Multi-Hop을 요구하는 복잡한 질의처리 검색 효율화를 위한 RAG와 Graph RAG의 결합 프레임워크

이성민[1], 김희은[1], 이동영[1], 강민선[1], 김민재[1], 양수열[2], 황영숙[1]*
[1]고려대학교 컴퓨터학과, [2]크라우드웍스
{ku2332sm, hehek, gooddino3, rndni0531, kmj200392, youngsook_hwang}@korea.ac.kr, sy.yang@crowdworks.kr

Enhancing Multi-Hop Complex Query Retrieval Efficiency through the Integration of RAG and Graph RAG

Seong-min Lee[1], He-eun Kim[1], Dong-young Lee[1], Min-seon Kang[1],
Min-jae Kim[1], Soo-yeol Yang[2], Young-sook Hwang[1]*
[1]Department of Computer Science and Engineering, Korea University, [2]Crowdworks

요 약

본 논문은 복잡한 Multi-Hop(다단계 추론) 질의 처리를 효율화하기 위해 Vector RAG(Retrieval-Augmented Generation)와 Graph RAG를 유기적으로 결합하는 적응형 멀티에이전트 시스템을 제안한다. 기존의 Vector RAG 시스템은 다단계 추론 과정에서 컨텍스트 손실(contextual loss) 및 관계 정보 활용 미흡으로 인해 성능 한계를 보였다. 제안하는 프레임워크는 이러한 한계를 극복하기 위해 Graph RAG의 관계 정보를 Vector RAG 질의 정제에 활용하는 양방향 정보 전달 메커니즘과 플레이스홀더 기반 컨텍스트 전파 메커니즘을 도입했다. 또한, TriageAgent, OrchestratorAgent, DataGathererAgent, ProcessorAgent로 구성된 멀티에이전트 아키텍처를 통해 질의의 복잡도와 유형에 따라 최적의 검색 전략을 동적으로 선택하게 함으로써 시스템의 유연성을 극대화하고, 이와 더불어 사전 점검(pre-check) 메커니즘을 통해 불필요한 연산을 사전에 차단함으로써 전체적인 시스템의 효율성을 향상시키고자 하였다.
시스템의 성능은 식품 산업 리서치 도메인을 대상으로 2-hop에서 4-hop 수준의 복잡성을 가진 200개의 테스트 질의를 사용하여 평가하였다. 특히 "집중호우가 농산물 가격에 미친 영향"과 같이 명확한 인과관계 체인 분석이 요구되는 질의에서, Vector RAG의 세부적이고 풍부한 정보와 Graph RAG의 구조적인 관계 정보가 상호 보완적으로 작용하여 기존 RAG 시스템 대비 향상된 검색 정확도와 추론 성능을 보임을 확인하였다.

주제어: Multi-Hop RAG, Graph RAG, Vector RAG, 적응형 멀티에이전트 시스템, 플레이스 홀더 기반 컨텍스트 전파 매커니즘, 양방향 정보전달 매커니즘, 사전점검 매커니즘

1. 서론

최근 대규모 언어 모델(Large Language Model, LLM)에 정보를 의존하는 사람들이 많아지고 있는데 반해, LLM은 환각(Hallucination)이라는 고질적인 문제를 가지고 있다. 이는 LLM 내의 학습 데이터에 최신 정보나 특정 분야의 세부 정보가 포함되어 있지 않아 거짓된 정보를 생성하는 현상이다. 이를 해결하기 위해 LLM에 외부 지식을 결합하는 방법으로 Retrieval-Augmented Generation(RAG) 기술이 널리 연구되고 있다. RAG는 벡터 임베딩 기반의 검색을 통해 관련 문서를 찾아내고, 이를 LLM의 입력으로 활용해 정확하고 풍부한 응답을 생성하는 방법이다.

기존의 벡터 기반 RAG는 여러 단계를 거쳐 정보를 검색해야 하는 Multi-Hop(다단계 추론) 질의 처리에서 명확한 한계를 드러낸다. 특히 식품 산업과 같은 복잡한 도메인은 생산, 유통, 영양, 가격 등 다양한 정보가 상호 연결되어 있어 다양한 관점에서 리서치를 수행할 때 Multi-Hop 질의가 빈번하게 발생한다. 예를 들어, "2025년 7-8월 집중호우가 피해지역 농산물 가격에 미친 영향"과 같은 질의는 '집중호우 피해지역', '해당 지역 농산물', '가격 변동' 등 여러 정보 요소 간의 관계 체인을 추론해야 한다. 이러한 복잡한 추론 과정에서 기존의 RAG 시스템은 중간 단계의 맥락(context)을 유지하는 데 어려움을 겪고, 정보 간의 구조적 관계를 효과적으로 활용하지 못해 정확한 답변을 제공하는 데 한계가 있다.

