Efficient Knowledge Graph Evidence Utilization for Retrieval-Augmented Question Answering

Jinho Choi(최진호)

Supervisor: Byungkook Oh

Graph & Language Intelligence Laboratory
Department of Computer Science and Engineering
Konkuk University

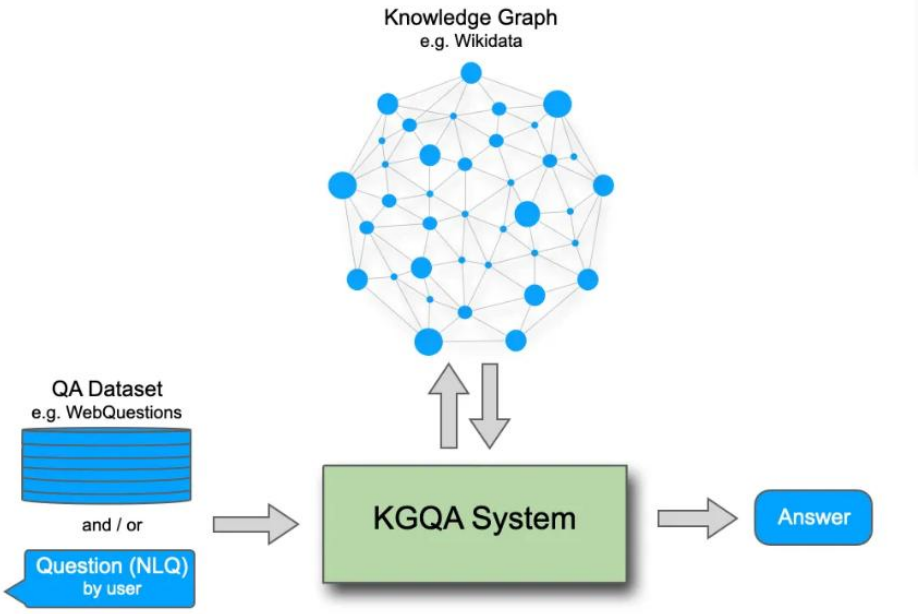
2026.01.20

CONTENTS

Background

What is the 'KGQA'?

- KGQA(Knowledge Graph Question Answering)는 자연어 질문에 대한 답변을 지식그래프의 정보를 바탕으로 생성하는 Task



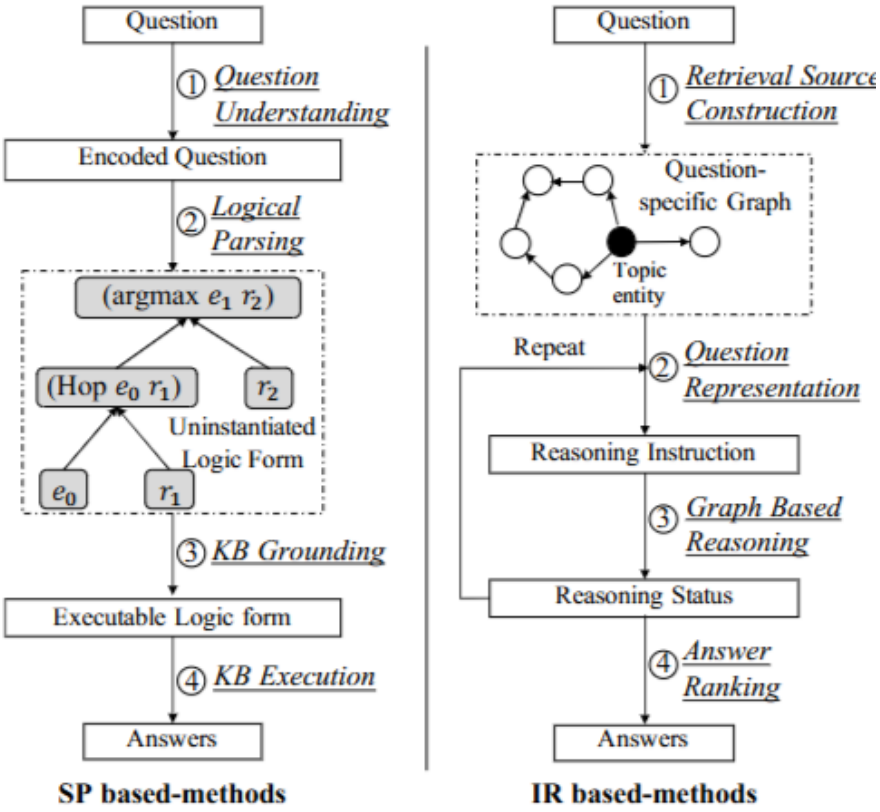
A) Where is the place of birth of Hawking ? ·····▶ Oxford

B) What did Stephen Hawking study at Oxford University ? ·····▶ physics

Background

Traditional KGQA(Knowledge Graph Question Answering)

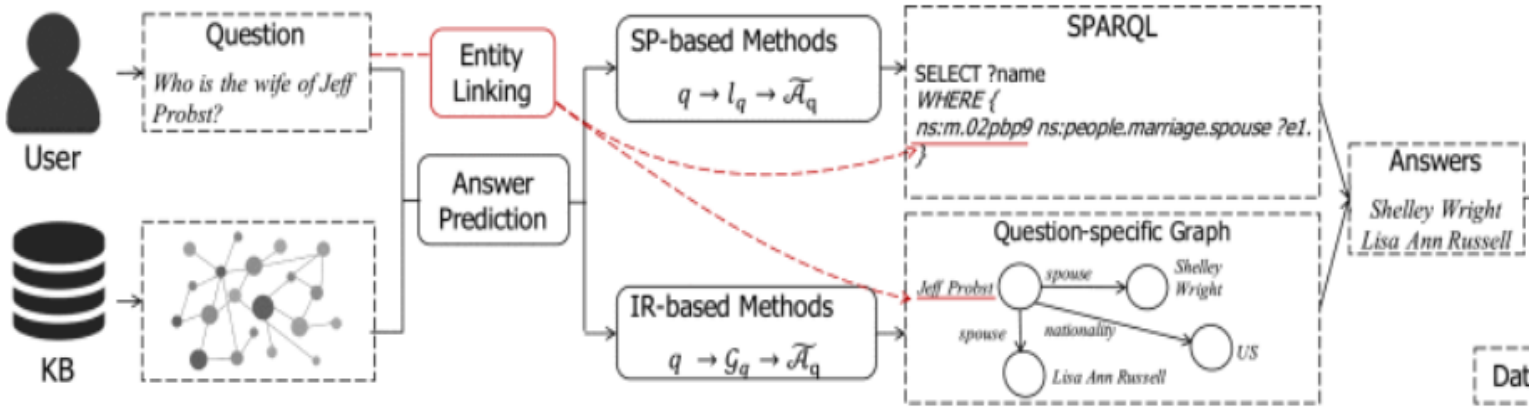
- **SP-based methods** (Semantic Parsing): 질문을 Logical form/Query로 변환해 KG 실행
- **IR-based methods** (Information Retrieval): 후보 Path/Subgraph/Triples을 뽑고 랭킹으로 답 선택



Background

Traditional KGQA

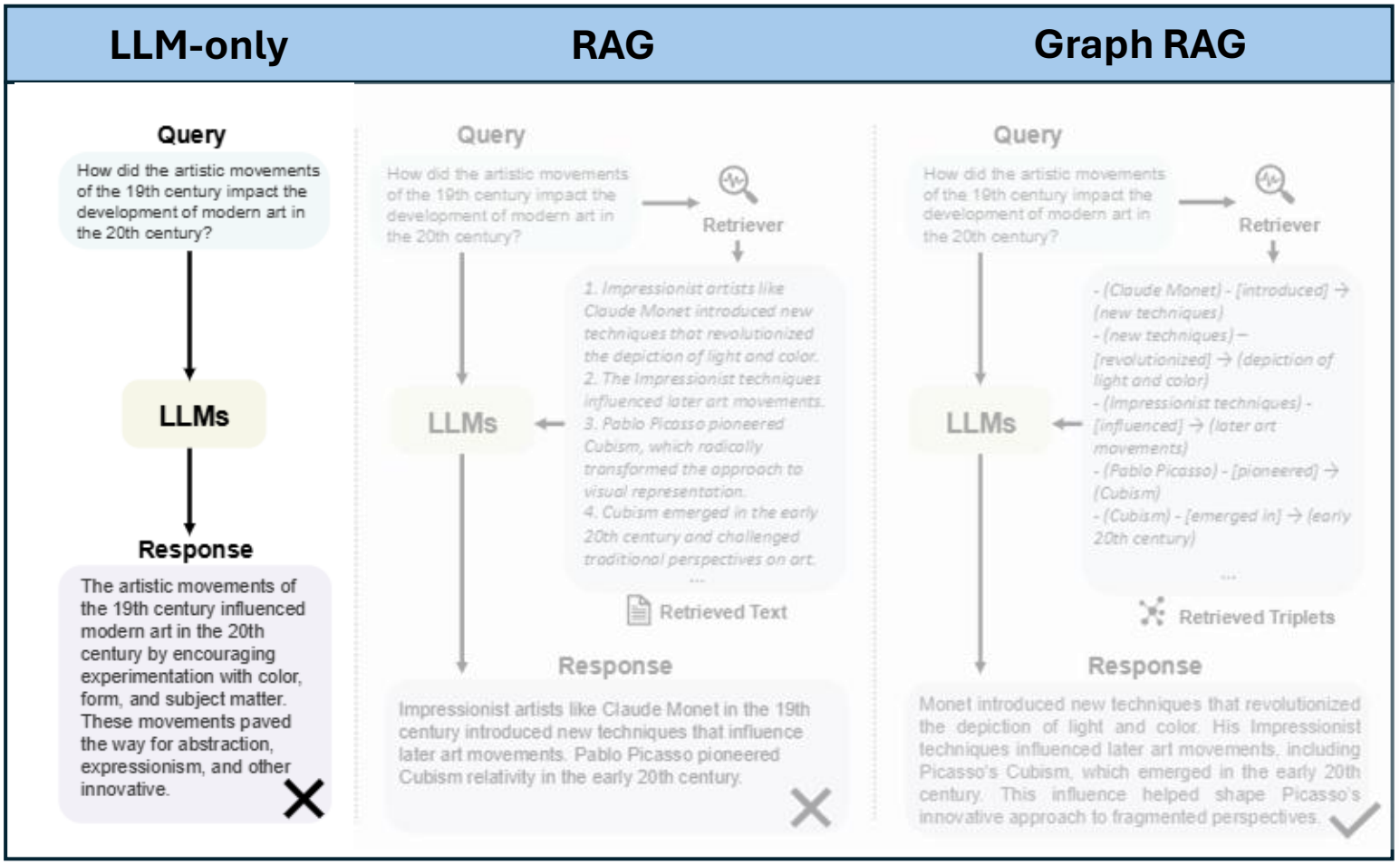
- **SP-based methods** (Semantic Parsing): 질문을 Logical form/Query로 변환해 KG 실행
- **IR-based methods** (Information Retrieval): 후보 Path/Subgraph/Triples을 뽑고 랭킹으로 답 선택



Background

LLM-only Question Answering

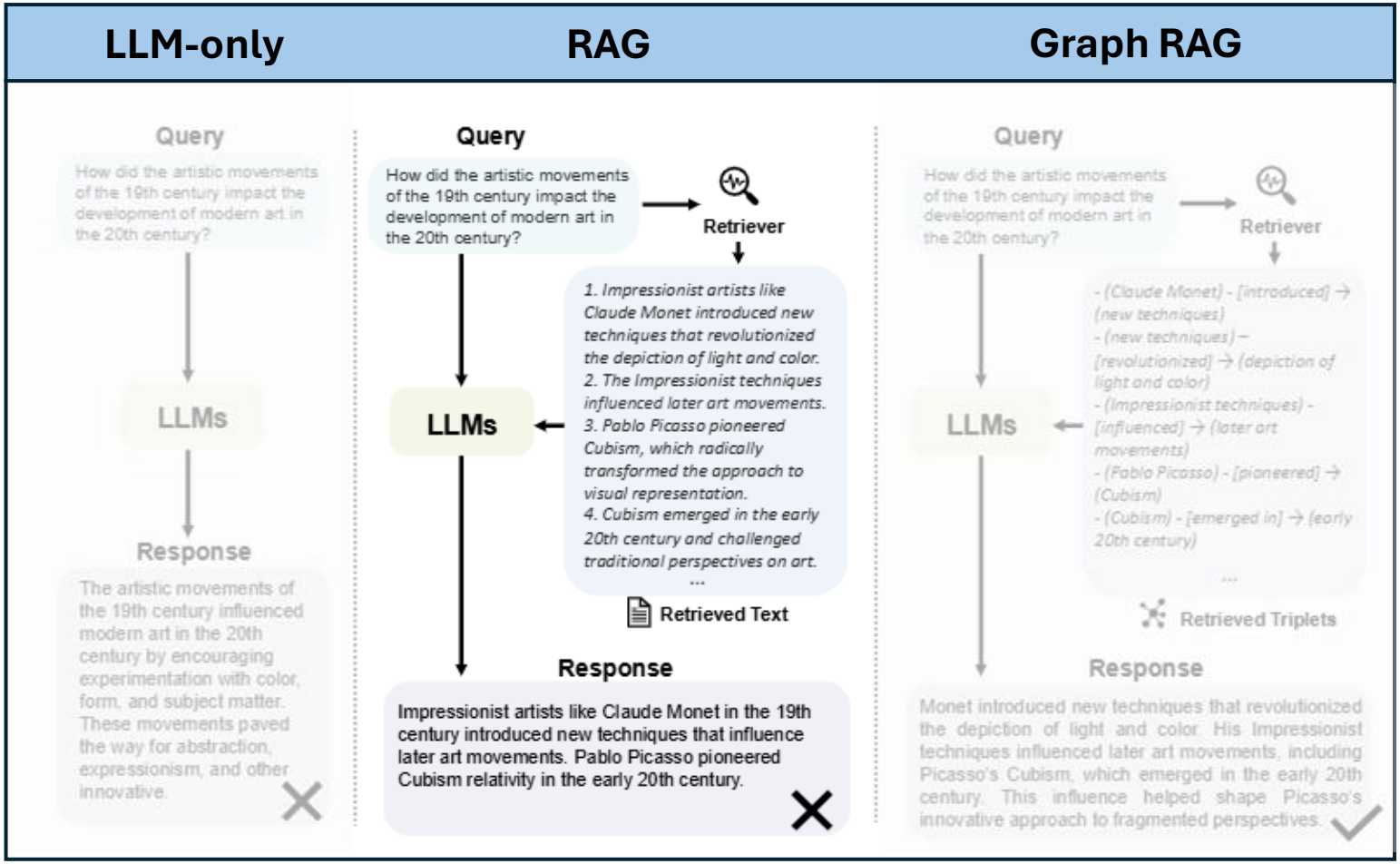
- LLM만으로 질의에 답하거나 reasoning을 수행
  → 빠르지만 hallucination & outdated information 문제 존재



