient conversation search over a vector database without explicit query generation. When evaluated on the MultiWOZ dataset with GPT-Neo-2.7B (Black et al., 2022), LLaMA-7B, and LLaMA-30B (Touvron et al., 2023) for few-shot DST, we find that the distilled conversation encoder is not only more efficient, but also more effective than a cascaded conversation retriever with explicit query generation. Our approach also significantly outperforms relevant baselines, which use annotated dialogues for retriever fine-tuning.

2 Conversation Retrieval with Summaries for LLM-based DST

In the context of task-oriented dialogues with multiple turns of interactions between a user and a system, the objective of DST is to predict the accumulated intents, slots and values at each user turn. In a LLM-based approach, the generation of a dialogue state is conditioned on a task-specific prompt. The prompt includes at least the test conversation and a set of k demonstration examples, from which we expect the LLM to learn to generalize. Considering the size limit of the prompt, k is expected to be small (3-5 examples). Each of the retrieved examples is an annotated conversation sharing similar features as the test conversation. We expect to retrieve these exemplars with a dense retriever from a "support set" (e.g., 100 annotations) that can be constructed with minimum effort for domain scaling.

There are two major challenges of conversation retrieval for LLM-based DST described above. First, a good representation of search keys and queries need to be found. As we analyze in Section 1, the similarity of two dialogues is not directly quantifiable by semantic distance, but rather requires more sophisticated structural matching mechanism or a higher-order similarity function. This requirement leads to the second challenge as to how to train an effective conversation retriever that can scale across domains. Previous works (Hu et al., 2022; Chen et al., 2023) mainly fine-tune pre-trained dense retrievers with annotated dialogues obtained from the support set. However, fine-tuning is not realistic in a few-shot setting and for every domain.

As shown in Figure 1-(a), our work introduces a query/key generation step in the LLM-based DST. The generation is performed with another LLM which transforms the raw dialogue context into a text summary whose similarity can be evaluated more easily with pretrained retrievers. Specifically, the text summary represents the user's intent up to the current point of the conversation. It grounds the latest user input onto the dialogue history, keeping only information related to the

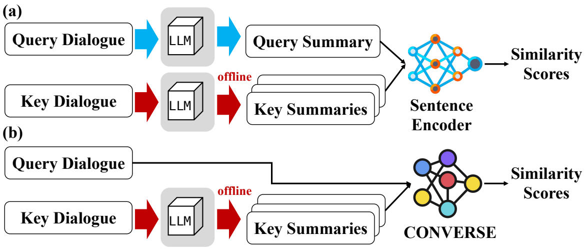


Figure 1: Comparison between **(a)** off-the-shelf retriever with query generation and **(b)** CONVERSE w/o query generation.

current user intent. Note that the summary is a contextual rewriting of the current user intent that is possibly expressed in multiple turns, with applied ellipsis recovery (Hardt, 1997) and co-reference resolution (Pradhan et al., 2012). Examples of the conversation summary are in Table 6. The summary can also be viewed as a text description of an updated dialogue state which is to be predicted by the LLM. Unlike the dialogue state, the summary does not maintain all conversation history but only includes information relevant to the current user input.

With the introduction of an explicit query/key generation step, we expect that the conversation retrieval becomes easier and the search index can be built more efficiently. To construct the search index, an offline process can be triggered to generate text summaries for every example in the support set. Note that search key generation does not add any inference cost. However, the query generation step comes at an extra cost since the generation needs to happen in an online process. In the next section, we describe how to make the conversation retrieval more efficient by stepping away from explicit query generation.

3 Conversation Encoder Distillation

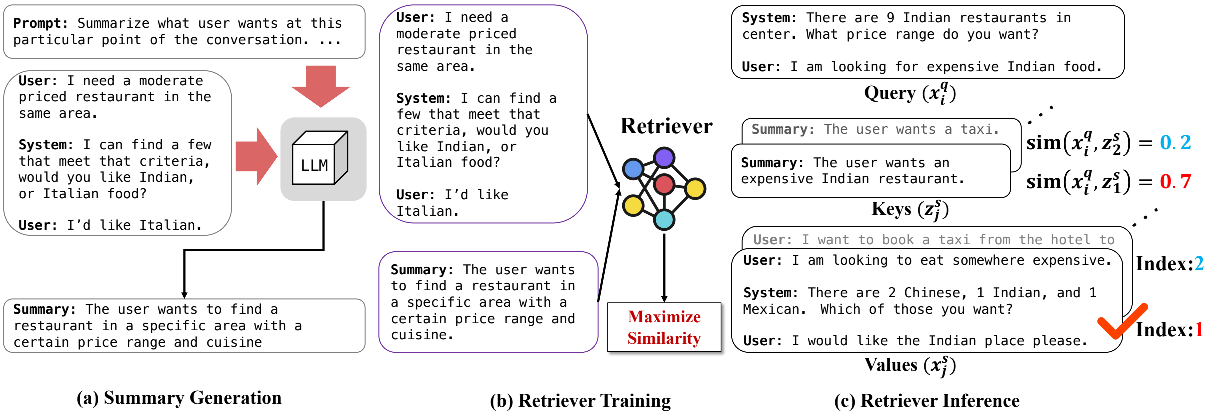
Note that in the proposed conversation retriever, the LLM-based conversation summarizer needs to be invoked for every test sample to generate the search query as shown in Figure 1-(a). To eliminate the extra inference cost, we propose to distill a light-weight conversation encoder which directly embeds a dialogue into a vector space similar to its summary, by maximizing their embedding similarity. The encoder is trained with large-scale dialogue-summary pairs generated by the conversation summarizer in an offline process. After training the model, as shown in Figure 1-(b), we can directly encode each dialogue into a query embedding for maximum inner product search. We call our conversation encoder CONVERSE, standing for **CON**versation embeddings for **VE**rsatile **R**etrieval with implicit **S**ummari**E**s. Next we explain the structure and training objective of CONVERSE.

3.1 Model

Preliminaries In our problem setup, we are given a set of unlabeled conversations between a user and system, denoted as $\mathcal{D}^u = \{\mathbf{x}_i\}_{i=1}^n$ where each conversation $\mathbf{x}_i$ consists of l_i utterances $(\mathbf{u}_{i,j}, \ldots, \mathbf{u}_{i,l_i})$ and each utterance $\mathbf{u}_{i,j}$ is a sequence of $T_{i,j}$ tokens $(x_{i,j,1}, \ldots, x_{i,j,T_{i,j}})$. As shown in Figure 2-(b), the training data of CONVERSE is prepared by invoking the conversation summarizer to generate a summary for each conversation $\mathbf{x}_i$, denoted as $\mathbf{z}_i$, which consists of T'_i tokens with $T'_i \ll \sum_{j=1}^{l_i} T_{i,j}$. We denote the dataset augmented with summaries as $\mathcal{D}^a = \{(\mathbf{x}_i, \mathbf{z}_i)\}_{i=1}^n$. For brevity, we omit the first subscript i if there is no ambiguity. Given the set of conversation-summary pairs, the goal is to train an encoder $f_\theta : \mathcal{V}^T \to \mathbb{R}^{T\times d}$ such that the similarity between a conversation and its summary is maximized, where $\mathcal{V}$ denotes a set of predefined tokens.

