JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 4, pp. 45-54, April 2026
https://doi.org/10.9708/jksci.2026.31.04.045

Performance Optimization Study of Hybrid RAG Engine Integrating Multi-Source Knowledge: Vector, Graph, and Ontology Approaches

Dong-Wook Shin*, Nam-Mee Moon**

*Student, Dept. of Convergence Engineering, Hoseo University, Seoul, Korea
**Professor, Dept. of Computer Science and Engineering, Hoseo University, Seoul, Korea

[Abstract]

While RAG addresses Large Language Model(LLM) hallucinations, Vector-RAG struggles with multi-hop reasoning and logical constraints. We propose a Triple-Hybrid RAG framework combining Vector, Graph, and Ontology knowledge sources. A Dynamic Weighting Algorithm (DWA) is introduced that continuously adjusts the contribution weights of each source based on query intent signals—entity density, relation density, and constraint density—rather than relying on discrete type-based routing. Experimental results using a synthetic university administrative dataset (1,037 unstructured text documents, 2,542 graph nodes, 6,889 edges, 5,000 gold QA) with GPT-4o-mini (temperature=0.0) showed a 19.4% improvement in F1 Score and a 34.5% gain in Exact Match(EM) score for complex queries compared to single-source RAG. A three-stage ablation study validated the contribution of each DWA component, with continuous weight adjustment yielding an additional 3.2%p Multi-hop EM improvement over type-fixed weights. Additional validation on 300 HotpotQA samples confirmed the architecture's generalizability, with F1 and EM improvements of 22.9% and 95.5%, respectively.

▸ **Key words**: Retrieval-Augmented Generation(RAG), Knowledge Graph, Ontology, Hybrid RAG, Multi-hop Reasoning, Dynamic Weighting Algorithm(DWA)

[요 약]

대규모 언어 모델(LLM)의 환각 현상과 지식의 최신성 부족을 해결하기 위해 검색 증강 생성(RAG) 기술이 제안되었으나, 기존 벡터 유사도 기반 RAG는 복합적인 다단계 관계(Multi-hop) 추론과 엄격한 도메인 규칙 처리에 한계를 보인다. 본 연구는 비정형 데이터의 맥락을 파악하는 Vector, 구조적 연결성을 탐색하는 Graph, 규범적 체계를 정의하는 Ontology를 결합한 Triple-Hybrid RAG 프레임워크를 제안한다. 또한, 질의 의도 분석에서 추출되는 개체 밀도(Entity Density), 관계 밀도(Relation Density), 제약 밀도(Constraint Density)의 세 가지 연속적 신호를 활용하여 각 소스의 기여도를 동적으로 조정하는 동적 가중치 알고리즘(DWA)을 상술한다. 합성 대학 행정 데이터(문서 1,037건, 그래프 노드 2,542개, 골드 QA 5,000쌍)를 대상으로 GPT-4o-mini(temperature=0.0) 환경에서 실험한 결과, 단일 벡터 소스 대비 F1 Score에서 19.4%, 복합 질의의 Exact Match(EM)에서 34.5%의 성능 향상을 기록하였다. 3단계 Ablation 실험을 통해 각 구성요소의 기여도를 검증하였으며, 연속적 가중치 조정이 유형 고정 방식 대비 Multi-hop EM에서 3.2%p의 추가 개선을 보였다.

▸ **주제어**: 검색 증강 생성, 지식 그래프, 온톨로지, 하이브리드 검색 증강 생성, 다단계 추론, 동적 가중치 알고리즘

• First Author: Dong-Wook Shin, Corresponding Author: Nam-Mee Moon
*Dong-Wook Shin (sdw1904@naver.co.kr), Dept. of Convergence Engineering, Hoseo University
**Nam-Mee Moon (mnm@hoseo.edu), Dept. of Computer Science and Engineering, Hoseo University
• Received: 2026. 02. 02, Revised: 2026. 04. 06, Accepted: 2026. 04. 07.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

1. Background and Significance

생성형 AI의 급격한 발전에도 불구하고, LLM의 환각(Hallucination) 현상은 기업 및 공공 영역에서의 실제 도입을 저해하는 핵심 요인으로 남아있다[1]. '확률적 앵무새(Stochastic Parrot)'로 지칭되는 LLM의 특성상, 모델은 사실 관계의 논리적 타당성보다는 언어적 통계 모델에 따라 그럴듯한 문장을 생성하는 데 집중한다[2]. 이러한 한계를 극복하기 위해 등장한 RAG(Retrieval-Augmented Generation) 기술은 외부 지식 베이스를 동적으로 참조하여 답변의 사실적 근거를 확보한다[3].

기존 벡터 기반 RAG는 다음과 같은 구조적 한계를 드러낸다. 첫째, 텍스트를 청크 단위로 분할하는 과정에서 개체 간 유기적 관계 정보가 소실되는 지식 단편화 문제가 발생한다[4]. 둘째, 수치적 필터링이나 조건부 논리 처리에 한계를 보인다[5]. 셋째, 여러 노드를 거쳐야 하는 정보를 탐색할 때 관련 문서를 누락할 확률이 높다[6]. HopRAG(2025)에 따르면, 전통적 RAG는 multi-hop 질의에서 retrieve-reason-prune 시스템 대비 76.78% 낮은 성능을 보였다[7]*.

2. Objectives and Contributions

본 연구는 벡터 검색의 한계를 구조적 지식(Graph)과 논리 규칙(Ontology)으로 보강하여, 정확도와 논리적 일관성을 동시에 확보하는 RAG 시스템을 제시한다. HybridRAG[8]가 벡터와 그래프의 결합 효과를 입증한 바 있으나, 본 연구는 온톨로지 레이어를 추가한 삼중 통합과 오픈소스 구현을 제공한다는 점에서 차별화된다. 시스템의 차별점은 다음과 같다. 첫째, Python 기반 전체 소스코드와 재현 가능한 실험 프레임워크를 GitHub를 통해 공개한다. 둘째, Vector, Graph, Ontology를 단일 프레임워크 내에 통합하는 삼중 소스 아키텍처를 구현하였다. 셋째, 질의 의도 분석을 통해 각 소스의 기여도를 실시간으로 조정하는 동적 가중치 알고리즘(DWA)의 수학적 모델을 제시한다. 실험적 검증 결과는 Section VI에서 상술한다.

