Lab Meetings

GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning.
C Mavromatis, G Karypis. University of Minnesota.

CAU
Junseo, Yu

DMAIS Lab Meeting
1.22.2025

Contents

Weekly Meetings

1. Introduction

- RAG
- Graph RAG
- Existing Methods

Introduction

RAG

Why Do we need RAG?

- **LLM**
  - LLMs are the **SOTA** in many NLP tasks.
  - However, LLMs **cannot easily adapt** to new or in-domain knowledge.
    - Because pretraining process is costly and time-consuming.
  - Moreover, LLM prone to **hallucinations**.

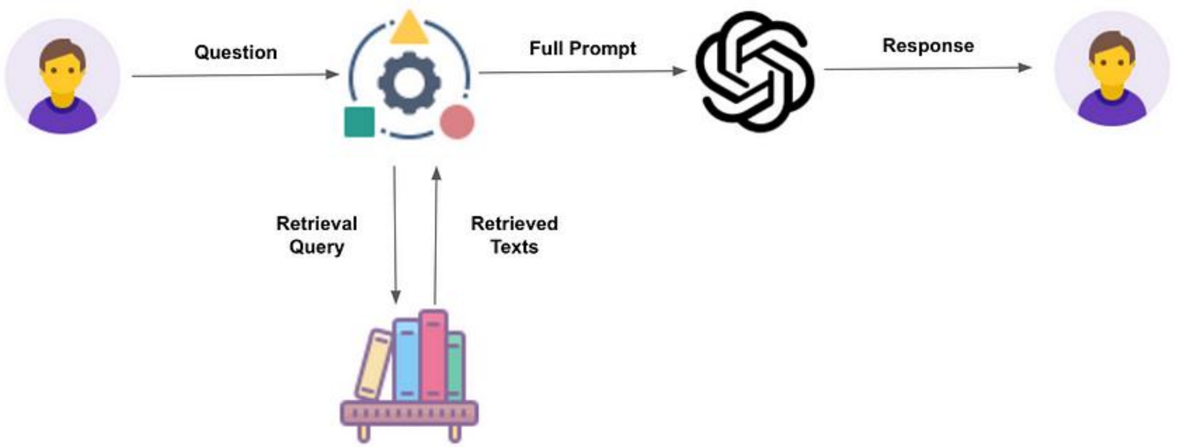
Introduction

RAG

Why Do we need RAG?

- **RAG, Retrieval-augmented generation**
  - RAG RAG retrieves relevant external information.
  - RAG can alleviate LLM hallucinations by enriching the input context with **accurate information**
    - **E.g.,** *Knowledge from RAG: Jamaica → language_spoken → English*

      Question: Which language do Jamaican people speak?

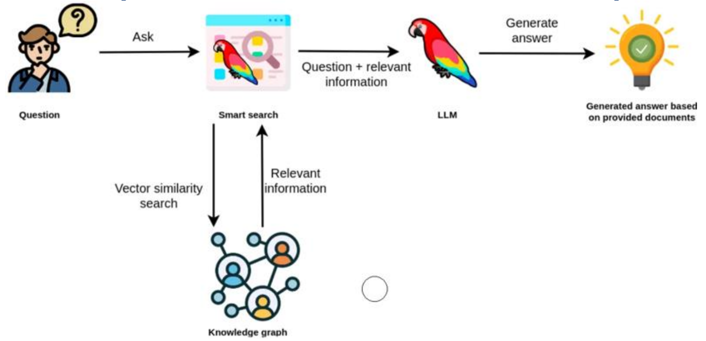


Introduction

Graph RAG

Why Do we need Graph RAG?

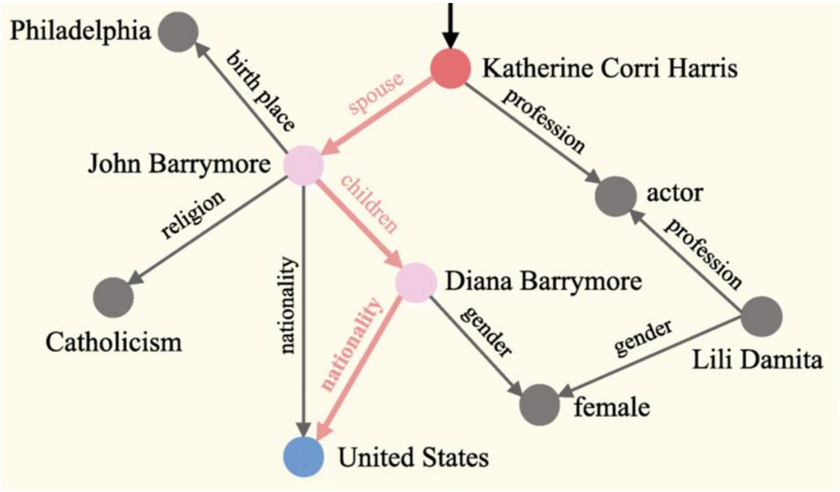
- Unlike the textual or visual data, It is beneficial to use **graph structure** to represent the **heterogeneous and relational information.**
  - E.g., KG(Knowledge Graph), Social Graph, and Document Graph
- Especially, KG is powerful resource to assist the LLM
- Retrieving the right information from graph requires distinctive graph processing.
  - Due to their diverse0formatted, interdependent, and domain-specific information.