Background

Retrieval-Augmented Generation(RAG) QA

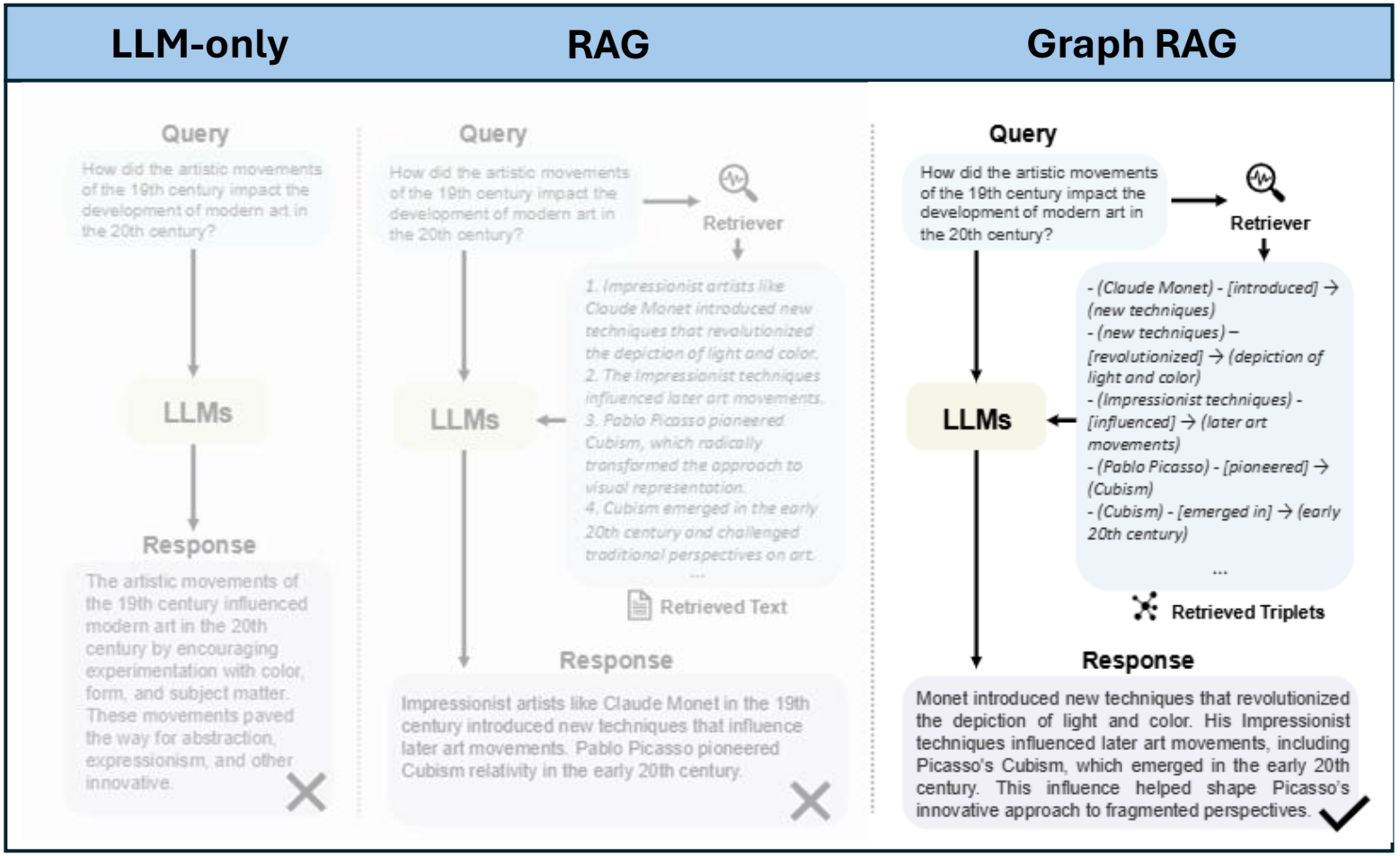
- Text corpus를 검색하여 LLM 입력
  → lost in the middle & Neglecting Relationships 한계 존재



Background

Graph RAG(Retrieval-Augmented Generation) QA

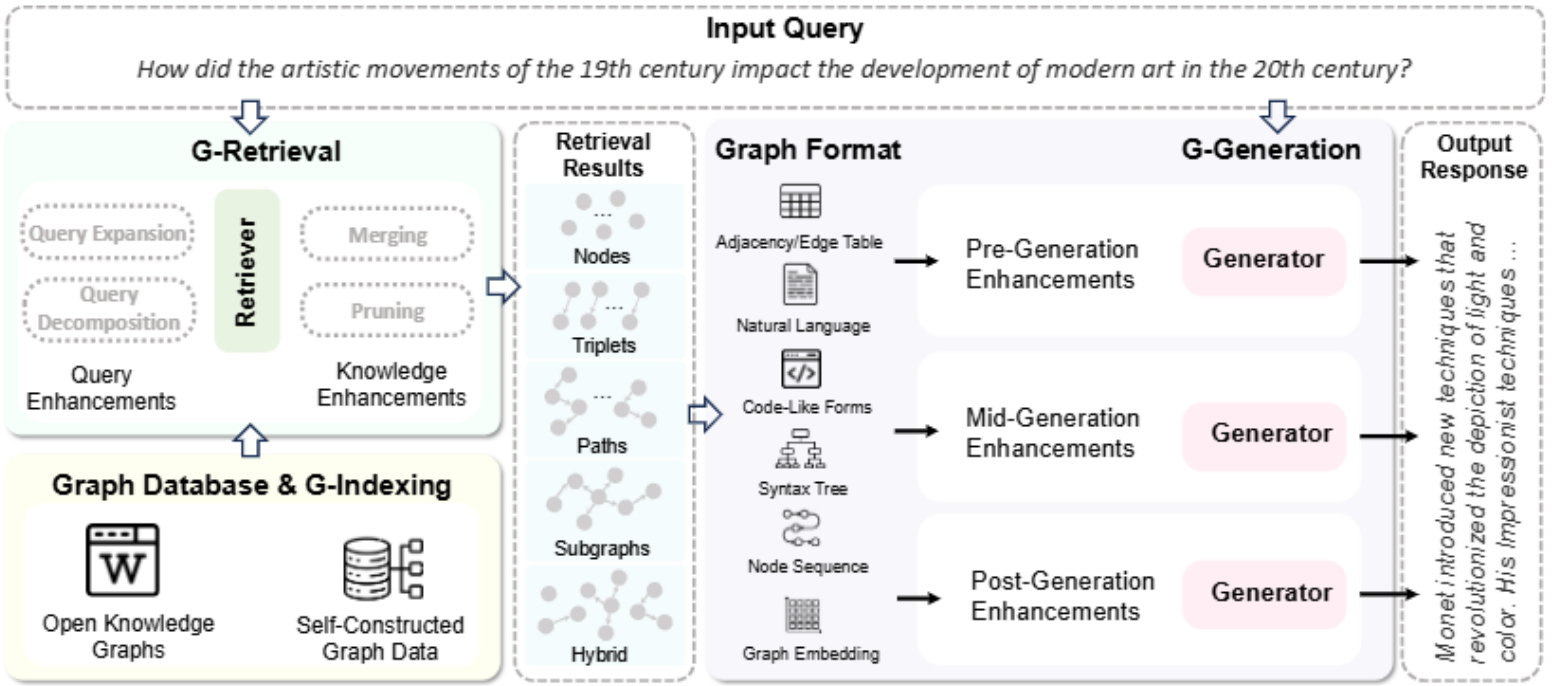
- KG의 triple 등 구조적 요소를 검색해 LLM에 제공
  → High structural consistency & Global information



Background

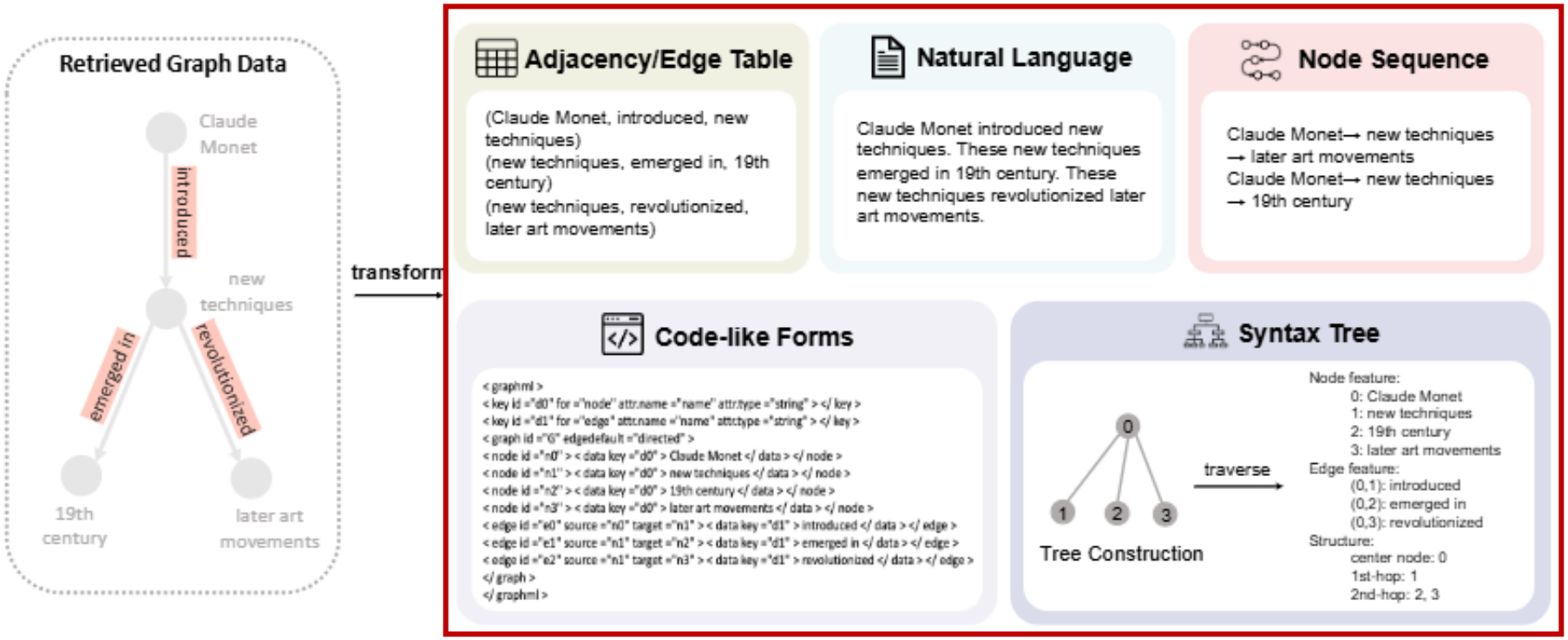
What is the ‘GraphRAG’

- GraphRAG Workflow 3단계
  - ✓ G-Indexing -> G-Retrieval -> G-Generation



Background

What is the ‘GraphRAG’