Conversation and Summary Embedding To match a conversation against a summary, we leverage the commonly used architecture in dense retrieval known as the dual encoder (Yih et al., 2011; Lee et al., 2019; Karpukhin et al., 2020a), where a conversation and a summary are encoded jointly for similarity comparison. State-of-the-art dual encoders (Khattab and Zaharia, 2020) represent each encoding as multiple vectors, typically the contextualized token vectors, to represent the text. These models largely improve the model expressiveness, and exhibit much stronger performance and robustness compared to their single-vector counterparts (Thakur et al., 2021). Based on it, we represent both the conversation embedding $f_\theta(\mathbf{x})$ and the summary embedding $f_\theta(\mathbf{z})$ as a matrix. While the summary encoder in the dual architecture can be directly integrated into off-the-shelf sentence encoders, our conversation encoder (CONVERSE) is designed to reflect the inductive bias of the summarization task.

CONVERSE Remember that the task of the conversation summarizer is to summarize the current user intent by grounding it to the conversation history. Hence the latest user input (the state delta) is most important and any past utterances irrelevant to the latest input should be dropped out.



Prompt: Summarize what user wants at this particular point of the conversation. ...

User: I need a moderate priced restaurant in the same area.

System: I can find a few that meet that criteria, would you like Indian, or Italian food?

User: I'd like Italian.

Summary: The user wants to find a restaurant in a specific area with a certain price range and cuisine

User: I need a moderate priced restaurant in the same area.

System: I can find a few that meet that criteria, would you like Indian, or Italian food?

User: I'd like Italian.

Summary: The user wants to find a restaurant in a specific area with a certain price range and cuisine.

System: There are 9 Indian restaurants in center. What price range do you want?

User: I am looking for expensive Indian food.

Summary: The user wants a taxi.

Summary: The user wants an expensive Indian restaurant.

User: I want to book a taxi from the hotel to

User: I am looking to eat somewhere expensive.

System: There are 2 Chinese, 1 Indian, and 1 Mexican. Which of those you want?

User: I would like the Indian place please.

(a) Summary Generation

(b) Retriever Training

(c) Retriever Inference

Figure 2: **Concept. (a)** Generating a summary of a dialogue with language model (LM). **(b)** Training the retriever to maximize a similarity between the dialogue and generated summary. **(c)** Given a test dialogue as a query, we retrieve the dialogue (value) of which summary (key) obtains the best similarity score with the query.

To reflect the nature of the summarization task, we explicitly model the grounding step between the latest user input $\mathbf{u}_l$ and past utterances $\mathbf{u}_1, \ldots, \mathbf{u}_{l-1}$ as a structural bias in CONVERSE. This is achieved with the introduction of a soft retrieval structure that softly retrieves past utterances or tokens which are relevant to the latest user input. Specifically, the soft retrieval is simulated with another neural network $g_\phi : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$, which outputs the relevance score of each token in the utterances $\mathbf{u}_1, \ldots, \mathbf{u}_{l-1}$ conditioned on the latest user utterance $\mathbf{u}_l$. Then, the relevance scores are used to downweight irrelevant token representations of the conversation $\mathbf{x}$:

$$\hat{f}_{\theta,\phi}(\mathbf{x}) = \begin{bmatrix} w_{1,1} f_\theta(\mathbf{x})_{1,1}^\top \\ \vdots \\ w_{l-1,T_{l-1}} f_\theta(\mathbf{x})_{l-1,T_{l-1}}^\top \\ f_\theta(\mathbf{x})_{l,1}^\top \\ \vdots \\ f_\theta(\mathbf{x})_{l,T_l}^\top \end{bmatrix} \in \mathbb{R}^{T \times d}$$

$$w_{j,t} = g_\phi(f_\theta(\mathbf{x})_{j,t}, s_l(\mathbf{x})) \in [0, 1] \qquad (1)$$

$$s_l(\mathbf{x}) = \frac{1}{T_l} \sum_{t=1}^{T_l} f_\theta(\mathbf{x})_{l,t} \in \mathbb{R}^d,$$

where $t \in \{1, \ldots, T_j\}$ for each $j \in \{1, \ldots, l-1\}$ and $f_\theta(\mathbf{x})_{j,t}$ is a contextual representation of t-th token in $\mathbf{u}_j$. Intuitively, an irrelevant token in the conversation history receives a small weight, reducing its contribution to the final similarity scoring against the summary. Conversely, a token in the latest user input always carries the highest weight 1 and contributes more to the similarity computation.

3.2 Training Objective

Given the conversation encoder $\hat{f}_{\theta,\phi}$ and summary encoder f_θ with a set of conversation and summary pairs $\mathcal{D}^a = \{(\mathbf{x}_i, \mathbf{z}_i)\}_{i=1}^n$, as illustrated in Figure 2-(b), we train the dual encoder to maximize the similarity between a dialogue and its summary with the contrastive loss (Henderson et al., 2017):

$$L(\theta, \phi; \mathcal{D}^a) = -\frac{1}{n} \sum_{(\mathbf{x},\mathbf{z}) \in \mathcal{D}^a} \log p_{\theta,\phi}(\mathbf{z}|\mathbf{x}) \qquad (2)$$

$$p_{\theta,\phi}(\mathbf{z}|\mathbf{x}) = \frac{\exp(\mathtt{sim}(\hat{f}_{\theta,\phi}(\mathbf{x}), f_\theta(\mathbf{z})))}{\exp(\sum_{(\mathbf{x}',\mathbf{z}') \in \mathcal{D}^a} \mathtt{sim}(\hat{f}_{\theta,\phi}(\mathbf{x}'), f_\theta(\mathbf{z}')))},$$

where $\mathtt{sim}$ is the multi-vector similarity function (Khattab and Zaharia, 2020), which computes the similarity between the conversation and its summary, denoted as $\mathtt{sim}(\hat{f}_{\theta,\phi}(\mathbf{x}), f_\theta(\mathbf{z}))$, by averaging maximum dot product between summary tokens and each conversation token as:

$$\frac{1}{T} \sum_{j=1}^{l} \sum_{t=1}^{T_j} \max_{t' \in \{1,\ldots,T'\}} \hat{f}_{\theta,\phi}(\mathbf{x})_{j,t}^\top f_\theta(\mathbf{z})_{t'}. \qquad (3)$$

In practice, due to computational costs, we sample a mini-batch $\mathcal{B} \subset \mathcal{D}^a$ for computing the denominator of the contrastive loss in equation 2.

