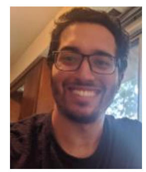
Retrieval-augmented Generation on Graph-structured Data


Yu Wang[1]

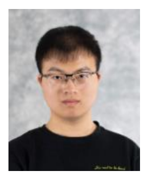

Haoyu Han[2]


Harry Shomer[2]


Kai Guo[2]


Yongjia Lei[1]


Jiayuan Ding[4]


Xianfeng Tang[3]


Qi He[3]


Jiliang Tang[2]

University of Oregon[1]
Michigan State University[2]
Amazon[3]
Hippocratic AI[4]

SDM25-GraphRAG

Addressing real-world tasks desire knowledge

Just can't remember……

What are they talking about?

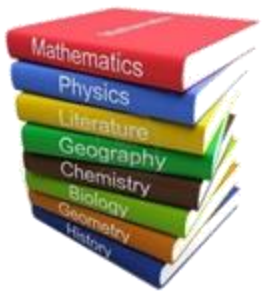
What should I look next?

Query/Task Request

Missing Knowledge!

Real-world knowledge is so much!

Textbook Knowledge Base



158 million books

ISBN DB 2023

Internet Knowledge Base

1.1 billion websites

Musemind 2024

Neural Knowledge Base

405 billion parameters

Hugging Face 2024

2.5 petabytes, 1 billion books

- **We remember meanings, not details.**
- **We forget on purpose.**
- **Tiny active memory, Larger long-term memory.**

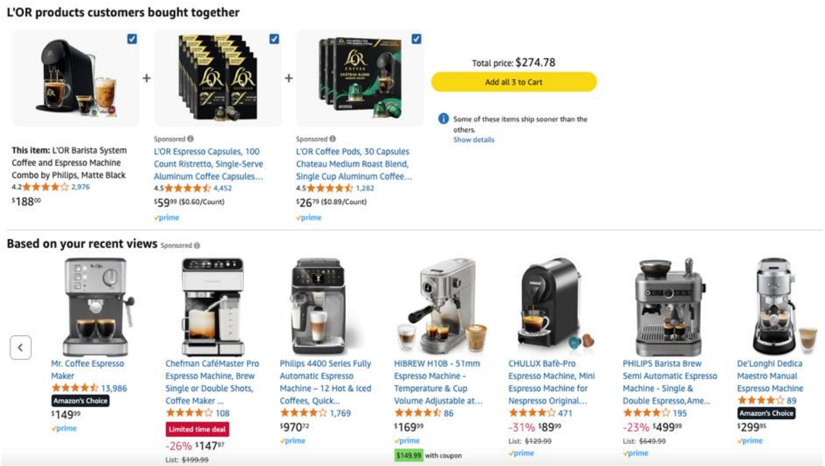
Retrieval Knowledge to Augment Downstream Task is Rather Important!

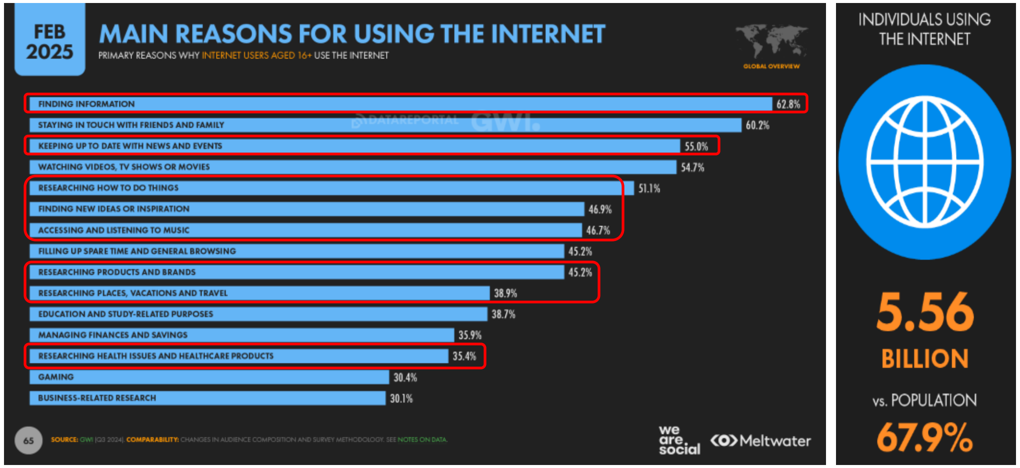
Retrieving External Knowledge

Open-book Exam

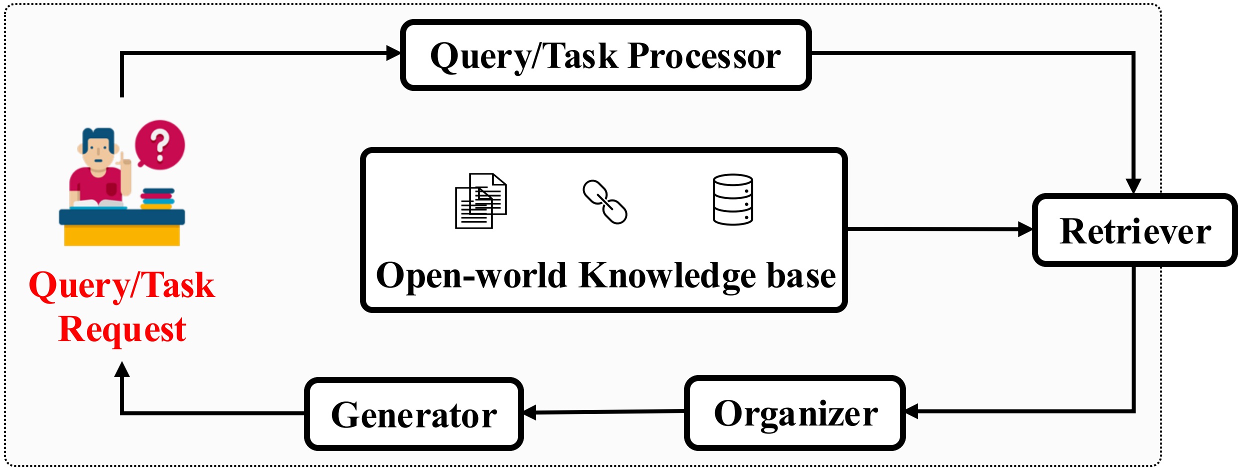
Google Search

Online Shopping

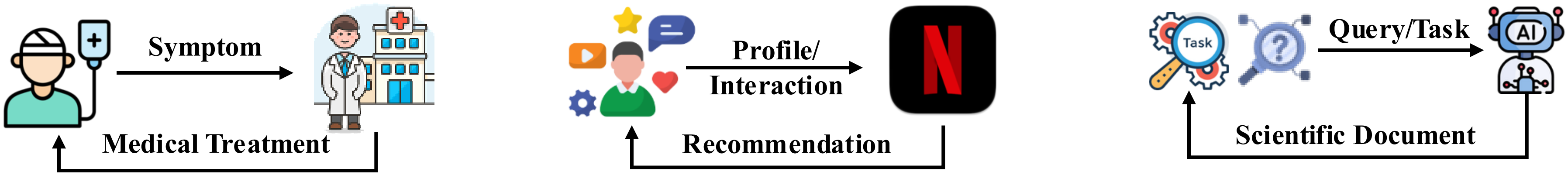




Retrieval-augmented Generation (RAG)



Query Q



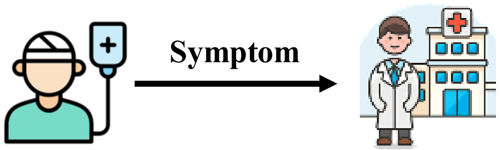
Any idea why I might be sick?

Can you recommend a mouse repellent that has a nice smell?

Find me papers that discuss improving condensers performance

Retrieval-augmented Generation (RAG)

Really tired.
Temperature is over 100.
Recently in France.
Drank a lot of tap water there.
Any idea why I might be sick?

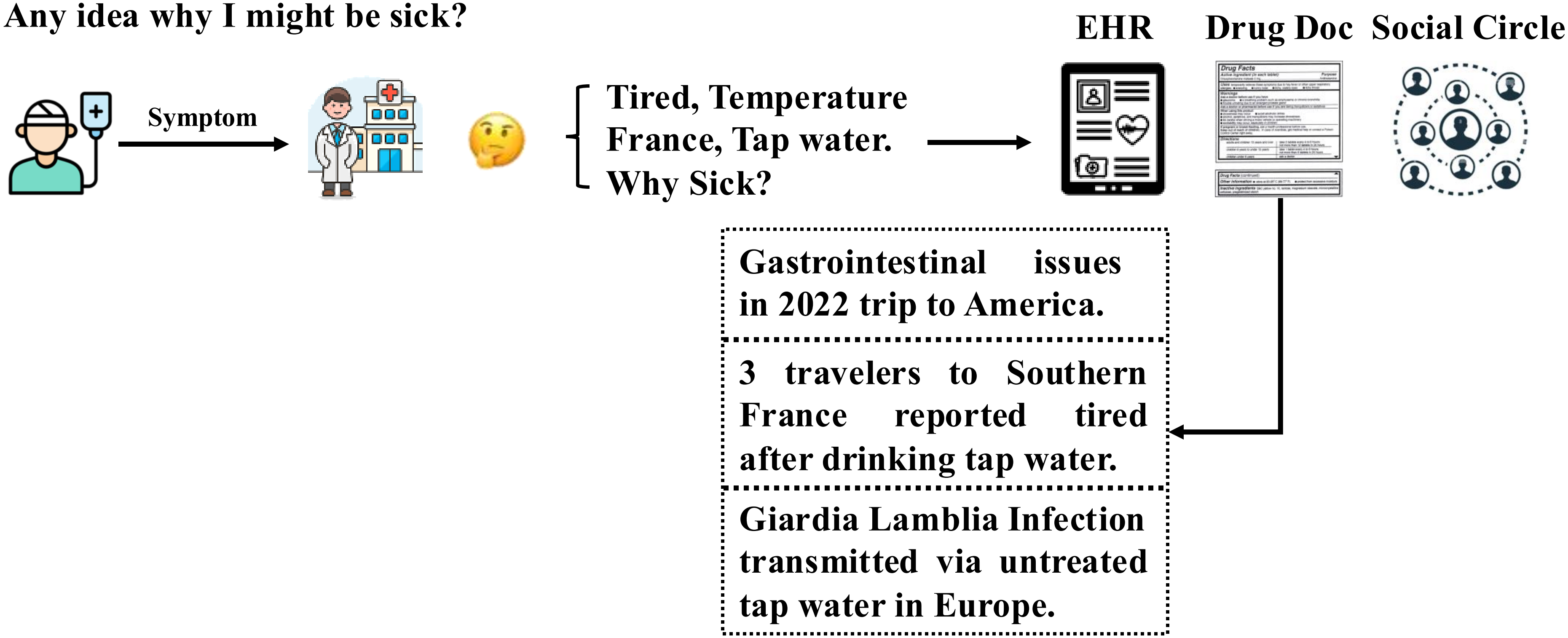


Tired, Temperature
France, Tap water.
Why Sick?

$$\hat{Q} = \mathbf{\Omega}^{\textbf{Processer}}(Q)$$

Retrieval-augmented Generation (RAG)

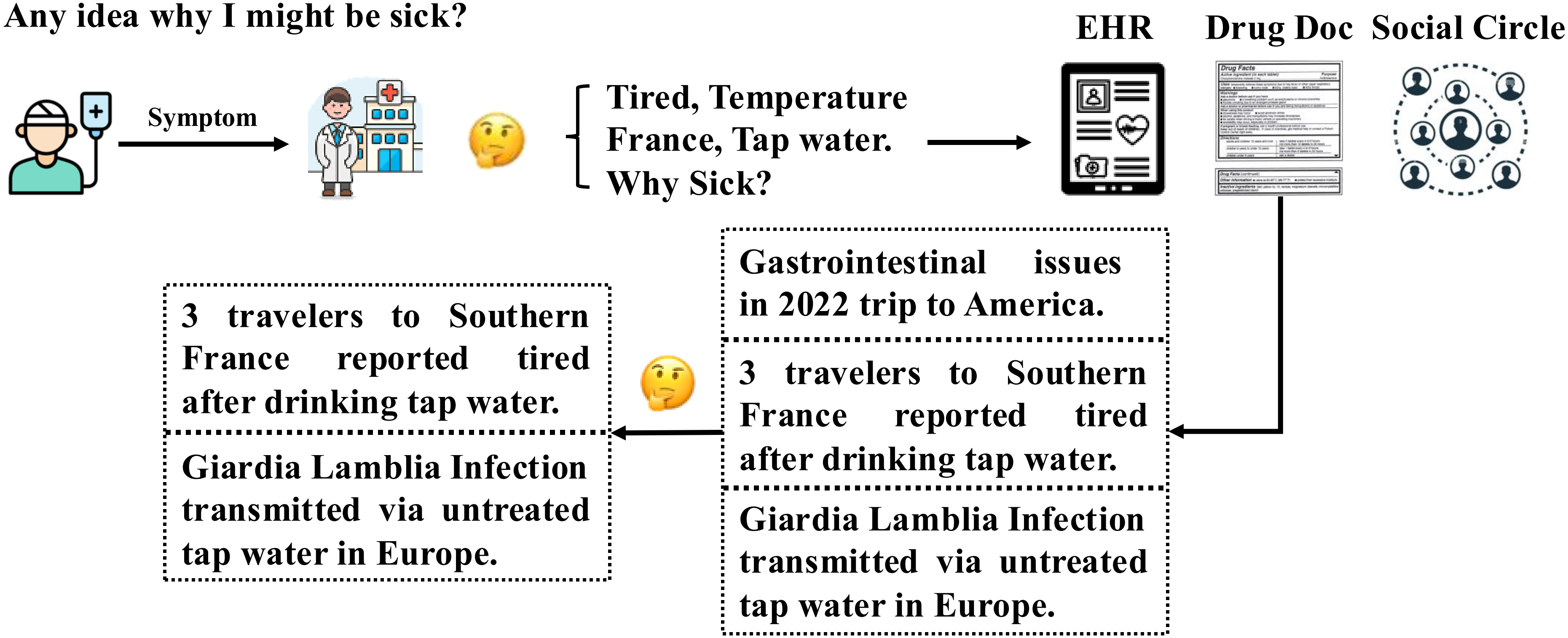
Really tired.
Temperature is over 100.
Recently in France.
Drank a lot of tap water there.
Any idea why I might be sick?



$$\hat{C} = \mathbf{\Omega}^{\text{Retriever}}(\hat{Q}, C)$$

Retrieval-augmented Generation (RAG)

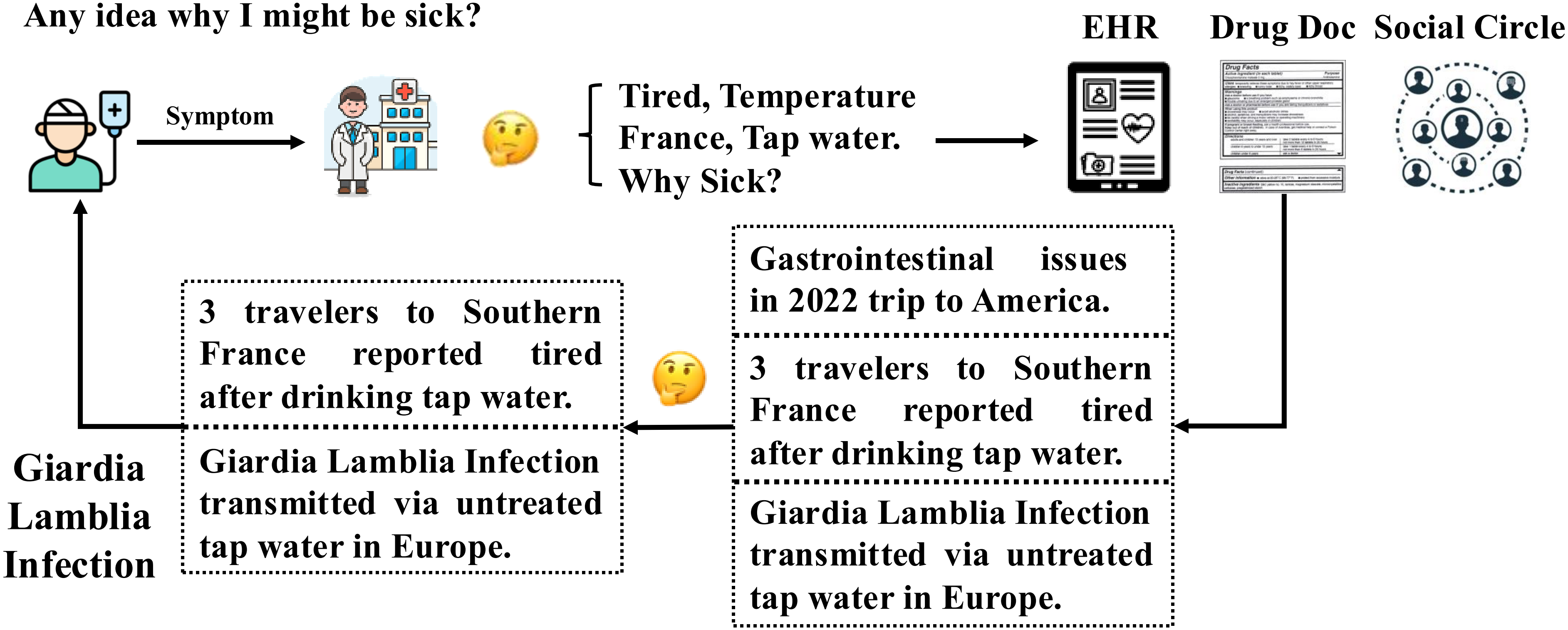
Really tired.
Temperature is over 100.
Recently in France.
Drank a lot of tap water there.
Any idea why I might be sick?



$$\hat{C} = \boldsymbol{\Omega}^{\mathbf{Organizer}}(\hat{Q}, C)$$

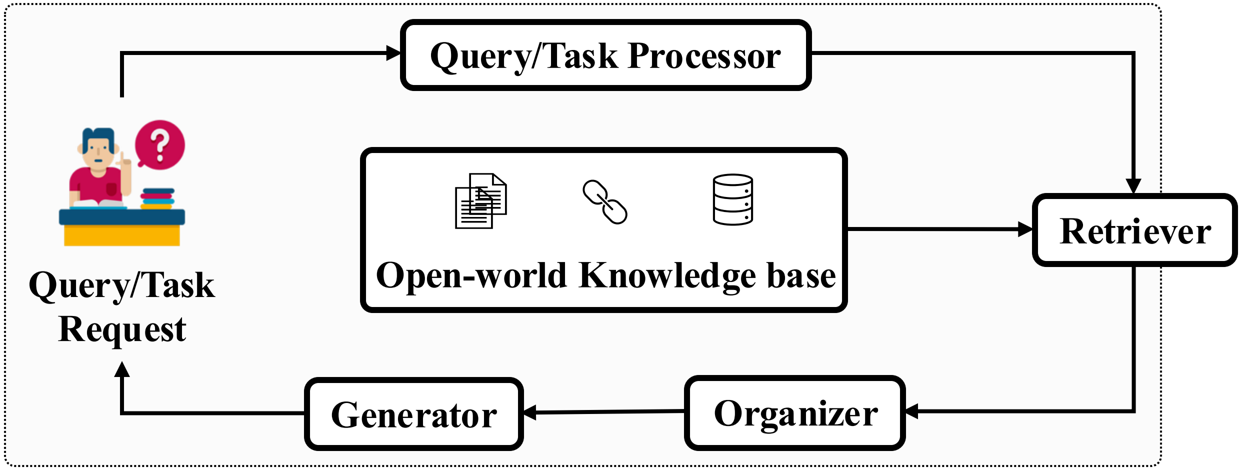
Retrieval-augmented Generation (RAG)

Really tired.
Temperature is over 100.
Recently in France.
Drank a lot of tap water there.
Any idea why I might be sick?



$$A = \mathbf{\Omega}^{\text{Generator}}(\hat{Q}, \hat{C})$$

Retrieval-augmented Generation (RAG)



(1) **Query** Q

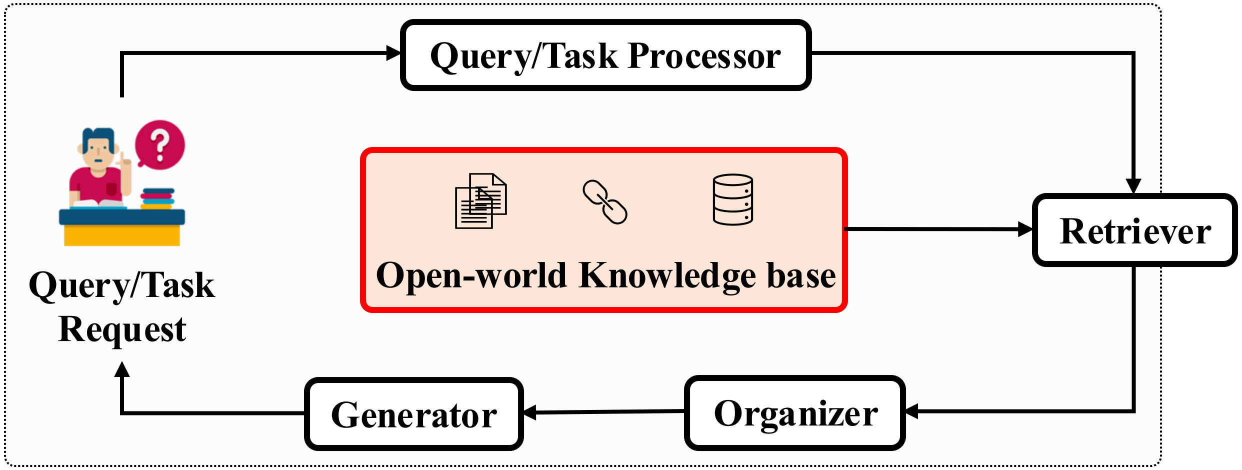
(2) $\hat{Q} = \boldsymbol{\Omega}^{\textbf{Processer}}(Q)$

(4) $C = \boldsymbol{\Omega}^{\textbf{Retriever}}(\hat{Q}, G)$

(5) $\hat{C} = \boldsymbol{\Omega}^{\textbf{Organizer}}(\hat{Q}, C)$

(6) $A = \boldsymbol{\Omega}^{\textbf{Generator}}(\hat{Q}, \hat{C})$

Retrieval-augmented Generation (RAG)



(1) **Query** Q

(2) $\hat{Q} = \mathbf{\Omega}^{\text{Processer}}(Q)$

(4) $C = \mathbf{\Omega}^{\text{Retriever}}(\hat{Q}, G)$

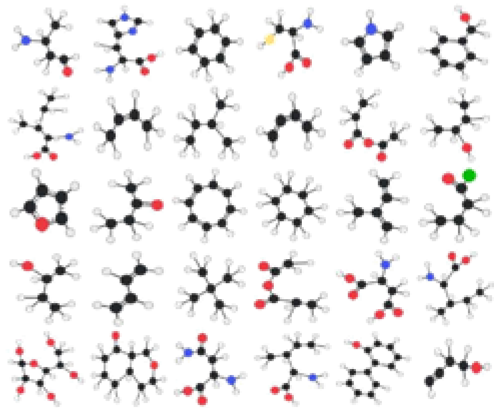
(5) $\hat{C} = \mathbf{\Omega}^{\text{Organizer}}(\hat{Q}, C)$

(6) $A = \mathbf{\Omega}^{\text{Generator}}(\hat{Q}, \hat{C})$

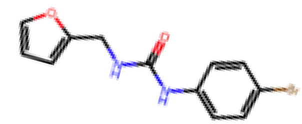
Real-world knowledge can be extremely complex and heterogeneous!

Retrieval-augmented Generation (RAG) – Drug Design

Optimizing the binding affinity ≤ -4.9 kcal/mol

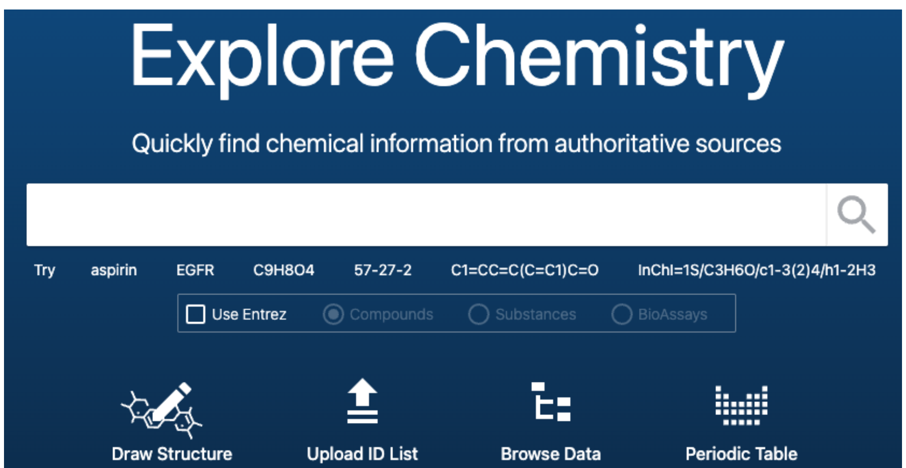


Compounds Knowledge Base



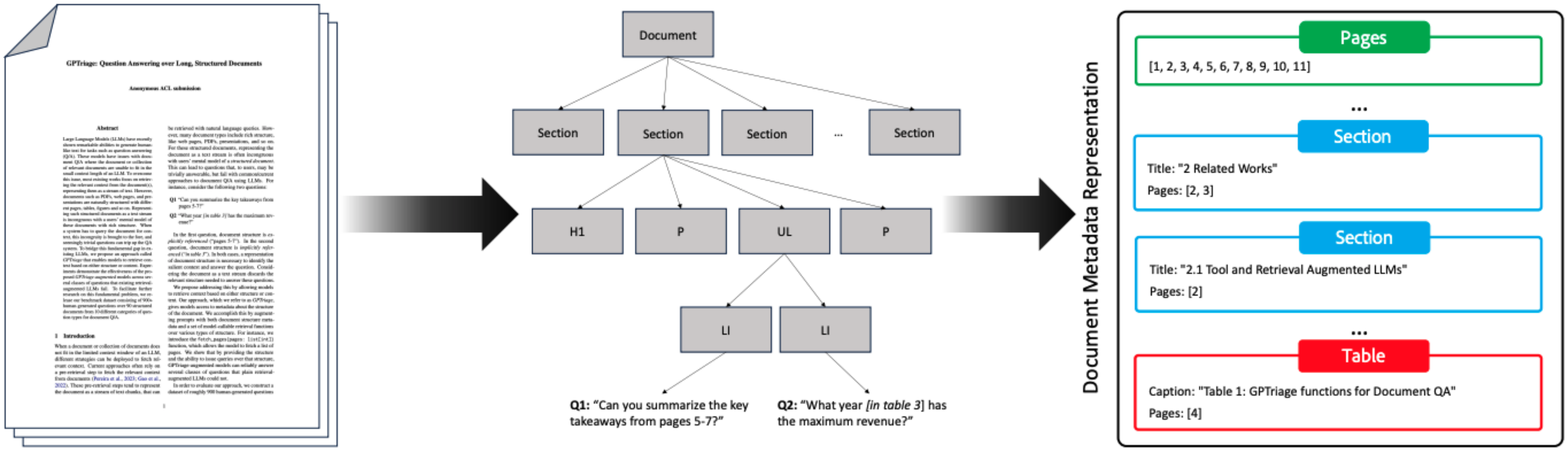
Chemical Property Depends on 3D structures

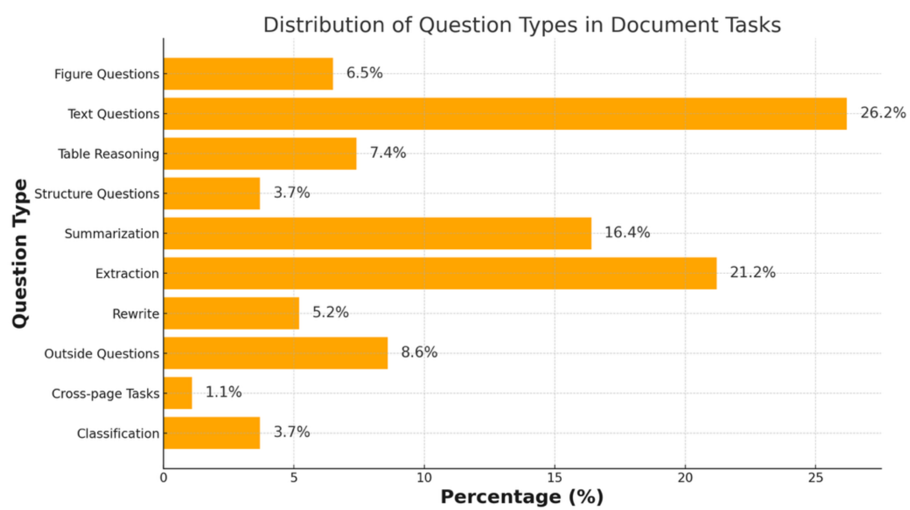
PubChem



- **119M** Compounds
- **329M** Substances
- **297M** Bioactivities
- **42M** Literature
- **54M** Patents pubchem

Retrieval-augmented Generation (RAG) – Document

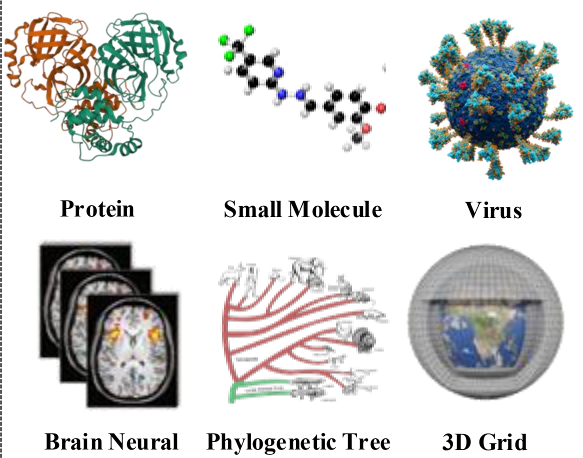




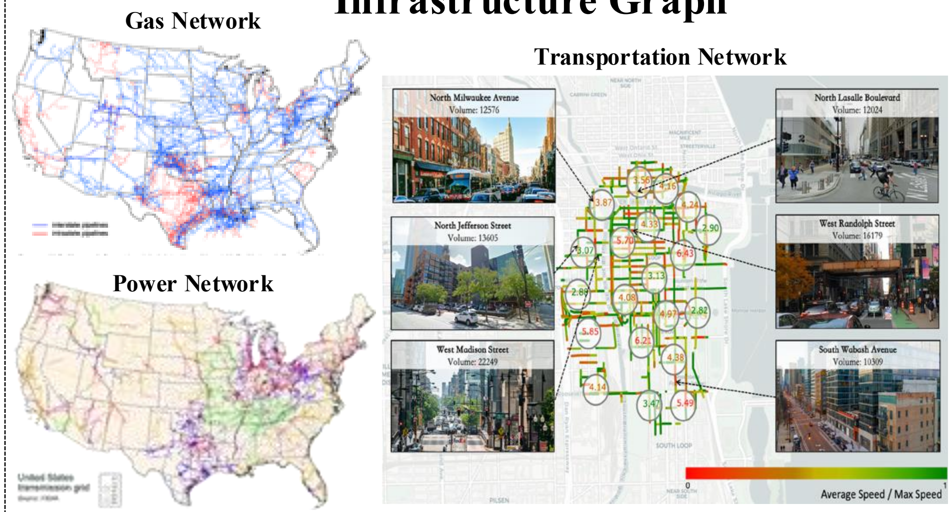
PDFTriage: Question Answering over Long, Structured Documents. EMNLP 2024

Heterogeneous knowledge can be represented as Graph

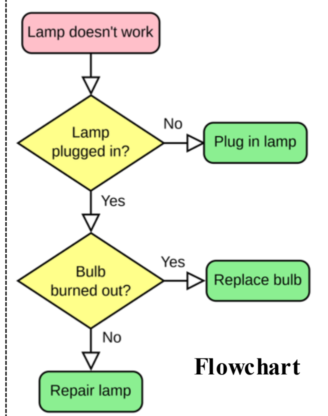
Scientific Graph



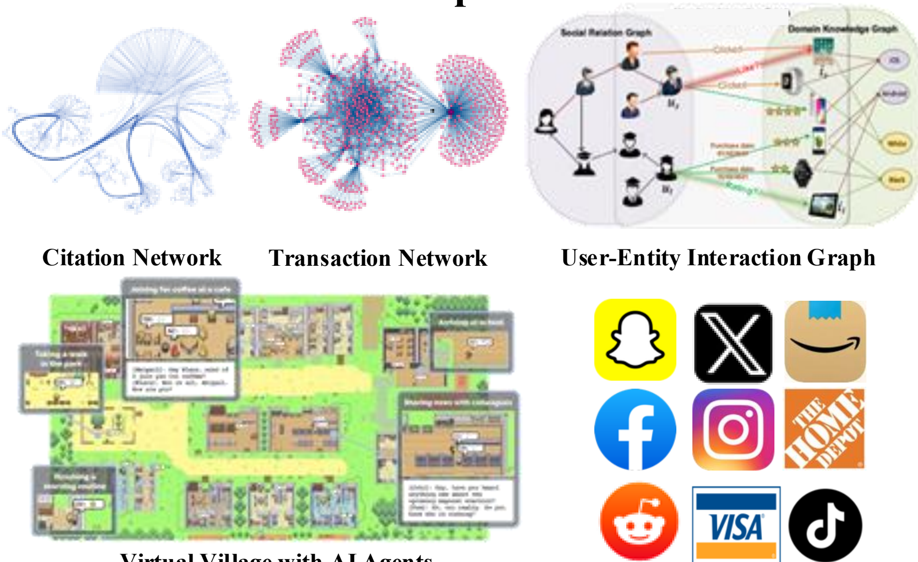
Infrastructure Graph



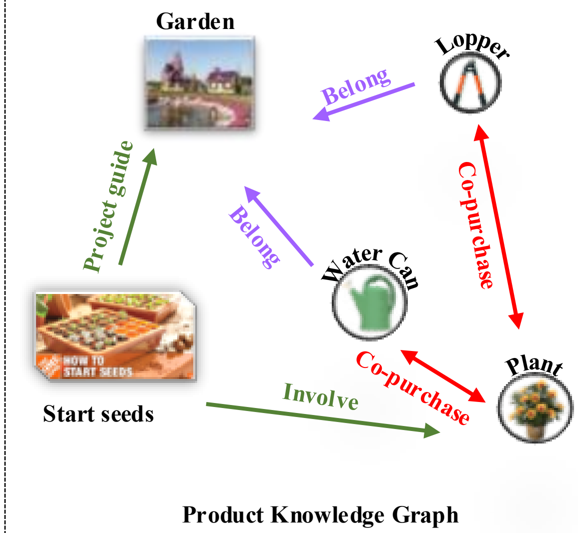
Reasoning/Planning Graph



Social Interaction Graph

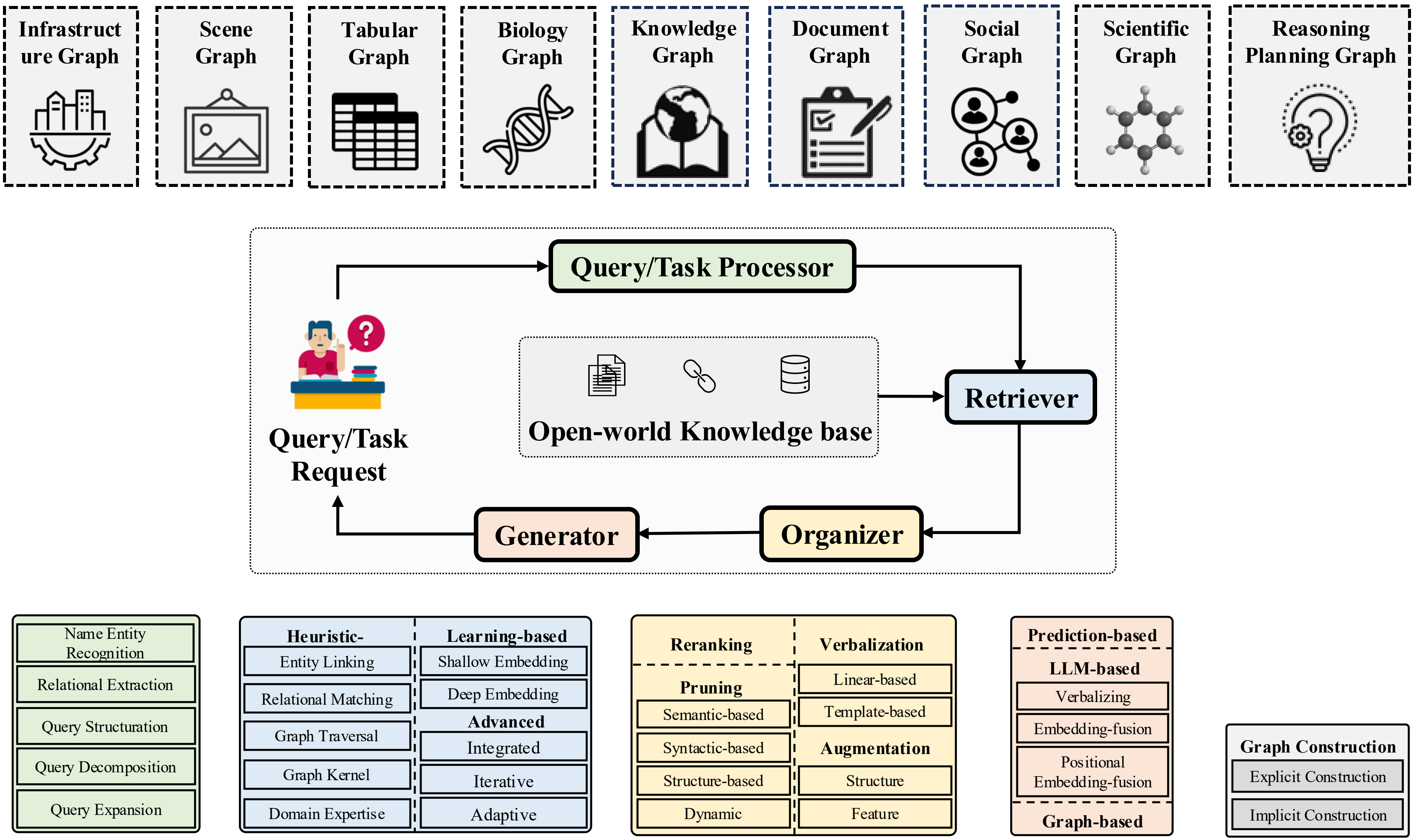


Knowledge Graph

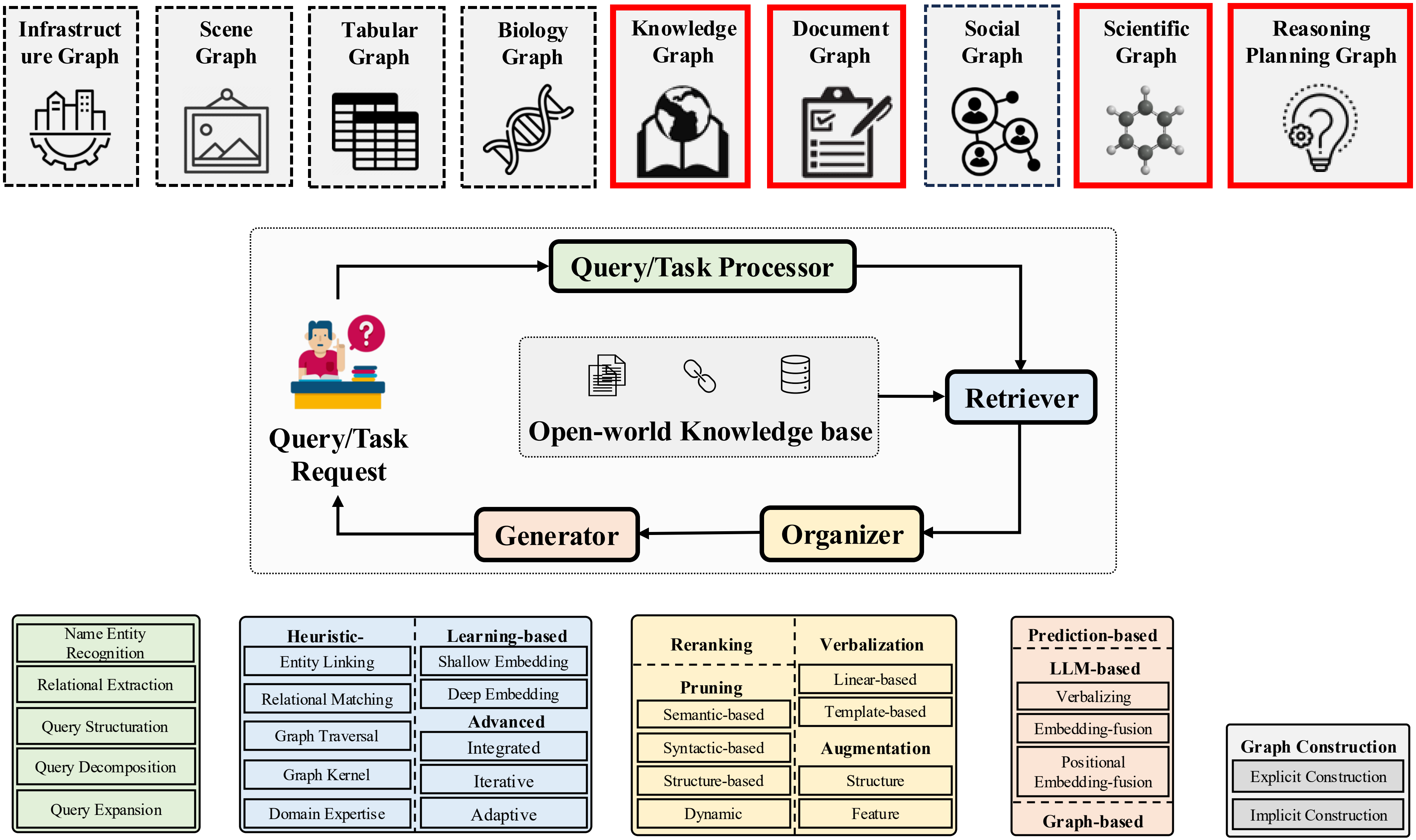


Document Graph

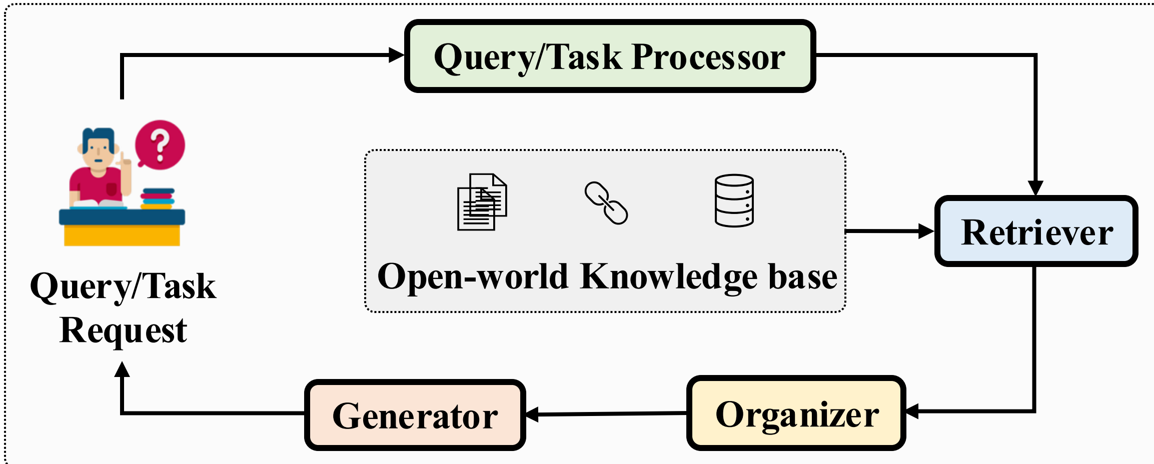
Graph Retrieval-augmented Generation (GraphRAG)