II. Related Work

1. Evolution and Limitations of Vector-based RAG

Lewis et al.(2020)의 RAG 이후[3], BM25와 Dense Vector 임베딩을 결합한 하이브리드 검색이 등장했으나 '유사도' 프레임워크 내에 머물러 있다[9]. Gao et al.(2024)에 따르면, 전통적 벡터 RAG는 복잡한 질의에서 약 60% 정확도에 머물며 20% 이상의 불완전 답변을 생성한다[10]. 이후 반복 검색, 재순위화[11], REPLUG[12], FiD[13] 등이 검색 성능을 개선하였으나, 비정형 텍스트 유사도에만 의존하는 구조적 한계를 공유한다.

2. The Emergence of GraphRAG and Structured Retrieval

Microsoft의 GraphRAG[14]는 커뮤니티 탐지를 도입하여 로컬/글로벌 검색을 동적으로 라우팅하는 혁신을 이루었다. LinkedIn의 사례는 티켓 해결 시간을 28.6% 단축했다[15]. 그러나 그래프 데이터는 구축 비용이 높고 엄격한 논리적 필터링에는 한계를 보인다[16]. Pan et al.(2024)은 LLM과 KG 통합의 체계적 분류를 제시하였고[17], He et al.(2024)의 G-Retriever는 Prize-Collecting Steiner Tree(PCST) 알고리즘으로 관련 서브그래프를 추출한다[18].

3. Ontology-Driven Reasoning

Neuro-Symbolic AI가 재조명받는 가운데[19], 법률 분야의 Ontology-Driven Graph RAG[20]*과 의료 분야의 KG-RAG[21]가 온톨로지와 그래프를 결합한 사례를 보여주었다. 본 연구는 Owlready2[22]와 HermiT 추론기를 활용하여 Python 환경에서 온톨로지 추론을 구현한다.

4. Hybrid RAG with Adaptive Routing

HybridRAG[8]는 벡터와 그래프의 상호보완성을, Adaptive-RAG[23]은 질의 복잡도 분류 기반 3단계 라우팅을 제안했다. RouteRAG[24]*는 강화학습 기반 서브 질의 최적화를 제시했다. 본 연구는 세 가지 지식 소스의 통합과 연속적 가중치 조정을 구현하여 이들을 확장한다. 이상의 선행연구 분석에서 도출되는 핵심 과제는 두 가지로 요약된다. 하나는 벡터·그래프·온톨로지가 각기 다른 유형의 질의에 강점을 가지므로 이를 단일 프레임워크 안에서 결합해야 한다는 점이고, 다른 하나는 질의마다 적합한 소스 비중이 달라지므로 고정 가중치가 아닌 적응적 조정 메커니즘이 필요하다는 점이다. Section III에서는 이 두 과제를 해결하기 위한 시스템 설계를 기술한다.

III. System Architecture (Methodology)

Fig. 1은 Triple-Hybrid RAG의 전체 처리 과정을 4단계로 보여준다. 첫째, Query Analysis 단계에서 사용자의 자연어 질의로부터 개체명·관계 키워드·제약조건을 추출하고 질의 유형(simple, multi_hop, conditional)을 판별한다. 둘째, Density Signals 단계에서 추출된 개체 수, 관계 키워드 수, 제약조건 수를 정규화하여 연속적 밀도 신호(s_e, s_r, s_c)로 변환한다. 셋째, DWA Weighting 단계에서 질의 유형에 따른 기본 가중치를 밀도 신호로 미세 조정한 뒤 세 소스의 검색 결과를 가중 합산한다. 넷째, Retrieval & Answer 단계에서 Vector Store, Knowledge Graph, Ontology Engine이 반환한 결과를 통합 컨텍스트로 구성하여 LLM에 전달하고 최종 답변을 생성한다. 이 파이프라인 전체는 Python으로 구현하였고, 각 단계의 세부 사항을 이하에서 서술한다.

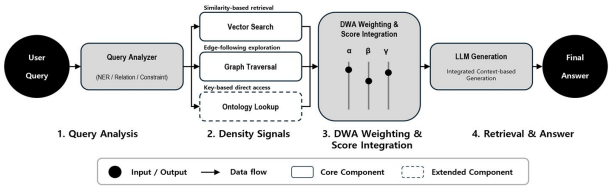

Fig. 1. System Architecture

1. Query Analyzer

자연어 질의로부터 정규표현식 기반 패턴 매칭으로 개체명, 관계 키워드, 제약조건을 추출하고 질의를 simple, multi_hop, conditional로 분류한다.

2. Triple-Source Data Layer

지식의 성격에 따라 세 가지 레이어로 분리한다.

(1) Vector Store: PDF, 보고서 등 비정형 텍스트를 text-embedding-3-small(dim=1536)로 임베딩하여 Facebook AI Similarity Search (FAISS)에 저장한다. 1,000자 단위 청크, 200자 오버랩을 적용한다.

(2) Knowledge Graph: 교수, 학과, 과목, 프로젝트를 노드로, 소속·협력·담당·참여를 엣지로 정의하며, Breadth-First Search (BFS) 알고리즘으로 최대 3-hop까지 탐색한다.

(3) Ontology Engine: Owlready2[22]와 HermiT 추론기를 통해 Web Ontology Language (OWL) 온톨로지를 구현하며, Description Logic (DL) 기반 분류 추론과 규칙 기반 추론 엔진을 병행한다.