Introduction

Graph RAG

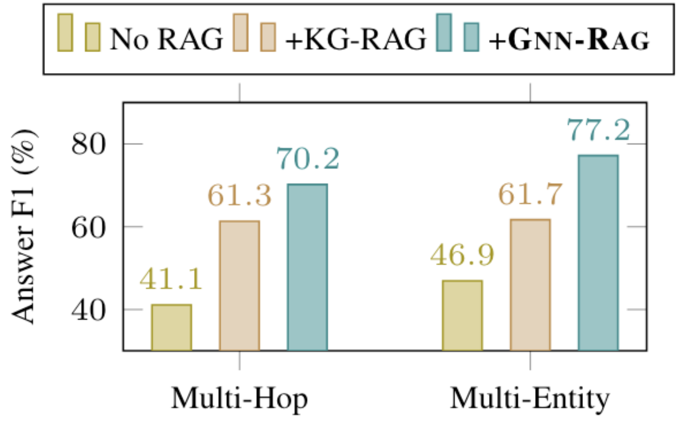
- KBQA, Knowledge Base Question Answering
  - Finding answers to questions expressed in natural language from a given knowledge base
  - ***E.g., Which language do Jamaican people speak?***
- Multi-hop KGQA
  - require a multi-hop reasoning procedure
  - ***E.g., What is nationality of Katherine Corri Harris's couple's children***

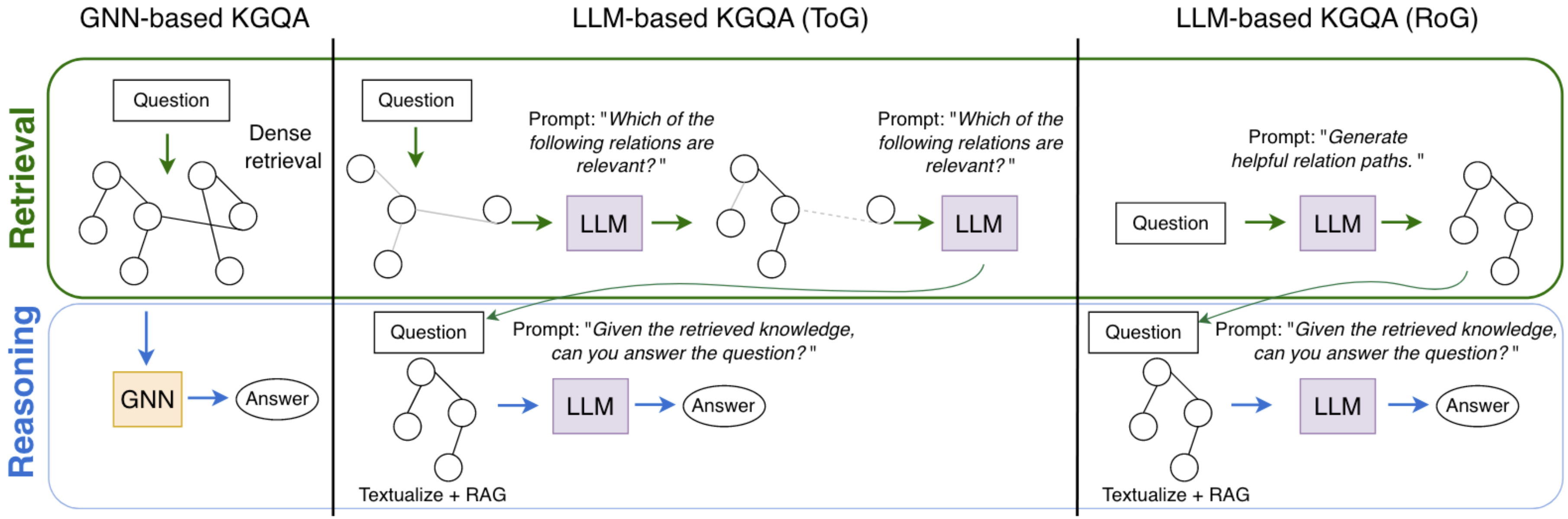


Introduction

Existing Methods

- ❑ LLM Based Graph RAG
  - Did not perform well in multi-hop KBQA.
- ❑ GNN Based Graph RAG
  - Can handle complex graph structure



