Hyper-RAG: combating LLM hallucinations using hypergraph-driven retrieval-augmented generation

Yifan Feng, Hao Hu, Shihui Ying, et al.

Nature Communications, 2026

Deep Shark Lab 인턴
양현민

Contents

RAG

LLM의 한계

1. 정보의 신뢰성

- Hallucination
- 지식의 접근 및 활용의 한계
- 답변의 출처 확인의 어려움

2. 지식 관리/업데이트

- 기존 지식 수정의 어려움
- 새로운 지식 추가 어려움

RAG의 효과

1. 정보의 신뢰성

- Hallucination 완화
- 외부 지식 활용
- 검색 근거를 답변에 활용 및 출처 연결 가능

2. 지식 관리/업데이트

- 새로운 지식 추가 용이
- 기존 지식 수정 및 업데이트 용이

Graph-RAG

RAG의 한계

1. 청크중심 검색

- 질문과 유사한 Top-k 청크 검색
- 여러 청크에 분산된 정보 연결이 어려움
- Entity 사이의 구조적 관계 활용이 제한적

2. 검색 범위의 한계

- 구체적인 정보와 전체적인 맥락을 함께 검색하기 어려움
- 복잡한 관계 질문에서 필요한 정보가 누락될 수 있음

Graph-RAG(Light-Rag)의 효과

1. 지식 그래프 기반 표현

- 문서에서 Entity와 Relation 추출
- Entity와 Relation를 그래프 형태로 저장
- 연결된 정보를 구조적으로 탐색

2. Dual-level Retrieval

- Local Query: 구체적인 개체와 세부 관계 검색
- Global Query: 주제와 전체적인 관계 정보 검색

Hyper-RAG

Graph-RAG의 한계

1. Low-Order 관계 중심

- 복잡한 상호작용 표현 한계

2. 구조적 정보 활용 제한

- 복잡한 Query 구조적 정보 부족

Hyper-RAG의 효과

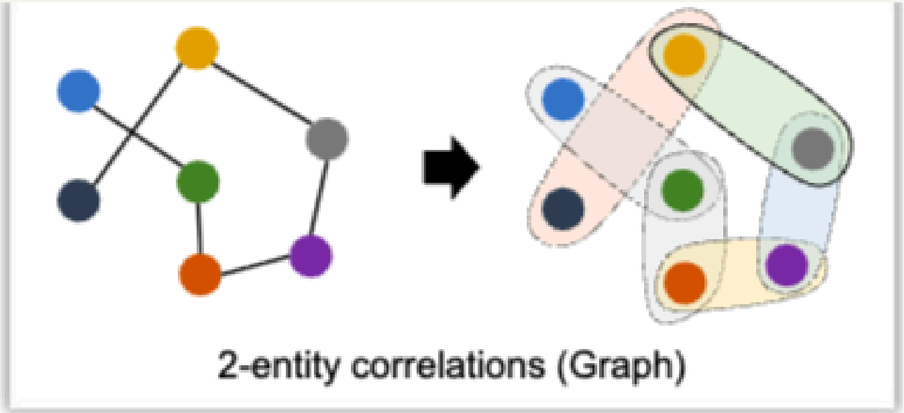
1. High-Order 표현

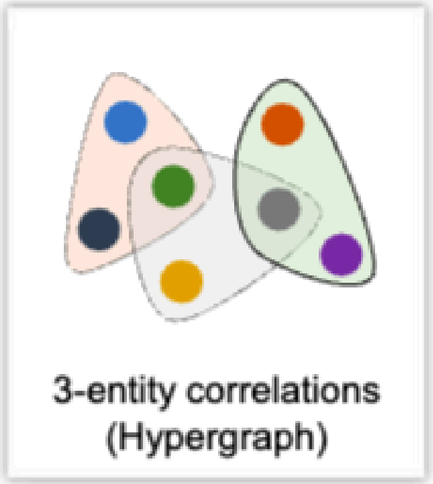
- 여러 Entity를 하나의 관계로 연결
- 복잡한 상호작용 직접 표현