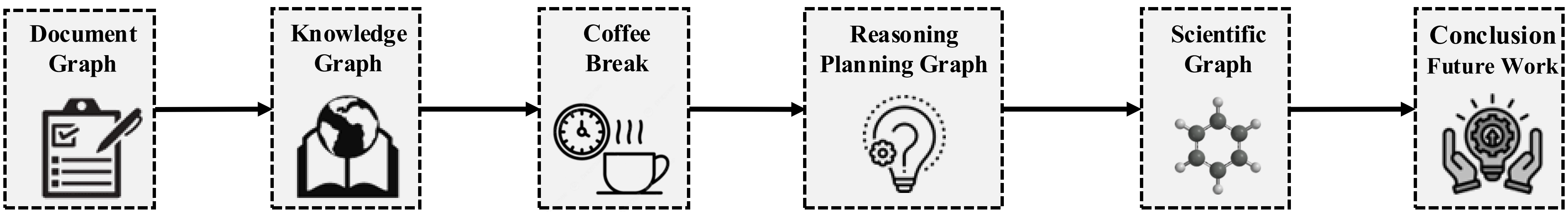


Graph Retrieval-augmented Generation (GraphRAG)

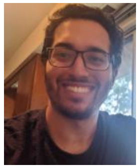


Outline





Haoyu Han
24 min



Harry Shomer
24 min

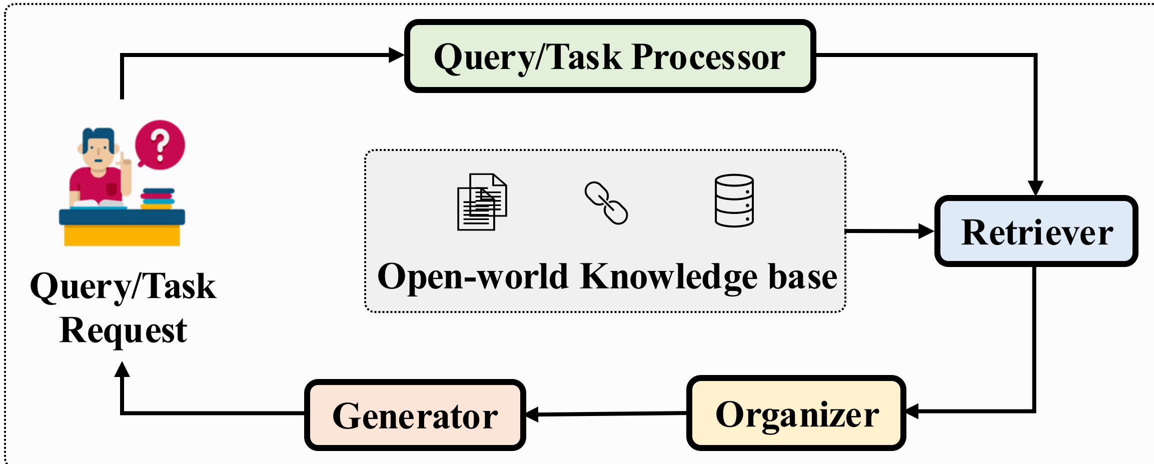
4 min

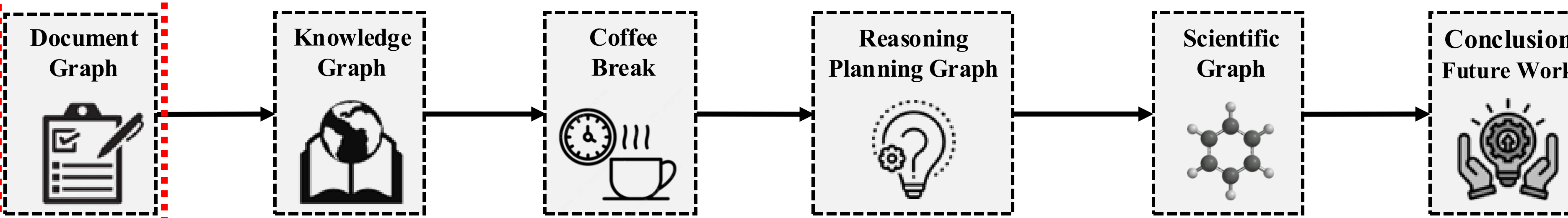
Yongjia Lei
24 min

Kai Guo
24 min

Yu Wang
5 min

Outline





Haoyu Han
24 min

Harry Shomer
24 min

4 min

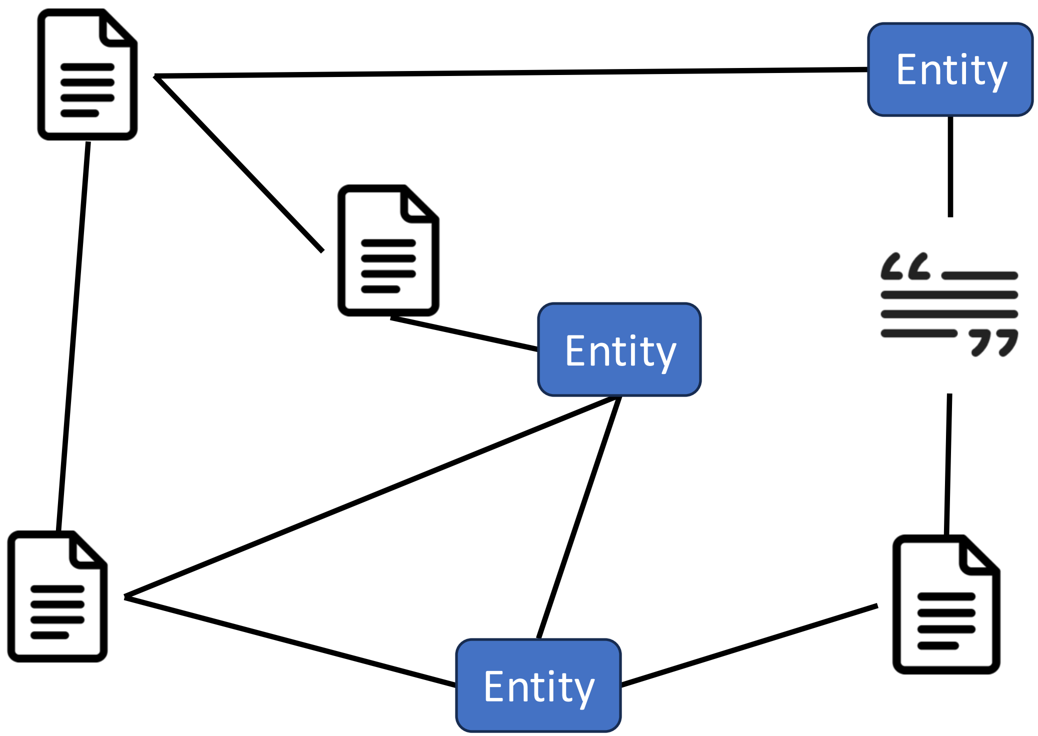
Yongjia Lei
24 min

Kai Guo
24 min

Yu Wang
5 min

Document Graph

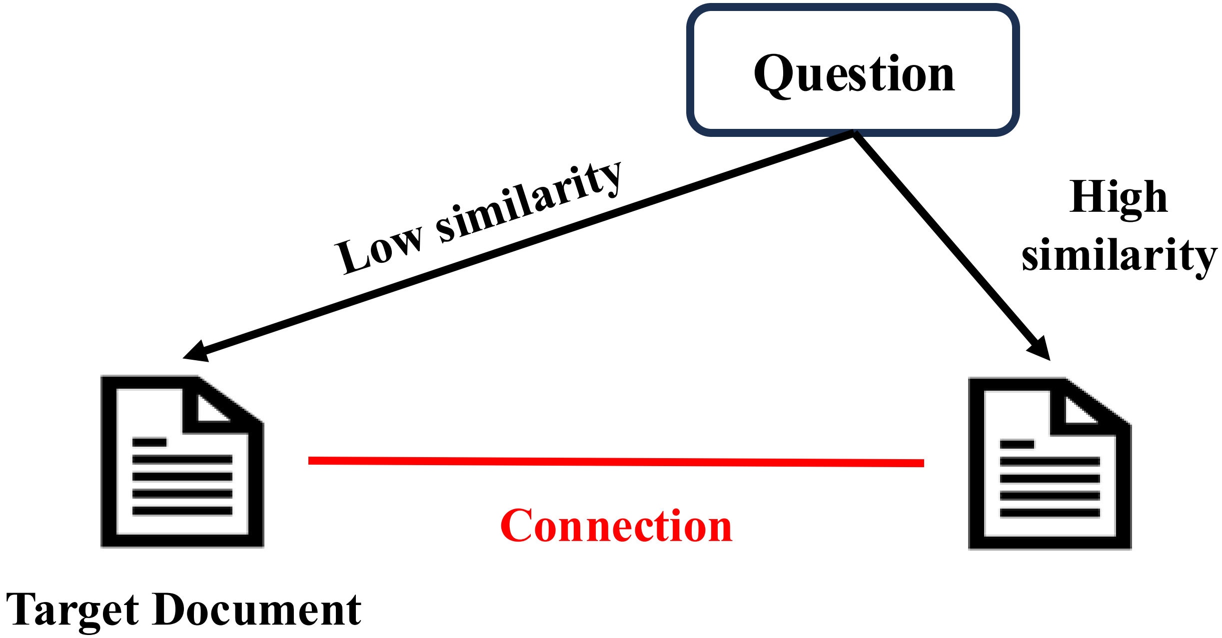
Connections between different documents or various granularity of documents.



Why should we build document graphs?

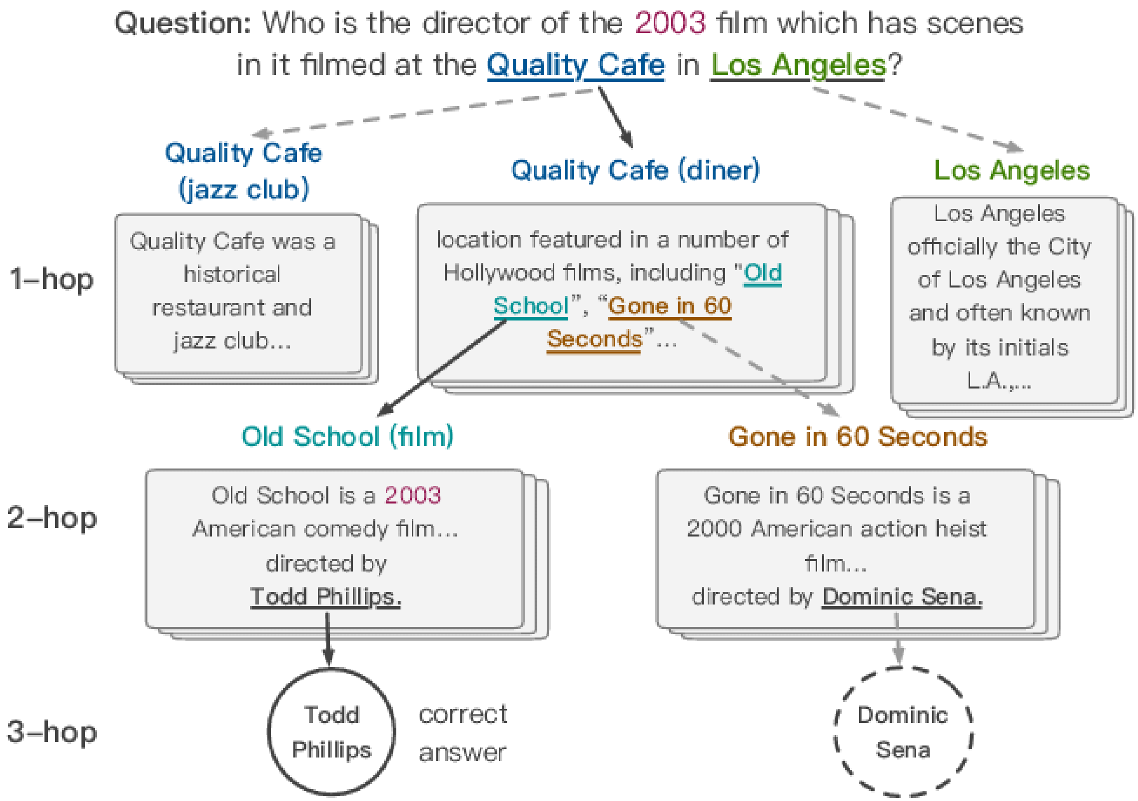
Document Graph Motivation - Beyond Semantic Similarity

Target documents may have low similarity with the question.
But can still be **retrieved via graph-based connections**.



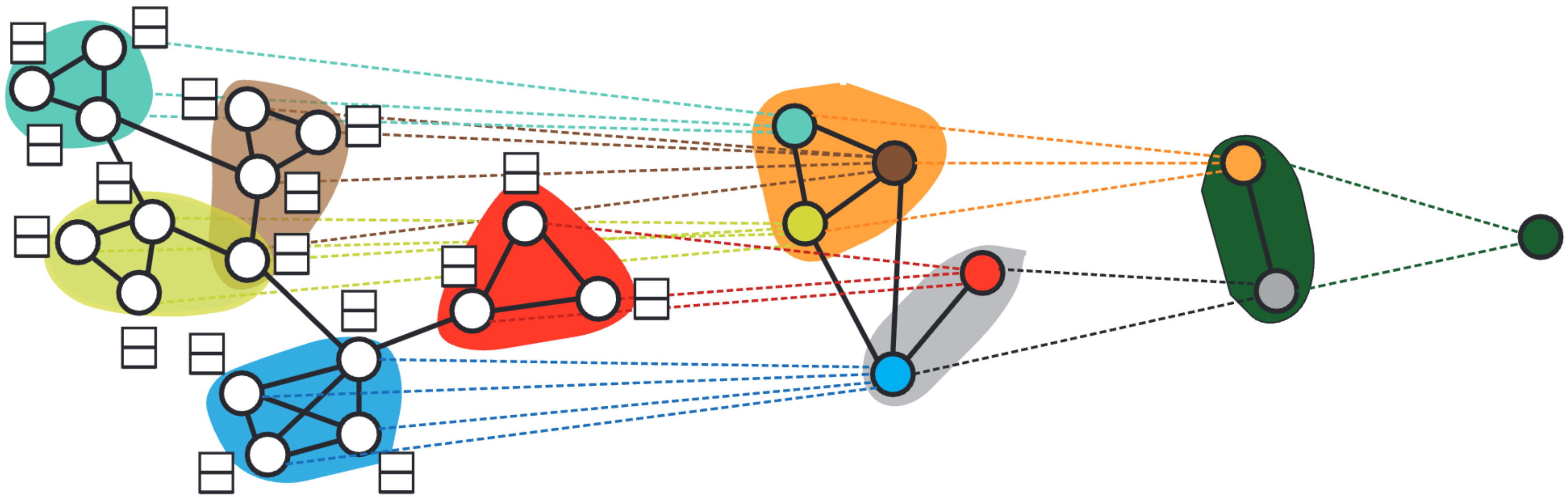
Document Graph Motivation - Multi-hop Reasoning

The graph structure inherently supports multi-hop reasoning.



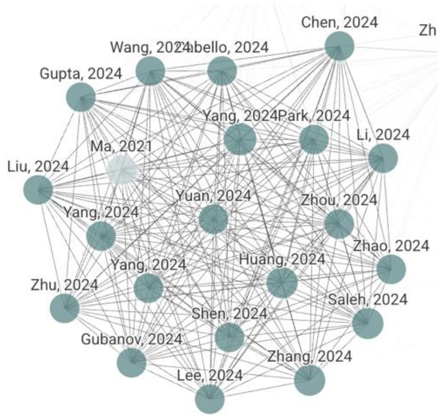
Document Graph Motivation - Global Summarization

Hierarchical graph structure supports global information retrieval.

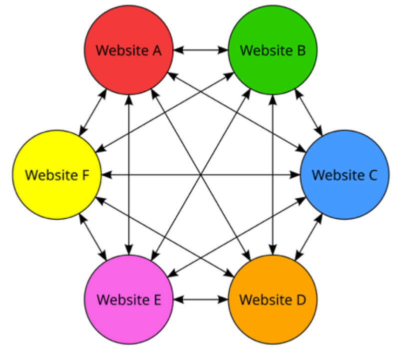


Document Graph Construction – Explicit Construction

Building graphs using (pre)-defined relationships present in the data.

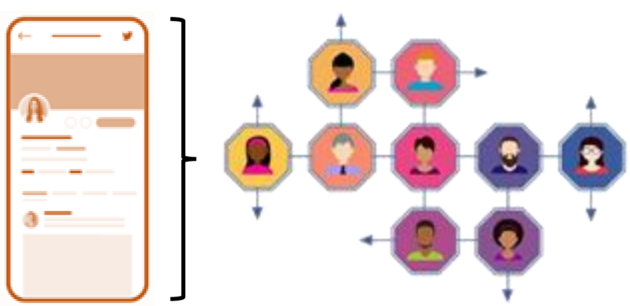


Citation

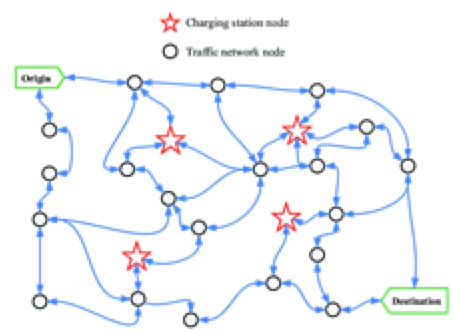


Web Hyperlinks

User Profile



Social Relation

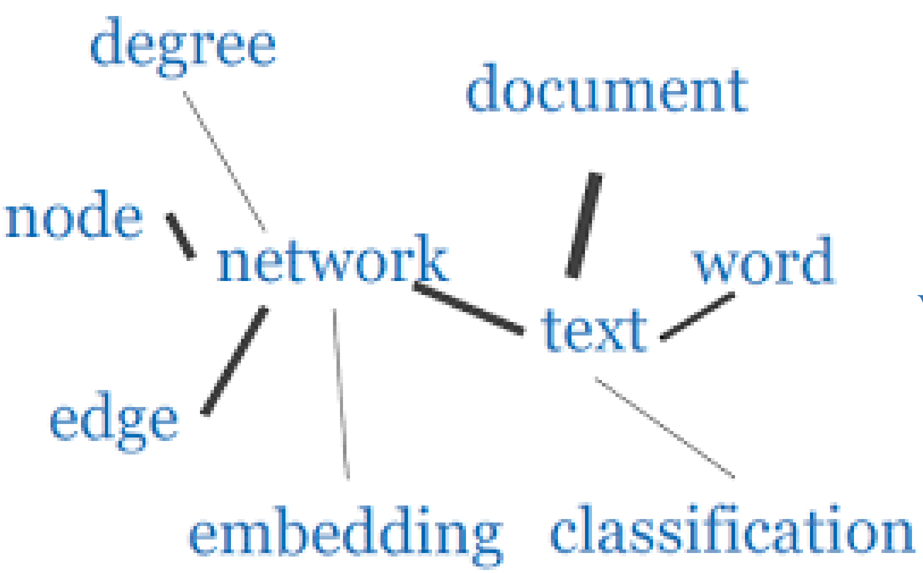


Traffic Document

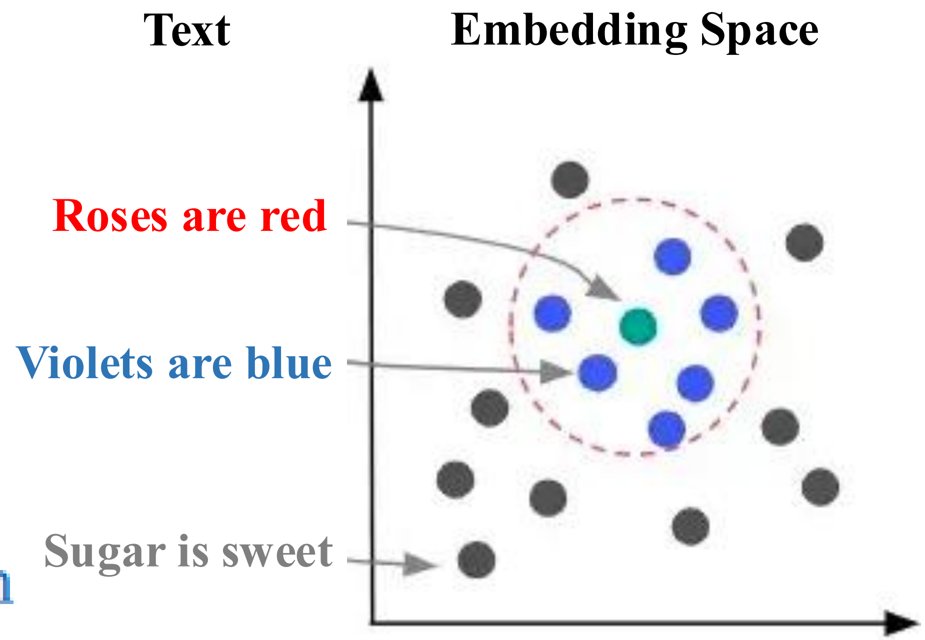
Spatial Relation

Document Graph Construction – Implicit Construction

Building graphs by leveraging latent or implicit relations between nodes



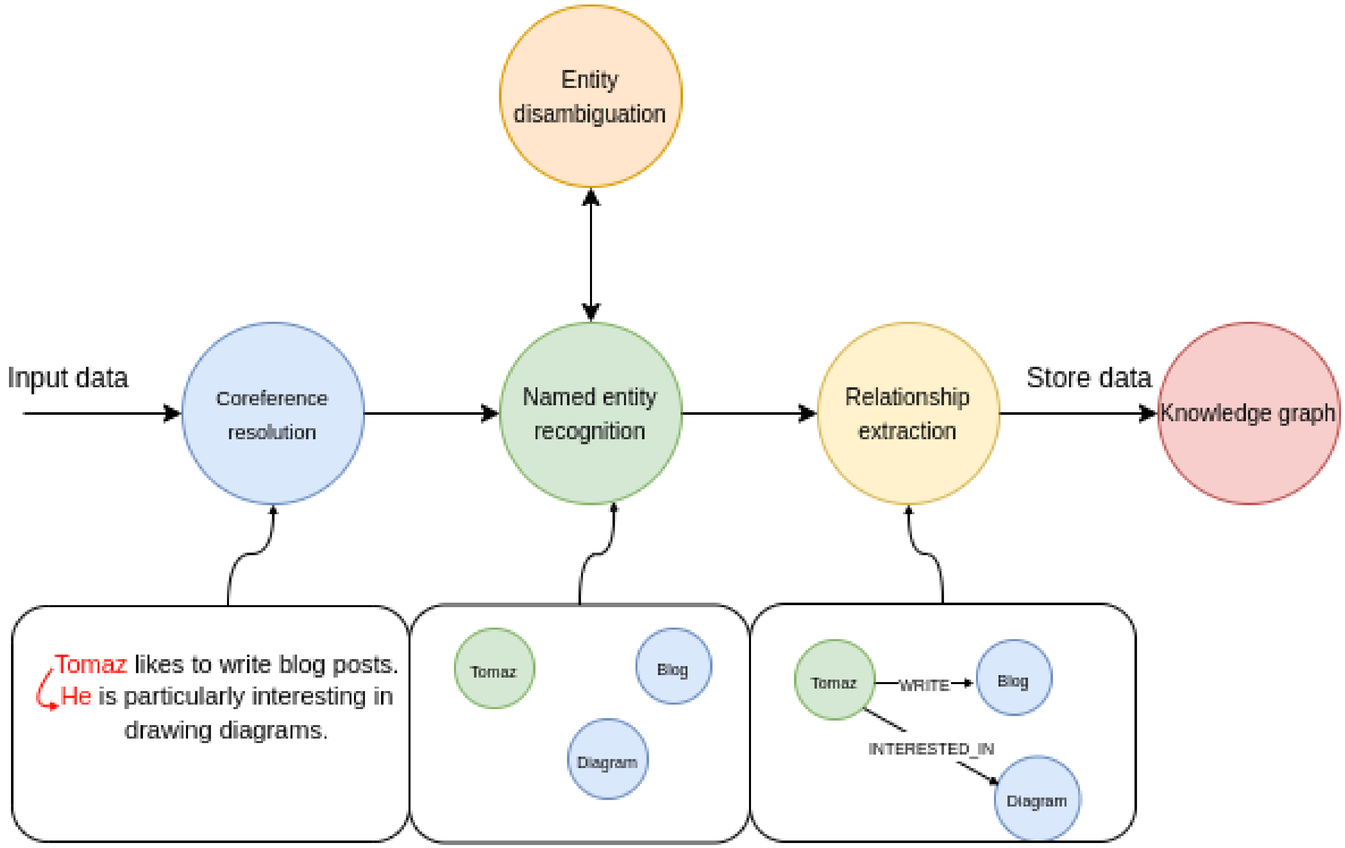
Word Co-Occurrence



Semantic Similarity

Document Graph Construction – Implicit Construction

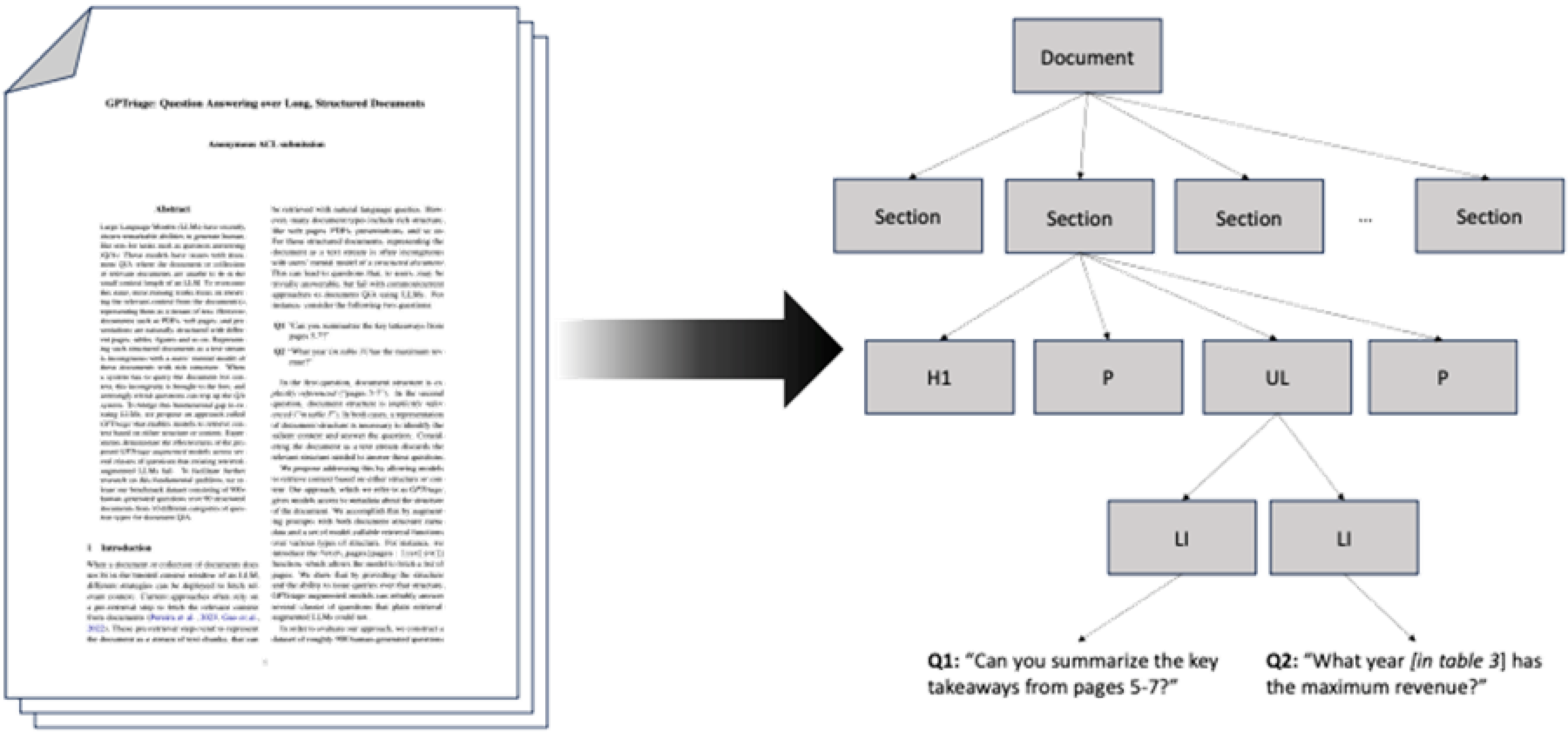
Building graphs by leveraging latent or implicit relations between nodes



Entity and Relation Extraction

Document Graph Construction – Implicit Construction

Building graphs by leveraging latent or implicit relations between nodes



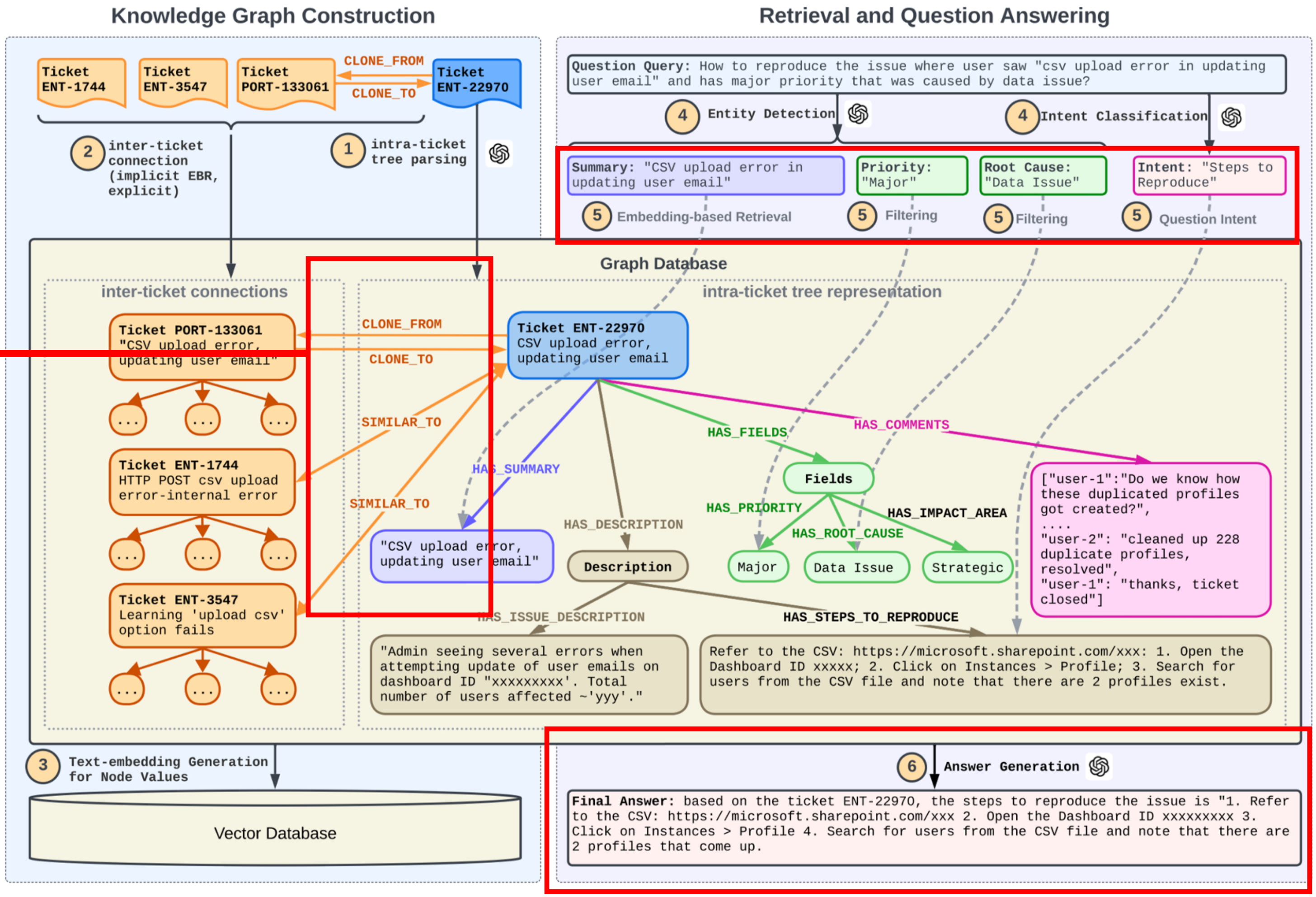
Document Structure

Document Graph – Question-Answering

Leverage the solution of previous tickets to answer the current ticket

Inter-Relations:

- **Clone**
- **Similarity**



Document Graph – Question-Answering

Leverage the solution of previous tickets to answer the current ticket

Table 1: Retrieval Performance

	MRR	Recall@K		NDCG@K	
		K=1	K=3	K=1	K=3
Baseline	0.522	0.400	0.640	0.400	0.520
Experiment	**0.927**	**0.860**	**1.000**	**0.860**	**0.946**

Table 2: Question Answering Performance

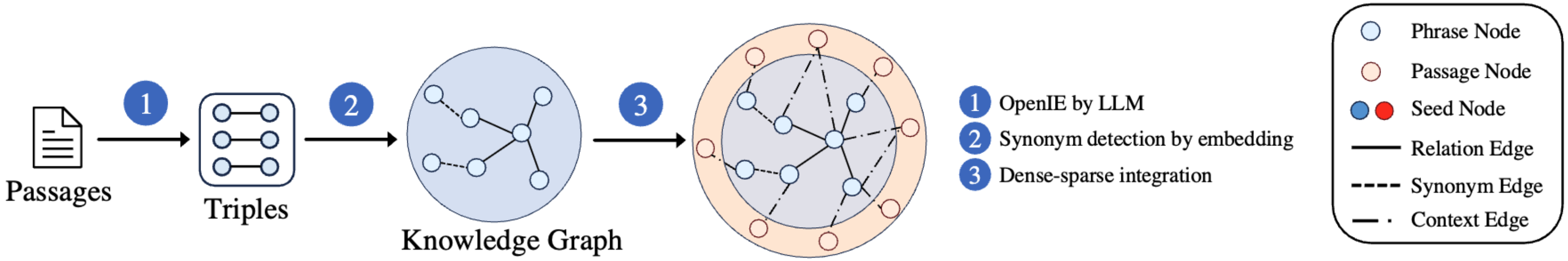
	BLEU	METEOR	ROUGE
Baseline	0.057	0.279	0.183
Experiment	**0.377**	**0.613**	**0.546**

Table 3: Customer Support Issue Resolution Time

Group	Mean	P50	P90
Tool Not Used	40 Hours	7 Hours	87 Hours
Tool Used	**15 hours**	**5 hours**	**47 hours**

Document Graph – Question-Answering

HippoRAG 2 - Implicit graph construction from documents



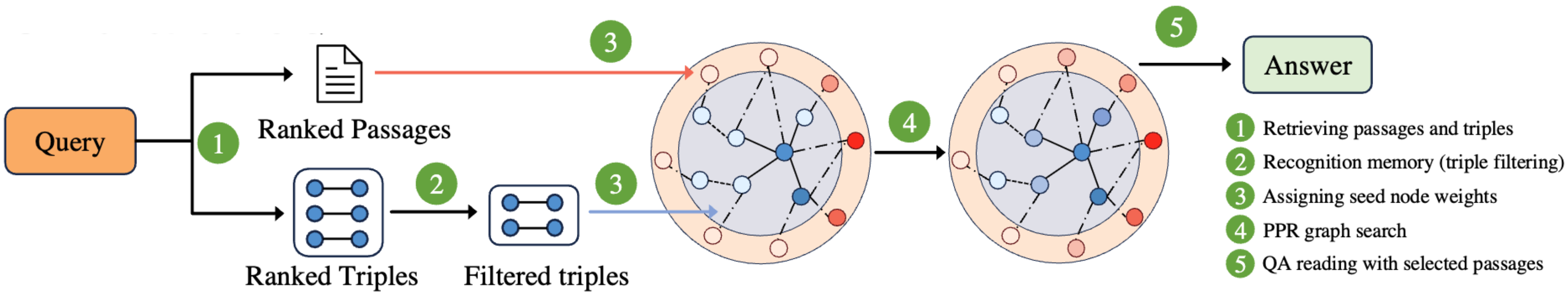
1. Triplet Construction: LLMs extract entities/relations

2. Identify synonymous entities and connect them

3. Connect Extracted Entities with Originating Passages

Document Graph – Question-Answering

HippoRAG 2 - Retrieval & QA



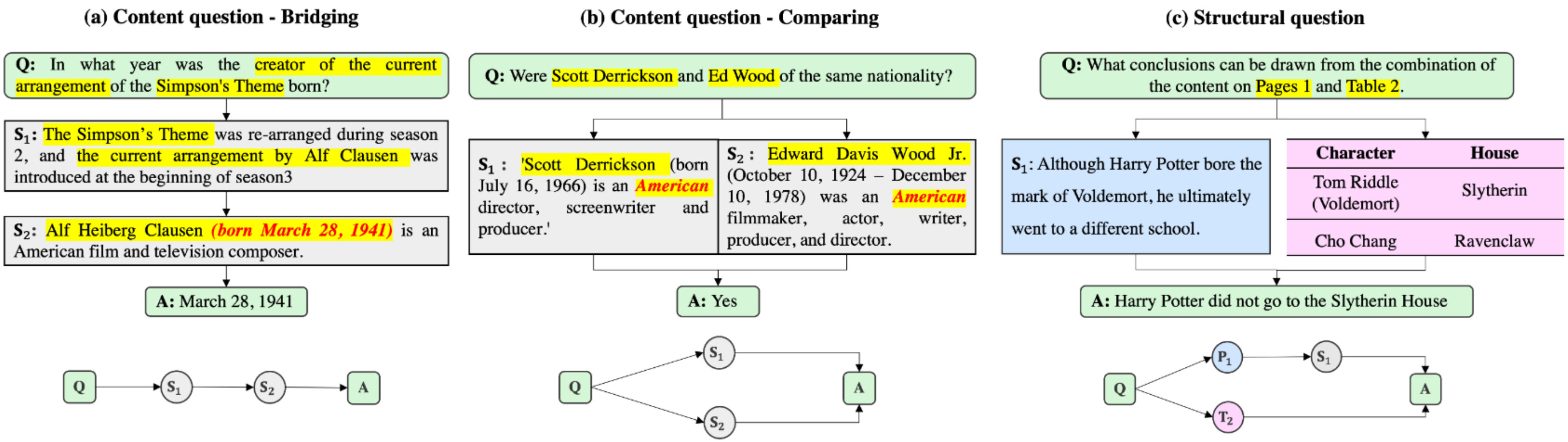
1. Passage Retrieval by Semantic Similarity
2. Triplets-Retrieval
    a. Query Entity Extraction and map to the graph
    b. Similarity (Query, Nodes)
    c. Similarity (Query, Triplets)
3. Retrieve on the Graph: Personalized PageRank search
4. Answer Generation

Document Graph – Question-Answering

Retrieval	Simple QA		Multi-Hop QA				Discourse Understanding	
	NQ	PopQA	MuSiQue	2Wiki	HotpotQA	LV-Eval	NarrativeQA	Avg
Simple Baselines								
None	54.9	32.5	26.1	42.8	47.3	6.0	12.9	38.4
Contriever (Izacard et al., 2022)	58.9	53.1	31.3	41.9	62.3	8.1	19.7	46.9
BM25 (Robertson & Walker, 1994)	59.0	49.9	28.8	51.2	63.4	5.9	18.3	47.7
GTR (T5-base) (Ni et al., 2022)	59.9	56.2	34.6	52.8	62.8	7.1	19.9	50.4
Large Embedding Models								
GTE-Qwen2-7B-Instruct (Li et al., 2023)	62.0	**56.3**	40.9	60.0	71.0	7.1	21.3	54.9
GritLM-7B (Muennighoff et al., 2024)	61.3	55.8	44.8	60.6	73.3	9.8	23.9	56.1
NV-Embed-v2 (7B) (Lee et al., 2025)	61.9	55.7	45.7	61.5	75.3	9.8	25.7	57.0
Structure-Augmented RAG								
RAPTOR (Sarthi et al., 2024)	50.7	56.2	28.9	52.1	69.5	5.0	21.4	48.8
GraphRAG (Edge et al., 2024)	46.9	48.1	38.5	58.6	68.6	11.2	23.0	49.6
LightRAG (Guo et al., 2024)	16.6	2.4	1.6	11.6	2.4	1.0	3.7	6.6
HippoRAG (Gutiérrez et al., 2024)	55.3	55.9	35.1	**71.8**	63.5	8.4	16.3	53.1
HippoRAG 2	**63.3**	56.2	**48.6**	71.0	**75.5**	**12.9**	**25.9**	**59.8**

GraphRAG is typically more effective for multi-hop QA.