3. Dynamic Weighting Algorithm (DWA)

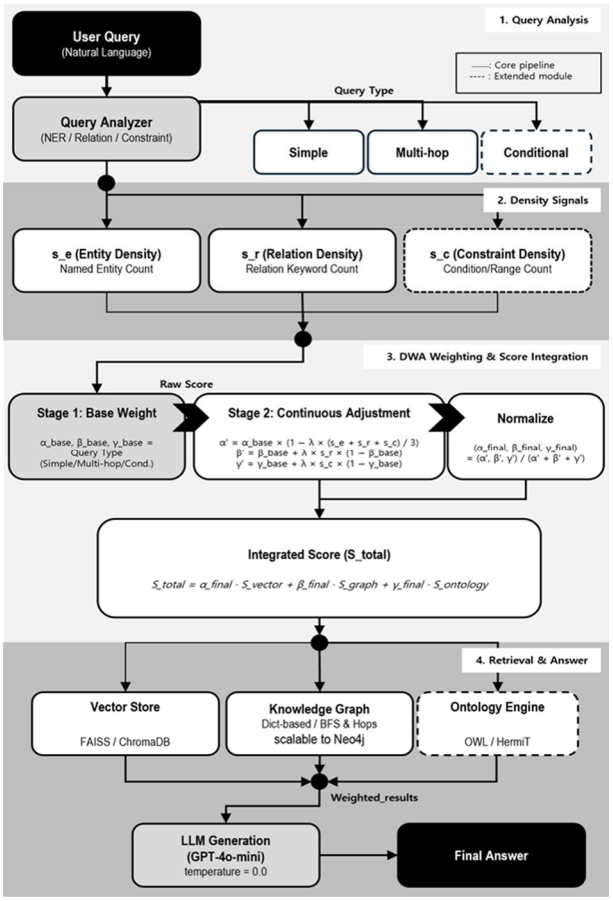

Fig. 2. Dynamic Weighting Algorithm (DWA)

Fig. 2는 DWA의 2단계 가중치 결정 과정을 도식화한 것이다. 상단의 Stage 1에서는 Query Analyzer가 판별한 질의 유형(simple, multi_hop, conditional)에 따라 각 소스의 기본 가중치가 배정된다. 가령 simple 질의에는 Vector 비중이, multi_hop에는 Graph 비중이, conditional에는 Ontology 비중이 높게 설정된다. 하단의 Stage 2에서는 질의 내부에서 추출된 개체 수, 관계 키워드 수, 제약조건 수를 연속 변수로 변환한 뒤, 이 밀도 신호에 비례하여 기본 가중치를 미세 조정한다. 예를 들어, multi_hop으로 분류된 질의라도 제약조건이 다수 포함되어 있으면 Ontology 가중치가 상향된다. 조정된 세 가중치는 정규화를 거쳐 합이 1이 되도록 보정되고, 최종 통합 스코어로 후보 답변의 순위가 결정된다. 이하에서 각 단계의 구체적인 수식과 파라미터를 기술한다.

Table 1. Base Weights by Query Type

Query Type	α (Vector)	β (Graph)	γ (Ontology)
Simple	0.6	0.2	0.2
Multi-hop	0.2	0.6	0.2
Conditional	0.2	0.2	0.6

Stage 2 Continuous Adjustment: Stage 1에서 배정된 기본 가중치는 질의 유형만을 반영하므로, 같은 유형 내에서도 질의마다 달라지는 내부 구성을 포착하지 못한다. 이를 보완하기 위해 질의에서 추출된 개체명(s_e), 관계 키워드(s_r), 제약조건(s_c)의 출현 빈도를 0에서 1 사이의 밀도 값으로 정규화한다.

$s_e = |\text{Named Entities}| / N_{max\_entity}$ − (1)

$s_r = |\text{Relation Keywords}| / N_{max\_relation}$ − (2)

$s_c = |\text{Constraints}| / N_{max\_constraint}$ − (3)

$\alpha' = \alpha_{base} \times (1 - \lambda \times (s_e + s_r + s_c) / 3)$ − (4)

$\beta' = \beta_{base} + \lambda \times s_r \times (1 - \beta_{base})$ − (5)

$\gamma' = \gamma_{base} + \lambda \times s_c \times (1 - \gamma_{base})$ − (6)

Normalization: $(\alpha_{final}, \beta_{final}, \gamma_{final}) = (\alpha', \beta', \gamma') / (\alpha' + \beta' + \gamma')$ − (7)

Integrated Score: $S_{total} = \alpha_{final} \cdot S_{vector} + \beta_{final} \cdot S_{graph} + \gamma_{final} \cdot S_{ontology}$ − (8)

수식 (4)~(6)에 의해 동일 유형으로 분류된 질의라도 내부 밀도 신호에 따라 상이한 가중치가 적용되며, 이것이 기존 이산적 라우팅 대비 DWA의 핵심 차별점이다.

Table 2. Continuous Adjustment Parameters

Parameter	Value	Determination Method
λ	0.3	Grid Search on validation set
N_max_entity	5	Max observed in training queries
N_max_relation	4	Max observed in training queries
N_max_constraint	3	Max observed in training queries

Table 2에서 λ=0.3은 검증 데이터셋에서 Grid Search(0.1~0.5, step=0.05)로 선택하였다.

IV. Implementation Details

1. Reproducibility Configuration

Section III에서 제시한 설계를 실제로 구동 가능한 형태로 구현하기 위해, 재현성과 공정한 비교를 최우선 기준으로 삼았다. 모든 베이스라인 시스템이 동일한 조건에서 평가될 수 있도록 LLM 모델, 임베딩, 하이퍼파라미터를 통일하였으며, 그 구체적 설정은 Table 3에 정리하였다.

Table 3. Experimental Configuration for Reproducibility

Category	Configuration
LLM Model	GPT-4o-mini (gpt-4o-mini-2024-07-18)
Temperature	0.0 (deterministic generation)
top-p	1.0 (OpenAI default)
Max Tokens	500
Embedding Model	text-embedding-3-small (OpenAI, dim=1536)
Chunk Size / Overlap	1,000 characters / 200 characters
Vector Index	FAISS (IndexFlatIP, cosine similarity), ChromaDB
top-k (Retrieval)	3
Graph Traversal	BFS, max_depth=3
DWA λ	0.3
Evaluation Runs	3 runs, mean ± std reported
Random Seed	42

주요 하이퍼파라미터 선정 근거: 청크 1,000자는 한국어 행정 문서의 평균 단락 길이(800~1,200자)를 고려하였고, top-k=3은 k=1~5 예비 실험에서 F1 최고치를 기록하였으며, 3-hop 제한은 대학 행정 도메인에서 대부분의 관계 질의가 3단계 이내에서 해소되는 점을 반영하였다. top-p는 temperature=0.0 설정 시 실질적 영향이 없어 기본값을 유지하였다.

