

계층적 데이터 처리를 위한 GraphRAG 연구

2026.02.12

오정현
데이터 인텔리전스 연구실
석사 과정

Contents

1. Introduction

2. Methodology

3. Experiments

4. Conclusion

Background

Retrieval-Augmented Generation (RAG)

- 외부 지식을 활용하여 LLM의 **사실적 응답 능력 강화**
- Hallucination 및 wrong answer 문제 완화
- 의료, 물류, 공정 등 **다양한 산업 도메인**에서 활용 확대
- **산업 도메인 데이터의 특성**
 - 문서, 개체, 사건 간 **계층적·관계적 구조**를 내포
 - 단순 텍스트 유사도만으로는 구조적 맥락 포착에 한계 존재

Text-based RAG의 한계

- 문서를 **chunk 단위로 분할** 후 임베딩·검색·생성
- 단순 질의응답에서는 높은 성능
- Long-context 처리 능력은 향상되었으나
 - 여전히 **context 외 관계성** 및
 - 다단계 추론(multi-hop reasoning)에는 취약

Problem Definition

Graph-based RAG의 등장

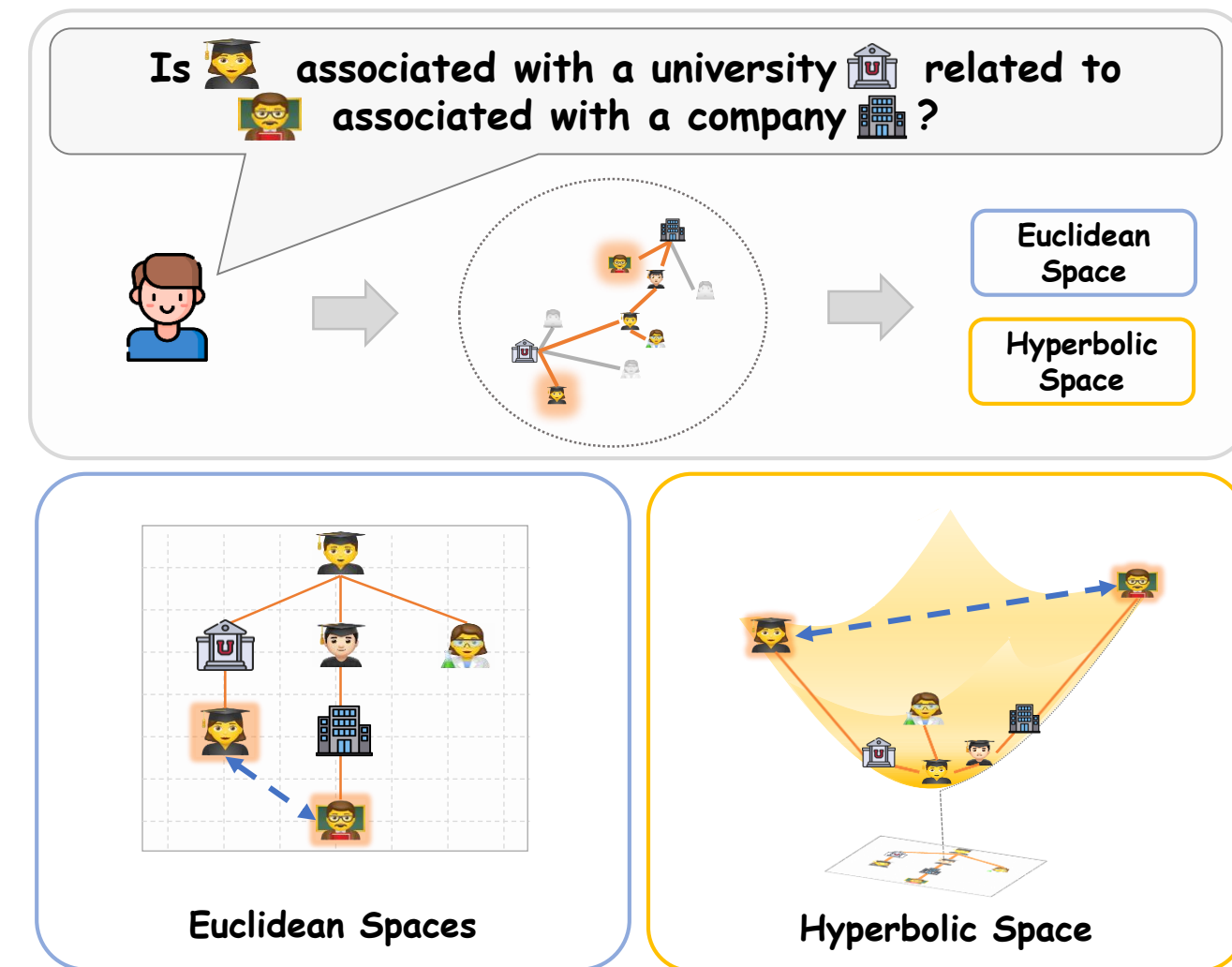
- 문서를 Knowledge Graph(KG)로 변환하여 검색·추론 수행
- Text-based RAG의 한계를 보완하며 **Multi-hop QA에서 강점**
- 그래프 형태
 - Chunk-level graph
 - Triplet (h, r, t) 기반 그래프
- Node·Edge를 자연어로 변환하여 **LLM 추론력 강화**
- 주요 기법
 - Clustering, Summarization
 - Hierarchical graph construction 등