3.3 Inference

In LLM-based DST, we are given a small support set of labeled dialogues $\mathcal{D}_1^s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^m$. The search keys can be pre-built offline by calling the conversation summarizer to generate a summary for each dialogue $\mathbf{x}_i^s$ from the support set $\mathcal{D}_1^s$, resulting in a set of (conversation, label, and summary) triplets denoted as $\mathcal{D}_2^s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s, \mathbf{z}_i^s)\}_{i=1}^m$. The search index is then built with the summary as the key, and a labeled conversation as the value. The summaries are encoded with the fine-tuned summary encoder described in Section 3.2.

During inference, for each conversation $\mathbf{x}_i^q$ from the test set $\mathcal{D}^q = \{\mathbf{x}_i^q\}_{i=1}^Q$, we embed the conversation with the CONVERSE encoder and compute its similarity with every search key using the similarity function in equation 3, i.e., $\mathtt{sim}(\hat{f}_{\theta,\phi}(\mathbf{x}_i^q), f_\theta(\mathbf{z}_j^s))$ for $j = 1, \ldots, m$. As shown in Figure 2-(c), the retriever ranks examples $(\mathbf{x}_j^s, \mathbf{y}_j^s)$ based on the similarity score and chooses the top-k exemplars. Finally, the retrieved exemplars are added to the prompt of the downstream LLM for dialog state generation.

4 Experiments

4.1 Experimental Setup

Common We evaluate LLM-based DST with the proposed conversation retriever on MultiWOZ 2.1 (Eric et al., 2020) and 2.4 (Ye et al., 2022).

To simulate few-shot scenario, we consider a support set of 100 labeled conversations as the default setting in our comparison. For each experimental run, we randomly sample 100 labeled conversations from the training data of MultiWOZ 2.1/2.4. The analysis of other support set sizes is deferred to an ablation study. During inference, we retrieve the top 5 examples from the support set. The examples along with a test conversation are inserted into the prompt, following Hu et al. (2022). This setting is applied to all comparisons. We use both GPT-Neo (Black et al., 2022) and LLaMA-7B/30B (Touvron et al., 2023) as the LLM for DST generation. For evaluation, we report average and standard deviation of Joint Goal Accuracy (JGA) and F1 score (Henderson et al., 2014) on all 7,368 test dialogues from MultiWOZ with three runs.

Baselines We compare the proposed conversation retriever with the following baselines.

1. IC-DST (Hu et al., 2022): It utilizes dialogue labels to construct positive and negative pairs for fine-tuning a pretrained SBERT (Reimers and Gurevych, 2019) or LinkBERT (Yasunaga et al., 2022) as a retriever. The retrieval key is a dialogue context, and the best dialogue context is reported to be previous dialogue state + current user input (which is better than a full dialogue).
2. SM2 (Chen et al., 2023): Similar to IC-DST, it fine-tunes SBERT on labeled dialogue data with contrastive loss, where conversations with partial matching slots or values are considered as positive samples. The retrieval key is a dialogue context similar to IC-DST.
3. GTR-T5-LARGE (Ni et al., 2022): It uses a T5 encoder, which is pretrained on large scale corpora for sentence representation, to compute the similarity between conversations for retrieving examples. The retrieval key is the full dialogue.
4. JINA-LARGE (Günther et al., 2023): Similar to GTR-T5, the pretrained sentence encoder Jina is used to compute similarity between conversations. The retrieval key is also the full dialogue.

Ours We use `gpt-3.5-turbo` (OpenAI, 2022) as the conversation summarizer, since it provides reliable summaries that satisfy the task requirement in the prompt (see human evaluation in 4.4 and the prompt specified in Appendix A). First, we evaluate the effectiveness of summary-based search key and query generation, using off-the-shelf retrievers GTR-T5-Large and Jina-Large, which are directly comparable with the baseline.

Second, we evaluate the distilled conversation encoder (CONVERSE). To train CONVERSE, we use the same conversation summarizer to generate a summary for every turn of every conversation from the full MultiWOZ training set, resulting in a total of 56,776 conversation-summary pairs. The parameters θ of the dual encoder f_θ and $\hat{f}_{\theta,\phi}$ are shared and initialized with LinkBERT (Yasunaga et al., 2022), and trained on the conversation-summary pairs for 20 epochs with the objective in equation 2. LinkBERT (Yasunaga et al., 2022) is chosen since we empirically find that it offers the best general-purpose weight of initialization. We use the AdamW optimizer (Loshchilov and Hutter, 2018) with learning rate $5 \cdot 10^{-5}$ and batch size 200. We use eight A100 GPUs for training the model.

4.2 Quantitative Results

Main Results The DST results are shown in Table 1 and Table 2. The first set of comparisons is between conversation retrieval with and without explicit query/key generation. We observe that using the summary as search keys/queries significantly improves the end-to-end (E2E) results, when evaluated with the same off-the-shelf retriever (GTR-T5 or Jina). The result is slightly behind IC-DST which fine-tunes the retriever with dialogue state information in the key. However, after introducing the distilled CONVERSE model, we achieve much