Document Graph – Question-Answering

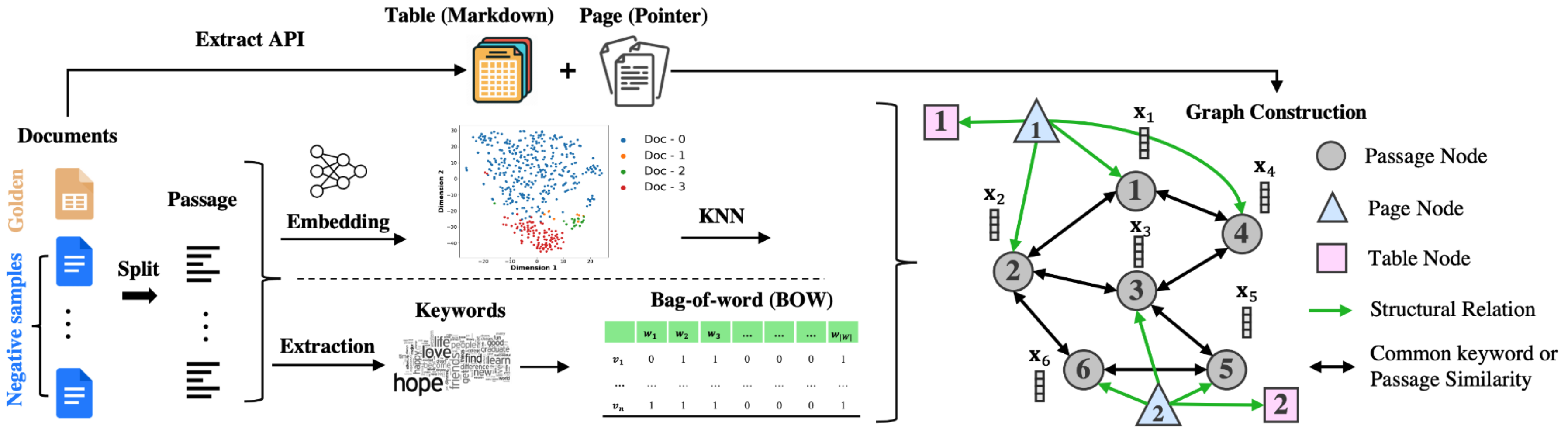


Lexical similarity

Semantic similarity

Document Structure

Document Graph – Question-Answering



1. **Graph Construction**

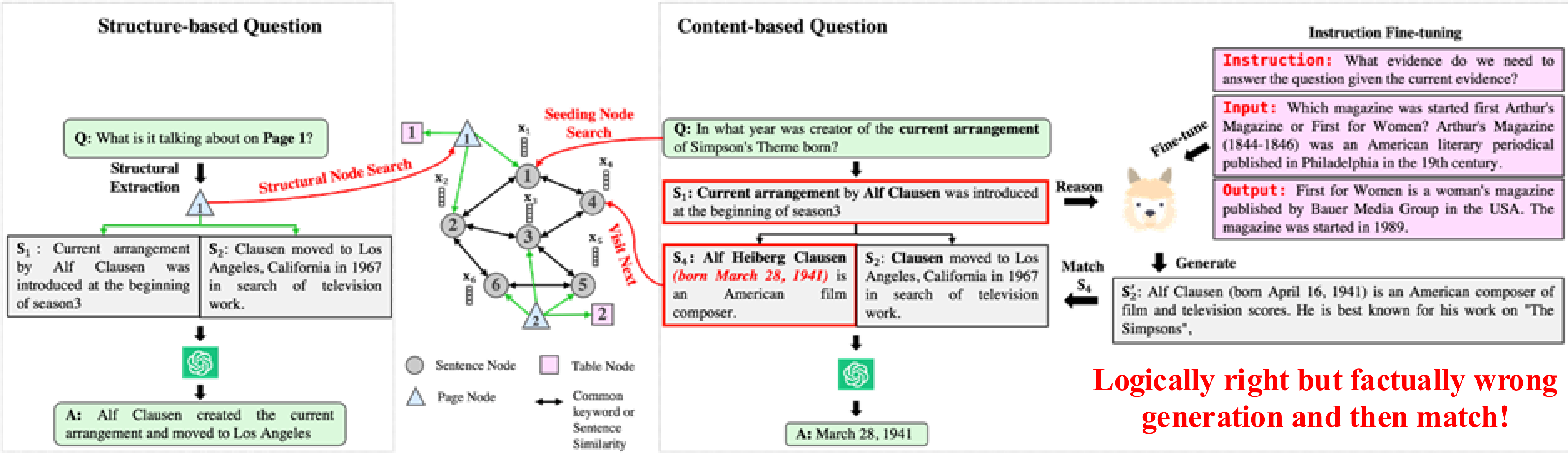
a. TF-IDF construction

b. KNN construction

c. Connect passages share same entity

d. Add Table/Page Document Meta-Structure

Document Graph – Question-Answering



2. **Retrieval (LLM traversal agent for reasoning and grounding)**

   a. Initialize the seeding passage with similarity search

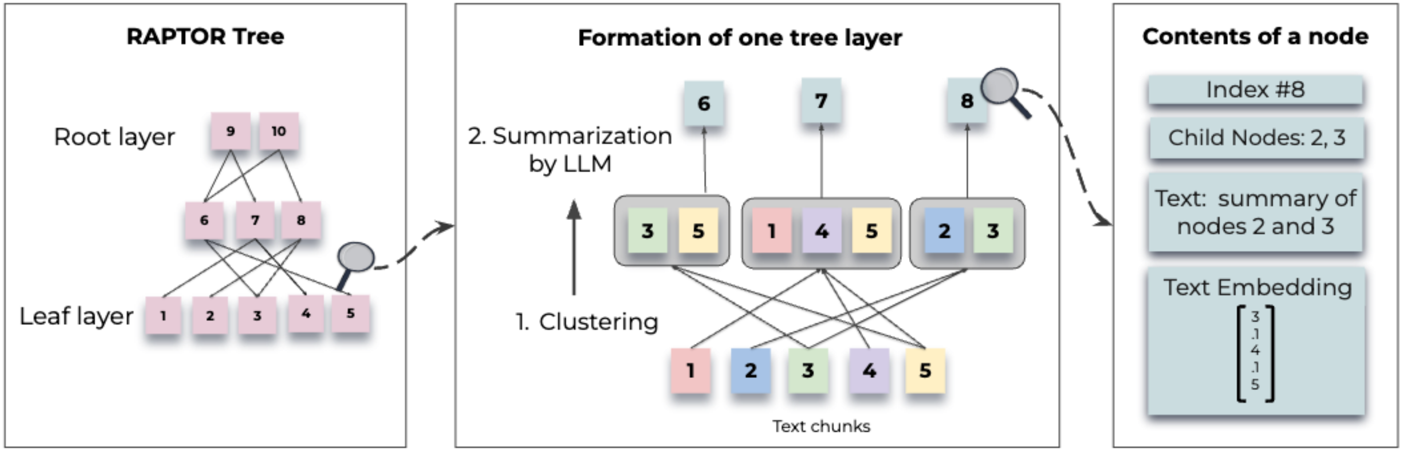
   b. LLMs predict the next passage to explore

   c. Retrieve passages based on LLM's generation

Document Graph – Question-Answering

RAPTOR – Tree-based Retrieval

Tree structure to capture **High/Low-level** information

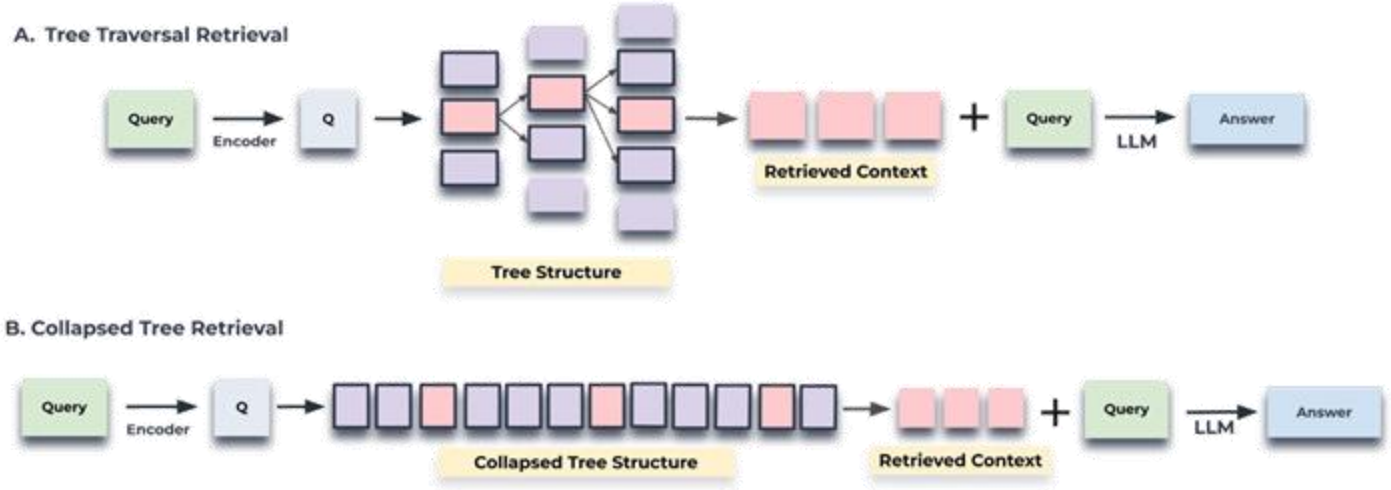


1. **Graph Construction**
    a. Represent each leaf node as a text chunk
    b. Apply clustering algorithms to group related chunks
    c. Summarize each cluster to form higher-level nodes
    d. Repeat the construction process

Document Graph – Question-Answering

RAPTOR – Tree-based Retrieval

Tree structure to capture **High/Low-level** information



2. **Retrieval**
   a. Tree Traversal Retrieval: Root-to-Leaf Traversal, Progressively Narrowing Down
   b. Collapsed Tree Retrieval: Flatten Tree Structure, Independently Retrieve

Document Graph – Question-Answering

RAPTOR – Tree-based Retrieval

Tree structure to capture **High/Low-level** information

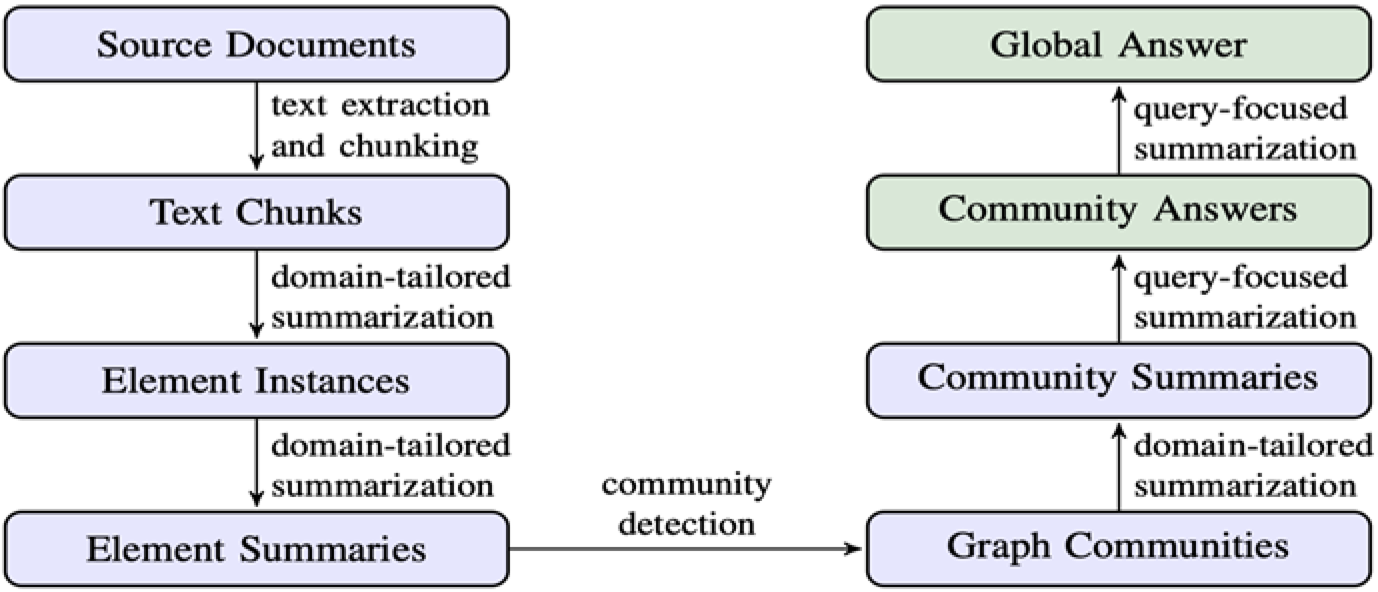
Model	ROUGE	BLEU-1	BLEU-4	METEOR
SBERT with RAPTOR	**30.87%**	**23.50%**	**6.42%**	**19.20%**
SBERT without RAPTOR	29.26%	22.56%	5.95%	18.15%
BM25 with RAPTOR	**27.93%**	**21.17%**	**5.70%**	**17.03%**
BM25 without RAPTOR	23.52%	17.73%	4.65%	13.98%
DPR with RAPTOR	**30.94%**	**23.51%**	**6.45%**	**19.05%**
DPR without RAPTOR	29.56%	22.84%	6.12%	18.44%

Tree-based retrieval improves global QA performance.

Document Graph – Document Summarization

Microsoft GraphRAG

Corpus to summarize too large **vs** LLM context window is limited

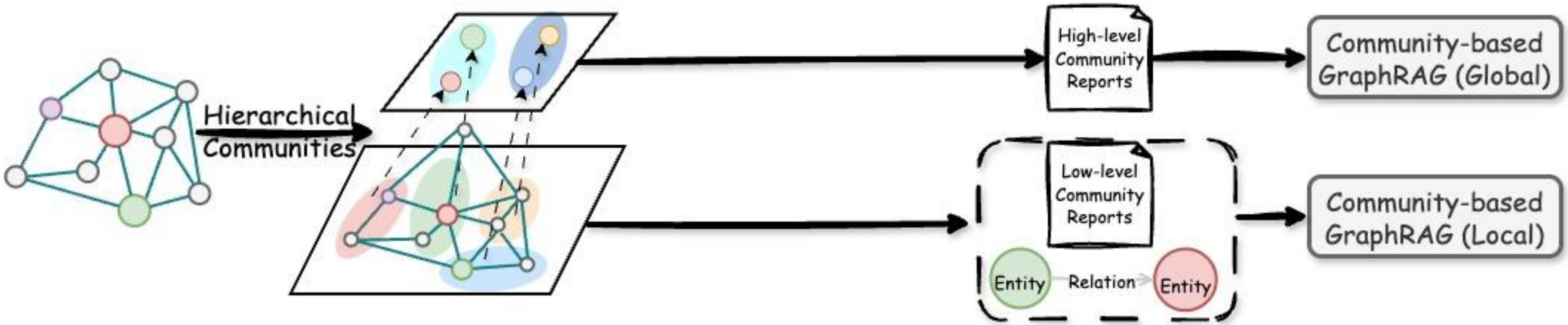


Extract a knowledge graph from the whole corpus.

Hierarchical Community Detection and Summarization Multiple Granularities

Document Graph – Document Summarization

Microsoft GraphRAG



1. **Local Retrieval** from leaf nodes

2. **Global Retrieval** from summarization nodes

Document Graph – Document Summarization

Microsoft GraphRAG

Podcast transcripts

	SS	TS	C0	C1	C2	C3
SS	50	17	28	25	22	21
TS	83	50	50	48	43	44
C0	72	50	50	53	50	49
C1	75	52	47	50	52	50
C2	**78**	**57**	**50**	**48**	**50**	**52**
C3	79	56	51	50	48	50

Comprehensiveness

	SS	TS	C0	C1	C2	C3
SS	50	18	23	25	19	19
TS	82	50	50	50	43	46
C0	77	50	50	50	46	44
C1	75	50	50	50	44	45
C2	81	57	54	56	50	48
C3	**81**	**54**	**56**	**55**	**52**	**50**

Diversity

	SS	TS	C0	C1	C2	C3
SS	50	42	57	52	49	51
TS	**58**	**50**	**59**	**55**	**52**	**51**
C0	43	41	50	49	47	48
C1	48	45	51	50	49	50
C2	51	48	53	51	50	51
C3	49	49	52	50	49	50

Empowerment

	SS	TS	C0	C1	C2	C3
SS	**50**	**56**	**65**	**60**	**60**	**60**
TS	44	50	55	52	51	52
C0	35	45	50	47	48	48
C1	40	48	53	50	50	50
C2	40	49	52	50	50	50
C3	40	48	52	50	50	50

Directness

News articles

	SS	TS	C0	C1	C2	C3
SS	50	20	28	25	21	21
TS	80	50	44	41	38	36
C0	72	56	50	52	54	52
C1	**75**	**59**	**48**	**50**	**58**	**55**
C2	79	62	46	42	50	59
C3	79	64	48	45	41	50

Comprehensiveness

	SS	TS	C0	C1	C2	C3
SS	50	33	38	35	29	31
TS	67	50	53	45	44	40
C0	62	47	50	40	41	41
C1	65	55	60	50	50	50
C2	**71**	**56**	**59**	**50**	**50**	**51**
C3	69	60	59	50	49	50

Diversity

	SS	TS	C0	C1	C2	C3
SS	50	47	57	49	50	50
TS	53	50	58	50	50	48
C0	43	42	50	42	45	44
C1	**51**	**50**	**58**	**50**	**52**	**51**
C2	50	50	55	48	50	50
C3	50	52	56	49	50	50

Empowerment

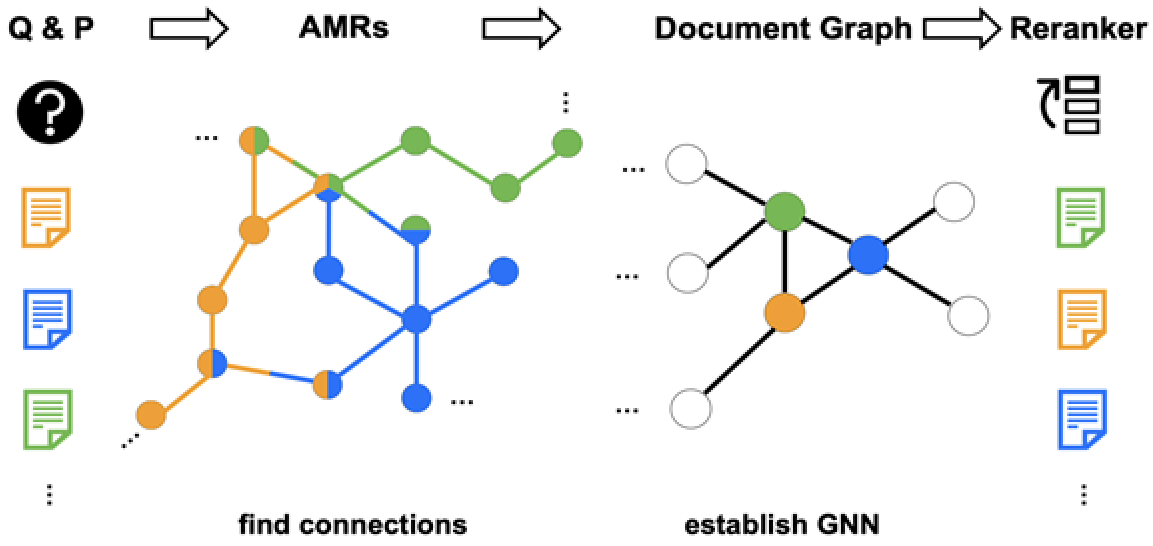
	SS	TS	C0	C1	C2	C3
SS	**50**	**54**	**59**	**55**	**55**	**54**
TS	46	50	55	53	52	52
C0	41	45	50	48	48	47
C1	45	47	52	50	49	49
C2	45	48	52	51	50	49
C3	46	48	53	51	51	50

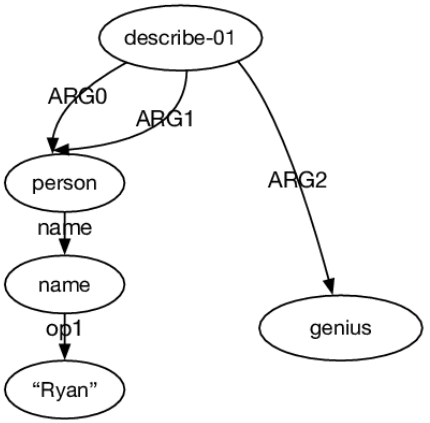
Directness

GraphRAG is typically superior in both comprehensiveness and diversity.

Document Graph – Document Retrieval

G-RAG: A document-graph-based reranker





AMR graph: Ryan's description of himself: a genius

1. **Graph Construction**
   a. Build Abstract Meaning Representation (AMR) graphs
   b. Connect documents share same nodes

Document Graph – Document Retrieval

G-RAG：A document-graph-based reranker

2. GNNs for Reranking

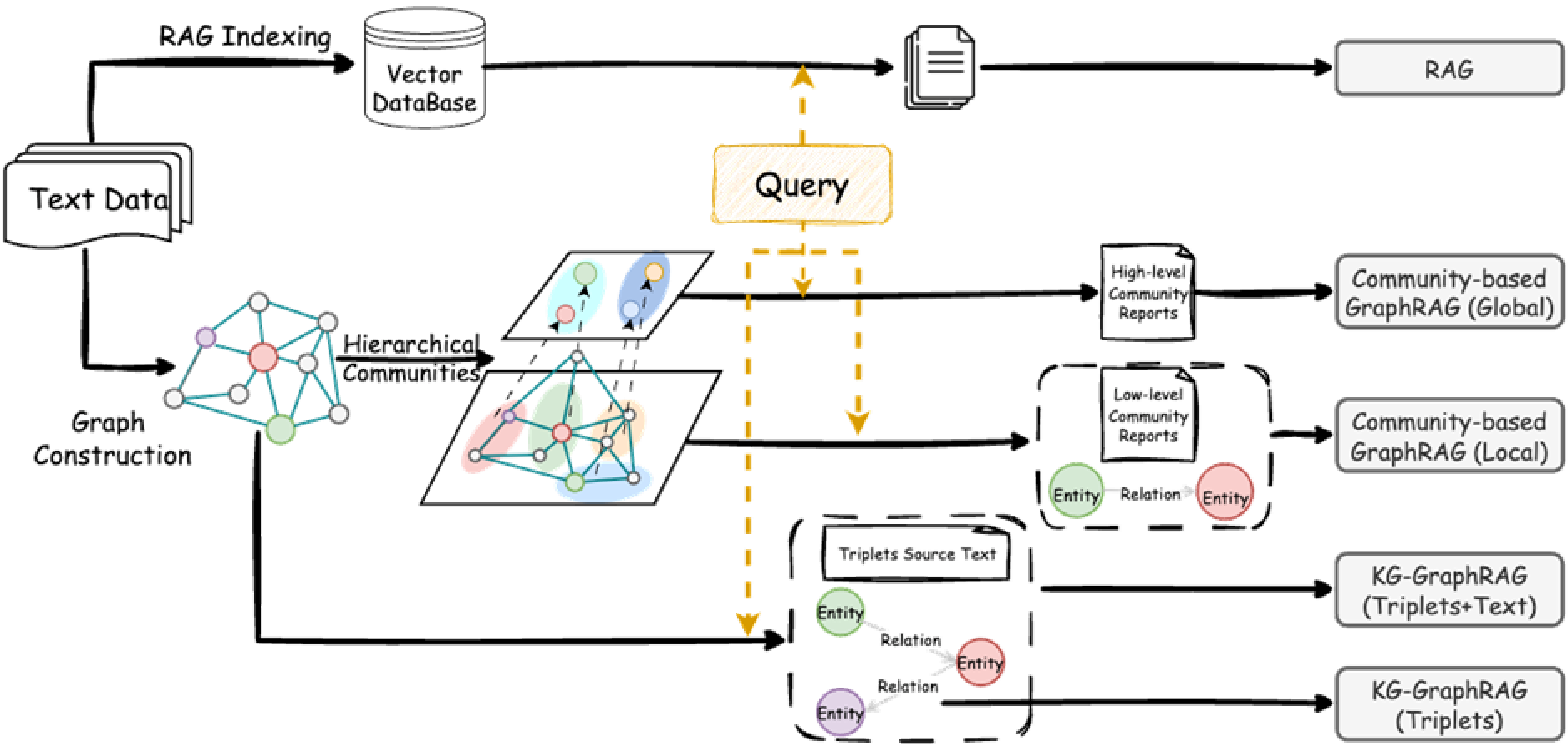
Document and query embedding: $$\mathbf{x}_v^{\ell} = g\left(\mathbf{x}_v^{\ell-1}, \bigcup_{u \in \mathcal{N}(v)} f\left(\mathbf{x}_u^{\ell-1}, \mathbf{e}_{uv}^{\ell-1}\right)\right) \qquad \mathbf{y} = \text{Encode}(q).$$

Ranking based on the similarity: $$s_i = \mathbf{y}^{\top} \mathbf{x}_{v_i}^{L}$$

Ranking loss $$\mathcal{RL}_q\left(s_i, s_j, r\right) = \max\left(0, -r\left(s_i - s_j\right) + 1\right),$$

RAG vs. GraphRAG

A systematic evaluation between RAG and GraphRAG.



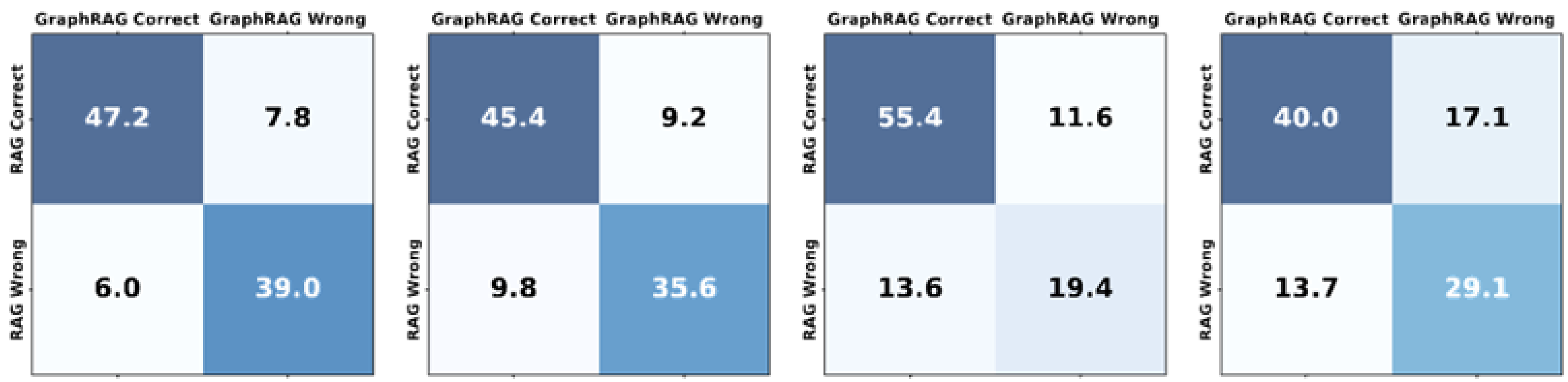
RAG vs. GraphRAG: QA Task

	Single-Hop						Multi-Hop					
Method	NQ						Hotpot					
	Llama 3.1-8B			Llama 3.1-70B			Llama 3.1-8B			Llama 3.1-70B		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
RAG	**71.7**	**63.93**	**64.78**	**74.55**	**67.82**	**68.18**	62.32	60.47	60.04	66.34	63.99	63.88
KG-GraphRAG (Triplets only)	40.09	33.56	34.28	37.84	31.22	28.50	26.88	24.81	25.02	32.59	30.63	30.73
KG-GraphRAG (Triplets+Text)	58.36	48.93	50.27	60.91	52.75	53.88	45.22	42.85	42.60	51.44	48.99	48.75
Community-GraphRAG (Local)	69.48	62.54	63.01	71.27	65.46	65.44	**64.14**	**62.08**	**61.66**	**67.20**	**64.89**	**64.60**
Community-GraphRAG (Global)	60.76	54.99	54.48	61.15	55.52	55.05	45.72	47.60	45.16	48.33	48.56	46.99

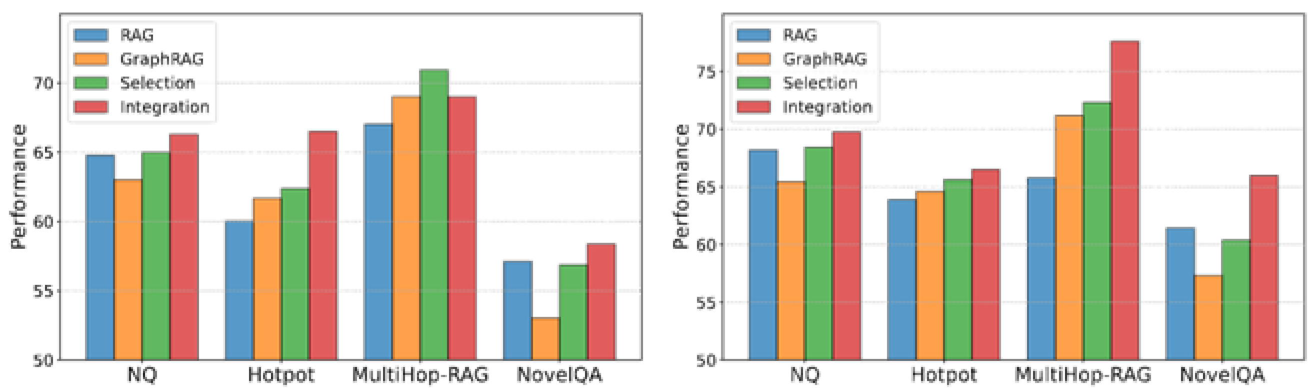
- RAG excels on detailed single-hop queries.
- GraphRAG, particularly CommunityGraphRAG (Local), excels on multi-hop queries.
- Community-GraphRAG (Global) often struggles on QA tasks.
- KG-based GraphRAG also generally underperform on QA tasks due to the incomplete graph.

RAG vs. GraphRAG: QA Task

RAG and GraphRAG are Complementary!



(a) NQ (b) Hotpot (c) MultiHop-RAG (d) NovelQA



(a) Llama3.1-8B (b) Llama3.1-70B

Combining RAG and GraphRAG yields better performance!

RAG vs. GraphRAG: Summarization Task

Ground Truth (Human Answer) as Judge

Table 4: The performance of query-based single document summarization task using Llama3.1-8B.

Method	SQuALITY						QMSum					
	ROUGE-2			BERTScore			ROUGE-2			BERTScore		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
RAG	15.09	8.74	10.08	74.54	81.00	77.62	21.50	**3.80**	6.32	**81.03**	84.45	**82.69**
KG-GraphRAG (Triplets only)	11.99	6.16	7.41	82.46	84.30	83.17	13.71	2.55	4.15	80.16	82.96	81.52
KG-GraphRAG (Triplets+Text)	15.00	**9.48**	10.52	**84.37**	**85.88**	**84.92**	16.83	3.32	5.38	80.92	83.64	82.25
Community-GraphRAG (Local)	**15.82**	8.64	10.10	83.93	85.84	84.66	20.54	3.35	5.64	80.63	84.13	82.34
Community-GraphRAG (Global)	10.23	6.21	6.99	82.68	84.26	83.30	10.54	1.97	3.23	79.79	82.47	81.10
Integration	15.69	9.32	**10.67**	74.56	81.22	77.73	**21.97**	**3.80**	**6.34**	80.89	**84.47**	82.63

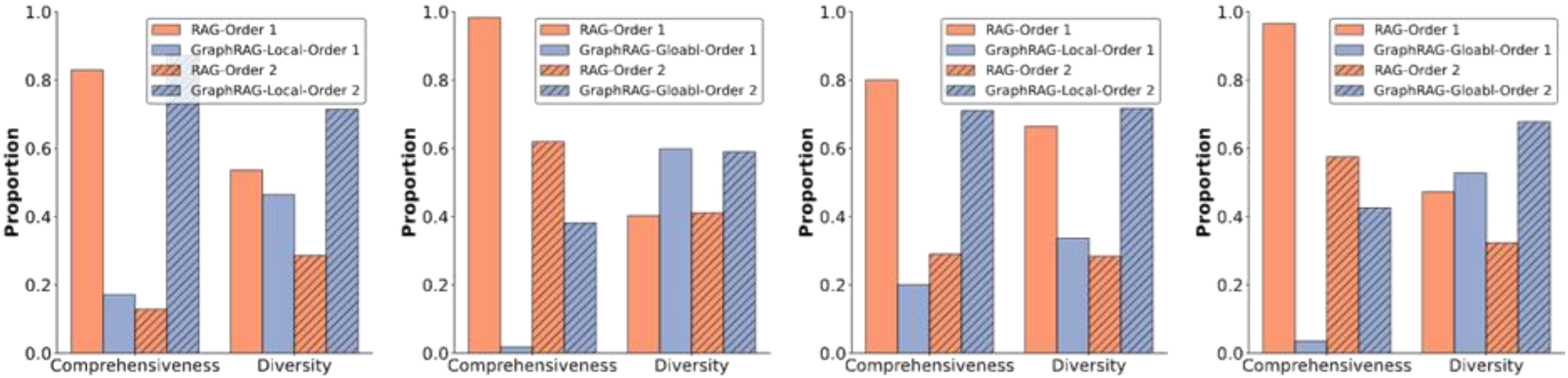
Table 5: The performance of query-based multiple document summarization task using Llama3.1-8B.

Method	ODSum-story						ODSum-meeting					
	ROUGE-2			BERTScore			ROUGE-2			BERTScore		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
RAG	**15.39**	8.44	**9.81**	**83.87**	**85.74**	**84.57**	15.50	**6.43**	**8.77**	**83.12**	**85.84**	**84.45**
KG-GraphRAG (Triplets only)	11.02	5.56	6.62	82.09	83.91	82.77	11.64	4.87	6.58	81.13	84.32	82.69
KG-GraphRAG (Triplets+Text)	9.19	5.82	6.22	79.39	83.30	81.03	11.97	4.97	6.72	81.50	84.41	82.92
Community-GraphRAG (Local)	13.84	7.19	8.49	83.19	85.07	83.90	15.65	5.66	8.02	82.44	85.54	83.96
Community-GraphRAG (Global)	9.40	4.47	5.46	81.46	83.54	82.30	11.44	3.89	5.59	81.20	84.50	82.81
Integration	14.77	**8.55**	9.53	83.73	85.56	84.40	**15.69**	6.15	8.51	82.87	85.81	84.31

RAG aligns more closely with human-written answers.