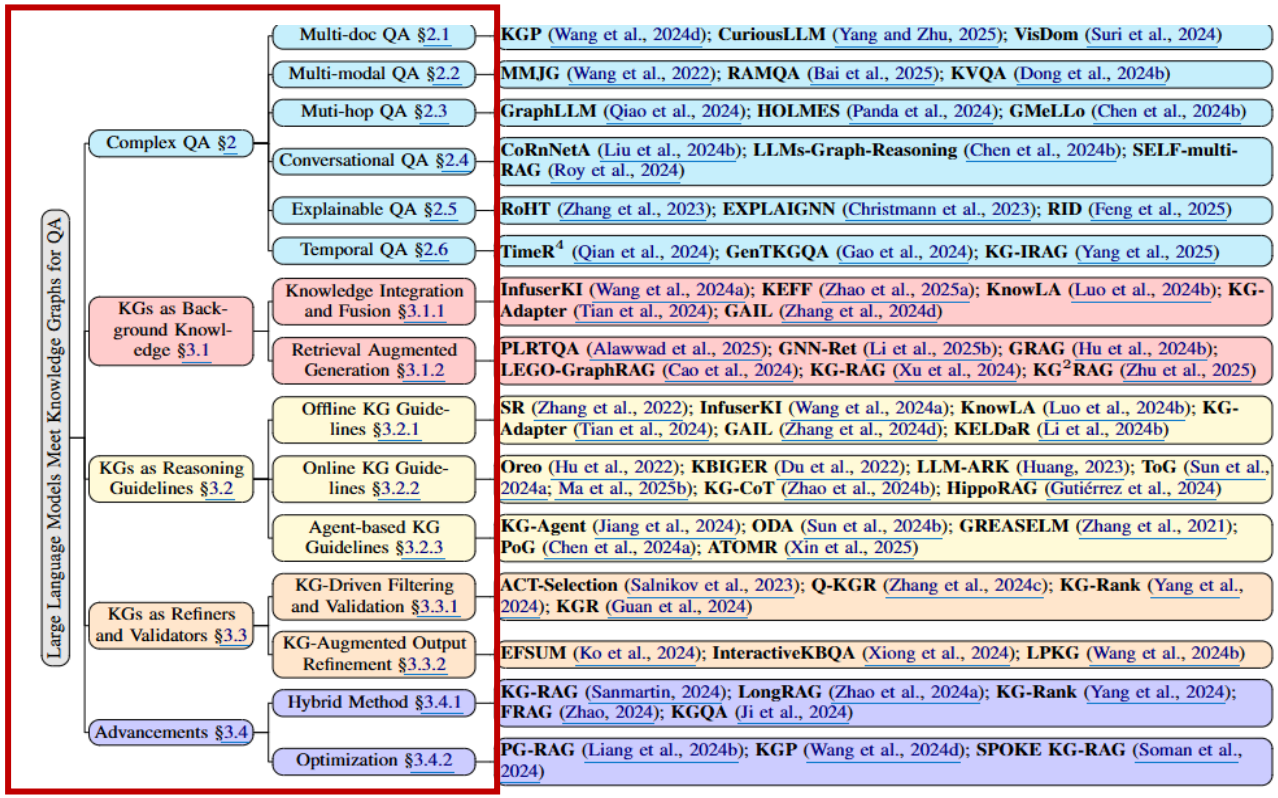


CONTENTS

Related Work

A Structured Taxonomy of Synthesizing LLMs and KGs for QA

- Complex qa – 여러 문서, 대화형, 시간 등이 섞인 어려운 질문으로, 한 조각이 아닌 여러 단서들을 잘 모아야 하는 QA
- KGs as Background Knowledge – KG를 참고서처럼 사용하여 query가 들어오면 관련 정보를 찾아 LLM등에 넣어주는 방식
- KGs as Reasoning Guidelines – KG를 하나의 guide로 써서 질문에 대해 어떤 순서로 reasoning 해야 하는지 안내하는 방식
- KGs as Refiners and Validators – LLM이 생성한 후보 답(claim)마다 KG로 근거를 찾아 검증하고 재정렬하는 방식
- Advancements – 위와 같은 방법들을 더 빠르고, 비용이 적게 만드는 방식



[2025][ACL-F][FGSP]

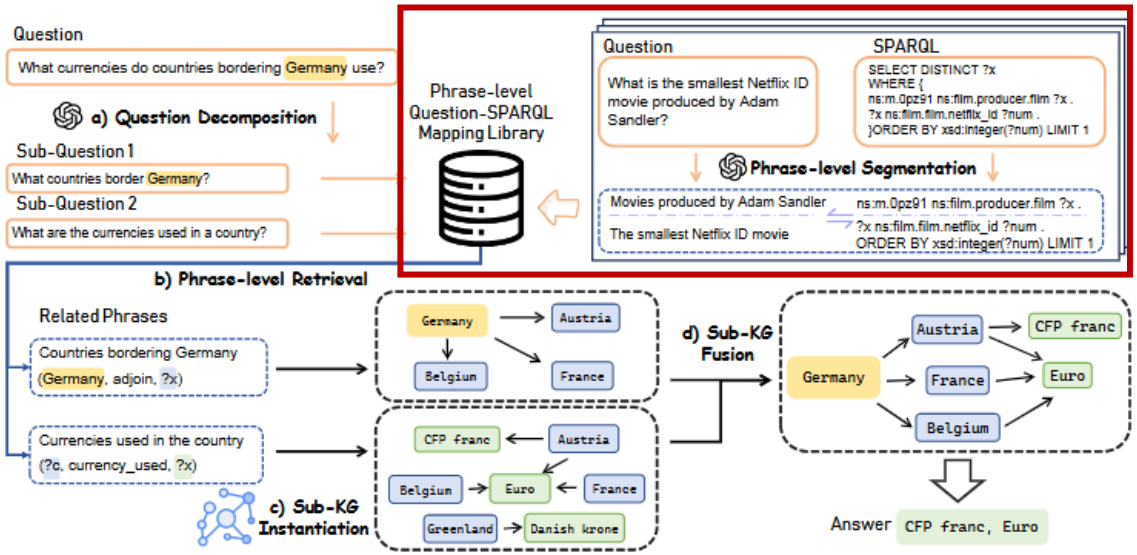
Semantic Parsing-based QA

From Phrases to Subgraphs: Fine-Grained Semantic Parsing for Knowledge Graph Question Answering

Yurun Song*, **Xiangqing Shen***, **Rui Xia**†
School of Computer Science and Engineering,
Nanjing University of Science and Technology, China
{yrsong, xqshen, rxia}@njust.edu.cn

1. Question-SPARQL Mapping Library Construction

- Offline 2단계
  - ✓ Rule-based SPARQL parser가 question 전처리(SPARQL Where절에서 triple pattern을 추출하여 병합)
  - ✓ 전처리된 SPARQL 조각 + 원본 SPARQL + Question input하여 Question-SPARQL pairs 생성



Phrase-level Segmentation (Logical Form)

[t1: : ?x ns:adjoin Ns:m.Germany]
[t2: ?x ns:currency_used ?currency]

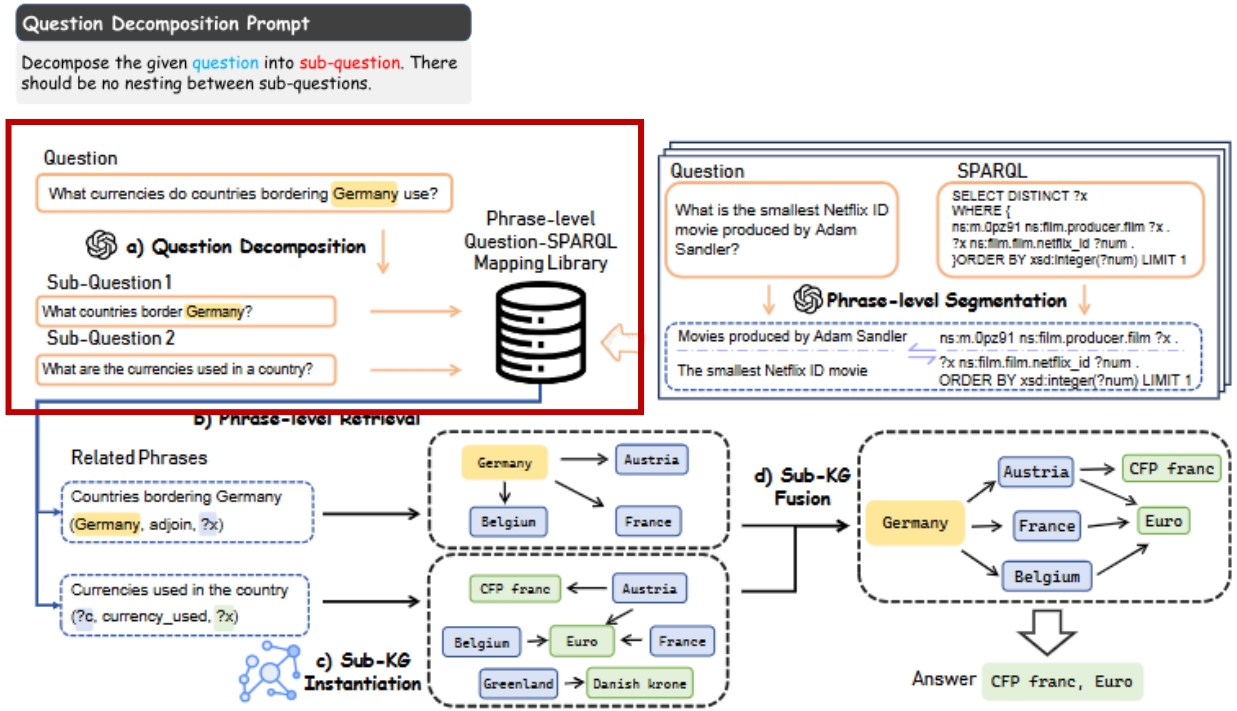
Query SPARQL(원문)

LLM

Question-SPARQL Mapping

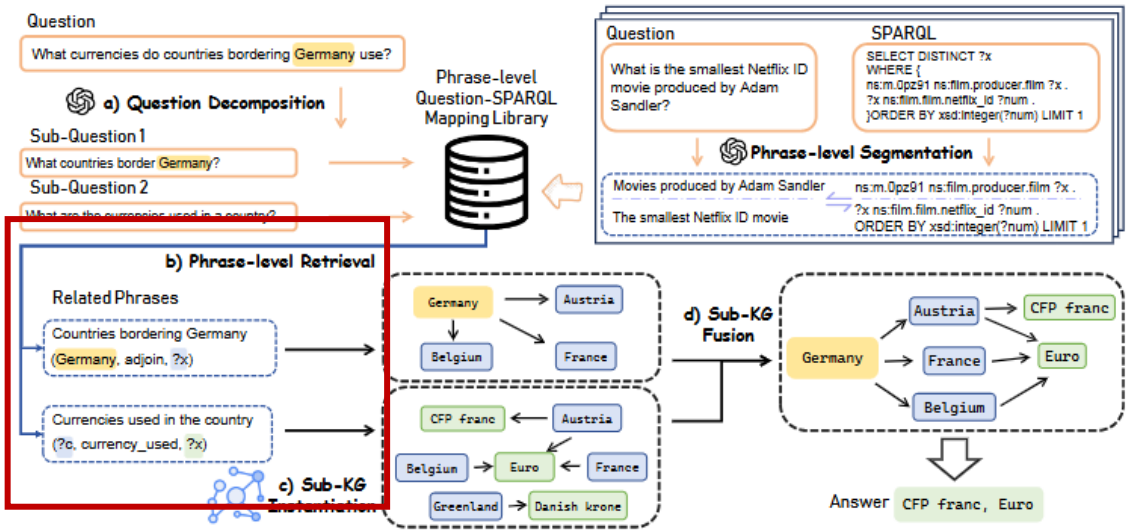
1. Question Decomposition

- Online
  - ✓ Query가 들어오면 Decomposition하여 아래처럼 sub-query로 분해



2. Phrase-level Retrieval

- 분해된 Sub-query별로 Cosine Similarity 기반 유사한 Pharse의 SPARQL을 가져옴



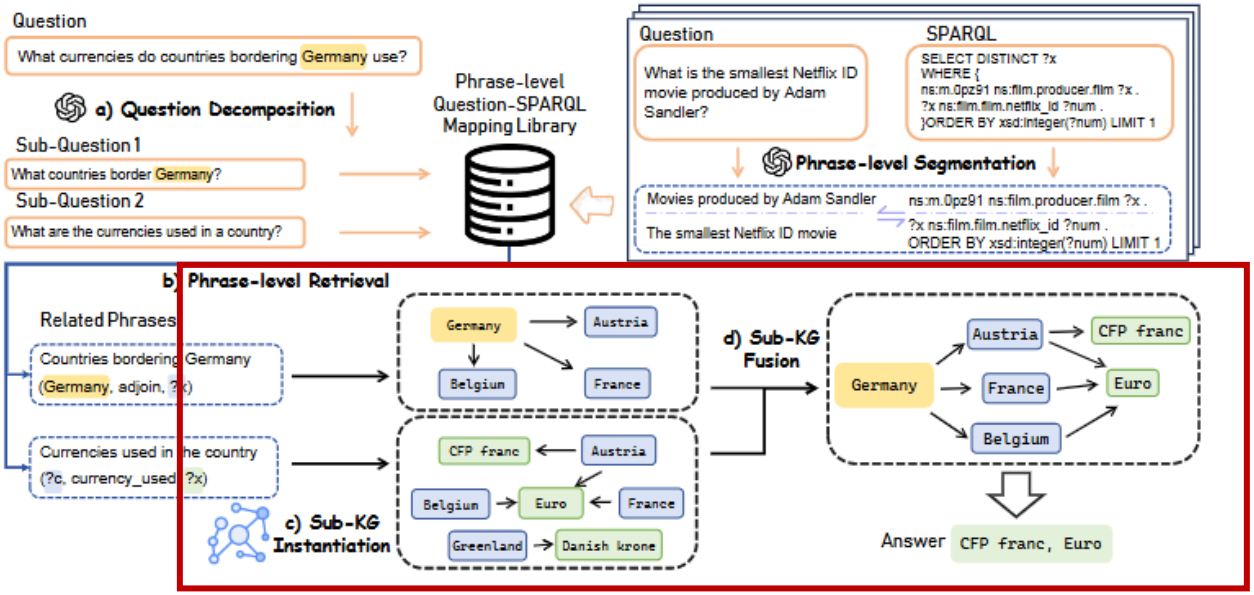
Sub-question: "What countries border Germany?"

