SG-RAG: Multi-Hop Question Answering
With Large Language Models Through Knowledge Graphs

Ahmmad O. M. Saleh
Sabanci University
ahmmad@sabanciuniv.edu

Gokhan Tur
University of Illinois
Urbana-Champaign
gokhan@illinois.edu

Yucel Saygin
Sabanci University
ysaygin@sabanciuniv.edu

Abstract

Large Language Models (LLM) such as GPT3 and Llama tend to hallucinate, especially for domain-specific questions. To alleviate this problem, Retrieval Augmented Generation (RAG) has been proposed but LLMs still suffer in multihop question answering even with RAG. Knowledge Graphs represent domain information in a structured manner and they have been used for reasoning in AI. In this work, we propose SubGraph Retrieval Augmented Generation (SG-RAG), a novel zero-shot Graph RAG method that exploits the structured information in Knowledge Graphs in order to accurately answer multihop questions with LLMs. We form a Cypher query based on the given question to retrieve the set of relevant subgraphs that is further provided as context to the Language Model. We implemented and tested our methodology on a benchmark question-answering data set on movies domain. Experiments show that the accuracy of 2-hop and 3-hop questions issued to LLAMA 8B Instruct and GPT4-Turbo significantly increases compared to LLAMA and GPT with and without RAG.

1 Introduction

Language Models have revolutionized how we represent knowledge and significantly impacted question-answering systems. Large Language Models (LLM) have proven to be very effective in generating convincing answers, especially for generic questions Touvron et al. (2023). However, they also tend to hallucinate when they encounter domain-specific questions Tonmoy et al. (2024). In the case of LLMs such as LLAMA, hallucination becomes a severe problem Li et al. (2024). In Table 1, we provide sample questions submitted to LLAMA3 8B Instruct where the answers show hallucinations of the model. In order to alleviate this problem, Retrieval Augmented Generation (RAG) was proposed by Lewis et al. (2020). With RAG, questions are answered based on a set of documents where documents most similar to the given question are retrieved and provided as context to the LLM. The semantic similarity of a question to the documents is calculated through word embeddings and the top few documents are provided as context. RAG eliminates most of the hallucinations in the case of single-hop questions such as "When has been the release year of the film No Looking Back", but for multihop questions like "Senator William Broyles Jr. wrote films with whom" (2-hop) and "When were the release years of the films led by Edges of the Lord as director"(3-hop), LLAMA3 8B Instruct fails to give correct answers. In order to understand the degree of hallucination, we evaluated LLAMA 8B on a benchmark Question-Answer data set where the questions and corresponding answers are provided. We observed that single-hop questions are answered with high accuracy, while for 2-hop questions the accuracy drops drastically, and for 3-hop questions the accuracy decreases even further.

In order to improve their performance, an alternative form of giving context to LLMs was proposed in the form of Knowledge Graphs (KGs). KGs provide domain information in a structured way. The term Graph RAG was coined in a blog by Microsoft Research Larson and Truitt (2024) where the authors highlighted the limitations of the standard RAG method in answering questions that require multiple pieces of information. They suggested transforming the unstructured documents into a knowledge graph as a solution to those limitations.

In this work, we introduce the SubGraph Retrieval Augmented generation (SG-RAG), a novel zero-shot Graph RAG method that exploits the relations stored in KGs to answer questions. An overview of SG-RAG is demonstrated in Figure 1. SG-RAG uses Cypher statements representing the semantics of the questions to retrieve the set of subgraphs containing relevant information from KG. SG-RAG then transforms the subgraphs into a tex-