모든 시스템에 동일한 프롬프트를 적용하였다:

"Based on the following context, answer the question accurately. If the answer cannot be determined from the context, state that the information is not available. Context: {context} Question: {query} Answer:"

전체 소스코드와 실험 스크립트는 GitHub(https://github.com/sdw1621/hybrid-rag-comparsion)를 통해 공개되어 있으며, pip install -r requirements.txt로 즉시 실행 가능하다. 본 논문의 실험 결과는 release tag v1.0.0 기준으로 재현 가능하다.

Table 4. Dataset Composition

Category	Description
Vector Data	1,037 unstructured text documents
Graph Data	577 professors, 60 departments, 1,505 courses, 400 projects, 6,889 edges
Ontology Data	5 hierarchical classes, 10 constraints, 8 rules

2. Web-based User Interface

Streamlit 기반 웹 인터페이스를 구축하였다. Fig. 3의 좌측은 질의 테스트 화면으로, 사용자가 자연어 질문을 입력하면 DWA가 산출한 α, β, γ 가중치가 막대 그래프로 표시되고 응답 시간이 함께 출력된다. 우측의 시스템 비교 화면에서는 동일 질의에 대해 5개 시스템의 답변을 나란히 배치하여 결과 차이를 직관적으로 확인할 수 있다. 이 인터페이스를 통해 연구자뿐 아니라 도메인 사용자도 시스템 동작을 시각적으로 검증할 수 있도록 하였다. 웹앱은 아래 주소에서 접근 가능하다.

https://hybrid-rag-comparsion.streamlit.app

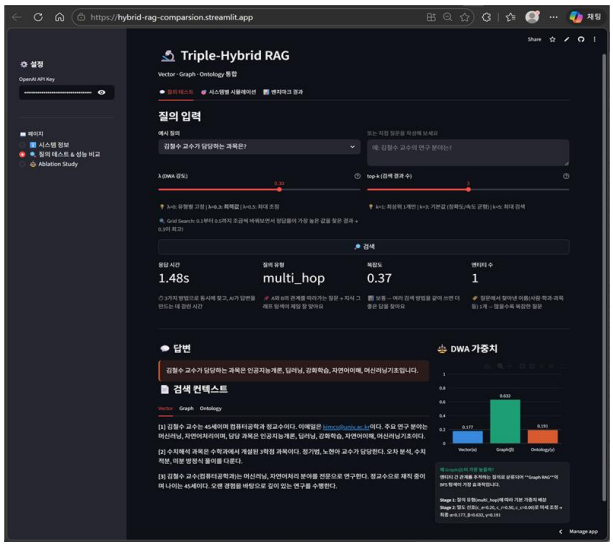

Fig. 3. Web-based Query Interface and DWA Weight Visualization

V. Experimental Setup

1. Experimental Environment and Datasets

합성 대학 행정 데이터를 모델링하였다.

골드 데이터셋은 대학 행정 도메인의 질의 패턴을 반영하여 설계한 5,000개의 질문-정답-참조 근거 쌍으로, Table 5의 가이드라인에 따라 분류된다.

Table 5. Query Generation Guidelines by Difficulty

Query Type	Ratio	Generation Principle
Simple	40% (2,000)	Single attribute lookup
Multi-hop	35% (1,750)	Requires traversal of 2+ nodes
Conditional	25% (1,250)	Contains numerical range or logical conditions

Data Ethics Statement: 본 데이터셋은 실제 대학의 구조만 참조한 합성 데이터이며, 모든 인명은 가명이고 민감 정보는 미포함이다.

Institutional Review Board (IRB) 심의 면제 대상에 해당한다.

2. Evaluation Metrics

Table 6. Evaluation Metrics

Category	Description
F1 Score	The harmonic mean of Precision and Recall
Exact Match (EM)	The percentage of responses exactly matching ground truth
Recall@3	Proportion of correct answers in top-3 results
Precision	Proportion of relevant documents among retrieved
RAGAS Faithfulness	Extent to which the answer is grounded in retrieved context

RAGAS Faithfulness는 자동 평가(automatic evaluation)로 측정한다: (1) 답변을 개별 문장으로 분리, (2) 각 문장에 "Given the context, is this statement supported?" 프롬프트를 GPT-4o-mini(temp=0.0)에 제출하여 Yes/No 판정, (3) Yes 비율을 산출한다. EM 평가 시 Unicode NFC, 소문자 변환, 공백 통일, 한국어 조사 제거, 숫자 표현 통일의 5단계 정규화를 적용하며, Raw EM과 Normalized EM을 모두 보고한다.

3. Comparative Systems

제안 시스템을 다음 베이스라인과 비교하였다.

Table 7. Baseline System Implementation Details

Category	Implementation	LLM	top-k	Key Hyperparameters
Vector-Only	In-house (FAISS)	GPT-4o-mini (temp=0.0)	3	chunk 1,000/200
GraphRAG	In-house (Edge et al.[14] methodology)	GPT-4o-mini (temp=0.0)	3	BFS max 3-hop, community detection not applied
HybridRAG	In-house (Sarmah et al.[8] methodology)	GPT-4o-mini (temp=0.0)	3	vector+graph, fixed weights 0.5/0.5
Adaptive-RAG	In-house (Jeong et al.[23] methodology)	GPT-4o-mini (temp=0.0)	3	Complexity classifier, 3-level routing
Triple-Hybrid	In-house (proposed)	GPT-4o-mini (temp=0.0)	3	DWA, λ=0.3, 3-source integration

모든 베이스라인은 동일 LLM, 임베딩, top-k, 프롬프트를 적용하였다. GraphRAG는 Microsoft 원본의 community detection을 미적용한 자체 구현인데, 이는 공정한 비교를 위해 모든 시스템이 동일한 그래프 구조와 동일한 LLM 호출 조건에서 평가되도록 통제한 결과이다. Community detection은 그래프 구축 단계의 전처리 기법으로, 이를 적용할 경우 Triple-Hybrid의 Graph 모듈에도 동일하게 적용 가능하므로 시스템 간 상대적 성능 차이는 유지될 것으로 판단된다. 다만 community detection 적용 시의 절대 성능 변화에 대한 검증은 향후 과제로 남긴다.