GraphRAG의 한계

- 주로 **Euclidean space** 기반 임베딩 활용
- 그래프 내 **의미적 연결성** 표현에는 효과적
- 병렬적·다중 관계 표현에는 유리하나
- 계층적 깊이(hierarchical depth)를 충분히 반영하지 못함

Hyperbolic Space의 장점

- 연속적인 음의 곡률(negative curvature)을 가진 공간
- Tree-like 구조를 갖는 **계층적 그래프 표현에 적합**
- 노드 간 깊이 차이를 자연스럽게 거리로 인코딩



Proposed idea

Recent Hyperbolic RAG의 한계

- 하이퍼볼릭 공간을 **main embedding / retrieval 단계**에 직접 적용
- 모든 질의-데이터셋이 계층 구조를 갖는 것은 아님
- 여전히 의미 정보 중심이며, 구조 정보는 간접적으로만 작용
- 의미 기반 신호와 구조 기반 신호 간 **Semantic-Structural Misalignment**
- 구조 정보가 오히려 성능 저하를 유발하는 경우 존재

Hyper-Reranking

- **Semantic-Structural Misalignment**를 개선하는 GraphRAG 기반 Reranking Module
- Main retriever가 아닌 **plug-in 형태의 reranking 모듈**로 설계
- **Hyperbolic space**에서 구조적 유사도를 계산하고 semantic-structural correlation에 따른 **선택적으로 반영**
- 실험 결과, 데이터셋의 계층성 정도에 따라 성능 향상 폭이 달라짐을 확인