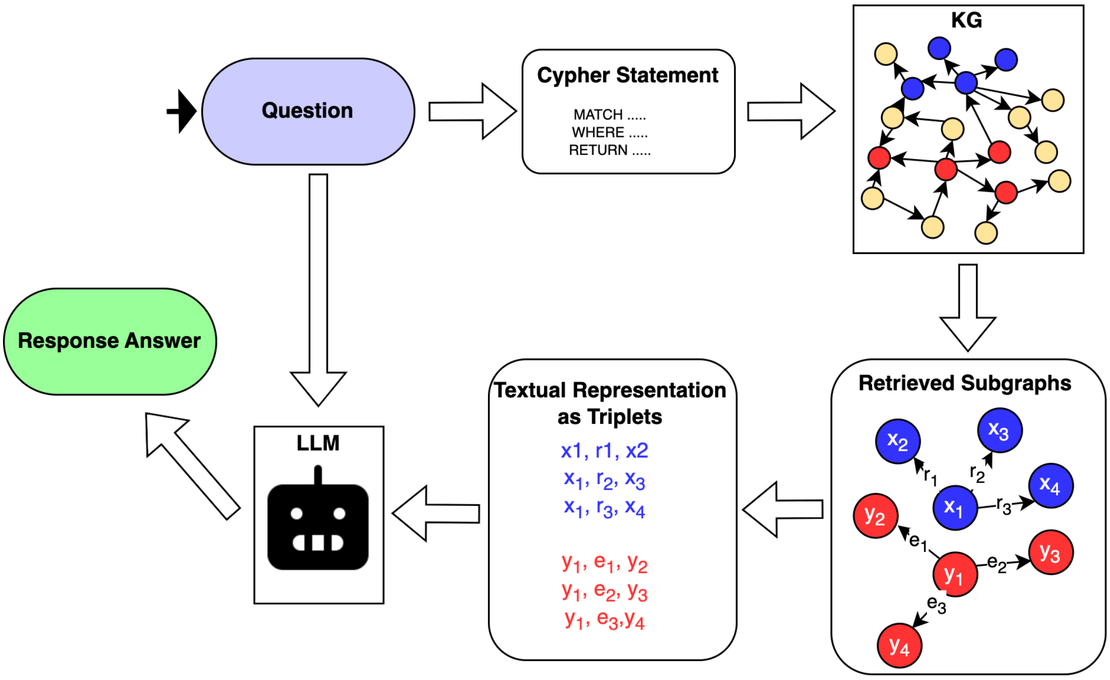


Figure 1: An Overview of SG-RAG Methodology.

tual representation in the form of triplets. Triplets are partitioned into groups based on the retrieved subgraphs as highlighted in Figure 1. Triplets are finally provided to the LLM as context to generate an answer. The task instruction sent to LLM highlights the structure of the triplets by using $(subject, relation, object)$ format where the direction of the $relation$ from $subject$ to $object$ is emphasized. The experiments show that SG-RAG provides more accurate answers compared to LLM with and without RAG, especially for multihop questions.

2 Background and Related Work

Large Language models (LLMs) are being used for language understanding and question answering Touvron et al. (2023); Reid et al. (2024). However, LLMs struggle to provide accurate answers for domain-specific questions, generating factually wrong, or irrelevant random responses which are referred to as model hallucination Tonmoy et al. (2024). Retrieval Augmented generation (RAG) works on solving the hallucination problem by embedding chunks of the domain information in the LLM prompt with the question Lewis et al. (2020); Gao et al. (2023). The choice of the information chunks shared with LLM is determined by measuring the semantic similarity between the question and the domain knowledge information. The RAG method with LLMs has been applied to question-answering systems targeting a wide range of domains including Finance by Setty et al. (2024), Medicine by Zakka et al. (2024), and Religion by Alan et al. (2024) to name a few. RAG method with semantic similarity search decreases the LLMs' hallucination on simple domain-specific questions, however, the semantic similarity search is not able to retrieve the indirect information required to answer multihop questions Larson and Truitt (2024).

After the success of LLMs on many Natural Language Processing tasks, researchers started to investigate the potential of using LLMs on graphs Jin et al. (2023). such as using LLMs on classifying graph nodes Chen et al. (2024). Edge et al. (2024) use LLMs to convert unstructured knowledge into a knowledge graph to answer global questions that require a global awareness of the domain knowledge. The method proposed by Edge et al. (2024) is based on creating groups, "communities" as they call them, of subgraphs. Each community has a summarization of the information stored in the community subgraph. The answer to any global question is based on collecting the summarization from the communities and providing an answer based on them. The most relevant work to ours is a recent paper by Jin et al. (2024) where authors propose Graph Chain-of-Thought (Graph-COT) approach. Their aim is to tackle questions that require reasoning by giving LLM the ability to interact with the knowledge graph directly through

Table 1: Examples of the LLAMA3 8B Instruct hallucination on multihop questions from movies domain (with and without RAG).