P_1: "Countries bordering Germany"
↔
S_1: ?c ns:location.country.adjoin Ns:m.Germany

P_2: "Currencies used in the country"
↔
S_2: ?c ns:location.country.currency_used ?currency

3. Sub-KG Instantiation and Fusion

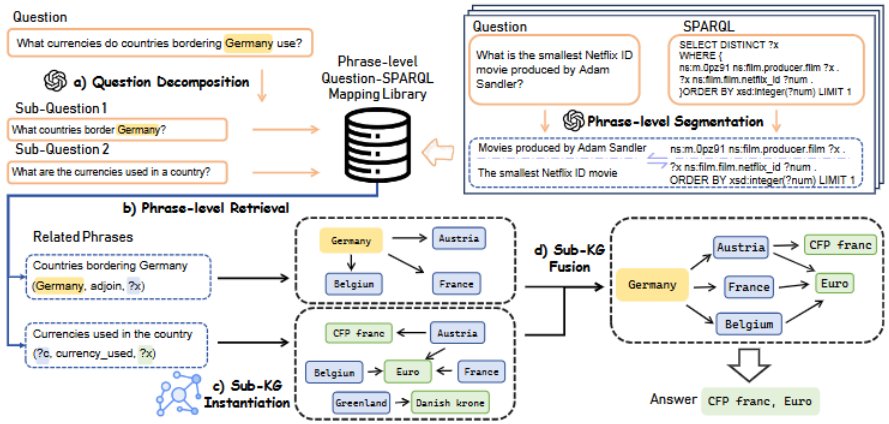
- 추출된 SPARQL 조각들에 대해 가능한 entity를 모두 연결하여 Local Subgraph 생성
- 이후 Rule-based로 Subgraph들을 fusion하여 하나의 Graph를 생성

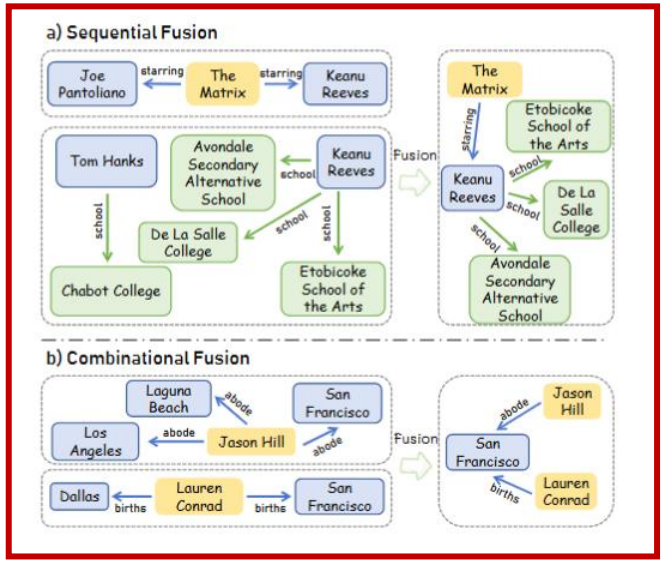


Subgraph
Germany –adjoins→ Austria
Germany –adjoins→ France
Germany –adjoins→ Belgium

4. Sub-KG Fusion

- Fusion 2가지 방식
  - ✓ Sequential Fusion
    - S1과 S2의 공통 변수 ?country 를 동일하게 두고 WHERE절에 함께 넣음
  - ✓ Combinational Fusion
    - S1과 S2에 대해 AND로 결합하여 SPARQL을 합성

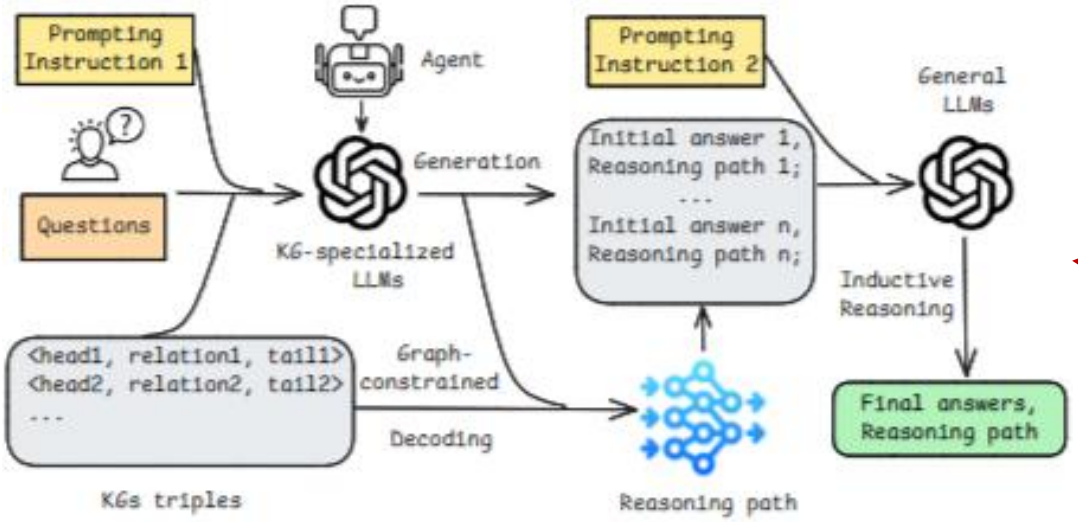


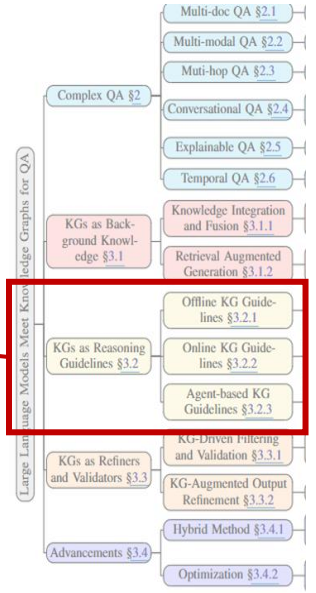


KGs as Reasoning Guideline

Agent-based KG Guidelines

- Agent가 질의 분해·KG 탐색을 조정하고, KG-specialized LLM들이 KG 기반의 초기 답안과 reasoning path를 생성
- General LLM이 이를 종합(Inductive reasoning)해 최종 답안과 근거를 생성





KGs as Reasoning Guideline

Agent-based KG Guidelines

KG-Agent: An Efficient Autonomous Agent Framework for Complex Reasoning over Knowledge Graph

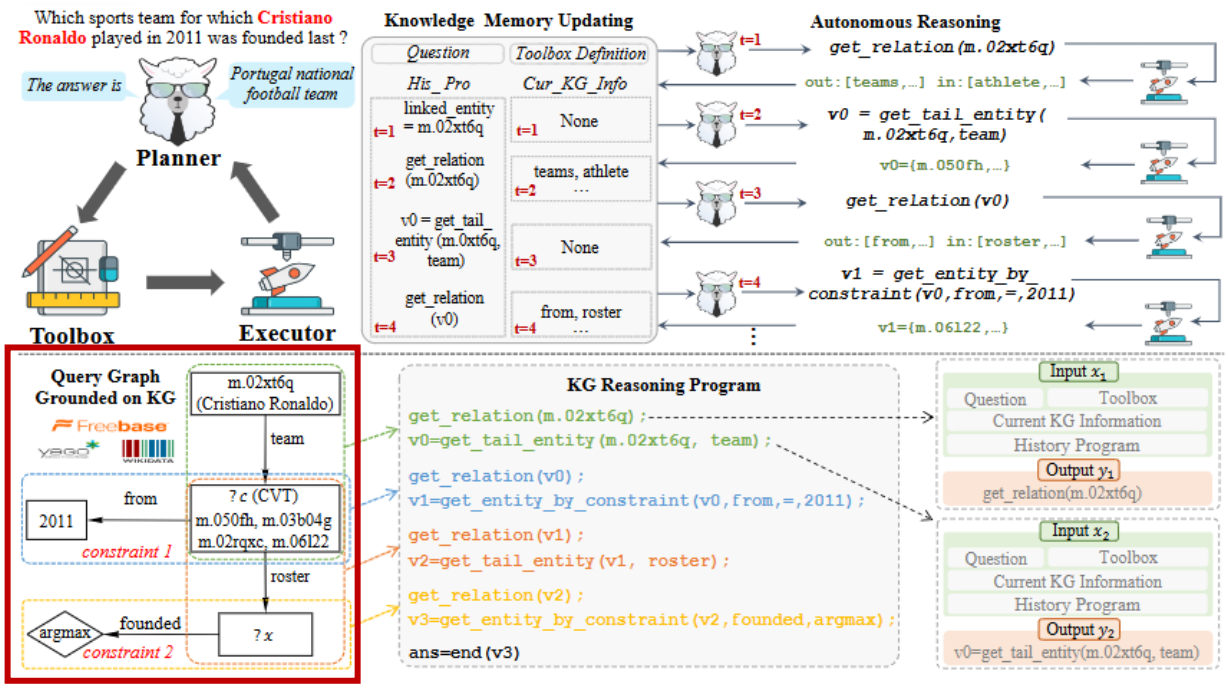
Jinhao Jiang[1], Kun Zhou[2], Wayne Xin Zhao[1]*, Yang Song[3]*,
Chen Zhu[4], Hengshu Zhu[4], Ji-Rong Wen[1]
[1]Gaoling School of Artificial Intelligence, Renmin University of China.
[2]School of Information, Renmin University of China.
[3]Nanbeige Lab, BOSS Zhipin. [4]Career Science Lab, BOSS Zhipin.
jiangjinhao@ruc.edu.cn, batmanfly@gmail.com

[2025][ACL][KG-Agent]

- Query 와 그에 대응하는 SQL/SPARQL를 가지고 reasoning program을 생성하여 자율적으로 추론할 수 있는 LLM 기반 에이전트를 개발하는 모델
- Offline
  - ✓ 질문에 해당하는 SPARQL을 가지고 Query Graph(Subgraph) 생성

Online

Offline

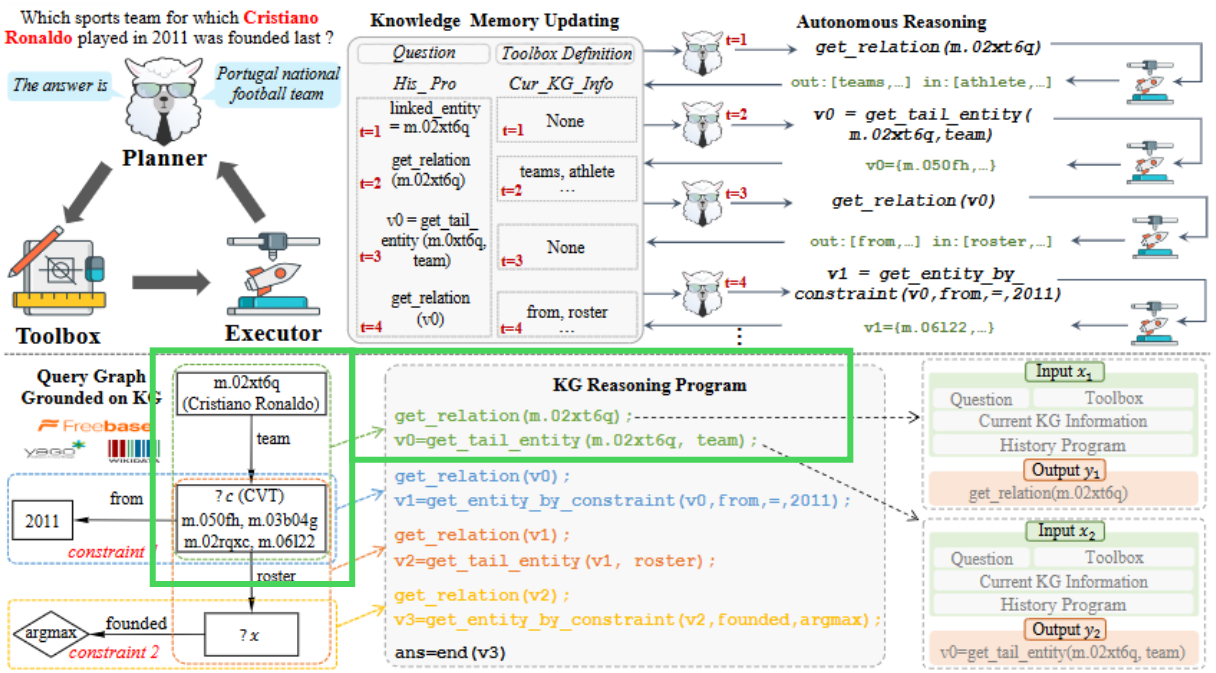


[2025][ACL][KG-Agent] KG-Agent: An Efficient Autonomous Agent Framework for Complex Reasoning over Knowledge Graph

[2025][ACL][KG-Agent]

- 정의된 Query Graph에서 각 박스(triple)마다 적절한 함수(Tool)를 호출
  - ✓ 이 때 Tool(get_relation, get_tail_entity)들은 Toolbox에 정의되어있고 Planner가 계획, Executor가 호출
    - Executor가 조회한 정보들은 Cureent KG Information에 저장됨
    - Planner가 생성해서 실행된 함수 호출 이력들은 History Program에 저장됨

Offline

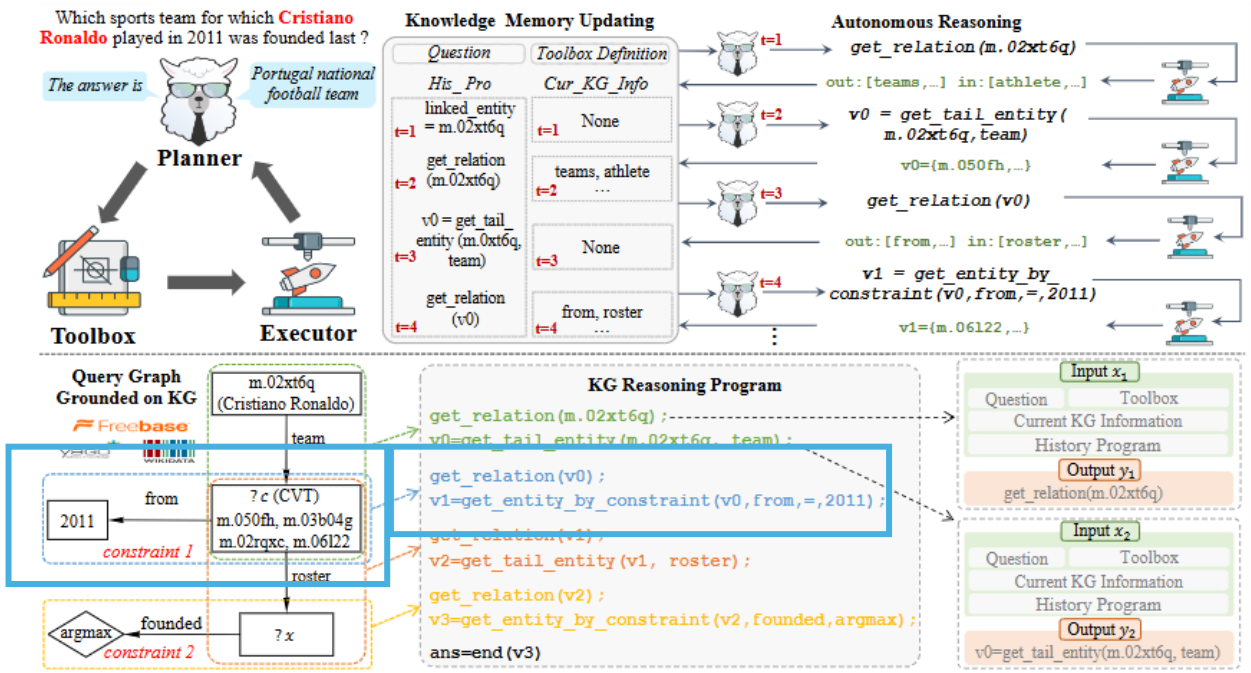


- 연두색 박스
  - ✓ get_relation 함수로 'team' 출력(y1)
  - ✓ get_tail_entity 함수로 호날두가 속한 team들을 출력(y2이자 새로운 input 변수 v0로 다음 함수에 input됨)