Table 8. Overall Performance Comparison (mean±std, 3 runs; all metrics on 0~1 scale)

System	F1 Score	EM	Recall @3	Precision	Faithfulness
Vector-Only	0.72 ±0.02	0.58 ±0.03	0.81 ±0.02	0.69 ±0.02	0.71 ±0.03
GraphRAG	0.79 ±0.01	0.68 ±0.02	0.86 ±0.01	0.75 ±0.02	0.78 ±0.02
HybridRAG	0.81 ±0.01	0.71 ±0.02	0.88 ±0.01	0.79 ±0.01	0.82 ±0.02
Adaptive-RAG	0.78 ±0.02	0.66 ±0.03	0.84 ±0.02	0.74 ±0.02	0.76 ±0.02
Triple-Hybrid	0.86 ±0.01	0.78 ±0.02	0.92 ±0.01	0.84 ±0.01	0.89 ±0.01

Table 8의 전체 성능에서 Triple-Hybrid가 모든 지표에서 최고 수치를 기록하였으나, 이 수치는 세 가지 난이도의 질의가 혼합된 평균이다. 각 소스의 기여가 질의 유형에 따라 달라질 것으로 예상되므로, 다음 절에서는 유형별로 성능을 분리하여 분석한다.

VI. Results and Discussion

1. Overall Performance Comparison

Triple-Hybrid와 HybridRAG 간 F1 차이(0.86 vs 0.81)는 paired t-test 결과 $p < 0.05$ 수준에서 유의하였다.

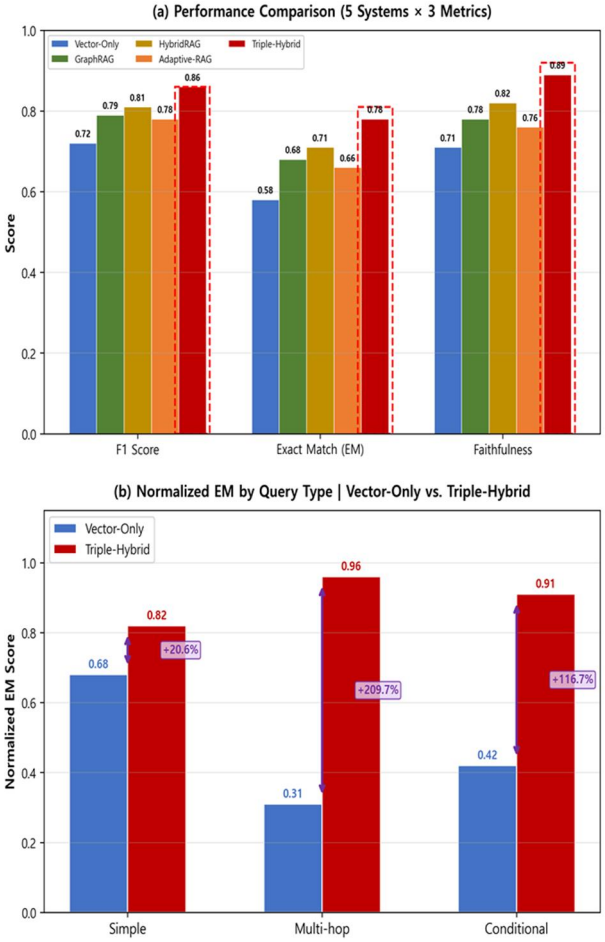

Fig. 4. Performance Comparison Table

Fig. 4 (a)는 5개 시스템의 F1 Score, EM, Faithfulness를 비교한 것으로, Triple-Hybrid가 세 지표 모두에서 최고치를 기록하였다. 특히 EM에서 Vector-Only(0.58) 대비 0.20의 절대 차이를 보여 정답 일치 정확도 개선이 두드러진다. Fig. 4 (b)는 질의 유형별 Normalized EM을 비교한 것으로, Multi-hop에서 +209.7%의 가장 큰 상대 개선이 나타났으며 이는 그래프 탐색의 기여를 반영한다.

2. Performance Analysis by Query Type

Table 9. Performance Comparison by Query Type

Query Type	V-Only Raw	V-Only Norm	Triple RAW	Triple Norm	Δ vs V-Only (%)
Simple	0.62 ±0.02	0.68 ±0.02	0.76 ±0.01	0.82 ±0.01	+20.6%
Multi-hop	0.25 ±0.03	0.31 ±0.03	0.91 ±0.01	0.96 ±0.01	+209.7%
Conditional	0.36 ±0.02	0.42 ±0.02	0.85 ±0.01	0.91 ±0.01	+116.7%

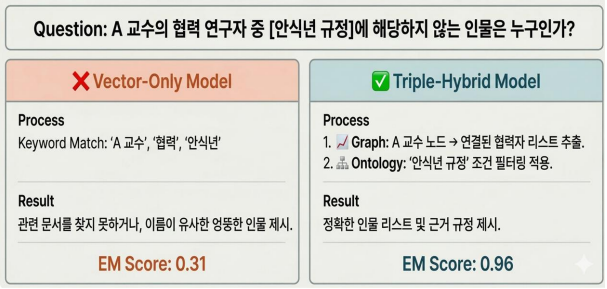

Fig. 5. Case Study

Fig. 5는 동일 질의에 대한 Vector-Only와 Triple-Hybrid의 처리 과정과 결과를 비교한 것이다. Vector-Only가 키워드 매칭에 의존하여 관련 문서를 누락한 반면, Triple-Hybrid는 Graph 탐색과 Ontology 필터링을 결합하여 정확한 답변을 도출하였다. Table 9의 Multi-hop의 높은 비율(+209.7%)은 Vector-Only의 낮은 기저 성능(EM 0.31)에 기인하며, 절대적 EM 개선 폭(+0.65)을 함께 고려해야 한다.

