HyKnow: End-to-End Task-Oriented Dialog Modeling with Hybrid Knowledge Management

Silin Gao[1*], **Ryuichi Takanobu**[1*], **Wei Peng**[2], **Qun Liu**[2], **Minlie Huang**[1†]
[1] DCST, BNRIST, Tsinghua University, Beijing, China
[2] Huawei Technologies, Shenzhen, China
[1] gsl16@tsinghua.org.cn, gxly19@mails.tsinghua.edu.cn, aihuang@tsinghua.edu.cn
[2] peng.wei1@huawei.com, qun.liu@huawei.com

Abstract

Task-oriented dialog (TOD) systems typically manage structured knowledge (e.g. ontologies and databases) to guide the goal-oriented conversations. However, they fall short of handling dialog turns grounded on unstructured knowledge (e.g. reviews and documents). In this paper, we formulate a task of modeling TOD grounded on both structured and unstructured knowledge. To address this task, we propose a TOD system with hybrid knowledge management, HyKnow. It extends the belief state to manage both structured and unstructured knowledge, and is the first end-to-end model that jointly optimizes dialog modeling grounded on these two kinds of knowledge. We conduct experiments on the modified version of MultiWOZ 2.1 dataset, where dialogs are grounded on hybrid knowledge. Experimental results show that HyKnow has strong end-to-end performance compared to existing TOD systems. It also outperforms the pipeline knowledge management schemes, with higher unstructured knowledge retrieval accuracy.

1 Introduction

Recently, Task-Oriented Dialog (TOD) systems (Mehri et al., 2019; Zhang et al., 2020a,b; Le et al., 2020; Hosseini-Asl et al., 2020; Peng et al., 2020) have achieved promising performance on accomplishing user goals. Most systems typically query *structured knowledge* such as tables and databases based on the user goals, and use the query results to guide the generation of system responses, as shown in the first dialog turn in Fig. 1.

However, real-world task-oriented conversations often step into dialog turns which are grounded on *unstructured knowledge* (Feng et al., 2020), such as passages and documents. For example, as the second dialog turn in Fig. 1 shows, the user asks about customers' favorite food at *Pizza Hut*, which is grounded on the customer reviews of this restaurant. Current TOD systems fall short of handling such dialog turns since they cannot utilize relevant unstructured knowledge. This deficiency may interrupt the dialog process, causing difficulties in tracking user goals and generating system responses.

In this work, we consider incorporating more various forms of domain knowledge into the TOD systems. Therefore, we define a task of modeling TOD whose turns involve either structured or unstructured knowledge. In turns involving struc-

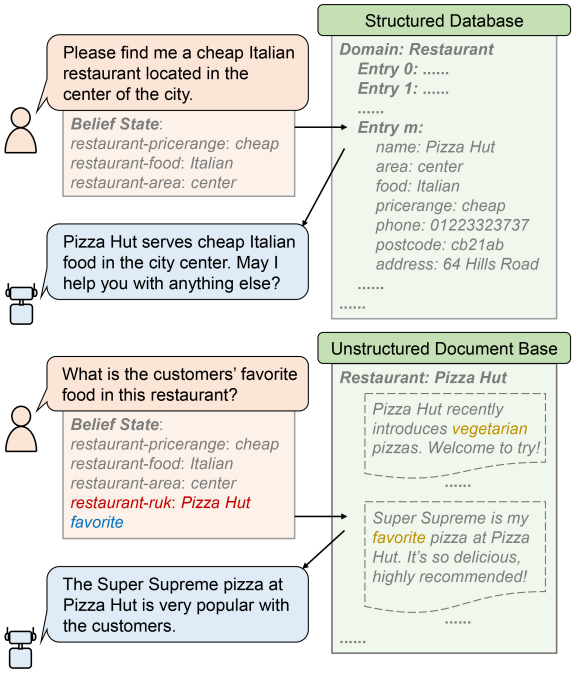

Figure 1: Illustration of task-oriented dialog modeling with hybrid knowledge management. Words in red and blue illustrate the new domain-slot-value triple and the topic of user utterance that we introduce into the belief state, respectively. Words in yellow illustrate the topics of documents that we extract through preprocessing.

[*] Equal contribution.
[†] Corresponding author.

tured knowledge, the system needs to track the user goals as triples and use them to perform database queries, whose results are used to generate the system response. While in turns involving unstructured knowledge, the system manages a document base to retrieve relevant references for generating the response.

To address our defined task, we propose a task-oriented dialog system with **Hy**brid **Know**ledge management (HyKnow). This model extends the belief state to handle TODs grounded on hybrid knowledge, and further uses the extended belief state to perform both database query and document retrieval, whose outputs are thereby used to generate the final response. We consider two implementations of our system, with different schemes of extended belief state decoding. Both implementations are in an end-to-end multi-stage sequence-to-sequence (Seq2Seq) (Lei et al., 2018; Liang et al., 2020; Zhang et al., 2020a,b) framework, where dialog modeling grounded on the two kinds of knowledge can be jointly optimized.

We evaluate our system on the modified version of MultiWOZ 2.1 (Kim et al., 2020) dataset, where dialogs are grounded on hybrid knowledge. Experimental results show that HyKnow outperforms existing TOD systems which do not leverage large pretrained language models, no matter whether they add extra unstructured knowledge management or not. It also has a higher accuracy in unstructured knowledge retrieval, compared to the pipeline knowledge management schemes.