[2025][ACL][KG-Agent]

- 정의된 Query Graph에서 각 박스(triple)마다 적절한 함수(Tool)를 호출
  - ✓ 이 때 Tool(get_relation, get_tail_entity)은 Toolbox에 정의 되어있고 Planner가 계획, Executor가 호출
  - ✓ 이 과정을 반복하여 생성된 Instruction data로 'Planner'를 Supervised Fine-tuning
    - LLM이 주어진 memory context에서 어떤 tool을 계획/호출해야 하는지를 학습함

Offline

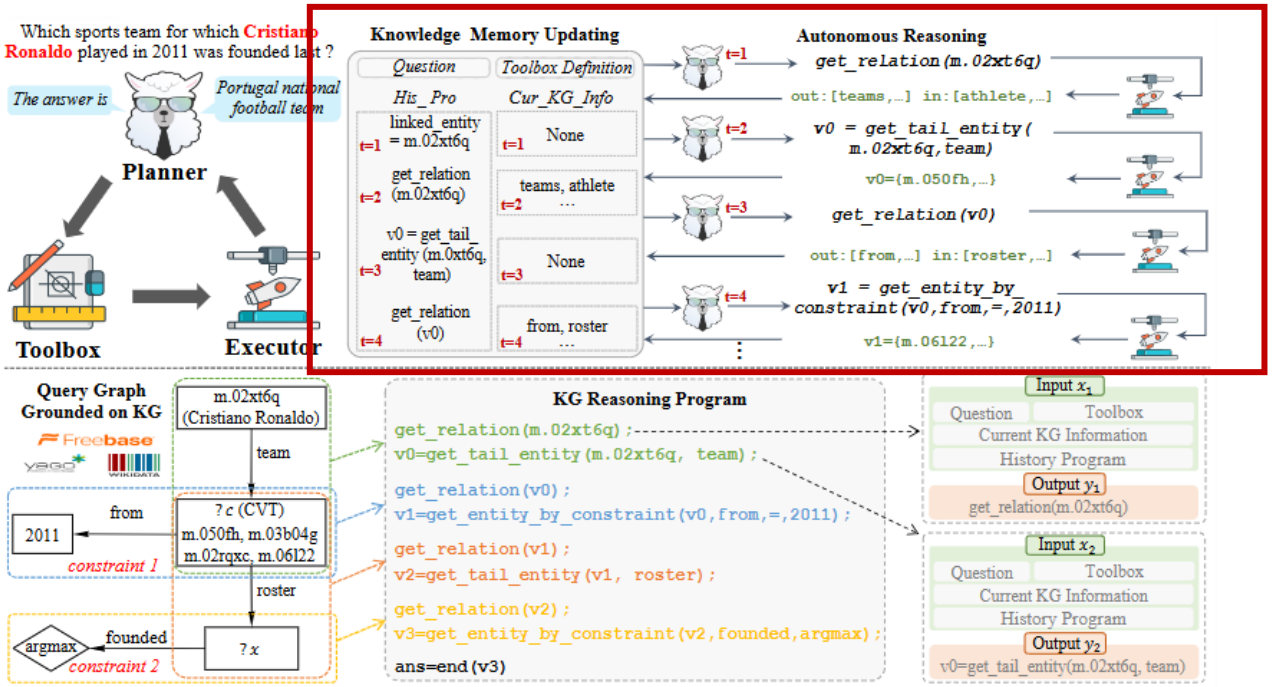


- 하늘색 박스
  - ✓ get_tail_entity 함수로 출력된 호날두가 속했던 팀들에 대해 다시 한번 get_relation으로 'from' 출력
  - ✓ 이후 출력된 호날두 '팀' 중에서 '2011'에 소속되어있던 '팀'을 출력

[2025][ACL][KG-Agent]

- 정의된 Query Graph에서 각 박스(triple)마다 적절한 함수(Tool)를 호출
  - ✓ 이 때 Tool(get_relation, get_tail_entity)은 Toolbox에 정의 되어있고 Planner가 계획, Executor가 호출
    - Offline에서는 KG grounding으로 완성된 query graph 탐색했지만, Online은 하나씩 직접 계획을 세우고 탐색하며 확장

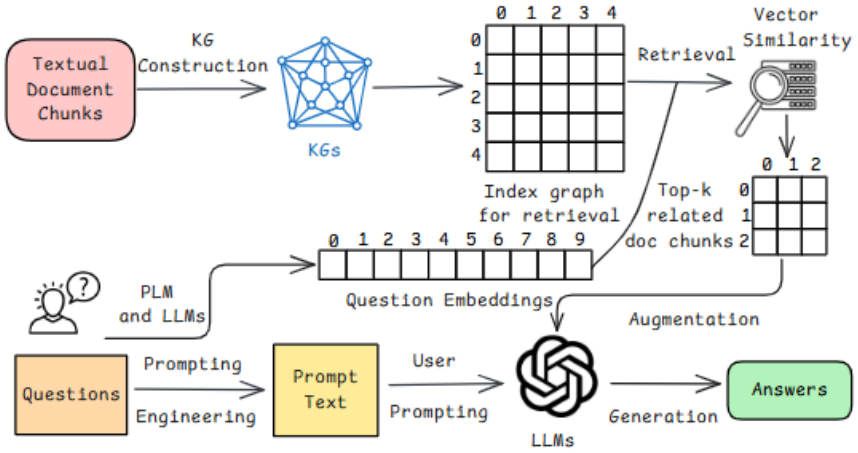
Online

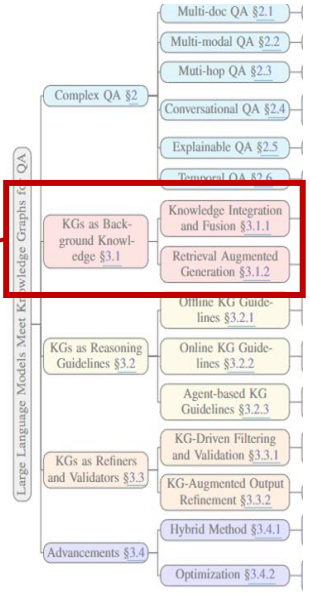


KGs as Background Knowledge

Retrieval Augmented Generation

- RAG(Retrieval-Augmented Generation)와 KG(knowledge graph)를 결합한 QA 파이프라인으로,
- 상단은 문서로부터 KG를 구성하고 그래프 기반 인덱스를 만들어 벡터 유사도로 관련 문서 청크 탐색
- 하단은 사용자 질문을 프롬프트로 가공해 LLM에 입력하고, 상단에서 찾은 top-k 컨텍스트로 LLM을 보강(augmentation)해 정답을 생성





KGs as Background Knowledge

GNN-Based QA

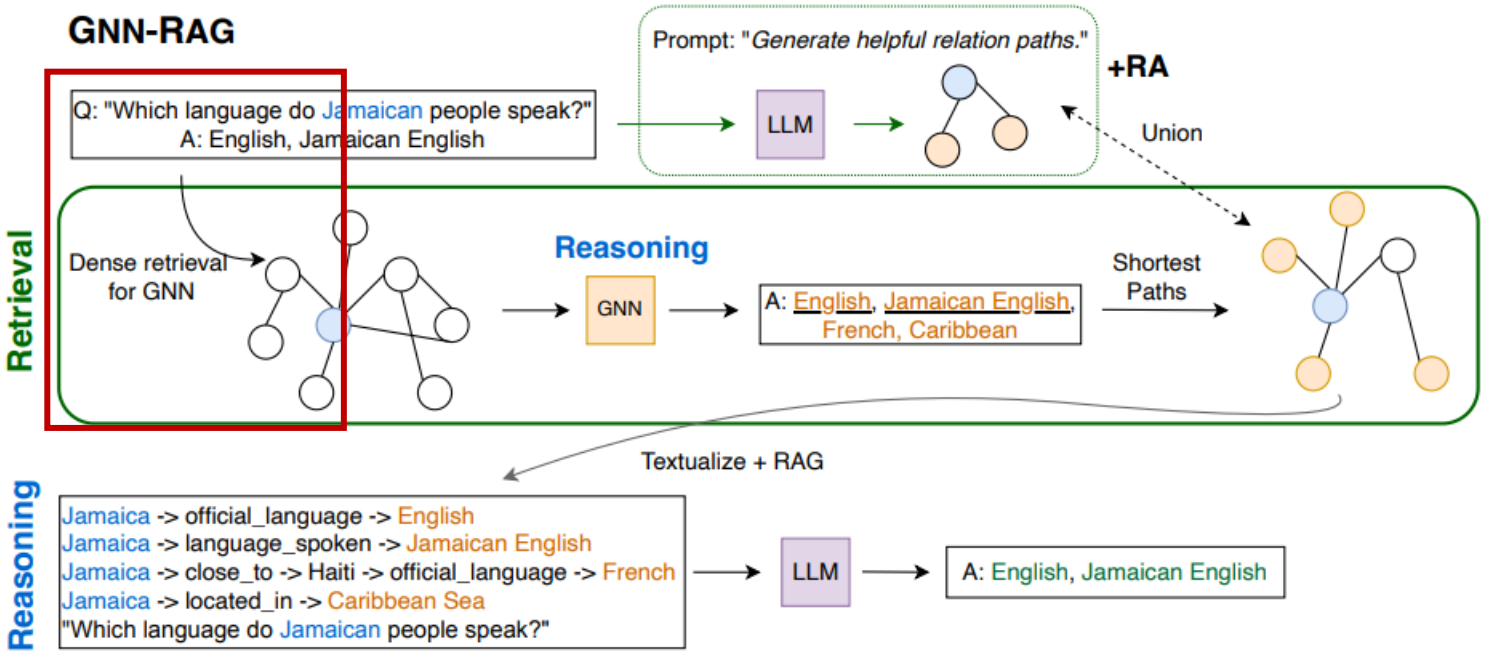
GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning

Costas Mavromatis
University of Minnesota
mavro016@umn.edu

George Karypis
University of Minnesota
karypis@umn.edu

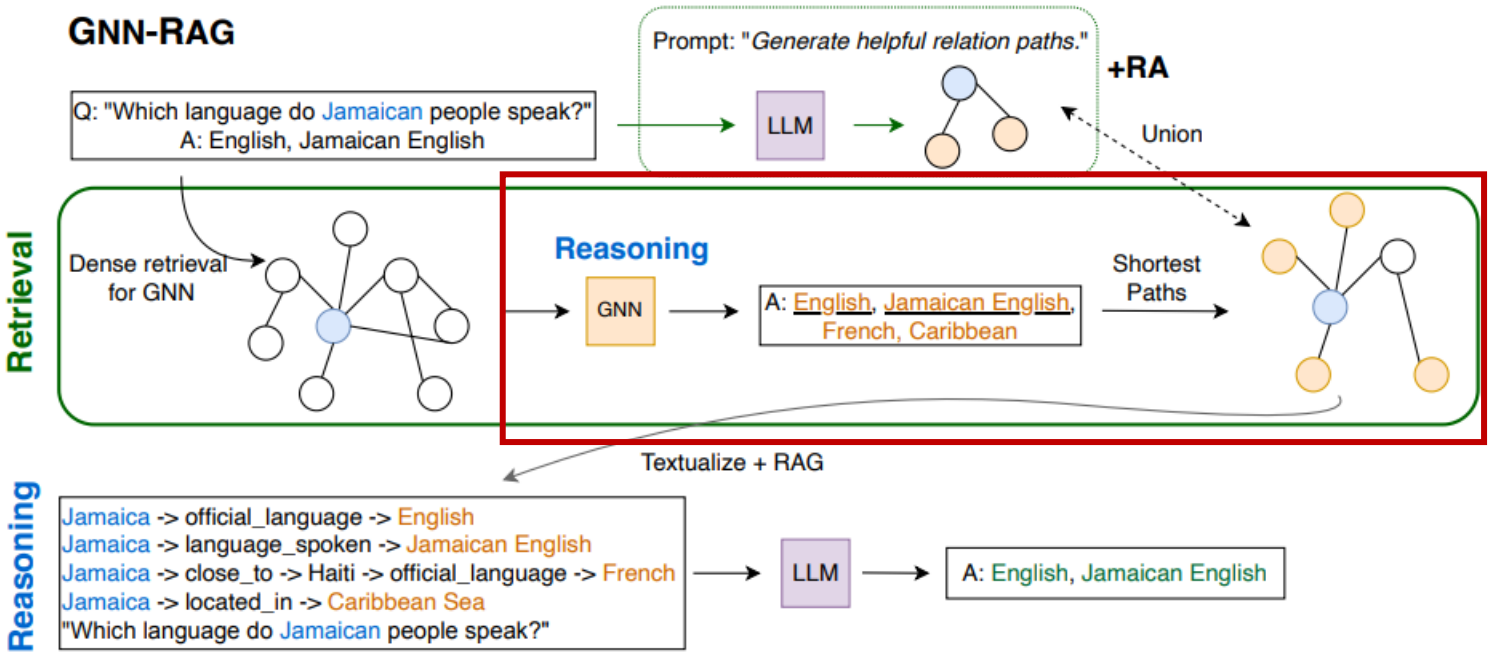
[2025][ACL-F][GNN-RAG]