1-hop	Question	When has been the release year of the film No Looking Back
	Gold Answer	1998
	LLAMA3-8B	2019
	LLAMA3-8B with RAG	June 28, 2024 | Japanese theaters | Annecy International Animation Film Festival (June 9-15, 2024) | Annecy Presents category | Annecy International Animation Film Festival | June 28, 2024 | Japanese theaters | Annecy International Animation Film Festival
2-hop	Question	Senator William Broyles Jr. wrote films with whom
	Gold Answer	Alvin Sargent, Claude Chabrol
	LLAMA3-8B	Tom Cruise Explanation: Tom Cruise and William Broyles Jr. co-wrote the film "The Polar Express". William Broyles Jr. is a screenwriter and his films include "The Polar Express" and "Cast Away". Tom Cruise is an actor who has starred in several films, including "Top Gun"
	LLAMA3-8B with RAG	Robert Redford, Steven Spielberg, Tom Cruise, Barry Levinson, Oliver Stone, Ron Howard, James Cameron, Brian De Palma, Martin Scorsese, Robert Zemeckis, Barry Sonnenfeld, Edward Zwick, Steven Soderbergh, Clint Eastwood, Sam Raimi
3-hop	Question	When were the release years of the films led by Edges of the Lord as director
	Gold Answer	1987, 1993
	LLAMA3-8B	1. The Lord of the Rings: The Fellowship of the Ring (2001) - December 19, 2001 2. The Lord of the Rings: The Two Towers (2002) - December 18, 2002 3. The Lord of the Rings: The Return of the King (200
	LLAMA3-8B with RAG	2001, 2002, 2005, 2007, 2008, 2002, 2005

a set of predefined functions. Hence, the LLM has the freedom to decide when and how to traverse the graph to gain the information needed to answer the question. The graph description, including the nodes' type, attributes, and outer edges, is augmented in the LLM prompt. The main limitation of Graph-COT is that the model can reach a dead-end in cases where the initial node does not have any outer edge as demonstrated in Figure 3 where the question is asking about the release years of the movies *Sharon Tate* acted. In this case the LLM in Graph COT will start with *Sharon Tate* node. Since all the edges connected to the *Sharon Tate* node are incoming edges as in Figure 2, LLM will not be able to traverse other nodes to find the release years of *The Wrecking Crew* and *Valley of the Dolls* from *Sharon Tate*, therefore LLM will not be able to answer the question correctly. Another point is that Graph-COT works on GPT3.5 Turbo which is an advanced black box model, however, when we run Graph-COT on LLAMA3-8B which is an open-source model with a much lower number of parameters compared to GPT, we observed many hallucinations for our benchmark questions.

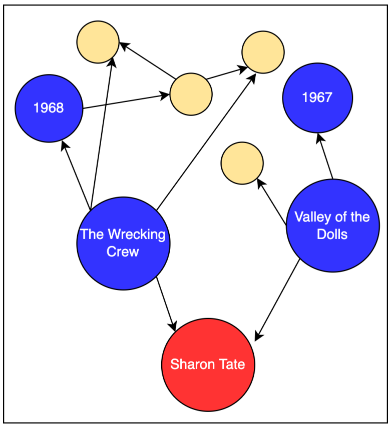


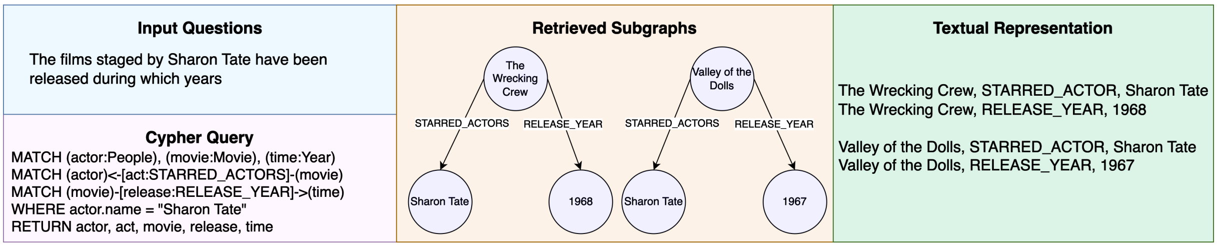
Figure 2: An Illustration of a Dead-end Scenario for Graph-COT.

Other individual and commercial experiments have been conducted on the LLM and KG, as highlighted by Kollegger (2024) stressing the importance of using KG with LLM and providing approaches to combine them, and the blog-post written byAlto (2024) as an implementation tutorial of applying a hybrid approach of RAG and KG with LLM using LangChain and Neo4j graph database. In this paper, we propose a novel Graph RAG methodology based on subgraph retrieval that we call SG-RAG to address the problem of multi-hop question answering.

3 Preliminaries and Problem Definition

In the following paragraphs, we define preliminary concepts that will be used in the problem statement.