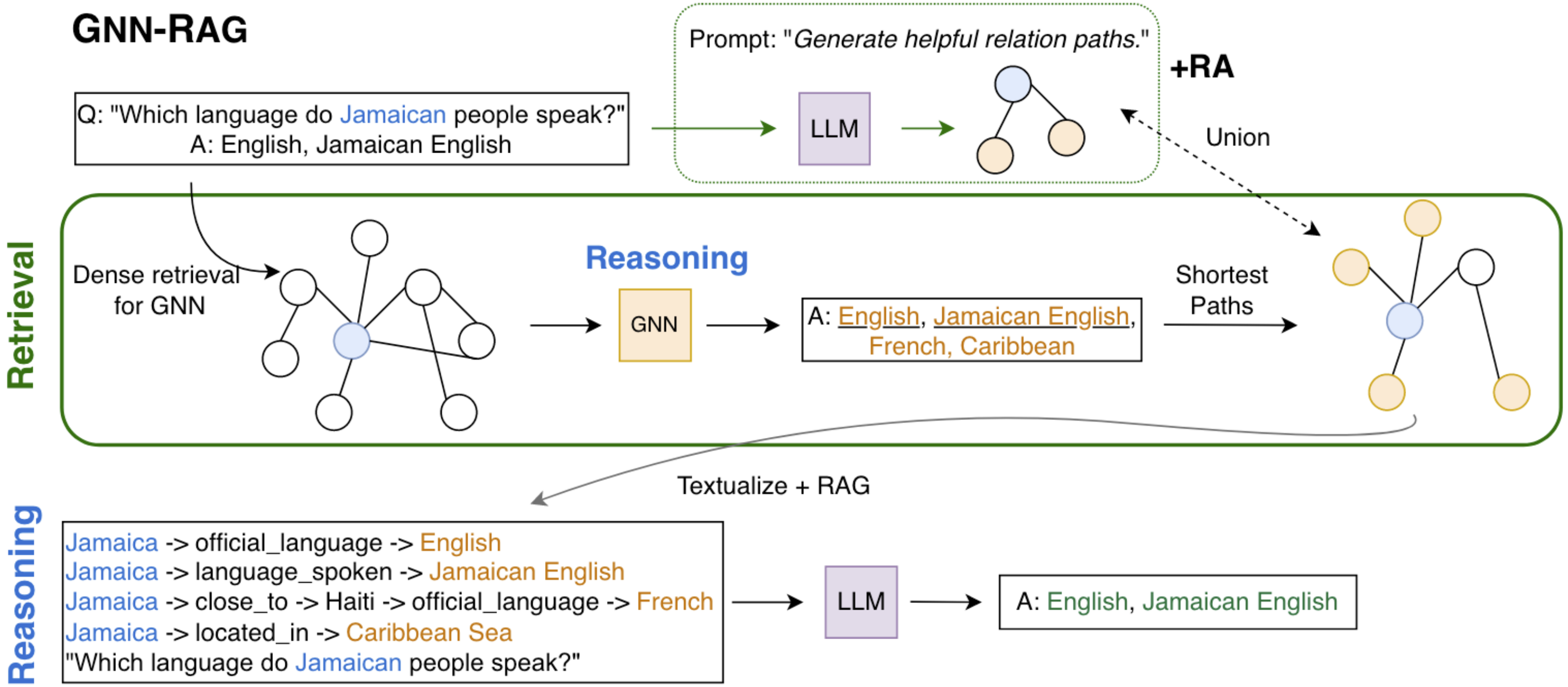


2. Methodology

- Overview
- Environment
- Detailed Methodology

Methodology

Overview



Overview

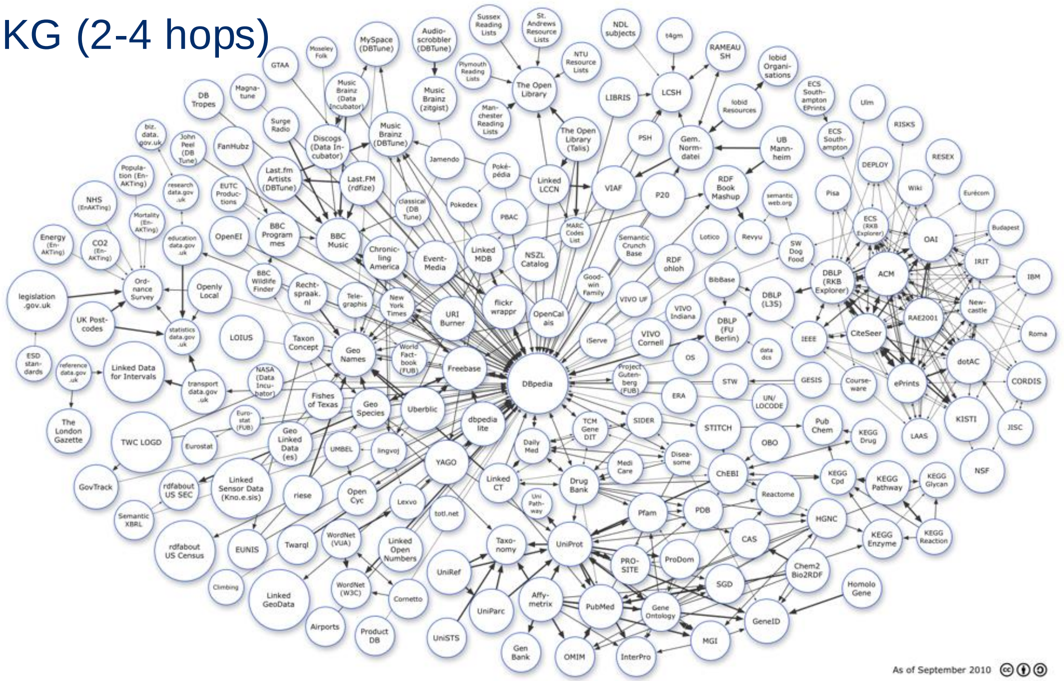
1. Retrieve the **subgraph**
2. Derive the candidate answer entities by **GNN**
3. Union with the other candidate answer entities by **LLM**
4. Textualize the reasoning path and then feed them to the **LLM**

Methodology

Environments

Datasets

- **Question-answer pairs**
  - Not the ground-truth paths that lead to the answer
  - Answerable using a subset of specific KG
  - The questions require **multiple-hops of reasoning** over the KG (2-4 hops)
- **Knowledge Graphs**