Our contributions are summarized as below:

- We formulate a task of modeling TOD grounded on both structured and unstructured knowledge, to incorporate more domain knowledge into the TOD systems.
- We propose a TOD system HyKnow to address our proposed task. It extends the belief state to manage hybrid knowledge, and is the first end-to-end model to jointly optimize dialog modeling grounded on the two kinds of knowledge.
- Experimental results show that HyKnow has strong performance in dialog modeling grounded on hybrid knowledge.[1]

2 Related Work

TOD systems usually use belief tracking, i.e. dialog state tracking (DST) to trace the user goals, i.e. *belief states*, through multiple dialog turns (Williams et al., 2013; Henderson et al., 2014). The states are converted into a representation of constraints based on different schemes to query the databases (El Asri et al., 2017; Budzianowski et al., 2018; Rastogi et al., 2020; Zhu et al., 2020). The entry matching results are then used to generate the system response.

With the development of intelligent assistants, the system should have a good command of massive external knowledge to better accomplish complicated user goals and improve user satisfaction. To realize this, some researchers (Zhao et al., 2017; Yu et al., 2017; Akasaki and Kaji, 2017) equip the system with chatting capability to address both task and non-task content in TODs. Other studies apply knowledge graph (Liao et al., 2019; Yang et al., 2020) or tables via SQL (Yu et al., 2019) to enrich the knowledge of TOD systems. However, all these studies are still limited in dialog modeling grounded on structured knowledge.

There are a couple of studies to integrate unstructured knowledge into TOD modeling recently. Kim et al. (2020) introduce knowledge snippets to answer follow-up questions out of the coverage of databases. Feng et al. (2020) formulate document-grounded dialog for information seeking tasks. However, they only focus on dialog turns grounded on unstructured knowledge instead. In this paper, we aims to fill the gap of managing domain-specific knowledge with various sources and structures in traditional TOD systems.

3 Task Definition

In this section, we introduce our formulation of modeling TOD grounded on hybrid knowledge. In particular, we assume that each dialog turn in TOD is grounded on either structured or unstructured knowledge. We formulate the modeling of the two kinds of dialog turns separately.

In turns that are grounded on structured knowledge, the system needs to track user goals, i.e. the belief state, as domain-slot-value triples, and then query a database (DB) to guide response generation. Specifically, we denote the user utterance and the system response at turn t as U_t and R_t respectively. Given the dialog context $C_t = [U_{t-k}, R_{t-k}, ..., U_t]$ and previous belief state B_{t-1}, the system needs to generate current belief state B_t, which is formulated as $B_t = f_b^{(s)}(C_t, B_{t-1})$. Then the system performs DB query based on B_t to get the matching

[1]The code is available at https://github.com/truthless11/HyKnow

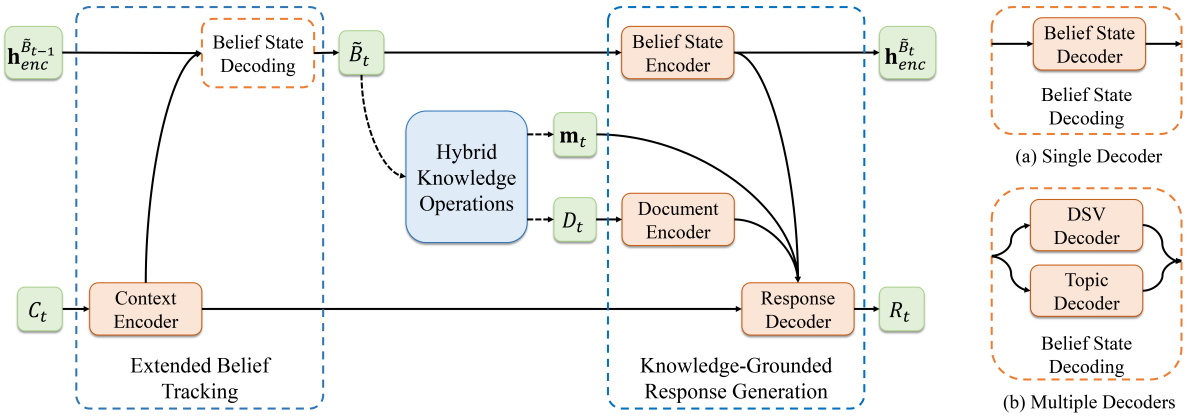

Figure 2: Overview of HyKnow. Solid arrows denote the input/output of the encoders or decoders. Dashed arrows denote the knowledge operations. C_t, $\mathbf{m}_t$, D_t and R_t represent turn t's dialog context, DB query result, relevant document and system response. $\widetilde{B}_t$ and $\mathbf{h}_{enc}^{\widetilde{B}_t}$ denote the extended belief state and its hidden states at turn t. The decoding of $\widetilde{B}_t$ (orange dashed box) is implemented in two different ways: (a) using a single decoder to generate the whole state, and (b) using two decoders to generate the domain-slot-value (DSV) triples and the topic separately.

result $\mathbf{m}_t$. In this paper, we follow Budzianowski et al. (2018) to represent $\mathbf{m}_t$ as a vector indicating the number of matched entities and whether the booking is available or not. Afterwards, the system generates the response R_t, formulated as $R_t = f_r^{(s)}(C_t, B_t, \mathbf{m}_t)$.

