Multi-Grained Knowledge Retrieval for End-to-End Task-Oriented Dialog

Fanqi Wan[1], **Weizhou Shen**[1], **Ke Yang**[1], **Xiaojun Quan**[1]*, **Wei Bi**[2]*
[1]School of Computer Science and Engineering, Sun Yat-sen University, China
[2]Tencent AI Lab
{wanfq, shenwzh3, yangk59}@mail2.sysu.edu.cn,
quanxj3@mail.sysu.edu.cn, victoriabi@tencent.com

Abstract

Retrieving proper domain knowledge from an external database lies at the heart of end-to-end task-oriented dialog systems to generate informative responses. Most existing systems blend knowledge retrieval with response generation and optimize them with direct supervision from reference responses, leading to suboptimal retrieval performance when the knowledge base becomes large-scale. To address this, we propose to decouple knowledge retrieval from response generation and introduce a multi-grained knowledge retriever (MAKER) that includes an entity selector to search for relevant entities and an attribute selector to filter out irrelevant attributes. To train the retriever, we propose a novel distillation objective that derives supervision signals from the response generator. Experiments conducted on three standard benchmarks with both small and large-scale knowledge bases demonstrate that our retriever performs knowledge retrieval more effectively than existing methods. Our code has been made publicly available.[1]

1 Introduction

When task-oriented dialog (TOD) systems try to accomplish a task such as restaurant reservations and weather reporting for human users, they generally resort to an external knowledge base (KB) to retrieve relevant entity information for generating an informative system response. Conventional pipeline systems comprise several modules such as dialogue state tracking and dialogue policy learning that require annotations for training, where intermediate predictions such as belief state can be used for the retrieval. By contrast, end-to-end task-oriented dialog (E2E-TOD) systems aim to eliminate the dependence on intermediate annotations and generate the response end-to-end (Wu et al., 2019). Apparently, knowledge retrieval is at the core of this task, which is non-trivial as no gold labels are available for training a retriever. Arguably, this problem has limited the performance of existing E2E-TOD systems considering that substantial progress has been made in natural language generation.

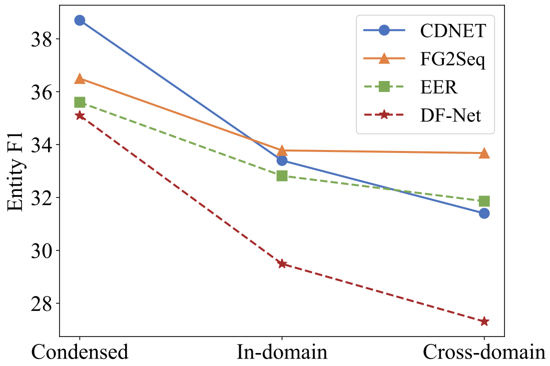


Figure 1: Performance of four end-to-end task-oriented dialog systems on MultiWOZ 2.1 when knowledge bases of different sizes are used. The evaluation metric is Entity F1 scores of entities in generated responses. "Condensed" means that each dialog is associated with a small-sized knowledge base, which is the default setting of many current systems. "In-domain" means that each dialog corresponds to a knowledge base of the same domain, while "Cross-domain" means that all dialogs share the same large-scale cross-domain knowledge base provided in the dataset.

Roughly, existing approaches for knowledge retrieval in E2E-TOD systems can be divided into three categories. First, the knowledge base can be embedded into a memory network and queried with the representations of dialogue context (Madotto et al., 2018; Qin et al., 2020; Raghu et al., 2021). Second, the serialized knowledge base records can be encoded together with dialog context by pre-trained language models (Xie et al., 2022; Wu et al., 2022; Tian et al., 2022). Third, the knowledge base can be embedded into model parameters through data augmentation to support implicit knowledge retrieval (Madotto et al., 2020; Huang et al., 2022). These approaches generally

*Corresponding authors

[1]https://github.com/18907305772/MAKER

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 11196–11210
July 9-14, 2023 ©2023 Association for Computational Linguistics

blend knowledge retrieval and response generation and train them by the supervision of reference responses, which has two limitations. First, the system response usually consists of pure language tokens and KB-related tokens (e.g., hotel names and phone numbers), and it is challenging to train a good retriever from the weak supervision of reference responses. Second, the systems may become inefficient when the scale of the knowledge base grows large. Our preliminary study[2] in Figure 1 confirms that when a large-scale cross-domain knowledge base is given, existing dialog systems suffer significant performance degradation.

In this paper, we propose a novel Multi-grAined KnowlEdge Retriever (MAKER) for E2E TOD systems to improve the acquisition of knowledge for response generation. The retriever decouples knowledge retrieval from response generation and introduces an entity selector and an attribute selector to select relevant entities and attributes from the knowledge base. Then, the response generator generates a system response based on the dialogue context and the multi-grained retrieval results. The retriever is trained by distilling knowledge from the response generator using the cross-attention scores of KB-related tokens in the response. We train the entity selector, attribute selector, and response generator jointly in an end-to-end manner.

We compare our system with other E2E TOD systems on three benchmark datasets (Eric et al., 2017; Wen et al., 2017; Eric et al., 2020). Empirical results show that our system achieves state-of-the-art performance when either a small or a large-scale knowledge base is used. Through in-depth analysis, we have several findings to report. First, our retriever shows great advantages over baselines when the size of knowledge bases grows large. Second, of the two selectors, the entity selector plays a more important role in the retriever. Third, our system consistently outperforms baselines as different numbers of records are retrieved, and works well even with a small number of retrieval results.

2 Related Work

2.1 End-to-End Task-Oriented Dialog

Existing approaches for knowledge retrieval in end-to-end task-oriented dialog systems can be divided into three categories. First, the knowledge base (KB) is encoded with memory networks, and KB records are selected using attention weights between dialogue context and memory cells. Mem2seq (Madotto et al., 2018) uses multi-hop attention over memory cells to select KB tokens during response generation. KB-Retriever (Qin et al., 2019) retrieves the most relevant entity from the KB by means of attention scores to improve entity consistency in the system response. GLMP (Wu et al., 2019) introduces a global-to-local memory pointer network to retrieve relevant triplets to fill in the sketch response. CD-NET (Raghu et al., 2021) retrieves relevant KB records by computing a distillation distribution based on dialog context.