	MultiWOZ 2.1		MultiWOZ 2.4	
Model	JGA	F1	JGA	F1
GPT-Neo 2.7B (Black et al., 2022)				
IC-DST (SBERT)	$6.76_{\pm 0.87}$	$42.91_{\pm 2.87}$	$6.81_{\pm 1.05}$	$43.42_{\pm 3.18}$
IC-DST (LinkBERT)	$6.39_{\pm 1.72}$	$40.11_{\pm 3.30}$	$6.35_{\pm 1.14}$	$40.78_{\pm 3.10}$
SM2	$5.44_{\pm 0.27}$	$35.15_{\pm 1.80}$	$5.33_{\pm 0.76}$	$35.03_{\pm 1.42}$
GTR-T5	$4.77_{\pm 0.66}$	$28.58_{\pm 0.79}$	$4.66_{\pm 0.57}$	$28.50_{\pm 0.84}$
Jina	$5.11_{\pm 0.18}$	$30.93_{\pm 1.29}$	$5.16_{\pm 0.40}$	$30.84_{\pm 1.33}$
Sum. + GTR-T5	$6.16_{\pm 0.54}$	$40.60_{\pm 2.51}$	$6.01_{\pm 0.60}$	$40.40_{\pm 2.34}$
Sum. + Jina	$6.09_{\pm 0.71}$	$40.48_{\pm 2.62}$	$6.13_{\pm 0.77}$	$40.84_{\pm 2.95}$
CONVERSE	$\mathbf{8.07}_{\pm 0.62}$	$\mathbf{44.11}_{\pm 2.45}$	$\mathbf{7.85}_{\pm 0.65}$	$\mathbf{44.92}_{\pm 2.16}$
LLaMA-7B (Touvron et al., 2023)				
IC-DST (SBERT)	$18.30_{\pm 2.81}$	$69.51_{\pm 3.36}$	$18.57_{\pm 3.17}$	$70.37_{\pm 3.54}$
IC-DST (LinkBERT)	$18.09_{\pm 0.08}$	$69.41_{\pm 0.65}$	$18.97_{\pm 0.53}$	$70.29_{\pm 0.59}$
SM2	$15.23_{\pm 1.56}$	$64.36_{\pm 2.36}$	$15.01_{\pm 1.72}$	$65.12_{\pm 2.36}$
GTR-T5	$13.64_{\pm 0.16}$	$57.95_{\pm 0.46}$	$13.61_{\pm 0.43}$	$58.26_{\pm 0.44}$
Jina	$15.58_{\pm 0.58}$	$60.89_{\pm 0.41}$	$15.50_{\pm 1.02}$	$61.48_{\pm 0.34}$
Sum. + GTR-T5	$17.54_{\pm 0.34}$	$68.36_{\pm 0.48}$	$17.74_{\pm 0.68}$	$69.14_{\pm 0.77}$
Sum. + Jina	$17.85_{\pm 0.41}$	$68.70_{\pm 0.46}$	$18.37_{\pm 0.61}$	$69.65_{\pm 0.87}$
CONVERSE	$\mathbf{19.33}_{\pm 0.91}$	$\mathbf{71.48}_{\pm 1.50}$	$\mathbf{20.35}_{\pm 1.03}$	$\mathbf{72.45}_{\pm 1.52}$

Table 1: JGA and F1 using labeled 100 conversations with GPT-Neo-2.7B and LLaMA-7B.

	MultiWOZ 2.1		MultiWOZ 2.4	
Model	JGA	F1	JGA	F1
LLaMA-30B (Touvron et al., 2023)				
IC-DST (SBERT)	$25.41_{\pm 1.82}$	$77.82_{\pm 2.16}$	$26.01_{\pm 2.17}$	$79.01_{\pm 2.52}$
SM2	$22.86_{\pm 1.35}$	$74.73_{\pm 1.95}$	$23.46_{\pm 1.80}$	$75.78_{\pm 2.41}$
GTR-T5	$25.10_{\pm 0.33}$	$68.42_{\pm 1.93}$	$19.94_{\pm 2.40}$	$68.90_{\pm 2.22}$
Jina	$22.51_{\pm 0.92}$	$72.31_{\pm 1.01}$	$22.42_{\pm 1.18}$	$72.95_{\pm 0.93}$
Sum. + GTR-T5	$26.06_{\pm 0.47}$	$78.55_{\pm 0.35}$	$26.75_{\pm 0.93}$	$78.55_{\pm 0.35}$
Sum. + Jina	$25.10_{\pm 0.33}$	$78.07_{\pm 0.54}$	$25.81_{\pm 1.02}$	$78.98_{\pm 0.66}$
CONVERSE	$\mathbf{27.35}_{\pm 0.77}$	$\mathbf{79.75}_{\pm 0.95}$	$\mathbf{28.23}_{\pm 1.58}$	$\mathbf{80.45}_{\pm 0.55}$

Table 2: JGA and F1 of LLaMA-30B with 100 labeled conversations.

better E2E results than all baselines. Although our motivation of distilling a conversation encoder is to reduce the inference cost, it turns out that the light-weight model is also helpful in E2E performance. We hypothesize that the improved performance brought by CONVERSE is attributed to two factors. The first and foremost is that we leverage a dual encoder architecture to optimize the matching between conversation-summary pairs. This suggests that the retrieval component is optimized for the task-specific keys and values. A secondary explanation of the performance gain is that the conversation encoder avoids error propagation in explicit summary decoding and re-embedding. It should be noted that the above findings are consistent across different datasets (MultiWOZ 2.1/2.4) and language models (GPT-Neo and LLaMA-7B/30B).

Comparison Against Few-Shot Finetuning Recently, Mosbach et al. (2023) have shown that few-shot fine-tuning outperforms in-context learning in some settings, which makes people wonder how few-shot fine-tuning behaves with 100 labeled dialogues for dialogue state tracking tasks. To answer this question, we compare our method, CONVERSE, against one of the strongest few-shot finetuning methods, DS2 (Shin et al., 2022), using BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) language models. As shown in Table 3, our method, CONVERSE, outperforms the few-shot fine-tuning method, DST. Note that the BART-based DS2 severely overfits to the small labeled dataset and the T5-based model performs worse than the in-context learning method, even though T5 model is pretrained on an additional large-scale labeled dialogue summarization dataset, SAMSum (Gliwa et al., 2019).

	JGA	
Model	MWZ-2.1	MWZ-2.4
DS2 + BART-Large	$7.60_{\pm 2.17}$	$5.86_{\pm 4.52}$
DS2 + T5-Large	$17.71_{\pm 1.84}$	$19.08_{\pm 1.23}$
CONVERSE + LLaMA-7B	$19.33_{\pm 0.91}$	$20.35_{\pm 1.03}$
CONVERSE + LLaMA-30B	$\mathbf{27.35}_{\pm 0.77}$	$\mathbf{28.23}_{\pm 1.58}$

Table 3: Comparison against few-shot finetuning methods.

	MultiWOZ 2.1		MultiWOZ 2.4	
Model	JGA	F1	JGA	F1
LLaMA-7B (Touvron et al., 2023)				
IC-DST (SBERT)	$12.52_{\pm 0.68}$	$62.11_{\pm 0.38}$	$12.43_{\pm 0.09}$	$62.45_{\pm 0.87}$
CONVERSE	$\mathbf{14.05}_{\pm 0.58}$	$\mathbf{63.37}_{\pm 1.53}$	$\mathbf{14.23}_{\pm 0.48}$	$\mathbf{64.18}_{\pm 1.47}$

Table 4: Out-of domain generalization using 100 labeled conversations with LLaMA-7B.

Out-of Domain Generalization To verify our hypothesis that our unsupervised retriever CONVERSE generalizes better to unseen domain than supervised methods, we hold out the hotel domain from the MultiWOZ dataset and train the retrievers, IC-DST and CONVERSE on the remaining four domains: train, restaurant, taxi, and attraction. Then we evaluate the performance of the few-shot in-context learning with the retrievers on test examples from the unseen domain, hotel. As shown in Table 4, our model CONVERSE outperforms IC-DST by a large margin, which empirically validates that our unsupervised retriever generalizes better to unseen domain than the supervised one.