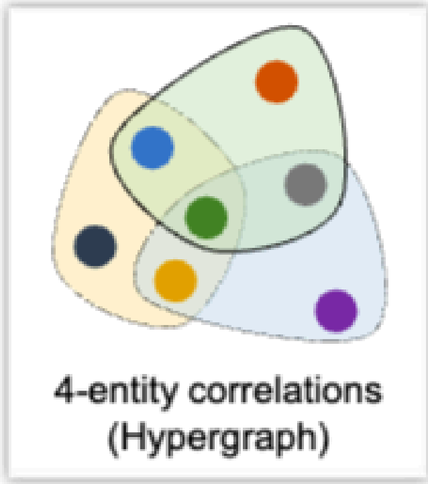
2. 지식 활용 향상

- Low-Order + High-Order 관계 활용
- 복잡한 Query에 더 풍부한 정보 제공

HyperGraph







HyperEdge

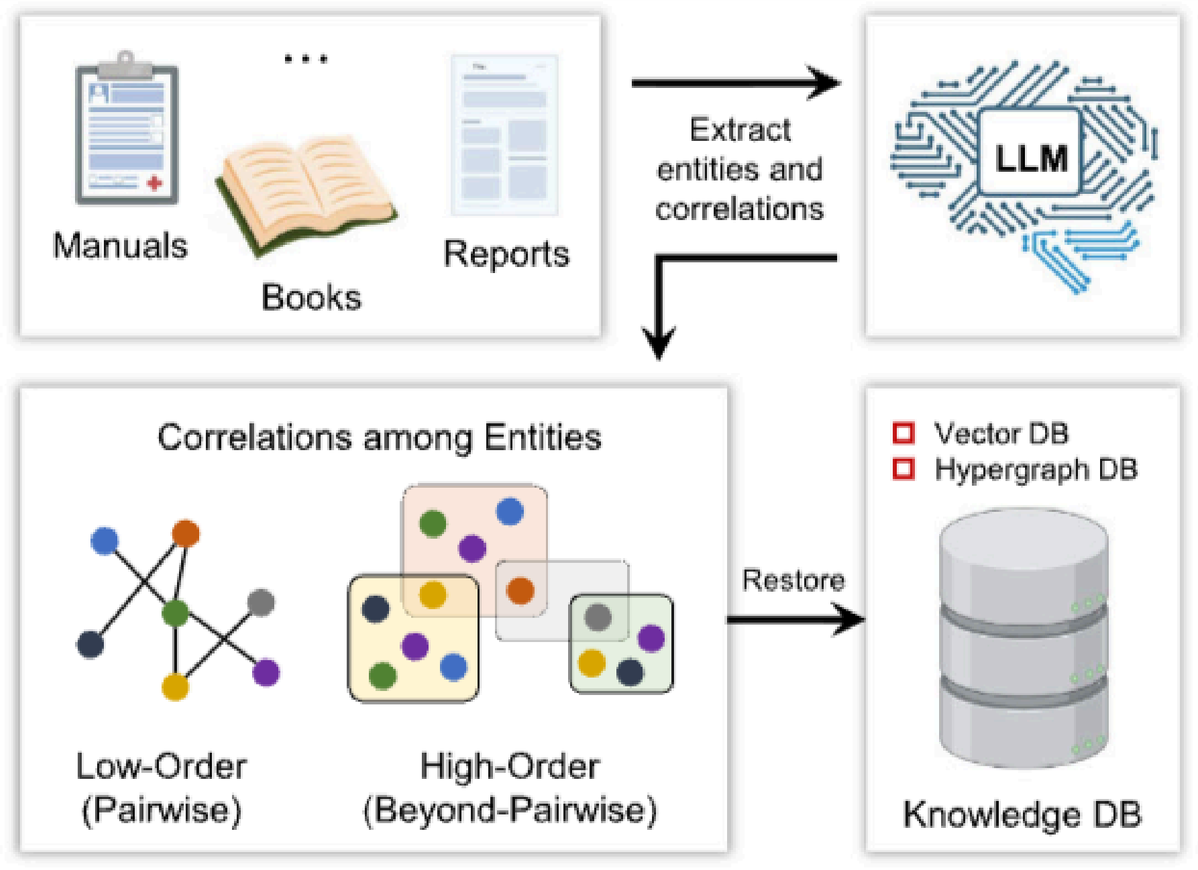
- 임의 개수의 Node를 동시에 연결
- 복수 Entity 간 High-Order 상관관계 표현 가능

Graph : $e = \{v_1, v_2\}$

HyperGraph : $e = \{v_1, v_2, \cdots, v_k\}$

Architecture

지식 구축



1. 데이터 입력

- 데이터 전처리(텍스트만 추출)
- Chunk 분할(1200토큰 100토큰 Overlap)

