Generative Subgraph Retrieval for Knowledge Graph–Grounded Dialog Generation

Anonymous ACL submission

Abstract

Knowledge graph–grounded dialog generation requires retrieving a dialog-relevant subgraph from the given graph and then seamlessly integrating it with the dialog history. Previous works typically represent a graph using either an external encoder such as graph neural networks and retrieve relevant triplets by similarity between single-vector representations of triplets and a dialog history. However, the external encoders cannot leverage the knowledge from trained language models, and the retrieval procedures are also suboptimal since the single-vector summarization of dialog history causes an information bottleneck. In this work, we propose Dialog generation with Generative Subgraph Retrieval (DialogGSR), which retrieves relevant knowledge subgraphs by generating their token sequences on top of language models. For effective generative subgraph retrieval, we introduce two methods: structure-aware knowledge graph linearization with self-supervised graph-specific tokens and graph-constrained decoding with graph structural proximity-based entity informativeness scores for valid and relevant generative retrieval. DialogGSR demonstrates the best performance in knowledge graph–grounded dialog generation, as evaluated on OpenDialKG and KOMODIS datasets.

1 Introduction

Dialog generation models aim to generate an informative and appropriate response given a dialog. Pretrained Language Models (PLMs) have shown promising performance on the task (Roberts et al., 2020; Touvron et al., 2023; Achiam et al., 2023). However, they often generate irrelevant, factually incorrect, or hallucinating responses since the generation process relies on the language models' internal parameters (Lewis et al., 2020; Shuster et al., 2021). To address this issue, many works have explored knowledge-grounded dialog generation models, which generate responses using external knowledge (Wang et al., 2020; Zhao et al., 2020). Some works consider unstructured texts such as Wikipedia articles (Dinan et al., 2019) and internet web pages (Ghazvininejad et al., 2018) as external knowledge sources. The other line of works uses external knowledge graphs (KGs) to leverage their structural and semantic information for generating dialog responses grounded on the knowledge graphs (Moon et al., 2019; Galetzka et al., 2021; Tuan et al., 2022; Kang et al., 2023).

Recent knowledge graph–grounded dialog generation models such as (Tuan et al., 2022; Kang et al., 2023) first retrieve context-relevant subgraphs from the given knowledge graph to filter out the irrelevant information before generating a response. Many works (Tuan et al., 2022; Kang et al., 2023) encode the dialog history into a single vector and use it on another encoder (*e.g.*, bi-encoder) to retrieve relevant triplets from the knowledge graph. However, it may lead to information bottleneck since a single vector has a limited capacity to represent a long multi-turn dialog (Humeau et al., 2020; Cao et al., 2021; Lee et al., 2022). They also require separate knowledge graph embeddings or models, such as graph neural networks (GNNs), to represent the knowledge graphs (Galetzka et al., 2021; Tuan et al., 2022; Kang et al., 2023), which cannot effectively leverage the knowledge from PLMs.

Other works such as Luo et al. (2024); Xu et al. (2023) address the information bottleneck issue by applying generative retrieval (Lee et al., 2022; Sun et al., 2023). It casts retrieval as an autoregressive generation to facilitate direct interactions between contexts and knowledge paragraphs. However, since they only targets on natural language contexts, they simply linearize knowledge triplets with conventional token representations and decoding strategies, which do not properly account for given graph's structure and properties.

To address the aforementioned issues, we propose a **Dialog** Generation model with **G**enerative **S**ubgraph **R**etrieval (**DialogGSR**), consisting of generative subgraph retriever and response generator. The proposed generative subgraph retrieval uses two graph-specialized methods: a structure-aware knowledge graph linearization for representing the graph and graph-constrained decoding for effective generative subgraph retrieval. Our knowledge graph linearization approach adds a small number of specific token embeddings to consider both the structural position of knowledge entities and reverse relations between entities. By self-supervising the special tokens with knowledge graph reconstruction loss, it effectively represents the knowledge graph. The graph-constrained decoding facilitates autoregressively retrieving the knowledge considering the graph structural information, thereby generating valid and relevant knowledge subgraphs. Since DialogGSR only uses language models for both subgraph retrieval and dialog generation, it can effectively leverage pre-trained language models' knowledge for 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 (Lewis et al., 2020; Chen et al., 2017; Izacard and Grave, 2021; Thorne et al., 2018). Recent studies have demonstrated that generative retrieval models can be more effective than conventional encoder-based retrieval models (Cao et al., 2021; Lee et al., 2022, 2023; Bevilacqua et al., 2022; Wang et al., 2022). 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 use pre-trained 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 have proposed to augment the models with external knowledge sources (Moon et al., 2019; Dinan et al., 2019; Lian et al., 2019). This approach is effective for generating factually accurate results 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., 2021, 2018; Tuan et al., 2019; Zhang et al., 2020). 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)). It directly uses gold knowledge, which is practically inapplicable, thereby difficult to fairly compare with the other approaches. 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.