In turns that are grounded on unstructured knowledge, the system manages a document base to guide response generation, which contains lists of documents characterized by different domains and entities, as showed in Fig. 1. Specifically, given the dialog context C_t, the system first retrieves a relevant document D_t in the document base, formulated as $D_t = f_d^{(u)}(C_t)$. Then the system generates the response R_t based on C_t and retrieved D_t, which is formulated as $R_t = f_r^{(u)}(C_t, D_t)$. Noting that the original belief state is not updated in the unstructured knowledge-grounded turns, namely $B_t = B_{t-1}$. However, in this paper, we introduce extra belief state extension to facilitate the document retrieval.

4 Proposed Framework

Fig. 2 shows an overview of our proposed system HyKnow with end-to-end sequence-to-sequence (Seq2Seq) implementations. It addresses our proposed task in three steps. First, it uses the **extended belief tracking** to track user goals through dialog turns that involve hybrid knowledge. Secondly, it performs **hybrid knowledge operations** based on the extended belief state, to search structured and unstructured knowledge that is relevant to the user goals. Finally, it uses the extended belief state and relevant knowledge to perform the **knowledge-grounded response generation**.

4.1 Extended Belief Tracking

Belief State Extension. We define an extended belief state $\widetilde{B}_t$ which is applicable to track user goals in TODs that are grounded on both structured and unstructured knowledge. Specifically, in turns that are grounded on structured knowledge, $\widetilde{B}_t$ is same as the original B_t, which describes user goals as domain-slot-value triples. While in turns that are grounded on unstructured knowledge, $\widetilde{B}_t$ has an additional slot *ruk* to indicate that current dialog turn **r**equires **u**nstructured **k**nowledge. The prefix and value of the slot *ruk* represent the involved domain and entity, e.g. *restaurant-ruk*: *Pizza Hut* colored in red in Fig. 1. We denote the combination of original and newly introduced domain-slot-value triples as DSV_t. In addition, the *topic* of U_t is abstracted in $\widetilde{B}_t$ as a word sequence T_t in each unstructured knowledge-grounded turn, e.g. *favorite* colored in blue in Fig. 1.

Extended Belief State Decoding. Following Seq2Seq framework, we first use the *context encoder* to encode the dialog context C_t, whose last output is used as the initial hidden state of decoders. Based on the hidden states of context encoder $\mathbf{h}_{enc}^{C_t}$ and previous extended belief state $\mathbf{h}_{enc}^{\widetilde{B}_{t-1}}$, we then decode the current extended belief state $\widetilde{B}_t$ under two schemes, which are described as below.

Since DSV_t and T_t are grounded on quite different vocabularies, we consider implementing the decoding of $\widetilde{B}_t$ in two ways: (a) using the belief state decoder to generate the whole $\widetilde{B}_t$, and (b) using the DSV decoder and the topic decoder to generate DSV_t and T_t separately. Each implementation has its own advantages over the other. Specifically, in the single-decoder implementation, the decoding of DSV_t and T_t can be jointly optimized via shared parameters:

$$\widetilde{B}_t = \text{Seq2Seq}^{(b)}(C_t|\mathbf{h}_{enc}^{\widetilde{B}_{t-1}}). \tag{1}$$

While in the multi-decoder implementation, the decoding of DSV_t and T_t are fitted to their own smaller decoding spaces (vocabularies), and thus the generation of $\widetilde{B}_t$ can be decomposed into two simpler decoding processes:

$$\begin{aligned} DSV_t &= \text{Seq2Seq}^{(dsv)}(C_t|\mathbf{h}_{enc}^{\widetilde{B}_{t-1}}), \\ T_t &= \text{Seq2Seq}^{(t)}(C_t|\mathbf{h}_{enc}^{\widetilde{B}_{t-1}}), \\ \widetilde{B}_t &= [DSV_t, T_t]. \end{aligned} \tag{2}$$

4.2 Hybrid Knowledge Operations

Based on the extended belief state $\widetilde{B}_t$, we conduct both DB query and document retrieval to get the query result $\mathbf{m}_t$ and the relevant document D_t, which are used to guide the generation of response. In the operation of DB query, we simply match the original triples in $\widetilde{B}_t$ with the DB entries. While in the operation of document retrieval, we first preprocess the document base to extract the topic of each document as its retrieval index, e.g. *vegetarian* and *favorite* colored in yellow in Fig. 1. Then we use the extended part of $\widetilde{B}_t$ to match the domain, entity and extracted topic of each document, and select the best-matched one as D_t.[2]

4.3 Knowledge-Grounded Response Generation

We generate system response based on the dialog context C_t, the extended belief state $\widetilde{B}_t$, and the outputs of hybrid knowledge operations $\mathbf{m}_t$ and D_t. We first use the same context encoder in Sec. 4.1 to encode C_t. Moreover, we use the *belief state encoder* and the *document encoder* to encode $\widetilde{B}_t$ and D_t into hidden states $\mathbf{h}_{enc}^{\widetilde{B}_t}$ and $\mathbf{h}_{enc}^{D_t}$, respectively. Based on the hidden states of all the encoders and the vector $\mathbf{m}_t$, we use the *response decoder* to generate the system response R_t, formulated as:

$$\begin{aligned} \mathbf{h}_{enc}^{\widetilde{B}_t} &= \text{Encoder}^{(\text{b})}(\widetilde{B}_t), \\ \mathbf{h}_{enc}^{D_t} &= \text{Encoder}^{(\text{d})}(D_t), \\ R_t &= \text{Seq2Seq}^{(r)}(C_t|\mathbf{h}_{enc}^{\widetilde{B}_t}, \mathbf{h}_{enc}^{D_t}, \mathbf{m}_t), \end{aligned} \tag{3}$$

where $\text{Encoder}^{(\text{b})}$ and $\text{Encoder}^{(\text{d})}$ denote the belief state encoder and the document encoder.