Definition 3.1. Graph. A graph $G = (V, E)$ is a data structure consisting of a set of nodes, denoted by V, and a set of edges, denoted by E. For any



The films staged by Sharon Tate have been released during which years

```
MATCH (actor:People), (movie:Movie), (time:Year)
MATCH (actor)<-[act:STARRED_ACTORS]-(movie)
MATCH (movie)-[release:RELEASE_YEAR]->(time)
WHERE actor.name = "Sharon Tate"
RETURN actor, act, movie, release, time
```

The Wrecking Crew, STARRED_ACTOR, Sharon Tate
The Wrecking Crew, RELEASE_YEAR, 1968

Valley of the Dolls, STARRED_ACTOR, Sharon Tate
Valley of the Dolls, RELEASE_YEAR, 1967

Figure 3: An Example of the SG-RAG Retrieval for a 2-hop Question.

edge $e_i \in E$, there exists two nodes, $v_j, v_k \in V$, such that e_i connects v_j and v_k. A graph can be directed or undirected where the edges in the former have a direction.

Definition 3.2. Subgraph Given a graph $G = (V, E)$, a graph $G' = (V', E')$ is a subgraph of G if and only if $V' \subseteq V$ and $E' \subseteq E$.

Definition 3.3. Knowledge Graph. A knowledge Graph, KG, represents domain knowledge in a graph data structure. In KG, a node represents a unit of information and an edge represents the relation(s) between the two nodes. Nodes and edges may have a label describing the type of knowledge or relations in the nodes or edges respectively. Each node $v_j \in V$ may have extra information embedded in the form of attributes. Attributes can differ based on the type of the nodes. For example in case of Movies KG, the possible node labels are Movies, People, Genre,...etc. The set of node labels with the candidate edge types represents the schema of the KG. Table 2 describes the schema of MetaQA-KG that we used in our experimental evaluation.

Definition 3.4. n-hop Question. An n-hop question is a question that requires one or more subgraphs from a KG such that each subgraph contains n edges. For example, in order to answer the 2-hop question in Figure 3, we need two subgraphs where each subgraph contains 2 edges, "STARRED _ACTORS" and "RELEASE _YEAR".

Problem Definition: For any domain D represented by a knowledge graph KG, our aim is to accurately answer n-hop questions Q about D. We assume that the questions are about the entities in KG with specific relations to other entities. For example, for the 1-hop question: "What are the movies directed by Sharon Tate?" we are interested in the *Movies* entity that is related to *Sharon Tate* entity with *directed by* relation. The expected answers to the questions in Q are the set of entities that satisfy the constraints in the form of relations provided in the question.

4 SG-RAG Methodology for Multihop Question Answering

In this section, we present SubGraph Retrieval Augmented Generation (SG-RAG), a novel zero-shot Graph RAG method for answering domain-specific multihop questions using KG and LLM. SG-RAG has two main steps, subgraph retrieval and response generation. The Subgraph Retrieval is based on querying KG using a Cypher statement representing the input question and then transforming the retrieved subgraphs into a set of triplets. The response generation step takes the input question and the resulting triplets from the retrieval step and augments them into a prompt with an instruction to LLM, then the prompt is sent to LLM to generate a response to the question. The flow of SG-RAG is demonstrated in Figure 1. The following subsections explain the subgraph retrieval and response generation in detail.

4.1 Subgraph Retrieval

Rather than retrieving a specific piece of information from KG as in Bratanič (2024); Alto (2024) such as list of movie names, the subgraph retrieval step relies on Cypher statements to retrieve a set of subgraphs from the KG containing the required relevant information to answer the input question. Then, SG-RAG transforms the retrieved subgraphs into a textual representation that will be provided as context to the LLM prompt during the response generation step.

Querying the Knowledge Graph: Cypher is a query language design by Neo4j for property graphs built following Graph Theory Francis et al. (2018). For an input question q, we use a statement in Cypher Query Language to be executed on the knowledge graph. The Cypher statement searches for the set of subgraphs with nodes containing the answer of q, then returns the found subgraphs as records. In the example shown in Figure 3, the Cypher statement aims to retrieve the movies in which Sharon Tate acts together with the release years of those movies. The result of executing the Cypher statement is two subgraphs shown in Figure 3. We use domain-specific Cypher templates to generate Cypher queries for the benchmark questions.

Transformation of Subgraps into Textual Representation: The transformation is based on converting each pair of nodes connected by an edge into a triplet of the form $(Subject, Relation, Object)$. More Precisely, given two nodes n and m connected by a directed edge e from n to m, the resulting triplet will be $(n,\ e,\ m)$. Textual transformation needs to preserve the partial order imposed by the retrieved subgraphs. Therefore the triplets from the same subgraph are grouped together. Grouping triplets based on the subgraphs helps the LLM extract the correct information and prevents it from getting confused between the different subgraphs. In Figure 3, two retrieved subgraphs were depicted where each subgraph contains 2 edges, hence the textual representation includes 4 triples split into two groups.