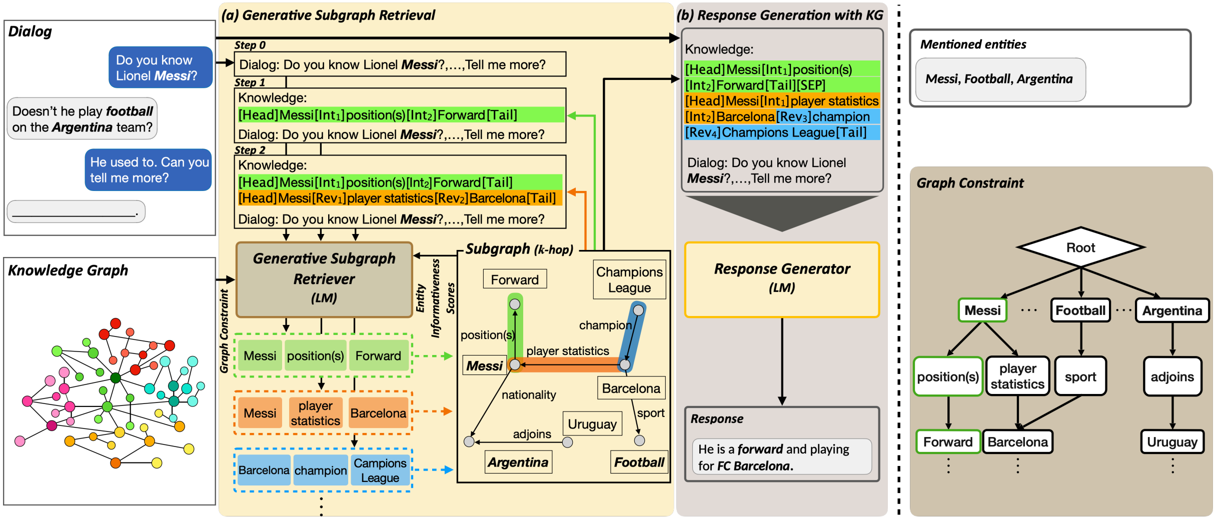
Dialog
Do you know Lionel Messi?
Doesn't he play football on the Argentina team?
He used to. Can you tell me more?
Knowledge Graph
(a) Generative Subgraph Retrieval
Step 0
Dialog: Do you know Lionel Messi?,...,Tell me more?
Step 1
Knowledge:
[Head]Messi[Int₁]position(s)[Int₂]Forward[Tail]
Dialog: Do 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
Uruguay
adjoins
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 aim to generate knowledge-grounded dialog responses with a retrieval augmented generation approach that retrieves context-relevant subgraphs from given knowledge graphs. Specifically, we propose a **Dialog** Generation model with **G**enerative **S**ubgraph **R**etrieval (DialogGSR), which consists of a generative subgraph retriever and a response generator. In this section, we first introduce the task of knowledge graph–grounded dialog generation. Next, we propose **G**enerative **S**ubgraph **R**etrieval (GSR), which generates token sequences of subgraphs through the generative subgraph retriever. For better retrieval, GSR represents the knowledge graph with structure-aware knowledge graph linearization and autoregressively retrieves the knowledge graph via graph-constrained decoding with entity-informativeness scores. Then, we present a response generator, which conducts subgraph–augmented dialog generation. Finally, we introduce the training details of DialogGSR including our self-supervised knowledge graph reconstruction loss. The overall 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, $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$, 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 extracting 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 $\boldsymbol{y} = [y_1, y_2, \ldots, y_t]$ is the output response, t is the length of the response, and $\boldsymbol{y}_{<j} = [y_1, \ldots y_{j-1}]$ denotes the generated sequence at the previous time steps. Since a KG can include a huge number of irrelevant entities and relations, extracting subgraphs related to the dialog context is crucial in KG-grounded dialog generation.