RAG vs. GraphRAG: Summarization Task

LLM as Judge



(a) QMSum Local (b) QMSum Global (c) ODSum-story Local (d) ODSum-story Global

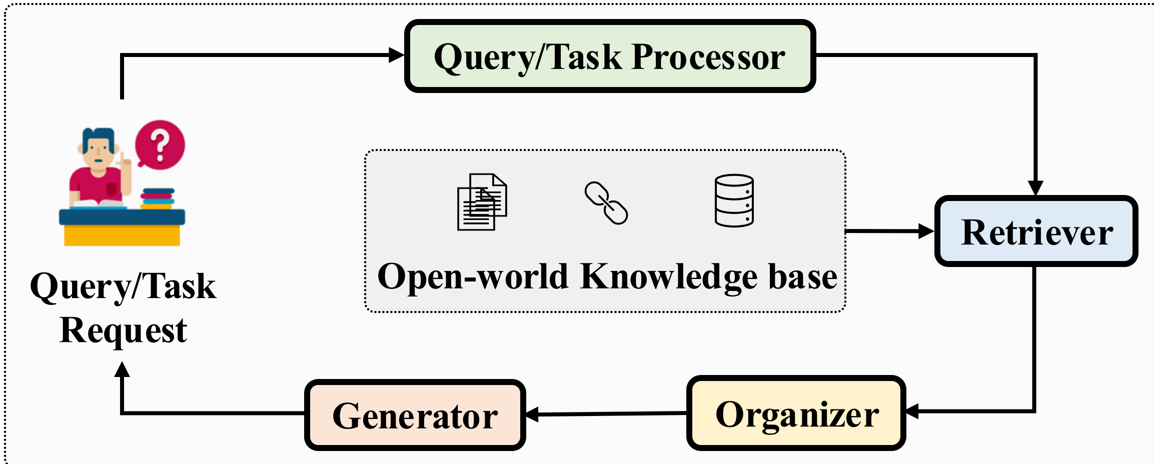
1. **Strong position bias is observed**

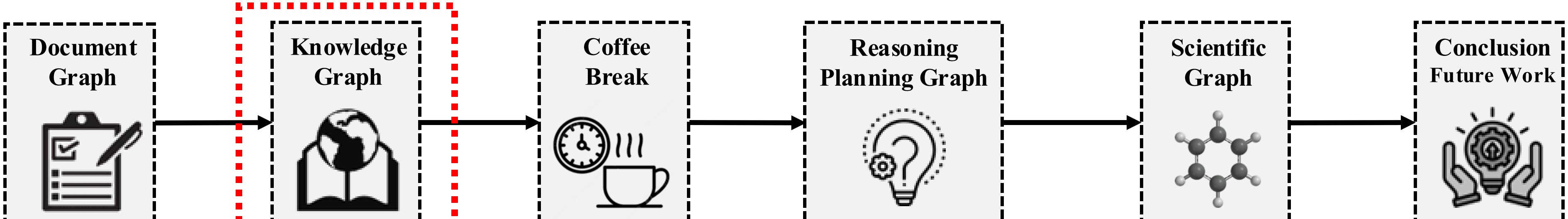
2. **Community-based GraphRAG with global search prefers corpus global structure**

Document Graph - Future Works

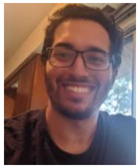
1. **Graph Construction**
   a. Task-specific graph construction
   b. Balancing efficiency and graph completeness
2. **Retrieval and Traversal**
   a. Adaptive retrieval strategies based on query type and complexity
   b. Multi-hop retrieval with reasoning over graph structure
3. **RAG and GraphRAG Integration**
   a. Analyzing the Pros and Cons of RAG and GraphRAG
   b. Designing methods to combine their strengths
4. **Evaluation**
   a. New benchmarks designed specifically for graph-based retrieval and generation
   b. Proposing fine-grained evaluation metrics

Outline





Haoyu Han
24 min



Harry Shomer
24 min

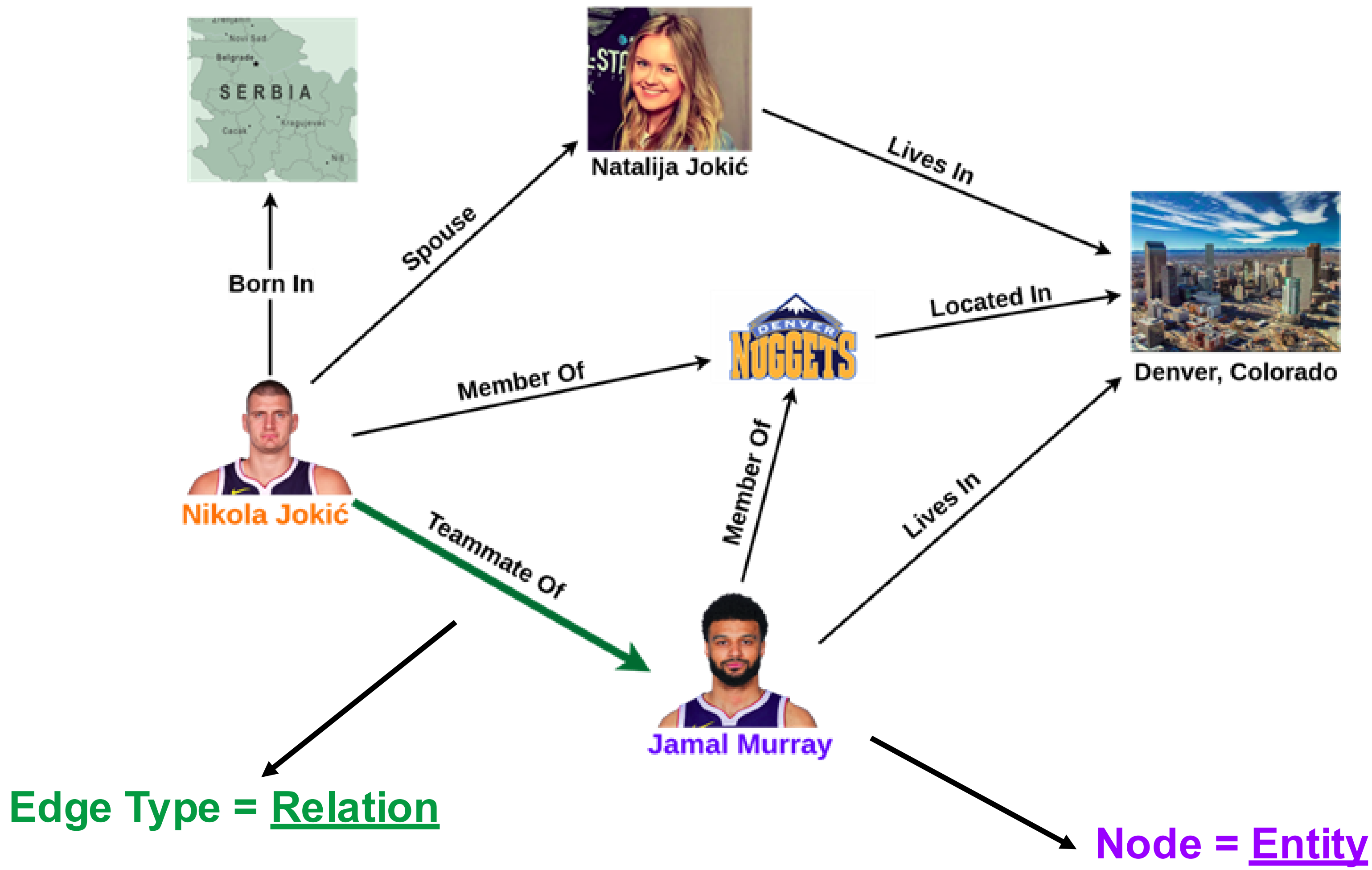
4 min

Yongjia Lei
24 min

Kai Guo
24 min

Yu Wang
5 min

Knowledge Graph - What are Knowledge Graph (KGs)?

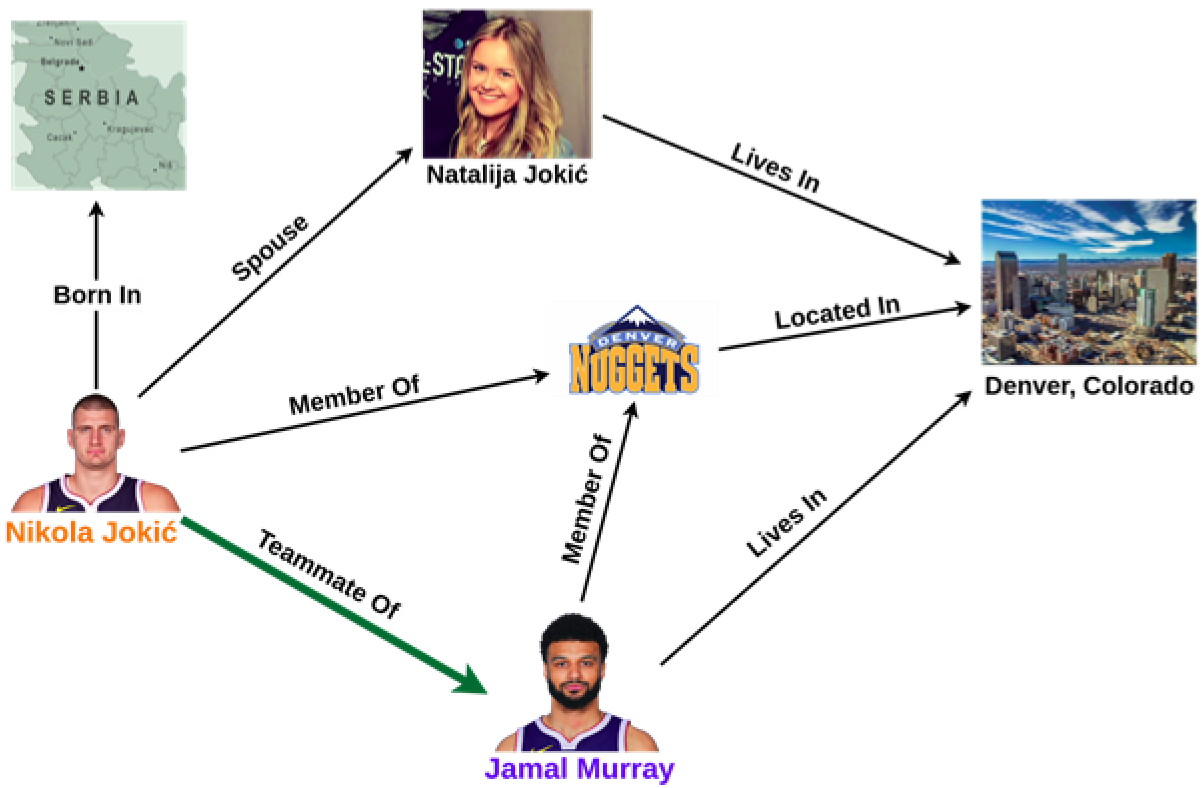


Knowledge Graph - What are Knowledge Graph (KGs)?

Fact = ( , Teammate of, )

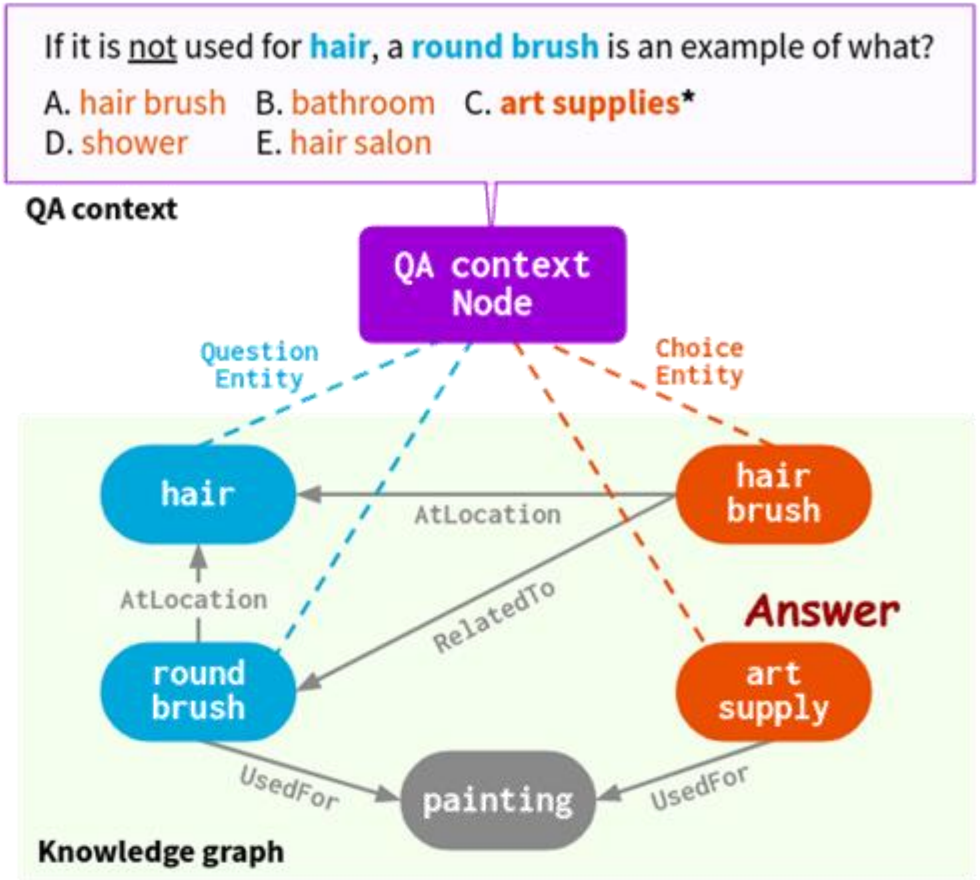
Nikola Jokić → ***Head*** Entity

Jamal Murray → ***Tail*** Entity



Knowledge Graph - Tasks

Question Answering



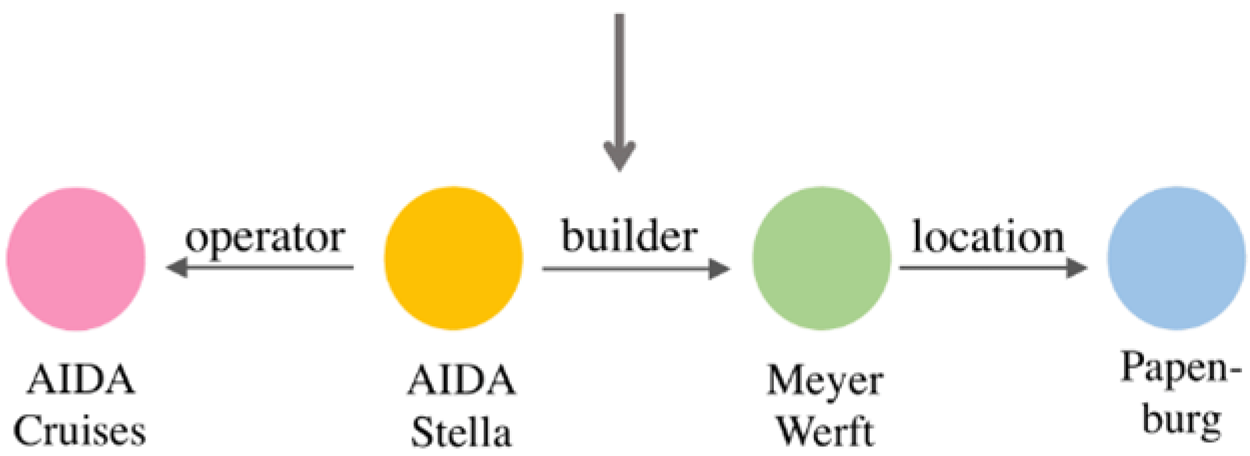
Knowledge Graph - Tasks

Fack Checking

Claim: Yeah! Actually AIDA Cruise line operated <u>a ship</u> which was built by <u>a company</u> in Papenburg!

Evidence:

DBpedia



Label: SUPPORTED

Knowledge Graph - Tasks

Knowledge Graph Completion

Given:

( , Lives In, ???)

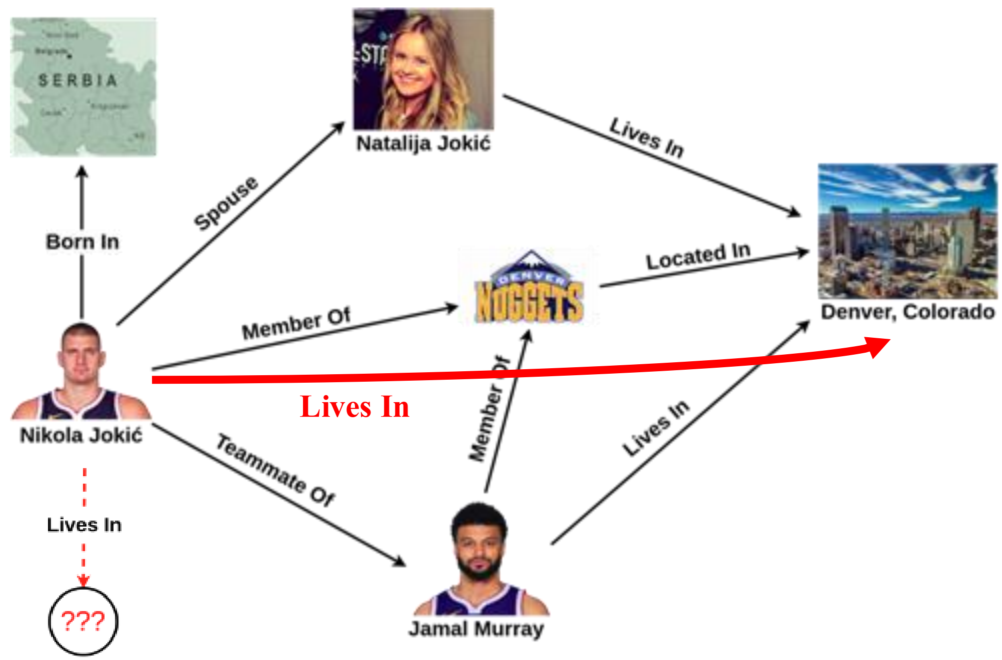
Nikola Jokić

Given:

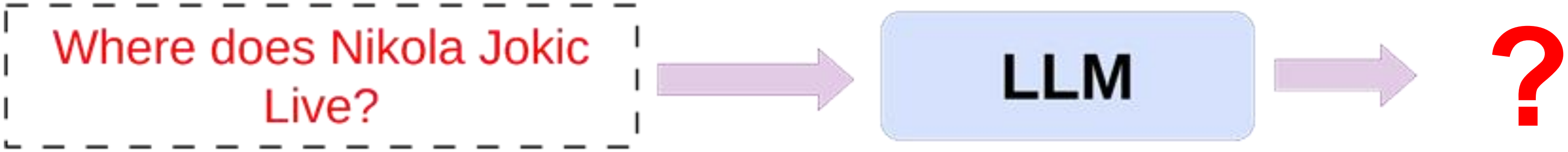
( , Lives In, )

Nikola Jokić

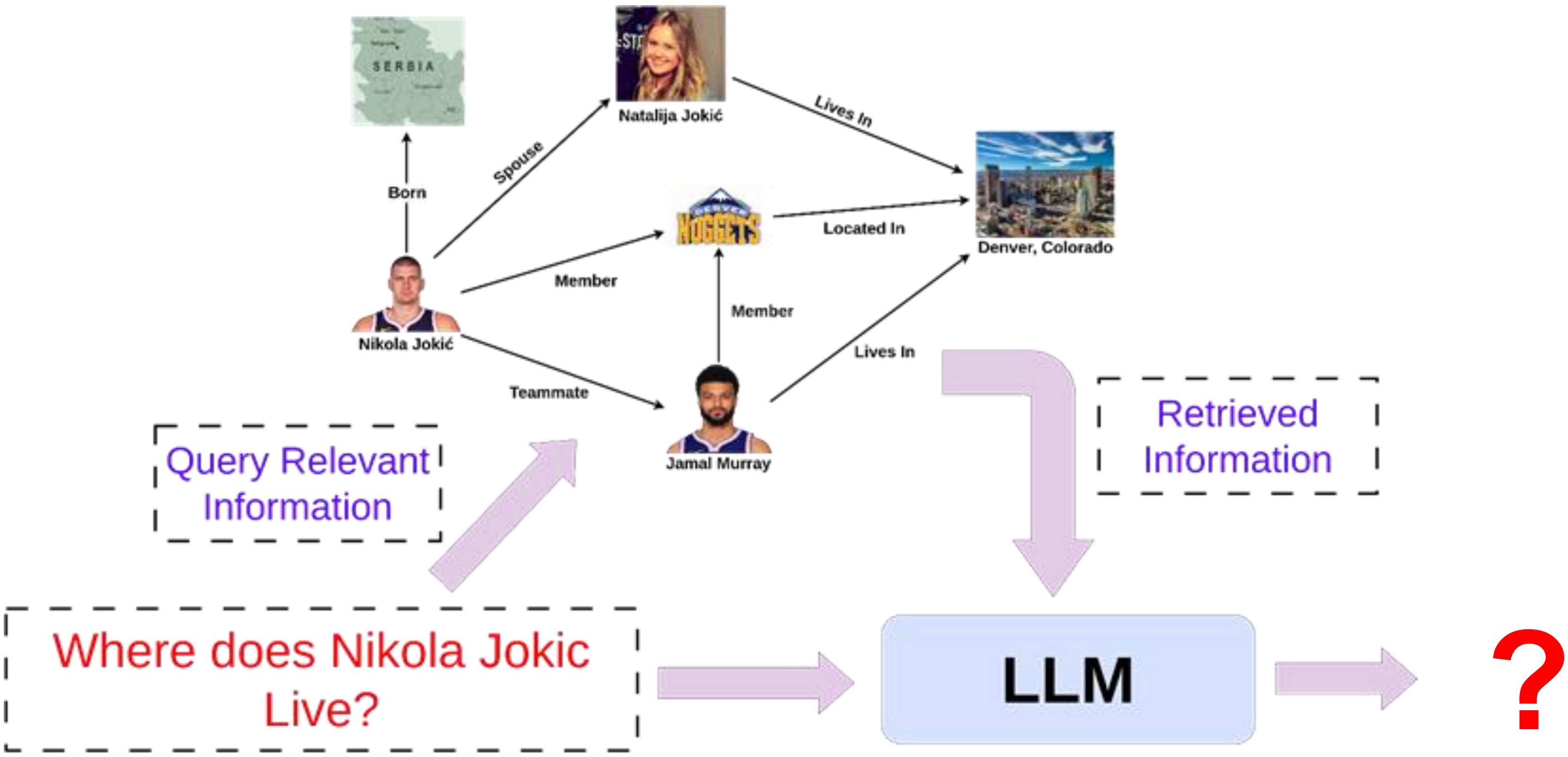
Denver, CO



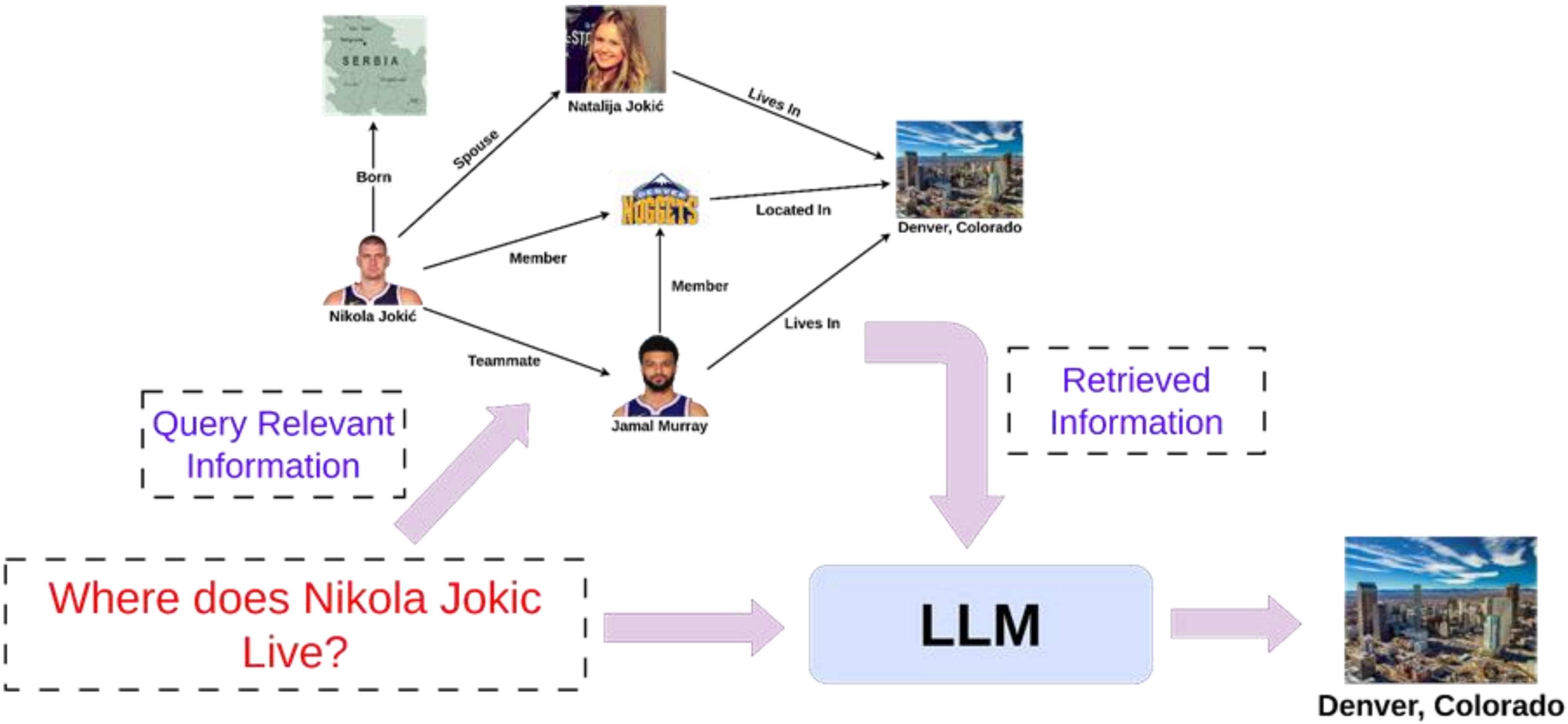
Knowledge Graph - Using KGs for GraphRAG



Knowledge Graph - Using KGs for GraphRAG



Knowledge Graph - Using KGs for GraphRAG



Knowledge Graph - How are KGs are Constructed?

1) Manual Construction

- Done via human annotation
- Popular example is the WikiData database

Knowledge Graph - How are KGs are Constructed?

Entity

Facts with Hinton as Head Entity

Knowledge Graph - How are KGs are Constructed?

1) Manual Construction

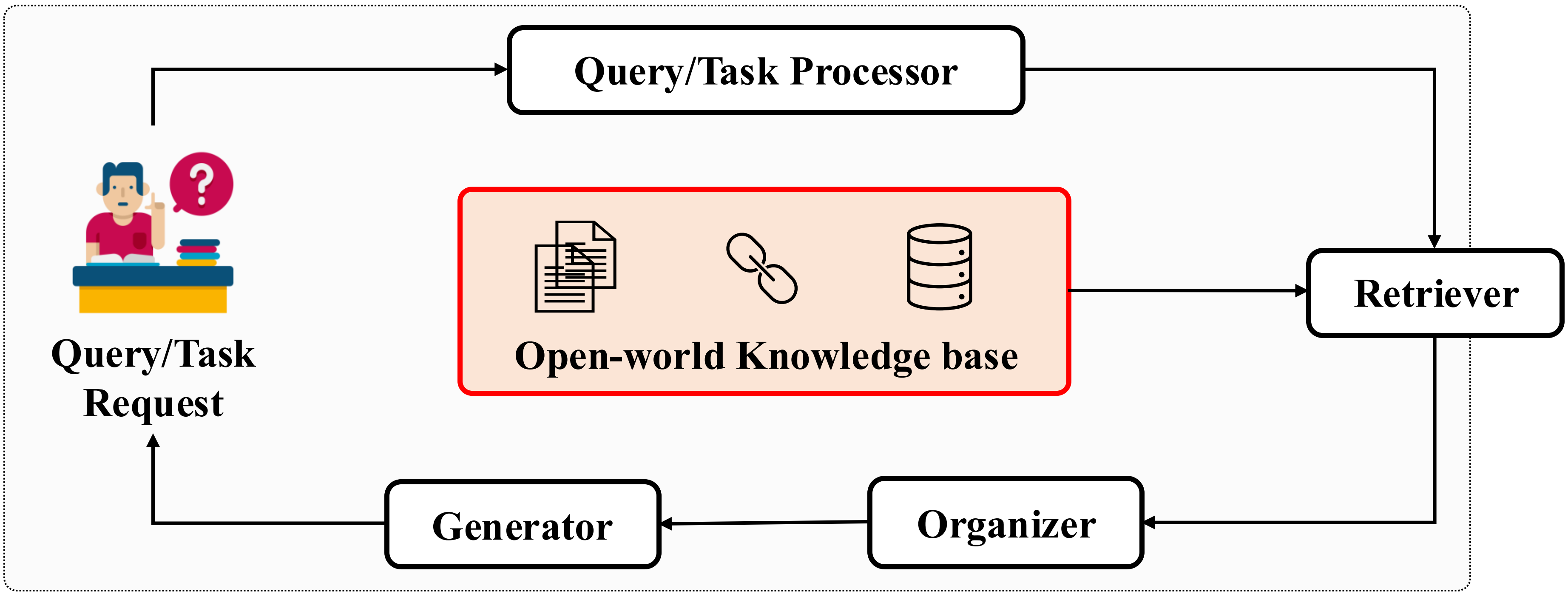
- Done via human annotation
- Popular example is the WikiData database [1]

2) Rule-Based Construction

3) LLM-Based Construction

Covered in last section

Knowledge Graph - Pipeline for GraphRAG on KGs



Knowledge Graph - GraphRAG for KGs

- A **key difference** in KG GraphRAG frameworks is the **retrieval method**
  - "*How do we retrieve relevant facts for our query?*"

- **Keys retrieval strategies:**
  - Subgraph-based
  - Traversal-based
  - GNN-based
  - Other (Agent, Semantic similarity)

Knowledge Graph - GraphRAG for KGs

- A **key difference** in KG GraphRAG frameworks is the **retrieval method**
  - "*How do we retrieve relevant facts for our query?*"

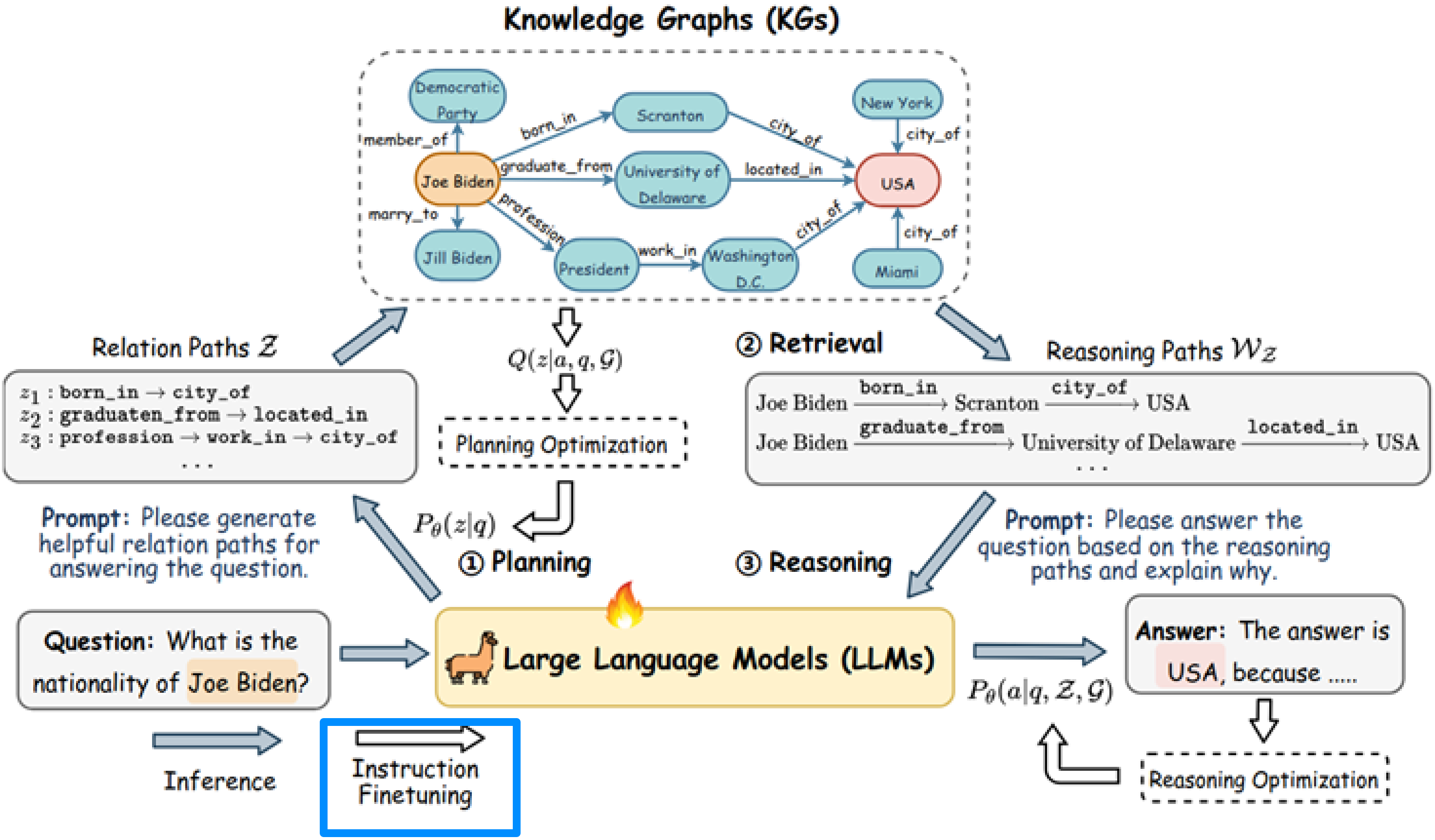
- **Keys retrieval strategies:**
  - **Subgraph-based: MindMap [1]**
  - **Traversal-based: RoG [2]**
  - **GNN-based: SubGraphRAG [3]**
  - Other (Agent, Semantic similarity)

[1] "MindMap: Knowledge Graph Prompting Sparks Graph of Thoughts in Large Language Models". ACL 2024.
[2] "Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning." ICLR 2024.
[3] "Simple is Effective: The Roles of Graphs and Large Language Models in Knowledge-Graph-Based Retrieval-Augmented Generation." ICLR 2025.

Knowledge Graph - Reasoning on Graph (RoG)

Motivation: How to extract a subset of "faithful and reliable" paths for the query?

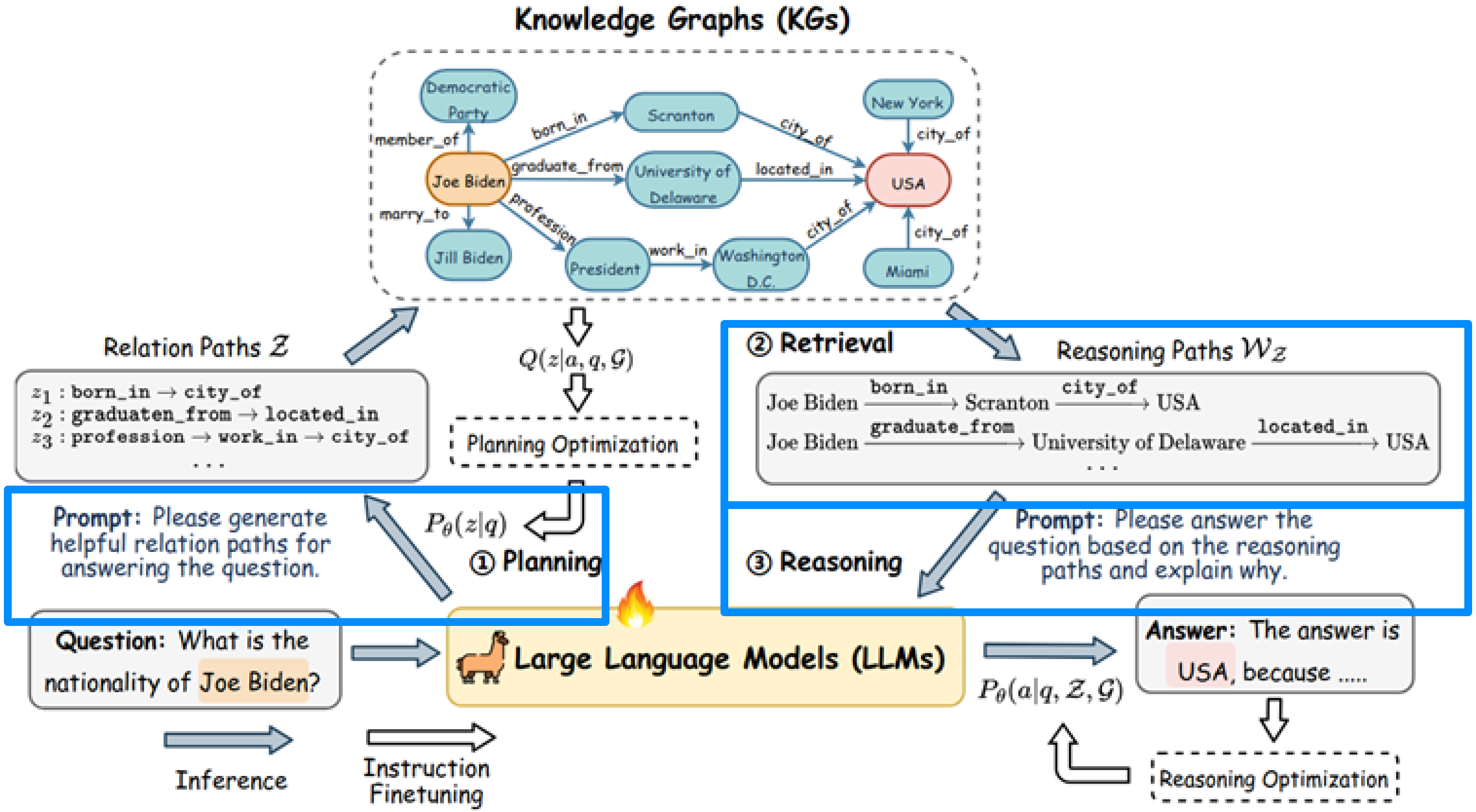
Basic Idea: Extract relevant paths from a KG for a given query



Knowledge Graph - Reasoning on Graph (RoG)

Motivation: How to extract a subset of “faithful and reliable” paths for the query?