4.3 Ablation Study

Size of Support Set We empirically study the size of the support set (labeled dialogues) in the conversation retrieval task. Notably, a smaller support set requires less annotation effort from the domain owner, placing more emphasis on general-

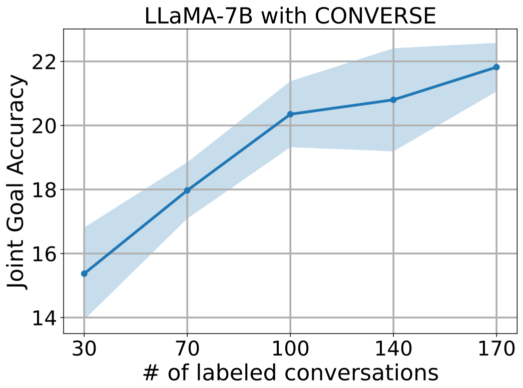


Figure 3: JGA of LLaMA-7B with CONVERSE as a function of the number of labeled data.

	JGA	
Model	**MultiWOZ 2.1**	**MultiWOZ 2.4**
CONVERSE + Rerank	19.86 ± 1.22	20.65 ± 1.28
CONVERSE	19.33 ± 0.91	20.35 ± 1.03

Table 5: Ablations of different ranking with LLaMA-7B.

ization to unseen dialogue structures. In contrast, a larger support set contradicts the fundamental motivation behind few-shot learning, but it is likely to improve the E2E accuracy, as more test dialogue structures are observable from the exemplars. In Figure 3, we plot the JGA of LLaMA-7B with CONVERSE on varying sizes of the support set constructed from MultiWOZ. As we expect, the JGA increases as the number of labeled conversation increases, even though we do not fine-tune the retriever with any labeled conversations.

Summary vs. state delta The conversation summary we adopt in this work concludes the user's current intent when the dialogue takes place. A limitation is that the summary does not directly highlight the state delta carried by the latest user input. As a remedy, we consider a multi-key and query retrieval setup, where we use both the summary and the latest user input as search keys and queries. More specifically, we first retrieve 20 dialogues with CONVERSE and re-rank the 20 dialogues based on the similarity of the latest utterance between the test sample and the support examples, using the pre-trained GTR-T5-Large. As shown in Table 5, re-ranking with the latest user utterance yields marginal performance gains. In future work, we aim to explore a better way of summarizing the conversation structure that reflects both the joint intent and the latest user input.

4.4 Qualitative Results

Visualization of history grounding As described in equation 1, CONVERSE softly retrieves conversation history based on the latest user utterance. Specifically, the network g_ϕ outputs a relevance score between 0 and 1 for each token of the conversation history. In Figure 4, we visualize this relevance score of each token in the history. The tokens with darker blue color indicates a higher weight, which are considered to be more relevant to the latest input.

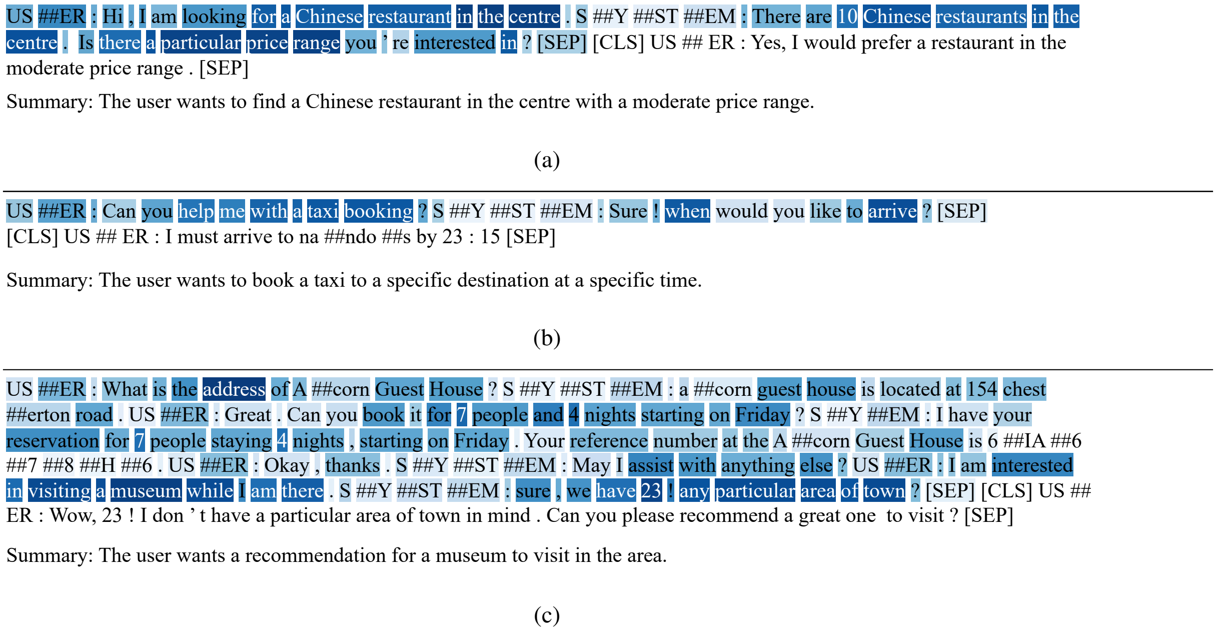
Conversation
USER: I need some tourist information please. I need to know about a hotel called the Arbury lodge guest house. **SYSTEM**: The Arbury lodge guest house is in the north area and has a moderate price range. $\cdots$ **USER**: I would like to book a stay for 3 people for 2 nights starting from Tuesday. **USER**: I am also looking to eat somewhere expensive, in the south area of town. $\vdots$ **USER**: I will also need a taxi , please. **SYSTEM**: Where would you like your taxi to pick you up and drop you off? **USER**: I want to be picked up at the hotel and dropped off at the restaurant.
Summary: The user wants to book a taxi to be picked up at a specific location and dropped off at another.

Table 6: The LLM successfully summarize the conversation based on the latest user utterance.

The examples in Figure 4 shows that the model successfully focuses on relevant part of history. For the first example in Figure 4a, the user wants to search for a Chinese restaurant in the center with moderate price range. The model assigns large weights to the tokens related to "Chinese", "center", and "price". Similarly, the tokens relevant to booking a taxi gets larger weights in Figure 4b. For the last example in Figure 4c, the model pays attention to the tokens related to a museum and ignores many irrelevant ones.

Human Evaluation on Conversation Summarizer The success of CONVERSE is highly dependent on the output quality of the conversation summarizer, which are used as labels for encoder distillation. We conduct human evaluation of 135 summaries generated by the conversation summarizer, namely `gpt-3.5-turbo`. Specifically, three human judges are asked to assess whether the generated summaries are consistent with the instructions in the prompt in Table 9. The results indicate that 90.3% of the 135 summaries are deemed consistent with the given prompt.