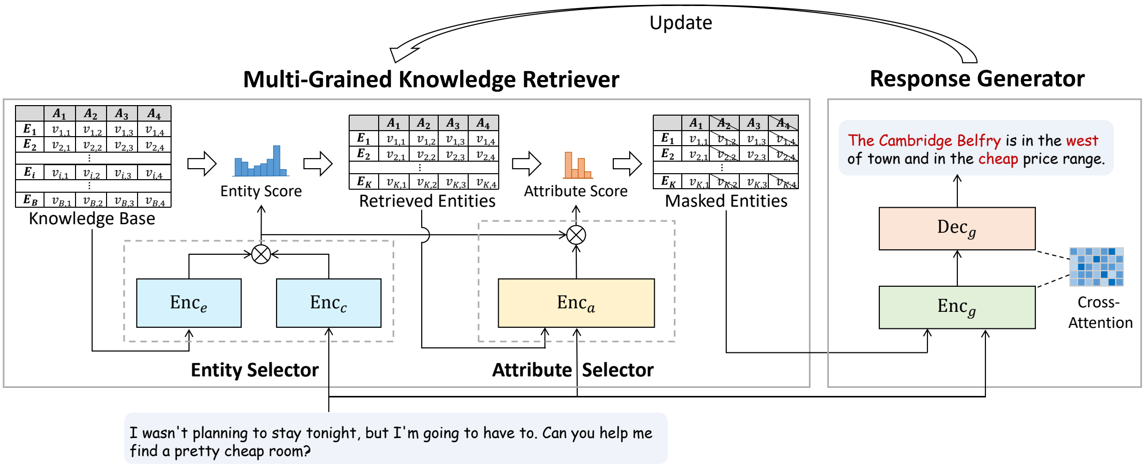
Second, the concatenation of knowledge base and dialogue context is taken as input for pre-trained language models. UnifiedSKG (Xie et al., 2022) uses a unified text-to-text framework to generate system responses. DialoKG (Rony et al., 2022) models the structural information of knowledge base through knowledge graph embedding and performs knowledge attention masking to select relevant triples. Q-TOD (Tian et al., 2022) proposes to rewrite dialogue context to generate a natural language query for knowledge retrieval.

Third, the knowledge base is stored in model parameters for implicit retrieval during response generation. GPT-KE (Madotto et al., 2020) proposes to embed the knowledge base into pre-trained model parameters through data augmentation. ECO (Huang et al., 2022) first generates the most relevant entity with trie constraint to ensure entity consistency in the response. However, these methods generally blend entity retrieval and response generation during response generation, which leads to sub-optimal retrieval performance when large-scale knowledge bases are provided.

2.2 Neural Retriever

With the success of deep neural networks in various NLP tasks, they have also been applied to information retrieval. One of the mainstream approaches is to employ a dual-encoder architecture (Yih et al., 2011) to build a retriever. Our work is mostly inspired by the retrieval methods in question answering. To train a retriever with labeled question-document pairs, DPR (Karpukhin et al., 2020) uses in-batch documents corresponding to other questions together with BM25-retrieved documents as negative samples for contrastive learning. To train a retriever with only question-answer pairs instead of question-document pairs, which is a weakly su-

[2] More details of this study are given in Appendix B.



The Cambridge Belfry is in the west of town and in the cheap price range.

I wasn't planning to stay tonight, but I'm going to have to. Can you help me find a pretty cheap room?

Figure 2: The overview of our end-to-end task-oriented dialog system, which consists of a knowledge retriever and a response generator. The retriever is further divided into an entity selector and an attribute selector to retrieve multi-grained knowledge, and optimized by distilling knowledge from the response generator.

pervised learning problem, researchers propose to distill knowledge from the answer generator to train the retriever iteratively (Yang and Seo, 2020; Izacard and Grave, 2020). Other researchers try to train the retriever and generator in an end-to-end manner. REALM (Guu et al., 2020), RAG (Lewis et al., 2020), and EMDR^2 (Singh et al., 2021) propose to train the retriever end-to-end through maximum marginal likelihood. Sachan et al. (2021) propose to combine unsupervised pre-training and supervised fine-tuning to train the retriever. Motivated by these works, we propose a multi-grained knowledge retriever trained by distilling knowledge from the response generator in E2E-TOD systems.

3 Methods

In this section, we first describe the notations and outline our method, and then introduce the knowledge retriever and response generator in detail.

3.1 Notations

Given a dialog $\mathcal{D} = \{U_1, R_1, ..., U_T, R_T\}$ of T turns, where U_t and R_t are the t-th turn user utterance and system response, respectively. We use C_t to represent the dialog context of the t-th turn, where $C_t = \{U_1, R_1, ..., U_{t-1}, R_{t-1}, U_t\}$. An external knowledge base (KB) is provided in the form of a set of entities, i.e., $\mathcal{K} = \{E_1, E_2, ..., E_B\}$, where each entity E_i is composed of N attribute-value pairs, i.e., $E_i = \{a^1, v_i^1, ..., a^N, v_i^N\}$. End-to-end task-oriented dialog systems take dialogue context C_t and knowledge base $\mathcal{K}$ as input and generate an informative response R_t.

3.2 System Overview

The architecture of our end-to-end task-oriented dialog system is shown in Figure 2. At each turn of conversation, our system resorts to a Multi-grAined KnowlEdge Retriever (MAKER) to retrieve a set of entities from the external knowledge base. Then, the response generator takes as input the retrieved entities together with the dialog context and generates a natural language response. The overall system is optimized in an end-to-end manner without the need for intermediate annotations.

The novelty of MAKER lies in that it decouples knowledge retrieval from response generation and provides multi-grained knowledge retrieval by means of an entity selector and an attribute selector. Specifically, the knowledge base is first encoded with an entity encoder Enc_e at entity level. Then, the dialogue context is encoded with a context encoder Enc_c and used to retrieve a set of relevant entities from the knowledge base, which is referred to as entity selection. Next, irrelevant attributes are filtered out with an attribute selector based on the interaction of dialog context and retrieved entities, where another encoder Enc_a is used. Finally, each retrieved entity is concatenated with the dialog context and passed to a generator encoder Enc_g to obtain their representations, based on which the generator decoder Dec_g produces a system response. To train the retriever, the cross-attention scores from KB-related tokens in the reference response to each retrieved entity are used as supervision signals to update the entity selector, while the attribute selector is trained by using the occurrences of attribute values in the dialogue as pseudo-labels. To better measure the relationship between entities and response, the whole training process involves two stages. First, the warming-up stage only trains the attribute selector and the response generator, with the entity selector not updated. As the above training converges, the second stage starts to update the entity selector together with other modules using cross-attention scores from the response generator.