$$D = \{ D_1, D_2, \cdots, D_n \}$$

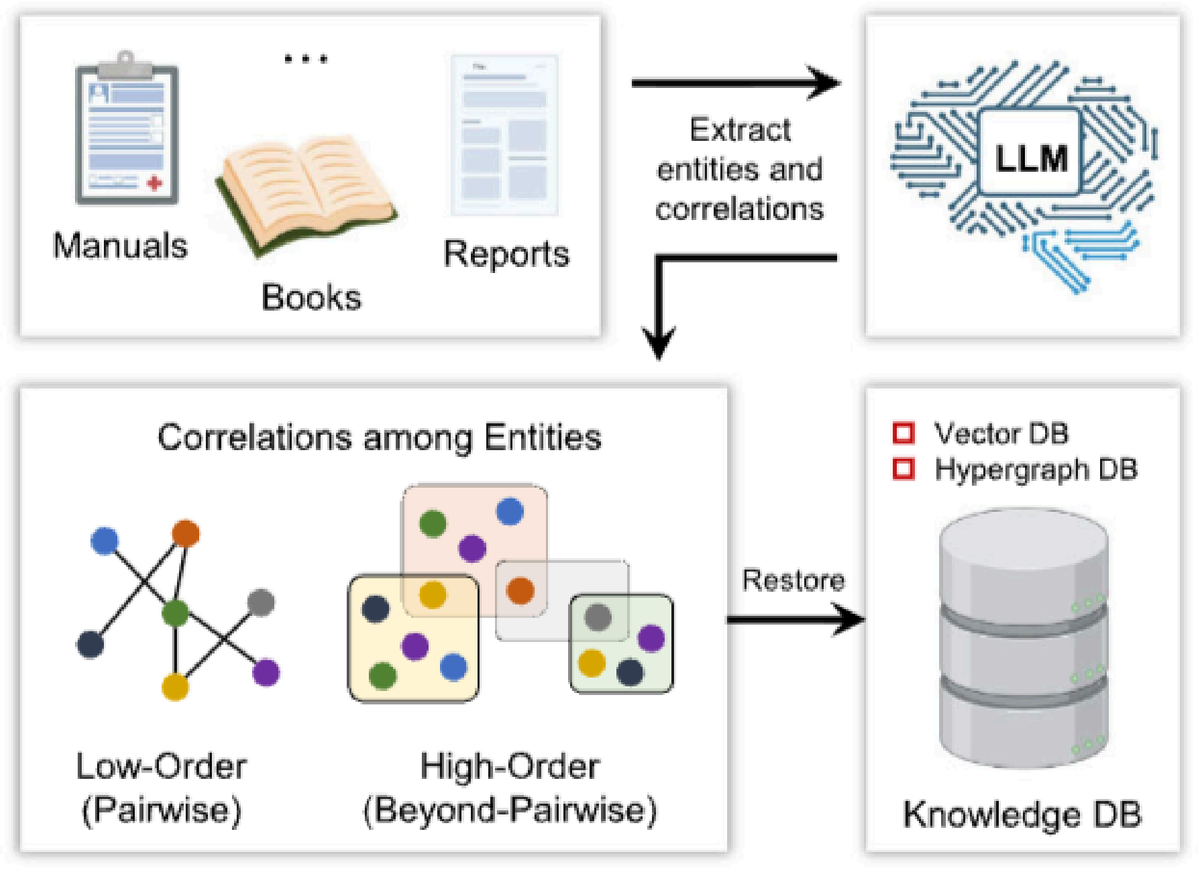
2. Entity 추출

- 각 Chunk에서 주요 Entity추출
- Entity 설명 생성

$$K_v = LLM(P_{ext\_entity}(D_i))$$

Architecture

지식 구축



3. 상관관계 추출

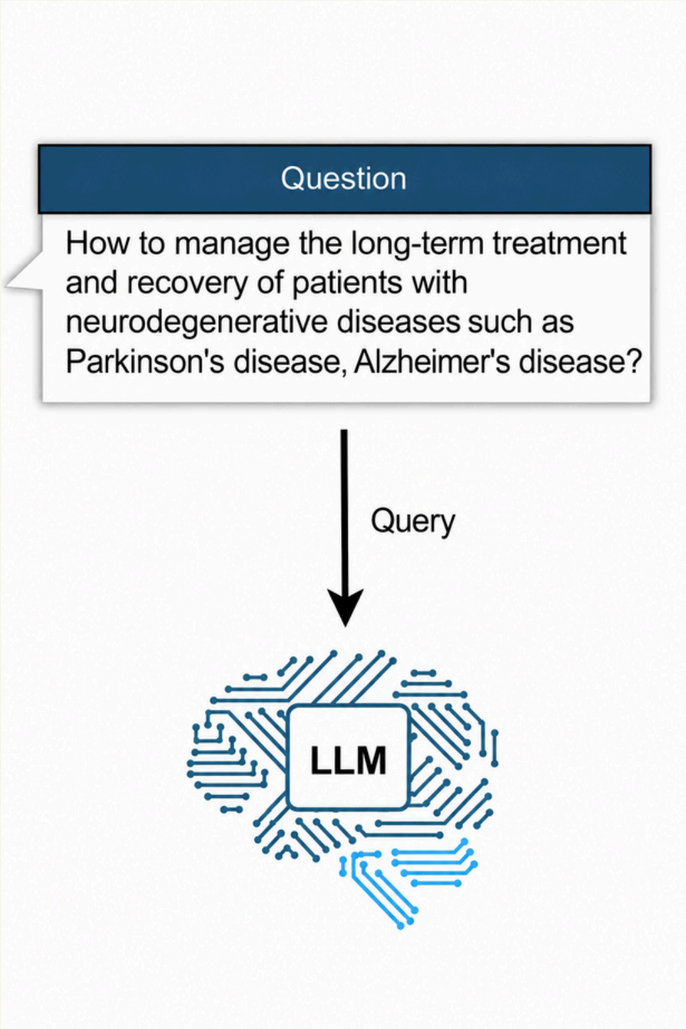
- Low-Order : 저차 상관관계
- High-Order : 고차 상관관계
- 관계 설명 생성

$$K_{\text{low\_}e} = \text{LLM}(P_{\text{ext\_low}}(D_i, K_v))$$
$$K_{\text{high\_}e} = \text{LLM}(P_{\text{ext\_high}}(D_i, K_v))$$