Following previous TOD systems with Seq2Seq architectures (Lei et al., 2018; Liang et al., 2020; Zhang et al., 2020a,b), we use one-layer, bidirectional GRU as encoders and standard GRU as decoders. We also apply global attention (Bahdanau et al., 2015) and copy mechanism (Gu et al., 2016) in all the Seq2Seq processes, to improve the context-awareness of decoding $\widetilde{B}_t$ and R_t.

4.4 Model Training

HyKnow is optimized through supervised training. Specifically, each dialog turn in the training data is initially labeled with the original belief state and the relevant document. We extend the belief state label based on the domain, entity and extracted topic of the relevant document. Then the extended belief state label and the reference response are used to calculate the cross-entropy loss with the generated $\widetilde{B}_t$ and R_t, respectively. We sum the two losses together and perform gradient descent in each turn to optimize the model parameters.[3]

5 Experimental Settings

5.1 Dataset

We evaluate our proposed system on the modified MultiWOZ 2.1 (Kim et al., 2020) dataset, where crowd-sourcing workers are hired to insert additional turns into the original MultiWOZ dialogs. Each newly inserted turn is grounded on unstructured knowledge in one of the four domains: restaurant, hotel, taxi and train, with the label of its relevant document in the document base. While the other three MultiWOZ domains (attraction, hospital and police) are not involved in these new turns.[4]

5.2 Baselines

We compare HyKnow with 1) existing end-to-end (E2E) TOD models and dialog state tracking (DST)

[2] See Appendix A for more details of the document preprocessing and matching.

[3] See Appendix B for more implementation details.

[4] See Appendix C for details of data statistics.

Model	Pretrained LM	Inform	Success	BLEU	METEOR	ROUGE-L	Combined
UniConv	none	71.5	61.8	18.5	37.8	40.5	85.7
LABES-S2S	none	76.5	65.3	17.8	36.8	39.9	88.7
UniConv + BDA	-	72.0	62.6	16.9	35.7	38.9	84.2
LABES-S2S + BDA	-	77.1	66.2	15.7	33.8	37.8	87.4
HyKnow (Single)	none	**81.9**	**68.3**	**19.0**	**38.5**	40.9	**94.1**
- w/o Joint Optim	none	78.5	65.7	18.3	36.9	39.6	90.4 (-3.7)
HyKnow (Multiple)	none	79.1	67.6	18.7	38.1	**41.0**	92.1
- w/o Joint Optim	none	77.7	65.4	18.0	36.6	39.5	89.6 (-2.5)
SimpleTOD	GPT-2	81.7	67.9	14.5	34.2	37.0	89.3
SimpleTOD + BDA	-	83.3	68.6	14.8	33.6	36.5	90.8

Table 1: End-to-end evaluation results on modified MultiWOZ 2.1. "+" denotes the combination of Beyond Domain APIs (BDA) with E2E TOD models. Best results among light-weight systems (i.e. above internal dividing line) are labeled in bold. Evaluation metrics are described and labeled in bold in Sec. 6.1.

Model	Inform	Success	BLEU	Combined
UniConv	84.2	71.8	19.0	97.3
LABES-S2S	83.6	74.2	18.3	97.2
UniConv + BDA	85.8	73.9	19.3	99.4
LABES-S2S + BDA	85.0	75.3	18.9	99.1
HyKnow	**87.2**	**76.5**	**19.5**	**101.4**
SimpleTOD	87.5	76.4	16.3	98.3
SimpleTOD + BDA	89.0	77.2	17.0	100.1

Table 2: Context-to-response generation results.

models, to show the benefits of incorporating unstructured knowledge management into TOD modeling. We also compare HyKnow with 2) unstructured knowledge management models, to investigate our system's document retrieval performance. For the comparison with pipeline systems that have hybrid knowledge management, we also consider the combinations of 1) and 2) as our baselines.

E2E TOD Models and DST Models. We consider three baseline E2E TOD models with different types of structures: **UniConv** (Le et al., 2020) uses a structured fusion (Mehri et al., 2019) design, **LABES-S2S** (Zhang et al., 2020a) uses a multi-stage Seq2Seq (Lei et al., 2018) architecture, and **SimpleTOD** (Hosseini-Asl et al., 2020) is based on a single auto-regressive language model initialized from GPT-2 (Radford et al., 2019). All three E2E models only manage structured knowledge (database) in their TOD modeling. In addition to E2E TOD models, we also compare HyKnow with existing DST models in the belief tracking evaluation. Specifically, we use **TRADE** (Wu et al., 2019) and **TripPy** (Heck et al., 2020) as two DST baselines, which are representative BERT-free and BERT-based DST models, respectively.

Unstructured Knowledge Management Models. We first compare our system with Beyond Domain APIs (**BDA**) (Kim et al., 2020). This baseline model uses two classification modules based on BERT (Devlin et al., 2019) to detect unstructured knowledge-grounded dialog turns and retrieve relevant documents, respectively. Moreover, we use standard information retrieval (IR) systems **TF-IDF** (Manning et al., 2008) and **BM25** (Robertson et al., 2009) as the other two baseline models.

Combinations. We combine the unstructured knowledge management model BDA with every DST or E2E TOD model. Specifically, BDA detects dialog turns that are grounded on unstructured knowledge, and uses a fine-tuned GPT-2 to generate responses in these turns, based on the dialog context and retrieved documents. While the DST or E2E TOD model handles the rest dialog turns which are grounded on structured knowledge.