3.3 Knowledge Retriever

In this section, we introduce the entity selector, attribute selector, and the training of the retriever.

Entity Selector To support large-scale knowledge retrieval, we model the entity selector as a dual-encoder architecture, where one encoder *Enc*$_c$ is used to encode the dialogue context and another encoder *Enc*$_e$ is to encode each entity (row) of the knowledge base, both into a dense vector. To encode an entity, we concatenate the attribute-value pairs of this entity into a sequence and pass it to *Enc*$_e$. The selection score $s_{t,i}$ for entity E_i is defined as the dot product between the context vector and the entity vector as:

$$s_{t,i} = \text{Enc}_c(C_t)^T \text{Enc}_e(E_i). \tag{1}$$

Then, the top-K entities are obtained by:

$$\mathcal{E}_t = \text{Top}K(s_{t,i}) = \{E_1, ..., E_K\}. \tag{2}$$

Retrieving the top-K entities can be formulated as maximum inner product search (MIPS), which can be accelerated to sub-linear time using efficient similarity search libraries such as FAISS (Johnson et al., 2019). We implement *Enc*$_c$ and *Enc*$_e$ with a pre-trained language model and allow them to share weights, where the final "[CLS]" token representation is used as the encoder output. Existing studies suggest that initializing *Enc*$_c$ and *Enc*$_e$ with BERT weights may lead to collapsed representations and harm the retrieval performance. Therefore, following KB-retriever (Qin et al., 2019), we initialize them by pre-training with distant supervision.[3]

Since the entity selector is updated by knowledge distillation, recalculating the embeddings of all entities after each update introduces considerable computational cost. Therefore, we follow EMDR^2 (Singh et al., 2021) to update the embeddings of all entities after every 100 training steps.

Attribute Selector To remove irrelevant attributes and values from the retrieved entities for finer-grained knowledge, we design an attribute selector as follows. We first concatenate dialog context C_t with each entity $E_i \in \mathcal{E}_t$ and encode them with an attribute encoder *Enc*$_a$, which is also a pre-trained language model. Then, the final "[CLS]" token representation of *Enc*$_a$ is extracted and mapped into a N-dimensional vector by a feed-forward network (FFN) for attribute scoring:

$$\mathbf{a}_{t,i} = \text{FFN}(\text{Enc}_a([C_t; E_i])), \tag{3}$$

where each element in $\mathbf{a}_{t,i} \in \mathbb{R}^N$ represents the importance of the corresponding attribute.

Note that $\mathbf{a}_{t,i}$ only measures the importance of attributes in E_i. To obtain the accumulated importance, we calculate the sum of $\mathbf{a}_{t,i}$ over all retrieved entities weighted by entity selection score $s_{t,i}$:

$$\mathbf{a}_t = \sigma(\sum_{i=1}^{K} s_{t,i}\mathbf{a}_{t,i}), \tag{4}$$

where σ represents the sigmoid function.

Finally, the attributes whose importance scores in $\mathbf{a}_t$ are greater than a pre-defined threshold τ are selected to construct an attribute subset. The retrieved entities clipped with these attributes are treated as multi-grained retrieval results denoted by $\hat{\mathcal{E}}_t$. Specifically, we obtain $\hat{\mathcal{E}}_t$ by masking irrelevant attribute-value pairs in each retrieved entity of $\mathcal{E}_t$.

$$\hat{\mathcal{E}}_t = \text{Clip}(\mathcal{E}_t, \mathbf{a}_t, \tau) = \{\hat{E}_1, ..., \hat{E}_K\}. \tag{5}$$

To train the attribute selector, we design an auxiliary multi-label classification task. The pseudo-label is a N-dimensional 0-1 vector $\mathbf{b}_t$ constructed by checking whether any value of an attribute in $\hat{\mathcal{E}}_t$ appears in dialogue context C_t or system response R_t. Then, we define a binary cross-entropy loss $\mathcal{L}_{att}$ for this classification task as:

$$\mathcal{L}_{att} = \text{BCELoss}(\mathbf{a}_t, \mathbf{b}_t). \tag{6}$$

Updating The entity selector is updated by distilling knowledge from the response generator as supervision signals. Specifically, since only KB-related tokens in the response are directly connected to the knowledge base, we regard the cross-attention scores from these tokens to each retrieved entity as the knowledge to distill. The rationality behind this is that the cross-attention scores can usually measure the relevance between each entity and the response. Supposing response R_t contains M KB-related tokens, we denote the cross-attention scores from each KB-related token to

[3] More pre-training details are given in Appendix C.

entity $\hat{E}_i$ by $\mathbf{C}_{t,i} \in \mathbb{R}^{|\hat{E}_i| \times M \times L}$, where $|\hat{E}_i|$ represents the number of tokens in $\hat{E}_i$ and L is the number of decoder layers. Then, we calculate an accumulated score for entity $\hat{E}_i$ as:

$$\hat{c}_{t,i} = \sum_{j=1}^{|\hat{E}_i|} \sum_{m=1}^{M} \sum_{l=1}^{L} \mathbf{C}_{t,i,j,m,l}. \tag{7}$$

Then, $\hat{c}_{t,i}$ is softmax-normalized to obtain a cross-attention distribution $\mathbf{c}_t$ over the K retrieved entities to reflect their importance for the response.

Finally, we calculate the KL-divergence between the selection scores $\mathbf{s}_t$ of retrieved entities and cross-attention distribution $\mathbf{c_t}$ as the training loss:

$$\mathcal{L}_{ent} = \mathcal{D}_{KL}(\mathbf{s}_t || \mathbf{c_t}). \tag{8}$$

3.4 Response Generator

Inspired by Fusion-in-Decoder (Izacard and Grave, 2020) in open-domain question answering, we employ a modified sequence-to-sequence structure for the response generator to facilitate direct interaction between dialog context and retrieved entities.

