Generative Subgraph Retrieval for Knowledge Graph–Grounded Dialog Generation

Jinyoung Park[1*] **Minseok Joo**[1] **Joo-Kyung Kim**[2†] **Hyunwoo J. Kim**[1†]
[1]Korea University, [2]Amazon AGI
{lpmn678, wlgkcjf87, hyunwoojkim}@korea.ac.kr jookyk@amazon.com

Abstract

Knowledge graph–grounded dialog generation requires retrieving a dialog-relevant subgraph from the given knowledge base graph and integrating it with the dialog history. Previous works typically represent the graph using an external encoder, such as graph neural networks, and retrieve relevant triplets based on the similarity between single-vector representations of triplets and the dialog history. However, these external encoders fail to leverage the rich knowledge of pretrained language models, and the retrieval process is also suboptimal due to the information bottleneck caused by the single-vector abstraction of the dialog history. In this work, we propose Dialog generation with Generative Subgraph Retrieval (DialogGSR), which retrieves relevant knowledge subgraphs by directly generating their token sequences on top of language models. For effective generative subgraph retrieval, we introduce two key methods: (i) structure-aware knowledge graph linearization with self-supervised graph-specific tokens and (ii) graph-constrained decoding utilizing graph structural proximity-based entity informativeness scores for valid and relevant generative retrieval. DialogGSR achieves state-of-the-art performance in knowledge graph–grounded dialog generation, as demonstrated on OpenDialKG and KOMODIS datasets.

1 Introduction

The goal of dialog generation is to generate an informative and appropriate response given an input dialog. Pretrained Language Models (PLMs) have demonstrated promising performance on the dialog generation (Roberts et al., 2020; Touvron et al., 2023; Achiam et al., 2023). However, they often generate irrelevant, factually incorrect, or hallucinatory responses since the generation process heavily depends on the internal parameters of the language models (Lewis et al., 2020; Shuster et al., 2021). To mitigate these issues, several studies (Wang et al., 2020; Zhao et al., 2020) have explored knowledge-grounded dialog generation models, which incorporate external knowledge to generate more factually accurate responses. Some approaches utilize unstructured texts such as Wikipedia articles (Dinan et al., 2019) and internet web pages (Ghazvininejad et al., 2018) while others (Moon et al., 2019; Galetzka et al., 2021; Tuan et al., 2022; Kang et al., 2023) leverage structured knowledge graphs (KGs) to capture both the relational and semantic information for grounding dialog responses.

Many existing knowledge graph–grounded dialog generation models (Tuan et al., 2022; Kang et al., 2023) employ encoder-based retrieval methods. They encode the dialog history into a single vector and then use it on another encoder (*e.g.*, bi-encoder) to retrieve relevant triplets from the KG. However, this approach can lead to an information bottleneck due to the limited capacity of a single vector to represent long and complex multi-turn dialogs (Humeau et al., 2020; Cao et al., 2021; Lee et al., 2022). Moreover, these methods (Galetzka et al., 2021; Tuan et al., 2022; Kang et al., 2023) often rely on separate models, such as graph neural networks (GNNs), to encode the knowledge graphs, which limits the integration of natural language comprehension capabilities of PLMs.

Recent studies (Lee et al., 2022; Sun et al., 2023) have addressed the information bottleneck issue by applying generative retrieval methods, which cast retrieval as an autoregressive generation process to facilitate direct interactions between query context and knowledge paragraphs. Despite this progress, most generative retrieval works focus solely on natural language-based knowledge, employing conventional token representations and decoding strategies, which do not fully capture the structure and properties of knowledge graphs.

[*] Part of this work was done during an internship at Amazon AGI.

[†] Co-corresponding authors.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 21167–21182
November 12-16, 2024 ©2024 Association for Computational Linguistics

To address the aforementioned issues, we propose **Dialog** Generation model with **G**enerative **S**ubgraph **R**etrieval (**DialogGSR**), which integrates generative subgraph retrieval with response generation. Our proposed method adopts two key graph-specialized techniques: (1) a structure-aware knowledge graph linearization for effective graph representation and (2) graph-constrained decoding for valid subgraph retrieval. Our knowledge graph linearization approach introduces a small set of special token embeddings to account for both the structural positioning of knowledge entities and the reverse relationships between them. By self-supervising these special tokens using a knowledge graph reconstruction loss, the method effectively represents the knowledge graph. The graph-constrained decoding facilitates autoregressively retrieving the knowledge considering the graph structural information, thus generating valid and relevant knowledge subgraphs. Since DialogGSR utilizes pretrained language models for both subgraph retrieval and dialog generation, it leverages the pretrained language models' internal knowledge in both tasks.

We evaluate DialogGSR on two KG–grounded dialog generation datasets: OpenDialKG (Moon et al., 2019) and KOMODIS (Galetzka et al., 2020). Our proposed method shows the best performance on both benchmark datasets.

Our contributions are three-fold as follows:

- We propose Dialog generation with Generative Subgraph Retrieval (DialogGSR), which retrieves the relevant knowledge subgraphs by generating their token sequences.
- We design knowledge graph linearization for effective graph representations and graph-constrained decoding for retrieving valid and relevant subgraphs.
- We show the state-of-the-art response generation performance on two benchmark datasets, OpenDialKG and KOMODIS.

2 Related Works

2.1 Generative Retrieval

Retrieving relevant information from a large corpus such as a text corpus or a knowledge base is crucial in many tasks (Chen et al., 2017; Thorne et al., 2018; Lewis et al., 2020; Izacard and Grave, 2021). Recent studies (Cao et al., 2021; Bevilacqua et al., 2022; Wang et al., 2022; Lee et al., 2022, 2023) have demonstrated that generative retrieval models can be more effective than conventional encoder-based retrieval models. They cast retrieval tasks as generation tasks, where relevant sequences are generated rather than retrieved given input queries. Several studies (Chen et al., 2022a; Thorne, 2022; Lee et al., 2022; Yu et al., 2023; Xu et al., 2023; Luo et al., 2024) have shown the effectiveness of generative retrieval in various knowledge-intensive natural language processing tasks. Motivated by these works, we propose a generative subgraph retrieval model with knowledge graph linearization and graph-constrained decoding for effective graph representation and generation.

2.2 Knowledge-Grounded Dialog Generation

Many language generation approaches leverage pretrained language models (PLMs) (Radford et al., 2019; Devlin et al., 2019; Roberts et al., 2020; Thoppilan et al., 2022; Touvron et al., 2023; Achiam et al., 2023), showing strong performance. However, they often suffer from the hallucination issue (Dušek et al., 2018; Balakrishnan et al., 2019; Dušek et al., 2020), which generates plausible but factually wrong responses since they rely on the models' internal parameters. To address this problem, recent works (Moon et al., 2019; Dinan et al., 2019; Lian et al., 2019) have proposed to augment the models with external knowledge sources. This approach is effective in generating factually accurate responses in various language generation tasks (Fernandes et al., 2019; Huang et al., 2020; Yasunaga et al., 2021; Yu et al., 2022; Zhang et al., 2022b). Regarding dialog generation, various works incorporate external knowledge graph into the generation (Moon et al., 2019; Zhou et al., 2018; Tuan et al., 2019; Zhang et al., 2020; Zhou et al., 2021). For instance, Space Efficient (Galetzka et al., 2021) proposes an efficient method to encode knowledge triplets. RHO (Ji et al., 2023) generates responses with the dialog history and knowledge graph represented by graph embedding methods (*e.g.*, TransE (Bordes et al., 2013)). DiffKG (Tuan et al., 2022) uses a graph reasoning encoder on top of sparse matrices for graph representations. SURGE (Kang et al., 2023) applies GNNs to retrieve context-relevant subgraphs. Different from these works, our work autoregressively retrieves the context-relevant subgraphs and then generates knowledge-grounded dialogs without requiring separate knowledge graph modules.