  - Freebase KG [Bollacker et al, 2008]



Methodology

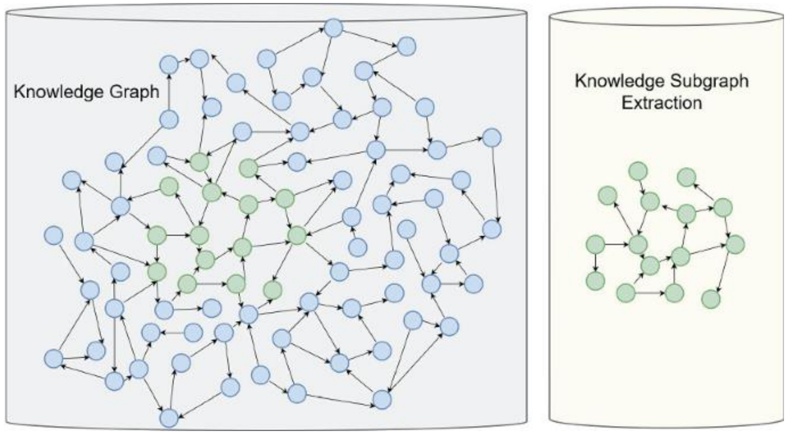
Detailed Methodology

Retrieve the subgraph

- ❑ **Linked Entities**
  - **Entity Recognition:** Identify and extract relevant entity from text
  - **Entity Linking:** Connect identified entities in text with their corresponding entities in KG
  - **Lexical Matching & Disambiguation:** Comparing text with entities name and resolve ambiguities
- ❑ **PageRank algorithm**
  - **PageRank-Nibble:** To identify the important entities from topic entities

Caution

- ❑ The correct answer may not exist in the subgraph.
- ❑ This method could be changed (Option)