Generator Encoder Each entity $\hat{E}_i$ in $\hat{\mathcal{E}}_t$ is first concatenated with dialog context C_t and encoded into a sequence of vector representations $\mathbf{H}_{t,i}$:

$$\mathbf{H}_{t,i} = \text{Enc}_g([C_t; \hat{E}_i]), \tag{9}$$

where Enc_g represents the encoder of the response generator. Then, the representations of all retrieved entities are concatenated into $\mathbf{H}_t$:

$$\mathbf{H}_t = [\mathbf{H}_{t,1}; ...; \mathbf{H}_{t,K}]. \tag{10}$$

Generator Decoder Taking $\mathbf{H}_t$ as input, the generator decoder Dec_g produces the system response token by token. During this process, the decoder not only attends to the previously generated tokens through self-attention but also attends to the dialogue context and retrieved entities by cross-attention, which facilitates the generation of an informative response. The probability distribution for each response token in R_t is defined as:

$$P(R_{t,i}) = \text{Dec}_g(R_{t,i} | R_{t,<i}, \mathbf{H}_t). \tag{11}$$

We train the response generator by the standard cross-entropy loss as:

$$\mathcal{L}_{gen} = \sum_{i=1}^{|R_t|} -\text{log} P(R_{t,i}), \tag{12}$$

where $|R_t|$ denotes the length of R_t.

Lastly, the overall loss of the system is the sum of entity selection loss $\mathcal{L}_{ent}$, attribute selection loss $\mathcal{L}_{att}$, and response generation loss $\mathcal{L}_{gen}$:

$$\mathcal{L} = \mathcal{L}_{ent} + \mathcal{L}_{att} + \mathcal{L}_{gen}. \tag{13}$$

3.5 Discussions

Although deriving much inspiration from open-domain question answering (QA) (Izacard and Grave, 2020), where the labels for retrieval are also not available, the scenario of this work is quite different. One major difference is that the answer in open-domain QA is completely from the external source of knowledge, while some responses and tokens in dialog systems may not be relevant to the external knowledge base. That means dialog systems need to accommodate both dialog context and external knowledge and generate a fluent and informative natural language response, making this task thornier than open-domain QA.

The main differences between our MAKER and existing knowledge retrieval methods in task-oriented dialog systems are twofold. First, MAKER decouples knowledge retrieval from response generation and provides multi-grained knowledge retrieval of both entities and attributes. The retrieval results are explicitly passed to the generator to produce a system response. Second, MAKER is trained by distilling knowledge from the response generator for supervision, which varies from existing attention-based approaches.

4 Experimental Settings

4.1 Datasets

We evaluate our system on three multi-turn task-oriented dialogue datasets: MultiWOZ 2.1 (MWOZ) (Eric et al., 2020), Stanford Multi-Domain (SMD) (Eric et al., 2017), and CamRest (Wen et al., 2017). Each dialog in these datasets is associated with a condensed knowledge base, which contains all the entities that meet the user goal of this dialog. For MWOZ, each condensed knowledge base contains 7 entities. For SMD and CamRest, the size of condensed knowledge bases is not fixed: it ranges from 0 to 8 with a mean of 5.95 for SMD and from 0 to 57 with a mean of 1.93 for CamRest. We follow the same partitions as previous work (Raghu et al., 2021). The statistics of these datasets are shown in Appendix A.

BLEU (Papineni et al., 2002) and Entity F1 (Eric et al., 2017) are used as the evaluation metrics. BLEU measures the fluency of a generated response based on its n-gram overlaps with the gold response. Entity F1 measures whether the generated response contains correct knowledge by micro-averaging the precision and recall scores of attribute values in the generated response.

4.2 Implementation Details

We employ BERT (Devlin et al., 2019) as the encoder of our entity selector and attribute selector, and employ T5 (Raffel et al., 2020) to implement the response generator. All these models are fine-tuned using AdamW optimizer (Loshchilov and Hutter, 2018) with a batch size of 64. We train these models for 15k gradient steps with a linear decay learning rate of 10^{-4}. We conduct all experiments on a single 24G NVIDIA RTX 3090 GPU and select the best checkpoint based on model performance on the validation set. More detailed settings can be found in Appendix E.

4.3 Baselines

We compare our system with the following baselines, which are organized into three categories according to how they model knowledge retrieval.

Memory network: These approaches embed the knowledge base into a memory network and query it with the representation of dialog context, including DSR (Wen et al., 2018), KB-Retriever (Qin et al., 2019), GLMP (Wu et al., 2019), DF-Net (Qin et al., 2020), EER (He et al., 2020b), FG2Seq (He et al., 2020a), CDNET (Raghu et al., 2021), and GraphMemDialog (Wu et al., 2022).

Direct fusion: These approaches encode serialized knowledge base records together with dialog context by pre-trained language models, including DialoKG (Rony et al., 2022), UnifiedSKG (Xie et al., 2022), and Q-TOD (Tian et al., 2022).

Implicit retrieval: These approaches embed the knowledge base into model parameters by data augmentation to provide implicit retrieval during response generation, including GPT-2+KE (Madotto et al., 2020) and ECO (Huang et al., 2022).

5 Results and Analysis

In this section, we first show the overall performance of the evaluated systems given a condensed knowledge base for each dialog. Then, we compare them with a more practical setting in which a large-scale knowledge base is provided. We also conduct an in-depth analysis of the proposed retriever. More experiments are presented in the appendix.

5.1 Overall Results

The overall results are shown in Table 1. We observe that our system with T5-Large as the backbone model achieves the state-of-the-art (SOTA) performance on MWOZ and SMD. Specifically, on MWOZ our system surpasses the previous SOTA, namely Q-TOD, by 1.15 points in BLEU and 4.11 points in Enity F1. On SMD, the improvements over Q-TOD are 4.58 points in BLEU and 0.19 points in Enity F1. On CamRest, our system only achieves the best performance in BLEU but underperforms the best-performing DialoKG slightly. The reason behind this phenomenon is that many dialogues in CamRest contain extremely small knowledge bases, with only 1-2 entities, leaving little space for our retriever to show its advantage.