- 검색된 근거들을 그래프로 구성하고 GNN으로 관계 추론한 뒤, evidence를 LLM에 주입해 더 정확한 답변을 유도하는 모델
- Dense subgraph extraction
  - ✓ PageRank 기반 neighborhood 확장하여 query-specific dense subgraph 생성



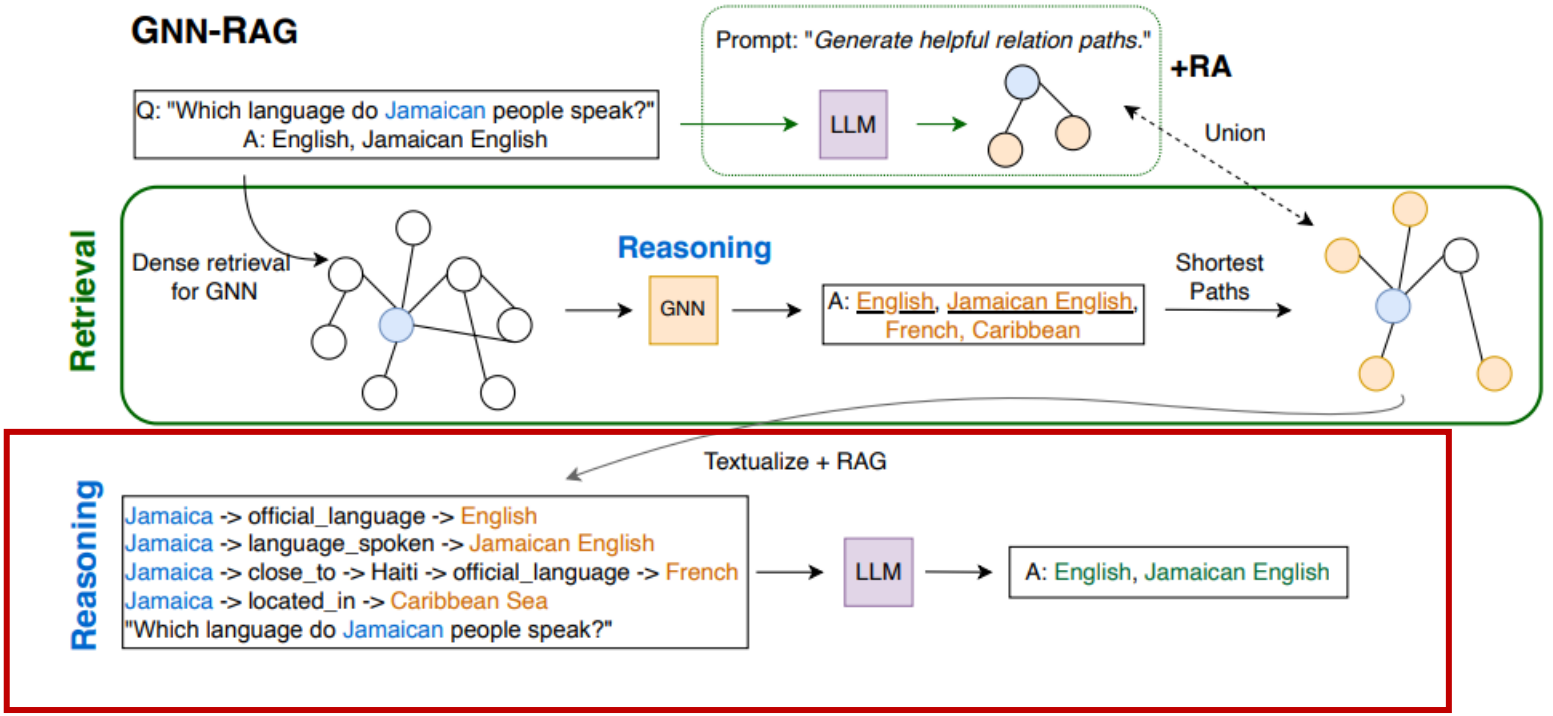
[2025][ACL-F][GNN-RAG]

- Reasoning paths 추출
  - ✓ 해당 Subgraph 위에서 GNN을 돌려 각 노드가 answer일 확률을 구하고, top K를 택
  - ✓ 선택된 top K의 후보 entity과 topic entity를 연결하는 shortest path를 추출
  - ✓ 해당 경로들을 "Jamaica → language_spoken → English" 형태로 텍스트화



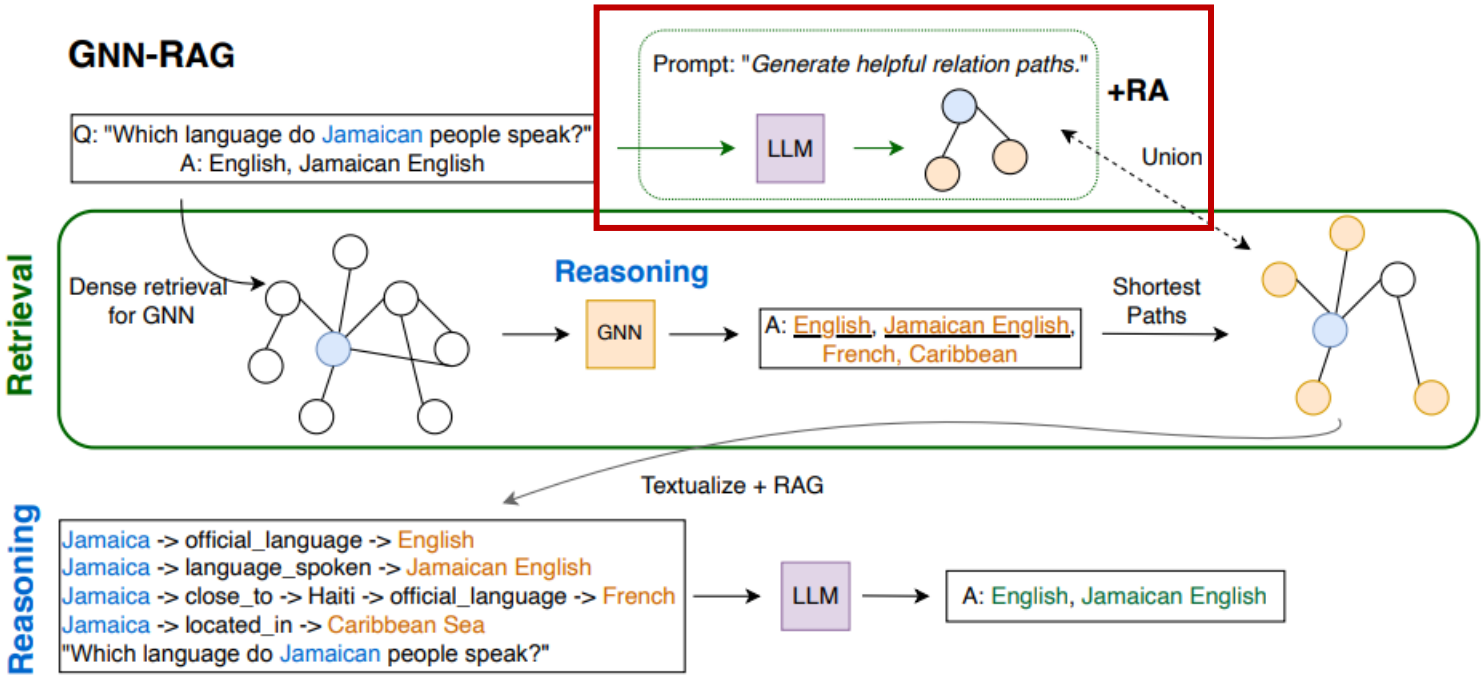
[2025][ACL-F][GNN-RAG]

- Verbalized paths-based Answer Generation
  - ✓ 텍스트 형태의 Reasoning paths을 LLM에게 전달하여 최종 Answer 생성



[2025][ACL-F][GNN-RAG]

- RA(Retrieval Augmentation) – (GNN/LLM 결과 병합)
  - ✓ GNN으로부터 온 경로들과 LLM 기반 경로들의 Union하여 정보 다양성과 리콜을 증가
    - GNN은 그래프 구조를 깊게 탐색하고 multi-hop 관계를 포착하는 데 강함
    - 반면 LLM은 자연어 이해를 이해하고 semantic matching에 강함



KGs as Background Knowledge

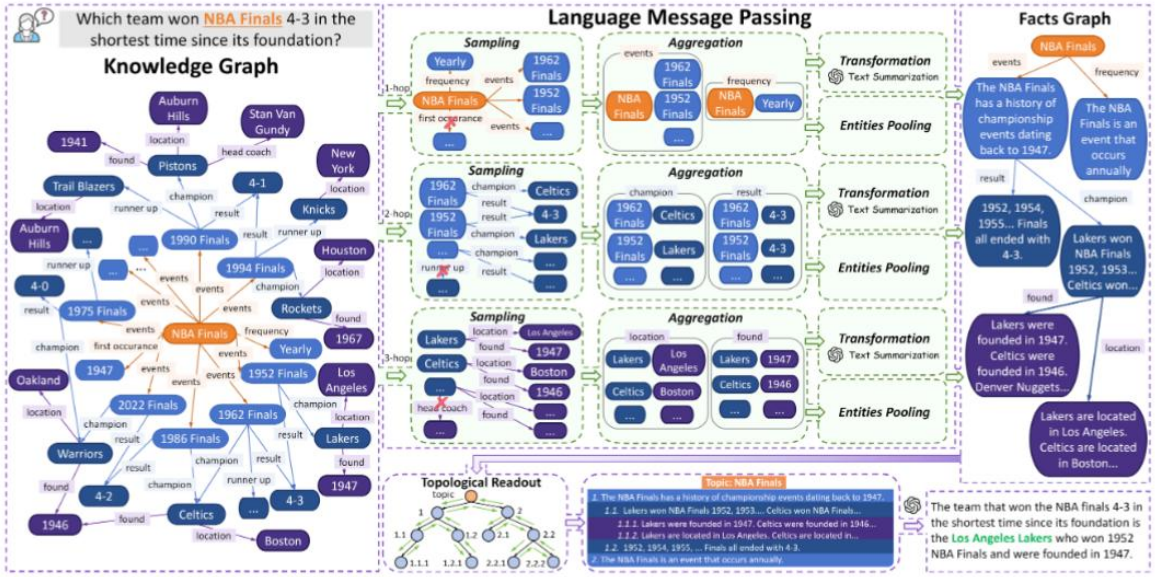
LLM-Based QA

Digest the Knowledge: Large Language Models empowered Message Passing for Knowledge Graph Question Answering

Junhong Wan, Tao Yu, Kunyu Jiang, Yao Fu*, Weihao Jiang, Jiang Zhu
Hikvision Research Institute, Hangzhou, China
{wanjunhong,yutao31,jiangkunyu,fuyao,jiangweihao5,zhujiang.hri}@hikvision.com

[2025][ACL][LMP]

- 기존 KGQA Logical path 방식의 한계(topology information 손실, 단순 keyword 나열로 인한 digest 어려움 등)을 언급하며 Message Passing 형태로 KGQA를 수행하는 모델
  - ✓ Sampling => Aggregation => Transformation => Readout 순으로 진행
  - ✓ GNN의 message passing을 text형태로 진행하는 모델



Aggregation prompt of LMP

The entity ⟨HeadEntity⟩ has relation ⟨Relation⟩ with following entities: ⟨TailEntity$_1$⟩, ⟨TailEntity$_2$⟩, . . . , ⟨TailEntity$_n$⟩.

Transformation prompt at layer 1 of LMP

Given the question, we have K facts about its topic and related relations that may helpful to answer the question.
question: ⟨Question⟩
topic: ⟨Topic⟩

Please summarize each following fact while only keeping every relevant information about the question and just return all summarized facts as following order in the same numbered list without explanation.
facts:
1. ⟨AgregatedInformation$_1$⟩
2. ⟨AgregatedInformation$_2$⟩
. . .
K. ⟨AgregatedInformation$_K$⟩

1. Sampling

- **Question :** "Who is the prime minister of the country that has Loma?"
  - **Topic Entity:** 'Loma'
  - **Gold Entity:** 'Hailemariam Desalegn'


```
====================output=======================
Here are the top 2 relations selected to explore about the topic:

1. location.administrative_division.country
2. location.location.containedby
```

2. Aggregation

- **Question :** "Who is the prime minister of the country that has Loma?"
  - Prompt: The <head entity> has <relation> with following entities: ~~


```
facts:

1. The entity Loma has relation location.administrative_division.country with following entities: Ethiopia.
2. The entity Loma has relation location.location.containedby with following entities: Dawro Zone; Southern Nations, Nationalities.
```

3. Transformation

❖ **Question :** "Who is the prime minister of the country that has Loma?"