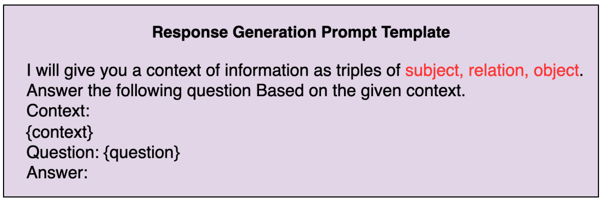
4.2 Response Generation

The LLM prompt used to generate a response contains the task instruction, the context as the set of triplets coming from the retrieval step, and the input question. The task instruction is a simple instruction explaining the task to the LLM and describing the structure of the triplets. During our initial trials, we explored different prompt templates that differ in the task instruction such as $(entity, relation, entity)$ which does not capture the directed edge structure of the KG. Using the $(subject, relation, object)$ gave the best results since it provides the LLM additional information about the direction of the relation such that the $relation$ is from $subject$ to $object$. The final version of the prompt template we used is demonstrated in Figure 4. After creating the prompt based on the prompt template, it is sent to the LLM to generate a response.

5 Experimental Setup

5.1 Dataset

MetaQA is a benchmark dataset introduced by Zhang et al. (2018). It includes a knowledge graph (MetaQA-KG) based on data about movies. In addition to the knowledge graph, it contains question-answer pairs about MetaQA-KG. The questions are generated through templates, and a paraphrased version of the questions called NTM is created by translating them to French and then back to English. Each question has a single category out of 49 categories. The question-answer pairs are divided into 1-hop, 2-hop, and 3-hops. For our experiments, we randomly picked 15K NTM questions with equal number of 1-hop, 2-hop, and 3-hop questions.



Response Generation Prompt Template

I will give you a context of information as triples of subject, relation, object.
Answer the following question Based on the given context.
Context:
{context}
Question: {question}
Answer:

Figure 4: The Prompt Template Used for SG-RAG Response Generation.

MetaQA-KG contains 9 types of relations: "directed by", "written by", "starred actors", "release year", "in language", "has tags", "has genre", "has imdb votes", and "has imdb rating". Based on the semantics of those relations, we divided the entities into 8 groups: Movies, People, Year, Language, Tag, Genre, IMDB Votes, and IMDB Rating. Hence, the Graph Schema of the MetaQA-KG becomes as shown in Table 2 such that each entity has a single attribute called name, while relations don't have attributes.

Table 2: The Graph Schema of MetaQA-KG after grouping the entities based on the semantics of the relations.

(:Movies)-[:DIRECTED _BY]->(:People)
(:Movie)-[:WRITTEN _BY]->(:People)
(:Movie)-[:STARRED _ACTORS]->(:People)
(:Movie)-[:IN _LANGUAGE]->(:Language)
(:Movie)-[:RELEASE _YEAR]->(:Year)
(:Movie)-[:HAS _GENRE]->(:Genre)
(:Movie)-[:HAS _TAGS]->(:Tag)
(:Movie)-[:HAS _IMDB _VOTES]->(:Vote)
(:Movie)-[:HAS _IMDB _RATING]->(:Rate)

5.2 Baselines

We consider the following baselines in our experiment:

- LLM: Using the LLM alone to answer the questions. The answers are based on the internal knowledge stored in the model's parameters.
- RAG: It is based on the original RAG method proposed by Lewis et al. (2020). The external

knowledge is represented by a set of plain-text documents.

5.3 Implementation Settings

Since the MetaQA benchmark does not contain the Cypher queries, we generated them based on templates. The generation process is based on creating a Cypher query template for each category. A subset of the query templates is provided in Table 3. Cypher statements are generated by replacing the "<entity>" tag with the entity name in the corresponding question.

Our baseline RAG Lewis et al. (2020) is based on indexing plain-text documents into a vector database using textual embedding. Since the knowledge in MetaQA is a graph structure, we retrieved Wikipedia documents about the entities that appear in our test questions. The retrieved Wikipedia documents are split into chunks with a maximum size of 100 words as in Lewis et al. (2020) that are indexed into a vector database. We used LLAMA-3 8B Instruct version AI@Meta (2024) as the backbone LLM for the baselines and SG-RAG.

5.4 Evaluation Metric

We evaluate the performance of SG-RAG and the baselines using the answer-matching rate inspired by the notion of entity-matching rate proposed by Wen et al. (2017) to evaluate the dialogue systems. The answer matching rate measures the ratio of the gold answers contained in the generated response. More specifically, let q be an input question, $Y = y_1, y_2, .., y_m$ be the gold answer, and $Y' = y'_1, y'_2, .., y'_n$ be the generated response, then:

$$MatchingRate(q) = \frac{\mid Y \cap Y' \mid}{\mid Y \mid} \tag{1}$$

The gold answers in MetaQA are a set of entity names whereas the LLM responses have a paragraph structure with explanations. Therefore, we have decided to use matching rate metric which considers only the part of the LLM generated text that is within the scope of our knowledge base.