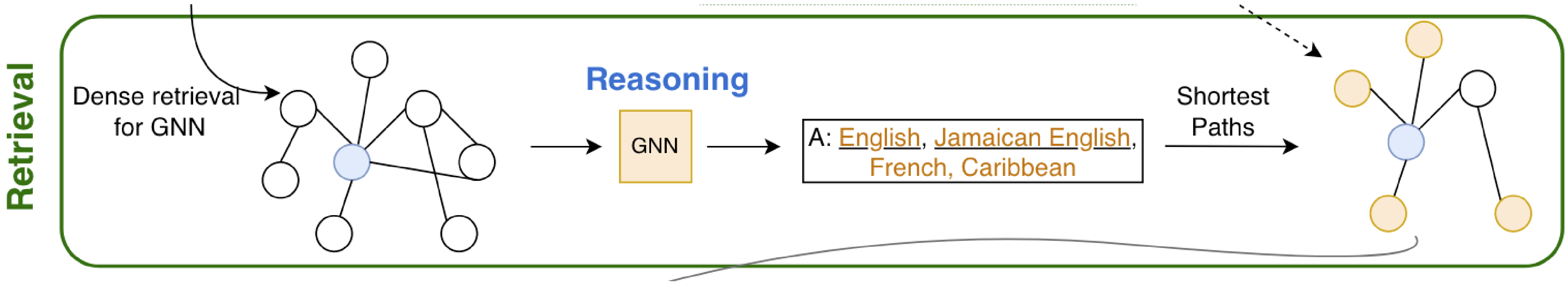


Methodology

Detailed Methodology

Derive the candidate answer entities by GNN - What

- Define the problem as **node classification** problem
  - All nodes in the subgraph are **scored** as answers vs non-answers based on their final GNN representations
  - The nodes above a probability threshold are returned as candidate answers along with the shortest paths
    - They are used as in put for LLM-based RAG (next step)



Methodology

Detailed Methodology

Derive the candidate answer entities by GNN - How

- **h_v**: the representation of node v
- **ω(q, r):** Measure how relevant the relation is to the question.
  - GNN reasoning depends on the question-relation matching operation ω(q, r).
  - **A common implementation:** $\phi(\boldsymbol{q}^{(k)} \odot \boldsymbol{r})$ $\boldsymbol{q}^{(k)} = \gamma_k\big(\mathrm{LM}(q)\big), \;\; \boldsymbol{r} = \gamma_c\big(\mathrm{LM}(r)\big),$
  - **The choice of LM** plays an **important role** regarding which answer nodes are retrieved.
    - It depends on how the relationship between the question and the relation is viewed.
    - Nevertheless, the performance was good regardless of the model used.

$$\boldsymbol{h}_v^{(l)} = \psi\Big(\boldsymbol{h}_v^{(l-1)}, \sum_{v' \in \mathcal{N}_v} \omega(q, r) \cdot \boldsymbol{m}_{vv'}^{(l)}\Big),$$

Methodology

Detailed Methodology

Derive the candidate answer entities by GNN - Why

- **Experimental Evidence**
  - **Answer Coverage:** whether the retriever is able to fetch at least one correct answer for RAG
  - RoG: the LLM based retriever
- Conclusion
  - GNN based retriever can retrieve useful multi-hop information more effectively
  - On the other hand, the LLM based retriever is better at 1-hop questions.
    - The authors explain this situation as accurate question-relation matching is more important than deep graph search

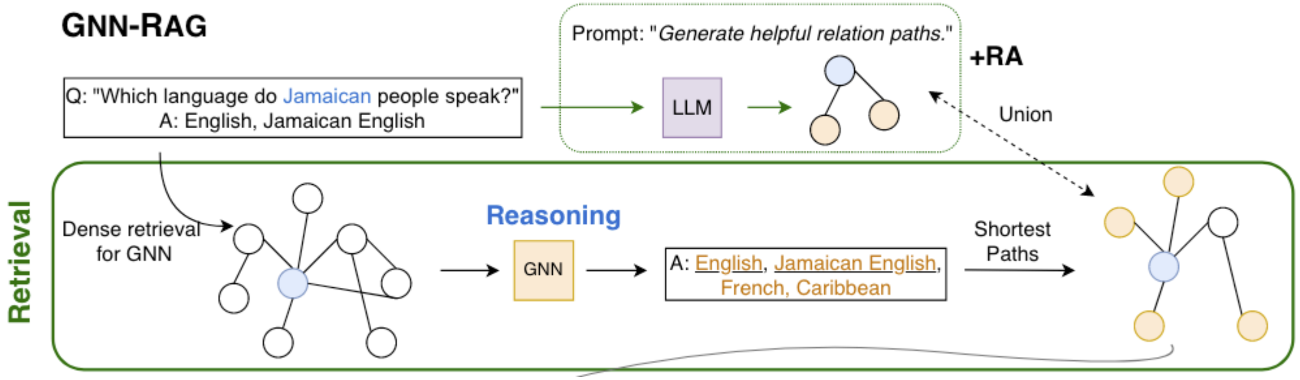
Retriever	**1-hop questions**		**2-hop questions**	
	#Input Tok.	%Ans. Cov.	#Input Tok.	%Ans. Cov.
RoG [Luo et al., 2024]	150	**87.1**	435	82.1
GNN ($L = 1$)	112	83.6	2,582	79.8
GNN ($L = 3$)	105	82.4	357	**88.5**