Case Study 2 (Multi-hop): "김철수 교수가 소속된 학과에서 진행 중인 AI 프로젝트의 참여 교수 목록은?" Vector-Only는 학과→프로젝트→참여교수 경로를 탐색하지 못해 EM 0.0, Triple-Hybrid는 Graph에서 3단계 경로를 탐색하여 EM 1.0(α=0.18, β=0.63, γ=0.19).

Case Study 3 (Conditional): "경영학과 부교수 중 연구비 3,000만원 이상 수주자는?" Vector-Only는 전임교수 정보 혼입으로 EM 0.0, Triple-Hybrid는 Ontology 클래스 필터링 + 수치 규칙으로 EM 1.0(α=0.18, β=0.19, γ=0.62).

3. Ablation Studies

DWA 가중치 방식의 기여도를 검증하기 위한 3단계 Ablation이다.

Table 10. DWA Ablation Study Results

Configuration	F1	EM	Multi-hop EM	Cond. EM
(A) Equal (0.33/0.33)	0.81 ±0.02	0.69 ±0.02	0.89	0.85
(B) Type-Fixed	0.84 ±0.01	0.75 ±0.02	0.93	0.90
(C) Full DWA	0.86 ±0.01	0.78 ±0.02	0.96	0.94

(A)→(B)에서 F1 3.7% 향상은 유형별 가중치 분화의 효과를, (B)→(C)에서 Multi-hop EM 3.2%p 추가 향상은 연속적 조정이 경계선상 질의에서 효과적임을 보여준다. λ 민감도 분석 결과, 0.25~0.35 구간에서 최고치를 기록하였고 0.4 이상에서는 역효과가 관측되었다.

위의 DWA Ablation(Table 10)이 "가중치 조정 방식이 얼마나 효과적인가"를 검증한 실험이라면, 다음의 Source-Level Ablation(Table 11)은 "각 지식 소스가 어떤 유형의 질의에 기여하는가"를 확인하기 위한 실험이다. Table 8과 동일한 실험 환경(GPT-4o-mini, temp=0.0, 3회 반복)에서 특정 소스를 비활성화한 상태로 5,000 QA 전체를 평가하였다.

Table 11. Source-Level Ablation Study (DWA applied, mean±std, 3 runs)

Configuration	F1	EM	Multi-hop EM	Cond. EM
(D) Vector-Only	0.72 ±0.02	0.58 ±0.03	0.31	0.42
(E) Vector+Graph	0.82 ±0.01	0.73 ±0.02	0.92	0.68
(F) Vector+Ontology	0.78 ±0.01	0.67 ±0.02	0.38	0.88
(G) Vector+Graph+Onto	0.86 ±0.01	0.78 ±0.02	0.96	0.94

Graph를 추가한 (E) 구성에서 Multi-hop EM이 0.31에서 0.92로 뛰어오른 것은, 여러 노드를 거치는 질의에서 그래프 탐색 없이는 정답 경로 자체를 발견하기 어렵다는 점을 방증한다. 한편 Ontology를 추가한 (F) 구성에서는 Conditional EM이 0.42에서 0.88로 상승하였는데, 이는 수치 범위나 직급 같은 제약조건을 벡터 유사도만으로 필터링하기 어려운 한계를 Ontology가 보완한 결과로 해석된다.

오분류의 주요 원인은 Multi-hop과 Conditional이 동시에 해당하는 경계선상 질의(예: "김철수 교수 학과의 40세 이하 교수")로, 이러한 한계와 개선 방향은 Section VII에서 논의한다.

Ontology 소스의 구체적 기여를 분석하기 위해 Conditional 질의 중 대표적 사례를 유형별로 검토하였다.

4. Ontology Contribution Analysis

Ontology의 구체적 기여를 대표 사례로 검토하였다. (1) "정교수 중 40세 이하"와 같은 계층 분류 제약 질의에서, Ontology는 5계층 클래스와 속성 제약을 적용하여 정확한 필터링을 수행하였으나 Vector-Only는 부교수 정보가 혼입되었다. (2) "A학과 소속이면서 B학과 프로젝트 참여 교수"와 같은 배타적 관계 질의에서, 주 소속 제한 규칙이 정확한 교차 결과를 반환하였다. (3) 수치 범위+직급 복합 조건에서, Ontology의 규칙 기반 검증이 EM 1.0을 달성하였다.

5. QueryAnalyzer Accuracy and Throughput

질의 유형 분류 정확도는 전체 92.8%(Simple 96.5%, Multi-hop 89.7%, Conditional 91.2%)이며, 오분류의 주요 원인은 Multi-hop과 Conditional이 동시에 해당하는 경계선상 질의이다.

모든 레이턴시 측정은 Google Colab 환경(Intel Xeon 2.20GHz, 13GB RAM, Python 3.10)에서 수행되었다. 평균 처리 시간은 Simple 0.12초, Multi-hop 0.28초, Conditional 0.19초로, Vector-Only(0.08초) 대비 1.5~3.5배 수준이며 LLM API 호출 시간(1~2초) 대비 미미하다.