본 논문에서는 식품산업 도메인을 대상으로 벡터 기반 RAG에 Graph RAG를 결합하여 이러한 문제를 해결하는 방안을 제안한다. 식품 도메인은 품목, 원산지, 영양성분, 가격, 유통 경로 등 명확한 엔터티와 엔터티 간의 관계가 존재하여 그래프 구조로 표현하기 적합하며, 동시에 상세한 보고서와 뉴스 기사 등 텍스트 문서도 풍부하여 벡터 검색의 장점도 활용할 수 있다.

GraphRAG와 Vector RAG를 결합한 새로운 아키텍처는 Graph-to-Vector와 Pre-check 두 가지 메커니즘으로 구성된다. Graph-to-Vector는 Graph에서 추출된 엔터티 관계 정보를 Vector RAG의 검색 쿼리 재구성에 활용하여 검색 퀄리티를 향상 시키는 방법이고, Pre-check 메커니즘은 Graph RAG를 사용하여 데이터의 관계 구조를 사전

에 파악하고, 이를 통해 에이전트가 계획을 수립함으로서, 효율적인 실행 계획을 수립할 수 있도록 한다. 본 연구의 목표는 식품 도메인에서 기존 RAG 기반 Multi-Hop 질의 처리 한계를 규명하고, GraphRAG와의 상호 보완적 결합을 통해 검색 효율성과 정확성을 향상 시키며, 복잡한 도메인 특화 질의에 대한 해결책을 제공한다.

2. 관련 연구

LLM의 성능을 높이기 위한 RAG 시스템 연구는 여러 방향에서 진행되어 왔다. 본 장에서는 주요 선행연구를 분석하고 본 연구의 위치를 명확히 한다.

Gao 등(2024)에 따르면, RAG의 패러다임은 인덱싱을 이용하는 전통적 Naive RAG부터 쿼리를 확장하거나 리랭킹을 하는 등 여러 전후처리를 추가한 Advanced RAG를 거쳐 여러 모듈을 유연하게 결합해 파이프라인을 구축하는 Modular RAG로 발전해왔다 [1].

Singh 등(2025)은 RAG에 자동화된 AI 에이전트를 도입한 Agentic RAG의 연구동향을 제시하고 있다. 기존 RAG가 정적 워크플로우만을 제공하고 다중 단계 추론에 취약하며 작업 유연성이 부족하다는 한계를 극복하기 위해 도입한 AI 에이전트가 동적으로 패턴을 실행한다. 덕분에 유연성, 확장성, 컨텍스트 적응력 면에서 크게 진화했다 [2]. 그러나 위 두 논문[1][2]은 특별하게 Multi-Hop 질의에 응답할 때의 RAG에 대한 문제를 다루지 않고 있다.

Multi-Hop RAG의 성능 평가와 관련하여, Tang과 Yang(2024)의 연구는 중요한 기준점을 제시한다. 연구진은 뉴스 데이터 기반의 벤치마크 데이터셋을 개발하여 RAG 시스템의 다단계 추론 능력을 측정했다. 실험 결과는 기존 시스템이 복잡한 연쇄 추론에서 현저한 성능 저하를 보인다는 사실을 밝혀냈다 [3]. 하지만 이 연구는 문제점 진단에 그쳤을 뿐, 실질적인 개선 방안까지는 다루지 않았다.

질의 분해 전략 측면에서 Zhang 등(2025)의 PAR RAG는 주목할 만한 접근법이다. 이들은 복잡한 질문을 체계적으로 분해하는 3단계 프레임워크를 설계했다. 계획 단계에서 하위 질문을 생성하고, 실행 단계에서 coarse-grained와 fine-grained 검색을 순차적으로 수행하며, 마지막으로 multi-granularity 검증을 통해 결과를 확인하는 구조다. 각 하위 질문은 사고 과정과 실제 질문으로 구성되어 추론의 투명성을 높였다 [4]. 그러나 이 시스템은 순차적 처리에만 의존하여 병렬 처리의 이점을 활용하지 못했고, 문서 간 관계 정보를 명시적으로 모델링하지 않았다.

그래프 기반 접근에서 Liu 등(2025)의 HopRAG는 혁신적인 시도였다. 문서 조각을 그래프 노드로 변환하고, 논리적 연결을 엣지로 표현하는 방식을 채택했다. 특히 가상 질문 생성을 통해 노드 간 관계를 자동으로 추론하는 메커니즘이 특징적이다. 검색 시에는 검색(retrieve), 추론(reason), 프루닝(prune)의 3단계를 거치며, Helpfulness 메트릭을 통해 결과를 재정렬한다 [5]. 다만 이 연구는 그래프 구조에만 집중하여 기존 벡터 검색의 장점을 충분히 활용하지 못했다는 한계가 있다.

메타데이터 활용 관점에서 Poliakov와 Shvai(2024)의 Multi-Meta-RAG는 실용적인 해법을 제시했다. 뉴스 데이터에 한정하여 문서의 출처와 출간 날짜를 LLM으로 추출하고, 이를 활용해 데이터베이스를 필터링하는 전략을 사용했다. 단순하지만 효과적인 이 방법은 특정 도메인에서 우수한 성능을 보였다 [6]. 그러나 메타데이터가 불명확하거나 복잡한 추론이 필요한 상황에서는 적용이 제한적이었다.

