Towards Large-Scale Interpretable Knowledge Graph Reasoning for Dialogue Systems

Yi-Lin Tuan[1], **Sajjad Beygi**[2], **Maryam Fazel-Zarandi**[2]
Qiaozi Gao[2], **Alessandra Cervone**[2], **William Yang Wang**[1]
[1] University of California, Santa Barbara [2] Amazon Alexa AI
{ytuan, william}@cs.ucsb.edu
{beygi, fazelzar, qzgao, cervon}@amazon.com

Abstract

Users interacting with voice assistants today need to phrase their requests in a very specific manner to elicit an appropriate response. This limits the user experience, and is partly due to the lack of reasoning capabilities of dialogue platforms and the hand-crafted rules that require extensive labor. One possible way to improve user experience and relieve the manual efforts of designers is to build an end-to-end dialogue system that can do reasoning itself while perceiving user's utterances. In this work, we propose a novel method to incorporate the knowledge reasoning capability into dialogue systems in a more scalable and generalizable manner. Our proposed method allows a single transformer model to directly walk on a large-scale knowledge graph to generate responses. To the best of our knowledge, this is the first work to have transformer models generate responses by reasoning over differentiable knowledge graphs. We investigate the reasoning abilities of the proposed method on both task-oriented and domain-specific chit-chat dialogues. Empirical results show that this method can effectively and efficiently incorporate a knowledge graph into a dialogue system with fully-interpretable reasoning paths.

1 Introduction

Nowadays, dialogue systems are ubiquitous in customer service and voice-based assistants. One of the main uses of this technology is supporting humans in accomplishing tasks that might require accessing and navigating large knowledge bases (e.g., movies search). A dialogue system architecture is typically composed of a natural language understanding (NLU) module, a dialogue management (DM) module, and a natural language generation (NLG) module (Jurafsky and Martin, 2009; Williams et al., 2016). First, the NLU component extracts a meaning representation from the user utterance based on which the DM generates the next system action by reasoning over the meaning representation and communicating with external applications if necessary. For example, the DM may retrieve information from external knowledge graphs (KG) to answer the user's query based on the dialogue history. This process requires the DM to convert the output of NLU to a query to be issued to the backend. Given the difficulty of this step, which is often domain-dependent, the DM component might require the design of hand-crafted rules. However, such rules are usually not scalable to different applications. They could require considerable effort to cover all possible cases/dialogue flows, leading to expensive costs to design new applications. Moreover, in several cases, users interacting with such assistants are forced to formulate specific queries in order to accomplish their objective, which might break user engagement.

To alleviate the problem of having to design expensive hand-crafted rules and breaking user experience, recent works have explored the possibility of building end-to-end dialogue systems (Wen et al., 2017) and all-in-one response generation models (Serban et al., 2016). Among them, since graph is one of the main structure to store knowledge, recent research (Ghazvininejad et al., 2018; Zhou et al., 2018; Moon et al., 2019; Tuan et al., 2019; Yang et al., 2020) has proposed methods to generate natural language responses according to both the dialogue history and external knowledge graph. Despite these innovative and inspiring methods, there are some shortcomings. For instance, these methods are either not fully-interpretable or limited to small-scale knowledge graphs.

In this paper, we propose a novel dialogue differentiable knowledge graph model (DiffKG). The DiffKG is a single transformer model that directly (1) generates a sequence of relations to perform multi-hop reasoning on a reified KG representation proposed by (Cohen et al., 2019), and then (2) generates responses using the retrieved entities. To

the best of our knowledge, this is the first dialogue model that can directly walk on a large-scale KG with flexibility and interpretability. DiffKG allows having flexible entity values in the KG and handling novel entity values with an arbitrarily defined number of tokens. The reasoning path of DiffKG consists of the predicted relations, thus allowing for transparency.

We run extensive experiments to test DiffKG performance on KG-grounded dialogues. We select Stanford Multi-domain Dialogues (SMD) (Eric et al., 2017) and propose a new dataset, SMD-Reasoning, to simulate scenarios requiring multiple reasoning types and select the OpenDialKG (Moon et al., 2019) to simulate scenarios requiring large-scale KG reasoning without preprocessing. We then compare DiffKG with state-of-the-art models on SMD and OpenDialKG and an additional baseline that flattens KGs into a textual form from which transformers can learn. Empirically, our experiments show that DiffKG can effectively be trained on large-scale KGs and demonstrate its robustness with modified triplets in a KG. From the perspective of computation, DiffKG leads to relatively low extra time and memory usage compared to transformer models not using any KG information.

In summary, our contributions are: 1) We propose DiffKG, a novel method that can effectively and flexibly incorporate large-scale KG; 2) We demonstrate that DiffKG is a model-agnostic method and can be applied to different model architectures; 3) We show that DiffKG is an interpretable method with low add-on latency at inference time. Our code and processed datasets are released in https://github.com/Pascalson/DiffKG-Dialog.

2 Related Work

Recent years have seen a surge of new methods proposing end-to-end models that try to both understand natural language input text and search information. Two of the widely explored tasks are question-answering (QA) and dialogue generation.

QA. Multiple QA methods (Weston et al., 2015; Yin et al., 2016; Hao et al., 2017; Rajpurkar et al., 2018; Verga et al., 2020; Eisenschlos et al., 2021) have been proposed to tackle tasks that go beyond what is explicitly stated in the linguistic context (Storks et al., 2019). For example, the benchmarks (Mihaylov et al., 2018; Reddy et al., 2019; Khot et al., 2020; Lin et al., 2021) are particularly useful for the model to extract information from external knowledge bases to answer questions. Nonetheless, these studies mostly take the retrieved information from KG as the answer to a single question, while in dialogue we have to formulate an informative response to multi-turn dialogue history.