Contents

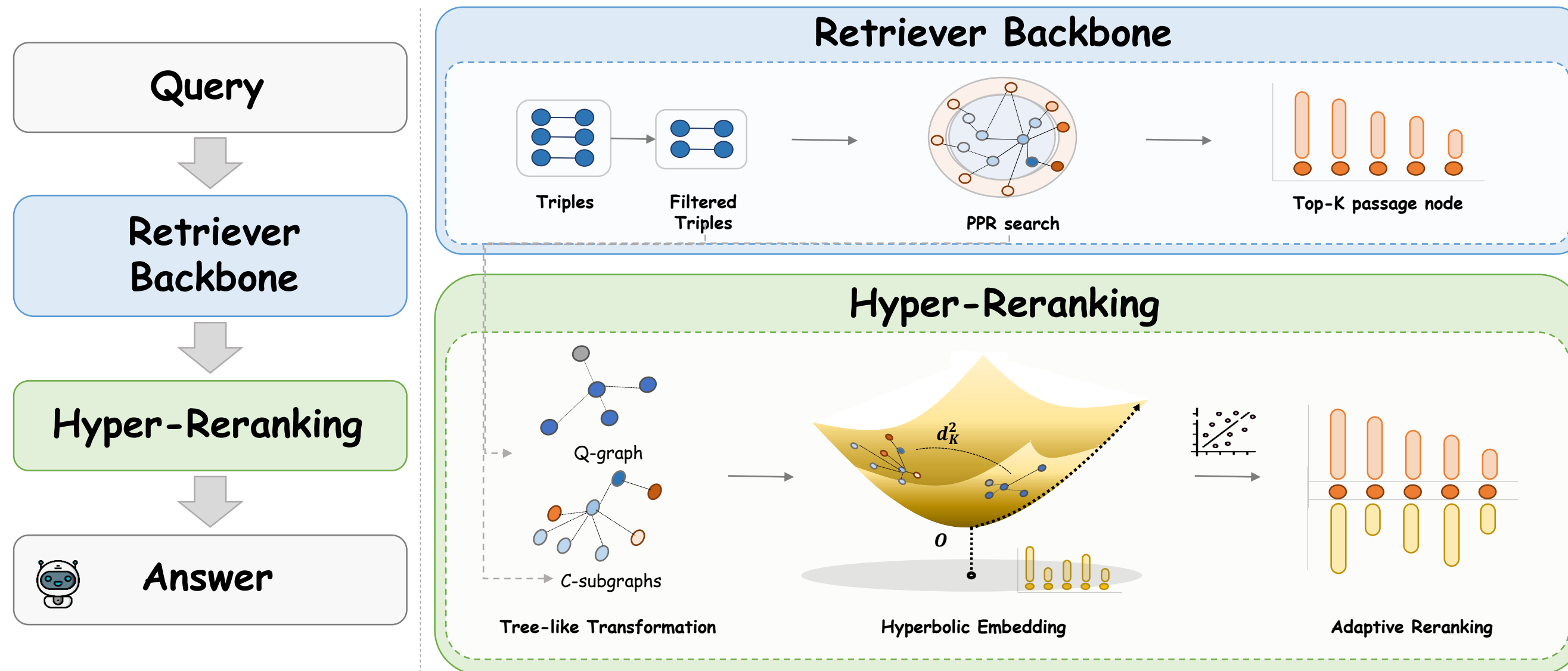
1. Introduction

2. Methodology

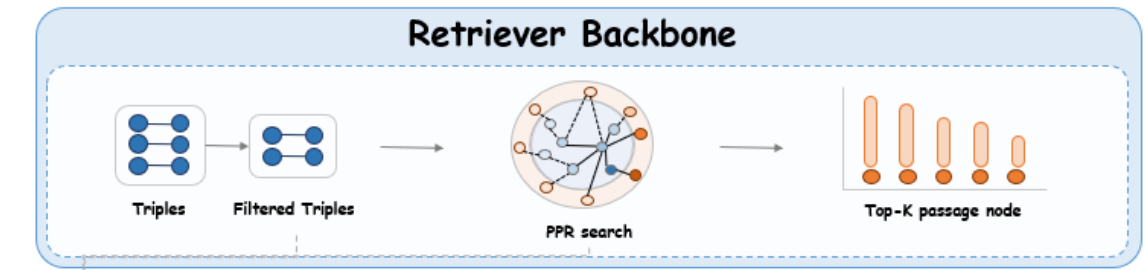
3. Experiments

4. Conclusion

Hyper-Reranking Overview



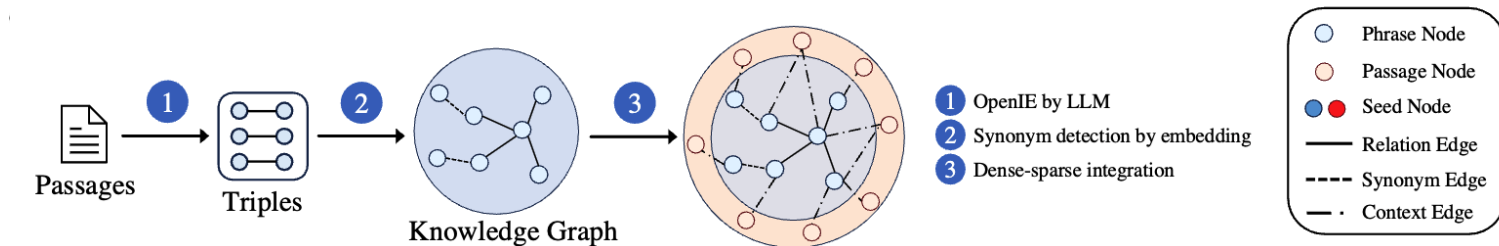
Retriever backbone



Retriever backbone 으로 HippoRAG2 사용

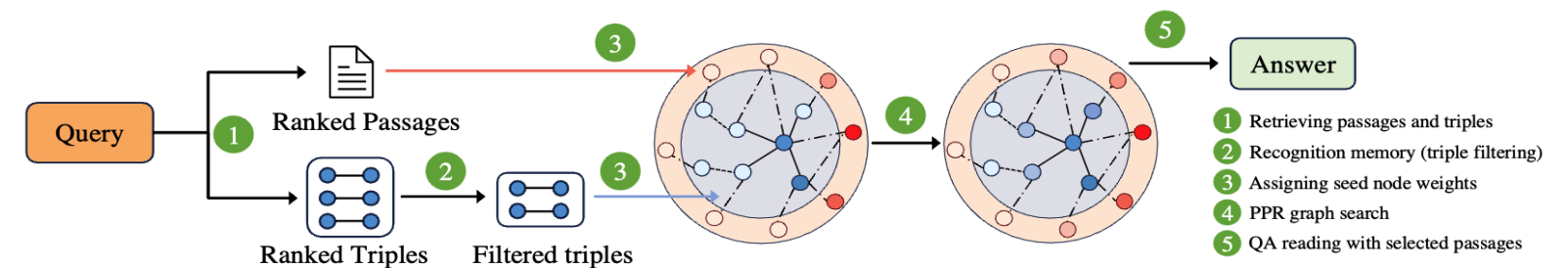
Indexing

- LLM을 활용해 passage로부터 triples (h, r, t) 추출
 - 추출된 triples로 초기 Knowledge Graph(KG) 구축
 - 각 entity가 포함된 passage를 노드로 추가
 - entity-passage 간 "contain" edge로 연결