Noting that TripPy and SimpleTOD use large-scale pretrained language models (LM) to improve their belief tracking and response generation performances, which require large model sizes and computing resources. For fair comparisons, we distinguish them from other light-weight models in our experiments.

6 Results and Analysis

We test our system's performances under both the single-decoder and multi-decoder belief state decoding implementations, denoted as HyKnow (Single) and HyKnow (Multiple), respectively. Both implementations of HyKnow come to the same conclusions when compared with the baseline models, which are described in detail below.

6.1 End-to-End Evaluation

Table 1 shows our experimental results of the end-to-end (E2E) evaluation, where we evaluate the task completion rate and language quality of system responses. In terms of the task completion rate,

Model	Pretrained LM	Joint Goal
TRADE	none	42.9
UniConv	none	45.5
LABES-S2S	none	46.0
TRADE + BDA	-	43.8
UniConv + BDA	-	46.5
LABES-S2S + BDA	-	46.8
HyKnow (Single)	none	**48.0**
- w/o Joint Optim	none	46.2 (-1.8)
HyKnow (Multiple)	none	47.6
- w/o Joint Optim	none	45.6 (-2.0)
TripPy	BERT	50.4
SimpleTOD	GPT-2	48.4
TripPy + BDA	-	51.2
SimpleTOD + BDA	-	49.8

Table 3: Original turns' belief tracking results on modified MultiWOZ 2.1. "+" denotes the combination of BDA with DST/E2E models. The best result among light-weight systems (i.e. above internal dividing line) is labeled in bold. The evaluation metric is described and labeled in bold in Sec. 6.3.

Model	Type	MRR@5	R@1
TF-IDF	standard IR	68.7	54.1
BM25	standard IR	69.2	52.5
BDA	classification	80.6	69.8
HyKnow (Single)	topic match	**81.7**	**80.2**
- w/o Joint Optim	topic match	80.1 (-1.6)	77.8 (-2.4)
HyKnow (Multiple)	topic match	81.1	79.5
- w/o Joint Optim	topic match	79.7 (-1.4)	77.4 (-2.1)

Table 4: Newly inserted turns' document retrieval results on modified MultiWOZ 2.1. Best results are labeled in bold. Evaluation metrics are described and labeled in bold in Sec. 6.3.

we measure whether the system provides correct entities (**Inform** rate) and answers all the requested information (**Success** rate) in a dialog, following Budzianowski et al. (2018). For the evaluation of language quality, we adopt commonly used metrics **BLEU** (Papineni et al., 2002), **METEOR** (Banerjee and Lavie, 2005) and **ROUGE-L** (Lin, 2004). Moreover, we use **Combined** score computed by $(Inform + Success) \times 0.5 + BLEU$ for overall evaluation, as suggested by Eric et al. (2020).

We find that HyKnow has better task completion rate than the light-weight E2E TOD models, which is comparable with SimpleTOD who uses large-scale pretrained GPT-2. It also generates responses with better language quality compared to all the E2E models. This is because our extended belief state can distinguish whether a dialog turn is grounded on structured or unstructured knowledge, which avoids the confusion between handling the two kinds of turns. In addition, we manage the document base to provide relevant references for generating the response, which guide our system to give more appropriate responses in turns that are grounded on unstructured knowledge.

We also observe that HyKnow outperforms the combinations of BDA and light-weight E2E TOD model. This indicates that our end-to-end model framework has advantages over the pipeline structures of combination models. In particular, dialog modeling grounded on the structured and unstructured knowledge are integrated in a uniform Seq2Seq architecture in our system, where they are jointly optimized to an overall better performance. Although HyKnow does not significantly outperform the combination of BDA and SimpleTOD, our system has lower deployment cost since it is trained end-to-end.

6.2 Context-to-Response Generation

We also conduct evaluations on the context-to-response (C2R) generation, where systems directly use the oracle belief state and knowledge to generate the response. The experimental results are shown in Table 2, where we observe the same conclusions as in the E2E evaluation (Table 1). This again shows our system's superiority in TOD modeling grounded on hybrid knowledge.

Additionally, we observe that HyKnow's performance gap between E2E and C2R evaluations is smaller than the baseline models, reflected in the smaller variations of the combined score. This shows that the belief state and knowledge provided by our system are probably closer to the oracle and may give stronger guidance to generate a response.

6.3 Knowledge Management

To further investigate our system's end-to-end performance, we conduct evaluations on the intermediate knowledge management. In particular, we evaluate the structured and unstructured knowledge management separately in the original and newly inserted dialog turns. In the original turns grounded on structured knowledge, we evaluate the belief tracking performance which directly determines the database query accuracy. Specifically, we use the **Joint Goal** accuracy (Henderson et al., 2014) to measure whether belief states are predicted correctly in a dialog turn. While in newly inserted turns grounded on unstructured knowledge, we adopt standard information retrieval metrics **R@1** and **MRR@5** to evaluate the document retrieval performance. Table 3 and 4 shows our evaluation

Test Set	Model	Joint Goal	Inform	Success	BLEU	METEOR	ROUGE-L	Combined
Original	LABES-S2S + BDA	49.0	82.1	69.8	17.8	37.1	40.2	93.8
	SimpleTOD + BDA	51.8	85.6	70.9	16.3	34.5	38.6	94.6
	HyKnow (Single)	49.2	82.3	69.4	18.0	37.3	40.2	93.9
Modified	LABES-S2S + BDA	46.8 (-2.2)	77.1	66.2	17.7	36.8	39.6	89.4 (-4.4)
	SimpleTOD + BDA	49.8 (-2.0)	83.3	68.6	15.8	33.6	37.8	91.8 (-2.8)
	HyKnow (Single)	48.0 (-1.2)	81.9	68.3	17.8	37.2	39.5	92.9 (-1.0)

Table 5: End-to-end evaluation results on the original and modified MultiWOZ 2.1 test set. The evaluation is conducted only in the original dialog turns.