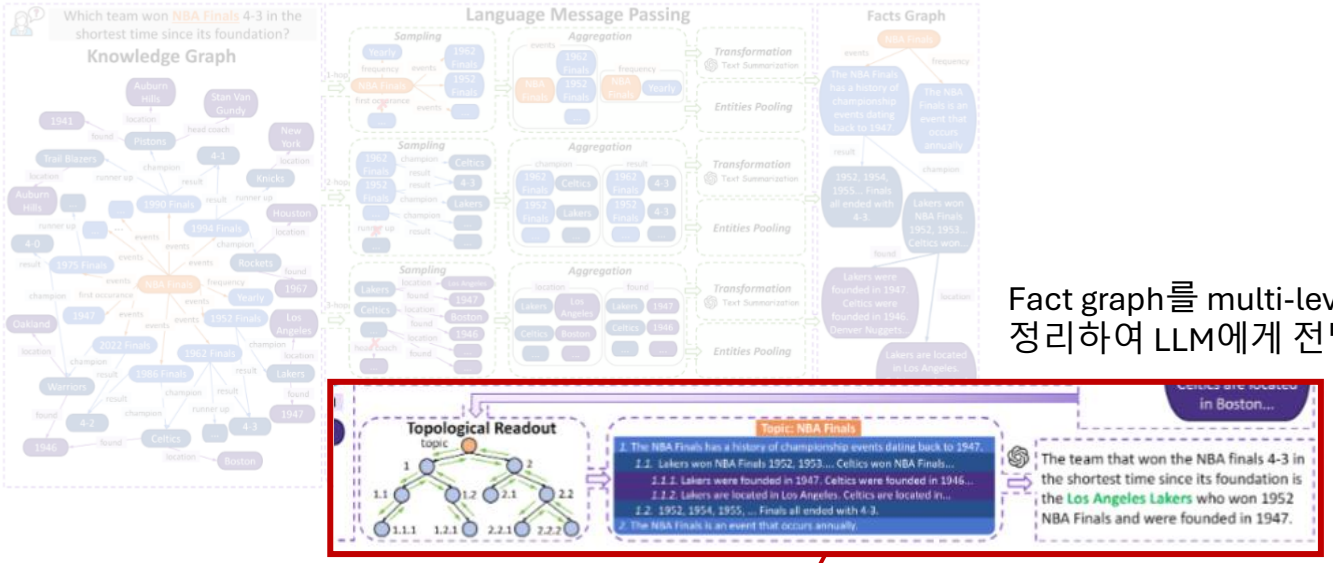
- Prompt:
  - Here is the list of summary:
    1. Summarized Fact
    2. Summarized Fact


```
==================output=====================
Here is the list of summarized facts:

1. Loma is located in the country of Ethiopia.
2. Loma is contained within Dawro Zone, Ethiopia, and the Southern Nations, Nationalities, and Peoples' Region.
==============================================
```

[2025][ACL][LMP] Digest the Knowledge: Large Language Models empowered Message Passing for Knowledge Graph Question Answering

4. Topological Readout



Fact graph를 multi-level numbered list 형태로 정리하여 LLM에게 전달하여 answer 생성

```
facts:
Here are some facts about topic Loma that may related to the question.

1. Loma is located in the country of Ethiopia.
    1.1. Ethiopia has the ISO 3166-1 alpha-2 code ET.
    1.2. Ethiopia has the ISO 3166-1 short name ETHIOPIA.
    1.3. Loma is located in the country of Ethiopia.

2. Loma is contained within Dawro Zone, Ethiopia, and the Southern Nations, Nationalities, and Peoples' Region.
    2.1. Ethiopia has the ISO 3166-1 alpha-2 code ET.
    2.2. Ethiopia has the ISO 3166-1 short name ETHIOPIA.
```

5. Limitation

- Multi-hop QA task의 경우 LLM 사용빈도가 굉장히 높은 편이며, 이에 크게 의존하고 있는 경우가 많음
- Question Answering task 자체가dataset 내 모든 query에 대해 분석을 진행하고 답을 생성하는 파이프라인을 가지고 있기에 상대적으로 비용과 시간이 많이 투자됨

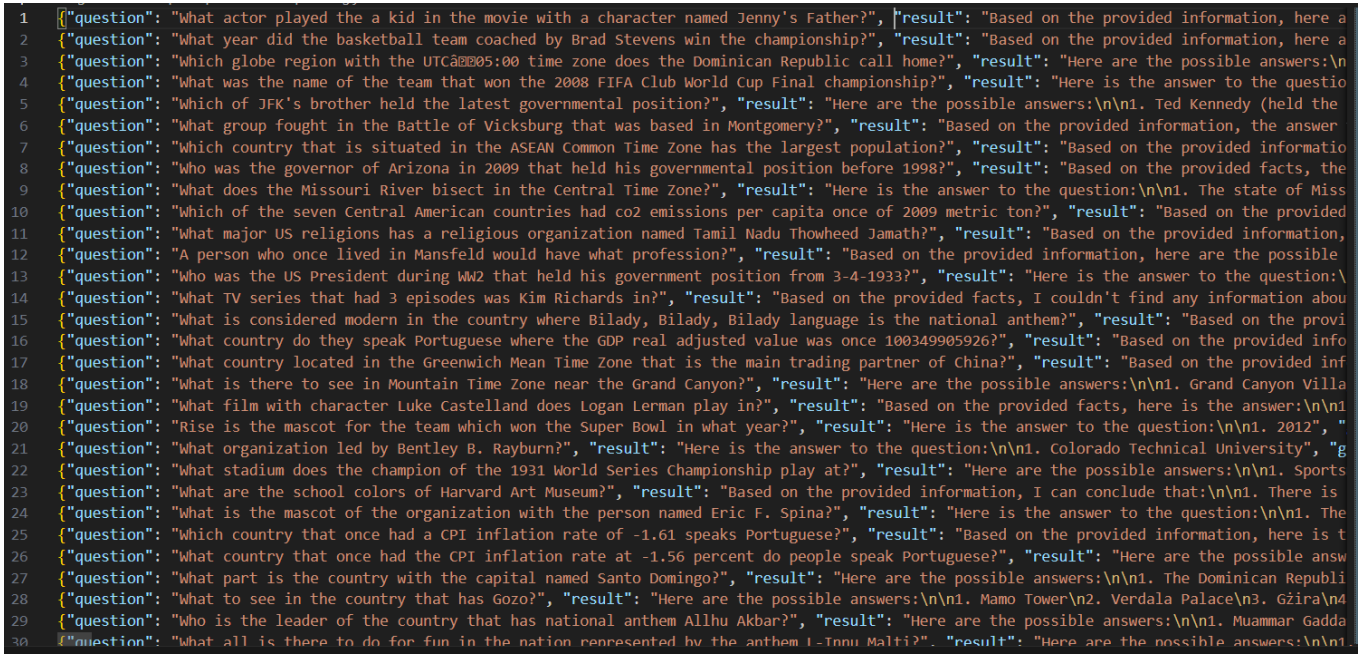
'LLM : 2l + 1, (l = hop)'

Cf. 3-hop의 경우 query당 7번

[2025][ACL][LMP] Digest the Knowledge: Large Language Models empowered Message Passing for Knowledge Graph Question Answering

5. Limitation

- Multi-hop QA task의 경우 LLM 사용빈도가 굉장히 높은 편이며, 이에 크게 의존하고 있는 경우가 많음
- Question Answering task 자체가dataset 내 모든 query에 대해 분석을 진행하고 답을 생성하는 파이프라인을 가지고 있기에 상대적으로 비용과 시간이 많이 투자됨



CONTENTS

GflowNet - KGQA

[2025][SIGIR][FD-PORT]

Flow-guided Direct Preference Optimization for Knowledge Graph Reasoning with Trees

Tiesunlong Shen
Yunnan University, Kunming, China
Nanyang Technological University, Singapore, Singapore
tensorshen@mail.ynu.edu.cn

Rui Mao*
Nanyang Technological University
Singapore, Singapore
rui.mao@ntu.edu.sg

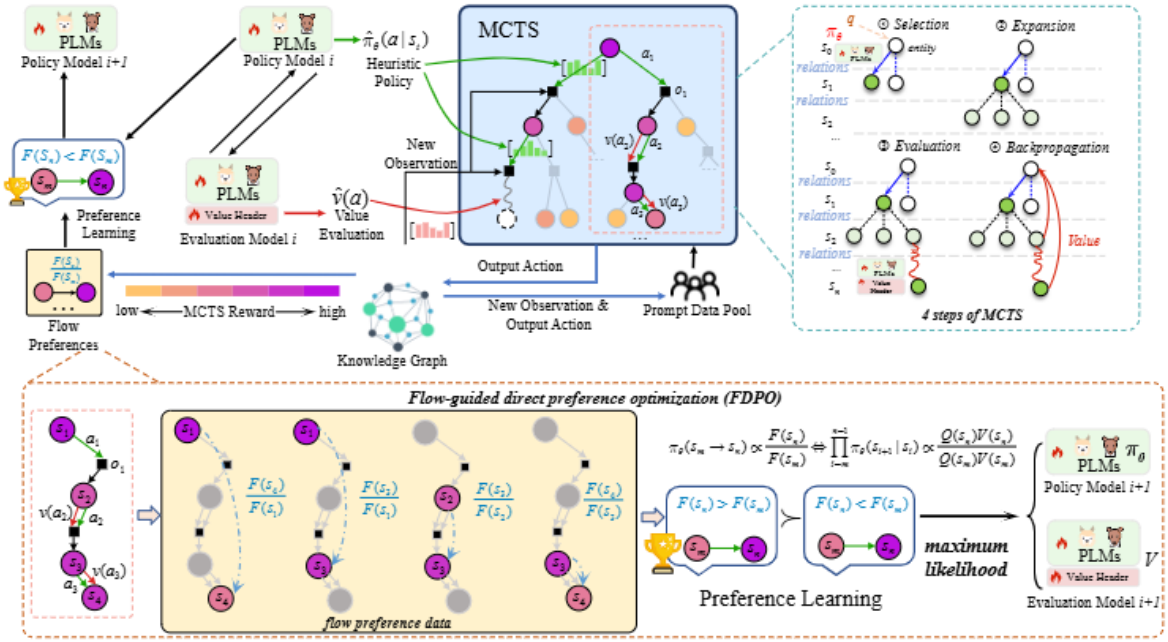
Jin Wang
Yunnan University
Kunming, China
wangjin@ynu.edu.cn

Xuejie Zhang
Yunnan University
Kunming, China
xjzhang@ynu.edu.cn

Erik Cambria
Nanyang Technological University
Singapore, Singapore
cambria@ntu.edu.sg

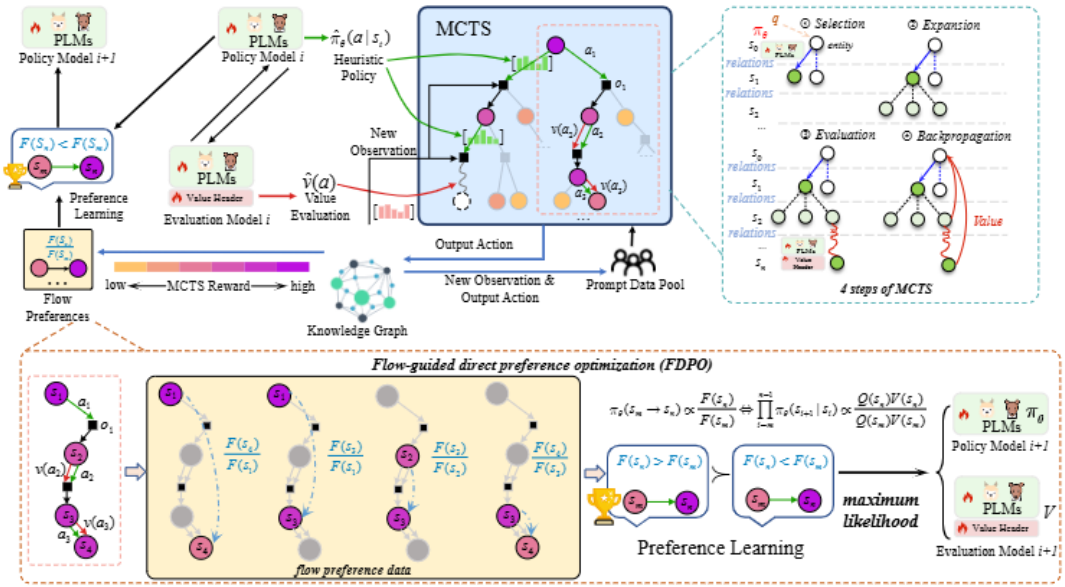
[2025][SIGIR][FD-PORT]

- MCTS로 KG 추론 트리를 탐색하고, Q·V로 만든 보상을 sub trajectory flow-balance FDPO로 선호학습해 KGQA 정확도(및 경로 일관성 및 Quality)를 높이는 모델
  - ✓ 기존 방식 한계
    - 전체 reasoning path를 하나의 시퀀스로 보고 **최종 answer만 비교하여 초반 error가 있다면 전파됨**
    - terminal state에만 reward를 부여하여 **중간 노드나 중간 결정의 가치를 학습 신호로 사용하지 않아** 특정 reasoning 패턴에만 집중하는 **mode collapse 문제 존재**
    - 따라서 좋고 다양한 path/subgraph들이 샘플링되도록 분포를 학습해보자 !