 - 의미적으로 유사한 entity 간 "synonym" edge 연결
- 구조·의미 정보를 모두 포함한 최종 KG 완성

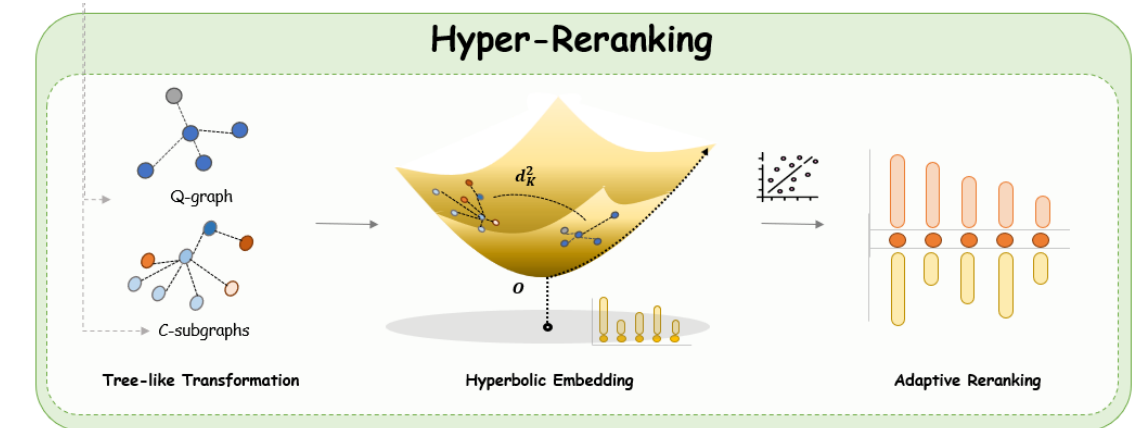


Retrieval

- Query로부터 관련 query triples 추출
- Query triples의 entity를 seed entities로 설정
- Personalized PageRank (PPR) 기반 그래프 탐색 수행
- 상위 점수 노드에 연결된 passage를 context로 선택
- 선택된 context를 활용해 QA 수행



Hyper-Reranking



Tree-like Transformation

- Q-graph : Query triples을 “question” 이라는 가상 노드로 tree-like graph 구축
- C-subgraphs : retriever backbone 선별한 상위 노드를 root node으로 PPR scores를 통해 tree-like graph 구축