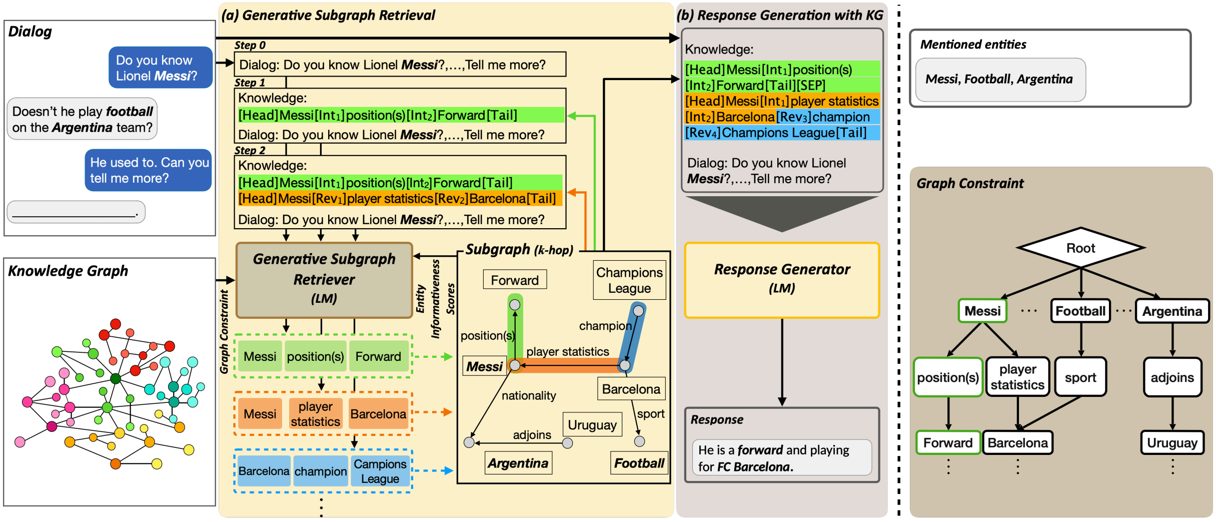
Dialog
Do you know Lionel Messi?
Doesn't he play football on the Argentina team?
He used to. Can you tell me more?
Knowledge Graph
(a) Generative Subgraph Retrieval
Step 0
Dialog: Do you know Lionel Messi?,...,Tell me more?
Step 1
Knowledge:
[Head]Messi[Int1]position(s)[Int2]Forward[Tail]
Dialog: Do you know Lionel Messi?,...,Tell me more?
Step 2
Knowledge:
[Head]Messi[Int1]position(s)[Int2]Forward[Tail]
[Head]Messi[Rev1]player statistics[Rev2]Barcelona[Tail]
Dialog: Do you know Lionel Messi?,...,Tell me more?
Generative Subgraph Retriever (LM)
Graph Constraint
Entity Informativeness Scores
Messi
position(s)
Forward
Messi
player statistics
Barcelona
Barcelona
champion
Campions League
Subgraph (k-hop)
Forward
Champions League
position(s)
champion
player statistics
Messi
Barcelona
nationality
sport
adjoins
Uruguay
Argentina
Football
(b) Response Generation with KG
Knowledge:
[Head]Messi[Int1]position(s)
[Int2]Forward[Tail][SEP]
[Head]Messi[Int1]player statistics
[Int2]Barcelona[Rev3]champion
[Rev4]Champions League[Tail]
Dialog: Do you know Lionel Messi?,...,Tell me more?
Response Generator (LM)
Response
He is a forward and playing for FC Barcelona.
Mentioned entities
Messi, Football, Argentina
Graph Constraint
Root
Messi
Football
Argentina
position(s)
player statistics
sport
adjoins
Forward
Barcelona
Uruguay

Figure 1: The overall inference process of DialogGSR. DialogGSR consists of a generative subgraph retriever and response generator. (a) Generative subgraph retrieval autoregressively retrieves subgraphs via generative subgraph retriever with graph-constrained decoding based on entity informativeness score. In step 0, given the dialog, GSR retrieves the most relevant triplets by referring to the graph constraint. In step 1, given the dialog and the prompt-augmented triplet, we generatively retrieve the next triplets. (b) Resposne generator generates the responses with the dialog and the prompt-augmented generated subgraph.

3 Methods

We propose a retrieval-augmented dialog generation approach that retrieves contextually relevant subgraphs from knowledge graphs to generate better responses. Our model, **Dialog** Generation model with **G**enerative **S**ubgraph **R**etrieval (DialogGSR) consists of a generative subgraph retriever and a response generator. We first define the task of knowledge graph–grounded dialog generation (Sec. 3.1). Next, we propose **G**enerative **S**ubgraph **R**etrieval (GSR), which autoregressively retrieves subgraph by applying structure-aware knowledge graph linearization and graph-constrained decdoing (Sec. 3.2). We then present a response generator, which performs subgraph–grounded dialog generation (Sec 3.3). Finally, we provide the training details of DialogGSR including our self-supervised knowledge graph reconstruction loss (Sec 3.4). The inference process of DialogGSR is illustrated in Figure 1.

3.1 KG–Grounded Dialog Generation

The goal of knowledge graph–grounded dialog generation is to generate a dialog response by jointly reasoning over a dialog history and a knowledge graph. We represent a dialog history as a token sequence, $\boldsymbol{x} = [x_1, x_2, \dots, x_n]$, where $x_i \in \mathcal{V}$ is the i-th token of the dialog history and $\mathcal{V}$ denotes the vocabulary set. A knowledge graph is defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ is the set of entities and $\mathcal{R}$ is the set of relations. $\mathcal{T}$ denotes the set of triplets, $(e_h, r, e_t) \in \mathcal{T}$, each of which are composed of a head entity $e_h \in \mathcal{E}$, a tail entity $e_t \in \mathcal{E}$, and a relation $r \in \mathcal{R}$ between the two entities. We use k-hop subgraph linked to the entities mentioned in the input dialog as retrieval candidates following previous works (Kang et al., 2023). The example of a extracted candidate subgraph is in Figure 3. We formulate knowledge graph–grounded dialog generation as follows:

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}, \mathcal{G}) = \prod_{j=1}^{t} p_\theta(y_j|\boldsymbol{x}, \boldsymbol{y}_{<j}, \mathcal{G}), \qquad (1)$$

where $\boldsymbol{y} = [y_1, y_2, \dots, y_t]$ is the output response, t is the length of the response, and $\boldsymbol{y}_{<j} = [y_1, \dots y_{j-1}]$ denotes the generated sequence at the previous time steps. Since a KG can include a huge number of irrelevant entities and relations, KG-grounded dialog generation works generally retrieve subgraphs related to the dialog context for the efficiency and effectiveness.

3.2 Generative Subgraph Retrieval

We introduce **G**enerative **S**ubgraph **R**etrieval (GSR), which autoregressively retrieves a knowledge subgraph $\hat{\mathcal{G}}$. Since a knowledge subgraph can be represented as a set of triplets, retrieving sequences of knowledge triplets is equivalent to subgraph retrieval. Many subgraph retrieval methods in dialog generation (Zhang et al., 2022a; Kang et al., 2023) compute the relevance score between the dialog history and each knowledge triplet and retrieve the triplets with the highest scores.

However, these methods often suffer from the information bottleneck problem (Izacard et al., 2020; Luan et al., 2021), as they encode long, multi-turn dialog histories into a single fixed-length vector, which has a limited capacity to accurately represent complex multi-turn dialogs. Moreover, these approaches require independent knowledge graph encoders to represent knowledge graphs, which cannot fully leverage the pre-trained knowledge embedded in the pretrained language models.