Examples of the generated summaries are shown in Table 6 and 7. For the first example, the model generates the summary about booking a taxi. It is



US ##ER : Hi , I am looking for a Chinese restaurant in the centre . S ##Y ##ST ##EM : There are 10 Chinese restaurants in the centre . Is there a particular price range you ’ re interested in ? [SEP] [CLS] US ## ER : Yes, I would prefer a restaurant in the moderate price range . [SEP]

Summary: The user wants to find a Chinese restaurant in the centre with a moderate price range.

(a)

US ##ER : Can you help me with a taxi booking ? S ##Y ##ST ##EM : Sure ! when would you like to arrive ? [SEP] [CLS] US ## ER : I must arrive to na ##ndo ##s by 23 : 15 [SEP]

Summary: The user wants to book a taxi to a specific destination at a specific time.

(b)

US ##ER : What is the address of A ##corn Guest House ? S ##Y ##ST ##EM : a ##corn guest house is located at 154 chest ##erton road . US ##ER : Great . Can you book it for 7 people and 4 nights starting on Friday ? S ##Y ##EM : I have your reservation for 7 people staying 4 nights , starting on Friday . Your reference number at the A ##corn Guest House is 6 ##IA ##6 ##7 ##8 ##H ##6 . US ##ER : Okay , thanks . S ##Y ##ST ##EM : May I assist with anything else ? US ##ER : I am interested in visiting a museum while I am there . S ##Y ##ST ##EM : sure , we have 23 ! any particular area of town ? [SEP] [CLS] US ## ER : Wow, 23 ! I don ’ t have a particular area of town in mind . Can you please recommend a great one to visit ? [SEP]

Summary: The user wants a recommendation for a museum to visit in the area.

(c)

Figure 4: **Visualization of importance scores.** Tokens with darker blue gets larger weights based on the latest user utterance.

Conversation
⋮ **SYSTEM**: Booking was successful. The table will be reserved for 15 minutes. · · · **USER**: Great. 1 more thing. Can you book a taxi between the 2 places? I would like to arrive at the restaurant in time for my reservation
Summary: The user wants to book a taxi to travel between two specific locations.

Table 7: A failure case of summarization with the LLM.

noteworthy that the model focuses on the latest user utterance while disregarding previous user requests for hotel and restaurant reservation. For the second example, the model misses out on the arrival time for generating the summary. Identification and correction of such errors are topics we will explore in future work. We include more examples in Appendix B.

Retrieved Exemplars In Table 8, we show the top three most similar examples retrieved by CONVERSE. In this example, the user asks to find an expensive Indian restaurant and a retriever needs to retrieve conversations about a restaurant. Indeed, our CONVERSE retriever assigns high similarity scores to pairs of the target conversation and summaries about finding a restaurant. Note that the language model (LLaMA-7B) with in-context learning successfully generalizes to decode test slot values from the exemplars, though the retrieved exemplars consist of values for food or price range, which are different from the target conversation.

5 Related Work

Dialog State Tracking Most of existing works on DST train a supervised model with large-scale labeled datasets (Wu et al., 2019; Zhang et al., 2020; Peng et al., 2021; Lin et al., 2020; Lee et al., 2021; Zhao et al., 2022; Kim et al., 2020; Heck et al., 2020; Hosseini-Asl et al., 2020; Ham et al., 2020; Cheng et al., 2020; Platanios et al., 2021). However, a supervised model does not scale well to new domains or annotation schemas. To address the problem, several recent works explore few-shot DST (Wu et al., 2020; Li et al., 2021; Gao et al., 2020; Lin et al., 2021; Campagna et al., 2020; Su et al., 2022). Most related are the works of Hu et al. (2022); Chen et al. (2023), who adopt in-context learning with LLM for dialog state generation. The work demonstrated the few-shot generalization ability of LLM applied to DST without parameter updates, but the dialog retriever is still fine-tuned with in-domain data.

Another work related to ours is Shin et al. (2022), which formulates DST as a summarization task. The authors train a T5 language model to decode text summaries, which are then transformed into dialog states with heuristic rules. Different from their work, we do not aim to alter the target of DST as summaries but rather our goal is to enable effective conversation retrieval.

Retrieval Our work mainly focuses on retrieving relevant conversations for in-context learning (Liu et al., 2022). There is a vast number of papers (Karpukhin et al., 2020b; Khattab and Za-

Target Conversation	⋮ **SYSTEM**: There are 9 Indian restaurants in centre what price range do you want? **USER**: I am looking for expensive Indian food.
Gold Label	restaurant-food: indian, restaurant-pricerange: expensive
Prediction	restaurant-food: indian, restaurant-pricerange: expensive
Exemplar #1	⋮ **USER**: I am also looking to eat somewhere expensive, in the south area of town. **SYSTEM**: There are 2 Chinese, 1 Indian , 1 Italian, and 1 Mexican restaurants. Which of those would you like? **USER**: I would like the Italian place please. **Summary**: The user wants to find a restaurant in a specific area with a certain price range and cuisine. **Label**: restaurant-food: italian.
Exemplar #2	⋮ **USER**: Actually, I also need a moderate priced restaurant in the same area. **SYSTEM**: I can find a few that meet that criteria, would you like Indian, or Italian food? **USER**: Well, everyone said it's my choice, so I think I would like Italian. **Summary**: The user wants to find a moderate priced restaurant in a specific area with a specific cuisine. **Label**: restaurant-food: italian.
Exemplar #3	⋮ **SYSTEM**: Good news I was able to get this for you. Reference i4dxhdjl. Can I help you find other things to do in the area as well ? **USER**: I am also looking to eat somewhere expensive, in the south area of town. **Summary**: The user wants to find a restaurant with an expensive price range in a specific area of town. **Label**: restaurant-pricerange: expensive, restaurant-area: south

Table 8: Given the target conversation, we show the top 3 most similar examples retrieved by our model CONVERSE.

haria, 2020; Izacard et al., 2022; Santhanam et al., 2022) proposing neural network based retrievers which encode queries and keys into low dimensional vectors and compute similarities between them. Hu et al. (2022); Chen et al. (2023) propose to utilize slots and values to represent a long history of conversation for retrieval. However, in order to train the retriever, their approaches require labeled dialogue data to construct positive and negative conversations for each query conversation. Recently, Ravfogel et al. (2023) retrieve texts based on abstract descriptions generated by a LLM.