Dialogue Generation. Recent works have investigated the grounded dialogue generation. These methods can be divided into three main categories. First, Dinan et al. (2018); Zhao et al. (2019); Tuan et al. (2020); Kim et al. (2020) extract useful knowledge from unstructured data to generate responses, such as information contained in passages and speaker's profiles. Second, Sordoni et al. (2015); Long et al. (2017); Zhu et al. (2017); Ghazvininejad et al. (2018); Zhou et al. (2018); Veličković et al. (2018); Joshi et al. (2020); Hosseini-Asl et al. (2020); Wang et al. (2021) utilize information from knowledge bases (either graphs or tables) to enhance the dialogue system. They usually train the entities and relations embeddings of the knowledge bases and incorporate these embeddings into the input representation to predict the response. Third, Moon et al. (2019); Tuan et al. (2019); Jung et al. (2020) formulate the reasoning process more explicitly, as a path traversal over knowledge graphs. These methods further improve the transparency and explainability of the conversational agent and share the most similar idea with us. However, they either only predict the reasoning path without generating responses or need subgraph sampling to reduce the scale of KG. In this work, our approach uses a transformer model to jointly predict explicit reasoning paths over a large-scale knowledge graph and generate dialogue responses based on the reasoning results.

3 Background

3.1 Knowledge Graph for Dialogue System

We assume that the knowledge of the system can be represented by a knowledge graph (KG) $\mathcal{G} = \{\mathcal{E}, \mathcal{R}\}$, where $\mathcal{E}$ denotes the entities and $\mathcal{R}$ denotes the relations. The knowledge graph $\mathcal{G}$ contains multiple triples describing the connections among entities and relations. We denote the k-th triple of this graph as (e^h_k, r_k, e^t_k) , where e^h_k, r_k, e^t_k are respectively the head entity, relation, and tail entity. The total numbers of triples, entities, and relations

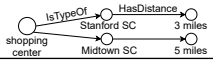
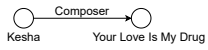
Reasoning Type			Example	Related Info. in KG
Semantic Form	KG reasoning		U: I need unleaded gas. R: inform Valero, 4 miles	
	Logical Reasoning	True/False	U: Is it going to snow this week at Corona? R: Yes	
		Selection	U: give me the direction to the nearest shopping mall. R: inform Stanford Shopping Center, 3 miles	
		Extraction	U: What gas stations are here? R: include poi_type gas station	No gas station in the available KG
NL Form	KG reasoning		U: Have you listen to any of the singer Kesha's song? R: I do enjoy in her music, especially "Your Love Is My Drug"	

Table 1: Example of different reasoning types and output formats (semantic and natural language forms) in a dialogue system with the related information in the accessible KGs.

are denoted as $N_{\mathcal{T}}$, $N_{\mathcal{E}}$, $N_{\mathcal{R}}$, respectively.[1]

3.2 Response Generation in Dialogue System

If we define the dialogue history as a sequence of tokens that occurred during the user and system interactions, then a flattened dialogue history can be written as:

$$\mathbf{x} = (x_1, x_2, ..., x_m, ..., x_M) \quad (1)$$

where x_m is the m-th token in the dialogue history with M tokens. In an end-to-end dialogue system, we assume a dialogue system parameterized by θ exists that can predict a probability distribution of responses $P_\theta(\cdot|\mathbf{x}, \mathcal{G})$. The generated responses are sampled from this probability distribution.

4 Problem Statement

We focus on understanding the ability of language models in performing reasoning during a conversation. We consider two tasks that are usually required in dialogue scenarios and call them semantic form and natural language (NL) form in Table 1. First, given a dialogue history and a user's query, the task of semantic form is to predict the next system action, corresponding to the output of the DM module, based on the available knowledge. In this case, we assume the expected output is the essential knowledge for an NLG module. We argue that this task could help better evaluate if the response is correct or not and which type of reasoning can be more successfully handled. Second, given a dialogue history and a user's query, the task of the NL form could be to directly output the response given by the system. This setting with annotated reasoning path can shed light on understanding if the model can learn to support chit-chat and reasoning at the same time.

Moreover, we aim to understand models' reasoning capability both in the form of logical reasoning and over the provided knowledge. As illustrated in Table 1, by KG reasoning, we refer to the ability of the model to retrieve information from an arbitrary scaled KG in multiple hops. Meanwhile, we refer to logical reasoning as the ability of the model to conduct operations such as evaluating whether a statement is true or false, selecting min/max from a list of alternatives, and extracting constraints.

We formulate the task that we focus on as follows: given the dialogue history $\mathbf{x}$ and currently accessible KG $\mathcal{G}$, can we extend a transformer model to predict a correct response y in either semantic or NL form? As illustrated in Table 1, this task not only requires the model to accurately retrieve information from the KG, but also needs to do further logical operations on the information. To solve this task, a model should also be able to effectively integrate the dialogue history $\mathbf{x}$ with the KG $\mathcal{G}$.

5 Proposed Approach

Figure 1 illustrates our proposed architecture which contains four main parts: a dialogue history encoder, a differentiable KG reasoning module, a learnable logical operations module, and a response decoder (the transformer model). Note that we experiment with two types of transformers: a causal language model GPT2 (Radford et al., 2019) and an encoder-decoder model T5 (Raffel et al., 2020). For GPT2, we reuse the same encoder that is used at the beginning of the process, i.e., f_{enc} in Figure 1, as the final transformer that generates the response

[1] An example of the triples in $\mathcal{G}$ is a triple e_k^h = gas station, r_k = IsTypeOf, and e_k^t = Chevron. That is, "gas station is the type of Chevron" to this system.