Model	Original			Newly Inserted		
	Cohe.	Info.	Corr.	Cohe.	Info.	Corr.
SimpleTOD	2.58	2.56	2.44	2.50	2.04	2.14
SimpleTOD + BDA	2.56	2.60	**2.46**	2.52	2.30	2.22
HyKnow (Single)	**2.60**	**2.62**	2.42	**2.56**	**2.36**	**2.50**

Table 6: Human evaluation results on modified MultiWOZ 2.1, results in original and newly inserted turns are shown separately.

results of belief tracking and document retrieval, respectively.

In terms of belief tracking, HyKnow outperforms the light-weight DST/E2E models. This is because our extended belief tracking can detect the newly inserted turns apart from the original turns (via the slot *ruk*), which improves our system's awareness on deciding when to update the original triples in the belief state. HyKnow also has better belief tracking performance than the combinations of BDA and light-weight DST/E2E model. This is because error propagation on updating belief states is eliminated in our system compared to the pipeline framework: The pipeline system either updates the belief state or retrieves the document in one turn, but HyKnow can perform both operations in the nature of its E2E design. Although the belief tracking performance of HyKnow is not as good as that of TripPy and SimpleTOD, our system does not use large-scale pretrained BERT or GPT-2 and is thus computational cheaper.[5]

In the document retrieval evaluation, we find that HyKnow outperforms the unstructured knowledge management models, especially on the R@1 metric. This shows that our system's document retrieval scheme with topic matching has a higher accuracy, compared to the classifier-based BDA and the standard information retrieval (IR) systems. Specifically, HyKnow retrieves documents based on the highly simplified semantic information, i.e. the topic, which reduces the complexity of the retrieval process. This makes the retrieval scheme of HyKnow more concise and effective than the baseline models, who directly calculate the relevance of dialog context to every document content.

[5]See Appendix D for details on model size comparison.

6.4 Single vs. Multiple Decoders

We then compare our two implementations of extended belief state decoding. We calculate the vocabularies of DSV triples, the topic and their combination (which are 709, 166 and 862), and observe that the last one approximately equals to the sum of the former two. This confirms our assumption in Sec. 4.1 that DSV triples and topic have quite different vocabularies, which motivates the multi-decoder implementation in belief state decoding.

However, we find that HyKnow (Single) outperforms HyKnow (Multiple) in both E2E and knowledge management evaluations, as shown in Table 1, 3 and 4. This shows that the decoding of DSV triples and topic can benefit from the joint optimization via shared parameters, although they are grounded on quite different vocabularies. The superiority of joint optimization further implies that the structured and unstructured knowledge management in TOD modeling have a positive correlation, since they commonly involve task-specific domain knowledge and entities. Therefore, the two kinds of knowledge management can learn from each other through joint training, and achieve overall better performance compared to separating them apart.

6.5 Ablation Study

We ablate the joint optimization of structured and unstructured knowledge-grounded TOD modeling to investigate its role in our framework, denoted as w/o Joint Optim in Table 1, 3 and 4. Specifically, we train two HyKnow models separately on the original and newly inserted dialog turns, and use them to handle TOD grounded on structured and unstructured knowledge, respectively. To determine which model should be used, the oracle label of slot *ruk* is used to judge which knowledge type

U_1: Hello, I would like to find a **hotel** that has WiFi in the north part of the town. B_1: hotel-area: north, hotel-internet: yes R_1: The **Arbury Lodge Guesthouse** is one of 12 options for you. Shall I make a reservation for you? U_2: Do they provide **Italian breakfast**?	
Human	B_2: hotel-area: north, hotel-internet: yes D_2: Full English **breakfast** is served every morning at Arbury Lodge Guesthouse. R_2: **No**, but they have a full English breakfast. Did you still want to book a room here?
SimpleTOD	B_2: hotel-area: north, hotel-internet: yes, **restaurant-food: Italian** R_2: Da Vinci Pizzeria serves Italian food and is located in the north.
SimpleTOD + BDA	B_2: hotel-area: north, hotel-internet: yes D_2: German, Italian, Spanish and English are **spoken** at Carolina Bed and Breakfast. R_2: **Yes**, the hotel has a Italian breakfast. Would you like to make a reservation?
HyKnow (Single)	$\widetilde{B}_2$: hotel-area: north, hotel-internet: yes, **hotel-ruk: Arbury Lodge Guesthouse**, **breakfast** D_2: Full English **breakfast** is served every morning at Arbury Lodge Guesthouse. R_2: **No**, they provides full English breakfast. Would you like me to book a room for you?

Table 7: Comparison of the belief tracking, document retrieval and response generation results when dialog steps into a newly inserted turn of modified MultiWOZ 2.1. "Human" denotes ground truth results.

the current dialog turn is grounded on.