6 Conclusion

The contribution of this work is twofold. First, we proposed an effective way of retrieving conversations in LLM-based DST with conversation summaries as search keys and queries. We then improved the efficiency of the retrieval system by distilling a conversation encoder capable of embedding a conversation into a vector space similar to its summary. This eliminates the cost of decoding an actual summary for each test sample during inference. We validated our CONVERSE encoder for LLM-based DST in a real few-shot setting with 100 conversations in the support set. Results showed that CONVERSE consistently improved both the efficiency and the performance of few-shot DST when using different LLMs, outperforming previous LLM-based DST baselines that rely on annotated dialogues for retriever fine-tuning.

Acknowledgement We would like to express our sincere gratitude to Dominik Wagner for the numerous helpful pieces of feedbacks.

References

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode# 5–Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136.

Paweł 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*, pages 5016–5026.

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.

Derek Chen, Kun Qian, and Zhou Yu. 2023. Stabilized in-context learning with pre-trained language models for few shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1506–1519.

Jianpeng Cheng, Devang Agrawal, Héctor Martínez Alonso, Shruti Bhargava, Joris Driesen, Federico Flego, Dain Kaplan, Dimitri Kartsaklis, Lin Li, Dhivya Piraviperumal, et al. 2020. Conversational semantic parsing for dialog state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8107–8117.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 422–428.

Shuyang Gao, Sanchit Agarwal, Di Jin, Tagyoung Chung, and Dilek Hakkani-Tur. 2020. From machine reading comprehension to dialogue state tracking: Bridging the gap. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 79–89.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79.

Michael Günther, Louis Milliken, Jonathan Geuter, Georgios Mastrapas, Bo Wang, and Han Xiao. 2023. Jina embeddings: A novel set of high-performance sentence embedding models. *arXiv preprint arXiv:2307.11224*.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 583–592.

Daniel Hardt. 1997. An empirical approach to vp ellipsis. *Computational Linguistics*, 23(4):525–541.

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.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

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. *Advances in Neural Information Processing Systems*, 33:20179–20191.

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

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020a. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020b. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

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.

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.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Seanie Lee, Hae Beom Lee, Juho Lee, and Sung Ju Hwang. 2022. Sequential reptile: Inter-task gradient alignment for multilingual learning. In *International Conference on Learning Representations*.

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

Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian McAuley. 2021. Zero-shot generalization in dialog state tracking through generative question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1063–1074.

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

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. Mintl: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405.

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

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. 2023. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In *Findings of the Association for Computational Linguistics: ACL*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855.

OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt.

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.

Emmanouil Antonios Platanios, Adam Pauls, Subhro Roy, Yuchen Zhang, Alexander Kyte, Alan Guo, Sam Thomson, Jayant Krishnamurthy, Jason Wolfe, Jacob Andreas, et al. 2021. Value-agnostic conversational semantic parsing. 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 3666–3681.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint conference on EMNLP and CoNLL-shared task*, pages 1–40.

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*, 21:140:1–140:67.

Shauli Ravfogel, Valentina Pyatkin, Amir DN Cohen, Avshalom Manevich, and Yoav Goldberg. 2023. Retrieving texts based on abstract descriptions. *arXiv preprint arXiv:2305.12517*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.

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.

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

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Chien-Sheng Wu, Steven CH Hoi, and Caiming Xiong. 2020. Improving limited labeled dialogue state tracking with self-supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4462–4472.

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.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016.

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

Wen-tau Yih, Kristina Toutanova, John C Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 247–256.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9604–9611.

Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. Description-driven task-oriented dialog modeling. *arXiv preprint arXiv:2201.08904*.

Appendix

A Prompt for Summarization

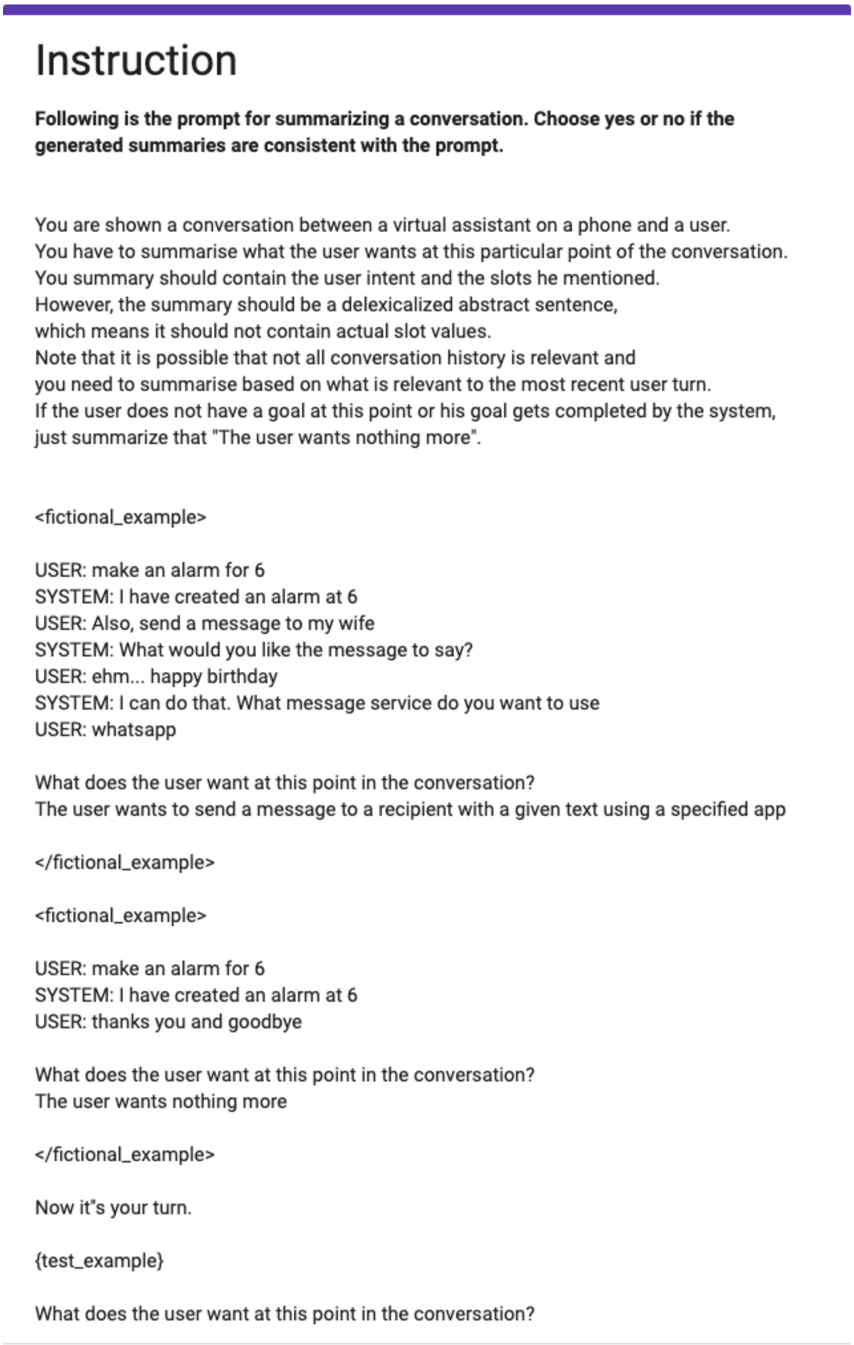
In Table 9, we provide an instruction and two examples that specify how to summarize a conversation based on the latest user utterance. The first example demonstrate how to ignore irrelevant history — request for setting an alarm and focus on sending a message. The second example is to generate "The user wants nothing more" instead of summarizing the conversation.