Note that with the same backbone generator (T5-Base/T5-Large), our system surpasses Q-TOD even though it relies on human annotations to train a query generator for knowledge retrieval. The possible reason is that while the retriever of Q-TOD is independent of response generation, ours is trained and guided by knowledge distillation from response generation. Moreover, in addition to retrieving entities from the knowledge base, our retriever also conducts a fine-grained attribute selection.

5.2 Large-Scale Knowledge Base

The experiments in Section 5.1 are conducted with each dialog corresponding to a condensed knowledge base. Although most previous systems are evaluated in this setting, it is not practical to have such knowledge bases in real scenes, where the systems may need to retrieve knowledge from a large-scale knowledge base. Therefore, we examine the performance of several well-recognized E2E TOD systems by implementing them on a large-scale cross-domain knowledge base (referred to as "full knowledge base") on MWOZ and CamRest, respectively, where the knowledge base is constructed by gathering the entities for all dialogs in the original dataset.[4] The results are shown in Table 2.

We observe that our system outperforms baselines by a large margin when the full knowledge

[4] Since the training scripts of Q-TOD is not released, we directly use its open-source checkpoint (T5-Large) and conduct inference with the full knowledge base.

Model	MWOZ		SMD		CamRest	
	BLEU	Entity F1	BLEU	Entity F1	BLEU	Entity F1
DSR (Wen et al., 2018)	9.10‡	30.00‡	12.70†	51.90†	18.30†	53.60†
KB-Retriever (Qin et al., 2019)	-	-	13.90	53.70	18.50	58.60
GLMP (Wu et al., 2019)	6.90‡	32.40‡	13.90‡	60.70‡	15.10§	58.90§
DF-Net (Qin et al., 2020)	9.40	35.10	14.40	62.70	-	-
GPT-2+KE (Madotto et al., 2020)	15.05	39.58	17.35	59.78	18.00	54.85
EER (He et al., 2020b)	13.60§	35.60§	17.20§	59.00§	19.20§	65.70§
FG2Seq (He et al., 2020a)	14.60§	36.50§	16.80§	61.10§	20.20§	66.40§
CDNET (Raghu et al., 2021)	11.90	38.70	17.80	62.90	21.80	68.60
GraphMemDialog (Wu et al., 2022)	14.90	40.20	18.80	64.50	22.30	64.40
ECO (Huang et al., 2022)	12.61	40.87	-	-	18.42	71.56
DialoKG (Rony et al., 2022)	12.60	43.50	20.00	65.90	23.40	**75.60**
UnifiedSKG (T5-Base) (Xie et al., 2022)	-	-	17.41	66.45	-	-
UnifiedSKG (T5-Large) (Xie et al., 2022)	13.69*	46.04*	17.27	65.85	20.31*	71.03*
Q-TOD (T5-Base) (Tian et al., 2022)	-	-	20.14	68.22	-	-
Q-TOD (T5-Large) (Tian et al., 2022)	17.62	50.61	21.33	71.11	23.75	74.22
Ours (T5-Base)	17.23	53.68	24.79	69.79	25.04	73.09
Ours (T5-Large)	**18.77**	**54.72**	**25.91**	**71.30**	**25.53**	74.36

Table 1: Overall results of E2E TOD systems with condensed knowledge bases on MWOZ, SMD, and CamRest. The best scores are highlighted in bold, and the second-best scores are underlined. †, ‡, §, $*$ indicates that the results are cited from (Qin et al., 2019), (Qin et al., 2020), (Raghu et al., 2021), and (Tian et al., 2022), respectively.

Model	MWOZ		CamRest	
	BLEU	Entity F1	BLEU	Entity F1
DF-Net	6.45	27.31	-	-
EER	11.60	31.86	20.61	57.59
FG2Seq	10.74	33.68	19.20	59.35
CDNET	10.90	31.40	16.50	63.60
Q-TOD	16.67	47.13	21.44	63.88
Ours (T5-Base)	16.25	50.87	**26.19**	72.09
Ours (T5-Large)	**18.23**	**52.12**	25.34	**72.43**

Table 2: Overall results of E2E TOD systems with a large-scale knowledge base on MWOZ and CamRest, respectively. The best scores are highlighted in bold, and the second-best scores are underlined.

Model	BLEU	Entity F1
Ours$_{\text{condensed}}$	17.23	53.68
w/o distillation	16.21 ($\downarrow$1.02)	51.05 ($\downarrow$2.63)
w/o attr_selector	15.72 ($\downarrow$1.51)	51.76 ($\downarrow$1.92)
w/o ent_selector	16.07 ($\downarrow$1.16)	50.67 ($\downarrow$3.01)
Ours$_{\text{full}}$	16.25	50.87
w/o distillation	15.85 ($\downarrow$0.40)	48.28 ($\downarrow$2.59)
w/o attr_selector	15.40 ($\downarrow$0.85)	48.55 ($\downarrow$2.32)

Table 3: Results of ablation study on MWOZ with T5-base, where "w/o" means without, "distillation" denotes distillation from response generation, "attr_selector" denotes the attribute selector, and "ent_selector" denotes the entity selector.

base is used. In addition, there are two other observations. First, comparing the results in Table 1 and Table 2, we note existing systems suffer a severe performance deterioration when the full knowledge base is used. For example, the Enity F1 score of DF-Net drops by 7.79 points on MWOZ, while our system only drops by 2.81/2.6 points. Second, our system with the full knowledge base still outperforms other systems when they use condensed knowledge bases, which is easier to retrieve. These observations verify the superiority of our system when applied to a large-scale knowledge base as well as the feasibility of applying it to real scenes.

5.3 Ablation Study

We conduct an ablation study of our retriever MAKER with both condensed and full knowledge bases on MWOZ, and show the results in the first and the second blocks of Table 3, respectively.