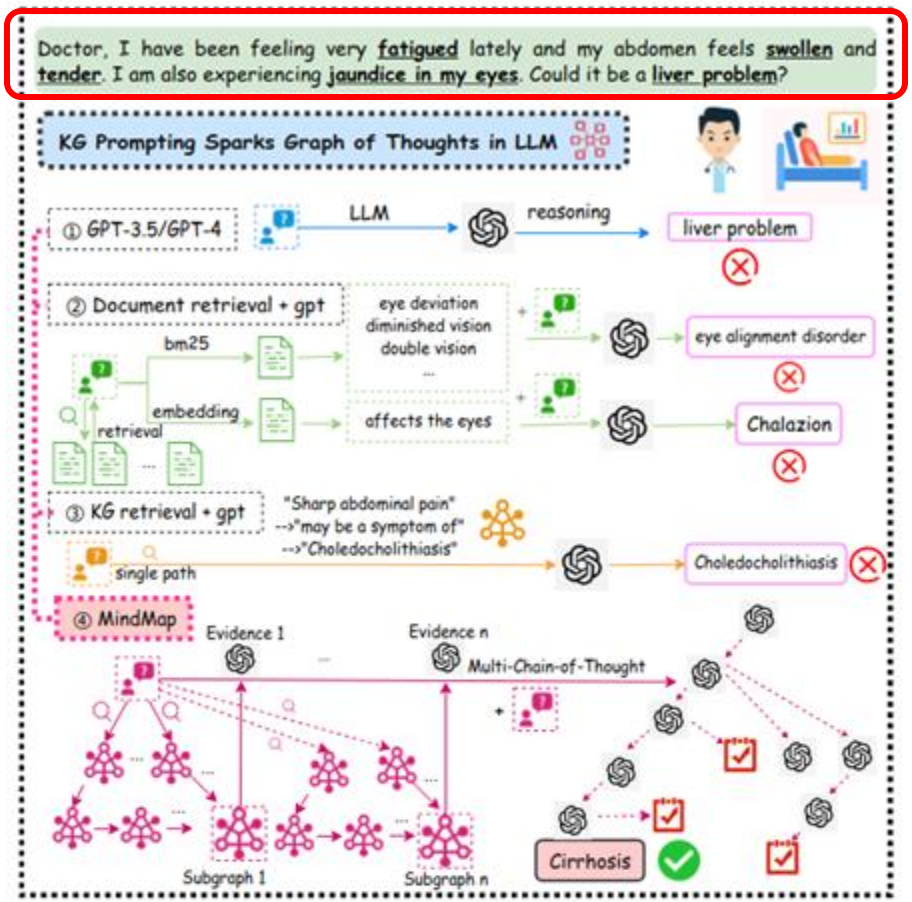
Basic Idea: Extract paths that follow <u>specific templates</u>, outputted by a LLM



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

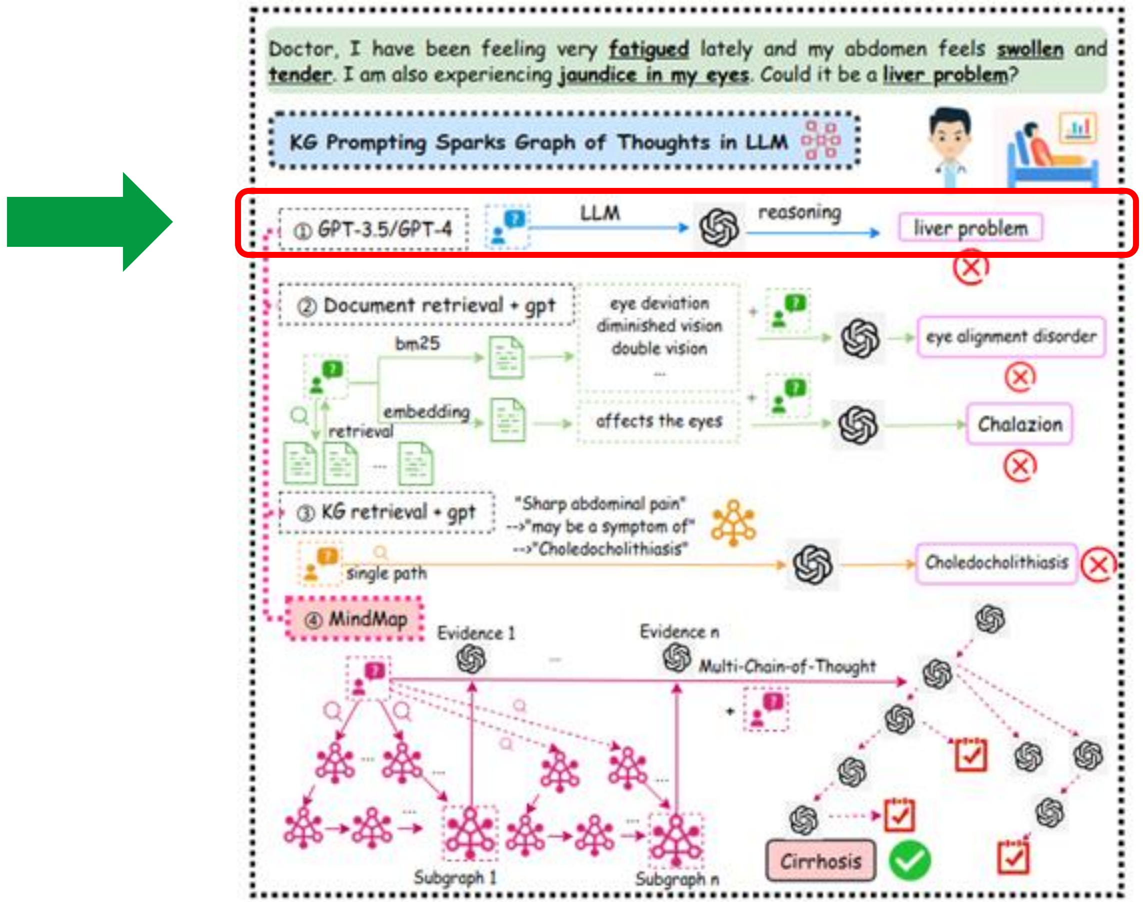
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

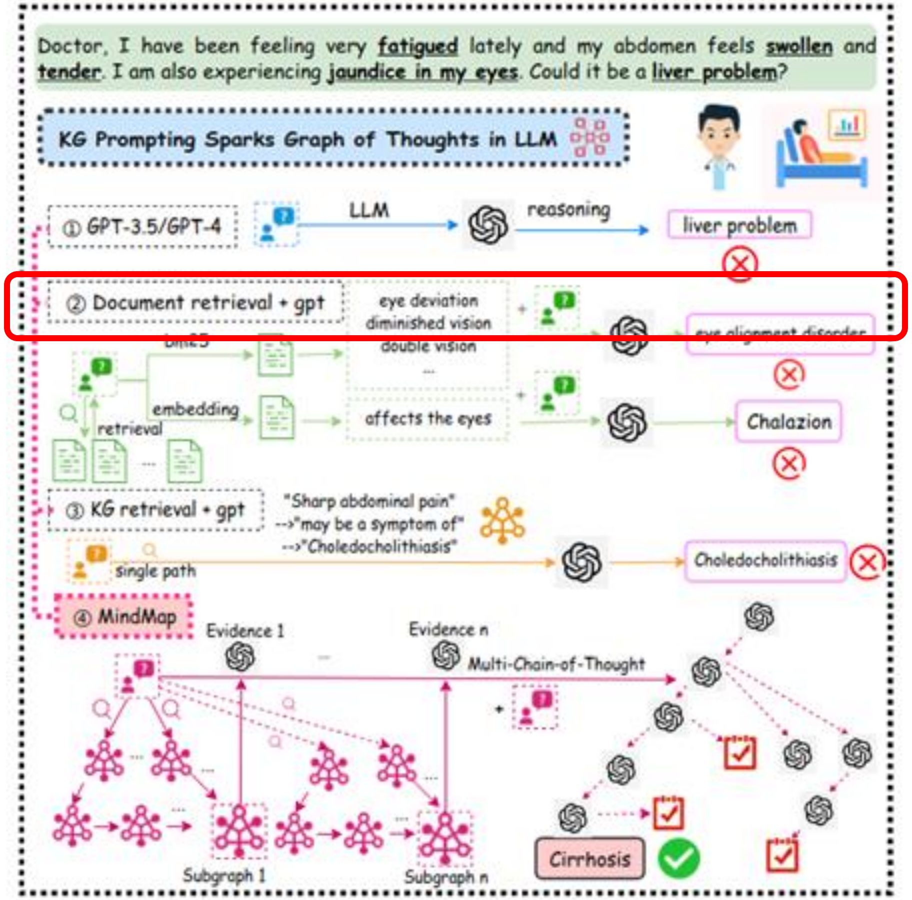
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

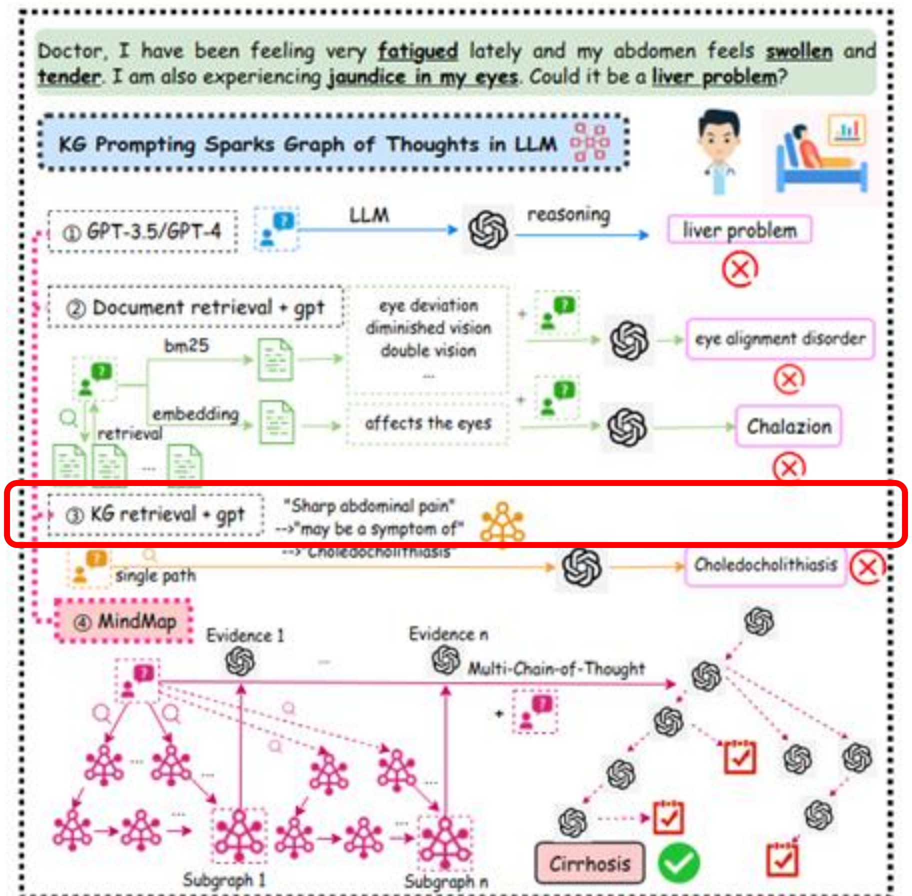
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

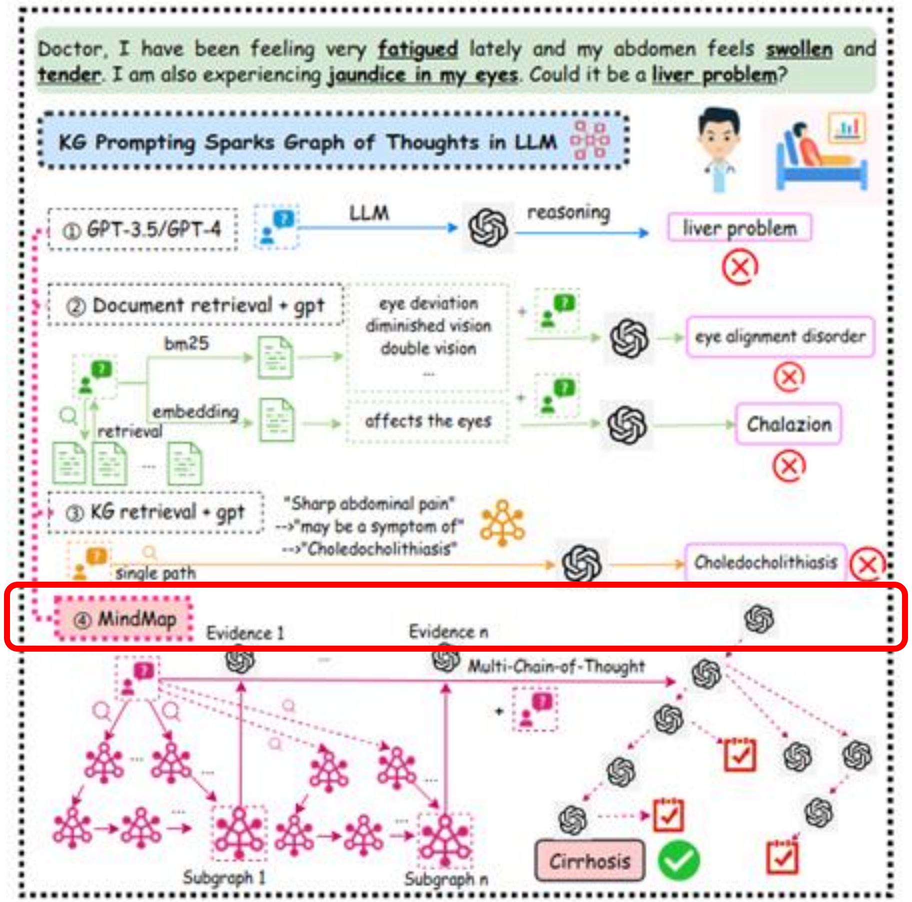
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

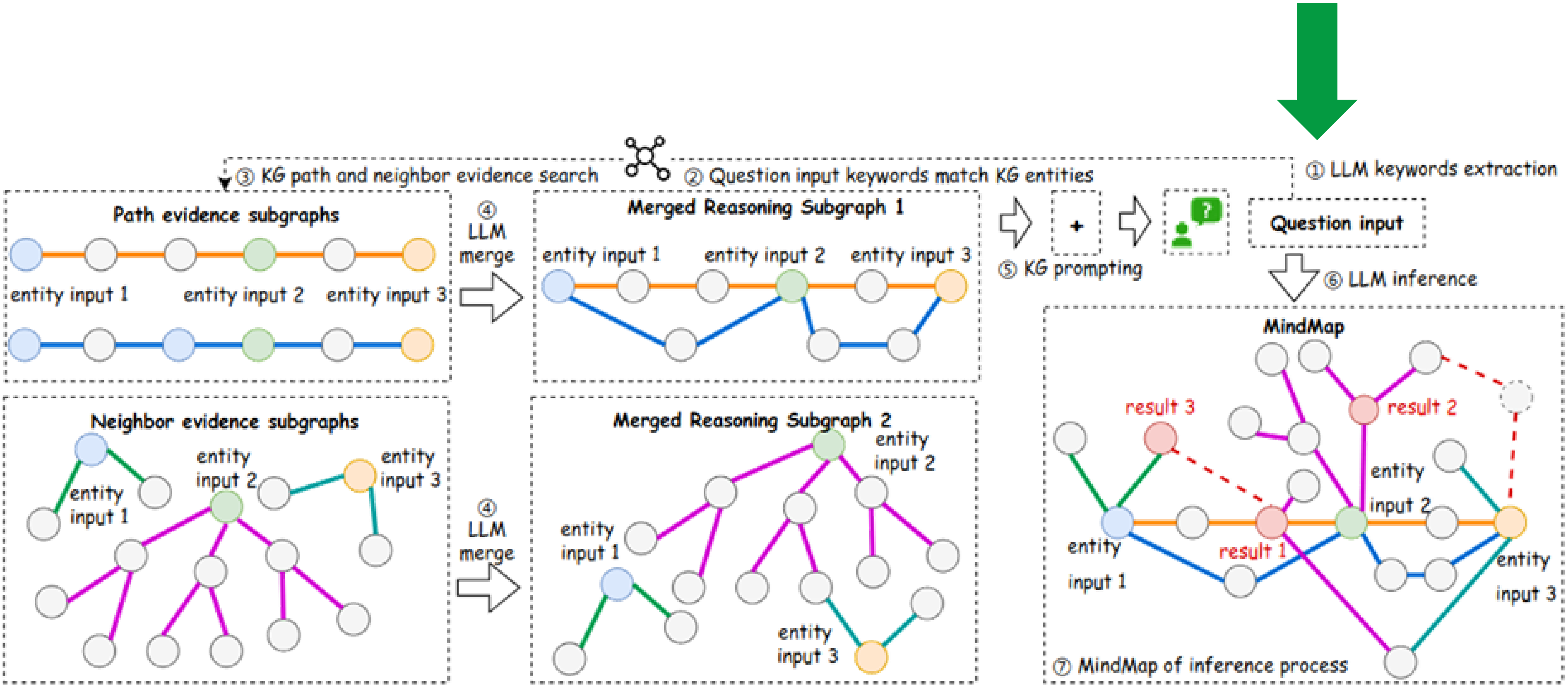
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

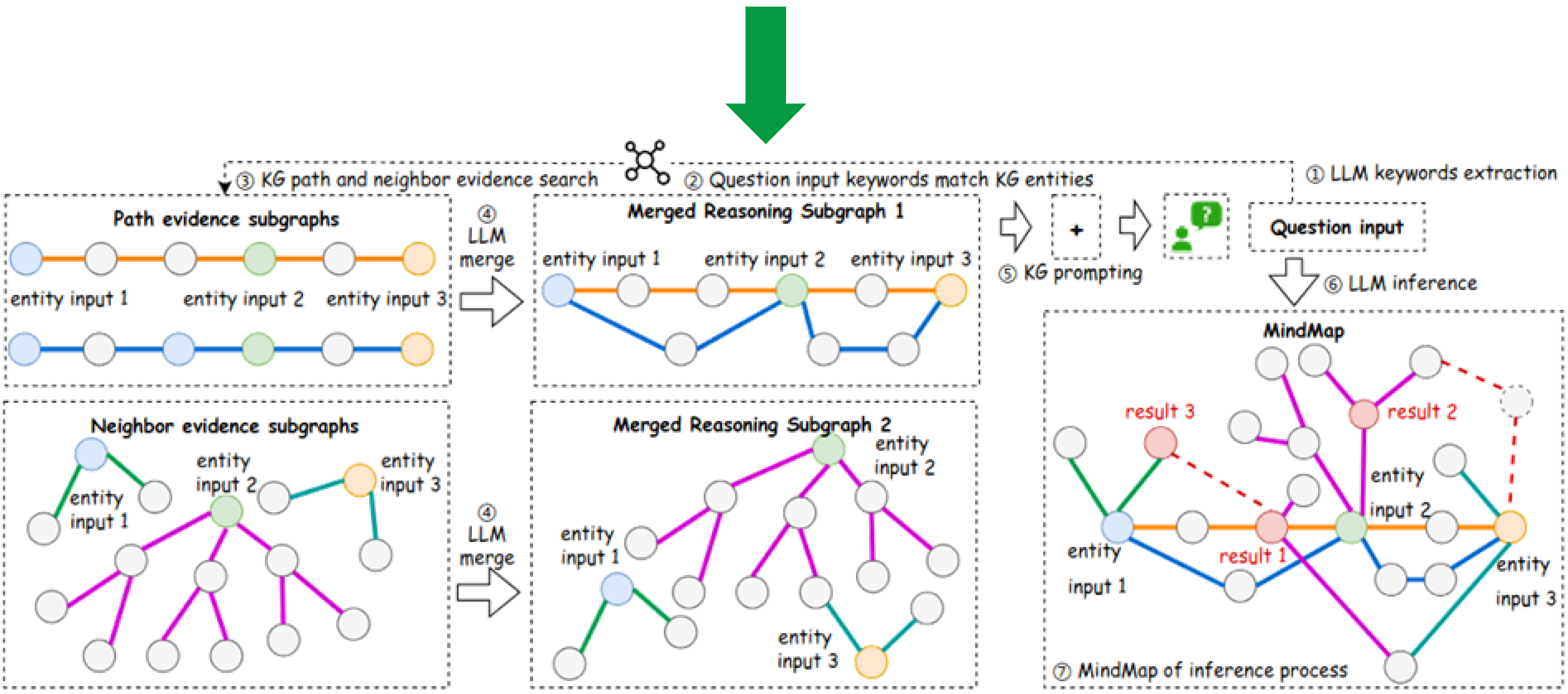
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

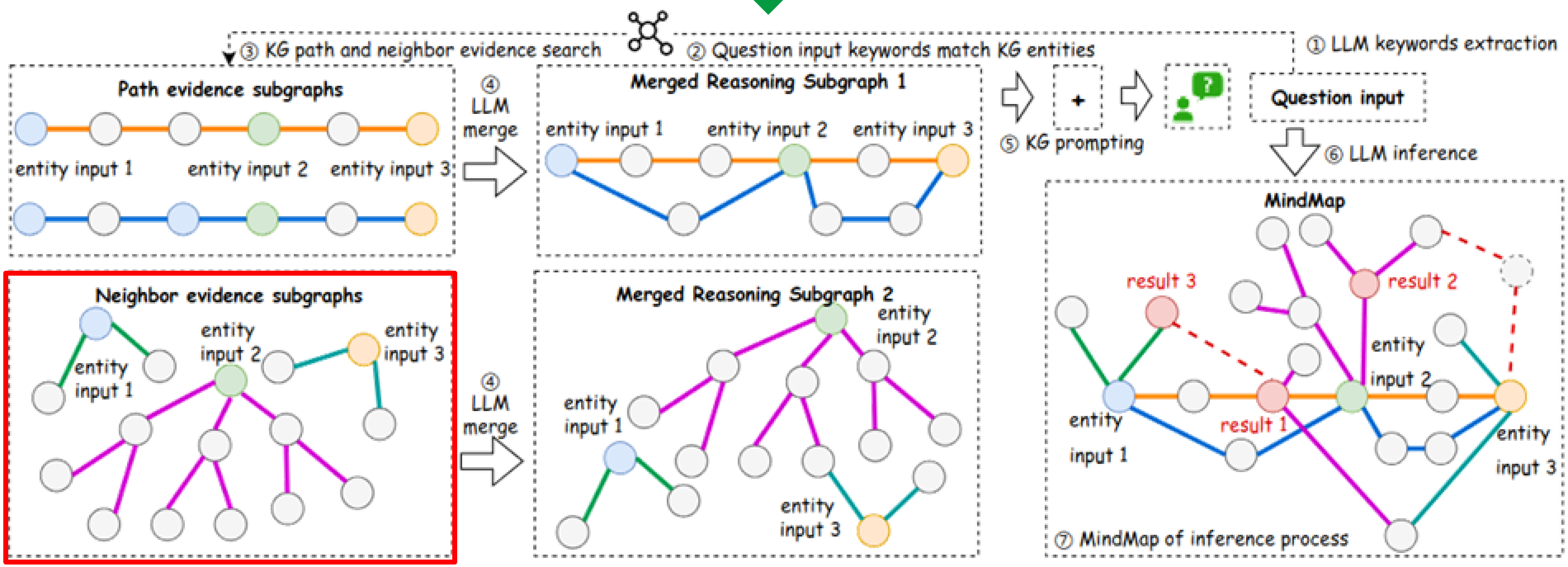
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

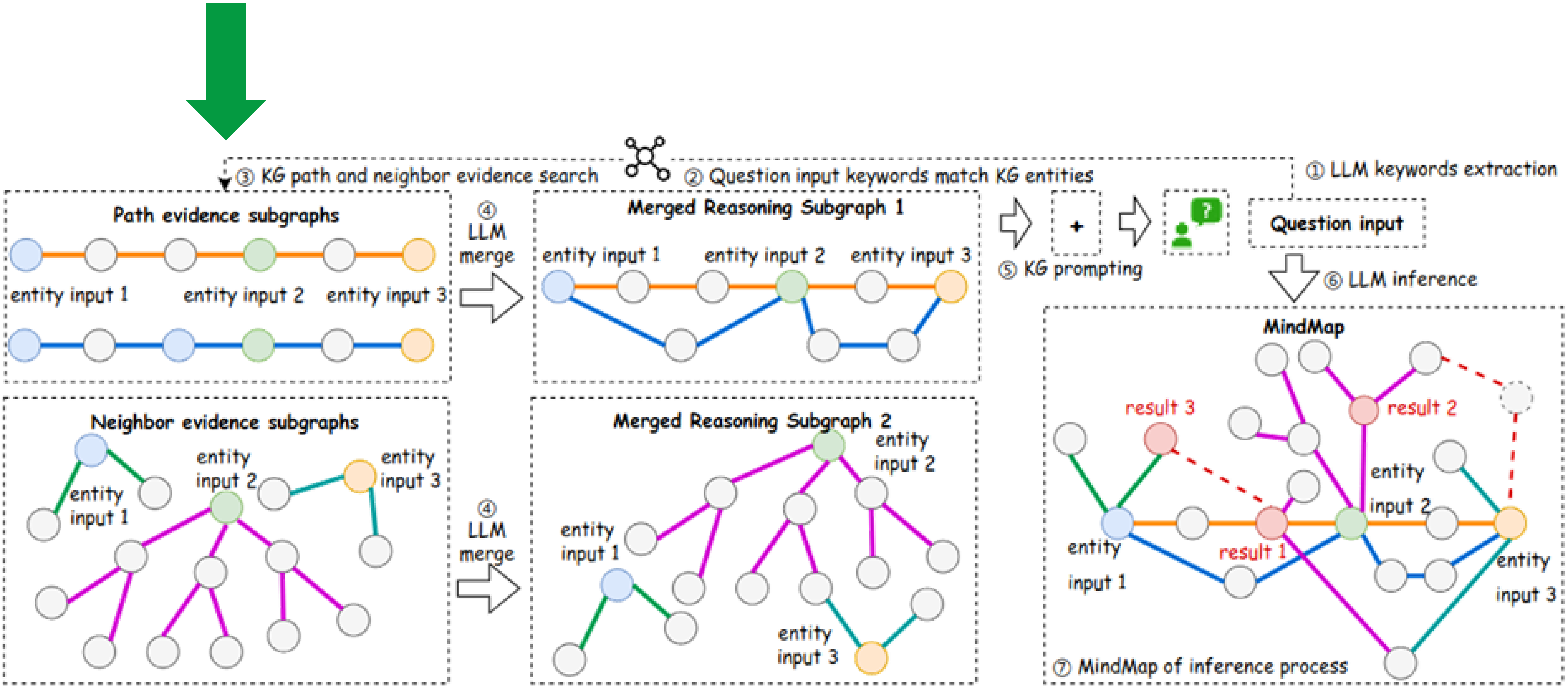
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

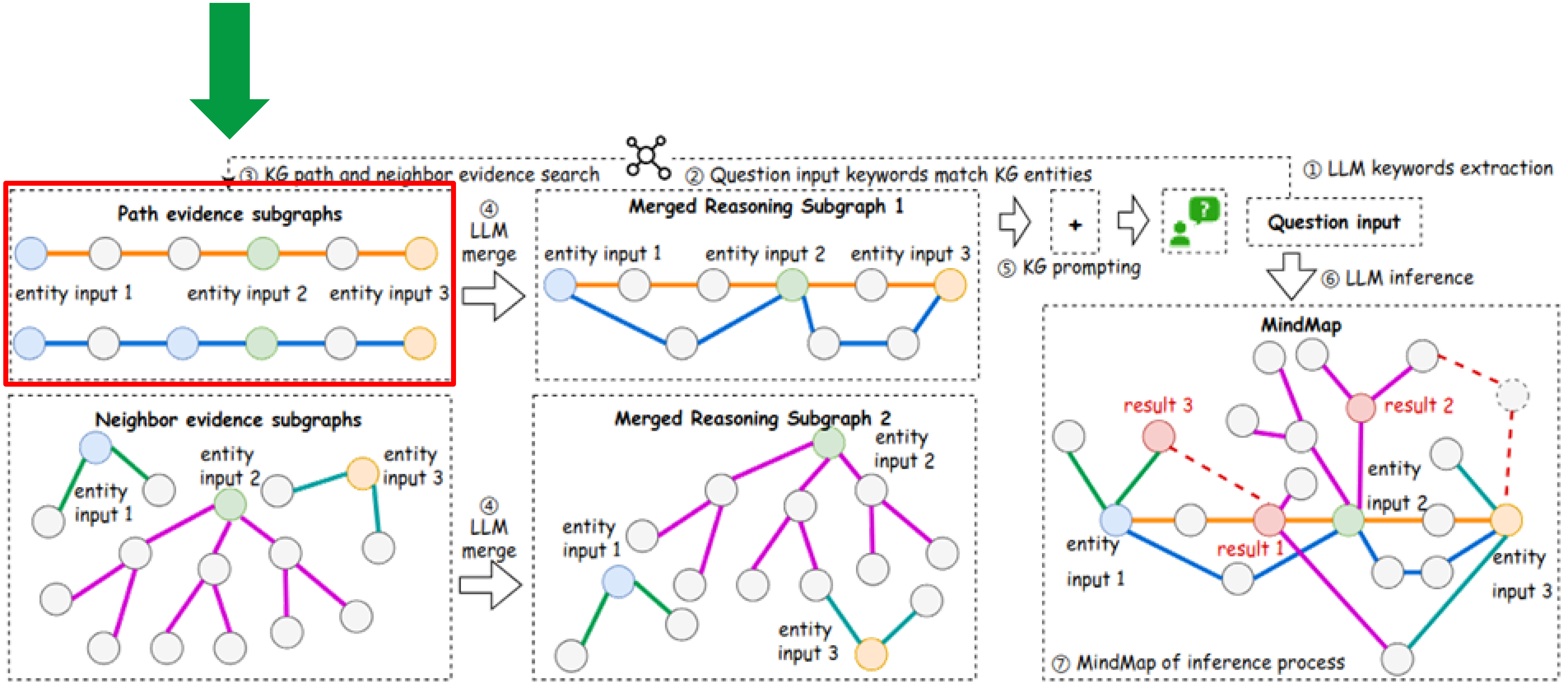
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

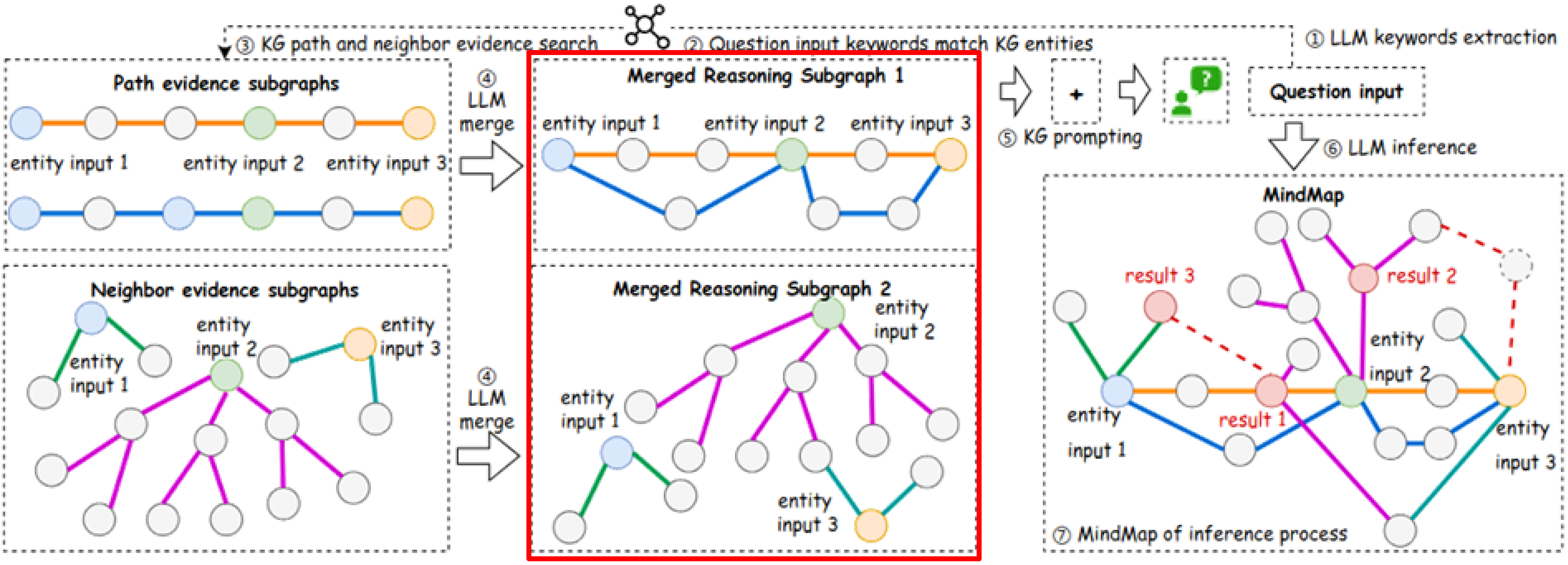
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

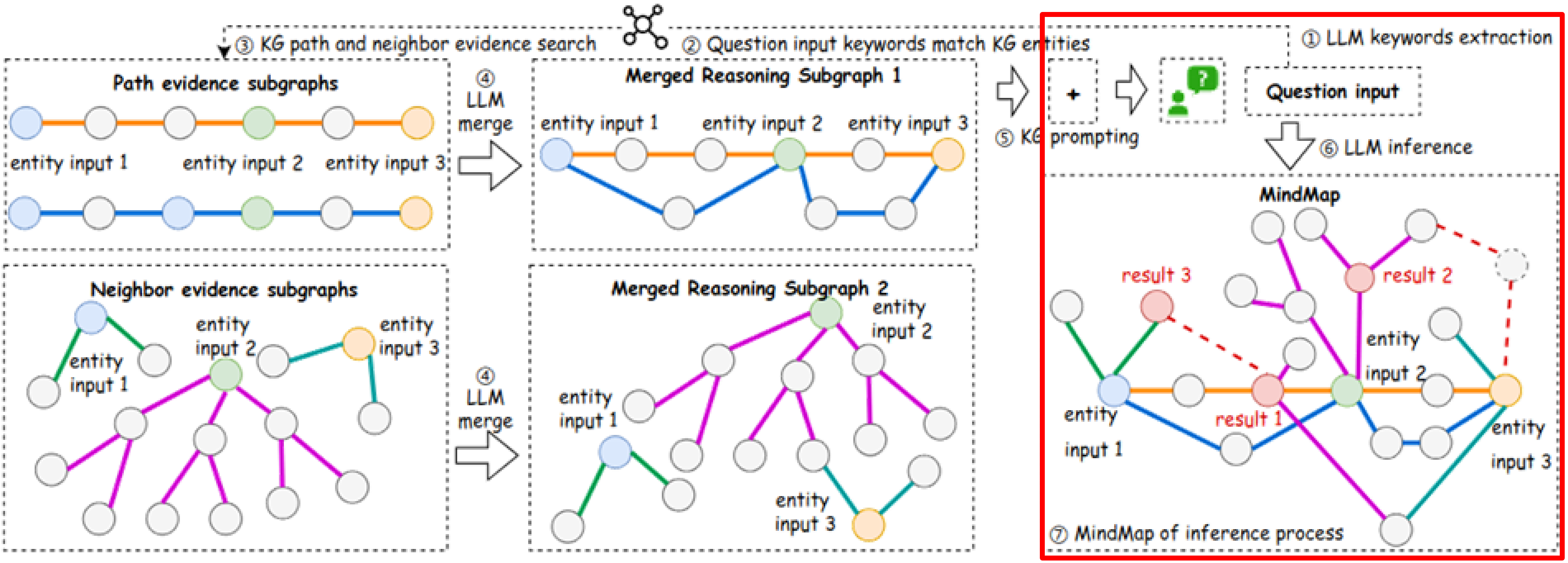
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - MindMap

Motivation: Explainable and diverse reasoning process to mitigate hallucinations

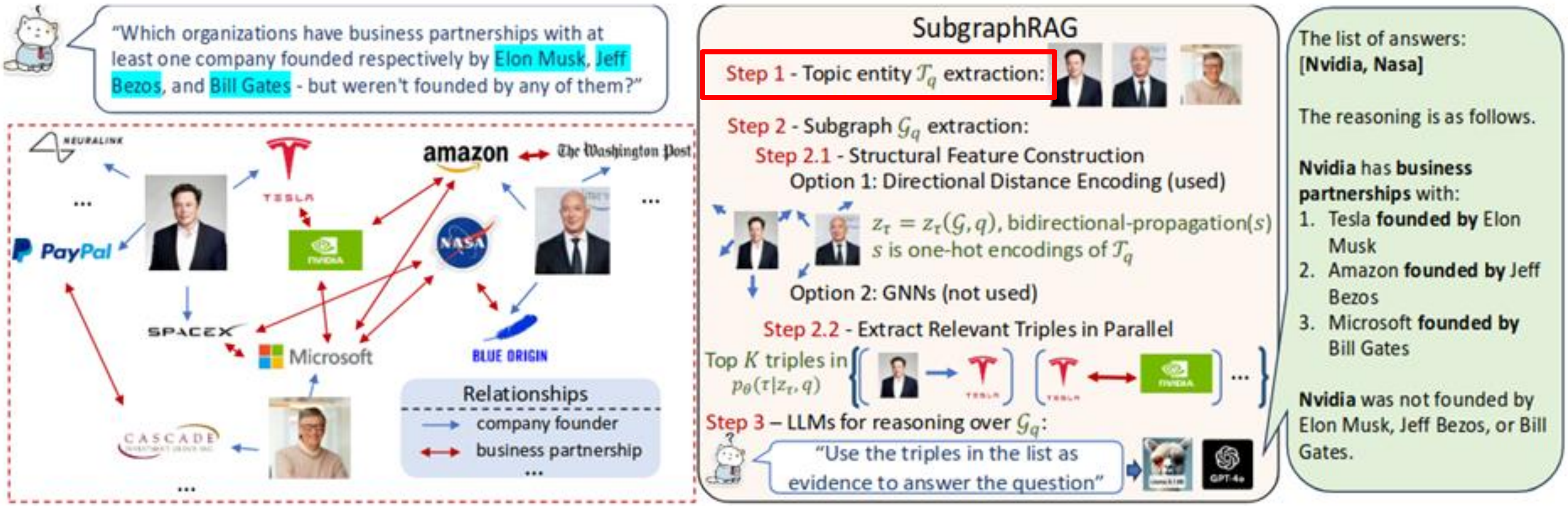
Basic Idea: For a query, extract both relevant subgraphs and paths



Knowledge Graph - SubGraphRAG

Motivation: There is a tradeoff between retrieval efficiency and reasoning abilities

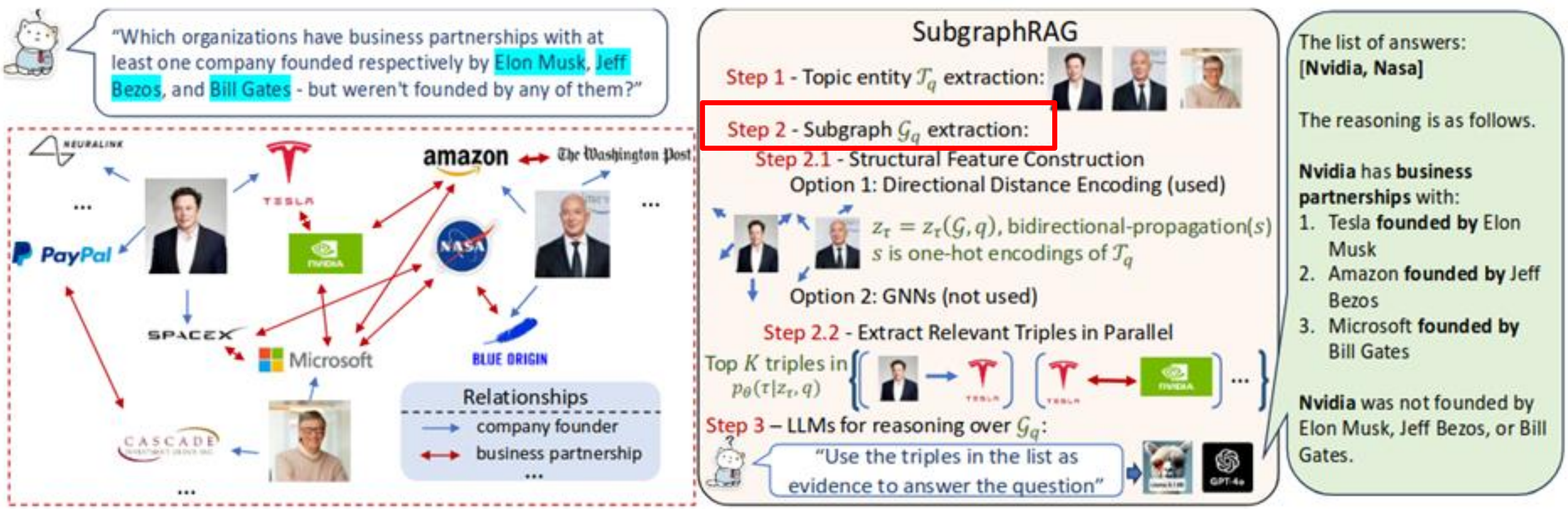
Basic Idea: Use a GNN to learn how to extract the important paths for the query



Knowledge Graph - SubGraphRAG

Motivation: There is a tradeoff between retrieval efficiency and reasoning abilities

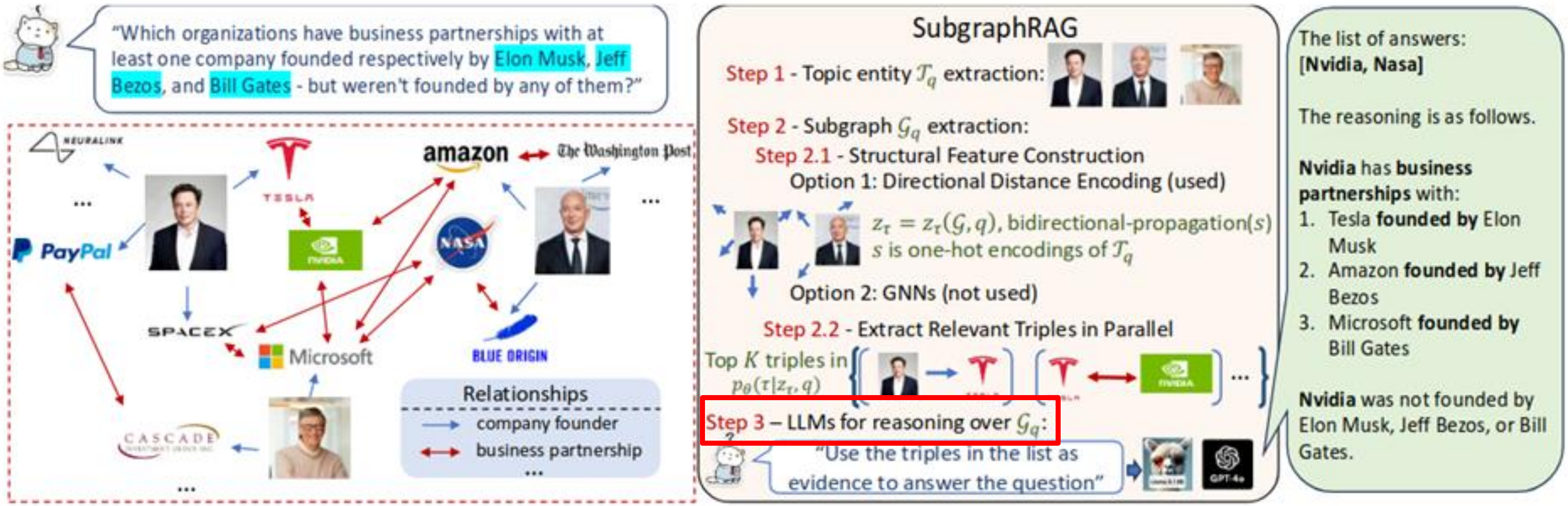
Basic Idea: Use a GNN to learn how to extract the important paths for the query



Knowledge Graph - SubGraphRAG

Motivation: There is a tradeoff between retrieval efficiency and reasoning abilities

Basic Idea: Use a GNN to learn how to extract the important paths for the query



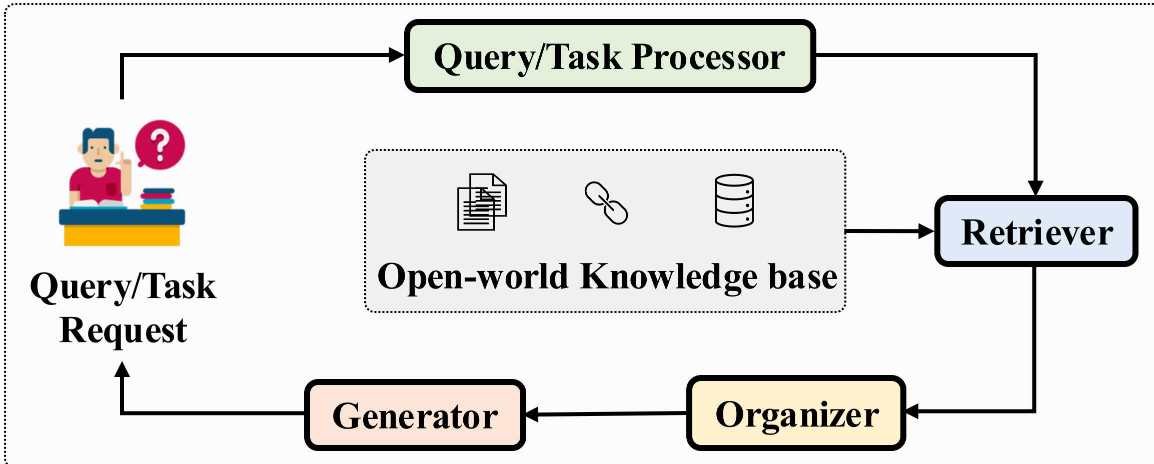
Knowledge Graph - Future Work

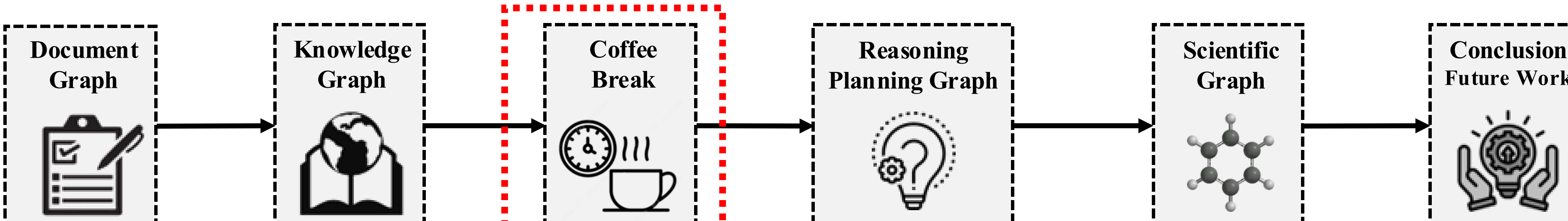
1. How to best **construct** KGs? What granularity should the node/edges be?