To address these limitations, generative subgraph retrieval casts graph retrieval as a graph generation, enabling more direct interaction between the dialog context and the knowledge graph by representing the graph with a token sequence. For effective generative retrieval, our GSR model incorporates two novel techniques: (1) Structure-aware knowledge graph linearization, which converts the knowledge graph into token sequences enriched with learnable special tokens that capture the connectivity and reverse relations between entities, and (2) Graph-constrained decoding, which ensures the language model to generate valid knowledge subgraphs by predicting the next tokens based not only on the language model's scores but also on the relational proximities of entities within the graph.

Structure-aware knowledge graph linearization. The goal of knowledge graph linearization is to convert a knowledge graph into a token sequence comprehensible to language models. Our structure-aware knowledge graph linearization augments a sequence of knowledge graph tokens with graph-specific learnable special tokens to help the language model understand the graph's structural information without separate graph encoders. Different from prior graph linearization methods such as Xu et al. (2023), which do not take into account multi-hop graph connections and reverse relations, our structure-aware knowledge graph linearization better captures and effectively represents the underlying structures of knowledge graphs.

Specifically, if there are connected triplets (*e.g.*, (e_1, r_1, e_2) and (e_2, r_2, e_3)), we efficiently represent the path as `[Head]` e_1 `[Int`$_1$`]` r_1 `[Int`$_2$`]` e_2 `[Int`$_3$`]` $r_2 \ldots e_{l+1}$ `[Tail]`. To represent multiple disconnected triplets or paths, we insert `[SEP]` between them. For more expressive representations of the special tokens, we use multiple consecutive tokens to represent each of `[Int]`,`[Rev]`, which improves the performance as in Section A.1.

Additionally, since a knowledge graph can contain reverse relations, representing them is crucial in knowledge graph processing (Feng et al., 2020; Qi et al., 2023; Zhu et al., 2024). Therefore, we introduce another special token `[Rev]` for reverse relations when (1) there is a mentioned entity that is the tail of a triplet because the decoding always starts with one of the mentioned entities, or (2) two triplets are connected with opposite directions (*e.g.*, (e_1, r_1, e_2) and (e_3, r_2, e_2)). We effectively represent reverse relations by adding special tokens `[Rev`$_1$`]` and `[Rev`$_2$`]` without modifying the relation tokens. For example, given a triplet (e_3, r_2, e_2), the corresponding triplet with the reverse relation $(e_2, \tilde{r}_2, e_3)$ is represented as `[Head]` e_2 `[Rev`$_1$`]` r_2 `[Rev`$_2$`]` e_3 `[Tail]`.

In sum, we represent the subgraph $\hat{\mathcal{G}}$ as the concatenation of the knowledge paths converted with the special tokens as follows:

$$\begin{aligned} \boldsymbol{z}_{\hat{\mathcal{G}}} = & \texttt{[Head]} e_1 \texttt{[Int}_1\texttt{]} r_1 \ldots \\ & e_{l+1} \texttt{[Tail][SEP][Head]} e_k \cdots . \end{aligned} \tag{2}$$

All the special tokens are learnable with soft prompting. They are learned with both downstream task loss and knowledge graph reconstruction loss, which will be introduced in Section 3.4. Our structure-aware knowledge graph linearization with the special tokens helps the language model capture knowledge graph information without any separate knowledge graph encoders, which leads to the full utilization of the power of PLMs.

Graph-constrained decoding. The language model is prone to generating invalid or irrelevant subgraphs due to its bias, often disregarding the knowledge graph structures (Cao et al., 2021; Chen et al., 2022b). To address this issue, we introduce a graph-constrained decoding method that ensures the generation of valid and relevant subgraphs. Formally, given a dialog $\boldsymbol{x}$ and the previously generated segments of linearized knowledge path $\pi_{<t}$, the log probability of the next token w is computed with $\log p_{\text{vocab}}(w|\boldsymbol{x}, \pi_{<t}, C_{\mathcal{M}})$. Here, $C_{\mathcal{M}}$ represents a prefix tree derived from the ego-graph (Zhu et al., 2021) of a set of mentioned entities $e_m \in \mathcal{M}$ as depicted in Figure 1 (right). The mentioned entities are the entities that appear in the input dialog history and correspond to entities in the knowledge

graph (Kang et al., 2023). For example, given the dialog “Do you know Lionel Messi?" in Figure 1, the entity ‘Messi’ is a mentioned entity since it exists in the knowledge graph. The next token prediction probability p_{vocab} is restricted to tokens within the valid set defined by the constraint $C_{\mathcal{M}}$ (*i.e.*, $(\pi_{<t}, w) \in \mathcal{C}_{\mathcal{M}}$). This constraint ensures that only valid knowledge subgraphs are generated.

In addition, to account for the importance of each entity in the knowledge graph, we introduce a graph-based next-token prediction probability, which is defined as:

$$\begin{aligned} &\log \tilde{p}(w|\boldsymbol{x}, \pi_{<t}, C_{\mathcal{M}}) \\ &= \alpha \cdot \log p_{\text{vocab}}\left(w|\boldsymbol{x}, \pi_{<t}, C_{\mathcal{M}}\right) \\ &+ (1-\alpha) \cdot \log p_{\text{graph}}\left(w|\pi_{<t}, C_{\mathcal{M}}\right), \end{aligned} \tag{3}$$

where p_{graph} is the probability of predicting the next token based on graph structure, and α is a hyperparameter controlling the balance between the language model and graph-based predictions. If the next token w corresponds to a tokenized entity, the probability p_{graph} is defined as:

$$p_{\text{graph}}\left(w|\pi_{<t}, C_{\mathcal{M}}\right) \propto \mathcal{S}(e_i, \mathcal{M}), \tag{4}$$

where $\mathcal{S}\left(e_i, \mathcal{M}\right)$ is the entity informativeness score of entity e_i with respect to the mentioned entity set $\mathcal{M}$. In cases where all entities have identical informativeness scores, the next token prediction is driven purely by p_{vocab}.

To capture the structural proximity between entity e_i and mentioned entities $e_m \in \mathcal{M}$ on the graph, we define the structure-based entity Informative Score (IS) as

$$\mathcal{S}(e_i, \mathcal{M}) = \frac{1}{|\mathcal{M}|} \sum_{e_m \in \mathcal{M}} s(e_i, e_m), \tag{5}$$

where $s(e_i, e_m)$ denotes the graph structural proximity between entity e_i and e_m. The proximity can be measured using methods such as the shortest path and common neighbors (Katz, 1953; Brin, 1998; Gasteiger et al., 2019). A typical approach for measuring graph structural proximity is counting the number of connections between node pairs, which can be defined as $s_{\text{con}}(e_i, e_m) = \sum_{\mathcal{N}(e_m)} \mathbf{1}(e_i = e_m)$, where $\mathcal{N}(e)$ is the neighborhood set of entity e.

However, the connection-based proximity measurement fails to account for multi-hop relations. To address this, we introduce a Katz index–based entity informativeness score ($\mathcal{IS}_{\text{katz}}$) (Katz, 1953), formulated as follows:

$$\mathcal{IS}_{\text{katz}}(e_i, \mathcal{M}) = \frac{1}{|\mathcal{M}|} \sum_{e_m \in \mathcal{M}} \sum_{k=1}^{K} \beta^k (\mathbf{A}^k)_{i,m}, \tag{6}$$

where $\mathbf{A}$ is the adjacency matrix of graph $\mathcal{G}$, K denotes the maximum length of knowledge paths and β^k means a weight of knowledge path of length k. Since the term $\mathbf{A}^k$ represents the number of paths between entity e_i and e_m, this Katz index–based entity informativeness score enables multi-hop relationship modeling, in contrast to the simple connection-based metrics.