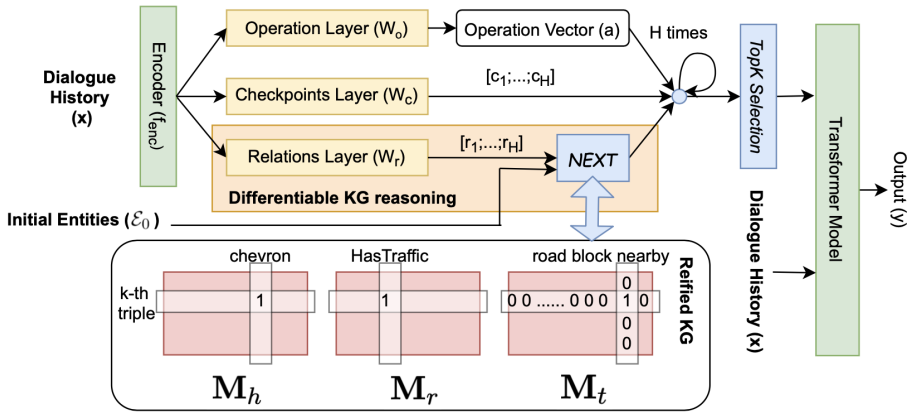


Figure 1: The illustration of proposed DiffKG, which leverages a pretrained transformer model (T5 or GPT2) and the Reified KG. The model generates the response depending on the predicted relation sequence $[r_1; ...; r_H]$, thus being fully interpretable in terms of the used reasoning path.

token by token. For T5, we reuse the same encoder as the encoder of the final transformer with a separate decoder that generates the response. Therefore, this method contains a single transformer model. In following sections we present each module in detail.

5.1 Dialogue History Encoder

We use encoder model to project $\mathbf{x}$ and obtain the dialogue history embedding through $\tilde{\mathbf{x}} = f_{enc}(\mathbf{x}) \in \mathbb{R}^d$, where d is the hidden size of the encoder. The embedding $\tilde{\mathbf{x}}$ is first fed into an operation layer with parameters $\mathbf{W}_o \in \mathbb{R}^{d \times d}$. The operation layer predicts the operation vector $\mathbf{a} = \mathbf{W}_o^T \tilde{\mathbf{x}} \in \mathbb{R}^d$. At the same time, the embedding $\tilde{\mathbf{x}}$ is also fed into a relation layer with parameters $\mathbf{W}_r \in \mathbb{R}^{d \times N_\mathcal{R} H}$. The relation layer predicts the concatenation of a sequence of relations $\mathbf{r} = \{\mathbf{r}_h | 1 \leq h \leq H\}$, where $\mathbf{r}_h \in \mathbb{R}^{N_\mathcal{R}}$ is the relation to be used at the h-th hop in the programmed walking block and H is the maximum number of hops. The embedding $\tilde{\mathbf{x}}$ is also fed into a checkpoints layer with parameters $\mathbf{W}_c \in \mathbb{R}^{d \times 2H}$. This layer produces the concatenation of a sequence of walk-or-check vectors $\mathbf{c} = \{c_h | 1 \leq h \leq H\}$, where $c_h \in \mathbb{R}^2$ is the walk-or-check vector at the h-th hop to determine the weights of the programmed walking module and the operation vector.

$$\begin{aligned} \tilde{\mathbf{x}} &= f_{enc}(\mathbf{x}) \,, \\ \mathbf{a} &= \mathbf{W}_o^T \tilde{\mathbf{x}} \,, \\ \mathbf{r} &= \mathbf{W}_r^T \tilde{\mathbf{x}} \,, \\ \mathbf{c} &= \texttt{softmax}(\mathbf{W}_c^T \tilde{\mathbf{x}}) \,. \end{aligned} \tag{2}$$

5.2 Differential Knowledge Graph Reasoning

To ensure that our model can scale to larger KGs, we adopt the reified KG representation proposed by (Cohen et al., 2019). The reified KG represents the graph $\mathcal{G}$ using three sparse matrices: head matrix $\mathbf{M}_h \in \mathbb{R}^{N_\mathcal{T} \times N_\mathcal{E}}$, relation matrix $\mathbf{M}_r \in \mathbb{R}^{N_\mathcal{T} \times N_\mathcal{R}}$, and tail matrix $\mathbf{M}_t \in \mathbb{R}^{N_\mathcal{T} \times N_\mathcal{E}}$. An entry (i, e) in $\mathbf{M}_h$ or $\mathbf{M}_t$ with value 1 indicates that the i-th triple in the KG has entity e as the head or the tail; an entry (i, r) in $\mathbf{M}_r$ with value 1 indicates that the i-th triple in the knowledge graph has the relation r. Since often in practical settings most entries in the three matrices are zero, saving them into sparse matrices can significantly reduce memory consumption (Cohen et al., 2019).