2. How do we **harmonize** the internal LLM knowledge and retrieved KG knowledge?

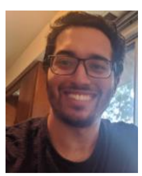
3. What's the best way of **organizing** the triples or paths for the LLM?

Outline





Haoyu Han
24 min



Harry Shomer
24 min

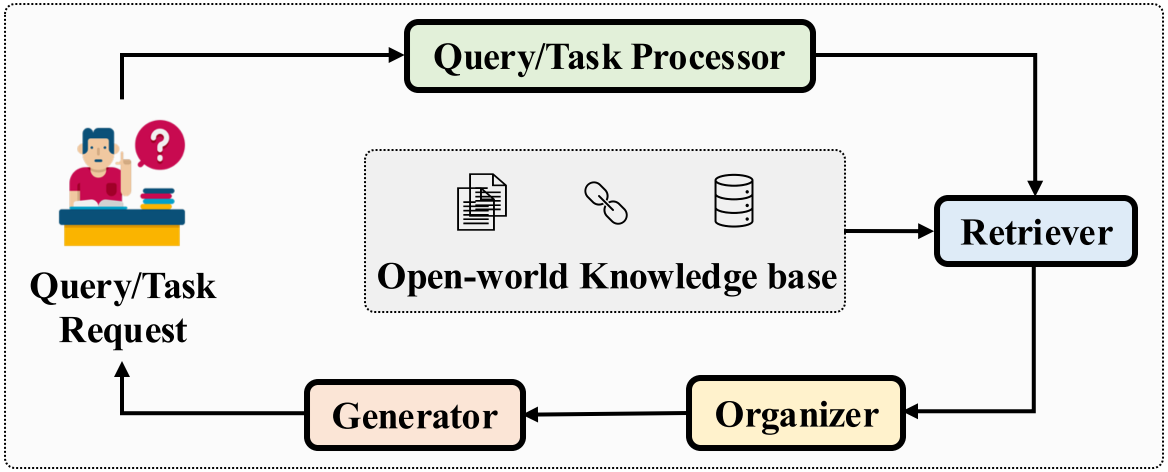
4 min

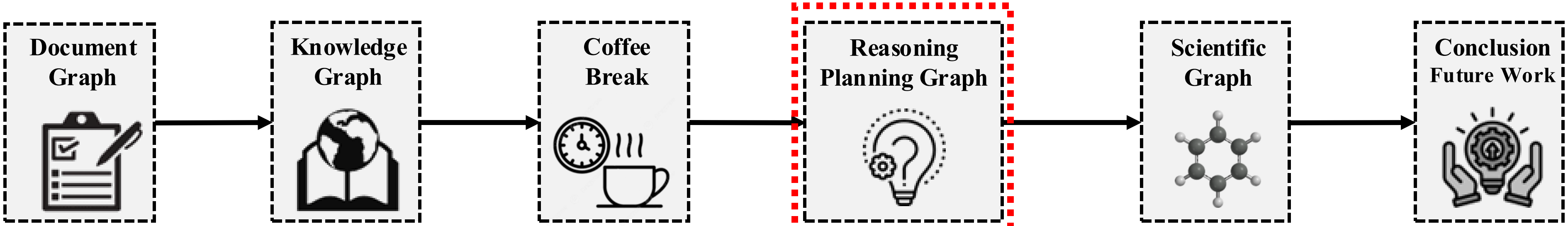
Yongjia Lei
24 min

Kai Guo
24 min

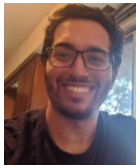
Yu Wang
5 min

Outline





Haoyu Han
24 min



Harry Shomer
24 min

4 min

Yongjia Lei
24 min

Kai Guo
24 min

Yu Wang
5 min

Reasoning & Planning Graph

What is Reasoning?

Thinking logically and systematically
Using Evidence/past experiences for drawing conclusion and decision-making

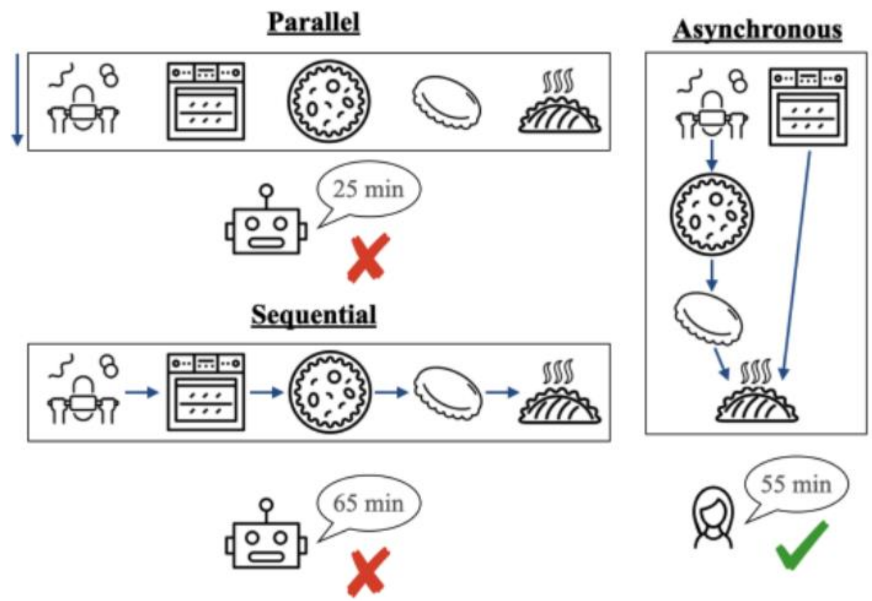
What is Planning?

Formulating a series of actions or operations to achieve a specific goal.

Reasoning and Planning are deeply interconnected in RAG

Make Calzones

Preheat the oven to 425 degrees – 10 minutes
Roll out the dough – 10 minutes
Add the filling – 15 minutes
Fold and pinch the dough – 5 minutes
Bake the calzones – 25 minutes

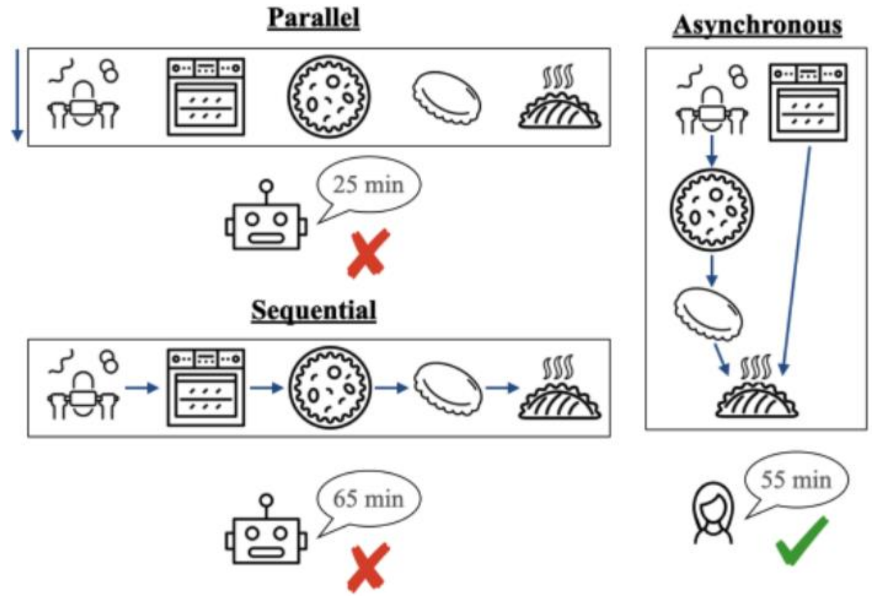


Reasoning & Planning Graph

Reasoning and Planning are deeply interconnected in RAG

Make Calzones

Preheat the oven to 425 degrees – 10 minutes
Roll out the dough – 10 minutes
Add the filling – 15 minutes
Fold and pinch the dough – 5 minutes
Bake the calzones – 25 minutes



- Retrieving task components, e.g., actions, time
- Reasoning about dependencies
- Planning the execution order

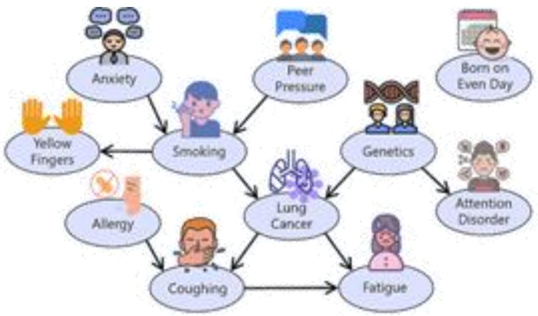
Reasoning & Planning Graph

Why Reasoning and Planning Graphs are Important in GraphRAG?

- Dependences/sequences to capture relations, e.g., Causal and Resource Dependency
- Structuring the Retrieval Process

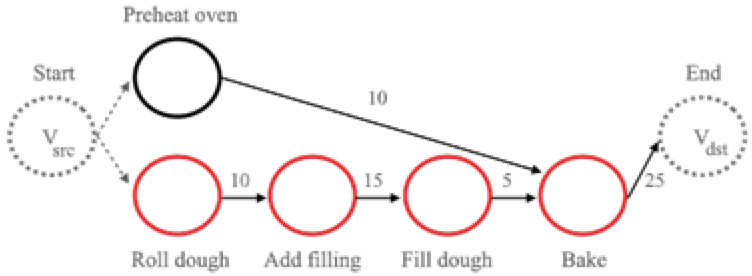
Resource Dependency

[Shen et al. 2024]



Causal Dependency

[LUCAS 2024]



Temporal Dependency

[Lin 2024]

Common Dependencies in Graph Construction

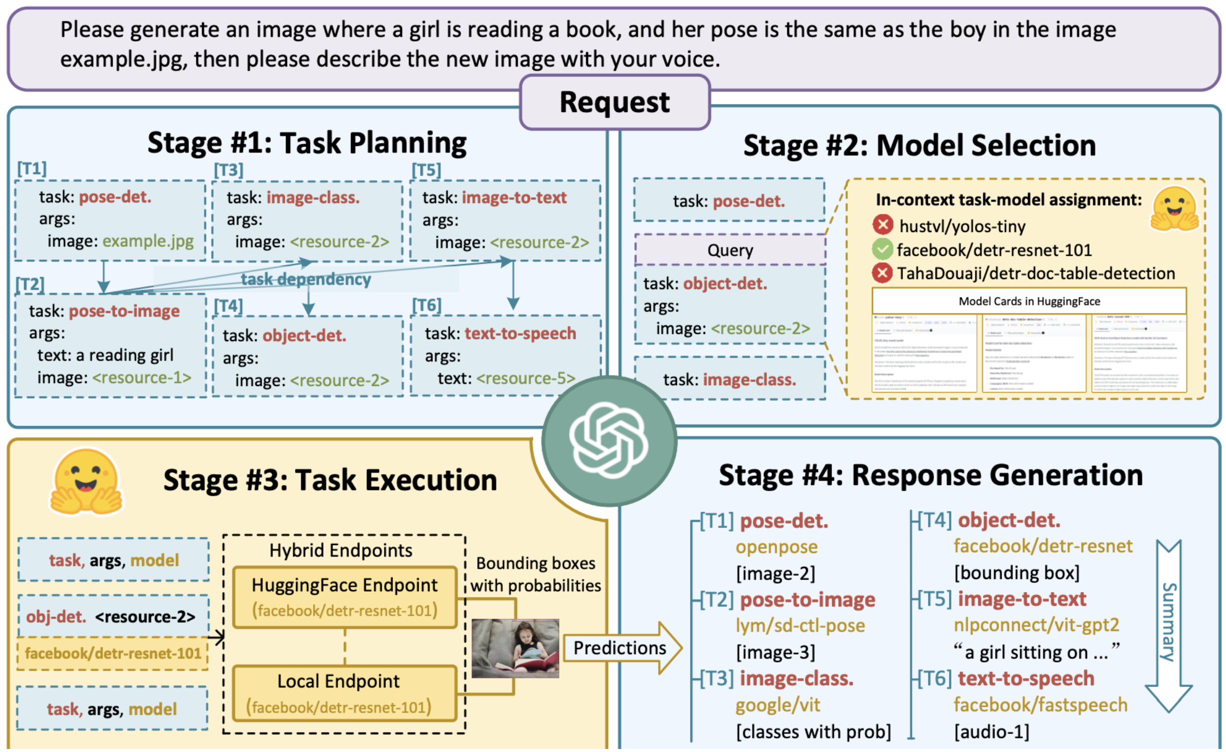
Reasoning & Planning Graph – Task Planning

Task Planning: Retrieve/generate plan of steps/tools in graph format

Planning Graph:

- Capture dependencies and execution orders
- Guide APIs retrieval
- Guide inter-model cooperation

HuggingGPT: Generation-based Planning



Reasoning & Planning Graph – Task Planning

How to enable LLMs conduct task planning?

- Specification-based Instruction
- Demonstration-based Parsing

Prompt	
#1 Task Planning Stage - The AI assistant performs task parsing on user input, generating a list of tasks with the following format: [{"task": task, "id", task_id, "dep": dependency_task_ids, "args": {"text": text, "image": URL, "audio": URL, "video": URL}}]. The "dep" field denotes the id of the previous task which generates a new resource upon which the current task relies. The tag "<resource>-task_id" represents the generated text, image, audio, or video from the dependency task with the corresponding task_id. The task must be selected from the following options: {{ *Available Task List* }}. Please note that there exists a logical connections and order between the tasks. In case the user input cannot be parsed, an empty JSON response should be provided. Here are several cases for your reference: {{ *Demonstrations* }}. To assist with task planning, the chat history is available as {{ *Chat Logs* }}, where you can trace the user-mentioned resources and incorporate them into the task planning stage.	
Demonstrations	
Can you tell me how many objects in e1.jpg?	[{"task": "object-detection", "id": 0, "dep": [-1], "args": {"image": "e1.jpg" }}]
In e2.jpg, what's the animal and what's it doing?	[{"task": "image-to-text", "id": 0, "dep":[-1], "args": {"image": "e2.jpg" }}, {"task":"image-cls", "id": 1, "dep": [-1], "args": {"image": "e2.jpg" }}, {"task":"object-detection", "id": 2, "dep": [-1], "args": {"image": "e2.jpg" }}, {"task": "visual-quesrion-answering", "id": 3, "dep":[-1], "args": {"text": "what's the animal doing?", "image": "e2.jpg" }}]
First generate a HED image of e3.jpg, then based on the HED image and a text "a girl reading a book", create a new image as a response.	[{"task": "pose-detection", "id": 0, "dep": [-1], "args": {"image": "e3.jpg" }}, {"task": "pose-text-to-image", "id": 1, "dep": [0], "args": {"text": "a girl reading a book", "image": "<resource>-0" }}]

Reasoning & Planning Graph – Task Planning

Challenges of Generation-based Task Planning

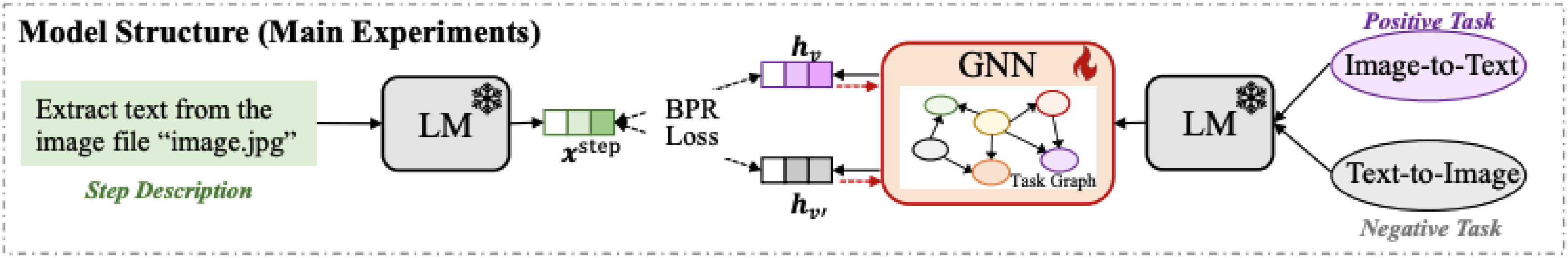
- Hallucinate non-existent tasks or dependencies (edges)

- Not invariant to graph isomorphism

- Performance degrades as the task graph scales

Reasoning & Planning Graph – Task Planning

Retrieval-based Task Planning



- Small frozen LM embeds sub-steps/task nodes in the pre-built task graph
- A GNN is applied over the task graph
  - Propagate information via pre-built dependencies
  - Refine node embeddings
- Retrieve matching tasks in the pre-built graph for sub-steps via similarity

Reasoning & Planning Graph – Multi-Step Reasoning

Multi-step Reasoning: Solving problems via multiple calculations/steps

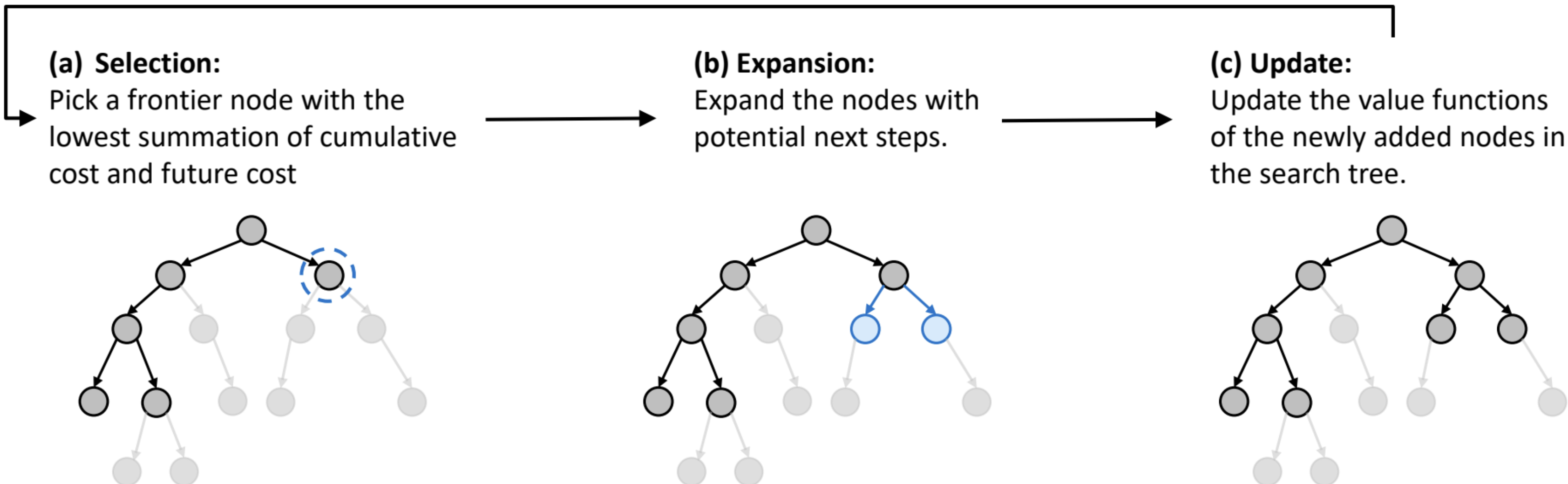
Question: Which publications from Altair Engineering authors focus on improving directional sensitivity across a wide range of frequencies?

↓

Institution *< Altair Engineering >* → **Author** → **Paper** *< improving directional sensitivity across a wide range of frequencies >*

Reasoning & Planning Graph – Multi-Step Reasoning

Toolchain*: Efficient Action Space Navigation



Multi-step Reasoning → Graph Search; Node → API Function Call; Edge → Possible Transition

Monte Carlo Tree Search vs. A* Search

- A*: one-step based on cost function

$f(n) = g(n) + h(n)$

  - $g(n)$ cumulative cost from the root node to the current node n
  - $h(n)$ heuristic estimation of the future cost from node n to the goal

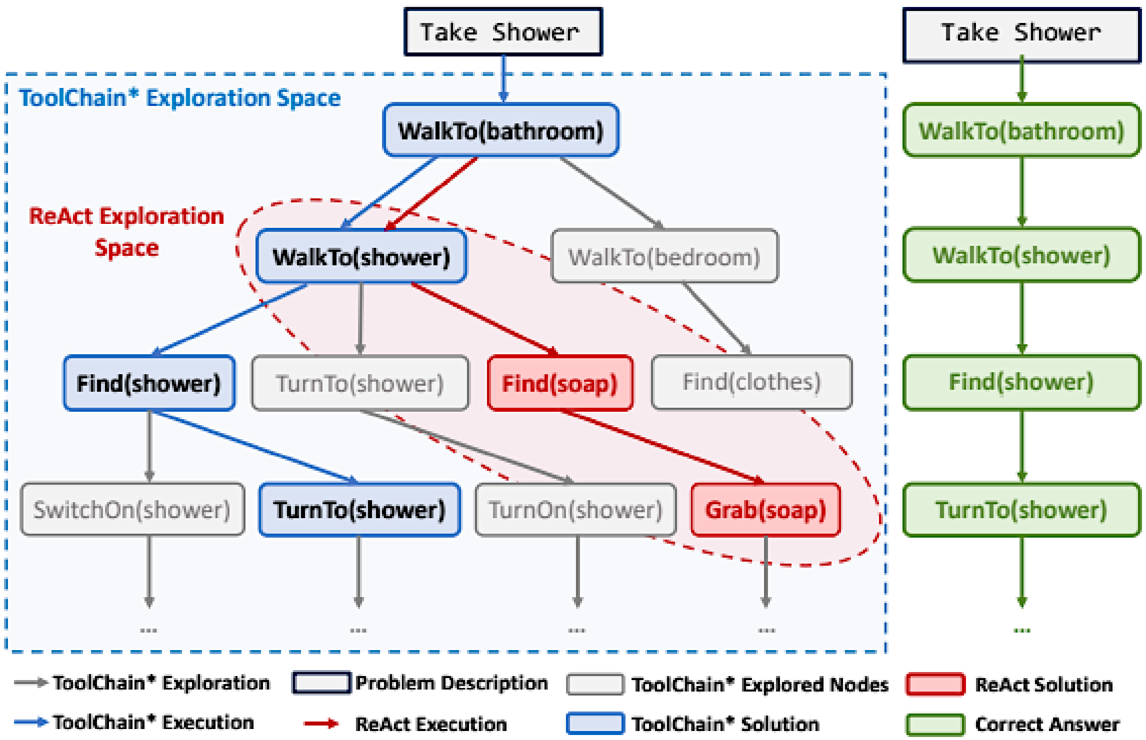
MCTS: Simulates many random rollouts to terminal states $Q(n,a) + c\sqrt{\frac{\log N(n)}{N(n,a)}}$

- $Q(n,a)$ average reward from history
- $\sqrt{\frac{\log N(n)}{N(n,a)}}$ encourages less-explored actions

Toolchain*: Efficient action space navigation in large language models with A* search. ICLR 2024

Reasoning & Planning Graph – Multi-Step Reasoning

Case Study Comparison

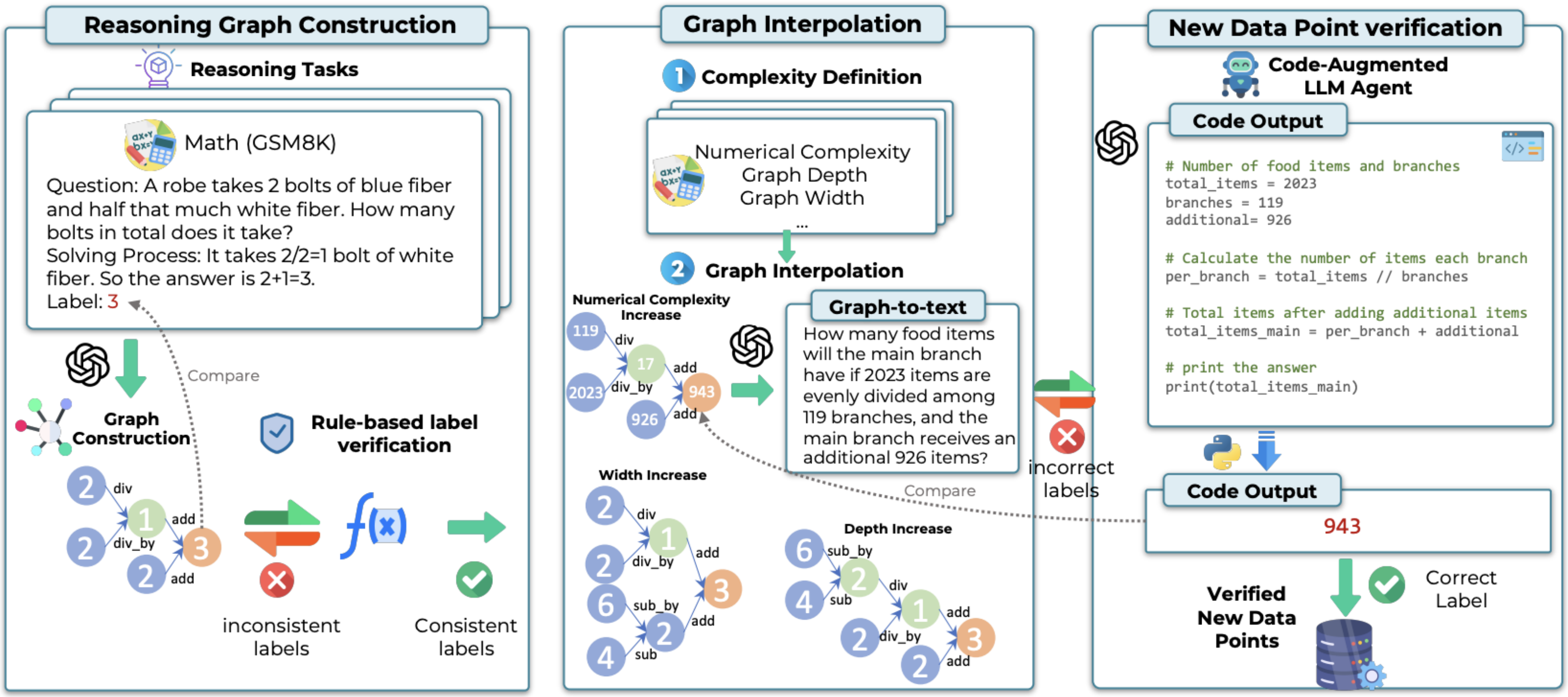


- Structured Exploration Instead of Greedy Paths
- Cost-guided Retrieval with Reasoning Flow

ToolChain* retrieves and reasons over a dynamically growing action graph

Reasoning & Planning Graph – Multi-Step Reasoning

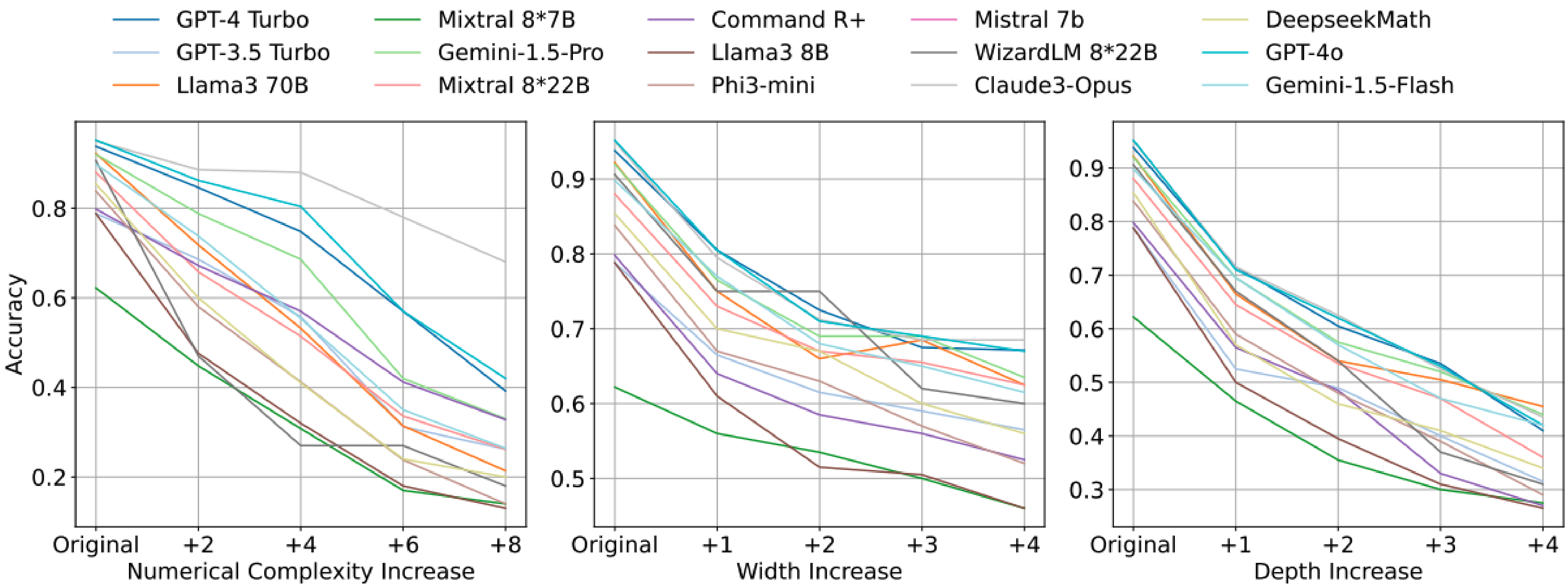
DARG: Dynamic Evaluation via Adaptive Reasoning Graph

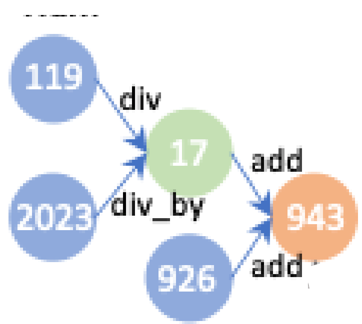


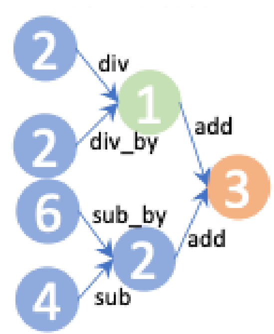
Reasoning graphs are powerful for reasoning ability evaluation

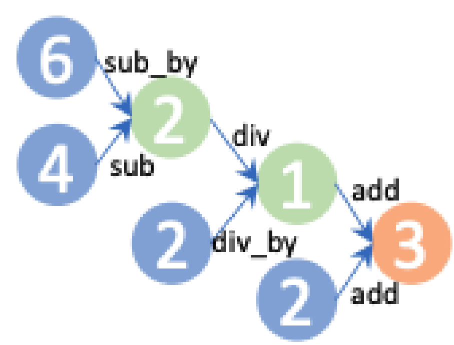
- Enable Structural Complexity Control
- Make LLM Reasoning Observable and Measurable
- Answer questions by retrieving underlying reasoning graph – Logic Fetching

Reasoning & Planning Graph – Multi-Step Reasoning



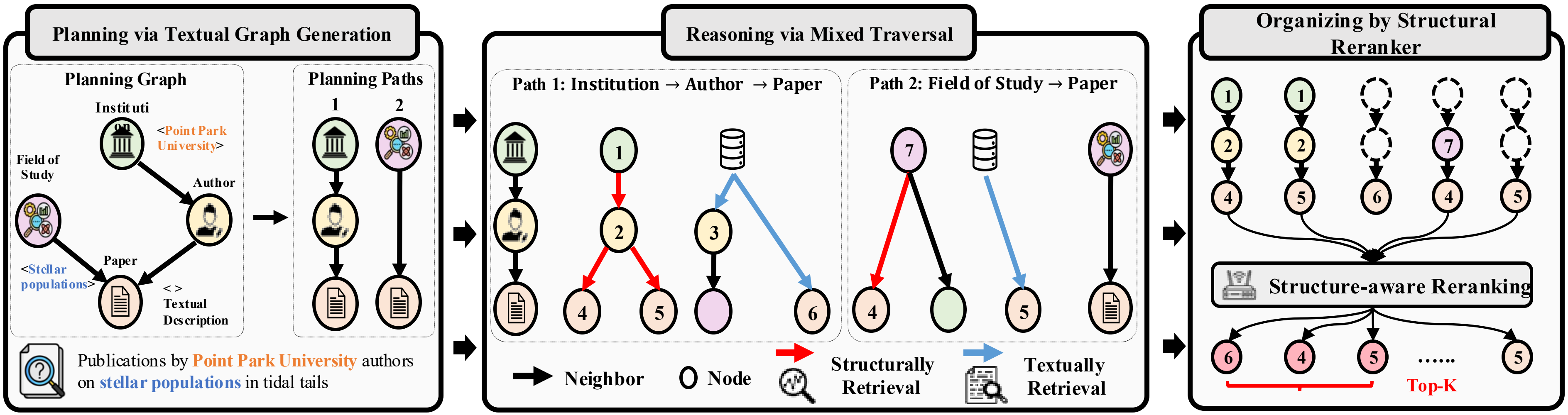






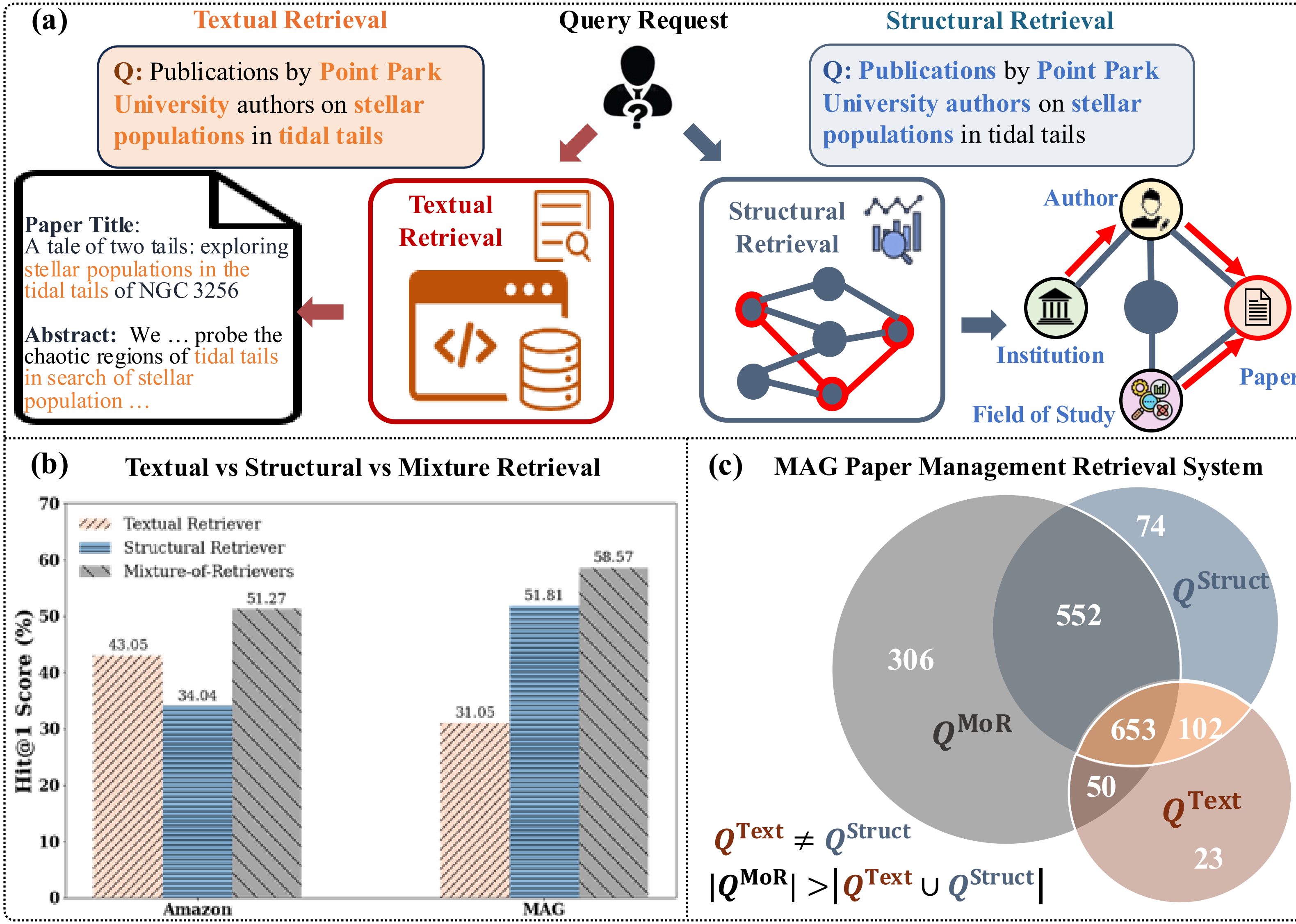
Reasoning & Planning Graph – Augment Retrieval Itself

MoR - Mixture of Structural and Textual Retrieval



- **Planning** - Given a query, generate its planning graph
- **Reasoning** - Mixed traversal guided by generated planning graph
  - Structural retrieval via graph traversal
  - Textual retrieval via textual matching
- **Organizing** - Structure-aware Rerank to select top-k candidates

Reasoning & Planning Graph – Augment Retrieval Itself



Reasoning & Planning Graph – Augment Retrieval Itself

Interleaved Reasoning and Retrieval via Reinforcement Learning

What is a suitable cloud server configuration on AWS Bedrock for training a LLaMA70B under $75/hour?

↓ Reasoning

<Reason> The user wants to train LLaMA-70B, which requires large-memory GPUs (≥ 80GB), multi-GPU parallelism, and high-speed networking.
<Retriever Type> Semantic Matching
<Input Query> GPU-Optimized Instances, LLaMA-70B

↓ Retrieving

<Retrieve> Capability Matching
<Info> AWS: 8×NVIDIA H100 80GB GPUs, $98/hour

↓ Reasoning

<Reason> The user may need a hybrid solution combining cheaper instances with memory optimization strategies.
<Retriever Type> Optimization Strategy Retriever
<Input Query> Optimization techniques for LLaMA-70B

↓ Retrieving

<Retrieve> Optimization Strategy Retriever (LLaMA-70B)
<Info> 4-bit quantization enable training LLaMA-70B on 16 × A100 40GB GPUs within $70/hour range.

Reasoning & Planning Graph – Augment Retrieval Itself

Interleaved Reasoning and Retrieval via Reinforcement Learning

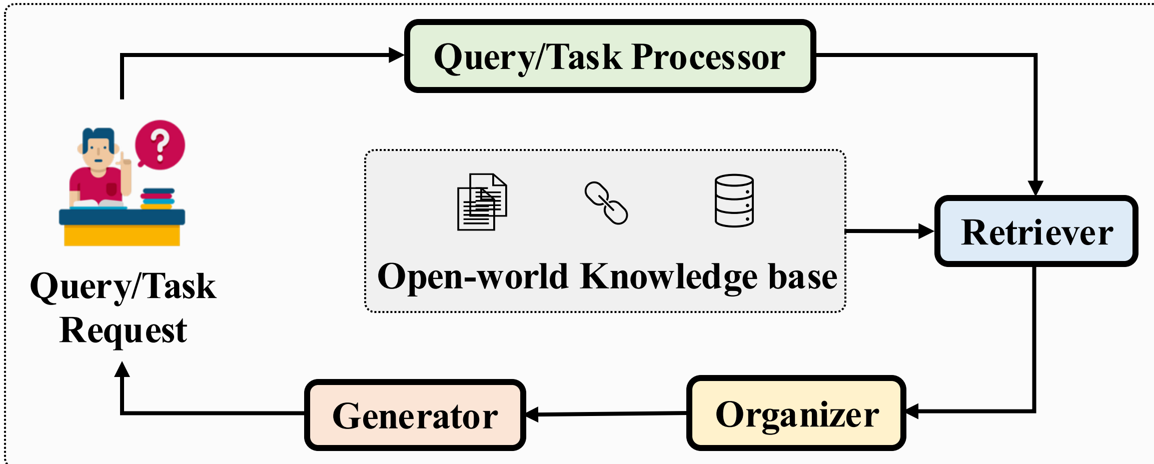
- Multi-turn reasoning with real-time search (<think>, <search>, <information> tokens)
- Retrieved token masking for stable RL training
- Simple outcome-based reward to supervise the reasoning + retrieval behavior.