Methodology

Detailed Methodology

Union with the other candidate answer entities by LLM

- **Retrieval augmentation (RA)**
  - Combines the retrieved KG information from different approaches to increase **diversity**
    - Complements the GNN retriever with an LLM-based retriever to combine **their strengths**
  - Experiment with the RoG retrieval
    - Take the union of the reasoning paths retrieved by the two retrievers.
    - A **downside** of **LLM-based** retrieval: Requires multiple generations (beam-search decoding) to retrieve diverse paths



Methodology

Detailed Methodology

Union with the other candidate answer entities by LLM

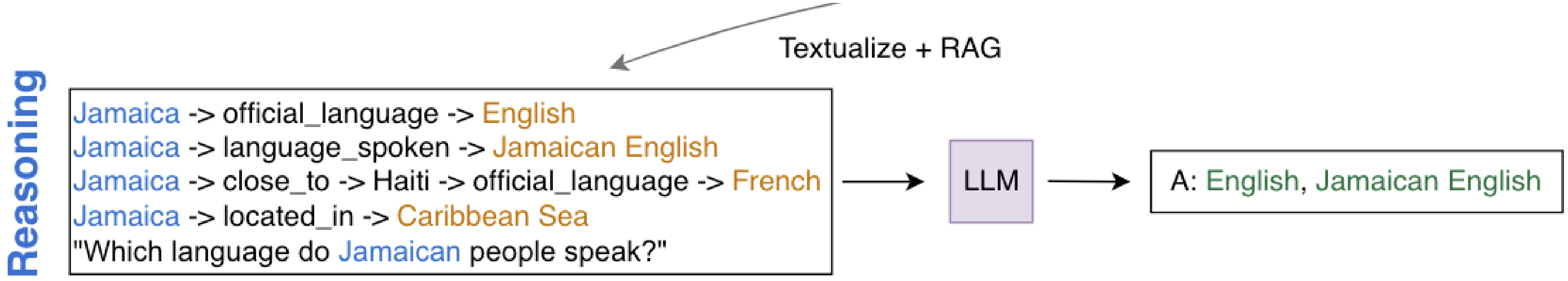
- **Cheaper Alternative**
  - By combining the outputs of different GNNs, which are equipped with different LMs in below equation.
    → **GNN-RAG+Ensemble**
  - The union of the retrieved paths of the two different GNNs as input for RAG.
    - GNN with **SBERT (Sentence-BERT)**
    - GNN with **$\mathbf{LM_{SR}}$ (LM for Structured Retrieval)**

Methodology

Detailed Methodology

Textualize the reasoning path and then feed them to the LLM

- ❑ **Verbalize** the obtained reasoning paths
- ❑ LLM model is fine-tuned based on the training question-answer pairs to generate correct answers
- ❑ **Prompts**
  - ***"{Reasoning Paths} \n Question: {Question}"***
    - The reasoning paths are verbalised as **"{question entity}** → **{relation}** → **{entity}** → **…** → **{relation}** → **{answer entity} \n"**



Weekly Meetings

3. Experiments

- Setup
- Results

Experiments

Setup

- **Datasets**
  - WebQuestionsSP (WebQSP)
  - Complex WebQuestions 1.1 (CWQ)
- **Implementation**
  - GNN-RAG model: ReaRev (SOTA)
  - LM making embeddings: SBERT and LM_{SR}
  - Prompt Tunning: RoG for RAG-based prompt tunning
- **Metrics**
  - Hit: If any of the true answers is found in the generated response
  - H@1
  - F1