4. Knowledge DB 저장

- Entity / Relation 임베딩 → Vector DB
- Vertex / Hyperedge 구조 → HyperGraph DB

Architecture



지식 검색 및 LLM증강

1. Query 입력

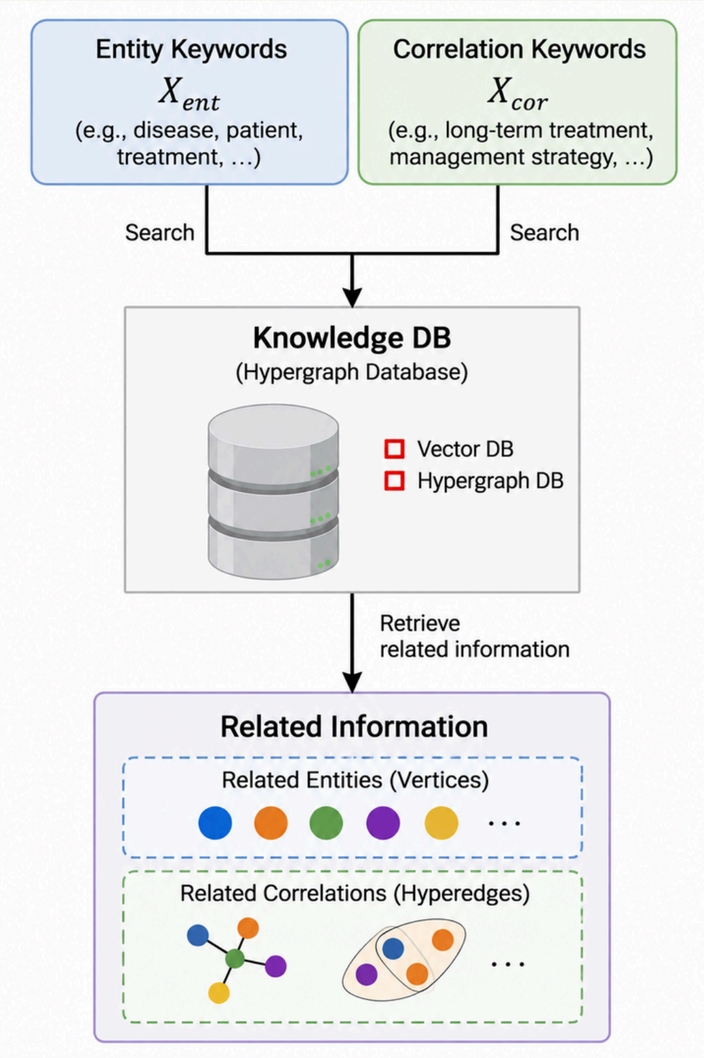
- 사용자 질문 q 입력
- 질문을 키워드 추출용 LLM에 전달

$$X_{\text{ent}}, X_{\text{cor}} = \text{LLM}(P_{\text{ext\_key}}(q))$$

2. 키워드 추출

- Entity 키워드 → X_{ent}
- 상관관계 키워드 → X_{cor}

Architecture



지식 검색 및 LLM증강

3. Entity 키워드 검색

- X_{ent}로 Vertex(entity) Vector DB 검색
- 질문과 유사한 Vertex V_{rel} 획득

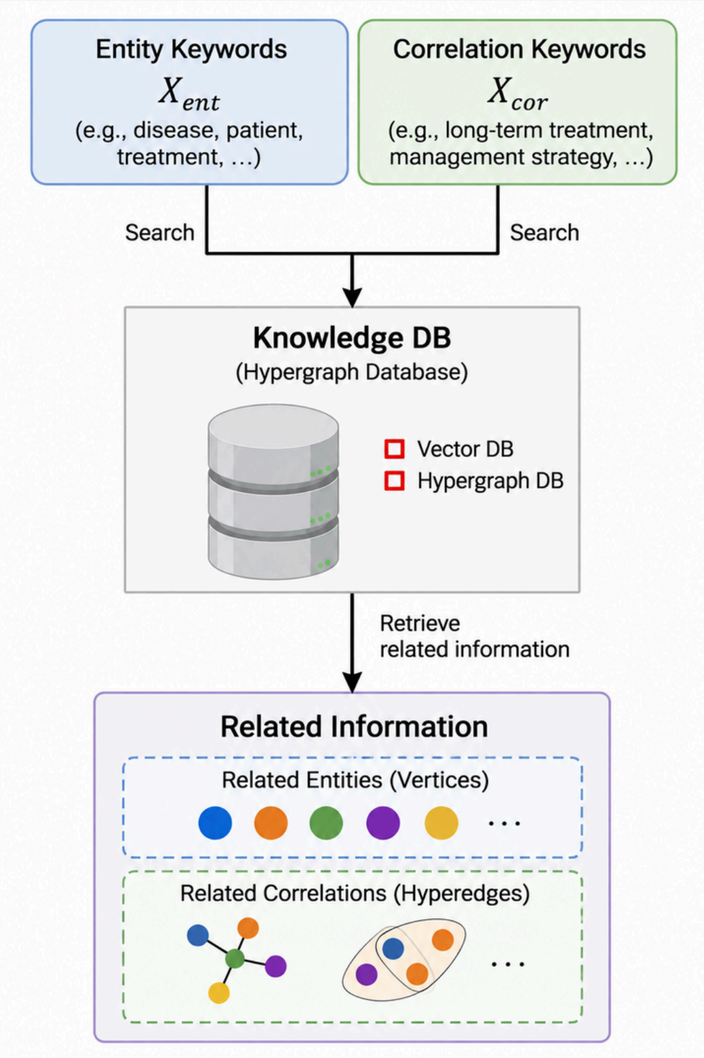
$$V_{\text{rel}} = \{\psi_{\text{ret}}(x_i, V) \mid x_i \in X_{\text{ent}}\}$$