After predicting the relation sequence $\mathbf{r}$, we start the graph traversal from a given set of initial entities $\mathcal{E}_0 \subseteq \mathcal{E}$. We first map the initial entities into a vector $\mathbf{e}_1 = [\mathbb{1}(e \in \mathcal{E}_0), \forall e \in \mathcal{E}]$. That is, each entry of $\mathbf{e}_1 \in \mathbb{R}^{N_\mathcal{E}}$ has value 1 if that entity is in the initial entities list $\mathcal{E}_0$, otherwise, the entry is zero. We then predict the next (temporary) entity vector $\mathbf{e}_2$ by conducting a `Next` module:

$$\mathbf{e}_{h+1}^r = \texttt{Next}(\mathbf{e}_h, \mathbf{r}_h) \,, \tag{3}$$

where

$$\texttt{Next}(\mathbf{e}_h, \mathbf{r}_h) = \frac{\mathbf{M}_t^T (\mathbf{M}_h \mathbf{e}_h \odot \mathbf{M}_r \mathbf{r}_h)}{||\mathbf{M}_t^T (\mathbf{M}_h \mathbf{e}_h \odot \mathbf{M}_r \mathbf{r}_h)||_2 + \epsilon} \,, \tag{4}$$

Here ϵ is an arbitrary small number to offset the denominator and prevent division by zero. We introduce the normalized `Next` to solve the issue with the method proposed by (Cohen et al.,

2019) for knowledge graph completion defined as $\texttt{Follow}(\mathbf{e}_h, \mathbf{r}_h) = \mathbf{M}_t^T(\mathbf{M}_h\mathbf{e}_h \odot \mathbf{M}_r\mathbf{r}_h)$; since in a dialogue model, we can seldom predict the relation vectors that perfectly match the entity vectors. That is, if directly using the `Follow` module in (Cohen et al., 2019), the $||\mathbf{e}_h||_2$ will not be one and will vanish as the hop number h increases. Specifically, note that in our proposed module, the predicted relations $\mathbf{r}_h$ are independent of the traversed entities $\mathbf{e}_h$. For instance, finding the "distance" of "the nearby gas station" is independent of whether the nearby gas station is "Chevron" or "Shell".

To allow the model to dynamically select the number of reasoning hops, we add a relation type "ToSelf" into $\mathcal{R}$ and connect each entity to itself by "ToSelf". More specifically, the KG will contain triples (e_k^h, r_k, e_k^t) for all $e_k^h = e_k^t \in \mathcal{E}$ and $r_k = \texttt{ToSelf}$.

5.3 Entity Embeddings

At each hop, we further conduct the operation vectors $\mathbf{a}$ on the entities weighted by the entity vector $\mathbf{e}_h$. First, we tokenize each entity and represent it by the concatenation of its token embeddings. This step allows (1) representing entities with longer texts such as phrases and sentences, and (2) eliminating the effort to retrain entity embeddings whenever new entity values are added. The entity embeddings can then be represented as a tensor $\mathbf{E} \in \mathbb{R}^{N_\mathcal{E} \times d \times m}$, where m is the maximum number of tokens of entities[2].

5.4 Learnable Logical Operation and Checkpoints

We compute the transformed entity embeddings by element-wise multiplication of the entity embeddings $\mathbf{E}$ with the entity vector $\mathbf{e}_h$ at the h-th hop. Next, the dot product of the operation vectors and the transformed entity embeddings is passed to a softmax layer as the entity vector at the next hop:

$$\mathbf{e}_{h+1}^a = \texttt{softmax}\left(\mathbf{a}(\mathbf{E} \odot \mathbf{e}_h)\right), \tag{5}$$

Further, at the h-th hop we use the walk-or-check vector $\mathbf{c}_h$ to combine the `Next` and operation modules above. The combined entity vector is given by:

$$\begin{aligned}\mathbf{e}_{h+1} &= \mathbf{c}_h^T \begin{bmatrix} \mathbf{e}_{h+1}^r \\ \mathbf{e}_{h+1}^a \end{bmatrix} \\ &= \mathbf{c}_h^T \begin{bmatrix} \texttt{Next}(\mathbf{e}_h, \mathbf{r}_h) \\ \texttt{softmax}\left(\mathbf{a}(\mathbf{E} \odot \mathbf{e}_h)\right) \end{bmatrix},\end{aligned} \tag{6}$$

5.5 Response Decoder

After H hops reasoning is done, the entities with top-k values in the entity vector $\mathbf{e}_H$ are selected, indicating that they have the highest probability to be retrieved from the graph. These entities are converted into their embeddings in E and multiplied by their values in $\mathbf{e}_H$. These entity embeddings are then concatenated with the dialogue history $\mathbf{x}$. The concatenated vectors are fed as the input into the transformer model to predict the response token by token. The predicted probability distribution over the output space can be written as $P(\cdot|\mathbf{x}, \mathbf{M}_h, \mathbf{M}_r, \mathbf{M}_t)$. Since all components are differentiable, all modules can be trained end-to-end with the dialogue history $\mathbf{x}$ and the reified KG representation $\{\mathbf{M}_h, \mathbf{M}_r, \mathbf{M}_t\}$ using the cross-entropy loss with the ground-truth output y as the labels.

$$L = \sum_{(\mathbf{x},y)} -\log P(y|\mathbf{x}, \mathbf{M}_h, \mathbf{M}_r, \mathbf{M}_t)\,. \tag{7}$$

During the inference time, the reasoning modules (relation layer, operation layer, and checkpoints layer) work exactly the same as the training stage, the only difference is that the response decoder is fed with predicted tokens in previous time steps (inference stage) instead of the ground-truth output (training stage).

6 Experiments

6.1 Datasets

We evaluate our proposed approach on three datasets. Among them, we use Stanford Multi-domain Dialogues (SMD) (Eric et al., 2017) and OpenDialKG (Moon et al., 2019) to test the methods generalizability on different dialogue types (task-oriented / chit-chat) and scales of structured knowledge (pairwise database / universal KG). To further analyze the reasoning ability, we propose a new dataset, SMD-Reasoning, by modifying the output of SMD dataset from natural language responses to actions paired with their reasoning types.

[2] In our experiments, we compute the maximum length of all entities and pad shorter entities to the length of m.