3.3 Response Generation

After retrieving the subgraphs, we generate a response based on both the dialog history and the retrieved subgraphs. To incorporate the retrieved knowledge subgraph $\hat{\mathcal{G}}$, we first apply the knowledge graph linearization to convert $\hat{\mathcal{G}}$ into a sequence of tokens, $\boldsymbol{z}_{\hat{\mathcal{G}}}$. This linearized subgraph is then concatenated with the dialog history $\boldsymbol{x}$, forming the input sequence for the dialog generation model as

$$\hat{\boldsymbol{x}} = \left[\boldsymbol{z}_{\hat{\mathcal{G}}}; \boldsymbol{x}\right], \tag{7}$$

where $[;]$ denotes concatenation operation. The combined sequence is fed into the response generation model to get the final response $\boldsymbol{y}$. By augmenting the dialog input with the knowledge graph, this method ensures that the generated response is both contextually relevant and knowledge-grounded.

3.4 Training DialogGSR

Our DialogGSR is trained in a multi-stage process. The training process consists of: (1) self-supervision through knowledge graph reconstruction, (2) training the generative subgraph retriever, and (3) optimizing the response generation model. These stages work in synergy to ensure the model effectively retrieves knowledge from graphs and generates coherent, knowledge-grounded responses. We also train the response generator by minimizing response generation loss.

Knowledge graph reconstruction. Inspired by masked language modeling techniques (Roberts et al., 2020; Devlin et al., 2019), we propose a self-supervised learning approach to learn the special tokens by masking either an entity token or a relation token in the token sequence of each knowledge path and reconstructing it. Specifically, we first sample k-hop path $\mathcal{G}'$ from the knowledge source graph $\mathcal{G}$ and convert it into token sequence $z_{\mathcal{G}'}$. During training, we randomly mask out either an entity token or a relation token from the sequence. The loss is formulated as

$$\mathcal{L}_{\text{GraphRecon}} = -\log p(\boldsymbol{z}_{\mathcal{G}'}|\hat{\boldsymbol{z}}_{\mathcal{G}'}), \tag{8}$$

where $\boldsymbol{z}_{\mathcal{G}'}$ is the token sequence of a sampled path and $\hat{\boldsymbol{z}}_{\mathcal{G}'}$ is its randomly masked sequence. For example, a knowledge triplet $\boldsymbol{z}_p = \langle$ 'Scarlet Letter', 'written by', 'N.Hawthorne' $\rangle$ can be randomly masked as

⟨`<M>, 'written by', 'N.Hawthorne'`⟩
⟨`'Scarlet Letter', <M>, 'N.Hawthorne'`⟩
⟨`'Scarlet Letter', 'written by', <M>`⟩.

Note that masking is done at the entity or relation level as done in Roberts et al. (2020). By minimizing the graph reconstruction loss, our framework self-supervise the special tokens `[Head]`,`[Int]`,`[Rev]`,`[Tail]` in (2), resulting in better knowledge graph representations. All the other parameters are frozen during this stage.

Knowledge subgraph retrieval. We train our generative subgraph retriever (GSR) to identify relevant subgraphs for dialog generation. Unlike conventional retrieval methods, our approach frames retrieval as a generation task, enabling a more seamless integration with the dialogue context. The loss is defined as follows:

$$\mathcal{L}_{\text{Ret}} = \mathbb{E}_{\boldsymbol{x}}\left[-\log p\left(\mathcal{G}^{\star}|\boldsymbol{x}\right)\right] \tag{9}$$

where $\mathcal{G}^{\star}$ is the gold subgraph and $\boldsymbol{x}$ is the dialog context We use cross-entropy loss to train the retriever, ensuring it generates subgraphs that are both relevant and informative.

Response generation. The final stage of training DialogGSR is response generation. We generate dialog responses with dialog history $\boldsymbol{x}$ and context-relevant knowledge subgraphs $\hat{\mathcal{G}}$ retrieved from GSR. The response generation loss is defined as follows:

$$\mathcal{L}_{\text{Gen}} = \mathbb{E}_{\boldsymbol{x}}\left[-\log p\left(\boldsymbol{y}^{\star}|\boldsymbol{x},\hat{\mathcal{G}}\right)\right], \tag{10}$$

where $\boldsymbol{y}^{\star}$ is the golden response.

4 Experiments

In this section, we evaluate the effectiveness of the proposed DialogGSR on knowledge graph–grounded dialog generation. We first introduce the two datasets (OpenDialKG (Moon et al., 2019) and KOMODIS (Galetzka et al., 2020)), and the experimental setup and metrics. Then, we demonstrate the effectiveness of DialogGSR on the two benchmark datasets. Lastly, we provide ablation studies, and analyses of our DialogGSR.

4.1 Datasets

OpenDialKG is an open-domain dialog dataset, which consists of 15K dialogs with 91K turns and 1.12M triplets from Freebase knowledge graph (Bast et al., 2014). The knowledge graph has 1,190,658 triplets, 100,813 entities, and 1,358 relations. There are 49% of the turns having gold knowledge triplets. Following (Galetzka et al., 2020), we randomly split the samples into train (70%), validation (15%), and test (15%) sets. We evaluate the response generation and retrieval performance of our DialogGSR with other baselines using OpenDialKG dataset.

KOMODIS is a closed-domain dialog dataset that consists of 7.5k dialogs with 103k turns and the corresponding KG, which contains 88K triplets. Following (Moon et al., 2019; Kang et al., 2023; Galetzka et al., 2020), we randomly split the dialogs into train (70%), validation (15%), and test (15%) sets for KOMODIS dataset, too. With KOMODIS dataset, we evaluate the response generation performance of our DialogGSR with other baselines following (Kang et al., 2023; Galetzka et al., 2021).

4.2 Experimental Setup

For fair comparisons with previous works, we use T5-small (Roberts et al., 2020) as the base PLM. We select the best model on the validation set to evaluate the performance of all experiments. More details are in Appendix B.

Evaluation metrics. We evaluate the dialog generation performance of different models with BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and unigram F1 score, by comparing the generated responses with the gold responses. In addition, we use the KQA metric (Kang et al., 2023), which measures whether the factually correct and necessary knowledge is contained in the generated response given the dialog history. We also evaluate

Method	BLEU				ROUGE			Unigram	KQA	
	B-1	B-2	B-3	B-4	R-1	R-2	R-L	F1	EM	F1
T5 (w/o KG)	15.79	9.19	5.61	3.43	19.67	7.13	19.02	22.21	12.25	20.69
Space Efficient (series)	16.15	10.03	6.66	4.50	21.15	8.56	20.44	24.55	36.60	42.64
Space Efficient (parallel)	16.33	10.22	6.81	4.64	21.42	8.85	20.68	24.87	38.54	44.34
EARL	11.49	6.34	4.06	2.75	15.36	4.37	14.61	16.88	32.47	35.88
DiffKG	15.68	9.13	5.60	3.46	19.50	7.07	18.84	22.26	12.25	20.99
SURGE (unsup.)	17.77	11.30	7.69	5.36	21.64	9.14	20.75	25.24	48.49	55.77
SURGE (semi-sup.)	17.70	11.21	7.61	5.28	21.43	8.85	20.57	25.07	51.00	57.63
SURGE (contrastive)	17.29	11.04	7.54	5.28	21.35	8.98	20.48	25.10	50.45	57.70
DialogGSR (Ours)	**19.30**	**12.10**	**8.30**	**5.83**	**22.32**	**9.24**	**21.23**	**25.50**	**54.61**	**60.57**

Table 1: Response generation performance comparison on OpenDialKG dataset.

Method	BLEU	ROUGE	F1
T5 (w/o KG)	7.58	18.54	16.60
Space Efficient (series)	8.34	22.36	17.37
Space Efficient (parallel)	9.33	22.80	17.72
SURGE (unsup.)	11.46	23.49	18.70
SURGE (semi-sup.)	11.28	23.58	18.68
SURGE (contrastive)	11.51	24.13	19.51
DialogGSR (Ours)	**11.96**	**24.47**	**19.60**

Table 2: Experimental results on KOMODIS dataset.