Experiments

Results

Type	Method	WebQSP			CWQ		
		Hit	H@1	F1	Hit	H@1	F1
Embedding	KV-Mem [Miller et al. [2016]]	–	46.7	38.6	–	21.1	–
	EmbedKGQA [Saxena et al. [2020]]	–	66.6	–	–	–	–
	TransferNet [Shi et al. [2021]]	–	71.4	–	–	48.6	–
	Rigel [Sen et al. [2021]]	–	73.3	–	–	48.7	–
GNN	GraftNet [Sun et al. [2018]]	–	66.7	62.4	–	36.8	32.7
	PullNet [Sun et al. [2019]]	–	68.1	–	–	45.9	–
	NSM [He et al. [2021]]	–	68.7	62.8	–	47.6	42.4
	SR+NSM(+E2E) [Zhang et al., 2022a]	–	69.5	64.1	–	50.2	47.1
	NSM+h [He et al. [2021]]	–	74.3	67.4	–	48.8	44.0
	SQALER [Atzeni et al. [2021]]	–	76.1	–	–	–	–
	UniKGQA [Jiang et al., 2023b]	–	77.2	72.2	–	51.2	49.1
	ReaRev [Mavromatis and Karypis, 2022]	–	76.4	70.9	–	52.9	47.8
	ReaRev + LM_{SR}	–	77.5	72.8	–	53.3	49.7
LLM	Flan-T5-xl [Chung et al., 2024]	31.0	–	–	14.7	–	–
	Alpaca-7B [Taori et al., 2023]	51.8	–	–	27.4	–	–
	LLaMA2-Chat-7B [Touvron et al., 2023]	64.4	–	–	34.6	–	–
	ChatGPT	66.8	–	–	39.9	–	–
	ChatGPT+CoT	75.6	–	–	48.9	–	–
KG+LLM	KD-CoT [Wang et al., 2023]	68.6	–	52.5	55.7	–	–
	StructGPT [Jiang et al., 2023a]	72.6	–	–	–	–	–
	KB-BINDER [Li et al., 2023]	74.4	–	–	–	–	–
	ToG+LLaMA2-70B [Sun et al., 2024]	68.9	–	–	57.6	–	–
	ToG+ChatGPT [Sun et al., 2024]	76.2	–	–	58.9	–	–
	ToG+GPT-4 [Sun et al., 2024]	82.6	–	–	**69.5**	–	–
	RoG [Luo et al., 2024]	85.7	80.0	70.8	62.6	57.8	56.2
GNN+LLM	G-Retriever [He et al., 2024]	–	70.1	–	–	–	–
	GNN-RAG (**Ours**)	85.7	80.6	71.3	66.8	61.7	59.4
	GNN-RAG+RA (**Ours**)	**90.7**	**82.8**	**73.5**	68.7	**62.8**	**60.4**

Method	WebQSP		CWQ	
	multi-hop	multi-entity	multi-hop	multi-entity
LLM (No RAG)	48.4	61.5	33.7	32.3
RoG	63.3	65.1	59.3	58.3
GNN-RAG	69.8	82.3	68.2	64.8
GNN-RAG+RA	71.1	88.8	69.3	65.6

- GNN-RAG achieve SOTA
- GNN-RAG is an effective retrieval method when deep graph search is important for successful KGQA.