Stanford Multi-domain Dialogues (SMD) The SMD dataset (Eric et al., 2017) is composed of two-speaker conversations, where a driver talks with the car assistant to tackle tasks in three domains: scheduling, navigation, and weather forecasting. Each dialogue focuses on one domain and is paired with a database having the related information. We convert the original database into two formats: (1) the natural language descriptions (NLD) and (2) the KG. The NLD form allows us to investigate the ability of the model to interpret unstructured knowledge, while the KG form could be a more extensible structured knowledge compared to tables.

OpenDialKG OpenDialKG dataset (Moon et al., 2019) is composed of two-speakers recommendation and chit-chat style conversations. Each turn in a dialogue is annotated with the reasoning path on the provided KG, which is filtered from Freebase (Bollacker et al., 2008). The resulting KG has 1,190,658 triples, 100,813 entities and 1,358 relations. We randomly split 70/15/15% for train/valid/test sets as described in (Moon et al., 2019; Jung et al., 2020) since they do not release their splits.

SMD-Reasoning To make SMD dataset suitable for more precise evaluation of reasoning abilities, we manually label and convert it to the SMD-Reasoning dataset. We first remove the natural language part from the original responses and only leave the action word (e.g., inform) along with the information being conveyed. We divide the dataset into three main reasoning types: informing items, selecting min/max, and evaluating true/false. To validate if the models can identify whether the needed knowledge is in the database, we add a new reasoning type for extracting constraints, by removing the needed knowledge from the database and changing the output to "include [knowledge description]" as shown in Table 1. See Appendix A,B for statistics of these datasets.

6.2 Evaluation Metrics

We use different evaluation methods for the three datasets. For SMD, we follow prior work (Yang et al., 2020) and use BLEU (Papineni et al., 2002), and Entity F1 scores on each domain. For OpenDialKG, we follow the descriptions in prior works (Moon et al., 2019; Jung et al., 2020) to evaluate the path@k scores, i.e., if the ground-truth path is ranked top-k in the predicted paths probabilities. Moreover, since our method not only can predict the reasoning path as prior works but also can predict the response, we also use the BLEU score to get the approximated evaluation of the response quality compared to ground-truth. Note that prior work has discussed that BLEU scores may not match human intuition (Liu et al., 2016), but we use them here as an approximated evaluation for reference.

For SMD-Reasoning, the output is more deterministic and does not include diverse sentence structures. Therefore, we compute the F1 score and the *exact match (EM)* score of prediction and the ground-truth. The EM score is calculated by removing the order of the prediction since the labels of SMD-Reasoning dataset follow the order of knowledge description appearing in the original ground-truth responses and may not have the same order as generated outputs. The EM score can be written as:

$$\texttt{EM} = \frac{1}{T}\sum \mathbb{1}(\texttt{sort}(\hat{y}) = \texttt{sort}(y)). \quad (8)$$

where $\hat{y}$ is inferred from the model using argmax sampling and T is total number of examples.

6.3 Implementation Details

Since the proposed method is model-agnostic, we implement it on GPT2 (Radford et al., 2019) and T5 (Raffel et al., 2020). Specifically for the T5 model, we use the unifiedQA-T5 model (Khashabi et al., 2020) which is pretrained on question answering tasks that also need to do reasoning. However, we empirically find that T5 generally has better performance than GPT2, thus using T5 model in most experiments. For OpenDialKG, since the ground-truth relations exist, we take them as an additional supervision signal as (Moon et al., 2019). Also, since we observe that there is only KG reasoning type in OpenDialKG, we do not use operation layer and checkpoints layer for the dataset. The hyperparameter settings are in Appendix C.

6.4 Baselines

We compare our proposed DiffKG model with the state-of-the-art models on OpenDialKG reported in (Moon et al., 2019; Jung et al., 2020) and the state-of-the-art graph-based model on SMD (Yang et al., 2020; Gou et al., 2021) with their reported baselines including sequence-to-sequence models with and without attention (S2S and S2S+Attn) (Luong et al., 2015), pointer to unknown (Ptr-Unk) (Gulcehre et al., 2016),

Model	BLEU	Entity F1			
		All	Sche.	Wea.	Nav.
S2S	8.4	10.3	9.7	14.1	7.0
S2S+Attn	9.3	19.9	23.4	25.6	10.8
Ptr-Unk	8.3	22.7	26.9	26.7	14.9
GraphLSTM	10.3	50.8	69.9	46.6	43.2
BERT	9.13	49.6	57.4	47.5	46.8
Mem2Seq	12.6	33.4	49.3	32.8	20.0
GLMP	12.2	55.1	67.3	54.1	48.4
GraphDialog	13.7	57.4	71.9	59.7	48.6
COMET-graph	14.4	56.7	71.6	48.7	50.4
T5-DiffKG	16.04	56.2	67.2	61.5	46.7

Table 2: The results on SMD dataset. S2S, S2S+Attn, Ptr-Unk, GraphLSTM, BERT, Mem2Seq, GLMP, GraphDialog are reported from (Yang et al., 2020) and COMET-graph from (Gou et al., 2021). Our DiffKG achieves the highest BLEU and comparable F1 scores with baselines.

Model	path@1	path@5	path@10	BLEU
Seq2Seq	3.1	29.7	44.1	-
Tri-LSTM	3.2	22.6	36.3	-
EXT-ED	1.9	9.0	13.3	-
DialKG	13.2	35.3	47.9	-
Seq2Path	14.92	31.1	38.68	-
AttnFlow	17.37	30.68	39.48	-
AttnIO-AS	23.72	43.57	52.17	-
T5-NoInfo	-	-	-	14.51
T5-DiffKG	26.80	54.33	61.75	15.37

Table 3: The results on OpenDialKG dataset. The four baselines from Seq2Seq to DialKG Walker are reported from (Moon et al., 2019) and the other three baselines from Seq2Path to AttnIO-AS are reported from (Jung et al., 2020). Our DiffKG achieves the highest path@k scores and is the only one that can simultaneously generate responses.