We observe that removing joint optimization brings HyKnow evident performance declines in the end-to-end evaluation (Table 1). This suggests that joint optimization plays a significant role in improving HyKnow's end-to-end performance, where TOD modeling grounded on the two kinds of knowledge can benefit each other by learning shared parameters. The ablation of joint optimization also causes performance declines in HyKnow's knowledge management (Table 3 and 4). This again indicates that the two kinds of knowledge management are positively correlative and can get benefit from joint training.

6.6 Between Structured and Unstructured Knowledge

In this section, we investigate how the newly inserted dialog turns (grounded on unstructured knowledge) affect systems' E2E performances in the original dialog turns (grounded on structured knowledge). Specifically, we evaluate systems' E2E performances on both the original and modified MultiWOZ 2.1 test sets. This evaluation is conducted only in the original dialog turns, which is different from the E2E evaluation conducted in all turns (Table 1). Table 5 shows the results of this experiment, where we compare HyKnow (Single) with strong combination models.

We find that all the models' performances are degraded when transferred from the original to the modified test set. This indicates that the inserted turns grounded on new knowledge may interrupt the original dialogs, which complicates the dialog process and causes difficulties in the original turns' dialog modeling.

However, we observe that HyKnow (Single) suffers from less reduction compared to the baseline combination models. This shows that our system has a stronger resistance to the interruptions of newly inserted turns, which benefits from our end-to-end modeling. Specifically, HyKnow jointly optimizes dialog modeling of the original and newly inserted turns in a uniform end-to-end framework. This unified modeling approach improves our system's flexibility in switching between the two kinds of turns, and thus makes it more competent in handling the complicated dialog process.

6.7 Human Evaluation

There is still a gap between the evaluation results of automatic metrics and the real E2E performances of TOD systems. Therefore, we conduct human evaluation to more adequately test our system's E2E performance. In particular, we compare HyKnow (Single) with a strong E2E baseline SimpleTOD and its combination with BDA.

We conduct human evaluation separately on the two types (original and newly inserted) of dialog turns. Specifically, we sample fifty dialog turns of each type and ask the judges to evaluate each turn's system response on three aspects. **Coherence** (Cohe.) measures how well the response is coherent with the dialog context. **Informativeness** (Info.) measures how well the response can provide sufficient information that meets the user requests. **Correctness** (Corr.) measures how well the information in response is consistent with the ground truth knowledge, i.e. relevant DB entries or documents. All the three aspects are scored on a Liker scale of 1-3, which denotes *bad*, *so-so* and *good*.

Table 6 shows our human evaluation results. In

the original dialog turns, HyKnow (Single) scores close to SimpleTOD and its combination with BDA on all the three aspects. This indicates that our proposed light-weight system is comparable with the large GPT-2 based models in managing structured knowledge to generate the response. In addition, our model outperforms the two baseline models in the newly inserted dialog turns. Specifically, HyKnow (Single) generates responses with significantly better informativeness and correctness than SimpleTOD. This again shows that the management of unstructured knowledge is beneficial for generating appropriate responses. Compared to the combination of SimpleTOD and BDA, the responses generated by HyKnow (Single) also achieve much better correctness, which benefits from our model's higher document retrieval accuracy (as shown in Table 4).

6.8 Case Study

An example dialog segment (U_1, B_1, R_1, U_2) and corresponding output results of each model (B_2, D_2, R_2) are presented in Table 7. Without access to unstructured document base, SimpleTOD misunderstands the user query, and instead recognizes the term "Italian" in user utterance as a constraint to update the belief state. As a result, the system makes an inappropriate recommendation. By combining with BDA, SimpleTOD predicts correct belief state, but fails in finding the relevant document, thus providing a wrong answer. This is because the wrong document's content has many common words with the dialog context, e.g. "Italian" and "Breakfast", which mislead the retrieval process of BDA. In contrast, HyKnow gives a proper response with accurate information as it identifies the entity ("Arbury Lodge Guesthouse") and captures the topic ("breakfast") during conversation to avoid the misleading of common words in document retrieval.

7 Conclusion

In this paper, we define a task of modeling TOD with access to both structured and unstructured knowledge. To address this task, we propose a TOD system HyKnow which uses an E2E framework to jointly optimize TOD modeling grounded on the two kinds of knowledge. In the experiments, HyKnow shows strong performance in modeling TOD with hybrid knowledge management, compared to existing TOD systems and their pipeline extensions. For future work, we plan to incorporate large-scale pretrained language models into our proposed system to further enhance its performance. Furthermore, we consider evaluating our system on different scenarios where dialogs are grounded on hybrid knowledge.

References

Satoshi Akasaki and Nobuhiro Kaji. 2017. Chat detection in an intelligent assistant: Combining task-oriented and non-task-oriented spoken dialogue systems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1308–1319.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

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.

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.

Layla El Asri, Hannes Schulz, Shikhar Kr Sarma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219.

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 12th Language Resources and Evaluation Conference*, pages 422–428.

Song Feng, Hui Wan, Chulaka Gunasekara, Siva Patel, Sachindra Joshi, and Luis Lastras. 2020. Doc2dial: A goal-oriented document-grounded dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8118–8128.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

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, 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. In *Advances in Neural Information Processing Systems*, volume 33.

Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu, and Dilek Hakkani-Tur. 2020. Beyond domain apis: Task-oriented conversational modeling with unstructured knowledge access. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 278–289.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Hung Le, Doyen Sahoo, Chenghao Liu, Nancy Chen, and Steven CH Hoi. 2020. Uniconv: A unified conversational neural architecture for multi-domain task-oriented dialogues. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1860–1877.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.