Hyperbolic Embedding

- **Feature Fusion (h_v)**
 - 기존 semantic feature e_v
 - 그래프 기반 structural feature s_v
 - 정규화된 depth + radial distance
 - 의미·구조 정보를 결합한 노드 표현 생성
- **Hyperbolic Graph Embedding**
 - 극점 o 를 기준으로 Lorentz manifold에 임베딩
 - HGAT 기반 message passing으로 그래프 전반의 구조 정보 반영
 - 구조적 유사도를 고려한 reranking score 산출

$$h_v = [e_v || s_v]$$

$$x_v = \exp_o(Ah_v + b)$$

$$z_G = \arg \min_{z \in H_K^n} \sum_{v \in V_G} d_K^2(z, x_v^{(1)})$$

Adaptive Reranking

- Structural similarity :
- Semantic-Structural Alignment :
- Query-dependent Gating :
- Final Score :

$$S_{struct}(q, d) = \sigma(-\tau d_K^2(z_q, z_d))$$

$$\rho(q) = \rho(S_{sem}(q, d), S_{struct}(q, d))$$

$$g(q) = g_{min} + (g_{max} - g_{min}) \cdot \sigma(1 - \rho(q))$$

$$S_{final}(q, d) = \alpha S_{sem}(q, d) + (1 - \alpha)g(q)S_{struct}(q, d)$$

Contents

1. Introduction

2. Methodology

3. Experiments

4. Conclusion

Overall Setup

- **Datasets :**
 - Multi-hop QA : 2WikimultihopQA, HotpotQA, MuSiQue
 - Simple QA : PopQA
- **Metrics :**
 - Exact Match (EM), F1
- **Baselines :**
 - Simple retriever : Contriever, BM25, GTR(T5-base)
 - Large Embedding Models : GritLM-7B, NV-Embed-v2(7B)
 - Structure-Augmented RAG : RAPTOR, GraphRAG, HippoRAG2(Backbone)
- **Implemental model :**
 - LLM : Qwen/Qwen2.5-7B-Instruct
 - Embedding : nvidia/NV-Embed-v2(7B)

QA Result

- **2WikiMultiHopQA**
 - 가장 큰 성능 개선 폭 관찰
 - 대부분의 기존 baselines 대비 우수한 성능
 - **Gating 효과 분석**
 - 2Wiki / HotpotQA / MuSiQue
 - Gating Off에서 더 안정적이거나 우수한 성능
 - QA 벤치마크는 이미 QA task에 맞게 정제되어 있어
→ 추가적인 gating이 필요하지 않은 경우가 많음
 - **HotpotQA의 예외적 특성**
 - Gating On에서 최고 성능 달성
 - 다수의 노이즈 문서.엔티티를 포함한 데이터셋
 - Gating이 구조적 정보 중 유효한 신호만 선택
 - **PopQA (Single-hop QA)**
 - HippoRAG2 대비 성능 향상은 확인
 - 그러나 기존 semantic baselines가 여전히 더 강력
 - 단순 질의에서는 구조 기반 재정렬의 이점이 제한적
- **Hyper-Reranking**은 복잡하고 계층적인 QA에서 가장 효과적
→ 특히 노이즈가 존재할 때, gating 메커니즘의 유효성 입증

Model	Multi-Hop QA			Simple QA
	2Wiki	HotpotQA	Musique	PopQA
BM25	9.70 / 22.16	21.90 / 35.99	5.10 / 12.07	20.60 / 33.06
Contriever	9.90 / 23.89	20.90 / 36.09	7.90 / 17.40	23.80 / 35.41
GTR-t5-base	9.90 / 25.45	18.30 / 35.00	8.00 / 17.82	24.50 / 36.68
GritLM	12.50 / 29.83	29.70 / 47.96	11.60 / 24.29	26.30 / 37.77
NVEmbedV2	15.60 / 32.43	30.30 / 49.22	10.90 / 22.97	<u>27.20</u> / 38.41
RAPTOR	19.20 / 22.62	17.80 / 21.87	6.10 / 8.56	18.10 / 28.96
GraphRAG	17.30 / 22.35	<u>33.50</u> / 42.75	11.30 / 17.97	34.20 / 45.37
HippoRAG2 (Base)	20.60 / 39.66	33.20 / <u>51.14</u>	<u>14.10</u> / <u>26.81</u>	25.80 / 38.52
Ours (Off)	22.90 / 42.20 (+11.17% / +6.41%)	32.90 / 51.08 (−0.90% / −0.12%)	14.30 / 26.94 (+1.42% / +0.48%)	26.30 / <u>39.31</u> (+1.94% / +2.05%)
Ours (On)	<u>22.20</u> / <u>41.53</u> (+7.77% / +4.71%)	34.60 / 52.80 (+4.22% / +3.25%)	13.70 / 26.58 (−2.84% / −0.86%)	26.20 / 38.81 (+1.55% / +0.75%)