GraphLSTM (Peng et al., 2017), BERT (Devlin et al., 2019), Mem2Seq (Madotto et al., 2018) and GLMP (Wu et al., 2019). We follow their metrics and train our model on their preprocessed data for fair comparisons. To further analyze the reasoning ability, we propose two more baselines based on different ways of leveraging pretrained language models. (1) **NoInfo** model does not take any format of knowledge as the input, aiming to test the performance of a fine-tuned vanilla transformer model on each dataset. (2) **FlatInfo** model constructs the input by concatenating the dialogue history with the NLD form of knowledge as (Beygi et al., 2022), allowing us to investigate the ability of the model to interpret unstructured knowledge.

Test KG	**Method**	**EM**	**F1**
Fixed	GPT2-NoInfo	10.71	43.78
	GPT2-FlatInfo	14.08	47.57
	GPT2-DiffKG	16.39	51.06
	T5-NoInfo	10.50	44.27
	T5-FlatInfo	28.99	66.15
	T5-DiffKG	27.52	63.93
Shuffled	T5-FlatInfo	17.02	54.51
	T5-DiffKG	27.52	64.00

Table 4: The results on SMD-Reasoning dataset.

6.5 Results

The results on SMD and OpenDialKG are shown in Table 2 and Table 3. On SMD dataset, we observe that DiffKG outperforms the baselines on BLEU by 11.4% (relative change of 16.04 and 14.4) and achieves comparable entity F1 scores with GLMP, GraphDialog and COMET-graph. DiffKG might not improve the entity F1 scores because that prior works group the text inside an entity together (e.g., "road block nearby" becomes a single word "road_block_nearby" in vocabularies). In contrast, we use a universal tokenizer so as to prevent heavy preprocessing and specialized vocabularies. This means that DiffKG can perform similarly with state-of-the-art to retrieve knowledge without a tokenizer specified for each dataset. On OpenDialKG dataset, we observe that DiffKG outperforms the baselines in terms of path@k scores and can simultaneously outperform T5 in terms of Entity F1 and BLEU. These demonstrate that DiffKG can retrieve accurate paths for reasoning and effectively incorporate reasoning into response generation.

We also investigate the results of SMD-Reasoning dataset as shown in Table 4. We find that DiffKG improves NoInfo by 16.6% and 44.4% F1 scores respectively on GPT2 and T5 models. This demonstrates that DiffKG can utilize knowledge effectively to improve the generation without access to information. In contrast, although FlatInfo gives similar performances as DiffKG on the SMD-Reasoning dataset, it cannot be run on OpenDialKG due to computational costs. More specifically, FlatInfo requires the knowledge graph to be transformed into sentences, which will result in at least a million tokens as the model inputs for OpenDialKG (since the number of triples is a million without designed subgraph sampling), which is not a practical number.

Method	Domains						Reasoning Types							
	Schedule		Navigation		Weather		Inform		Selection		Extraction		True/False	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
GPT2-NoInfo	3.49	45.7	4.63	41.6	27.5	46.8	5.03	45.2	1.45	47.4	3.06	24.0	68.6	68.6
GPT2-DiffKG	9.30	53.0	9.65	47.6	34.4	56.5	8.04	50.8	0.00	48.5	31.6	53.5	56.9	56.9
T5-NoInfo	0.00	44.6	4.63	40.9	29.0	50.7	3.02	44.9	8.70	49.1	1.02	25.2	70.6	70.6
T5-DiffKG	20.9	63.8	19.3	61.9	48.1	68.1	18.1	61.7	11.6	62.4	50.0	73.5	70.6	70.6

Table 5: Detailed Evaluation Results of SMD-Reasoning dataset

SMD-Reasoning	User: check the date and time of my doctor's appointment (Reasoning Path: Doctor Appointment $\xrightarrow{\text{HasDate, HasTime, ToSelf}}$ Tuesday, 11am, doctor appointment) DiffKG: inform 11 am tuesday doctor appointment
	User: Car I need to get to a gas station, please show me the nearest one Assistant: There is Valero 7 miles away with moderate traffic on our way User: Alright, where is it located? (Reasoning Path: Gas Station $\xrightarrow{\text{IsTypeOf}}$ Valero $\xrightarrow{\text{HasAddress, ToSelf}}$ 200 Alester Ave, Valero) DiffKG: inform 200 Alester Ave Valero
OpenDialKG	Speaker A: Do you have any info on Toni Kroos? (Reasoning Path: Toni Kroos $\xrightarrow{\sim\text{Player Statistics}}$ Germany national football team) DiffKG: Toni Kroos is German footballer who plays for the Germany national football team.

Table 6: Generated examples and the reasoning path.

6.6 Quantitative Analysis

To test the robustness of the methods towards accurately locating information, we shuffle the information order. This evaluation is to simulate the cases that extra information is arbitrarily added when deploying a dialogue system. Specifically, the order of the knowledge context for FlatInfo and the order of knowledge triples are changed during inference time. As shown in the last two rows in Table 4, the performance of FlatInfo drops while DiffKG remains about the same. This indicates that the slight superior performance of FlatInfo with the original order can come from the blackbox tricks to group the nearby knowledge in the inputs. When this implicit trick is broken down, the DiffKG shows much better robustness and performance.