Weixin Liang, Youzhi Tian, Chengcai Chen, and Zhou Yu. 2020. Moss: End-to-end dialog system framework with modular supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8327–8335.

Lizi Liao, Ryuichi Takanobu, Yunshan Ma, Xun Yang, Minlie Huang, and Tat-Seng Chua. 2019. Deep conversational recommender in travel. *arXiv preprint arXiv:1907.00710*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Christopher D Manning, Hinrich Schütze, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Cambridge university press.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 165–177.

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.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Few-shot task-oriented dialog with a single pretrained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Jason D Williams, Antoine Raux, Deepak Ramachandran, and Alan W Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413.

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.

Shiquan Yang, Rui Zhang, and Sarah Erfani. 2020. Graphdialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1878–1888.

Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, et al. 2019. Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases. 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 1962–1979.

Zhou Yu, Alan W Black, and Alexander I Rudnicky. 2017. Learning conversational systems that interleave task and non-task content. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4214–4220.

Yichi Zhang, Zhijian Ou, Min Hu, and Junlan Feng. 2020a. A probabilistic end-to-end task-oriented dialog model with latent belief states towards semi-supervised learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9207–9219.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020b. 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.

Tiancheng Zhao, Allen Lu, Kyusong Lee, and Maxine Eskenazi. 2017. Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 27–36.

Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020. Crosswoz: A large-scale chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics*, 8:281–295.

A Document Preprocessing and Matching

We preprocess the document base of the modified MultiWOZ 2.1 (Kim et al., 2020) dataset to extract the topic of each document, which are used as its retrieval index in the unstructured knowledge management. Based on the TF-IDF (Manning et al., 2008) algorithm, we perform the topic word extraction domain-by-domain in a two-step procedure. First, we choose the top-three keywords with the highest TF-IDF scores in each document as its topic candidates. Then we filter the candidates to further select our desired topic words.

Noticing that different entities in the same domain usually have documents covering similar topics, we assume that a desired topic word should typically appear in multiple entities' documents, and therefore have a high frequency of occurrence among the topic candidates. So we calculate a cumulative average TF-IDF (CA-TF-IDF) score for each topic word in the candidates, which synthetically measures the word's document-level TF-IDF and entity-level occurrence frequency. Specifically, CA-TF-IDF sums the TF-IDF score of a topic word's each occurrence in the candidates, and divides it by the entity number in the domain. We filter out the topic candidates with low CA-TF-IDF scores and retain the rest to form the final retrieval indexes. The filtering thresholds are 2.3, 2.7, 6.9 and 7.3 for the domain of restaurant, hotel, taxi and train, respectively. While other domains are not involved in the document base. After the preprocessing, each document has one to three topic words extracted.

In the document retrieval process, we use the prefix and value of slot *ruk* in our proposed extended belief state to locate the document list of involved domain and entity. Then we use the topic of user utterance in our extended belief state to match the extracted topic of each document in the involved list, and select the best-matched one as the relevant reference. The topic matching is conducted by using the fuzzy string matching toolkit[6]. Noting that the relevant document is set to *none* if the slot *ruk* or the topic of user utterance is not available.

B Implementation Details

The dialog context C_t in our system is set as the concatenation of previous system response R_{t-1}

[6] https://github.com/seatgeek/fuzzywuzzy

Model	Pretrained LM	Model Size
TRADE	none	10M
UniConv	none	16M
LABES-S2S	none	3.8M
HyKnow (Single)	none	4.1M
HyKnow (Multiple)	none	5.3M
TripPy	BERT	110M
SimpleTOD	GPT-2	81M

Table 8: Comparison of model size.

and current user utterance U_t. We use GloVe (Pennington et al., 2014) to initialize the embedding matrix, and set batch size, embedding size, hidden size and vocabulary size as 40, 50, 200 and 3000, respectively. We also set dropout rate as 0.35 and use greedy decoding to generate the belief state and response. Moreover, we use Adam optimizer (Kingma and Ba, 2014) with a learning rate of $7e^{-4}$, which is selected via grid search from $\{4e^{-4}, 5e^{-4}, 6e^{-4}, 7e^{-4}, 8e^{-4}, 9e^{-4}, 1e^{-3}\}$. We halve the learning rate when no improvement of overall performance (combined score) is observed on the development set in two consecutive epochs, and we stop the training when no improvement is observed in four consecutive epochs. The average training time is about 80 minutes per epoch, and the total number of training epoch is around 15. Model training is performed on NVIDIA TITAN-Xp GPU.

C Statistics of Modified MultiWOZ 2.1

There are totally 8449/1001/1004 dialogs[7] in the training, development and testing set of modified MultiWOZ 2.1, where 6501/836/847 dialogs have new turns inserted, respectively. After the modification, each dialog has 8.93 turns on average, which is longer than the original 6.85. The ontology of modified MultiWOZ 2.1 is the same as the original, with 32 slot types (excluding *ruk*) and 2426 corresponding slot values.

D Model Size Comparison

Table 8 shows the model size of our proposed HyKnow and some baseline models. We find that HyKnow has a comparable model size with the lightweight baseline models, which do not leverage pretrained language models (LM). But its model size is much smaller than that of TripPy and SimpleTOD, which use pretrained BERT and GPT-2, respectively. Therefore, HyKnow requires much less computational resources, compared to TripPy and SimpleTOD that use large-scale pretrained LM.

[7] These are slightly more compared to the original MultiWOZ 2.1, because some of the original dialogs are modified twice with different turns inserted.