3.2 Generative Subgraph Retrieval

In this section, we describe **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 $\{\tau^{(1)}, \dots, \tau^{(k)}\}$, where k is the number of triplets consisting the subgraph, a retrieval of knowledge triplet sequences is regarded as the subgraph retrieval. Many existing subgraph retrieval methods for dialog generation (Kang et al., 2023; Zhang et al., 2022a) obtain the knowledge triplet by measuring the relevance score between the dialog history and each knowledge triplet and retrieving the triplets with high scores.

Since these methods encode the long dialog history into a single fixed length of vector, they have a limited capacity to represent multi-turn dialog. This is called the information bottleneck problem (Izacard et al., 2020; Luan et al., 2021). Due to this problem, they suffer from retrieving accurate knowledge, especially when the dialog has many turns. Also, they require independent knowledge graph encoders to represent knowledge graphs, which cannot fully leverage the knowledge available in PLMs.

To address these limitations, generative subgraph retrieval methods are used to cast the graph retrieval as graph generation providing a more direct interaction between a dialog context and a knowledge graph by representing the graph with a token sequence. For effective generative retrieval, our GSR uses the following two methods: (1) Structure-aware knowledge graph linearization that converts the knowledge graph into token sequences with learnable special tokens considering the connectivity and reverse relations between entities and (2) Graph-constrained decoding that ensures the language model to generate valid knowledge subgraphs and predicts the next tokens considering not only the language model's scores but also the proximities among the entities on the graph.

3.3 Structure-Aware Knowledge Graph Linearization

The goal of the structure-aware knowledge graph linearization is to convert the 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 model represent the knowledge graph without separate graph encoders. Different from prior graph linearization methods such as Luo et al. (2024); Xu et al. (2023) that do not take into account graph connections and reverse relations, our structure-aware knowledge graph linearization considers and more effectively represents the knowledge graph structures.

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 $\texttt{[Head]}$ e_1 $\texttt{[Int}_1\texttt{]}$ r_1 $\texttt{[Int}_2\texttt{]}$ e_2 $\texttt{[Int}_3\texttt{]}$ $r_2 \dots e_{l+1}$ $\texttt{[Tail]}$. To represent multiple disconnected triplets or paths, we insert [SEP] between them. For more expressive representations of special tokens, we use multiple consecutive tokens to represent each of [Int],[Rev], which improves the performance as described in Section A.2.

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 representing 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 $\texttt{[Rev}_1\texttt{]}$ and $\texttt{[Rev}_2\texttt{]}$ 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 $\texttt{[Head]}$ e_2 $\texttt{[Rev}_1\texttt{]}$ r_2 $\texttt{[Rev}_2\texttt{]}$ e_3 $\texttt{[Tail]}$.

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

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

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 3.6. 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.

3.4 Graph-Constrained Decoding

We introduce a graph-constrained decoding method to generate valid and relevant subgraphs. Without the graph constraints, the language model is prone to generating invalid or irrelevant subgraphs due to its bias, disregarding the graph structures (Chen et al., 2022b; Cao et al., 2021). To address this issue, our proposed graph-constrained decoding method injects the knowledge graph information into the language model in the decoding step.

Formally, given the dialog $\boldsymbol{x}$ and the output sequence $\tilde{\pi}_{<t}$ at the previous time step, the log probability of the next token w is computed with $\log p_{\text{vocab}}(w|\boldsymbol{x}, \tilde{\pi}_{<t}, C_{\mathcal{M}})$. $C_{\mathcal{M}}$ is the prefix tree based on 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 entities appearing in the input dialog history among the entities of a knowledge graph. (Kang et al., 2023) For example, in Figure 1 given the dialog “Do you know Lionel Messi?", the entity ‘Messi’ corresponds to the mentioned entity since it is in the knowledge graph. The next token prediction probability p_{vocab} is calculated only on the tokens included in the potential next token set of the constraint $C_{\mathcal{M}}$ (*i.e.*, $(\tilde{\pi}_{<t}, w) \in \mathcal{C}_{\mathcal{M}}$). Therefore, the generation model is forced to generate valid knowledge only.