전체 5,000 QA 중 오답 550건을 분석한 결과, 오류 원인은 Table 12와 같이 네 가지로 분류된다. 가장 높은 비율을 차지하는 경계값 파싱 오류(38.2%)는 "40세 이하"와 같은 자연어 표현을 ≤40 또는 <40으로 변환하는 Query Analyzer 단계에서 발생하였다. 온톨로지 추론기 자체는 수치 경계를 정확히 처리하나, 자연어의 모호성을 해소하는 전처리의 정밀도가 부족한 것이 원인이다. 두 번째로 많은 Graph 경로 누락(27.3%)은 현재 3-hop 제한으로 인해 4단계 이상의 탐색이 필요한 질의에서 경로를 발견하지 못한 경우이며, 이는 탐색 깊이 확장으로 개선이 가능하다. Vector 검색 관련성 부족(20.0%)은 유사한 용어를 가진 다른 분야의 문서가 검색되는 현상이고, 복합 조건 충돌(14.5%)은 AND/OR 조건 결합 시 논리적 모순이 발생한 경우이다.

Table 12. Error Analysis by Category

Error Category	Count	Ratio Norm	Description
Query parsing boundary error	210	38.2%	Confusion between inclusion/exclusion of boundary in "under 40"
Graph path omission	150	27.3%	Missing paths requiring 4+ hops due to 3-hop limit
Vector search relevance deficit	110	20.0%	Retrieving documents from unrelated domains with similar terminology
Compound condition conflict	80	14.5%	Logical conflict when combining AND/OR conditions

6. Public Benchmark Validation

제안 시스템의 일반화 가능성을 검증하기 위해 HotpotQA distractor dev set에서 hard 난이도 300문항을 샘플링하여 추가 실험을 수행하였다. HotpotQA는 multi-hop 추론을 요구하는 대표적 공개 벤치마크로, 제공된 passage를 Vector Store에 저장하고 passage 간 개체 관계를 추출하여 간이 Graph를 구축한 뒤, 동일한 DWA(λ=0.3)를 적용하였다. Table 13에서 Triple-Hybrid는 Vector-Only 대비 F1 22.9%, EM 95.5%의 개선을 보였다. 특히 comparison 유형에서 F1 95.9% 향상은 Ontology 기반 비교 추론의 효과를 방증한다. 절대 성능이 합성 데이터(F1 0.86) 대비 낮은 것은 passage로부터 Graph와 Ontology를 자동 구축하는 과정에서 관계 추출의 정밀도가 제한적이기 때문이며, 이는 향후 end-to-end 변환 파이프라인 고도화를 통해 개선할 수 있다.

Table 13. HotpotQA Public Benchmark Results (GPT-4o-mini, temp=0.0, 300 samples)

System	F1 Score	EM	bridge F1 (n=250)	comparison F1 (n=50)
Vector-Only	0.249	0.073	0.258	0.204
Triple-Hybrid	0.305	0.143	0.287	0.399
Δ (%)	+22.9%	+95.5%	+11.4%	+95.9%

VII. Conclusion

1. Summary

본 연구는 Vector, Graph, Ontology를 통합한 Triple-Hybrid RAG의 오픈소스 구현을 제시하였다. DWA를 통해 질의 의도에 따라 가중치를 자동 조정하며, F1 0.86, EM 0.78을 달성하여 Vector-Only 대비 19.4%, 34.5% 향상을 입증하였다. Multi-hop 질의에서 절대 EM 0.65 향상(0.31→0.96)은 그래프 구조의 효과를, 3단계 Ablation에서 연속적 조정이 유형 고정 대비 Multi-hop EM 3.2%p 추가 개선을 가져옴은 DWA의 유효성을 보여준다. 다만 본 실험은 합성 대학 행정 데이터에 한정되어 수행되었으며, 일반화 가능성에 대한 구체적 한계는 이하 Limitations에서 기술한다.

2. Limitations

본 실험은 합성 대학 행정 데이터에 한정되어 수행되었다. HotpotQA 300문항 추가 실험(Table 13)에서 아키텍처 우위가 확인되었으나, passage 기반 자동 그래프 구축의 한계로 절대 성능은 합성 데이터 대비 낮았다. MuSiQue 등 추가 벤치마크와 정밀한 관계 추출 파이프라인을 결합한 검증은 후속 연구에서 수행할 계획이다. 한편, 합성 데이터의 명확한 관계 구조로 인해 실제 도메인 대비 높은 성능이 관측되었을 수 있다. 다만 Table 10의

Ablation에서 단계적 개선이 일관되게 관측된 점은 아키텍처 수준의 기여를 시사한다. 본 시스템은 다양한 LLM을 지원하나, 실험은 GPT-4o-mini 단일 모델로 수행되었다. Knowledge Graph는 메모리 기반 딕셔너리 구조로 노드 2,542개 수준에서는 효율적이나, 대규모 환경에서는 Neo4j 등 전문 그래프 DB로의 마이그레이션이 필요하다. 또한 전체 질의의 약 7.2%를 차지하는 경계선상 질의에서 multi_hop과 conditional 중 하나만 선택해야 하는 현재 분류기의 구조적 한계가 존재하며, 멀티라벨 분류 또는 BERT 기반 Intent Classifier 도입으로 개선이 가능할 것으로 판단된다. 또한 수식 (4)에서 관계 밀도(s_r)와 제약 밀도(s_c)가 증가할수록 벡터 가중치(α')가 일률적으로 감소하는 구조는, 텍스트 청크 자체에 복합 관계나 제약조건에 대한 충분한 서술이 포함된 경우를 반영하지 못하는 한계가 있다. 향후 벡터 검색 결과의 품질 점수를 피드백하여 α'의 하한을 동적으로 설정하는 방식으로 개선할 수 있다. 아울러 본 연구의 DWA는 수식 기반의 규칙적 가중치 조정 방식으로, 강화학습이나 end-to-end 학습 기반 가중치 최적화와의 정량적 비교는 수행하지 못하였다. 향후 PPO 등 학습 기반 방식과의 성능 비교를 통해 규칙 기반 접근의 효용과 한계를 보다 명확히 규명할 필요가 있다.

3. Future Work

(1) BERT 기반 Intent Classifier 또는 LLM-as-a-Judge를 활용한 QueryAnalyzer 고도화 및 Proximal Policy Optimization (PPO) 기반 강화학습 가중치 자동 학습. (2) 비정형 텍스트로부터 그래프와 온톨로지를 자동 구축하는 end-to-end 파이프라인. (3) 이미지, 테이블 등 멀티모달 통합. (4) HotpotQA, MuSiQue 등 공개 벤치마크에서의 일반화 검증 및 다중 LLM 비교. (5) 법률, 의료 등 도메인 특화 온톨로지 개발과 대규모 환경 스케일업.