Experiments

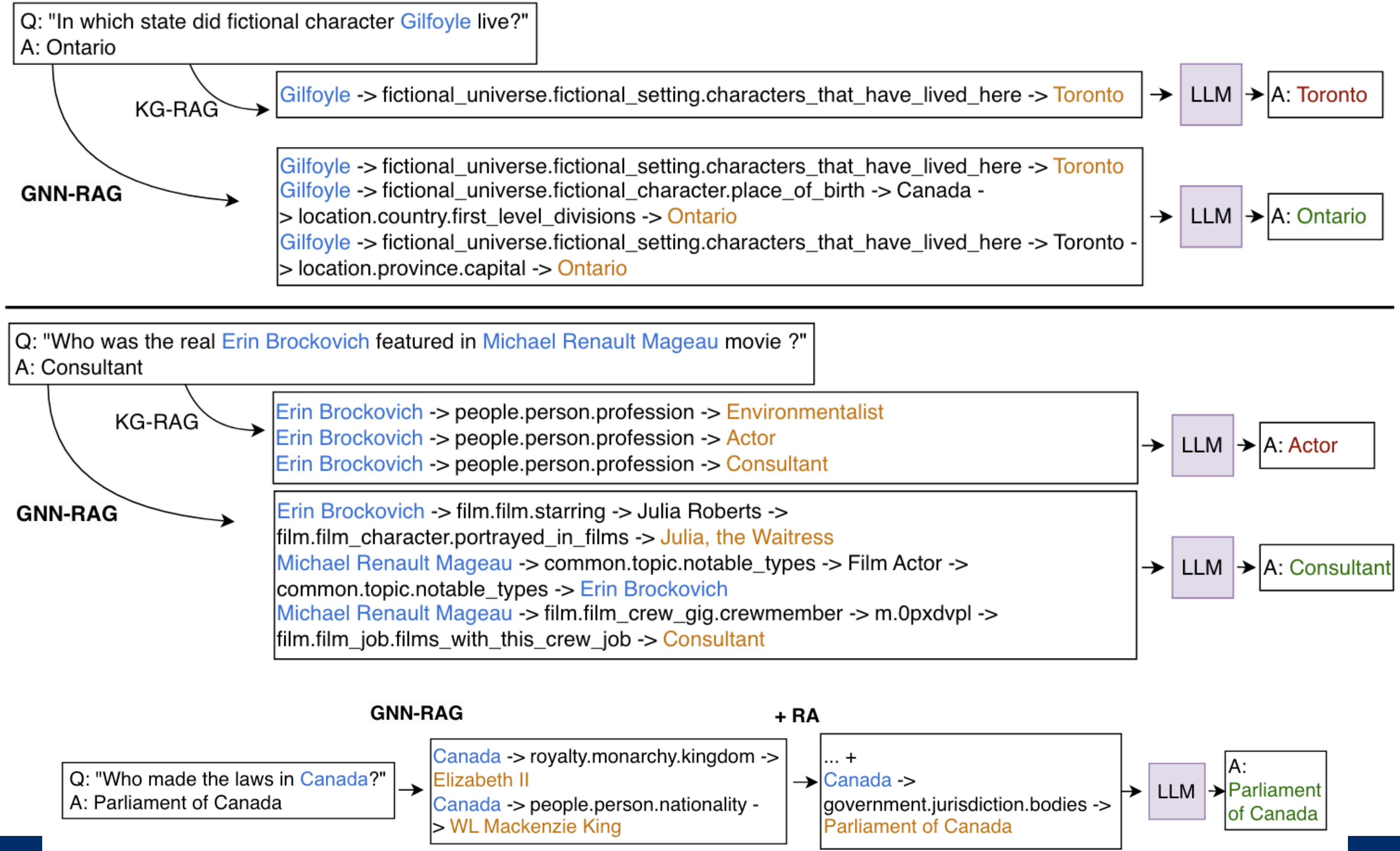
Results

Retriever	*KGQA Model*	*Input/Graph Statistics* #LLM Calls	#Input Tokens WebQSP / CWQ	*KGQA Performance* F1 (%) WebQSP / CWQ
a) Dense Subgraph	(i) GNN + SBERT (Eq. 3)	0	–	70.9 / 47.8
b) Dense Subgraph	(ii) GNN + LM_{SR} (Eq. 3)	0		72.8 / 49.1
c) None	LLaMA2-Chat-7B (tuned)	0	59 / 70	49.7 / 33.8
d) (iii) RoG (LLM-based; Eq. 2)		3	202 / 325	70.8 / 56.2
e) GNN-RAG (*default*): (i)		0	144 / 207	71.3 / 59.4
f) GNN-RAG: (ii)		0	124 / 206	71.5 / 58.9
g) GNN-RAG+Ensemble: (i) + (ii)	LLaMA2-Chat-7B (tuned)	0	156 / 281	71.7 / 57.5
h) GNN-RAG+RA (*default*): (i) + (iii)		3	299 / 540	**73.5** / 60.4
i) GNN-RAG+RA: (ii) + (iii)		3	267 / 532	73.4 / **61.0**
j) GNN-RAG+All: (i) + (ii) + (iii)		3	330 / 668	72.3 / 59.1

- GNN-based retrieval is more efficient and effective than LLM-based retrieval.
- Combining GNN-induced reasoning paths with LLM-induced reasoning paths is better.
- Augmenting all retrieval approaches does not necessarily cause improved performance

Experiments

Results



Weekly Meetings

4. Conclusion

- Contributions
- Future Directions

Conclusion

Contributions

- Effective Integration of GNN and LLM
- Achieving State-of-the-Art Performance on KGQA Benchmarks
- Introduction of Retrieval Augmentation

Future Directions

- Apply similar method into other fields not QA
- Assumption of a situation where the correct entity does not exist
- Risks of the shortest path assumption