In the graph-constrained decoding process, we also reflect the importance of each entity in the knowledge graph by defining the graph-based next-token prediction probability, which is formulated as:

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

where p_{graph} is the probability of predicting the next token based on graph information, and α is a hyperparameter. If the segment of a sequence $(\tilde{\pi}_{<t}, w)$ is the part of the tokenized entity e_i, the graph-based next-token probability is defined as:

$$p_{\text{graph}}(w|\tilde{\pi}_{<t}, C_{\mathcal{M}}) \propto \mathcal{S}(e_i, \mathcal{M}), \tag{4}$$

where $\mathcal{S}$ is the entity informativeness score of entity e_i on the mentioned entity set $\mathcal{M}$. If all entities have the same informative score, the prediction probability p_{graph} of all the entities becomes equal. The next token is decided only with the probability p_{vocab} produced by the language model. So, the structural information of the knowledge graph is not used when retrieving the subgraph.

To address it, we introduce a graph structure-based entity informativeness score. To consider the structural proximity between entity e_i and mentioned entities $e_m \in \mathcal{M}$ on the graph, we define the structure-based entity informativeness score (IS) as

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

where $s(e_i, e_m)$ is the graph structural proximity between entity e_i and e_m. The proximity between entities can be measured with structural information related to the node pairs such as degree, shortest path, and common neighbors (Katz, 1953; Gasteiger et al., 2019; Brin, 1998). Typical methods for graph structural proximity are the number of connections of the 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. Since it cannot capture multi-hop relations, we introduce a Katz index–based entity informativeness score (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 $\mathcal{G}$. Since $\mathbf{A}^k$ can be interpreted as the number of paths between entity e_i and e_m, this Katz index–based entity informativeness score captures multi-hop relationships between the node pair different from the simple connection-based score.

3.5 Response Generation

After retrieving the subgraphs, we generate the response given the dialog history and the retrieved subgraphs. We first apply the knowledge graph linearization to the retrieved subgraph $\hat{\mathcal{G}}$ to transform it into a token sequence, $\boldsymbol{z}_{\hat{\mathcal{G}}}$. Then, we compose the input sequence $\hat{\boldsymbol{x}}$ for our dialog generation model with the linearized subgraph $\boldsymbol{z}_{\hat{\mathcal{G}}}$ and the dialog history $\boldsymbol{x}$ (*i.e.*, $\boldsymbol{x} = \left[\boldsymbol{z}_{\hat{\mathcal{G}}}; \boldsymbol{x}\right]$, where $[;]$ denotes concatenation). The knowledge graph–augmented dialog input is fed into DialogGSR to generate the response $\boldsymbol{y}$.

3.6 Training DialogGSR

For training DialogGSR, we first self-supervise the special tokens with knowledge graph reconstruction loss. Then, we train our GSR to find knowledge graphs informative to dialog generation. We also train the dialog generator by minimizing response generation loss.

Knowledge graph reconstruction loss. Similar to the training scheme of masked language modeling (Roberts et al., 2020; Devlin et al., 2019), we propose a self-supervised method to learn the special tokens for the linearization by masking either an entity token or a relation token in the token

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.

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 $z_{\mathcal{G}'}$. Then, we randomly mask out either an entity or a relation of it. The loss is formulated as

$$\mathcal{L}_{\text{GraphRecon}} = -\log p(z_{\mathcal{G}'}|\hat{z}_{\mathcal{G}'}), \tag{7}$$

where $z_{\mathcal{G}'}$ is the token sequence of a sampled path and $\hat{z}_{\mathcal{G}'}$ is its randomly masked sequence. For example, a knowledge triplet $z_p = \langle$ ‘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 tuning.

Graph retrieval loss. To train our generative subgraph retriever, our method minimizes the knowledge subgraph retrieval loss defined as:

$$\begin{aligned}\mathcal{L}_{Ret} &= \mathbb{E}_{\boldsymbol{x}}\left[-\log p\left(\mathcal{G}^{\star}|\boldsymbol{x}\right)\right] \\ &= \mathbb{E}_{\boldsymbol{x}}\left[-\log p\left(\tilde{\tau}^{\star(1)},\ldots,\tilde{\tau}^{\star(k)}|\boldsymbol{x}\right)\right],\end{aligned} \tag{8}$$

where $\mathcal{G}^{\star}$ is the gold subgraph, and $\tau^{\star(1)}\ldots\tau^{\star(k)}$ are gold triplets.