Method	path@1	path@3
Seq2Seq	3.1	18.3
Tri-LSTM	3.2	14.2
EXT-ED	1.9	5.8
DialKG Walker	13.2	26.1
AttnFlow	17.37	24.84
AttnIO	23.72	37.53
DiffKG	26.12	44.50
SURGE	16.76	28.64
DialogGSR (Ours)	**28.96**	**46.76**

Table 3: Retrieval performance on OpenDialKG.

the performance of the retriever with path@k metrics, which are the recall@k of ground-truth paths following (Moon et al., 2019; Jung et al., 2020).

4.3 Experimental Results

We compare our DialogGSR with existing knowledge–grounded dialog generation models on OpenDialKG dataset. Table 1 shows that DialogGSR achieves the best performance in all metrics (BLEU, ROUGE, KQA, and F1 score). In particular, DialogGSR outperforms other baselines on KQA metrics by a large margin (4.61 on EM metric), which indicates that the proposed method generates more factually correct responses with relevant knowledge. In addition, our method achieves a 1.53 performance gain on BLEU-1 metric compared to the best baseline method, which is an 8.61% improvement. The performance gain of DialogGSR compared to SURGE, which retrieves the subgraph with a bi-encoder and uses graph neural networks for graph representations, indicates that our generative retrieval is effective in retrieving relevant knowledge and generating more accurate responses based on the retrieved knowledge.

Method	**DialogGSR (Ours)**	SURGE
Consistency	**2.57 (0.168)**	2.41 (0.196)
Informativeness	**2.28 (0.136)**	1.81 (0.260)
Fluency	**2.64 (0.200)**	2.53 (0.286)

Table 4: Human evaluation results. () indicates standard deviation.

We also conduct experiments on KOMODIS (Galetzka et al., 2020) dataset. Similar to the OpenDialKG result, Table 2 demonstrates that our DialogGSR achieves the best performance compared to all the previous approaches. To further validate the effectiveness of our generative subgraph retrieval, we compare the retrieval performance by path@k metrics. Table 3 shows that DialogGSR achieves the best performance compared to the other baselines. This result indicates that our generative subgraph retrieval successfully retrieves context-relevant subgraphs from the knowledge graph by fully utilizing the power of pretrained language models.

4.4 Human Evaluation

We conduct human evaluation to assess the generated responses of our dialog generation model. The detailed process of human evaluation is in Appendix C. Table 4 shows the experimental results of the human evaluation, where DialogGSR outperforms SURGE in all the metrics (Consistency, Informativeness, Fluency). In particular, on the Consistency and Informativeness metrics, DialogGSR achieves statistically significant performance

Graph Const.	Special tokens	B-1	B-2	B-3	B-4	path@3
w/o Const.	with Special tokens (w/o Recon.)	17.02	10.96	7.53	5.25	10.00
Hard Const.	w/o Special tokens	18.44	11.68	7.93	5.44	35.83
Hard Const.	with Special tokens (w/o Recon.)	18.77	11.74	8.03	5.48	39.53
Hard Const.	with Special tokens (with Recon.)	18.83	11.84	8.01	5.49	43.27
Connection Const.	with Special tokens (with Recon.)	19.17	11.90	8.15	5.68	45.85
Katz Const.	with Special tokens (with Recon.)	**19.30**	**12.10**	**8.30**	**5.83**	**46.76**

Table 5: Ablation study of each component in DialogGSR on OpenDialKG dataset.

Method	B-1	B-2
Base (w/o KG)	18.68	11.96
DialogGSR (w/o Const.)	19.60	13.32
DialogGSR (ours)	**21.10**	**14.44**

Table 6: Experimental results on OpenDialKG dataset with large language model `Llama-3-8b` under the fine-tuning with LoRA (Hu et al., 2022). 'Const.' denotes graph-constrained decoding.

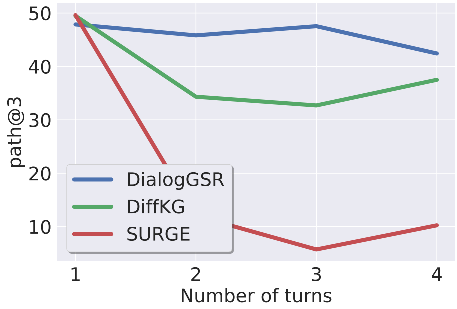

Figure 2: Retrieval performance according to the number of turns.

gains of 0.16 and 0.47 over SURGE (based on t-test with p-value < 0.05), which indicates that our generative subgraph retrieval performs significantly better in retrieving informative knowledge compared to existing retrieval methods. Our DialogGSR provides a relatively small performance gain of 0.11 on the Fluency metric. Since the Fluency metric is more influenced by the language model's performance than the knowledge retrieval performance, it is reasonable to expect similar fluency scores when using the same base language model (T5-small) for fair comparisons.

4.5 Analysis

We analyze DialogGSR to answer the following research questions: **[Q1]** Does each component of DialogGSR contribute to a performance improvement? **[Q2]** Are graph-constrained decoding and the entity informativeness score helpful for retrieving context-relevant subgraphs? **[Q3]** Is GSR robust to the information bottleneck issue? **[Q4]** Is DialogGSR effective with large language models (LLMs)?

Ablation studies. We provide the ablation studies to answer **[Q1], [Q2]** by empirically showing the contribution of each component of DialogGSR in Table 5. **w/o Const.** is generative retrieval without graph-constrained decoding. **Hard const.** is the retrieval with graph-constrained decoding but not considering entity informativeness score. **Connection** and **Katz** use entity informativeness scores based on Connection ($\mathcal{IS}_{\text{con}}$) and Katz metrics ($\mathcal{IS}_{\text{Katz}}$) referred in Section 3.4, respectively. **with Special tokens (w/o Recon.)** uses special tokens to linearize the knowledge graph without graph reconstruction learning while **with Special tokens (w/ Recon.)** uses prompts learned with graph reconstruction. Table 5 shows that each component contributes to the performance improvement of the model. In particular, graph-constrained decoding is crucial in our generative approach.

In addition, the models with graph constraints show improvements compared to the model without the constraints, which indicates that the graph constraint is important for the generative retrieval of knowledge subgraphs. Also, using entity informative score (Connection, Katz) performs better than graph constraints without it since the entity informativeness score reflects graph structural proximity in the decoding process.

Effectiveness of DialogGSR with LLMs. To assess the effectiveness of our DialogGSR with Large Language Models (LLMs) (**[Q4]**), we apply it to LLaMA-3 (Meta, 2024). The experimental result is shown in Table 6. From the table, the performance gain of DialogGSR compared to the base model is 2.42 in BLEU-1 score. In addition, the experimental result demonstrates that our proposed graph-constrained decoding is still important in LLMs. This indicates that DialogGSR is also effective in LLMs.

Dialog	Gold response	SURGE (Baseline)	DialogGSR (Ours)
(a) Do you like Shaun White? (b) I know he's an Olympic snowboarder he was funny in Friends With Benefits. (a) Oh, I've never seen that movie, isn't Mila Kunis in it? I love her! (b) She is. Justin Timberlake and Woody Harrelson were also in it. Shaun just played a small part. (a) Do you by any chance remember who Mila Kunis is married too, I totally forgot.	She's married to Ashton Kutcher.	Mila Kunis is married to Jennifer Lawrence.	Mila Kunis is married to Ashton Kutcher.
Knowledge triplets τ retrieved by Baseline ⟨ 'Justin Timberlake', 'place musical career began', 'Shelby Forest' ⟩ ⟨ 'Justin Timberlake', 'place musical career began', 'Millington' ⟩ ⟨ 'Justin Timberlake', 'romantic relationship (with celebrities)', 'Scarlett Johansson'⟩			
Knowledge triplets τ retrieved by DialogGSR (ours) ⟨ 'Ashton Kutcher', 'romantic relationship (with celebrities)', 'Mila Kunis' ⟩ ⟨ 'Friends with Benefits', 'starred_actors', 'Mila Kunis' ⟩ ⟨ 'Friends with Benefits', 'starred_actors', 'Patricia Clarkson' ⟩			

Table 7: Comparison on responses generated by SURGE (Baseline) and DialogGSR given a dialog.