4. 구조적 확산

- 검색된 Vertex와 연결된 Hyperedge 탐색
- 보충 정보 E_{more} 획득

$$E_{\text{more}} = \{e \mid v \in e,\ v \in V_{\text{rel}}\}$$

Architecture



지식 검색 및 LLM증강

5. 상관관계 키워드 검색

- X_{cor}로 Hyperedge Vector DB 검색
- 질문과 관련된 Hyperedge E_{rel} 획득

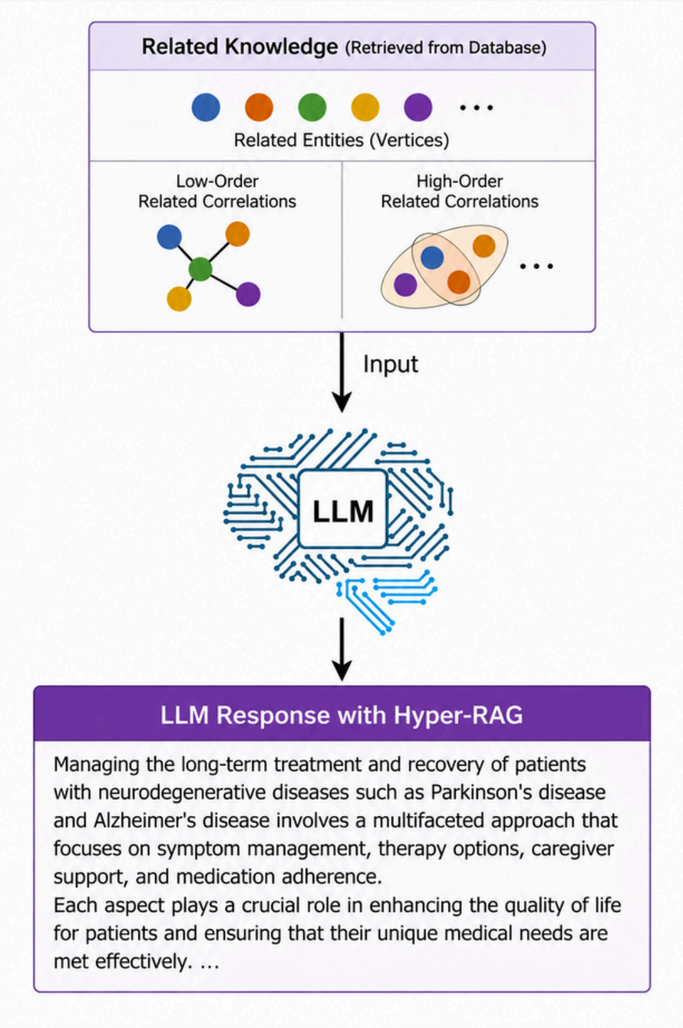
$$E_{\text{rel}} = \{\psi_{\text{ret}}(x_i, E) \mid x_i \in X_{\text{cor}}\}$$

6. 1-Step Diffusion

- 검색된 Hyperedge에 포함된 Vertex 탐색
- 보충 정보 V_{more} 획득

$$V_{\text{more}} = \{v \mid v \in e,\ e \in E_{\text{rel}}\}$$

Architecture



지식 검색 및 LLM증강

1. 정보 선택

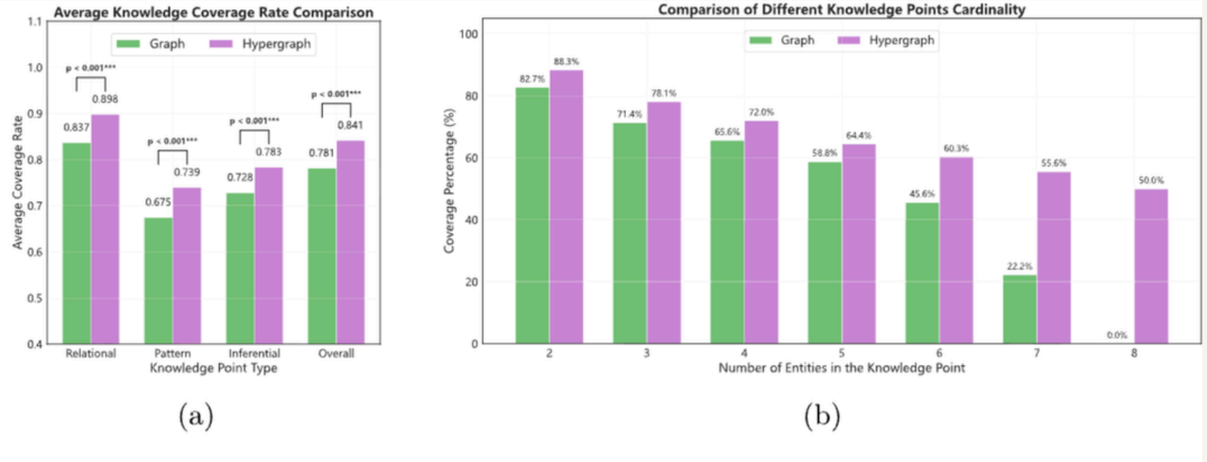
- 검색 결과를 관련도 순으로 정렬
- 입력 길이에 맞게 상위 정보만 선택

2. 원본 Chunk 통합

- 관련 Vertex / Hyperedge의 원본 Chunk 통합
- 최종 LLM입력
  = 사용자 입력 + 검색된 지식 + 원본 청크

Experiment

Knowledge Capture Experiment



Knowledge Point(KP)

- Semantic Chunk에서 추출한 핵심 지식 단위
- Gpt-5 활용해 추출
- Relational / Pattern / Inferential