Response generation loss. 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:

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

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

4 Experiments

4.1 Experimental Setup

For fair comparisons with previous works, we use T5-small (Roberts et al., 2020) as the base PLM. For datasets, we use two datasets (OpenDialKG (Moon et al., 2019) and KOMODIS (Galetzka et al., 2020)). **OpenDialKG** is a dataset that consists of 15K dialogs with 91K turns and 1.12M triplets from Freebase (Bast et al., 2014) knowledge graph. **KOMODIS** is a dataset that consists of 7.5k dialogs with 103k turns and the corresponding KG, which contains 88K triplets. We follow Galetzka et al. (2020); Kang et al. (2023) to split the dialogs into train (70%), validation (15%), and test (15%) sets for both datasets. More details are in Appendix B.

Graph Const.	Special tokens	B-1	B-2	B-3	B-4	path@3
w/o Const.	with Soft Prompt (w/o Recon.)	17.66	10.96	7.32	5.14	10.00
Hard Const.	w/o Special tokens	18.51	11.68	7.90	5.60	35.83
Hard Const.	with Special tokens (w/o Recon.)	18.90	11.91	7.90	5.44	39.53
Hard Const.	with Special tokens (with Recon.)	19.09	11.87	7.96	5.44	43.27
Connection Const.	with Special tokens (with Recon.)	19.14	11.87	8.08	5.57	45.85
Katz Const.	with Special tokens (with Recon.)	**19.30**	**12.10**	**8.30**	**5.83**	**46.76**

Table 4: 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 5: 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.

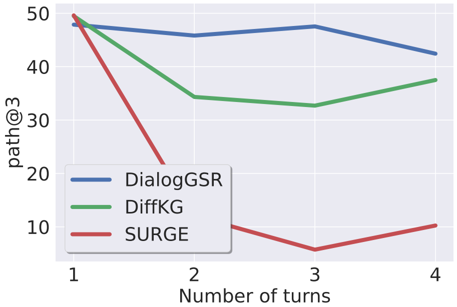

Figure 2: Retrieval performance according to the number of turns.

4.2 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.

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.

In addition to the quantitative results above, we also provide human evaluation in Appendix A.1. It shows that our DialogGSR generates more preferable responses than baselines.

4.3 Analysis

We analyze DialogGSR to answer the following research questions: **[Q1]** Does each component of DialogGSR contributes 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 our 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 4. **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 4 shows that each component contributes to the performance improvement of the model. In particular, graph-constrained decoding is crucial in our generative approach.

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

In addition, the models with graph constraints show improvements compared to the model without the constraints, which means that the graph constraint is important for 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.

Information bottleneck issue. Information bottleneck issue (Humeau et al., 2020; Cao et al., 2021; Lee et al., 2022) usually occurs when a long text sequence, such as a dialog history, is encoded into a fixed single vectors. 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 dialog histories whereas the other methods often deteriorate as the number of turns increases.

Effectiveness of DialogGSR with LLMs. To assess the effectiveness of our DialogGSR in other LLMs (**[Q4]**), we apply it to LLaMA-3 (Meta, 2024) in Table 5. From the table, the performance gain of DialogGSR compared to the base model is 2.42 in BLEU-1 score. In addition, our proposed graph-constrained decoding is still important in LLMs. These results indicate that DialogGSR is also effective in other LLMs.

Qualitative analysis. We perform qualitative analysis by comparing responses generated from SURGE and DialogGSR. Table 6 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", our 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.3.

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 design graph-constrained decoding for valid and relevant generative retrieval. Our experiments demonstrate the effectiveness of our proposed method in knowledge grounded dialog generation.

Limitations

DialogGSR generatively retrieves token sequences of the subgraph from the 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 retrieves only the knowledge contained in the knowledge graph. Second, the benchmark dataset for knowledge graph–grounded dialog generation is limited. Except for OpendialKG (Moon et al., 2019) dataset, there is no dataset on dialog generation with a large-scale knowledge graph. So, new benchmark datasets on dialog generation with knowledge graphs deserve more attention.