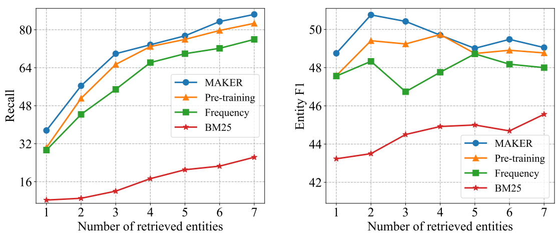
When condensed knowledge bases are used, the system suffers obvious performance drops with the removal of distillation (*w/o* distillation) or entity selection (*w/o* ent_selector). This indicates that despite the quality of condensed knowledge bases, our retriever can further learn to distinguish between the entities by distilling knowledge from the response generator. Besides, the performance of the system drops when the attribute selector is aban-

Retrieval Method	BLEU	Entity F1	Recall@7
Oracle	16.17	51.45	100.00
MAKER	17.18	49.05	86.47
Pre-training	16.67	48.77	82.71
Frequency	16.60	48.00	75.94
BM25	16.21	45.56	26.32

Table 4: Comparison of different retrieval methods on the full knowledge base. *Oracle* refers to using the condensed knowledge base for each dialog as the retrieval result. *Frequency* means measuring the relevance by the frequency of attribute values occurring in the dialogue context. *BM25* measures the relevance using the BM25 score between dialogue context and each entity.

Method	BLEU	Entity F1
Weighted	16.25	50.87
Average	16.46	48.81
Threshold	16.25	50.87
Top-K	16.31	46.89
All	15.40	48.55

Table 5: Comparison of attribute selection methods for MAKER on the full knowledge base. The upper two rows are methods for accumulating attribute importance scores across retrieved entities, and the bottom three rows are methods for filtering out irrelevant attributes.



(a) Recall (b) Entity F1

Figure 3: Performance of different retrieval methods as the number of retrieved entities changes on the full knowledge base in Recall (a) and Entity F1 (b) scores.

doned (*w/o* attr_selector), showing that attribute selection is also indispensable in the retriever.

When the full knowledge base is used, entity selection is more necessary for the system. Therefore, we only ablate the distillation component and the attribute selector. The results show that the system suffers significant performance degradation when distillation is removed (*w/o* distillation). Attribute selection is also shown important as the performance drops upon it is removed (*w/o* attr_selector).

5.4 Comparison of Retrieval Methods

To further demonstrate the effectiveness of our multi-grained knowledge retriever, we compare different retrieval methods on the full knowledge base of MWOZ. Specifically, we first retrieve the top-K entities with different retrieval methods and employ the same response generator to generate the system response. Moreover, we propose a new metric, i.e., Recall@7, to measure whether the suggested entities in the system response appear in the 7 retrieved entities. As shown in Table 4, the proposed retriever achieves the best performance compared with other methods except Oracle, which uses condensed knowledge bases without retrieval, in both generation metrics (BLEU, Entity F1) and the retrieval metric (Recall@7).

To investigate the effect of different numbers of retrieved entities on system performance, we report the Entity F1 and Recall@x scores of the above retrieval methods as the number of entities changes, while Oracle is not included because we cannot rank its entities. We observe in Figure 3(a) that the Recall@x scores for all methods improve as the number of entities grows, while our retriever consistently achieves the best performance. In Figure 3(b), we observe no positive correlation between the Entity F1 score and the number of entities, suggesting that noisy entities may be introduced as the number of entities increases. We can also observe that the number of entities corresponding to the peak of the Entity F1 scores varies for different methods, while our retriever only requires a small number of entities to reach the peak performance.

5.5 Attribute Selection Methods

In Section 3.3, we calculate an accumulated importance score for each attribute weighted by entity selection scores to determine which attributes are preserved based on a given threshold. In Table 5, we compare different methods for accumulating the attribute scores as well as different approaches for filtering out irrelevant attributes. It can be observed that direct averaging rather than weighting by entity selection scores hurts the Entity F1 score. This indicates that the retriever can select attributes more appropriately based on the selection scores of retrieved entities. We also observe that using top-K instead of a threshold to select attributes leads to a lower Entity F1 score than preserving all attributes. We believe the reason is that the number of attributes to be selected varies for each dialogue context, and therefore simply selecting the top-K attributes results in sub-optimal attributes.

6 Conclusion

We propose a novel multi-grained knowledge retriever (MAKER) for end-to-end task-oriented dialog systems. It decouples knowledge retrieval from response generation and introduces an entity selector and an attribute selector to acquire multi-grained knowledge from the knowledge base. The retriever is trained by distilling knowledge from the response generator. Empirical results show that our system achieves state-of-the-art performance when either a small or a large-scale knowledge base is provided for each dialog. Through in-depth analysis, our retriever shows great advantages over baselines when the size of knowledge bases grows large. Of the two selectors, the entity selector is shown to be more prominent in the retriever.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No.62176270), the Guangdong Basic and Applied Basic Research Foundation (No.2023A1515012832), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No. 2017ZT07X355), and the Tencent AI Lab Rhino-Bird Focused Research Program. We thank Yingqi Gao and Canbin Huang for their efforts in the preliminary experiments.

Limitations

Our system employs a modified sequence-to-sequence architecture to implement the response generator. Since the length of dialogue context increases as the dialogue continues, the generator needs to input multiple long dialogue contexts to the encoder simultaneously, each for a retrieved entity. This may cause redundancy in the input and lowers the proportion of KB-related information. We will explore more efficient architectures for the response generator in future work.

Ethics Statement

All the experiments are conducted on publicly available datasets, which don't include any private information. Our work doesn't involve identity characteristics or any gender and racial discrimination.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Zhenhao He, Yuhong He, Qingyao Wu, and Jian Chen. 2020a. Fg2seq: Effectively encoding knowledge for end-to-end task-oriented dialog. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8029–8033. IEEE.

Zhenhao He, Jiachun Wang, and Jian Chen. 2020b. Task-oriented dialog generation with enhanced entity representation. In *INTERSPEECH*, pages 3905–3909.

Guanhuan Huang, Xiaojun Quan, and Qifan Wang. 2022. Autoregressive entity generation for end-to-end task-oriented dialog. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 323–332.

Gautier Izacard and Edouard Grave. 2020. Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick 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)*, pages 6769–6781.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

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

Andrea Madotto, Samuel Cahyawijaya, Genta Indra Winata, Yan Xu, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Learning knowledge bases with parameters for task-oriented dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2372–2394.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478, Melbourne, Australia. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Libo Qin, Yijia Liu, Wanxiang Che, Haoyang Wen, Yangming Li, and Ting Liu. 2019. Entity-consistent end-to-end task-oriented dialogue system with KB retriever. 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 133–142, Hong Kong, China. Association for Computational Linguistics.