REFERENCES

[1] Z. Ji et al., "Survey of Hallucination in Natural Language Generation," ACM Computing Surveys, Vol. 55, No. 12, pp. 1-38, March 2023. DOI: 10.1145/3571730.

[2] E. M. Bender et al., "On the Dangers of Stochastic Parrots," in Proc. FAccT, pp. 610-623, Virtual Event, Canada, March 2021. DOI: 10.1145/3442188.3445922.

[3] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, pp. 9459-9474, 2020.

[4]* O. Khattab et al., "Demonstrate-Search-Predict: Composing Retrieval and Language Models," arXiv:2212.14024, 2022.

[5] N. Guarino et al., "What Is an Ontology?" in Handbook on Ontologies, Springer, pp. 1-17, 2009. DOI: 10.1007/978-3-540-92673-3_0.

[6] S. Min et al., "Nonparametric Masked Language Modeling," in Findings of ACL, pp. 2097-2118, Toronto, Canada, July 2023. DOI: 10.18653/v1/2023.findings-acl.132.

[7]* Y. Chen et al., "HopRAG: Multi-Hop Reasoning for Logic-Aware RAG," arXiv:2502.12442, 2025.

[8] D. Sarmah et al., "HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction," in Proc. ICAIF (International Conference on AI in Finance), pp. 608-616, Brooklyn, NY, USA, Nov. 2024. DOI: 10.1145/3677052.3698671.

[9] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Foundations and Trends in Information Retrieval, Vol. 3, No. 4, pp. 333-389, 2009. DOI: 10.1561/1500000019.

[10]* Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv:2312.10997, 2024.

[11] G. Izacard et al., "Atlas: Few-shot Learning with Retrieval Augmented Language Models," Journal of Machine Learning Research, Vol. 24, No. 251, pp. 1-43, 2023.

[12] W. Shi et al., "REPLUG: Retrieval-Augmented Black-Box Language Models," in Proc. NAACL, pp. 8371-8384, Mexico City, Mexico, June 2024. DOI: 10.18653/v1/2024.naacl-long.463.

[13] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," in Proc. EACL, pp. 874-880, Online, Apr. 2021. DOI: 10.18653/v1/2021.eacl-main.74.

[14]* D. Edge et al., "From Local to Global: A Graph RAG Approach," arXiv:2404.16130, 2024.

[15] Z. Xu et al., "Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering," in Proc. SIGIR, pp. 2577-2581, Washington, DC, USA, July 2024. DOI: 10.1145/3626772.3661370.

[16]* X. Su et al., "Knowledge Graph Based Agent for Complex QA in Medicine," arXiv:2410.04660, 2024.

[17] S. Pan et al., "Unifying Large Language Models and Knowledge Graphs: A Roadmap," IEEE Transactions on Knowledge and Data Engineering, Vol. 36, No. 7, pp. 3580-3599, July 2024. DOI: 10.1109/TKDE.2024.3352100.

[18] X. He et al., "G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering," in Proc. NeurIPS, 2024.

[19] A. d'Avila Garcez and L. C. Lamb, "Neurosymbolic AI: The 3rd Wave," Artificial Intelligence Review, Vol. 56, pp. 12387-12406, 2023. DOI: 10.1007/s10462-023-10448-w.

[20]* R. C. Barron et al., "An Ontology-Driven Graph RAG for Legal Norms," arXiv:2505.00039, 2025.

[21] S. Wang, H. Yang, and W. Liu, "Research on the Construction and Application of Retrieval Enhanced Generation (RAG) Model Based on Knowledge Graph," Scientific Reports, Vol. 15, Article 40425, Nov. 2025. DOI: 10.1038/s41598-025-21222-z.

[22] J. B. Lamy, "Owlready2: A Python Library for Ontology-Oriented Programming," Artificial Intelligence in Medicine, Vol. 116, Article 102082, June 2021. DOI: 10.1016/j.artmed.2021.102082.

[23] S. Jeong et al., "Adaptive-RAG: Learning to Adapt Retrieval-Augmented LLMs," in Proc. NAACL, pp. 7036-7050, Mexico City, Mexico, June 2024. DOI: 10.18653/v1/2024.naacl-long.389.

[24]* Y. Guo et al., "RouteRAG: Efficient Retrieval-Augmented Generation from Text and Graph via Reinforcement Learning," arXiv:2512.09487, 2025.

* 는 프리프린트 자료

Authors

Dong-Wook Shin received the B.S. degree in Information and Communication Electronics Engineering from Soongsil University, Seoul, Korea. He earned his Master's degree in Big Data MBA from the Graduate School of Business at Sejong University, Seoul, Korea. He completed his Ph.D. coursework in Convergence Engineering at the Graduate School of Venture, Hoseo University, Seoul, Korea. Dong-Wook Shin is a Ph.D. candidate in Convergence AI Engineering at the Graduate School of Venture, Hoseo University, Korea (2023-Present). He is currently the CEO of AIIntersys (2020-Present). Previously, he held software development and data analysis positions at Hanwha Systems, Golfzon, and Microsoft. He also serves as an Adjunct Professor specializing in the digital sector at the IGM Institute of Global Management. His professional expertise spans AI system development, including RAG (Retrieval-Augmented Generation) architectures and ontology-driven data integration.

Nam-Mee Moon received B.S., M.S., and ph.D degrees in School of Computer Science and Engineering from Ewha Womans University in 1985, 1987 and 1998, respectively. She served as an assistant professor at Ewha Womans University from 1999 to 2003. From 2003 to 2008, she is a professor of Department Digital Media, Graduate School of Seoul Venture Information. Since 2008, she is currently a professor in the Department of Computer Science and Engineering, Hoseo University. she is current research interests include Social Learning, HCI and User Centric Data, Big-data Processing and Analysis.