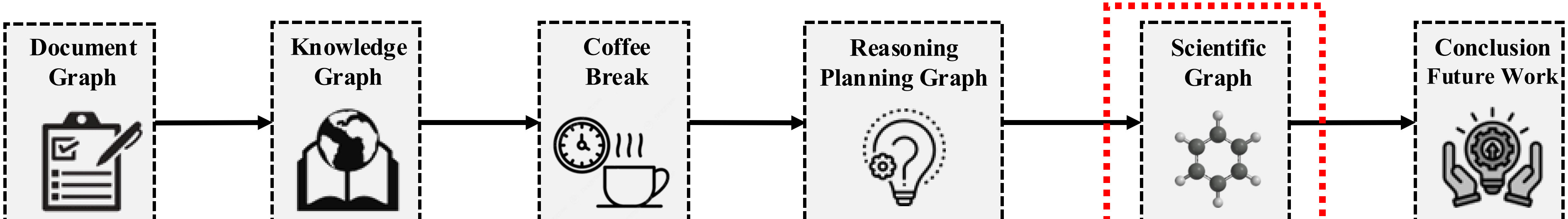
Algorithm 1 LLM Response Rollout with Multi-Turn Search Engine Calls

Require: Input query x, policy model π_θ, search engine $\mathcal{R}$, maximum action budget B.
Ensure: Final response y.

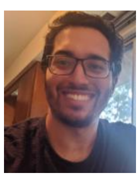
Initialize rollout sequence $y \leftarrow \emptyset$
Initialize action count $b \leftarrow 0$
while $b < B$ **do**
    Initialize current action LLM rollout sequence $y_b \leftarrow \emptyset$
    while True **do**
        Generate response token $y_t \sim \pi_\theta(\cdot \mid x, y + y_b)$
        Append y_t to rollout sequence $y_b \leftarrow y_b + y_t$
        if y_t in [</search>, </answer>, <eos>] **then** break
        end if
    end while
    $y \leftarrow y + y_b$
    if <search> </search> detected in y_b **then**
        Extract search query $q \leftarrow$ Parse(y_b, <search>, </search>)
        Retrieve search results $d = \mathcal{R}(q)$
        Insert d into rollout $y \leftarrow y +$ <information>d</information>
    else if <answer> </answer> detected in y_b **then**
        return final generated response y
    else
        Ask for rethink $y \leftarrow y +$ "My action is not correct. Let me rethink."
    end if
    Increment action count $b \leftarrow b + 1$
end while
return final generated response y

Outline






Haoyu Han
24 min


Harry Shomer
24 min

5 min


Yongjia Lei
24 min

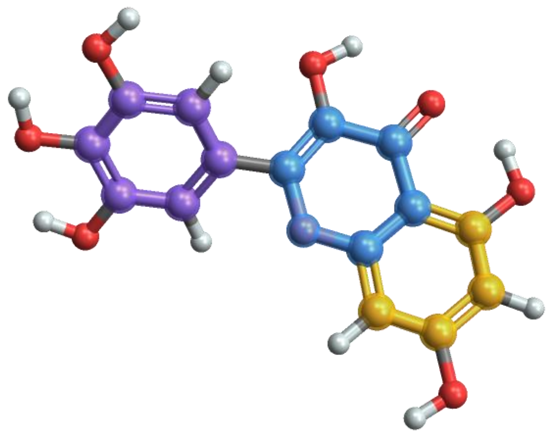

Kai Guo
24 min


Yu Wang
5 min

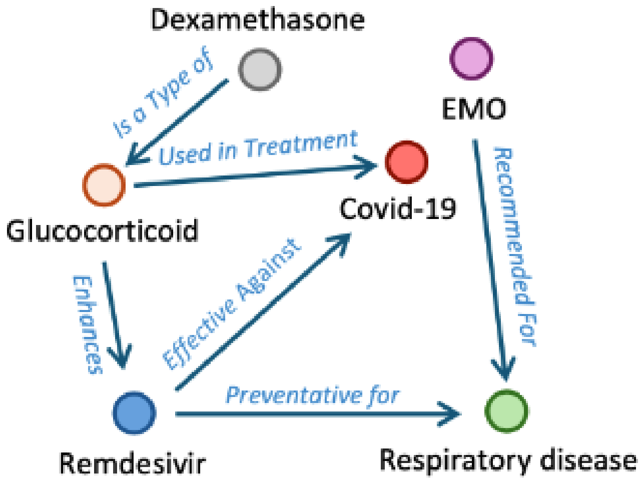
Scientific Graph

What is scientific graph?

Microscopic



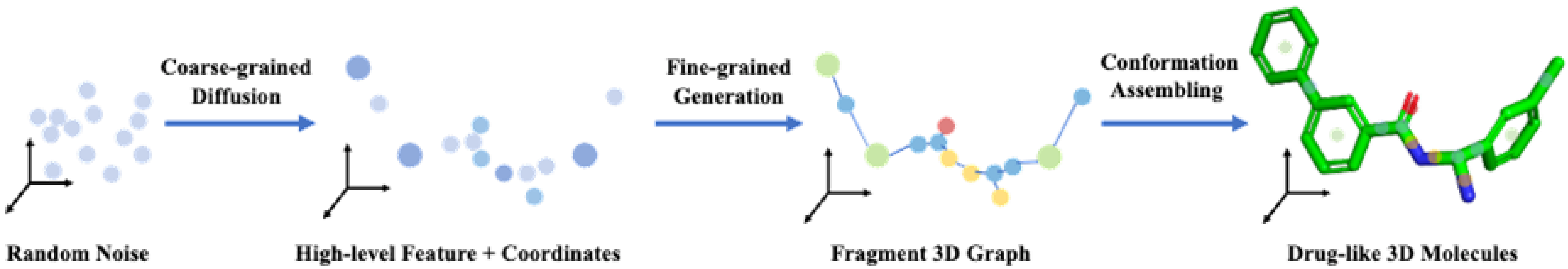
Macroscopic



What kind of tasks can we do on scientific graph?

Scientific Graph - Molecule Generation

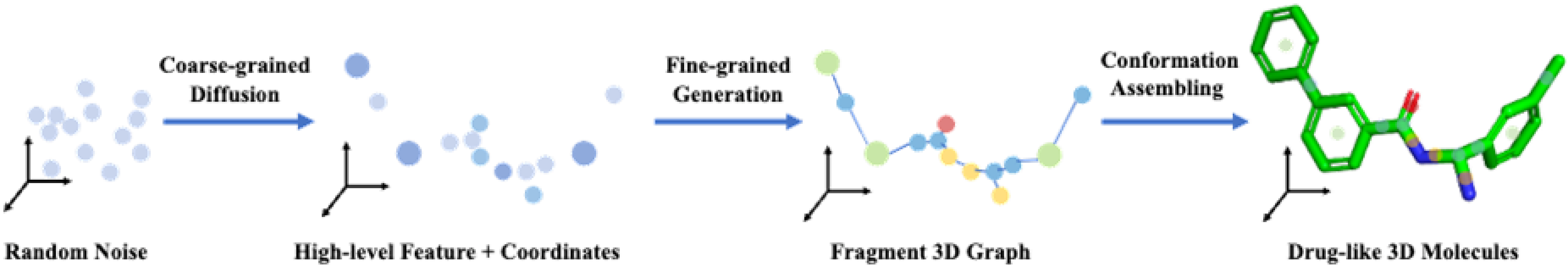
Molecule Generation



- **Random Noise:** Initialize fragment feature vectors and 3D positions as random noise.
- **Coarse-grained Diffusion:** Diffusion-denoise fragment features and coarse 3D positions to form a high-level scaffold.
- **Fine-grained Generation:** Employ an Equivariant GNN with iterative refinement to predict fragment bonds and precise atomic coordinates.
- **Conformation Assembling:** Assemble fragments into a complete 3D molecule.

Scientific Graph - Molecule Generation

Why GraphRAG for Molecule Generation?



- **Slow, resource-heavy generation:** Efficient generation guided by retrieved high-performing exemplar molecules.
- **Lack of prior chemical knowledge:** Introduce real molecules or fragments as structural priors to improve generation quality.
- **Lack of controllability:** Guide the generation direction precisely based on retrieved molecules with desired properties.

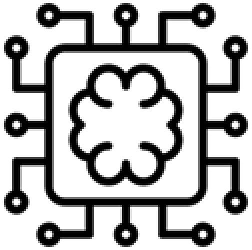
Molecule Property Prediction

Molecule property prediction is the task of using LLMs to predict a molecule's chemical properties from its structural representation.

Instruction: Lumo is the lowest unoccupied molecular orbital energy.

What's the Lumo value of this molecule?

Molecule SMILES: CCC(O)CN(C)C=O



Answer: 0.03eV

SMILES is a textual encoding of molecules topology, such as atom types, bond types, and branching

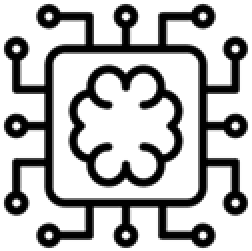
Scientific Graph - Molecule Property Prediction

Why GraphRAG for Molecule Property Prediction?

Instruction: The assay is PUBCHEM-BIOASSAY: NCI human tumor cell line growth inhibition assay.

Question: Is this molecule effective to this assay?

Input: CNC=O



Answer: Yes

Instruction: The assay is PUBCHEM-BIOASSAY: NCI human tumor cell line growth inhibition assay. Here are some examples.

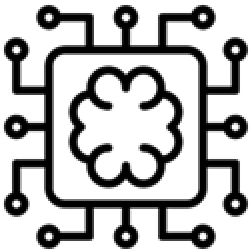
Examples:

CC(C)C(N)=O No

O=CNC=Cc1ccccc1 No

Question: Is this molecule effective to this assay?

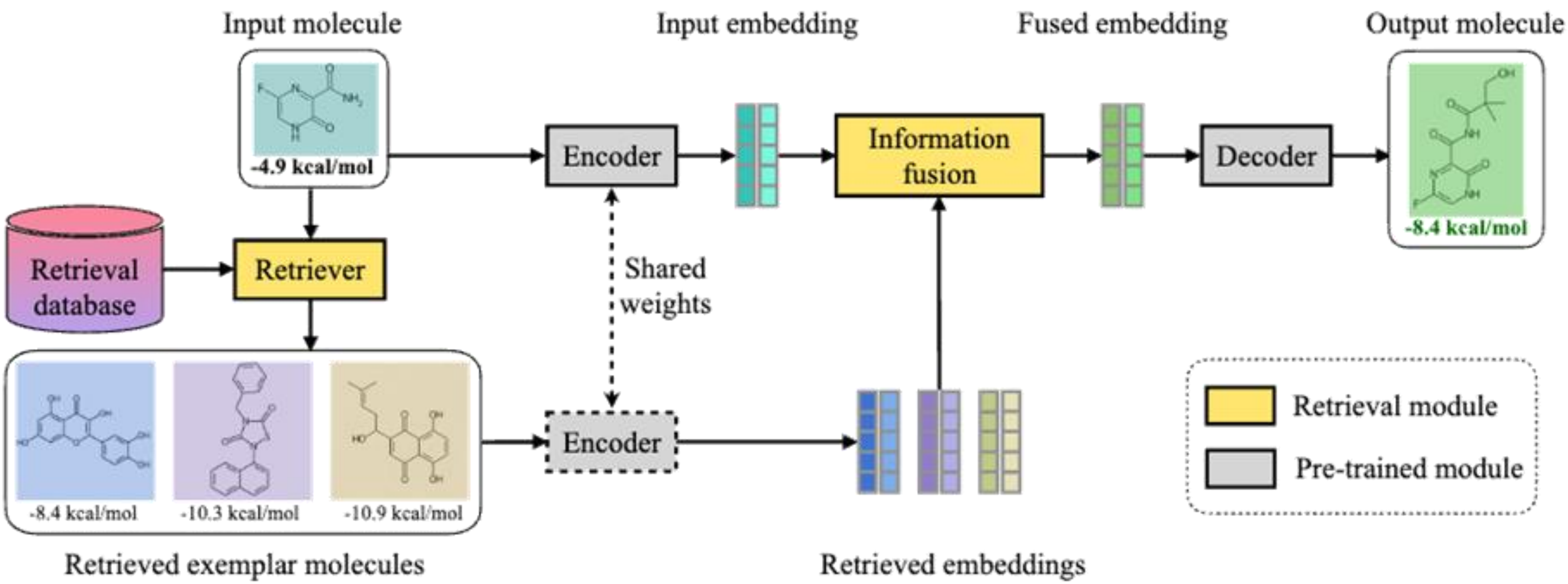
Input: CNC=O



Answer: No

By retrieving exemplar molecules **structurally similar** to CNC=O as demonstration and including them in the prompt, the LLM can make accurate predictions.

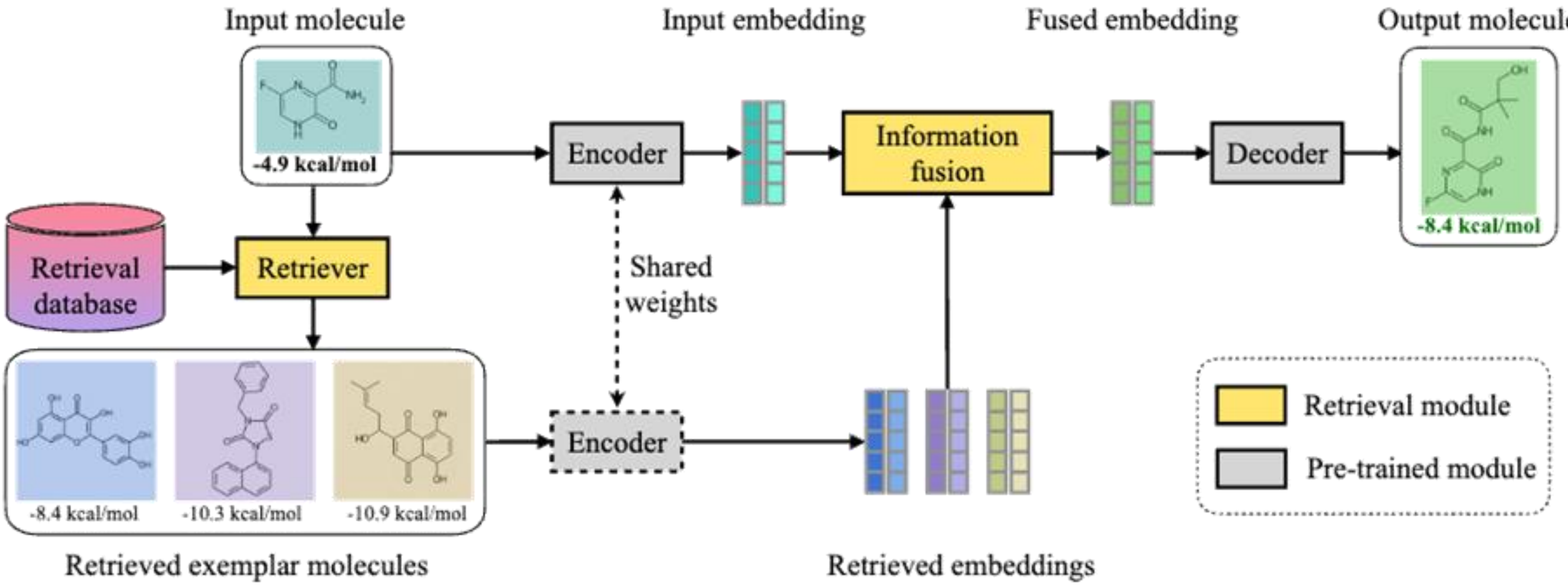
Scientific Graph - Molecule Generation



Main problem: Data is scarce and Molecular Property Control is Difficult

Core idea: Retrieve a set of exemplar molecules to guide the generation model.

Scientific Graph - Molecule Generation

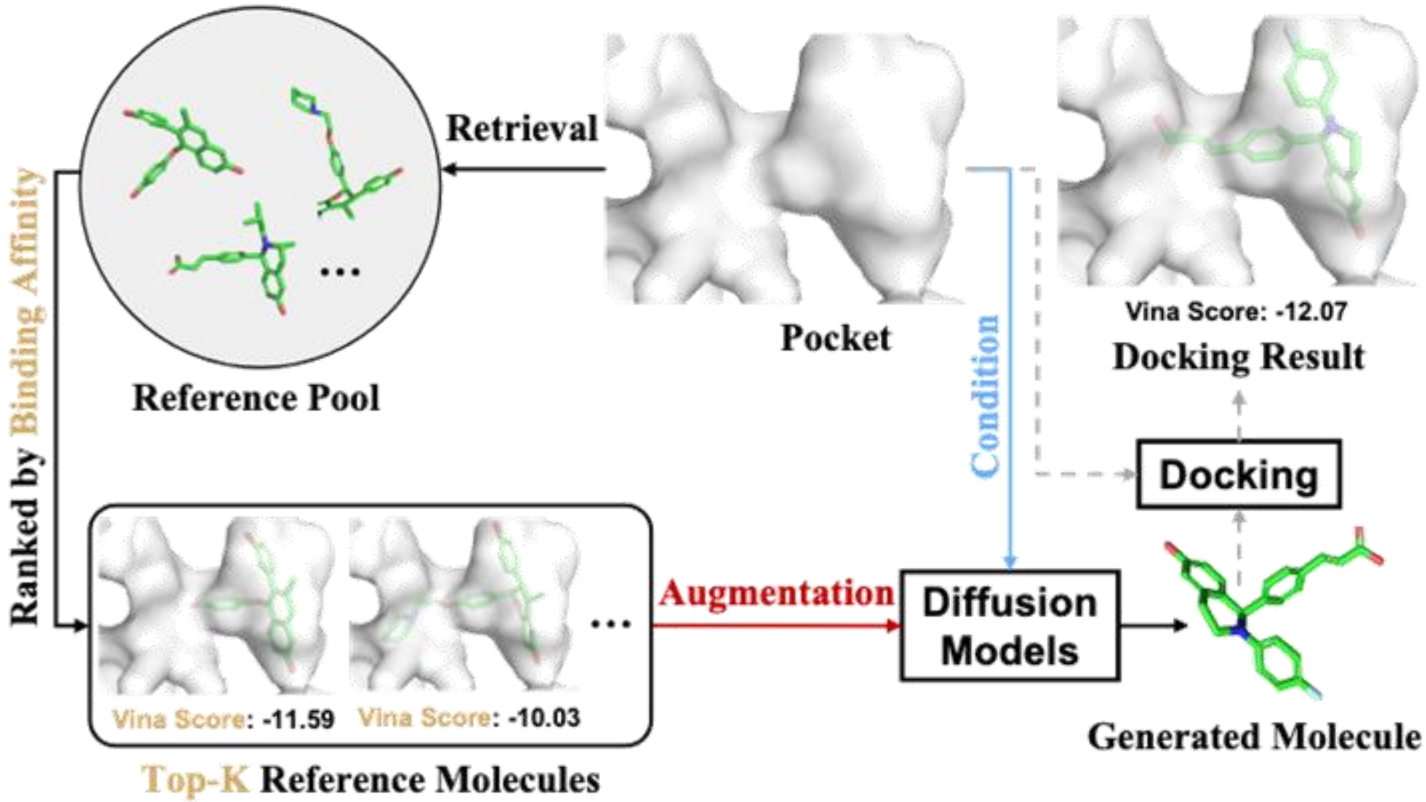


Retrieval Database: Collect exemplar molecules with desired properties.

Molecule Retrieval: Property filtering, then select top-K similar molecules using KNN.

Information Fusion: Use cross-attention to fuse input and exemplar embeddings for molecule generation via a pre-trained transformer-based model.

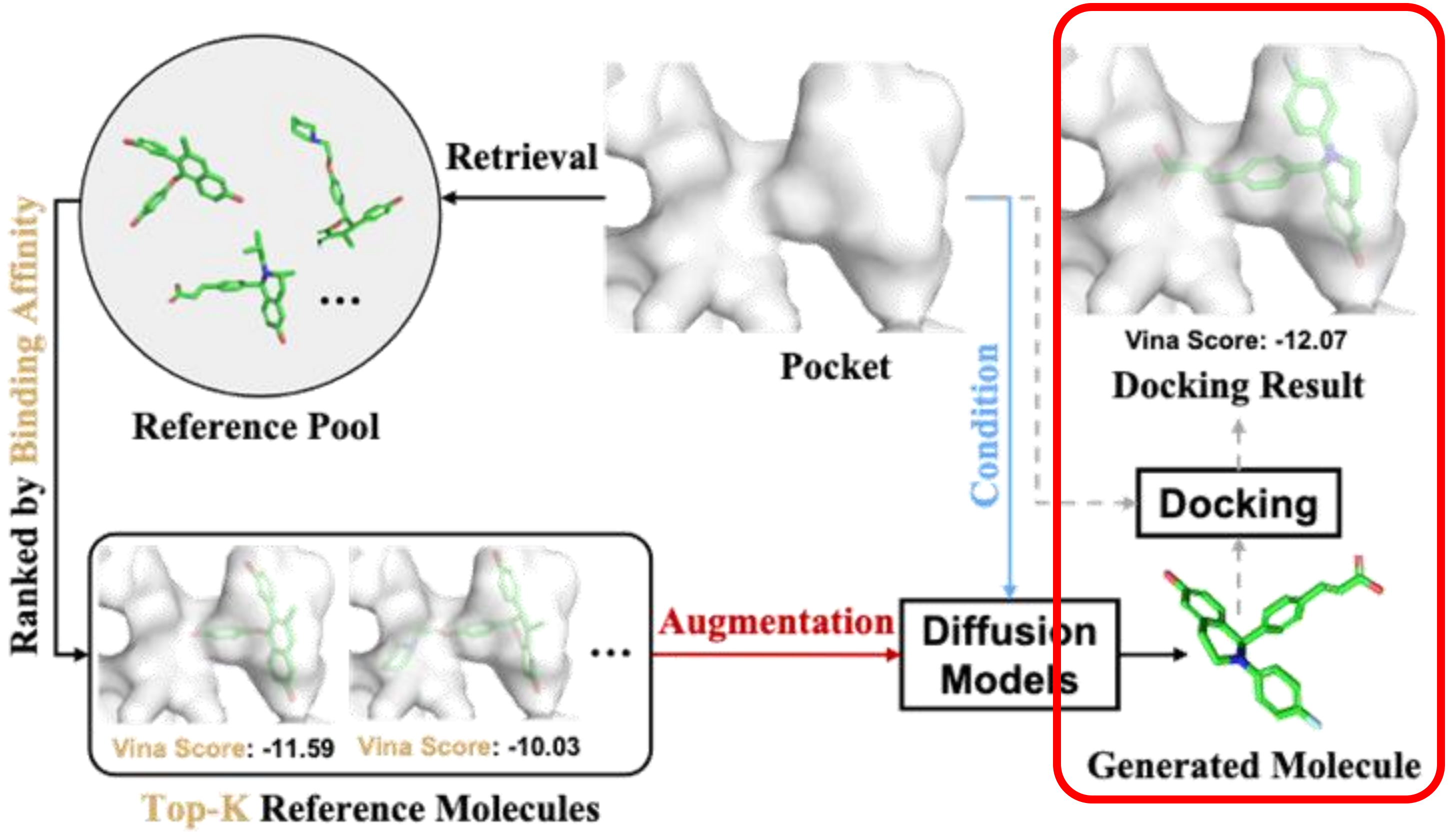
Scientific Graph - Molecule Generation



Main problem: Molecule generation without target awareness → poor binding.

Core idea: Retrieve binding-aware references → guide diffusion to generate target-specific, high-affinity molecules.

Scientific Graph - Molecule Generation

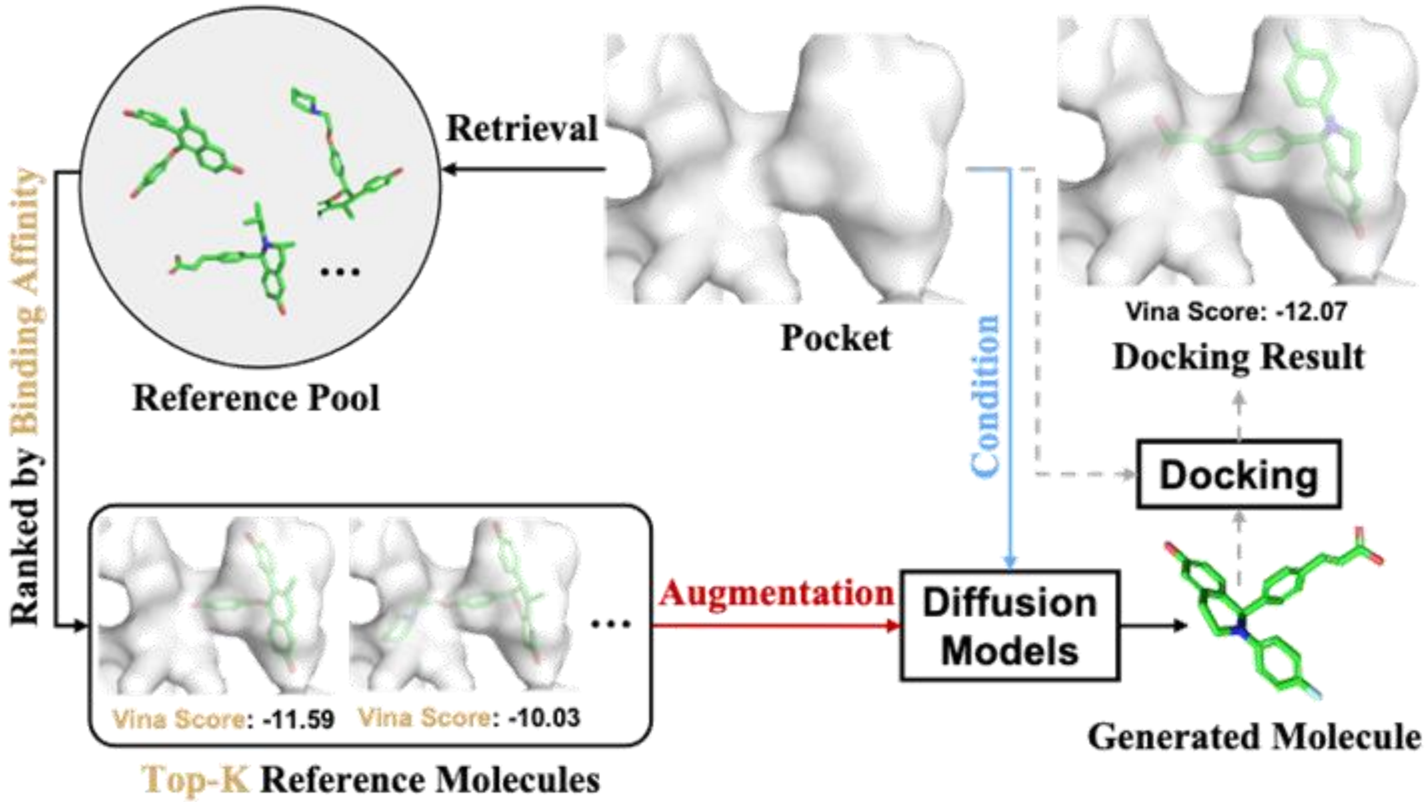


What is docking?

Predict and select small **molecules** that can **effectively bind to disease-related protein targets.**

Starting points for further optimization toward the development of drug candidates.

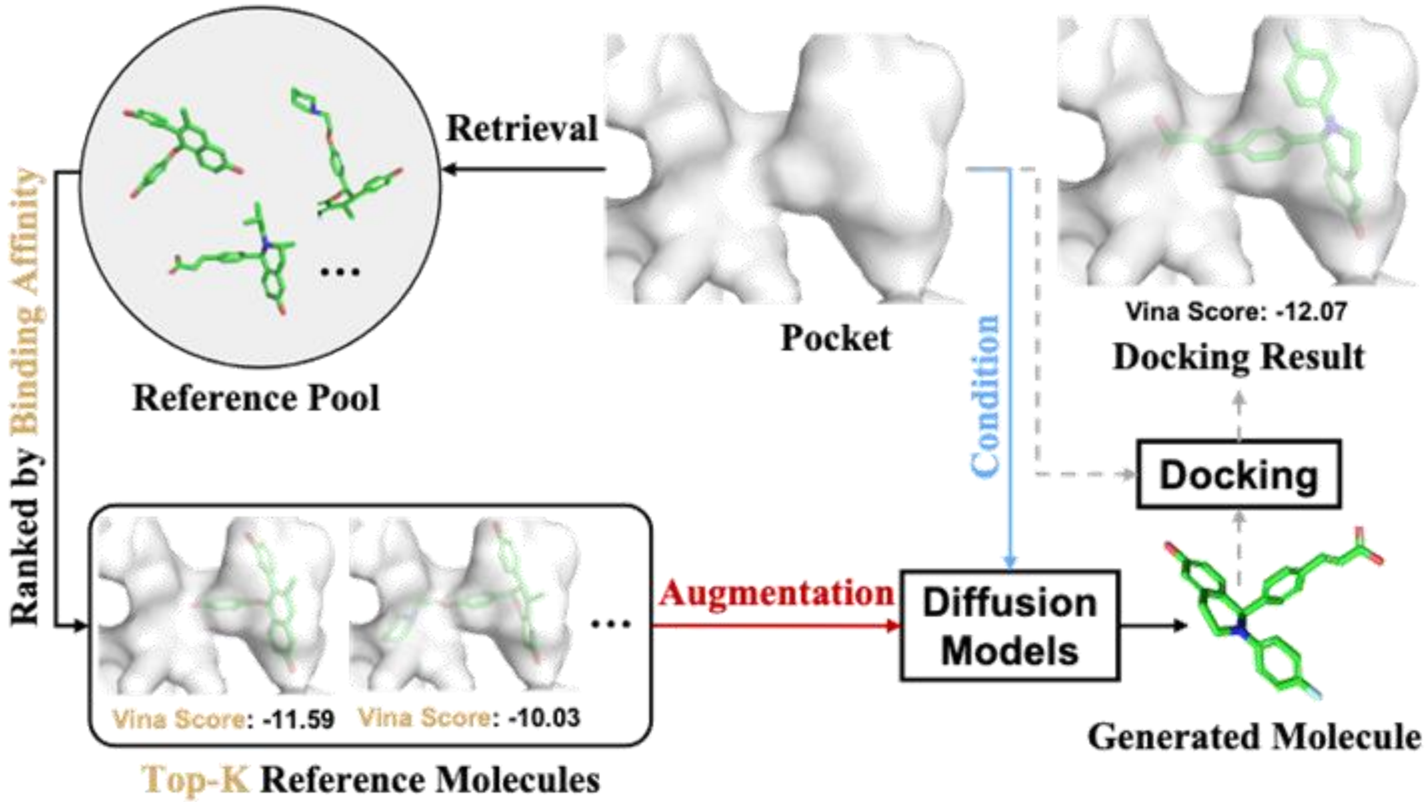
Scientific Graph - Molecule Generation



Why use graph-based retrieval?

- Ignore target protein structure → Poor binding when evaluated by docking.
- Retrieve strong-binding reference molecules → Guide diffusion model → Generate protein-specific, high-affinity molecules.

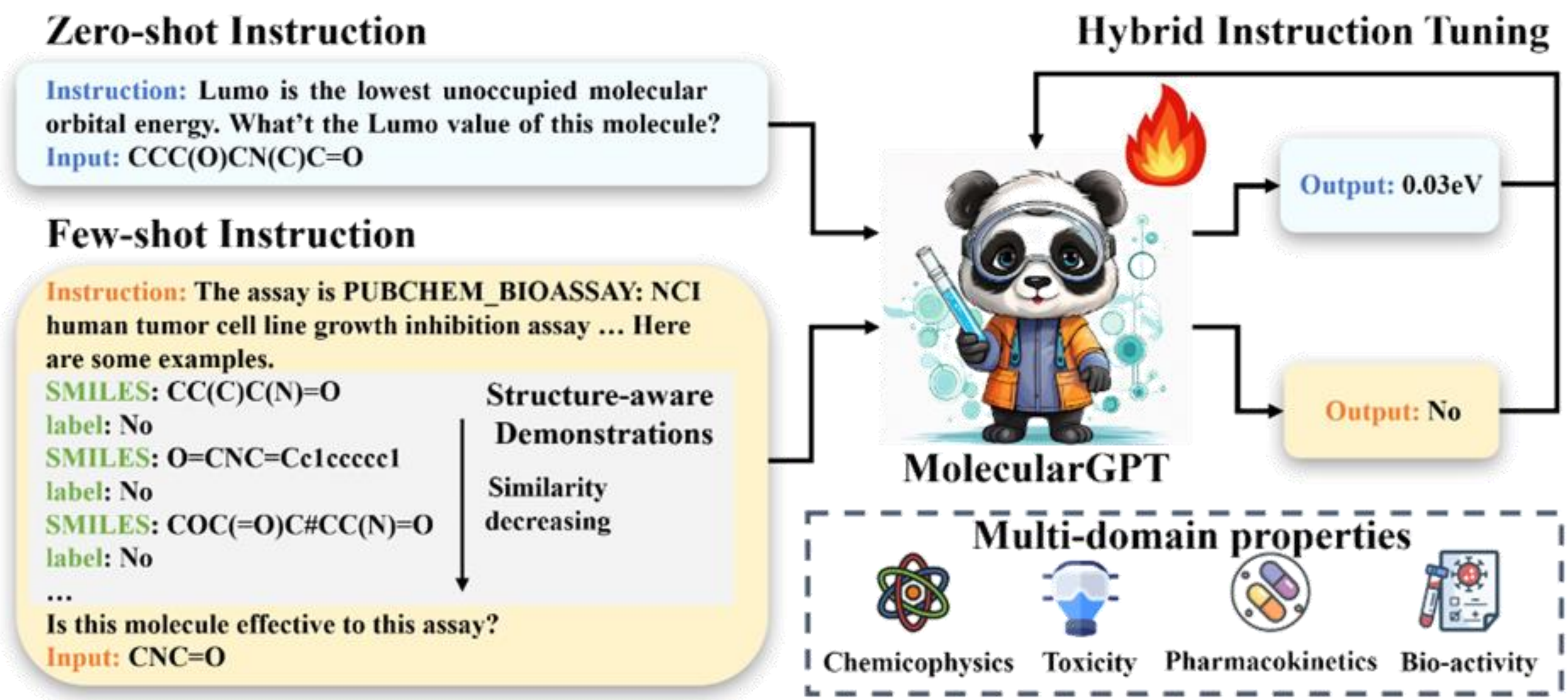
Scientific Graph - Molecule Generation



How to retrieve reference molecules?

- Target Pocket Encoding
- Precompute Reference Pool
- Similarity Search (L2 Distance)
- Retrieve Top-K Molecules
- Use for Generation

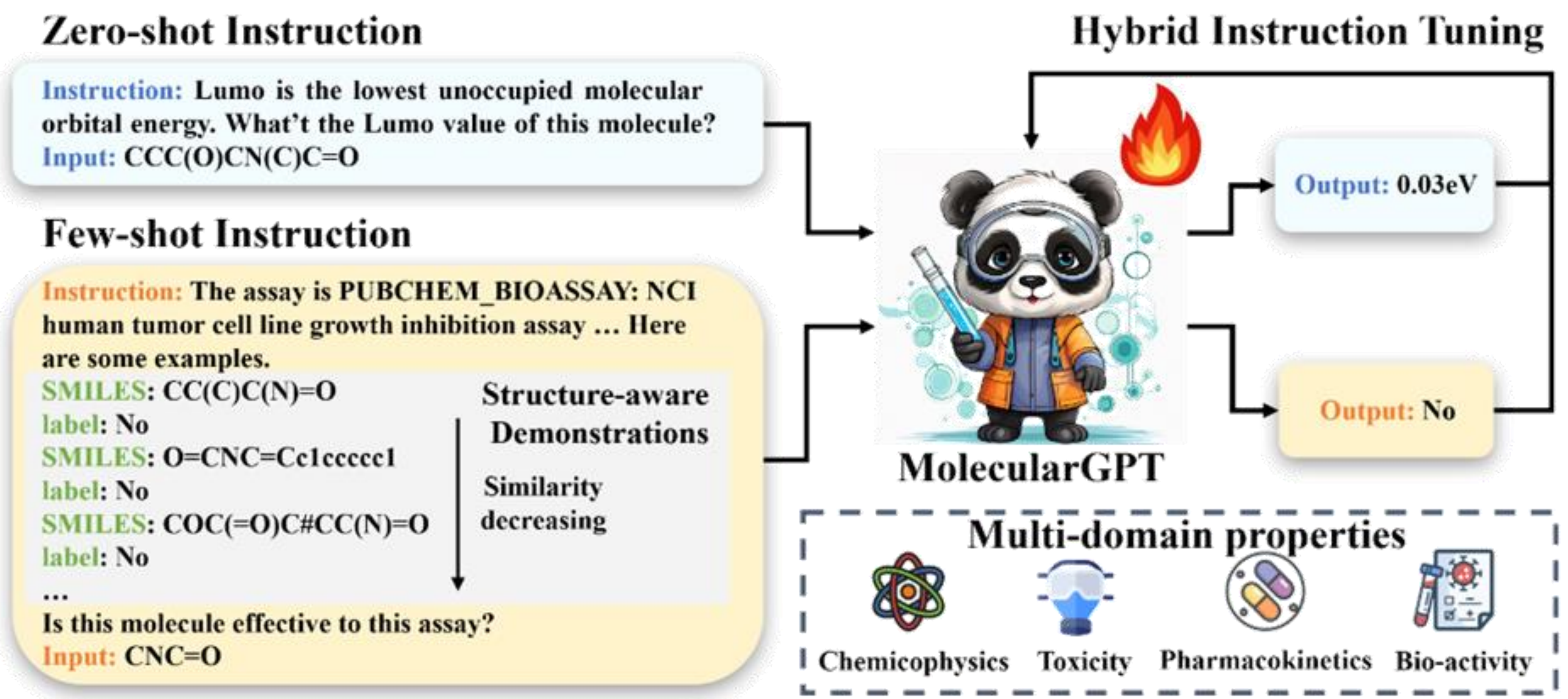
Scientific Graph - Molecule Property Prediction



Main problem: LLMs lack domain-specific Knowledge

Core idea: MolecularGPT retrieve relevant molecules based on structure to enhance LLM.

Scientific Graph - Molecule Property Prediction



Data Preparation: Collect (molecule, property) pairs

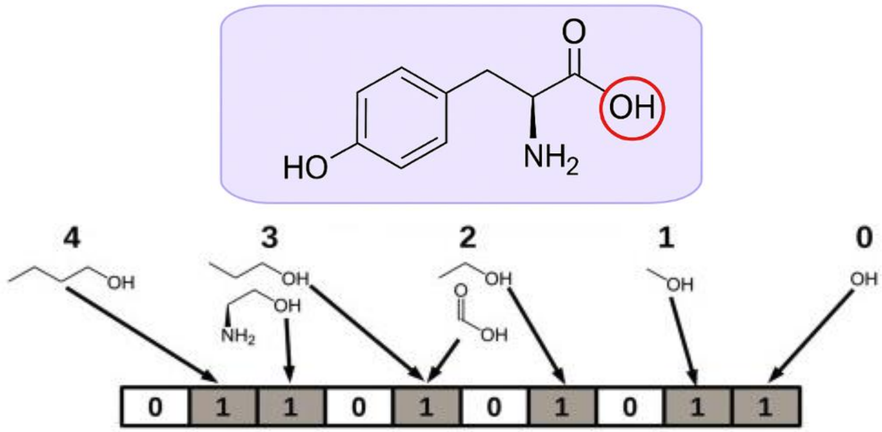
SMILES Conversion: Represent molecules as SMILES strings for input.

Neighbor Retrieval: Tanimoto similarity

Scientific Graph - Molecule Property Prediction

What is Tanimoto similarity?