To investigate the difficulty of each domain and reasoning type, we divide the results accordingly in Table 5. As presented in the domains part, the models achieve the highest EM and F1 on the weather domain. We conjecture the reason is that the weather domain includes more reasoning types (weather:4, navigate:3, schedule:2 as in Appendix A Table 9), thus reflecting more balanced reasoning ability. In the reasoning types part, we observe that true/false is less well coped by DiffKG; however, DiffKG improves the extraction. This shows that DiffKG can effectively check the existence of required knowledge and then query the database.

Regarding to the computational costs (on SMD-Reasoning dataset using T5 model), we found that DiffKG requires about 5.85GB memory during training and has 30ms inference latency. This could be an acceptable add-on memory usage and inference time compared to a model without knowledge reasoning (3.13GB; 30ms). Especially when a baseline like FlatInfo consumes much more (18.56GB; 50ms).

6.7 Qualitative Analysis

We visualize the generated examples and the symbolic reasoning path by DiffKG on SMD-Reasoning and OpenDialKG datasets in Table 6. The examples show that DiffKG can capture some naturally occurring phenomena in this dataset: (1) the KG reasoning path can be 1 to multiple hops; (2) the reasoning will diffuse to multiple paths (e.g., DiffKG simultaneously applies "HasDate","HasTime","ToSelf" to "Doctor Appointment"). Along with analyses in previous subsections, we observe that DiffKG can extract interpretable reasoning paths and generate corresponding outputs using reasonable computational costs.

However, even though DiffKG can maintain or improve performance while doing interpretable reasoning on any scaled KG, errors might happen in

Error Type	Example
Unclear Information Requirement	User: What gas stations are here? Ground-truth: inform Chevron DiffKG: inform 5 miles, Chevron (Our comment: Although this prediction is considered wrong for the EM metric, the "5 miles" of DiffKG output is the correct distance from "Chevron" and might be needed in a good response.)
Incomplete Reasoning but Faithful Response	User: Where is the closest grocery store? Ground-truth: Inform 4 miles, Whole Foods, Safeway. DiffKG: inform 4 miles, grocery store, 819 Alma St, Whole Foods (Our comment: The 4 miles, grocery store, 819 Alma St are all correct entities about Whole Foods. Nonetheless, this reasoning process neglects another grocery store Safeway which is also 4 miles away.)
Correct Reasoning but Wrong Response	Speaker A: Do you know Don Hall? Ground-truth: Don Hall wrote the Princess and the Frog a romance story starring Jenifer Lewis. Do you like Romance? Reasoning Path: Don Hall $\xrightarrow{\sim\text{written by}}$ The Princess and the Frog DiffKG: Yes, he wrote The Little Dolls. (Our comment: The reasoning path is correct to find out the script written by Don Hall. However, the generation process fails to properly utilize the retrieved entity.)

Table 7: The error analysis with three major error types across datasets.

some cases. As listed in Table 7, we found that across the datasets, the three main error types of DiffKG are: (1) unclear information requirement in the dataset, (2) incomplete reasoning ability but faithful response generation, and (3) correct reasoning but hallucinated response prediction. We argue that the first error type mainly comes from the mismatch among data points in the dataset and may not be able to be dealt with by models. The second error type indicates that the KG reasoning module sometimes cannot retrieve all the needed information. The third error type indicates that the module producing final output may not fully utilize the retrieved information. These three points might provide a direction for further improvement.

7 Conclusion and Future Work

For a dialogue system, an effective reasoning method over structured databases is important. In this work, we proposed DiffKG, an end-to-end model-agnostic method that does symbolic reasoning on any scale of KGs to enhance response generation. Experiments demonstrated that using DiffKG, models are able to generate responses with interpretable KG reasoning paths at a modest extra cost.

This work can be extended in various ways. While we solely consider efficient large-scale KG reasoning in dialogue generation, future work can incorporate domain fusion methods to consider the generalizability over domains or simultaneously use relation information. Moreover, since DiffKG is a simple large-scale structured knowledge-empowered transformer with flexible entity values, future work can extend it to dialogue generation that needs to do table and text mixed reasoning and that needs to do both KG reasoning and other goals such as personalized dialogues, storytelling, etc.

References

Sajjad Beygi, Maryam Fazel-Zarandi, Alessandra Cervone, Prakash Krishnan, and Siddhartha Reddy Jonnalagadda. 2022. Logical reasoning for task oriented dialogue systems. *arXiv preprint arXiv:2202.04161*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*.

William W Cohen, Haitian Sun, R Alex Hofer, and Matthew Siegler. 2019. Scalable neural methods for reasoning with a symbolic knowledge base. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Julian Martin Eisenschlos, Maharshi Gor, Thomas Müller, and William W Cohen. 2021. Mate: Multi-view attention for table transformer efficiency. *arXiv preprint arXiv:2109.04312*.

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*.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yanjie Gou, Yinjie Lei, Lingqiao Liu, Yong Dai, and Chunxu Shen. 2021. Contextualize knowledge bases with transformer for end-to-end task-oriented dialogue systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1.

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, pages 20179–20191. Curran Associates, Inc.

Rishabh Joshi, Vidhisha Balachandran, Shikhar Vashishth, Alan Black, and Yulia Tsvetkov. 2020. Dialograph: Incorporating interpretable strategy-graph networks into negotiation dialogues. In *International Conference on Learning Representations*.

Jaehun Jung, Bokyung Son, and Sungwon Lyu. 2020. Attnio: Knowledge graph exploration with in-and-out attention flow for knowledge-grounded dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Dan Jurafsky and James H. Martin. 2009. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

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*.

Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William Cohen. 2021. Differentiable open-ended commonsense reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Yinong Long, Jianan Wang, Zhen Xu, Zongsheng Wang, Baoxun Wang, and Zhuoran Wang. 2017. A knowledge enhanced generative conversational service agent. In *Proceedings of the 6th Dialog System Technology Challenges (DSTC6) Workshop*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

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)*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the 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*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5.

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

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building End-to-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Shane Storks, Qiaozi Gao, and Joyce Y Chai. 2019. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*.

Yi-Lin Tuan, Yun-Nung Chen, and Hung-Yi Lee. 2019. Dykgchat: Benchmarking dialogue generation grounding on dynamic knowledge graphs. 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)*.

Yi-Lin Tuan, Wei Wei, and William Yang Wang. 2020. Knowledge injection into dialogue generation via language models. *arXiv preprint arXiv:2004.14614*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. *International Conference on Learning Representation*.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W Cohen. 2020. Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge. *arXiv preprint arXiv:2007.00849*.

Qingyue Wang, Yanan Cao, Junyan Jiang, Yafang Wang, Lingling Tong, and Li Guo. 2021. Incorporating specific knowledge into end-to-end task-oriented dialogue systems. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *European Association for Computational Linguistics (EACL)*, pages 438–449.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

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*.

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)*.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural generative question answering. In *Proceedings of the Workshop on Human-Computer Question Answering*.

Xueliang Zhao, Wei Wu, Chongyang Tao, Can Xu, Dongyan Zhao, and Rui Yan. 2019. Low-resource knowledge-grounded dialogue generation. In *International Conference on Learning Representations*.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*.

Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. 2017. Flexible end-to-end dialogue system for knowledge grounded conversation. *arXiv preprint arXiv:1709.04264*.

A Dataset Details

The statistics of datasets are in Table 8 and Table 9. The OpenDialKG dataset is under CC-BY-NC-4.0 license. These datasets can be used for research purposes.

Data	Train	Validation	Test
SMD	2425	302	304
OpenDialKG	10971	2351	2351

Table 8: The number of dialogues in each data split.

	Schedule	Navigation	Weather
Inform	364	1133	236
Selection	-	686	39
True/False	-	-	543
Extraction	173	474	214

Table 9: The statistics of SMD Reasoning dataset with respect to domains and reasoning types. Note that since reasoning types are classified on turn-level, the total number in this table is larger than in Table 8 that counted on dialogue-level.

B SMD KG Construction

We write a simple, automatic program to construct KGs for SMD dataset mapped from the original annotated tables.

For the schedule and navigation domains in SMD, we directly map their table attributes to the relations $\mathcal{R}$ in our constructed KG. For the weather domain, we split each weather report into low temperature, high temperature, and weather. The resulting number of relations is 29, and the relations are listed in Table 10.

In the schedule and navigation domain, each item in the original database with multiple attributes are transformed to KG triples as (event/point-of-interest, attribute, attribute value), e.g., `(tennis activity, HasTime, 7pm)` in schedule domain or `(Chevron, HasType, gas station)` in navigation domain.

In the weather domain, we add additional entities named "ReportID$digits$", where $digits$ will be replaced with an ID number. Each item in the original database is in the format: (item, location, $location), (item, $date, $weather_report), where the $weather_report contains multiple information not simultaneously needed. To make the KG of weather consistent with the KGs of schedule and navigation, we transform each item into (ReportID, location, $location), (ReportID, HasDate, $date), (ReportID, HasWeather, $weather), (ReportID, HasLowTemp, $low_temparature), (ReportID, HasHighTemp, $high_temparature).

Domain	Relations
Schedule	HasTime, HasDate, HasParty, HasRoom, HasAgenda, IsTimeOf, IsDateOf, IsPartyOf, IsRoomOf, IsAgendaOf
Navigation	HasAddress, HasType, HasTraffic, HasDistance, IsAddressOf, IsTypeOf, IsTrafficOf, IsDistanceFrom
Weather	IsEqualTo, HasLocation, HasWeather, HasLowTemp, HasHighTemp, HasDate, IsWeatherOf, IsLowTempOf, IsHighTempOf, IsLocationOf, IsDateOf

Table 10: The relations used in each domain in SMD dataset.

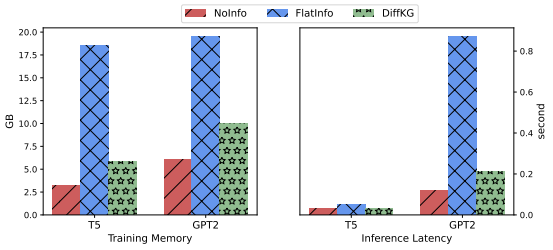


Figure 2: The comparison of the consumed training memory and inference latency.

C Experiment Details

The hyperparameters we set for DiffKG are d=the hidden size of the used pretrained trasnformer (T5-small: d=512; GPT2:d=768), H=5, max norm=1.0, batch size=16, and gradient accumulation steps=2 for at most 50 epochs and train the model learning rate$\in \{5 \times 10^{-5}, 6.25 \times 10^{-5}\}$ (found that 6.25×10^{-5} is better) without learning rate decay. Our experiments were single runs with random initialization and were not further fine-tuned.

D Computational Cost Analysis

As plotted in Figure 2, on SMD-Reasoning dataset, the consumed memory of FlatInfo is thrice the memory needed for DiffKG at training time, and its

latency is about twice at inference time. The difference in inference latency is even larger with GPT2 as the backbone model. The reason is that the computational cost of a causal language model such as GPT2 largely depends on the input sequence length, which is one of the main issues of FlatInfo.