Ethics Statement

Our DialogGSR does not have any direct negative social impacts, 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.

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 preprint 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 *Advances in Neural Information Processing Systems*, volume 26.

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 *International Conference on Learning Representations*.

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*.

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*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *arXiv*.

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 *Findings of EMNLP*, 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 *Proceedings of The Third Workshop on Simple and Efficient Natural Language Processing (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 *Findings of ACL*, 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*.

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.

A Additional experiments

A.1 Human Evaluation

In this subsection, we provide human evaluation results to assess the generated responses of our dialog generation model. 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 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.

Table 7 shows the experimental results of the human evaluation. As shown in the table, our DialogGSR shows better performance compared to SURGE in all the metrics (Consistency, Informativeness, Fluency). In particular, on the Consistency and Informativeness metrics, DialogGSR achieves statistically significant performance 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.

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 7: Human evaluation results. () indicates standard deviation.

A.2 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 triplet 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.

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.3 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 Datasets

OpenDialKG[1] is an open-domain dialog dataset that consists of 15K dialogs with 91K turns and 1.12M triplets from Freebase (Bast et al., 2014) knowledge graph. 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 dialogs into train (70%), validation (15%), and test (15%) sets. With OpenDialKG dataset, we evaluate the response generation and retrieval performance of our DialogGSR with other baselines following (Kang et al., 2023; Tuan et al., 2022). The baseline results in Table 1 of the main paper are reported from (Kang et al., 2023).

KOMODIS[2] 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. With KOMODIS dataset, we evaluate the response generation performance of our DialogGSR with other baselines following (Kang et al., 2023; Galetzka et al., 2021). The baseline results in Table 2 of the main paper are reported from (Kang et al., 2023).

B.2 Implementation details

In this section, we describe the implementation details not included in our main paper. For all the experiments, we use PyTorch[3] (Paszke et al., 2019) and Transformer module of Huggingface[4] (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. The TopK-Select$(\cdot)$ function of the algorithm selects Top-k candidates from all candidates $w \in \mathcal{V}$ by calculating $\log \tilde{p}\left(w|\boldsymbol{x}, \boldsymbol{y}_{<t}^{(i)}\right) + \log \tilde{p}\left(\boldsymbol{y}_{<t}^{(i)}|\boldsymbol{x}\right)$. We use $\alpha = 0.8$ and $k = 2$ for calculating Katz (Katz, 1953) index-based entity informativeness score. p_{graph} is defined in Equation 6 of the main paper, and b is 5.

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 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).

C Baselines

C.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)**[5] (Roberts et al., 2020): T5-small is an encoder-decoder Transformer architecture for various natural language processing tasks.

[1] Licensed under CC-BY-NC-4.0

[2] Copyright (c) 2020 Fabian Galetzka, Licensed under MIT license

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

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

[5] Licensed under the Apache license

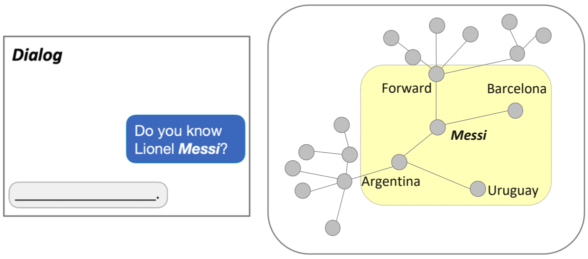

Figure 3: An example of extracting a 1-hop candidate subgraph from the knowledge graph. Yellow region indicates the 1-hop candidate subgraph centered on mentioned entity “Messi".

- **Space Efficient (series)**[6] (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.

C.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 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.

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

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 GSR ⟨ ‘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 GSR ⟨ ‘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 GSR ⟨ ‘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 GSR ⟨ ‘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.

The following is a conversation between a user and a knowledgeable system. The system's utterances are grounded on the background knowledge:
Knowledge:
(Iron man, starred_actors, Robert Downey Jr.)
Conversation:
User : Do you like Iron Man?
System: Sure do! Robert Downey Jr. is a favorite.

. . .

The following is a conversation between a user and a knowledgeable system. The system's utterances are grounded on the background knowledge:
Knowledge:
{Retrieved knowledge}
Conversation:
{Dialog}
System:

Table 10: Response generation prompt for Mixtral 8x7B.