Libo Qin, Xiao Xu, Wanxiang Che, Yue Zhang, and Ting Liu. 2020. Dynamic fusion network for multi-domain end-to-end task-oriented dialog. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6344–6354, 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*, 21:1–67.

Dinesh Raghu, Atishya Jain, Sachindra Joshi, et al. 2021. Constraint based knowledge base distillation in end-to-end task oriented dialogs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5051–5061.

Md Rashad Al Hasan Rony, Ricardo Usbeck, and Jens Lehmann. 2022. Dialokg: Knowledge-structure aware task-oriented dialogue generation. *arXiv preprint arXiv:2204.09149*.

Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. 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 6648–6662.

Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. *Advances in Neural Information Processing Systems*, 34:25968–25981.

Xin Tian, Yingzhan Lin, Mengfei Song, Siqi Bao, Fan Wang, Huang He, Shuqi Sun, and Hua Wu. 2022. Q-tod: A query-driven task-oriented dialogue system. *arXiv preprint arXiv:2210.07564*.

Haoyang Wen, Yijia Liu, Wanxiang Che, Libo Qin, and Ting Liu. 2018. Sequence-to-sequence learning for task-oriented dialogue with dialogue state representation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3781–3792, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019. Global-to-local memory pointer networks for task-oriented dialogue. In *International Conference on Learning Representations*.

Jie Wu, Ian G Harris, and Hongzhi Zhao. 2022. Graphmemdialog: Optimizing end-to-end task-oriented dialog systems using graph memory networks.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *arXiv preprint arXiv:2201.05966*.

Sohee Yang and Minjoon Seo. 2020. Is retriever merely an approximator of reader? *arXiv preprint arXiv:2010.10999*.

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.

A Statistics of Datasets

The statistics of the datasets are shown in Table 6.

Dataset	Domains	# Dialogues
		Train/Val/Test
MWOZ (Eric et al., 2020)	Restaurant, Attraction, Hotel	1839/117/141
SMD (Eric et al., 2017)	Navigate, Weather, Schedule	2425/302/304
CamRest (Wen et al., 2017)	Restaurant	406/135/135

Table 6: Statistics of the three datasets.

B Preliminary Study

The detailed results of our preliminary study for condensed, in-domain, and cross-domain knowledge bases are shown in Table 7. The results of baseline models on condensed knowledge bases are cited from (Raghu et al., 2021). We produce their results on in-domain and cross-domain knowledge bases by using the officially released code.

Model	Condensed		In-Domain		Cross-Domain	
	BLEU	Entity F1	BLEU	Entity F1	BLEU	Entity F1
DF-Net	9.40	35.10	7.24	29.49	6.45	27.31
EER	13.60	35.60	11.44	32.82	11.60	31.86
FG2Seq	14.60	36.50	10.53	33.78	10.74	33.68
CDNET	11.90	38.70	11.70	33.40	10.90	31.40

Table 7: Comparison of end-to-end task-oriented dialog systems with different sizes of knowledge bases.

C Pre-training for Entity Selector

Given a dialogue context and the system response, we use the entity with the most occurrences of its attribute values in the dialogue context and system response as the label. Then we apply supervised contrastive learning for optimization (Gao et al., 2021). Specifically, the positive example of a dialogue context is the corresponding labeled entity, while the negative examples are the labeled entities of other examples in the same mini-batch. Then, we employ the InfoNCE loss as the training objective to pull close the sentence representations of positive samples and push away that of negative samples. We conduct this pre-training on the MWOZ and CamRest datasets. Since the knowledge base of the SMD dataset is strictly specific to each dialog, we cannot collect a global knowledge base from the dialogs. Thus, the pre-training is not conducted on SMD. The hyperparameters for the pre-training are shown in Table 8.

Hyperparameters	MWOZ	CamRest
Optimizer	AdamW	AdamW
Batch size	128	108
Epoch	10	15
Learning rate schedule	Linear	Linear
Learning rate	5e-5	5e-5
Weight decay	0.01	0.01
Temperature	0.05	0.05
Max length	128	128
Pooler type	CLS	CLS
Pooler dimension	128	128

Table 8: Hyperparameter setting for pre-training our entity selector on the full knowledge base of MWOZ and CamRest datasets, respectively.

D Domain-Wise Results

We report the domain-wise results with condensed knowledge bases on MWOZ and SMD in Table 9 and Table 10, respectively. The results of baseline models are cited from (Raghu et al., 2021), (Rony et al., 2022), and (Tian et al., 2022).

E More Implementation Details

The hyperparameters of our system with condensed and full knowledge bases are shown in Table 11 and Table 12, respectively. Our method has three contributions: knowledge distillation, entity selection, and attribute selection. We list the application of these contributions with condensed and full knowledge base in Table 13 and Table 14, respectively.

F Case Study

In Figure 4, we provide a dialogue example from the MWOZ dataset. We can observe that, for a given user utterance, our system can retrieve entities that satisfy the user goal, while masking irrelevant attributes. Then, it generates appropriate system responses. Note that when the user goal changes, e.g., in the second turn of this case when the user wants a cheap restaurant, our retriever can retrieve the corresponding one, with the attribute of price range being preserved.

Model	**BLEU**	**Entity F1**	**Hotel**	**Attraction**	**Restaurant**
DSR	9.10	30.00	27.10	28.00	33.40
GLMP	6.90	32.40	28.10	24.40	38.40
DF-Net	9.40	35.10	30.60	28.10	40.90
GPT-2+KE	15.00	39.60	33.40	43.30	37.10
EER	13.60	35.60	35.70	43.00	34.20
FG2Seq	14.60	36.50	34.40	37.20	38.90
CDNET	11.90	38.70	36.30	38.90	41.70
GraphMemDialog	14.90	40.20	36.40	48.80	42.80
DialoKG	12.60	43.50	37.90	39.80	46.70
Q-TOD (T5-Large)	17.62	50.61	45.25	54.81	55.78
Ours (T5-Large)	**18.77**	**54.72**	**46.97**	**65.08**	**62.12**

Table 9: Domain-wise performance on MWOZ.