Information bottleneck issue. Information bottleneck issue (Humeau et al., 2020; Lee et al., 2022) usually occurs when a long text sequence, such as a dialog history, is encoded into a single fixed length of vector. To explore the robustness of DialogGSR to the information bottleneck issue (**[Q3]**), we compare the retrieval performance of DialogGSR with the baselines such as DiffKG and SURGE with respect to the number of turns in dialog histories in Figure 2. The result shows that DialogGSR is robust for long dialogs whereas the other methods often deteriorate as the number of turns increases.

Qualitative analysis. We perform qualitative analysis by comparing responses generated from SURGE and DialogGSR. Table 7 shows a sampled **Gold response** and the responses generated by SURGE (**Baseline response**) and DialogGSR (**DialogGSR response**) given a multi-turn dialog. From the table, DialogGSR retrieves more informative knowledge to generate responses compared to the baseline. Given the last turn "Do you by any chance remember who Mila Kunis is married too, I totally forgot", DialogGSR successfully retrieves the knowledge information related to 'Mila Kunis' to help provide the appropriate response from the question while the baseline fails to retrieve information related to answer the question. In contrast, the baseline incorrectly retrieves knowledge information related to "Justin Timberlake", who is mentioned in the past turn (4th turn), which results in a factually incorrect response. This demonstrates that generative retrieval is effective in retrieving informative knowledge and generating knowledge-grounded multi-turn dialogs. More qualitative results are included in Appendix A.2.

5 Conclusion

We have presented DialogGSR, a dialog generation model with generative subgraph retrieval. DialogGSR retrieves context-relevant subgraphs, by generating the subgraph token sequences considering both the dialog context and the graph information. We have proposed novel knowledge graph linearization to convert knowledge triplets into token sequences with self-supervised graph-specific tokens to represent knowledge graphs without separate knowledge graph modules. In addition, we have formulated a graph-constrained decoding for valid and relevant generative retrieval. Our experiments demonstrate the effectiveness of our proposed method in knowledge–graph grounded dialog generation. Our codes are publicly available at `https://github.com/mlvlab/DialogGSR`.

Limitations

The proposed DialogGSR generatively retrieves token sequences of the subgraph from a knowledge graph and then generates a response with the retrieved subgraph. However, similar to works using graph retrieval on knowledge-grounded dialog generation, our generative subgraph retrieval can retrieve only the knowledge information contained in the knowledge graph. Second, the benchmark datasets for knowledge graph–grounded dialog generation are limited. Therefore, new benchmark datasets on dialog generation with knowledge graphs warrants greater attention.

Ethics Statement

Our DialogGSR does not have any direct negative social impacts, but it can potentially be used maliciously, similar to other dialog generation models. These models may produce factually incorrect or biased responses, particularly in sensitive areas such as politics, religion, and diplomacy. To address these risks, we advocate for the release of benchmark datasets without private information and emphasize the need for research into the methods that detect the source of texts. These measures are essential for the responsible development and use of dialog generation technologies.

Acknowledgements

This work (for authors: HJK, JP, and MJ) was partly supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2024-RS-2020-II201819, 10 %), and National Research Foundation of Korea grant (NRF-2023R1A2C2005373, 90 %).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv:2303.08774*.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural NLG from compositional representations in task-oriented dialogue. In *ACL*, pages 832–844.

Hannah Bast, Florian Bäurle, Björn Buchhold, and Elmar Haußmann. 2014. Easy access to the freebase dataset. In *WWW*, pages 95–98.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. In *NeurIPS*, pages 31668–31683.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*, pages 2787–2795.

Sergey Brin. 1998. The pagerank citation ranking: bringing order to the web. *Proceedings of ASIS, 1998*, 98:161–172.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *ICLR*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *ACL*, pages 1870–1879.

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022a. Gere: Generative evidence retrieval for fact verification. In *SIGIR*, pages 2184–2189.

Xiang Chen, Zhixian Yang, and Xiaojun Wan. 2022b. Relation-constrained decoding for text generation. In *NeurIPS*, pages 26804–26819.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *ICLR*.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *INLG*, pages 322–328.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG challenge. *Computer Speech & Language*, 59:123–156.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *EMNLP*, pages 1295–1309.

Patrick Fernandes, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Structured neural summarization. In *ICLR*.

Fabian Galetzka, Chukwuemeka U Eneh, and David Schlangen. 2020. A corpus of controlled opinionated and knowledgeable movie discussions for training neural conversation models. In *LREC*, pages 565–573.

Fabian Galetzka, Jewgeni Rose, David Schlangen, and Jens Lehmann. 2021. Space efficient context encoding for non-task-oriented dialogue generation with graph attention transformer. In *ACL-IJCNLP*, pages 7028–7041.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. 2019. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*.

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 *AAAI*, pages 5110–5117.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*.

Luyang Huang, Lingfei Wu, and Lu Wang. 2020. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *ACL*, pages 5094–5107.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *ICLR*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *EACL*, pages 874–880.

Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Sebastian Riedel, and Edouard Grave. 2020. A memory efficient baseline for open domain question answering. *arXiv:2012.15156*.

Ziwei Ji, Zihan Liu, Nayeon Lee, Tiezheng Yu, Bryan Wilie, Min Zeng, and Pascale Fung. 2023. RHO: reducing hallucination in open-domain dialogues with knowledge grounding. In *ACL-findings*, pages 4504–4522.

Jaehun Jung, Bokyung Son, and Sungwon Lyu. 2020. Attnio: Knowledge graph exploration with in-and-out attention flow for knowledge-grounded dialogue. In *EMNLP*, pages 3484–3497.

Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2023. Knowledge graph-augmented language models for knowledge-grounded dialogue generation. *arXiv:2305.18846*.

Leo Katz. 1953. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43.

Hyunji Lee, Jaeyoung Kim, Hoyeon Chang, Hanseok Oh, Sohee Yang, Vlad Karpukhin, Yi Lu, and Minjoon Seo. 2023. Nonparametric decoding for generative retrieval. In *ACL-findings*, pages 12642–12661.

Hyunji Lee, Sohee Yang, Hanseok Oh, and Minjoon Seo. 2022. Generative multi-hop retrieval. In *EMNLP*, pages 1417–1436.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*, pages 9459–9474.

Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to select knowledge for response generation in dialog systems. In *IJCAI*, pages 5081–5087.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *TACL*, 9:329–345.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *ICLR*.

AI Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *ACL*, pages 845–854.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Prasanna Parthasarathi and Joelle Pineau. 2018. Extending neural generative conversational model using external knowledge sources. In *EMNLP*, pages 690–695.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035.

Chengwen Qi, Bowen Li, Binyuan Hui, Bailin Wang, Jinyang Li, Jinwang Wu, and Yuanjun Laili. 2023. An investigation of llms' inefficacy in understanding converse relations. In *EMNLP*, pages 6932–6953.

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.

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

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *EMNLP-findings*, pages 3784–3803.

Weiwei Sun, Pengjie Ren, and Zhaochun Ren. 2023. Generative knowledge selection for knowledge-grounded dialogues. In *EACL-findings*, pages 2032–2043.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*, pages 3104–3112.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. LaMDA: Language models for dialog applications. *arXiv:2201.08239*.

James Thorne. 2022. Data-efficient auto-regressive document retrieval for fact verification. In *SustaiNLP*, pages 44–51.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL*, pages 809–819.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*.