6 Results and Discussion

Using the MetaQA dataset and the matching rate metric, we evaluated SG-RAG and compared it with the baselines. The results are demonstrated in Table 4. From the result in Table 4, we observe that the performance of the LLM alone is poor compared to other methods. This shows that relying on LLM internal knowledge alone is not enough to answer questions on a specific domain, such as Movies.

RAG has better performance compared to the LLM alone. However, the performance of RAG decreases for 2-hop and 3-hop questions. The reason behind that is the external knowledge shared with the LLM as a context is determined by the semantics of the question which is not enough to know the extra information required to answer the question. Coming back to the example in Figure 3, using the semantics of the question, RAG retrieved the documents about "Sharon Tate" which include the names of the movies she acted such as "The Wrecking Crew" and "Valley of the Dolls", but those documents do not contain extra information about the movies such as the release year, the language, or the name of the cast. RAG cannot retrieve all the necessary documents about "The Wrecking Crew" or "Valley of the Dolls" by the mere semantics of the question. This problem of RAG is addressed by SG-RAG which we can observe in Table 4 where SG-RAG outperforms the baseline methods for 1-hop, and even more for 2-hop, and 3-hop questions. SG-RAG uses the KG as an external knowledge source where the relations between the entities are represented in the structure of the graph. Moreover, we use Cypher queries to retrieve information from the KG and fully capture the structural information provided by the KG. This can also be seen in the example provided in Figure 3 where Cypher query asked to retrieve all the movies in which "Sharon Tate" was an actress, and the release year of those movies. This way, the LLM received all the information needed to answer the question.

Generating Documents based on Knowledge Graph: The low performance of the RAG with Wikipedia documents on the 1-hop questions may be caused by the fact that Wikipedia does not include the answers to our questions. To analyze that issue, we also generated documents based on the information in our knowledge graph. The generation process started with extracting the entities in our questions. Then, for each entity, we extracted the subgraph containing the targeted entity node and the neighborhood of the node. After that, we asked Gemini 1.5 Flash to generate a 100-word document about the targeted entity containing the information in the subgraph. The subgraph is embedded in the Gemini prompts as a set of triplets. Figure 5 shows the prompt template we used to construct

Table 3: Sample question categories and their corresponding Cypher templates.

Type	Category	Cypher Template

1-hop	movie to language	MATCH (m:Movie)-[r:IN _LANGUAGE]->(l:Language)
		WHERE m.name="<entity>"
		RETURN m, r, l

	director to movie	MATCH (m:Movie)-[r:DIRECTED _BY]->(d:People)
		WHERE d.name="<entity>"
		RETURN m, r, d

2-hop	writer to movie to genre	MATCH (w:People)<-[r1:WRITTEN _BY]-(m:Movie)
		-[r2:HAS _GENRE]->(g:Genre)
		WHERE w.name="<entity>"
		RETURN w, r1, m, r2, g

	actor to movie to year	MATCH (a:People)<-[r1:STARRED _ACTORS]-(m:Movie)
		-[r2:RELEASE _YEAR]->(y:Year)
		WHERE a.name="<entity>"
		RETURN a, r1, m, r2, y

3-hop	movie to director to movie to actor	MATCH (m1:Movie)-[r1:DIRECTED _BY]->(d:People) <-[r2:DIRECTED _BY]-(m2:Movie) -[r3:STARRED _ACTORS]->(a:People) WHERE m1.name="<entity>" RETURN m1, r1, d, r2, m2, r3, a
	movie to writer to movie to language	MATCH (m1:Movie)-[r1:WRITTEN _BY]->(w:People) <-[r2:WRITTEN _BY]-(m2:Movie) -[r3:IN _LANGUAGE]->(l:Language) WHERE m1.name="<entity>" RETURN m1, r1, w, r2, m2, r3, l

Table 4: The evaluation results of SG-RAG with LLAMA3-8B Instruct and the baselines on the MetaQA selected test set.