Model	**BLEU**	**Entity F1**	**Schedule**	**Weather**	**Navigate**
DSR	12.70	51.90	52.10	50.40	52.00
GLMP	13.90	59.60	70.20	58.00	54.30
DF-Net	14.40	62.70	73.10	57.60	57.90
GPT-2+KE	17.40	59.80	72.60	57.70	53.50
EER	17.20	59.00	71.80	57.80	52.50
FG2Seq	16.80	61.10	73.30	57.40	56.10
CDNET	17.80	62.90	75.40	61.30	56.70
GraphMemDialog	18.80	64.50	75.90	62.30	56.30
DialoKG	20.00	65.90	77.90	**72.70**	58.40
Q-TOD (T5-Large)	21.33	71.11	**81.42**	69.18	**62.91**
Ours (T5-Large)	**25.91**	**71.30**	78.56	72.69	62.15

Table 10: Domain-wise performance on SMD.

Hyperparameters	**MWOZ**		**SMD**		**CamRest**	
	T5-Base	**T5-Large**	**T5-Base**	**T5-Large**	**T5-Base**	**T5-Large**
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Batch size	2	1	2	2	2	2
Gradient accumulation steps	32	64	32	32	32	32
Training gradient steps	1500	1500	1500	1500	1000	1000
Learning rate schedule	Linear	Linear	Linear	Linear	Linear	Linear
Entity selector learning rate	5e-5	1e-4	1e-4	1e-4	1e-4	1e-4
Attribute selector learning rate	5e-5	1e-4	1e-4	1e-4	1e-4	1e-4
Response generator learning rate	1e-4	1e-4	1e-4	1e-4	1e-4	7e-5
Weight decay	0.01	0.01	0.01	0.01	0.01	0.01
Gradient clipping	1.0	1.0	1.0	1.0	1.0	1.0
Entity selector max length	128	128	128	128	128	128
Attribute selector max context length	200	200	200	200	200	200
Attribute selector max kb length	100	100	200	200	100	100
Response generator max context length	200	200	200	200	200	200
Response generator max kb length	100	100	200	200	100	100
Max output length	64	64	128	128	64	64
Top-K retrieval entities	6	7	8	8	6	4
Attribute selection threshold	0.1	0.1	0.0	0.0	0.0	0.0
Distillation start gradient steps	625	938	1500	1500	750	750

Table 11: Hyperparameter settings of our system when condensed knowledge bases are used on the MWOZ, SMD, and CamRest datasets.

Hyperparameters	MWOZ		CamRest	
	T5-Base	T5-Large	T5-Base	T5-Large
Optimizer	AdamW	AdamW	AdamW	AdamW
Batch size	2	1	2	1
Gradient accumulation steps	32	64	32	64
Training gradient steps	1500	1500	1500	1500
Learning rate schedule	Linear	Linear	Linear	Linear
Entity selector learning rate	5e-5	1e-4	1e-4	1e-4
Attribute selector learning rate	5e-5	1e-4	1e-4	1e-4
Response generator learning rate	1e-4	1e-4	1e-4	1e-4
Weight decay	0.01	0.01	0.01	0.01
Gradient clipping	1.0	1.0	1.0	1.0
Entity selector max length	128	128	128	128
Attribute selector max context length	200	200	200	200
Attribute selector max kb length	100	100	100	100
Response generator max context length	200	200	200	200
Response generator max kb length	100	100	100	100
Max output length	64	64	64	64
Top-K retrieval entities	7	7	7	7
Attribute selection threshold	0.2	0.2	0.1	0.1
Distillation start gradient steps	938	938	938	938

Table 12: Hyperparameter settings of our system when the full knowledge base is used on MWOZ and CamRest.

Contributions	MWOZ		SMD		CamRest	
	T5-Base	T5-Large	T5-Base	T5-Large	T5-Base	T5-Large
Knowledge distillation	✓	✓	×	×	✓	✓
Entity Selection	✓	×	×	×	✓	✓
Attribute Selection	✓	✓	×	×	×	×

Table 13: Hyperparameter settings of whether to apply each contribution to our system when condensed knowledge bases are used on the MWOZ, SMD, and CamRest datasets.

Contributions	MWOZ		CamRest	
	T5-Base	T5-Large	T5-Base	T5-Large
Knowledge distillation	✓	✓	✓	✓
Entity Selection	✓	✓	✓	✓
Attribute Selection	✓	✓	✓	✓

Table 14: Hyperparameter settings of whether to apply each contribution to our system when the full knowledge base is used on MWOZ and CamRest.

User Utterance	I am looking for a restaurant. The restaurant should be in the north and should serve italian food.							
Retrieved Knowledge	**name**	**address**	**area**	**food**	~~**phone**~~	~~**postcode**~~	~~**pricerange**~~	**type**
	da vinci pizzeria	20 milton road chesterton	north	italian	~~1223351707~~	~~cb41jy~~	~~cheap~~	restaurant
	hakka	milton road chesterton	north	chinese	~~1223568988~~	~~cb41jy~~	~~expensive~~	restaurant
	...	...	...	...	...	...	...	...
Generated Response	Da vinci pizzeria is located at 20 milton road chesterton.							
Gold Response	Da vinci pizzeria at 20 milton road chesterton.							
User Utterance	Is that restaurant cheap?							
Retrieved Knowledge	**name**	**address**	**area**	**food**	~~**phone**~~	~~**postcode**~~	**pricerange**	**type**
	da vinci pizzeria	20 milton road chesterton	north	italian	~~1223351707~~	~~cb41jy~~	cheap	restaurant
	royal spice	victoria avenue chesterton	north	indian	~~1733553355~~	~~cb41ch~~	cheap	restaurant
	...	...	...	...	...	...	...	...
Generated Response	Yes it is.							
Gold Response	Yes the restaurant is cheap.							

Figure 4: An example of dialogue to illustrate our system. Blue font refers to knowledge base-related information.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☑ A2. Did you discuss any potential risks of your work?
Section Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4.1 and Section 4.2

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1 and Section 4.2

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.1 and Appendix A

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4.1 and Appendix A

C ☑ Did you run computational experiments?

Section 4 and Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Sections 4.2 and Appendix E

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Our experiments are extensive and computationally expensive. All experiments are based on the same random seed (111).

☒ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
We use the same evaluation metrics (BLEU, Entity-F1) as previous works and provide proper citations in Section 4.1 Datasets

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.