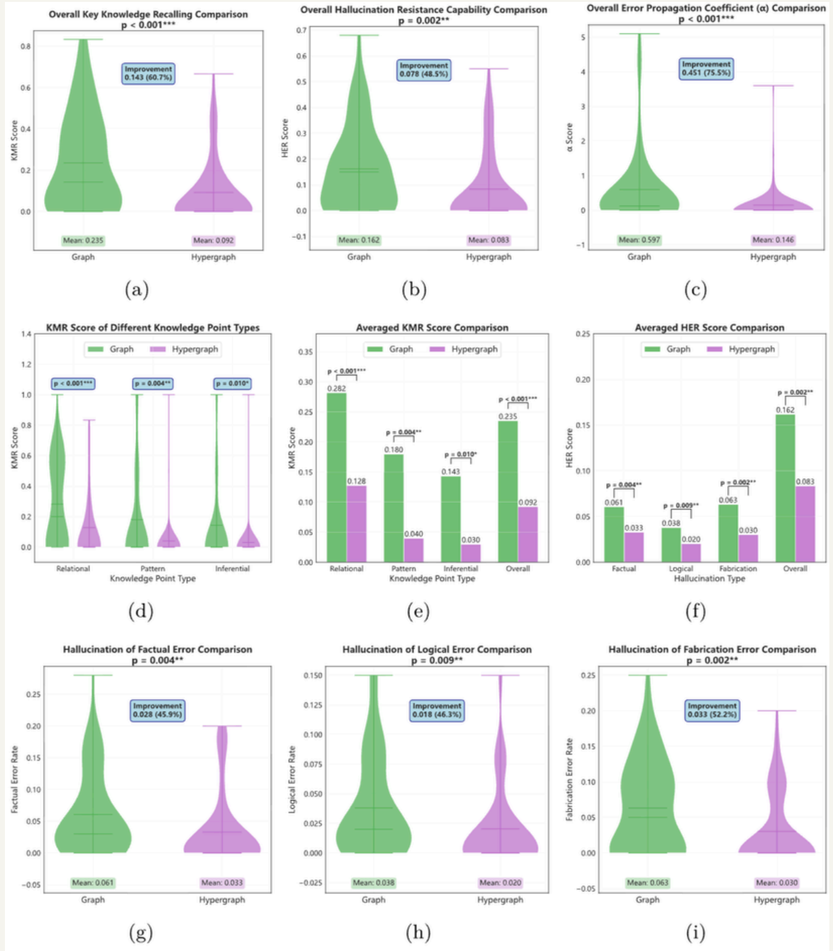
예시

약물A는 수용체 B를 억제한다. 이로 인해 신호C가 감소하고, 결국 증상 D가 완화된다.

- KP1 : 약물 A가 수용체 B를 억제한다
- KP2 : 수용체 B의 억제로 신호 C가 감소한다
- KP3 : 신호 C의 감소로 증상 D가 완화된다
- KP4 : 약물 A → 수용체B → 신호C → 증상D

Experiment

Hallucination Mitigation Expeiment



KMR (Key Knowledge Missing Rate)

- 정답 기준 핵심 지식과 검색 결과 비교
- all-MiniLM-L6-v2 기반 의미적 유사도 측정
- 검색 단계의 핵심 지식 누락 정도 측정

HER (Hallucination Error Rate)

- 최종 LLM 응답의 hallucination 오류 측정
- 사실/논리/날조 오류로 구분
- 최종 응답의 사실,논리,날조,오류를 평가하여 정규화한 환각 오류 지표

α (Error Propagation Coefficient)

- 검색 오류의 환각 전파 정도 측정
- α = HER / KMR
- 0.59 → 0.14, 즉 76% 감소

Experiment

Ablation Experiment

Method	$\mathcal{D}$	$\mathcal{E}_{low}$	$\mathcal{E}_{high}$	Comp.	Dive.	Empo.	Logi.	Read.	Overall	Rank
LLM	✗	✗	✗	77.00	58.60	67.26	82.80	82.60	73.65	8
-	✗	✓	✗	83.40	74.96	74.72	86.06	86.24	81.08	6
-	✗	✗	✓	84.40	75.00	75.68	86.46	86.22	81.55	5
-	✗	✓	✓	85.90	78.34	77.14	87.02	86.70	83.02	3
RAG	✓	✗	✗	81.90	69.24	74.00	85.06	82.62	78.56	7
LightRAG	✓	✓	✗	85.80	77.20	76.56	86.58	86.84	82.60	4
-	✓	✗	✓	88.26	78.80	77.52	87.34	87.04	83.79	2
Hyper-RAG	✓	✓	✓	**88.40**	**79.60**	**77.84**	**87.76**	**87.22**	**84.16**	1