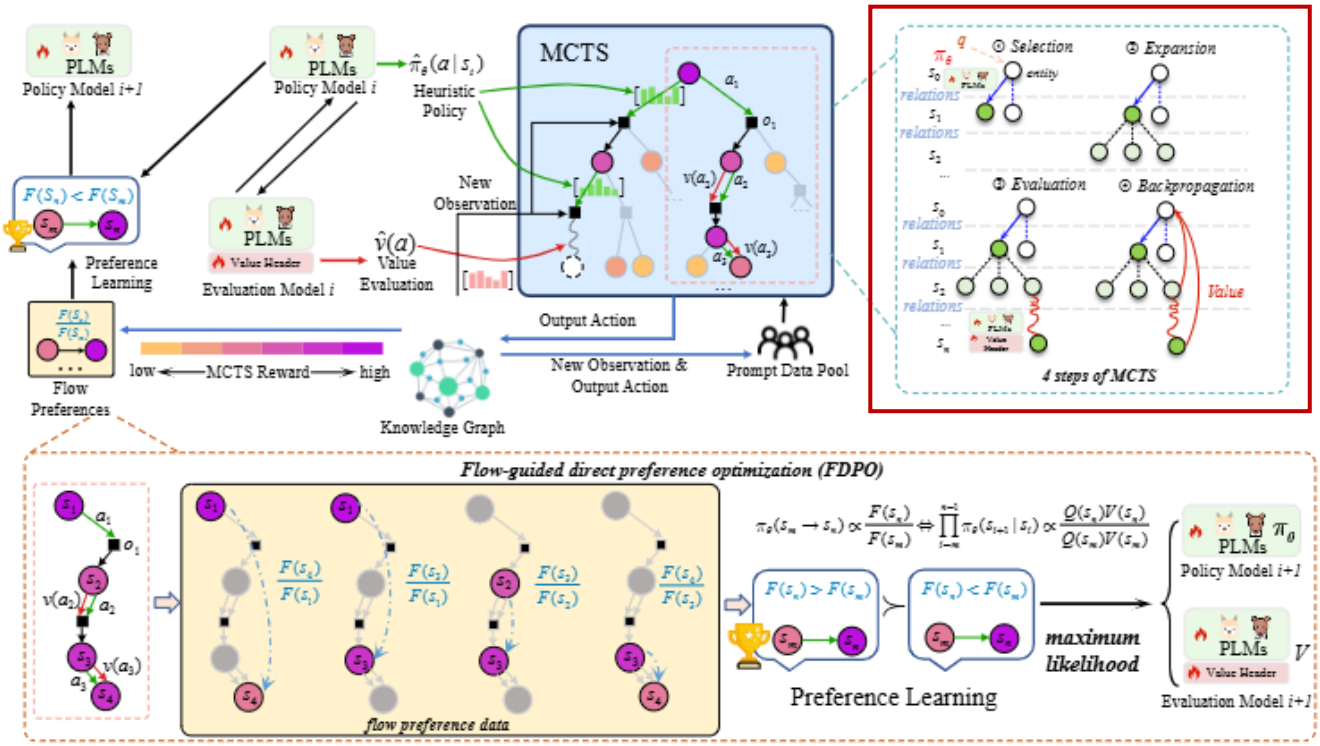
[2025][SIGIR][FD-PORT]

- MCTS로 KG 추론 트리를 탐색하고, Q·V로 만든 보상을 sub trajectory flow-balance FDPO로 선호학습해 KGQA 정확도(및 경로 일관성 및 Quality)를 높이는 모델
  - ✓ Policy model
    - Action(relation sampling)으로 Graph Expansion
  - ✓ Evaluator model(PLM)
    - 언어모델 기준 얼마나 좋은 장소(state)인지 판단하는 Vφ(s) 값을 출력
  - ✓ MCTS(Monte Carlo Tree Search)
    - 시뮬레이션을 통해 해당 장소(state)가 얼마나 가치있는지를 나타내는 Q(s) 출력



[2025][SIGIR][FD-PORT]

- MCTS는 4단계 Loop 반복
  (**Selection** -> Expansion -> Evaluation -> Backpropagation)
  - ✓ **Selection**
    - Topic entity에서 어디로 확장할지 하나의 action을 샘플링한다.



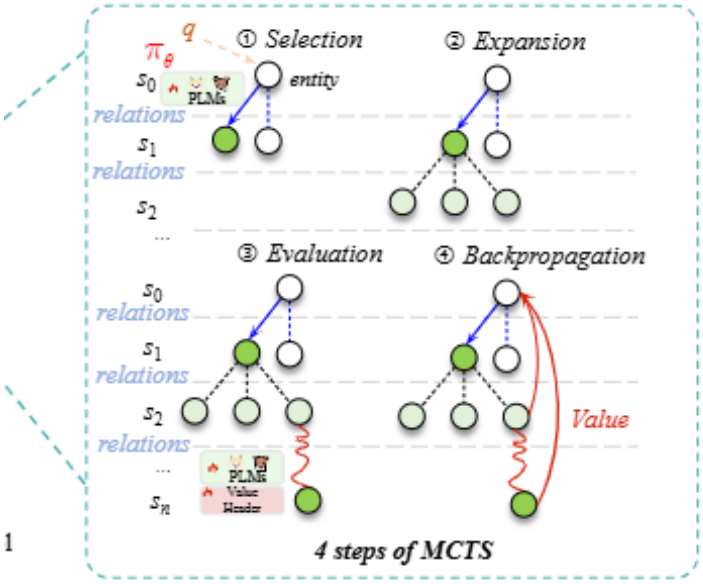
Topic ->

Policy model

Topic -> rel1 -> Leaf

[2025][SIGIR][FD-PORT]

- MCTS로 KG 추론 트리를 탐색하고, Q·V로 만든 보상을 sub trajectory flow-balance FDPO로 선호학습해 KGQA 정확도(및 경로 일관성 및 Quality)를 높이는 모델
  - ✓ Action 샘플링 확률(다양한 경로를 탐색할 수 있도록)



Evaluator model

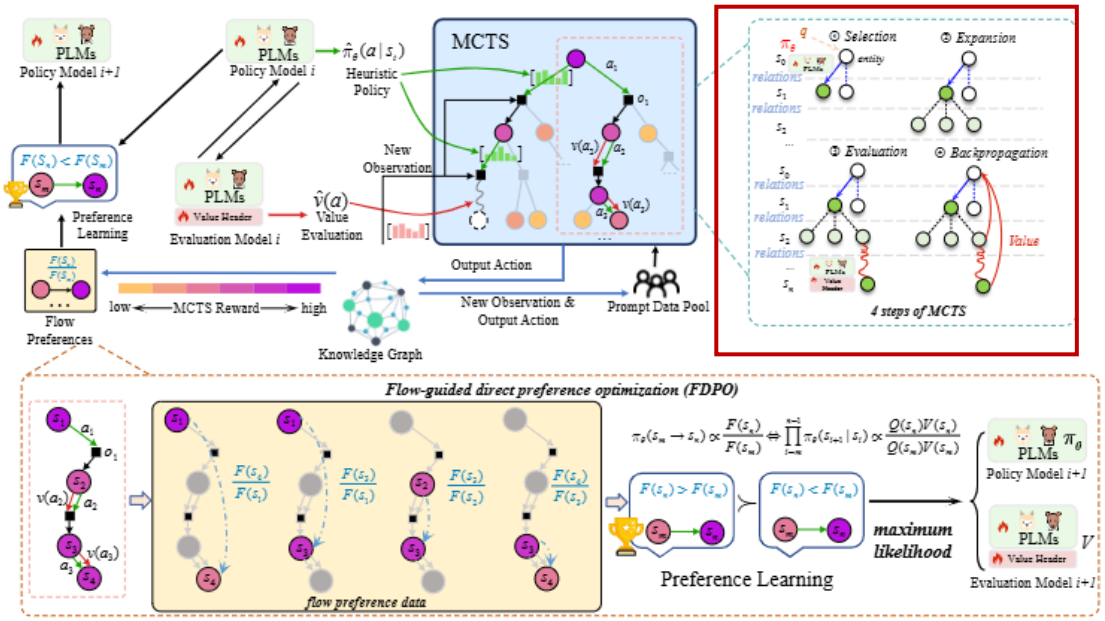
Policy model

$$(a \mid s_t) = (1 - p_s)\,\mathrm{Softmax}\left(Q(s_t, \cdot) + c_{puct}\,P(\cdot \mid s_t)\sqrt{\frac{N(s_t)}{1 + N(s_t, \cdot)}}\right) + p_s\,\mathcal{U}(C(s_t)),$$

Random Search

[2025][SIGIR][FD-PORT]

- MCTS는 4단계 Loop 반복
  (Selection -> **Expansion** -> Evaluation -> Backpropagation)
  - ✓ **Expansion**
    - 샘플링 후 Leaf node에서 연결 가능한 action 후보들 나열



Topic ->

Topic -> rel1 -> Leaf

Leaf -> rel2 -> child1

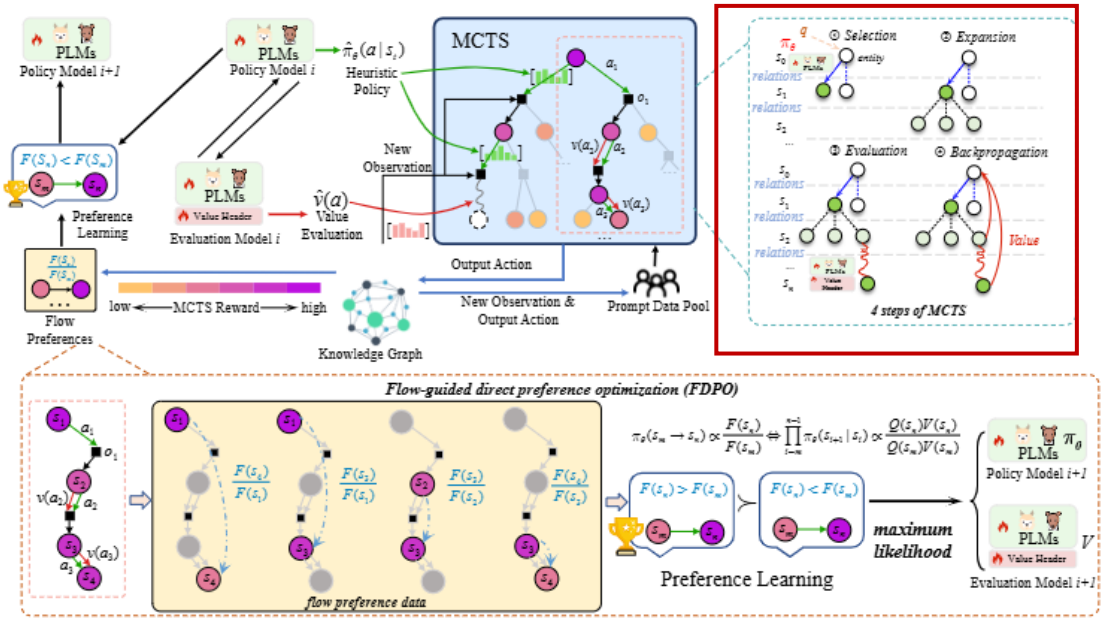
Leaf -> rel3 -> child2

Leaf -> rel4 -> child3

만약 child가 terminal이라면 Reward 1
아니면 1-step Rollout

[2025][SIGIR][FD-PORT]

- MCTS는 4단계 Loop 반복
  (Selection -> Expansion -> **Evaluation** -> Backpropagation)
  - ✓ **Evaluation**
    - 1 step rollout 실행하여 terminal이라면



Topic ->

Topic -> rel1 -> Leaf

Leaf -> rel2 -> child1

Leaf -> rel3 -> child2

Leaf -> rel4 -> child3

Rollout (시뮬)

Child3 -> rel5 -> node5

각 Child 1,2,3에 대해 가능한 모든 action 중 한 개가 sampling, Node 5의 terminal 여부에 따라 V(s) = 1 or 0 부여

[2025][SIGIR][FD-PORT]

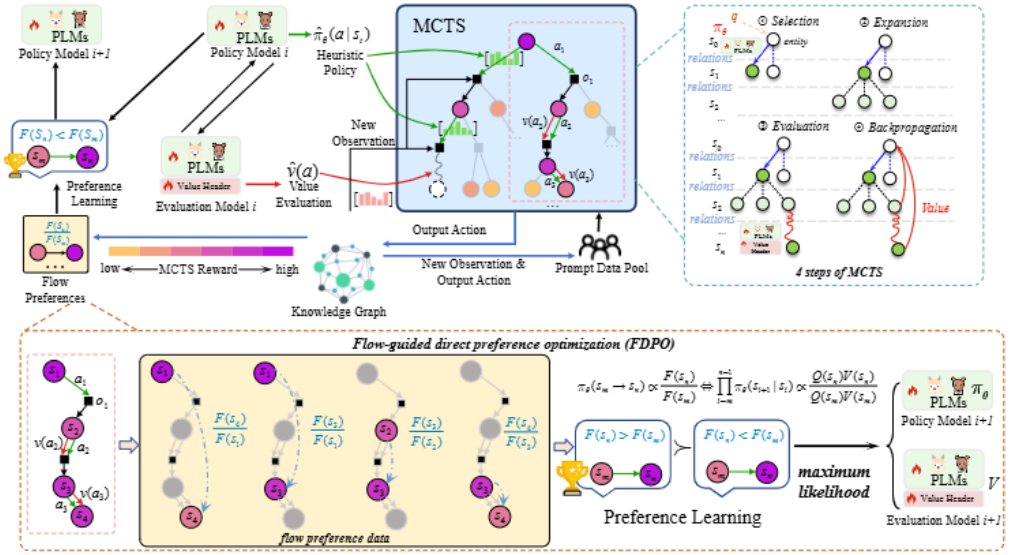
- MCTS는 4단계 Loop 반복
  (Selection -> Expansion -> Evaluation -> **Backpropagation**)
  - ✓ **Backpropagation**
    - MCTS를 통해 state별 Q(s,a)(MCTS)와 V(s)(evaluator)값을 얻음
    - **R(s)==F(s)==Q(s)Vϕ(s)** 값을 가지고 FDPO loss 기반 Evaluation과 Policy model을 학습하여
      더 좋은 방향으로 path를 탐색할 수 있도록 업데이트되고, 이후 다시 한번 MCTS를 돌리면 이전보다 더 좋은 Q값을 도출해냄.

$$Q(s',a') \leftarrow Q(s',a') + \frac{1}{N(s',a')}(V(s) - Q(s',a')).$$

$$\mathcal{L}_{\text{FDPO}} = \mathcal{L}_{\text{policy}} + \lambda \mathcal{L}_{\text{evaluate}},$$

$$\mathcal{L}_{policy}(\hat{\pi}) = \sum_{\tau \in \mathcal{D}} \sum_{m<n} \left(\log \frac{Q(s_m)\, V_\phi(s_n)}{Q(s_n)\, V_\phi(s_m)} - \sum_{i=m}^{n-1} \log \hat{\pi}(s_{i+1} \mid s_i) \right)^2.$$

$$\mathcal{L}_{\text{evaluate}} = \max(0, \gamma - (V_\phi(s_t, a_w) - V_\phi(s_t, a_l))),$$



Thank you for listening

Jinho Choi (jhc12931293@gmail.com)

Graph & Language Intelligence Laboratory
Konkuk University

2026.01.20