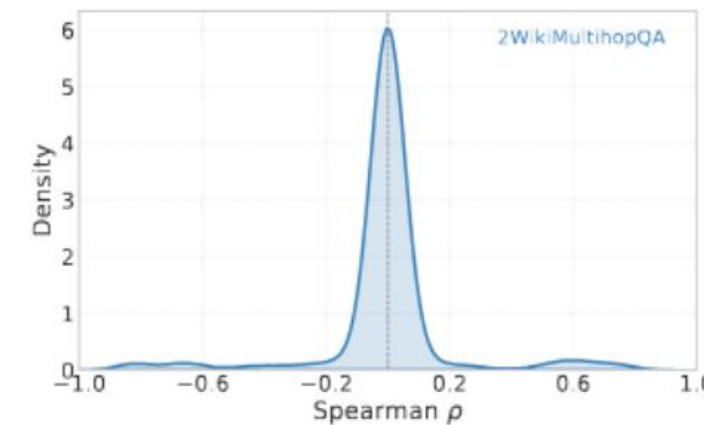
Ablation Study

- HotpotQA 데이터셋 기준 분석
 - **w/o Structural Features**
 - 가장 큰 성능 하락 발생 ($\Delta F1 = -5.47\%$)
 - 구조 정보가 Hyper-Reranking의 핵심 기여 요소임을 명확히 보여줌
 - **w/o Gating / w/o HGAT**
 - 모두 일관된 성능 감소 관찰
 - 구조 신호를 선택적으로 조절하고 집약하는 과정이 필요함을 시사
- 모든 구성 요소가 상호 보완적으로 작용
- Hyper-Reranking의 성능 향상은 단일 요소가 아닌 구조적 설계 전반의 기여 결과

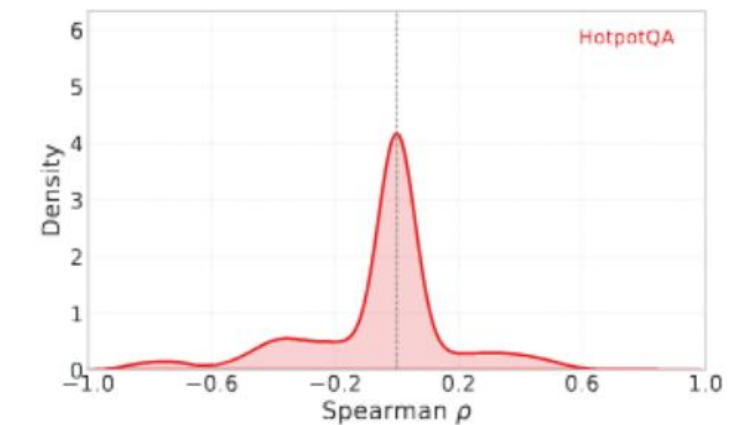
Components	EM	F1	$\Delta F1$
Ours	34.60	52.80	-
w/o Gating	32.90	51.08	-3.26 %
w/o Struct Feats	31.80	49.91	-5.47 %
w/o HGAT	33.80	51.46	-2.53 %

Impact analysis of Dataset Structure

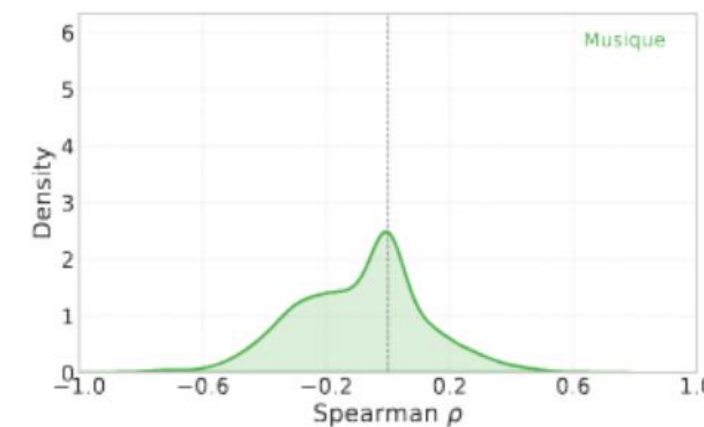
- **2WikiMultiHopQA / HotpotQA**
 - 의미-구조 유사도 간 상관계수 $\rho \approx 0$
 - 의미적 유사도만으로는 계층적 구조를 포착하지 못함
 - Hyper-Reranking이 미활용 구조 정보 보완
→ 큰 성능 향상 (+6.41%, +3.25% F1)
 - **MuSiQue / PopQA**
 - 의미-구조 유사도가 부분적으로 중첩
 - 이미 semantic embedding이 일부 구조 정보 반영
→ 개선 폭 제한적
 - **MuSiQue 추가 관찰**
 - 본질적으로 **2-hop** 구조
 - 3-hop 확장은 노이즈 유입 → 구조 일관성 저하
 - depth 2 사용 시 F1 26.94 → 27.56 개선
- **Semantic-Structural misalignment**가 클수록
Hyper-Reranking 효과 큼
- 데이터셋별 계층 구조 특성에 맞춘 구조 활용이 핵심