	1-hop	2-hop	3-hop
LLAMA3-8B	0.24	0.13	0.17
RAG(Wiki Docs) Top-1	0.33	0.19	0.21
RAG(Wiki Docs) Top-2	0.36	0.20	0.20
RAG(Wiki Docs) Top-3	0.38	0.22	0.20
RAG(Wiki Docs) Top-5	0.40	0.23	0.18
RAG(Wiki Docs) Top-10	0.42	0.27	0.19
SG-RAG	0.90	0.73	0.58

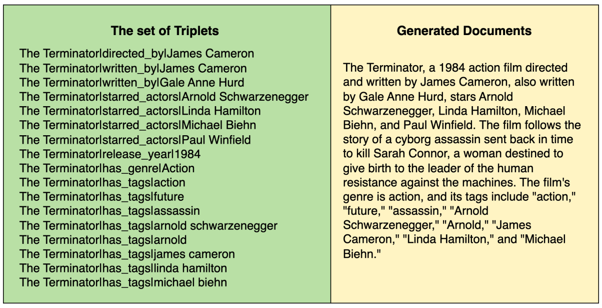
the templates we sent to Gemini to generate the document. Figure 6 provides an example of the generated document about *The Terminator* movie by Gemini based on the set of triplets representing the subgraph containing *The Terminator* node and its neighborhood. We randomly sampled a set of 1547 1-hop questions, 1589 2-hop questions, and 1513 3-hop questions, to apply this experiment within a limited time frame. From the results in Table 5, we can see that applying RAG on the generated documents based on KG achieved higher performance than the RAG on Wikipedia documents since each document contains the information of a 1-hop neighborhood around the targeted entity. However, the performance of RAG on both the generated and Wikipedia documents is comparable with 2-hop and 3-hop questions while SG-RAG has superior performance for 1-hop, 2-hop, 3-hop questions.

Document Generation Prompt Template

Write a paragraph to me about "{entity}" using these relation triplets. The paragraph should include all the information in the relation triples. Each triplet is separated by ' ; '. The paragraph should be at most 100 words long.
the relation triples:
{triplets}

Figure 5: The Prompt Template Used With Gemini for Documents Generation

Using GPT4-Turbo as backbone LLM: The low performance of RAG compared to SG-RAG even on the Gemini generated documents may be caused by the LLAMA3-8B Instruct that we chose as a backbone LLM for our evaluation. To analyze that issue further, we evaluated SG-RAG, and RAG on the Gemini generated documents on GPT4-Turbo. We did this experiment on the same small test set we used earlier to apply this experiment within a limited time frame. From the results in Table 6, we can see the superior performance of SG-RAG on 1-hop, 2-hop, and 3-hop questions. For RAG, we can notice that increasing the number of documents shared with GPT4 on 2-hop and 3-hop questions affected GPT4 negatively and decreased its performance.



The set of Triplets

The Terminator|directed_by|James Cameron
The Terminator|written_by|James Cameron
The Terminator|written_by|Gale Anne Hurd
The Terminator|starred_actors|Arnold Schwarzenegger
The Terminator|starred_actors|Linda Hamilton
The Terminator|starred_actors|Michael Biehn
The Terminator|starred_actors|Paul Winfield
The Terminator|release_year|1984
The Terminator|has_genre|Action
The Terminator|has_tags|action
The Terminator|has_tags|future
The Terminator|has_tags|assassin
The Terminator|has_tags|arnold schwarzenegger
The Terminator|has_tags|arnold
The Terminator|has_tags|james cameron
The Terminator|has_tags|linda hamilton
The Terminator|has_tags|michael biehn

Generated Documents

The Terminator, a 1984 action film directed and written by James Cameron, also written by Gale Anne Hurd, stars Arnold Schwarzenegger, Linda Hamilton, Michael Biehn, and Paul Winfield. The film follows the story of a cyborg assassin sent back in time to kill Sarah Connor, a woman destined to give birth to the leader of the human resistance against the machines. The film's genre is action, and its tags include "action," "future," "assassin," "Arnold Schwarzenegger," "Arnold," "James Cameron," "Linda Hamilton," and "Michael Biehn."

Figure 6: The set of Triplets Representing the Subgraph Containing *The Terminator* Node and Its Neighborhood on the Left and Gemini Generated Document on the Right.

Table 5: Comparison between SG-RAG, RAG on Wikipedia documents, and RAG on Gemini generated documents using LLAMA3-8B Instruct.