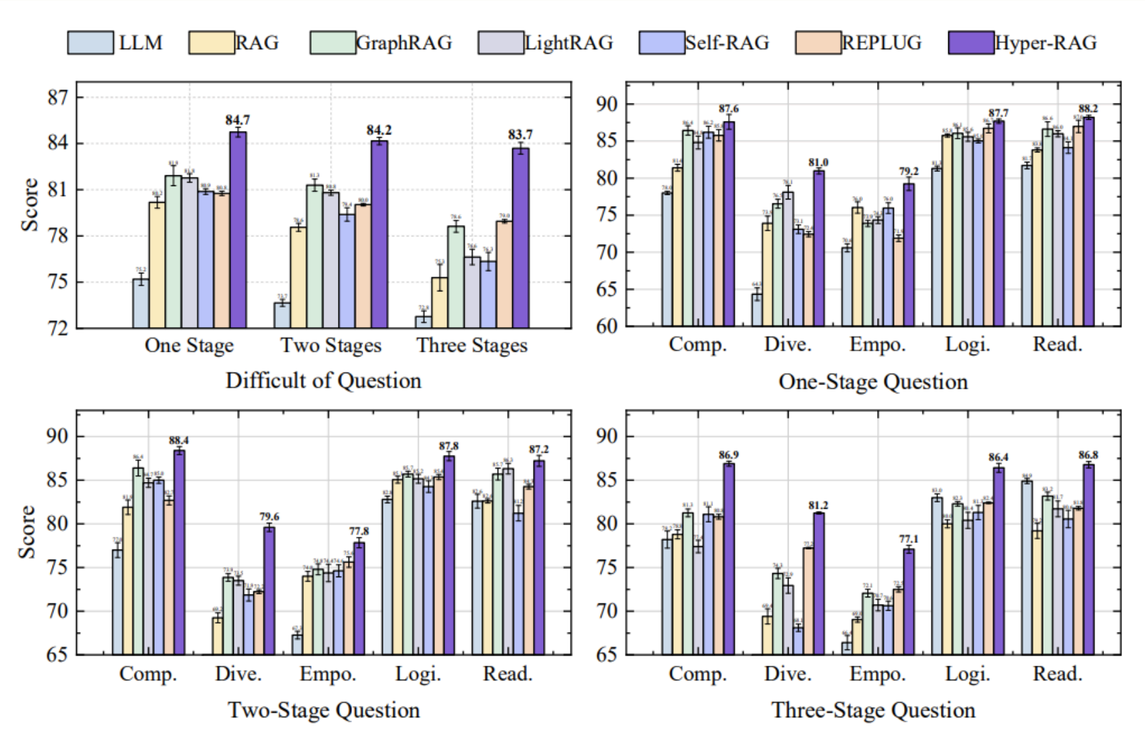
1. 구조화된 지식 표현의 효과

2. 원본 Chunk의 보완 효과

3. High-order Relation의 높은 정보 효율

Experiment

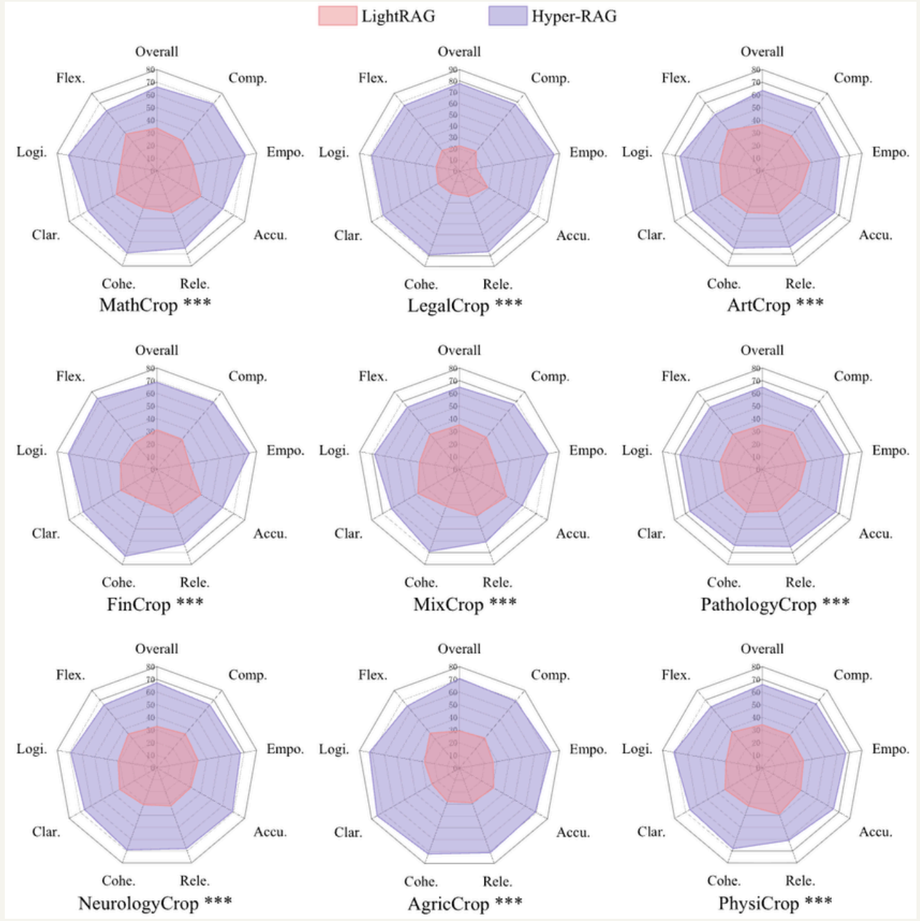
Query Complexity Evaluation



- 질문 복잡도가 증가할수록 direct LLM 대비 상대적 성능 하락 안정적
- 84.7 → 84.2 → 83.7

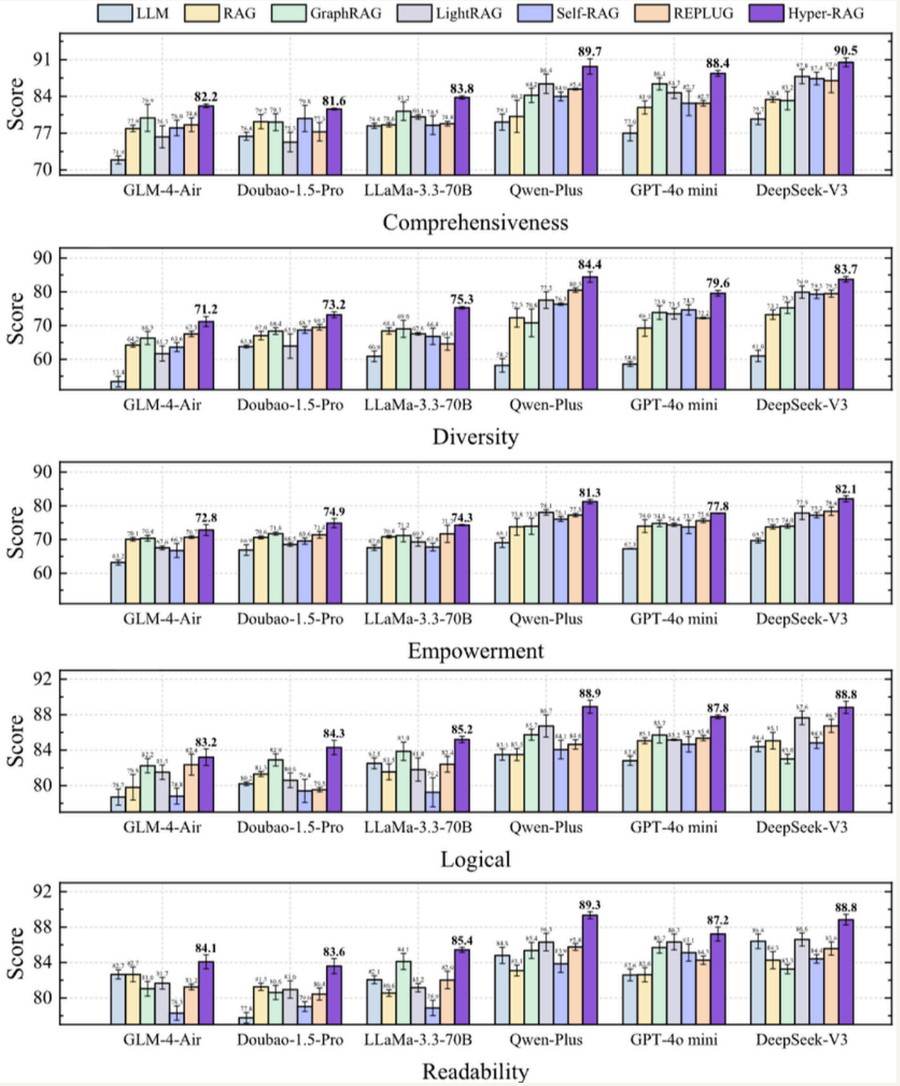
Experiment

Domain-wise Evaluation



- 다양한 도메인에서도 일관된 성능 향상 확인
- Hyper-RAG가 LightRAG 대비 평균 35.5% 향상

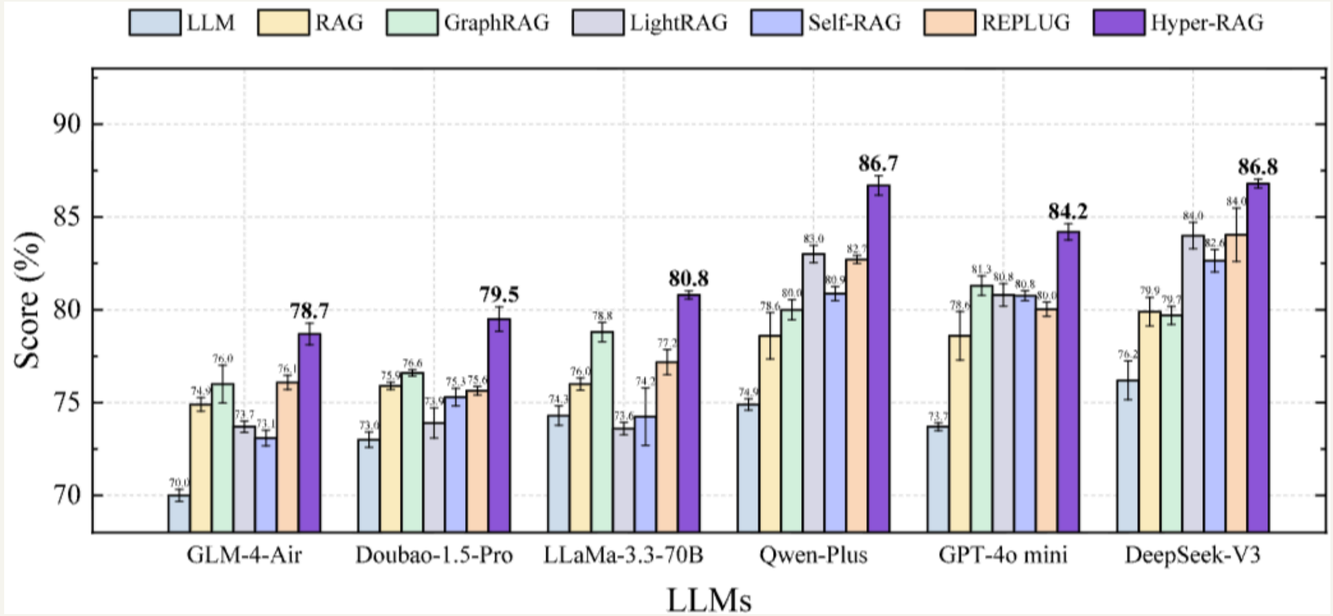
Experiment



- Comprehensiveness: 포괄성
- Diversity: 다양성
- Empowerment: 신뢰성
- Logicality: 일관성
- Readability: 가독성

- Light Rag대비 평균 성능 35.5% 향상

Experiment



- direct LLM 대비 평균 12.3% 향상

Limitations

1. Knowledge Base 구축 비용

- High-order Relation 추가 추출 필요
- Knowledge Construction 단계 증가

2. Cross-Chunk Relation 추출 한계

- 서로 다른 Chunk에 걸친 관계 직접 추출 어려움
- 후처리 기반 관계 병합 필요

Future Work

- 청크 간 관계 융합
- 문서 간 관계 모델링
- 지식 표현의 자동화 및 확장성 개선

Thank You