Yi-Lin Tuan, Sajjad Beygi, Maryam Fazel-Zarandi, Qiaozi Gao, Alessandra Cervone, and William Yang Wang. 2022. Towards large-scale interpretable knowledge graph reasoning for dialogue systems. In *ACL-findings*, pages 383–395.

Yi-Lin Tuan, Yun-Nung Chen, and Hung-yi Lee. 2019. DyKgChat: Benchmarking dialogue generation grounding on dynamic knowledge graphs. In *EMNLP*, pages 1855–1865.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

Jian Wang, Junhao Liu, Wei Bi, Xiaojiang Liu, Kejing He, Ruifeng Xu, and Min Yang. 2020. Improving knowledge-aware dialogue generation via knowledge base question answering. In *AAAI*, pages 9169–9176.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. 2022. A neural corpus indexer for document retrieval. In *NeurIPS*, pages 25600–25614.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv:1910.03771*.

Yi Xu, Shuqian Sheng, Jiexing Qi, Luoyi Fu, Zhouhan Lin, Xinbing Wang, and Chenghu Zhou. 2023. Unsupervised graph-text mutual conversion with a unified pretrained language model. In *ACL*, pages 5130–5144.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *NAACL-HLT*, pages 535–546.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *AAAI*, pages 4970–4977.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2022. Jaket: Joint pre-training of knowledge graph and language understanding. In *AAAI*, pages 11630–11638.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *ICLR*.

Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *ACL*, pages 2031–2043.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022a. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *ACL*, pages 5773–5784.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022b. GreaseLM: Graph REASoning enhanced language models. In *ICLR*.

Xueliang Zhao, Wei Wu, Chongyang Tao, Can Xu, Dongyan Zhao, and Rui Yan. 2020. Low-resource knowledge-grounded dialogue generation. In *ICLR*.

Hao Zhou, Minlie Huang, Yong Liu, Wei Chen, and Xiaoyan Zhu. 2021. EARL: Informative knowledge-grounded conversation generation with entity-agnostic representation learning. In *EMNLP*, pages 2383–2395.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2024. DyVal: Graph-informed dynamic evaluation of large language models. In *ICLR*.

Qi Zhu, Carl Yang, Yidan Xu, Haonan Wang, Chao Zhang, and Jiawei Han. 2021. Transfer learning of graph neural networks with ego-graph information maximization. In *NeurIPS*, pages 1766–1779.

Reverse	Multiple	B-1	B-2	path@3
		18.74	11.99	41.28
✓		19.07	12.03	44.54
	✓	19.22	12.01	45.06
✓	✓	**19.30**	**12.10**	**46.76**

Table 8: Ablation studies on special tokens with OpenDialKG dataset. ‘Reverse’ denotes reverse tokens and ‘Multiple’ denotes multiple tokens.

A Additional experiments

A.1 Additional Quantitative Analysis

We also conduct experiments to verify the contribution of using [Rev] to represent reverse relations and multiple consecutive tokens to represent each [Rev] or [Int] in Table 8. By adding **reverse tokens** to the knowledge, which allows mentioned entities that are tail entities in the provided triplets to be the starting points for the decoding, the performance is improved by 0.33 on BLEU-1 metric. Also, using **multiple consecutive tokens** to represent each [Rev] or [Int] (e.g., [Head] e_1 [Int$_{11}$] [Int$_{12}$] r_1 [Int$_{21}$] [Int$_{22}$] e_2 [Tail]) gives the performance gain on all the metrics since using the multiple tokens improve the capacity of representing the entities and the relations on top of language models. By adding all the components, performance significantly improves by 0.56 on BLEU-1 metric compared to the linearized knowledge graph without any special tokens, which demonstrates the effectiveness of our proposed knowledge graph linearization approaches with special tokens. Interestingly, adding reverse tokens with using multiple consecutive tokens improves the overall performance compared to adding reverse tokens without using multiple consecutive tokens, which indicates that representing reverse relations is more effective when the capacity of the knowledge representation is increased.

A.2 Additional Qualitative Analysis

In Table 9, we provide additional qualitative examples for what we have shown in Table 7 of the main paper. Our DialogGSR often generates high-quality responses similar to the main paper. For example, in the first example, our DialogGSR generates a factually correct response "It was written by Frank Beddor" based on the retrieved triplet ⟨‘The Looking Glass Wars’, ‘written_by’, ‘Frank Beddor’⟩ while SURGE generates a factually incorrect response "Terry Pratchett" with the same triplet ⟨‘The Looking Glass Wars’, ‘written_by’, ‘Frank Beddor’⟩. It demonstrates that our DialogGSR is more effective in generating responses even with the same knowledge information given. In the second example, DialogGSR successfully generates a factually correct response by retrieving context-relevant knowledge triplets whereas the factually incorrect response is generated by the baseline due to the retrieval of irrelevant knowledge. These results demonstrate that our generative retrieval is effective in retrieving informative knowledge and generating knowledge-grounded dialogs.

B Experimental details

B.1 Implementation details

In this section, we describe the implementation details not included in our main paper. For all the experiments, we use PyTorch[1] (Paszke et al., 2019) and Transformer module of Huggingface[2] (Wolf et al., 2019) as our code base. All experiments are conducted with 48GB NVIDIA RTX A6000 GPU. We select the best model on the validation set to evaluate the performance of all experiments. The epoch for training is set to 50 and the weight decay is 0.1. We use AdamW optimizer (Loshchilov and Hutter, 2019) to train our model and adopt learning rate decay.

Knowledge graph–constrained decoding. Without the graph constraints, the language model is prone to generate invalid or irrelevant subgraphs due to the language model's bias (Chen et al., 2022b; Cao et al., 2021). To inject the knowledge graph information into the language model in the decoding step, we present a knowledge graph–constrained decoding method. We use $\alpha = 0.8$ and $k = 2$ for calculating Katz (Katz, 1953) index-based entity informativeness score. p_{graph} is defined in Eq. (6) of the main paper, and b is 5.

C Details of Human Evaluation

We first randomly selected 30 dialogs from OpenDialKG test dataset (Moon et al., 2019) and generated responses using our model and SURGE (Kang et al., 2023) for the comparison. We recruited 22 participants who were not involved in our research and allowed the use of external sources, such as the

[1] Copyright (c) 2016-Facebook, Inc (Adam Paszke), Licensed under BSD-style license

[2] Copyright 2018-The Hugging Face team, Licensed under the Apache license

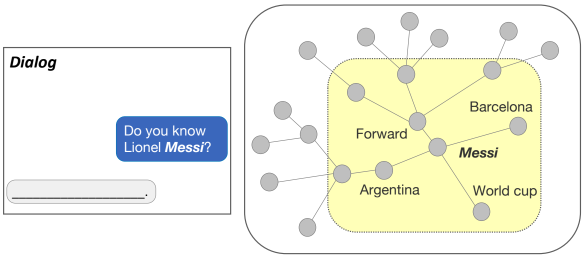

Figure 3: An example of extracting a 2-hop candidate subgraph from the knowledge graph. Yellow region indicates the 2-hop candidate subgraph centered on the mentioned entity "Messi".

Internet, to verify the factual correctness of generated responses. Following the process outlined in the other work (Kang et al., 2023), we utilized a 3-point Likert-like scale to evaluate three criteria: Consistency, Informativeness, and Fluency. **Consistency** measures the coherence and logical flow within the context of the conversation, **Informativeness** assesses the correctness and usefulness of the information in the generated responses, and **Fluency** focuses on the naturalness and linguistic quality of the dialog. With the human evaluation metrics and the automatic metrics in the main paper, we establish a comprehensive evaluation framework that enables accurate comparisons between models, enhancing the reliability of our assessment.

D Baselines

D.1 Response Generation

In our experiments, the following baseline models are used for comparing the response generation performance with our DialogGSR.