B Instruction for Human Evaluation

We conduct human evaluation of 135 summaries generated by the conversation summarizer, `gpt-3.5-turbo`. As shown in Figure 5, three human judges are given an instruction for the evaluation and they are asked to assess whether the generated summaries are consistent with the prompt specified in Table 9.

C Example of Summaries

In Tables 10 to 14, we provide more summaries generated by the LLM, `gpt-3.5-turbo`.



Instruction

Following is the prompt for summarizing a conversation. Choose yes or no if the generated summaries are consistent with the prompt.

You are shown a conversation between a virtual assistant on a phone and a user.
You have to summarise what the user wants at this particular point of the conversation.
You summary should contain the user intent and the slots he mentioned.
However, the summary should be a delexicalized abstract sentence,
which means it should not contain actual slot values.
Note that it is possible that not all conversation history is relevant and
you need to summarise based on what is relevant to the most recent user turn.
If the user does not have a goal at this point or his goal gets completed by the system,
just summarize that "The user wants nothing more".

<fictional_example>

USER: make an alarm for 6
SYSTEM: I have created an alarm at 6
USER: Also, send a message to my wife
SYSTEM: What would you like the message to say?
USER: ehm... happy birthday
SYSTEM: I can do that. What message service do you want to use
USER: whatsapp

What does the user want at this point in the conversation?
The user wants to send a message to a recipient with a given text using a specified app

</fictional_example>

<fictional_example>

USER: make an alarm for 6
SYSTEM: I have created an alarm at 6
USER: thanks you and goodbye

What does the user want at this point in the conversation?
The user wants nothing more

</fictional_example>

Now it"s your turn.

{test_example}

What does the user want at this point in the conversation?

Figure 5: A screenshot of the instruction for human evaluation on summaries generated by gpt-3.5-turbo.

You are shown a conversation between a virtual assistant on a phone and a user. You have to summarise what the user wants at this particular point of the conversation. You summary should contain the user intent and the slots he mentioned. However, the summary should be a delexicalized abstrast sentence, which means it should not contain actual slot values. Note that it is possible that not all conversation history is relevant and you need to summarise based on what is relevant to the most recent user turn.
If the user does not have a goal at this point or his goal gets completed by the system,
just summarize that "The user wants nothing more".

<fictional_example>
USER: make an alarm for 6
SYSTEM: I have created an alarm at 6
USER: Also, send a message to my wife
SYSTEM: What would you like the message to say?
USER: ehm... happy birthday
SYSTEM: I can do that. What message service do you want to use
USER: whatsapp

What does the user want at this point in the conversation?
The user wants to send a message to a recipient with a given text using a specified app
</fictional_example>

<fictional_example>
USER: make an alarm for 6
SYSTEM: I have created an alarm at 6
USER: thanks you and goodbye

What does the user want at this point in the conversation?
The user wants nothing more
</fictional_example>

Now it's your turn.
{test_example}
What does the user want at this point in the conversation?

Table 9: Prompt with two exemplars for summarizing a conversation.

Conversation
USER: I am looking for a hotel in the east , and i would like free wifi please. **SYSTEM**: I have found express by holiday inn Cambridge. 1517 Norman way, Coldhams business park **USER**: Does that hotel include free parking and have 2 stars? **SYSTEM**: No, thank you , that s all I need. **USER**: Sure thing what's the area and or name? **SYSTEM**: Thank you for contacting Cambridge Towninfo centre. Have a nice trip! Good Bye! **USER**: Actually, I also need a moderate priced restaurant in the same area
Summary: The user wants to find a moderate priced restaurant in a specific area.

Table 10: Example of conversation summary.

Conversation
USER: I need some tourist information please. I need to know about a hotel called the Arbury lodge guest house. **SYSTEM**: The arbury lodge guest house is in the north area and has a moderate price range. It is 4 stars and has free parking and internet. **USER**: I would like to book a stay for 3 people for 2 nights starting from Tuesday. **SYSTEM**: Good news I was able to get this for you. Reference i4dxhdjl. Can I help you find other things to do in the area as well? **USER**: I am also looking to eat somewhere expensive, in the south area of town. **SYSTEM**: there are 2 Chinese, 1 Indian, 1 Italian, and 1 Mexican restaurants. Which of those would you like? **USER**: I would like the Italian place please. **SYSTEM**: Frankie and Benny s is an expensive restaurant which is expensive would you like for me to book it? **USER**: yeah , book for 3 people on tuesday at 18:30 and give me the reference number
Summary: The user wants to book a table at an expensive Italian restaurant for 3 people at a specific date and time and receive a reference number.

Table 11: Example of conversation summary.

Conversation
USER: I would like to get to Cambridge by train, but I am afraid I need to travel on Sunday. Will that be possible? **SYSTEM**: From where are you departing? **USER**: I am leaving from Broxbourne. **SYSTEM**: There are 19 trains running from Broxbourne to Cambridge on Sunday.
Summary: The user wants to find a train from a specific location to a specific destination arriving before a certain time.

Table 12: Example of conversation summary.

Conversation
USER: Please help me reserve a taxi from saint catharine s college **SYSTEM**: Sure! When would you like to arrive? **USER**: I would like to arrive by 01:15. **SYSTEM**: Great! And what is your destination? **USER**: I am going to Loch Fyne.
Summary: The user wants to reserve a taxi from a specific location to a specific destination at a specific time.

Table 13: Example of conversation summary.

Conversation
USER: Are there any concert halls in the west? **SYSTEM**: I am sorry there are no listings for concert halls in the west. Is there something else I can assist you with today? **USER**: What about colleges in the west **SYSTEM**: There are 5 colleges in that area. 3 have no entrance fees. Would you like additional information on any of these? **USER**: No thank you. I am looking for information El Shaddai hotel. **SYSTEM**: The El Shaddai is a guest house in the centre part of town. It has free internet , parking and has 0 stars. Would you like me to book it for you? **USER**: Could you? That would be great. There are 5 of us and we plan to arrive on Thursday. We'd like to stay for 5 nights.
Summary: The user wants to book a hotel for a group of 5 people for a specific duration of time.

Table 14: Example of conversation summary.