	1-hop	2-hop	3-hop
RAG(Wiki Docs) Top-1	0.33	0.19	0.21
RAG(Wiki Docs) Top-2	0.35	0.20	0.20
RAG(Wiki Docs) Top-3	0.36	0.22	0.20
RAG(Generated Docs) Top-1	0.64	0.15	0.17
RAG(Generated Docs) Top-2	0.66	0.12	0.13
RAG(Generated Docs) Top-3	0.66	0.12	0.16
SG-RAG	0.91	0.72	0.60

7 Conclusions

LLM with RAG has significantly impacted question-answering systems in multiple domains such as Finance by Setty et al. (2024), Medicine by Zakka et al. (2024), and Religion by Alan et al. (2024), to name a few. However, RAG is still suffering from hallucinations on multi-hop questions. In this work, we propose SG-RAG, a zero-shot Graph RAG method to answer multi-hop domain-specific questions that use Cypher statement representing the question to retrieve the set of subgraphs containing the required information to answer the question. SG-RAG is a method designed to exploit the structured information in Knowledge Graphs to increase the LLMs performance on multi-hop domain-specific questions. For an input question, SG-RAG uses a Cypher query representing the input question to retrieve the set of subgraphs containing the required information, then shares it as a context to the LLM. We evaluate our method on a question-answering benchmark dataset on movies. Our experiments show a significant increase in performance in general and specifically on 2-hop and 3-hop questions.

Table 6: Comparison between SG-RAG, and RAG on Gemini generated documents using GPT4-Turbo.

	1-hop	2-hop	3-hop
RAG(Generated Docs) Top-1	0.765	0.286	0.204
RAG(Generated Docs) Top-2	0.776	0.181	0.177
RAG(Generated Docs) Top-3	0.784	0.179	0.180
SG-RAG	0.941	0.815	0.520

Limitations

This work mainly focuses on introducing SG-RAG as a zero-shot Graph RAG method to answer multi-hop domain-specific questions. During our experiment, the Cypher statements are generated manually using domain-specific Cypher templates. To overcome the challenge of manually generating the domain-specific Cypher templates, we are working on automatically generating the Cypher statement representing the targeted question based on the KG schema as an extension to SG-RAG. In our initial trials, we observed that LLAMA3-8B and Gemini are very poor at generating valid Cypher queries. GPT-4 can generate Cypher queries, but accuracy needs to be improved. In order to address this problem we plan to fine-tune an LLM such as LLAMA3-8B to give it the ability to generate a Cypher query given the question and the graph schema.

Within the limited time frame, we evaluated SG-RAG on GPT4-Turbo over a small test set; however, we are working on extending the evaluation over a larger sample size and comparing its performance with Graph COT proposed by Jin et al. (2024).

Ethics Statement

Large Language Models (LLM) have achieved outstanding performance in natural language processing and generation, specifically in question-answering systems Touvron et al. (2023). However, the hallucination of these models can generate factual mistakes in answers or misleading information Tonmoy et al. (2024) that can be later propagated amoung people as facts. We are proposing SG-RAG as a potential solution to reduce and eradicate misinformation by exploiting the structured information in Knowledge Graphs to increase the LLMs performance on multi-hop domain-specific questions.

Acknowledgements

This research was partially funded by The Scientific and Technological Research Council of Turkey (TUBITAK)'s Industrial PhD program under project number 118C056.

References

AI@Meta. 2024. Llama 3 model card.

Ahmet Yusuf Alan, Enis Karaarslan, and Omer Aydin. 2024. A rag-based question answering system proposal for understanding islam: Mufassirqas llm. *arXiv preprint arXiv:2401.15378*.

Valentina Alto. 2024. Introducing graphrag with langchain and neo4j. Accessed 25/06/2024.

Tomaž Bratanič. 2024. Using a knowledge graph to implement a rag application. Accessed 25/06/2024.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2024. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. 2018. Cypher: An evolving query language for property graphs. In *Proceedings of the 2018 international conference on management of data*, pages 1433–1445.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2023. Large language models on graphs: A comprehensive survey. *arXiv preprint arXiv:2312.02783*.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Suhang Wang, Yu Meng, and Jiawei Han. 2024. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. *arXiv preprint arXiv:2404.07103*.

Andreas Kollegger. 2024. Knowledge graphs for rag. https://www.deeplearning.ai/short-courses/knowledge-graphs-rag/. Accessed 27/06/2024.

Jonathan Larson and Steven Truitt. 2024. Graphrag: Unlocking llm discovery on narrative private data. Accessed 25/06/2024.

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

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. *arXiv preprint arXiv:2401.03205*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Spurthi Setty, Katherine Jijo, Eden Chung, and Natan Vidra. 2024. Improving retrieval for rag based question answering models on financial documents. *arXiv preprint arXiv:2404.07221*.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*.

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 preprint arXiv:2307.09288*.

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

Cyril Zakka, Rohan Shad, Akash Chaurasia, Alex R Dalal, Jennifer L Kim, Michael Moor, Robyn Fong, Curran Phillips, Kevin Alexander, Euan Ashley, et al. 2024. Almanac—retrieval-augmented language models for clinical medicine. *NEJM AI*, 1(2):AIoa2300068.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *AAAI*.