- **T5–small (w/o KGs)**[3] (Roberts et al., 2020): T5-small is an encoder-decoder Transformer architecture for various natural language processing tasks.

- **Space Efficient (series)**[4] (Galetzka et al., 2021): Space Efficient (series) is the model proposed in (Galetzka et al., 2021). It utilizes all knowledge triplets related to the entities by matching the entities of KG and the entities mentioned in dialog history without any retrieval process. This model sequentially encodes knowledge triplets and feeds them into the encoder.

- **Space Efficient (parallel)** (Galetzka et al., 2021) : This model is also proposed by (Galetzka et al., 2021). Different from Space Efficient (series), this model constructs a segmentation block for each entity and encodes the relation in the segmentation block to reflect relational information.

- **Diff-KG** (Tuan et al., 2022): Diff-KG reasons differentiable knowledge paths to jointly generate a response with the dialog history. After the path reasoning, entities included in the path are concatenated with dialog history, and they are fed into a pretrained language model.

- **SURGE (unsup.)** (Kang et al., 2023): SURGE is a graph neural network–augmented Transformer-based dialog generation model that encodes knowledge triplets with graph neural networks. SURGE also retrieves context-relevant triplets via a subgraph retriever. This model trains the retriever without the guidance of gold knowledge and is implicitly trained with response generation loss.

- **SURGE (semi-sup.)** (Kang et al., 2023): SURGE (semi-sup.) uses gold knowledge to train the retriever.

- **SURGE (contrastive)** (Kang et al., 2023): SURGE (contrastive) uses both the retrieval supervision from SURGE (Semi-sup.) and contrastive learning to encourage the encoder output and the decoder output to be closer.

D.2 Knowledge Retrieval

The models below are used as the baselines for validating the effectiveness of our DialogGSR on knowledge subgraph retrieval.

- **Seq2Seq** (Sutskever et al., 2014): Seq2Seq is used as a baseline in (Moon et al., 2019; Tuan et al., 2022). Given all of the dialog contexts, Seq2Seq generates entity paths.

- **Tri-LSTM** (Young et al., 2018): Tri-LSTM is another baseline in (Moon et al., 2019; Tuan et al., 2022). It encodes dialog contexts and related 1-hop knowledge from a KG to retrieve knowledge paths.

- **Ext-ED (Extended Encoder-Decoder)** (Parthasarathi and Pineau, 2018): Extended Encoder-Decoder is also

[3] Licensed under the Apache license

[4] Copyright (c) 2021 Fabian Galetzka, Licensed under MIT license

one of the baselines in (Moon et al., 2019; Tuan et al., 2022). It generates a response conditioned on an external knowledge vector input, which is encoded by GloVe embedding.

- **DialKG Walker** (Moon et al., 2019): DialKG Walker is an attention-based knowledge path retrieval model designed to traverse a knowledge graph with dialog context and knowledge paths.
- **AttnFlow** (Jung et al., 2020): AttnFlow is an attention-based knowledge path retrieval model based on GAT (Veličković et al., 2018) and the encoded dialog context. It only uses incoming attention flow to update knowledge representation.
- **AttnIO** (Jung et al., 2020): AttnIO is an extension of AttnFlow, where both incoming and outcoming attention flows are used to represent knowledge paths with dialog contexts and entity features.

Dialog	Gold response	SURGE (Baseline)	DialogGSR (Ours)
(a) Could you recommend and books by the author of Colour of Magic? (b) The Colour of Magic has genre fantasy. So do you want to read fantasy books? (a) Like Through the Looking Glass? Sure I like Fantasy okay. (b) yes like The Looking Glass Wars it's really a good book. I suggest reading it. (a) Do you know who wrote it by any chance?	Yes Frank Beddor wrote it, who also wrote Seeing Redd.	Terry Pratchett	It was written by Frank Beddor.
Knowledge triplets τ retrieved from Baseline ⟨ 'The Colour of Magic', 'written_by', 'Terry Pratchett' ⟩ ⟨ 'The Looking Glass Wars', 'written_by', 'Frank Beddor' ⟩			
Knowledge triplets τ retrieved from DialogGSR ⟨ 'The Looking Glass Wars', 'written_by', 'Frank Beddor' ⟩ ⟨ 'Frank Beddor', 'is-a', 'Film Producer' ⟩			
(a) I like the book Where'd You Go, Bernadette. Do you have any other suggestions for me? (b) Definitely! That's a great book by Maria Semple. Do you like her? (a) I do! Has she written anything else?	She is a screenwriter, television producer, and she produced Mad About You.	She's written a lot of books, including Where'd You Go, Bernadette. Have you read that one?	She has. She also wrote the TV program, Mad About You. Have you heard of that one?
Knowledge triplets τ retrieved from Baseline ⟨ 'Where'd You Go, Bernadette', 'written_by', 'Maria Semple' ⟩ ⟨ 'Where'd You Go, Bernadette', 'release_year', '2012' ⟩ ⟨ '2012', 'release_year (reverse)', 'Where'd You Go, Bernadette' ⟩			
Knowledge triplets τ retrieved from DialogGSR ⟨ 'Where'd You Go, Bernadette', 'written_by' ,'Maria Semple' ⟩ ⟨ 'Maria Semple', 'tv program produced', 'Mad About You' ⟩ ⟨ 'Mad About You', 'has_genre', 'sitcom' ⟩			
(a) Do you know any movies directed by Bennett Miller? (b) He has some great ones. Have you seen Moneyball or Capote? (a) I haven't seen Moneyball, who stars in it?	Steve Zaillian wrote Moneyball. It starred Brad Pitt along with Tammy Blanchard. It's a really good movie!	Capote stars Seymour Hoffman and Ben Stiller. It's a romantic comedy.	Tammy BLanchard and Brad Pitt are in it. Do you like action movies?
Knowledge triplets τ retrieved from Baseline ⟨ 'Moneyball', 'starred_actors', 'Philip Seymour Hoffman' ⟩ ⟨ 'Capote', 'starred_actors','Philip Seymour Hoffman' ⟩ ⟨ 'Philip Seymour Hoffman', 'starred_actors (reverse)', 'Moneyball' ⟩			
Knowledge triplets τ retrieved from DialogGSR ⟨ 'Moneyball', 'starred_actors', 'Tammy Blanchard' ⟩ ⟨ 'Moneyball', 'starred_actors', 'Brad Pitt' ⟩ ⟨ 'Moneyball', 'starred_actors', 'Robin Wright' ⟩			
(a) Can you tell me some Beatrix Potter books? (b) Would you prefer her biography, or children's literature? (a) Children's literature please. (b) Would you like The Tale of Peter Rabbit? Or The Tale of Mrs Tiggy-Winkle? I hear they're both good. (a) Ill try Peter Rabbit, out of curiosity when were these books written?	I believe they were written in the year 1905, to be exact.	The Tale of Mrs. Tiggy-Winkle was written in 1851.	I'm not sure when they were written, but they were released in the year 1905.
Knowledge triplets τ retrieved from Baseline ⟨ 'The Complete Tales of Beatrix Potter', 'written_by', 'Beatrix Potter ' ⟩ ⟨ 'The Complete Adventures of Peter Rabbit', 'written_by', 'Beatrix Potter ' ⟩ ⟨ 'The Tale of Mrs. Tiggy-Winkle', 'written_by', 'Beatrix Potter ' ⟩			
Knowledge triplets τ retrieved from DialogGSR ⟨ 'The Tale of Mrs. Tiggy-Winkle', 'written_by', 'Beatrix Potter' ⟩ ⟨ 'The Tale of Mrs. Tiggy-Winkle', 'release_year', '1905' ⟩ ⟨ 'The Return Of Sherlock Holmes', 'release_year', '1905' ⟩			

Table 9: Comparison on responses generated by SURGE (Baseline) and DialogGSR given a dialog.