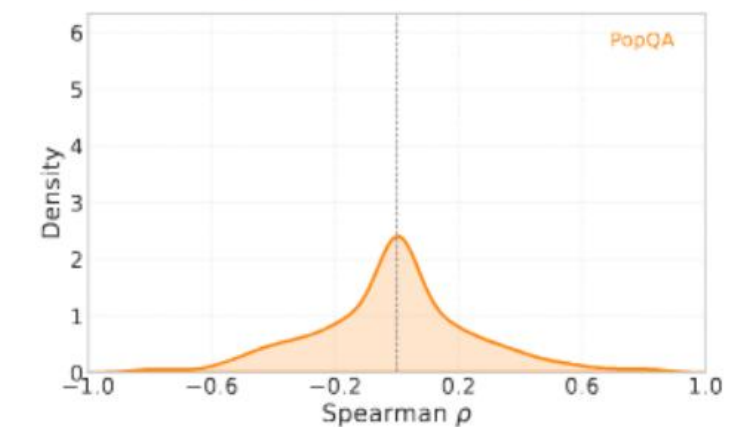
(a) 2WikiMultiHopQA



(b) HotpotQA



(c) MuSiQue



(d) PopQA

Case Study

- 기존 문제 상황
 - Reranking 이전에도 상위 후보에 정답 노드가 이미 존재
 - Kevin Durant(2016), Stephen Curry(2017) 등
의미적으로 유사하지만 계층적 깊이가 다른 노드들이 함께 상위에 위치하며 혼란 발생
- Hyper-Reranking 효과
 - 하이퍼볼릭 구조 정보를 반영해 계층 깊이 차이를 분리
 - Durant-Curry 노드의 순위 조정으로 구조적 혼란 제거
 - 결과적으로 정답 노드가 명확히 상위로 부각

Question	
In what year was the Golden State NBA player, who was part of the Cavaliers–Warriors rivalry, named NBA Finals Most Valuable Player?	
Gold Answer	
2015	
Before	After
The Golden State player was Kevin Durant in 2016 and Stephen Curry in 2017. If we consider the first instance, it would be 2016 .	2015
1. Cavaliers–Warriors rivalry 2. 2015 NBA Finals 3. Andre Iguodala 5. Kevin Durant 6. Stephen Curry	1. Cavaliers–Warriors rivalry 2. 2015 NBA Finals 3. Andre Iguodala 4. Stephen Curry 10. Kevin Durant

Contents

1. Introduction

2. Methodology

3. Experiments

4. Conclusion

Conclusion

- **Hyper-Reranking**은 semantic-structural misalignment를 완화하는 structure-aware GraphRAG reranking module
- Hyperbolic space에서 Q-graph와 C-subgraphs를 정렬하여 Euclidean 임베딩에서 손실된 구조적 신호를 복원
- Retriever backbone을 수정하지 않고 reranking 단계만으로 추론 품질을 효과적으로 향상
- 2WikiMultiHopQA, HotpotQA와 같이 계층 구조를 반영하지 못하는 데이터셋에서 가장 큰 성능 개선 확인
- Semantic-only retriever가 활용하지 못한 구조 정보를 복원하여 multi-hop QA의 핵심 한계를 해결
- 향후 generation-time reasoning 결합 및 selective-hop C-subgraph 확장을 통한 추가 성능 향상 가능성 제시

감사합니다.