A similarity metric between two binary fingerprints A and B



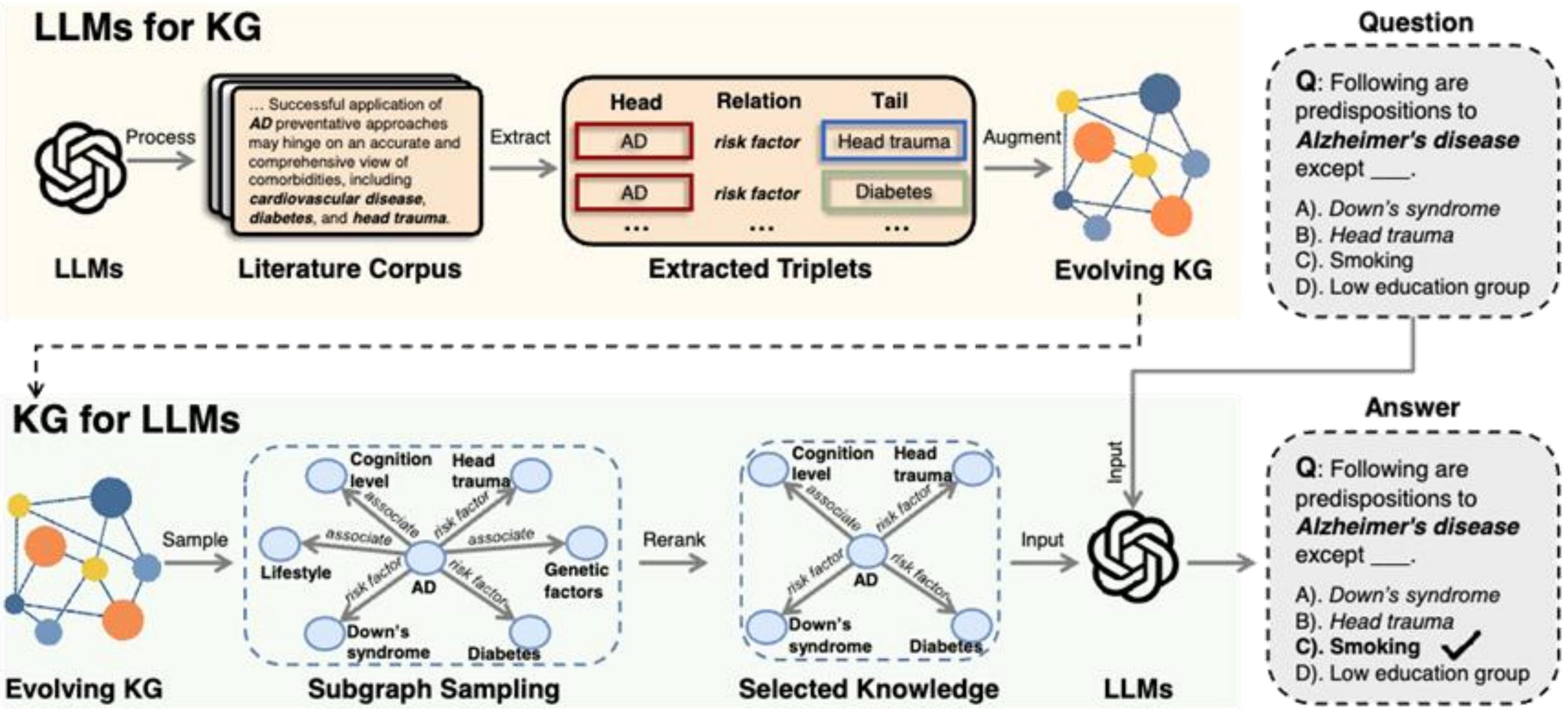
$$\text{Tanimoto}(A, B) = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Data Preparation: Collect (molecule, property) pairs

SMILES Conversion: Represent molecules as SMILES strings for input.

Neighbor Retrieval: **Tanimoto similarity**

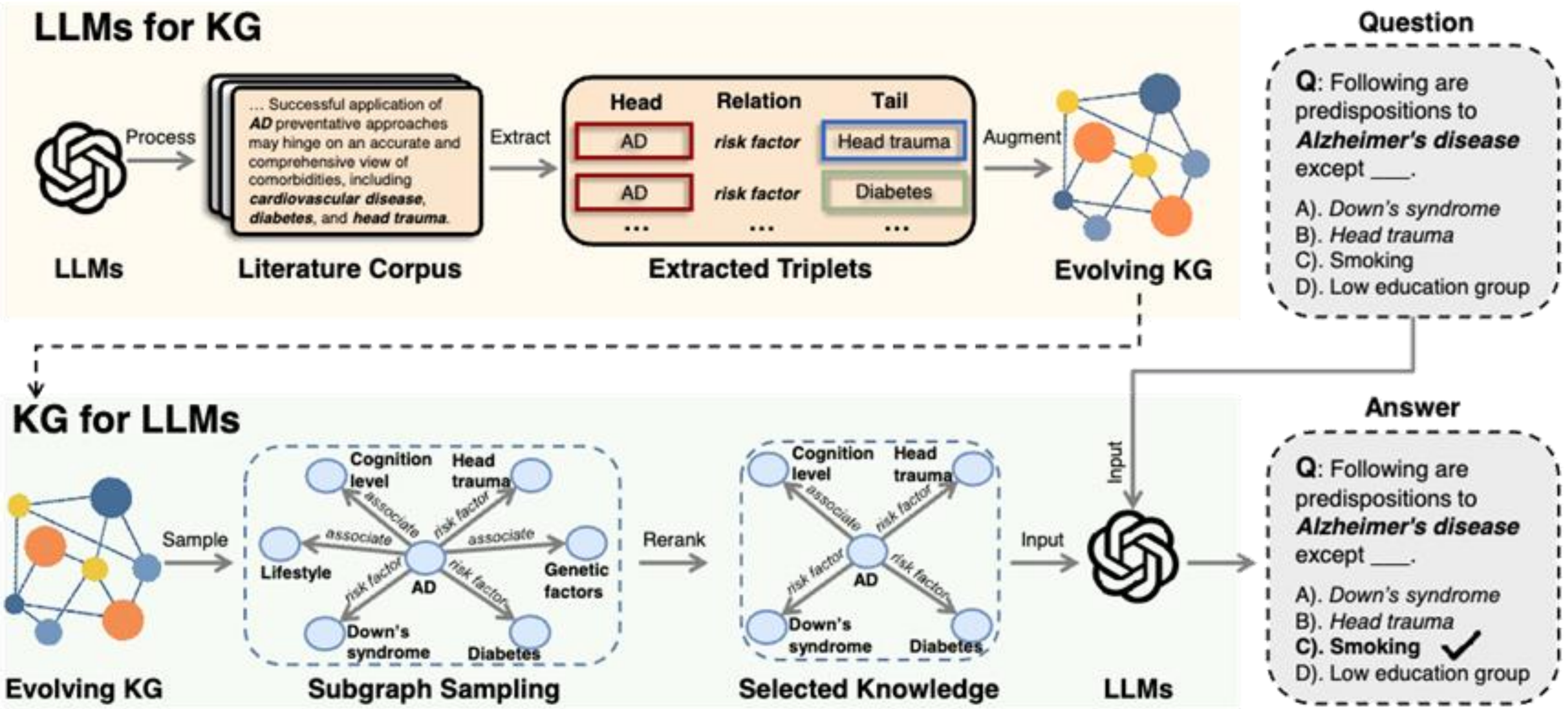
Scientific Graph - Question Answering



Main problem: LLMs struggle to answer Alzheimer's Disease (AD) questions due to limited integration of specialized biomedical knowledge.

Core idea: DALK augments LLMs with a scientific literature-derived knowledge graph to improve reasoning on AD-related questions.

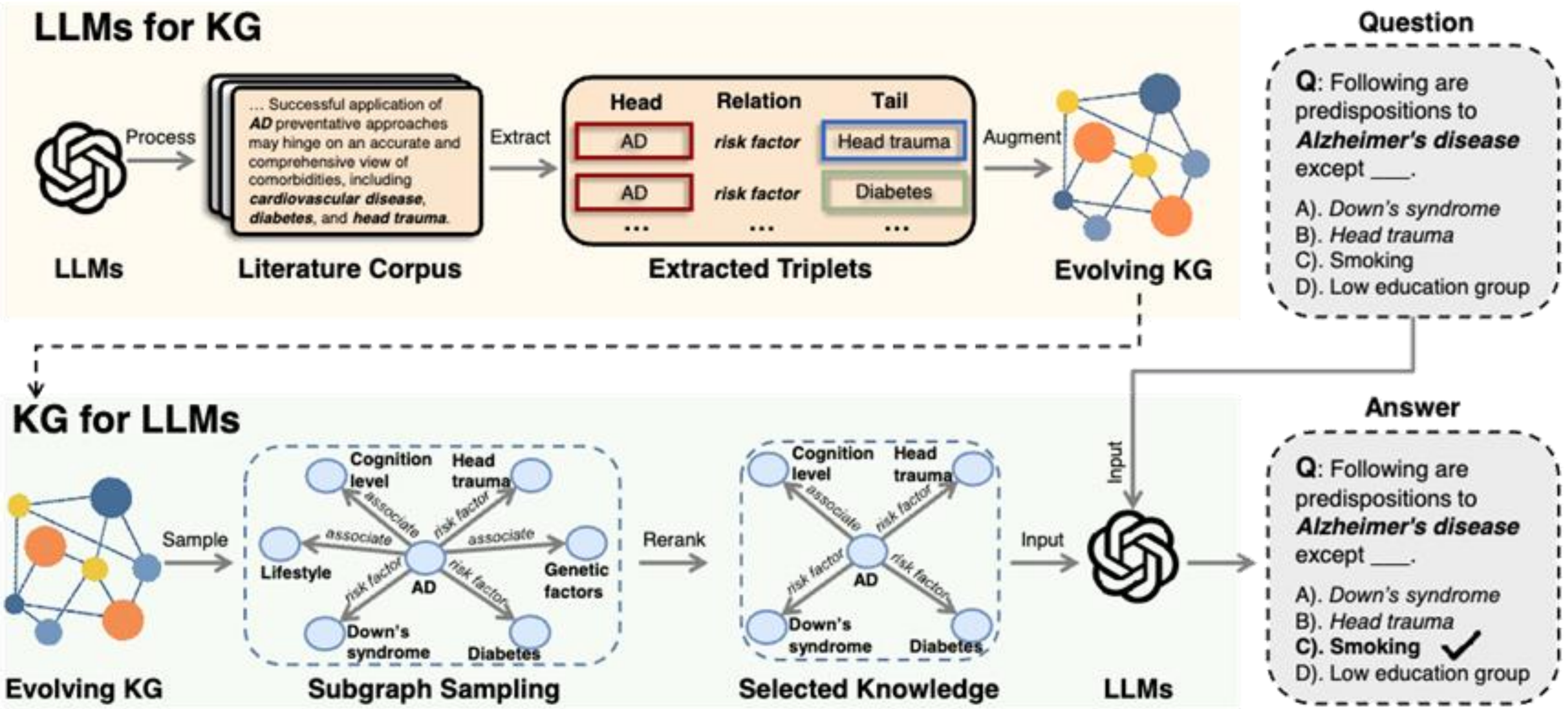
Scientific Graph - Question Answering



Entity Recognition: Use PubTator Central to identify biomedical entities.
Relation Extraction:
- Pairwise: LLMs describe pairwise relations.
- Generative: LLMs generate all triplets.

Evolving KG: Update KG to reflect new discoveries annually

Scientific Graph - Question Answering



Entity Extraction and Linking for query

Path Exploration: K-hop path triplets of seeding nodes and their induced subgraph

Neighbor Exploration: Neighbor of seeding nodes and their induced subgraph

Scientific Graph - Future Direction

Multi-modal GraphRAG for Scientific Graph

Motivation: Scientific data is inherently multi-modal:

- **Text** (Papers, Document)
- **Image** (Medical Images: MRI and CT)
- **Table**

Current GraphRAG mainly focus on text and structure separately.

Scientific Graph - Future Direction

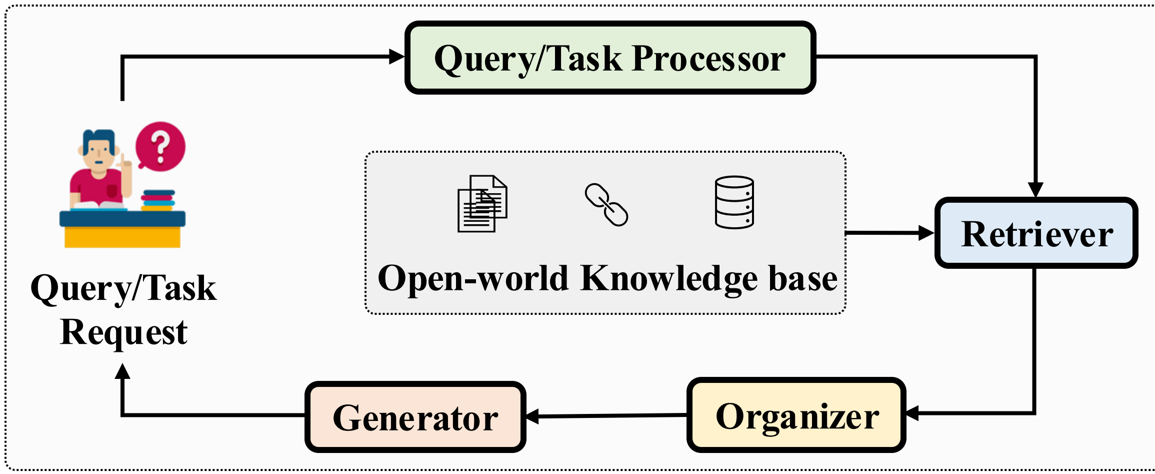
Towards Trustworthy GraphRAG for Scientific Graphs

Motivation: GraphRAG has been really deployed in many high-stake scenarios

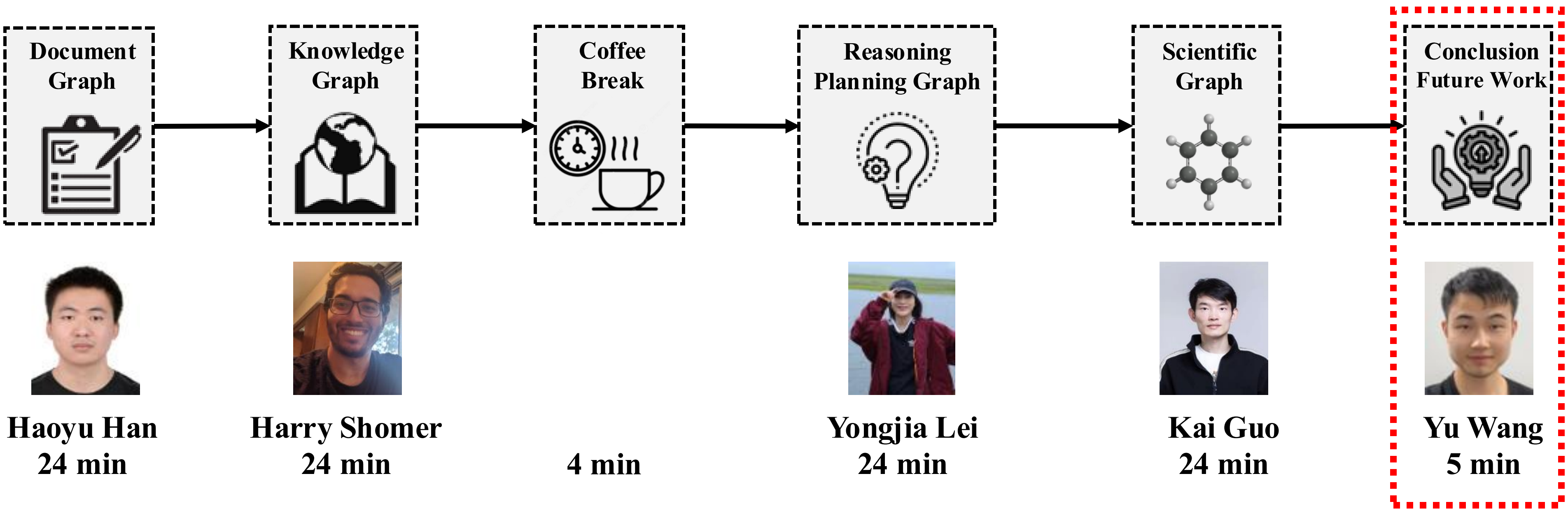
- **Retrieval focuses on associative facts, not verified causal relations**
- **Generated answers lack scientific rigor and are less trustworthy**

Building Causal Evidence Rule-based Retrieval-augmented Generation

Outline

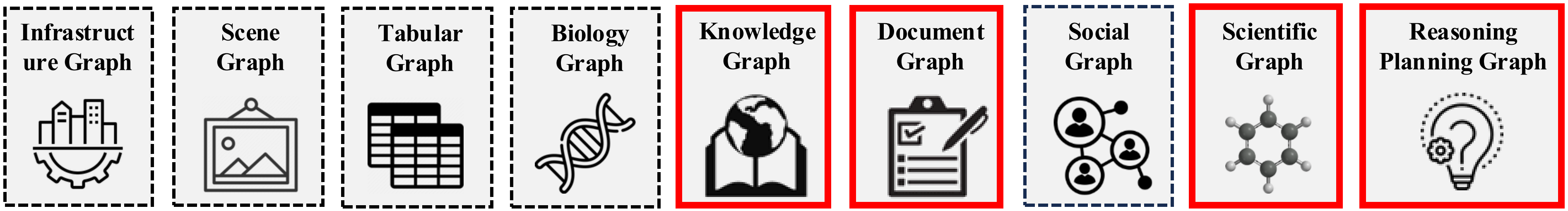


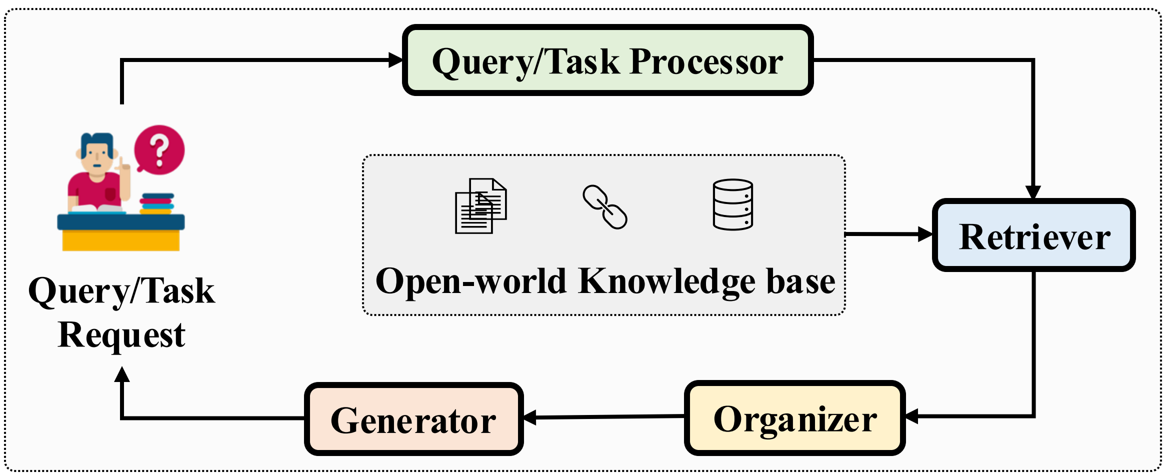
Query/Task Processor
Query/Task Request
Open-world Knowledge base
Retriever
Generator
Organizer



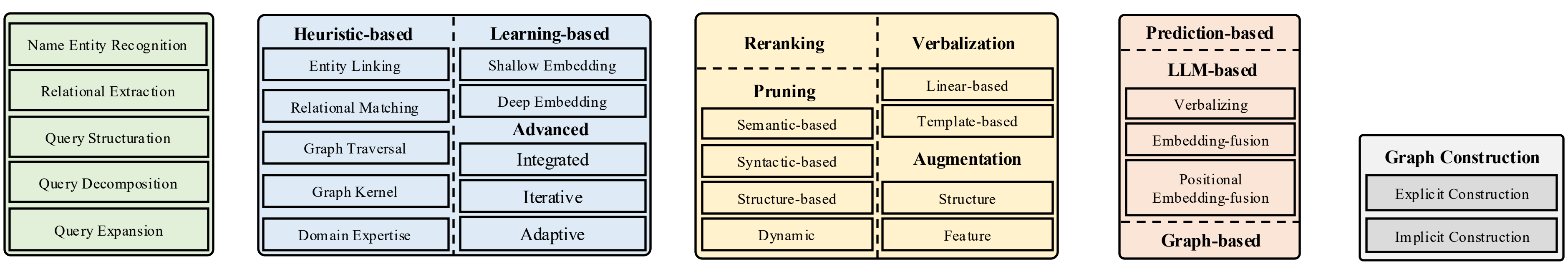
Document Graph
Knowledge Graph
Coffee Break
Reasoning Planning Graph
Scientific Graph
Conclusion Future Work
Haoyu Han 24 min
Harry Shomer 24 min
4 min
Yongjia Lei 24 min
Kai Guo 24 min
Yu Wang 5 min

Conclusion

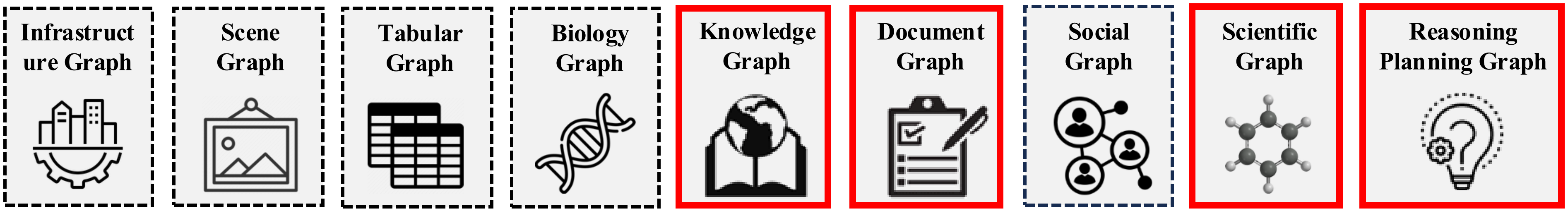


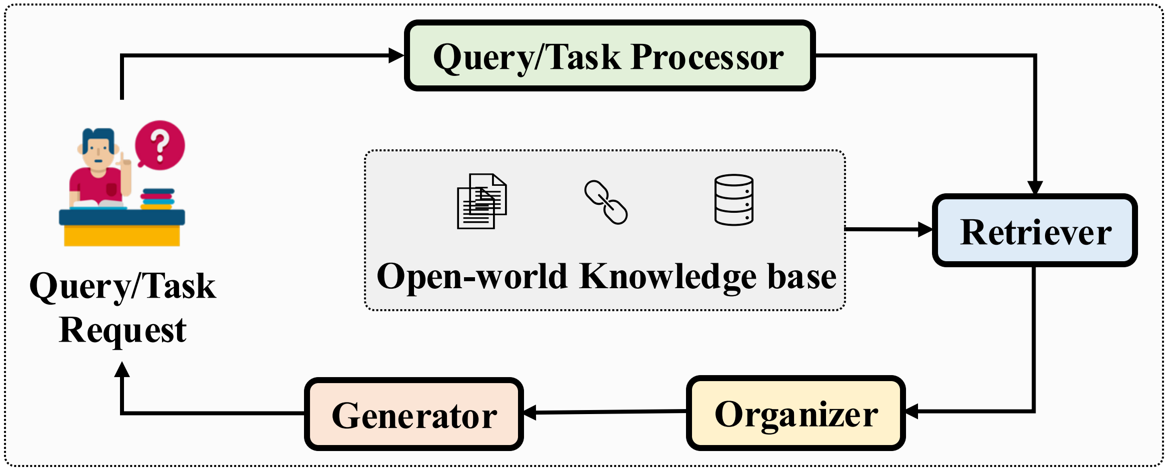


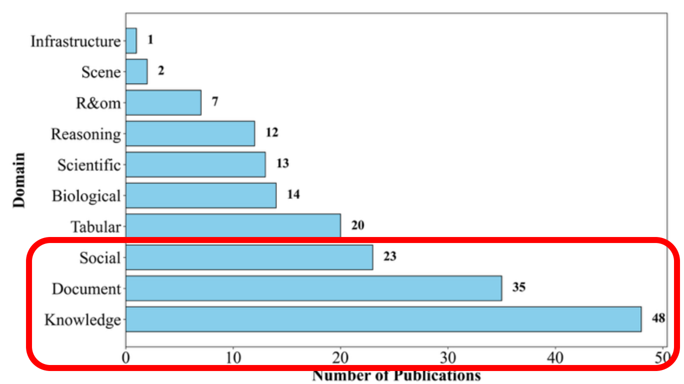
https://github.com/Graph-RAG/GraphRAG/



Future Work 1 – GraphRAG on other domains







Statistics surveyed until 12/31/2024

Name Entity Recognition
Relational Extraction
Query Structuration
Query Decomposition
Query Expansion

Heuristic-based
Entity Linking
Relational Matching
Graph Traversal
Graph Kernel
Domain Expertise

Learning-based
Shallow Embedding
Deep Embedding

Advanced
Integrated
Iterative
Adaptive

Reranking

Pruning
Semantic-based
Syntactic-based
Structure-based
Dynamic

Verbalization
Linear-based
Template-based

Augmentation
Structure
Feature

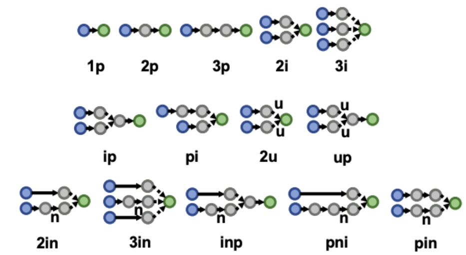
Prediction-based

LLM-based
Verbalizing
Embedding-fusion
Positional Embedding-fusion

Graph-based

Graph Construction
Explicit Construction
Implicit Construction

Future Work 2 – GraphRAG Module Design



Ren et al

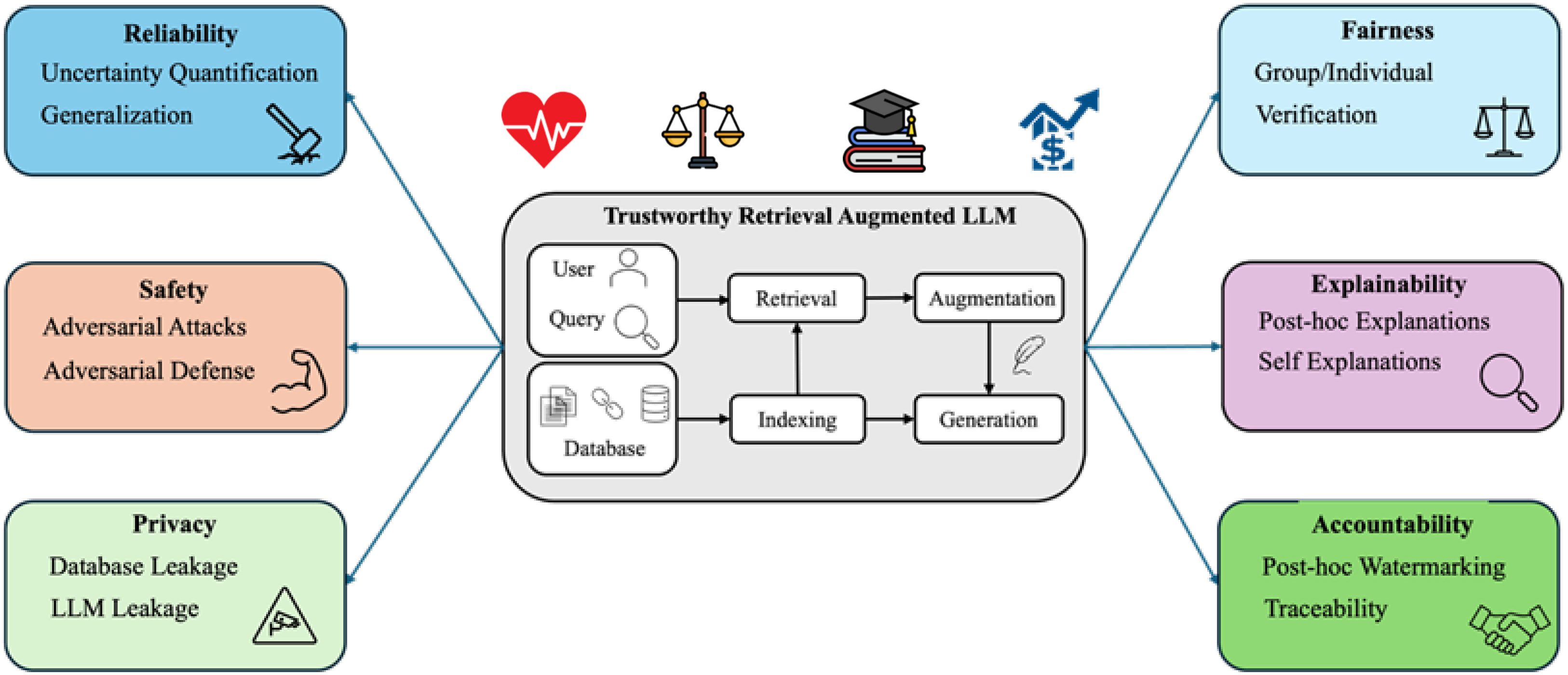
- **Query Preprocessor –** Analyze Query Structure and Topology

- **Retriever**
    - Harmonizing Internal and External Knowledge
    - How to embedding different types of structured knowledge (e.g., cluster vs path)
    - Reasoning, planning, and thinking along the way (e.g., Search-R1)

- **Organizer**
    - Retrieved Graph can be large, balancing completeness and conciseness (e.g., exponentially growth receptive field)
    - Optimal Data Structuring that generator can leverage
    - Align retrieved resources from different parties (e.g., multi-modality graph)

- **Generator**
    - Correct Format of Prompting (e.g., adjacent list, markdown format, ……)
    - Structural Encoding for expressing the graph structure

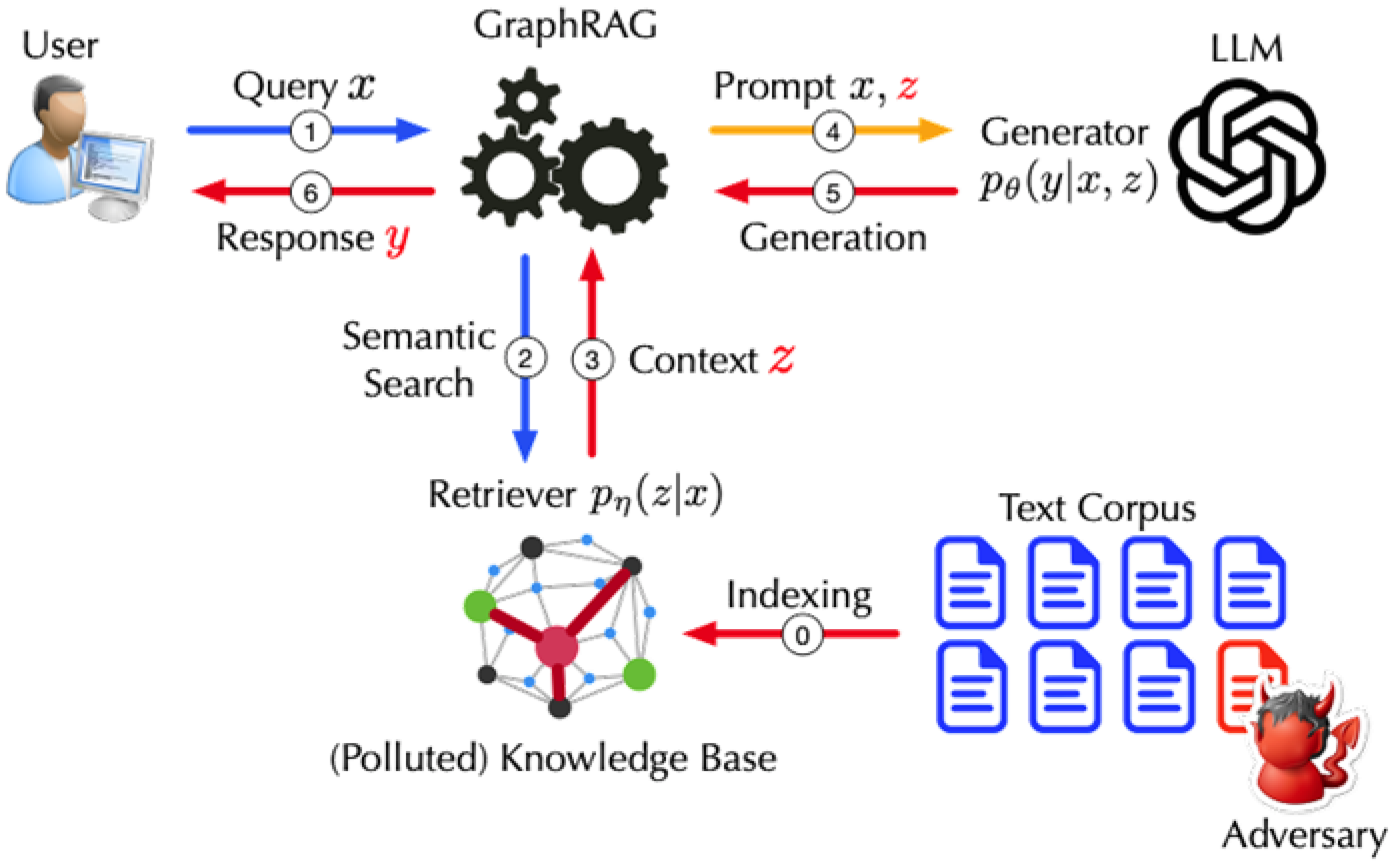
Future Work 3 – Trustworthy GraphRAG



How about the unique trustworthy challenges caused graph structure?

https://github.com/Arstanley/Awesome-Trustworthy-RAG

Future Work 3 – Trustworthy GraphRAG



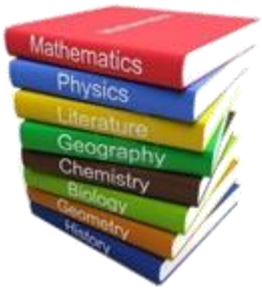
How about the unique trustworthy challenges caused graph structure?

https://github.com/Arstanley/Awesome-Trustworthy-RAG

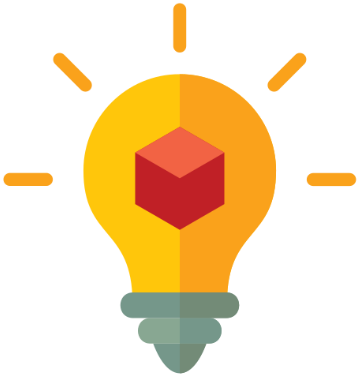
Future Work 4 – Data-centric GraphRAG

- **Balance Internal and External Knowledge**

V.S.



- **Trade-off Among Accuracy, Diversity, and Novelty**



Thank you for your listening!

Retrieval-Augmented Generation with Graphs (GraphRAG)

Haoyu Han[1,*], Yu Wang[2,*], Harry Shomer[1], Kai Guo[1], Jiayuan Ding[5], Yongjia Lei[2], Mahantesh Halappanavar[3], Ryan A. Rossi[4], Subhabrata Mukherjee[5], Xianfeng Tang[6], Qi He[6], Zhigang Hua[7], Bo Long[7], Tong Zhao[8], Neil Shah[8], Amin Javari[9], Yinglong Xia[7], Jiliang Tang[1]
[1]Michigan State University, [2]University of Oregon, [3]Pacific Northwest National Laboratory
[4]Adobe Research, [5]Hippocratic AI, [6]Amazon, [7]Meta ,[8]Snap Inc., ,[9]The Home Depot,
{hanhaoy1, shomerha, guokai1, tangjili}@msu.edu,
{yuwang, yongjia}@uoregon.edu, hala@pnnl.gov, ryarossi@gmail.com,
{jiayuan, subho}@hippocraticai.com, {xianft, qih}@amazon.com,
{zhua, bolong, yxia}@meta.com, {tong, nshah}@snap.com, amin_javari@homedepot.com

Abstract

Retrieval-augmented generation (RAG) is a powerful technique that enhances downstream task execution by retrieving additional information, such as knowledge, skills, and tools from external sources. Graph, by its intrinsic "nodes connected by edges" nature, encodes massive heterogeneous and relational information, making it a golden resource for RAG in tremendous real-world applications. As a result, we have recently witnessed increasing attention on equipping RAG with Graph, i.e., GraphRAG. However, unlike conventional RAG, where the retriever, generator, and external data sources can be uniformly designed in the neural-embedding space, the uniqueness of graph-structured data, such as diverse-formatted and domain-specific relational knowledge, poses unique and significant challenges when designing GraphRAG for different domains. Given the broad applicability, the associated design challenges, and the recent surge in GraphRAG, a systematic and up-to-date survey of its key concepts and techniques is urgently desired. Following this motivation, we present a comprehensive and up-to-date survey on GraphRAG. Our survey first proposes a holistic GraphRAG framework by defining its key components, including query processor, retriever, organizer, generator, and data source. Furthermore, recognizing that graphs in different domains exhibit distinct relational patterns and require dedicated designs, we review GraphRAG techniques uniquely tailored to each domain. Finally, we discuss research challenges and brainstorm directions to inspire cross-disciplinary opportunities. Our survey repository is publicly maintained at https://github.com/Graph-RAG/GraphRAG/.

GraphRAG

SDM25-GraphRAG

Towards Trustworthy Retrieval Augmented Generation for Large Language Models: A Survey

Bo Ni[1], Zheyuan Liu[†2], Leyao Wang[†1] Yongjia Lei[†3], Yuying Zhao[1], Xueqi Cheng[1], Qingkai Zeng[2], Luna Dong[4], Yinglong Xia[4], Krishnaram Kenthapadi[5], Ryan Rossi[6], Franck Dernoncourt[6], Md Mehrab Tanjim[6], Nesreen Ahmed[7], Xiaorui Liu[8], Wenqi Fan[9], Erik Blasch[10], Yu Wang[*3], Meng Jiang[*2], Tyler Derr[*1]

[1]Vanderbilt University, [2]University of Notre Dame, [3]University of Oregon, [4]Meta, [5]Oracle Health AI, [6]Adobe Research, [7]Cisco AI Research, [8]North Carolina State University, [9]The Hong Kong Polytechnic University, [10]Air Force Research Lab

{bo.ni, leyao.wang, yuying.zhao, xueqi.cheng, tyler.derr}@vanderbilt.edu,
{zliu29, qzeng, mjiang2}@nd.edu, {yongjia, yuwang}@uoregon.edu,
{lunadong, yxia}@meta.com, krishnaram.kenthapadi@oracle.com,
{ryrossi,dernonco,tanjim}@adobe.com, nesahmed@cisco.com,
xliu96@ncsu.edu, wenqi.fan@polyu.edu.hk, erik.blasch.1@us.af.mil

Abstract

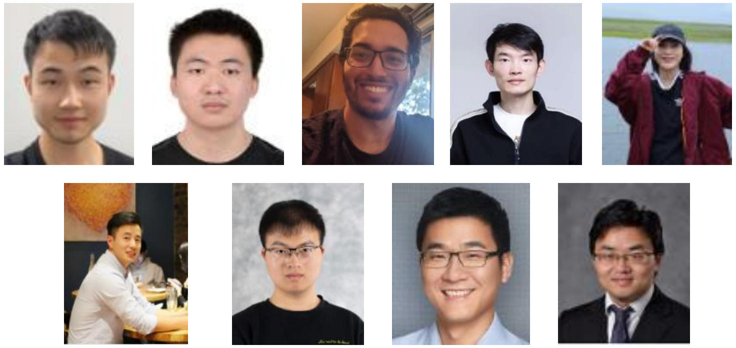
Retrieval-Augmented Generation (RAG) is an advanced technique designed to address the challenges of Artificial Intelligence-Generated Content (AIGC). By integrating context retrieval into content generation, RAG provides reliable and up-to-date external knowledge, reduces hallucinations, and ensures relevant context across a wide range of tasks. However, despite RAG's success and potential, recent studies have shown that the RAG paradigm also introduces new risks, including robustness issues, privacy concerns, adversarial attacks, and accountability issues. Addressing these risks is critical for future applications of RAG systems, as they directly impact their trustworthiness. Although various methods have been developed to improve the trustworthiness of RAG methods, there is a lack of a unified perspective and framework for research in this topic. Thus, in this paper, we aim to address this gap by providing a comprehensive roadmap for developing trustworthy RAG systems. We place our discussion around five key perspectives: reliability, privacy, safety, fairness, explainability, and accountability. For each perspective, we present a general framework and taxonomy, offering a structured approach to understanding the current challenges, evaluating existing solutions, and identifying promising future research directions. To encourage broader adoption and innovation, we also highlight the downstream applications where trustworthy RAG systems have a significant impact. For more information about the survey, please check our GitHub repository*.

Trustworthy RAG

Society for Industrial and Applied Mathematics

We really appreciate the travel support from SIAM for some of our teammates in presenting this tutorial!

Lead Tutors



Survey Collaborators
(Order by Random)