본 연구는 기존 연구의 한계를 극복하기 위해 다음과 같은 새로운 접근법을 제안한다. 첫째, 그래프 검색과 벡터 검색이 상호 보완적으로 작동할 수 있게 설계하였다. Pre-check 메커니즘으로 Graph RAG가 데이터 구조를 사전에 파악하여 효율적인 계획 수립이 가능하도록 하고, Graph-to-Vector 메커니즘으로 Graph의 관계 정보를 바탕으로 Vector RAG 검색 쿼리를 재구성한다. 둘째, 멀티에이전트 아키텍처를 통해 질의 특성에 따라 적절한 계획을 동적으로 수립하며, DataGathererAgent가 계획에 따라 정보 수집을 진행한다. 플레이스홀더 기반 컨텍스트 전파로 Multi-Hop 추론 시 정보 손실을 최소화했다. [step-X의 결과] 플레이스홀더로 전체 추론 체인을 명시적으로 유지하여 4-Hop 이상에서도 맥락이 온전히 전달되도록 한다.

3. VectorRAG/Graph RAG 시스템 구성 및 결합방법

3.1 VectorRAG의 구축

본 논문은 한국어 농식품 도메인에 특화된 Vector RAG 시스템을 구축한다. 이 시스템은 임베딩 모델로 한국어에 최적화된 BGE-M3-ko[7]를 사용하며, 대량의 웹 크롤링 데이터를 효율적으로 처리하기 위해 CrowdWorks의 Knowledge Compiler 서비스를 활용하여 문서를 파싱하고 정제한다.[1] 이 과정을 통해 정교한 임베딩 청크가 구축되며, 이는 이후 검색의 정확도를 높이는 기반이 된다.

또한 제안하는 시스템은 Dense 검색과 Sparse 검색을 결합하는 하이브리드 검색 전략을 채택한다. Dense 검색은 의미적 유사성을 기반으로 하며, Sparse 검색은 BM25 알고리즘을 활용한 키워드 매칭을 통해 문서를 검색한다. 이 두 검색 결과는 다음과 같은 수식에 따라 통합된다:

Score_final = α × Score_dense + (1-α) × Score_sparse (수식 1)

여기서 α 값을 0.5로 설정하여 두 방법의 균형을 맞춘다. 특히, Dense 검색에서는 상위 k=20개의 문서를 검색

1) Crowdworks의 KnowledgeCompiler(https://www.crowdworks.ai/en/agent/alpykc)를 활용하여 대량의 식품 도메인의 PDF 문서를 파싱하고, 파싱 결과를 이용하여 문서의 섹션단위 텍스트와 표를 구분하여 청크를 구성하였다.

하고, 100개의 후보군 중에서 근사 최근접 이웃(Approximate Nearest Neighbor) 탐색을 수행하여 효율성을 확보한다.

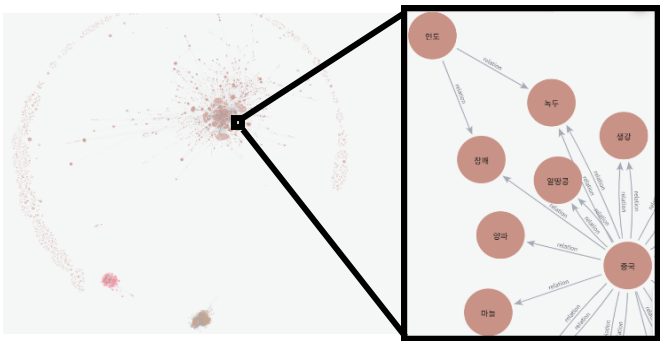
또한 농식품 도메인의 다양한 문서 특성을 고려하여 차별화된 인덱싱 전략을 적용한다. 시세 데이터에는 시계열 메타데이터를 포함하고 1.5의 가중치를 부여하며, 연구 보고서는 섹션별 청킹과 요약 임베딩을 추가하여 1.2의 가중치를 적용한다. 뉴스 기사는 시의성과 출처 신뢰도를 반영하여 1.0의 가중치를, 통계 테이블은 구조화 데이터 처리 및 컬럼명 인덱싱을 통해 1.3의 가중치를 부여한다. 마지막으로, 검색된 문서의 최종 품질을 향상시키기 위해 Cross-encoder 기반의 리랭커 모델[8]을 사용하여 검색 결과를 재정렬한다.

3.2 Graph RAG의 구축

Graph RAG 구축은 문서에서 엔터티와 관계를 추출하여 지식 그래프를 생성하는 과정으로 이루어진다. 먼저 RAG에 저장된 보고서에서 엔터티와 엔터티 간의 관계, 그리고 관계의 속성을 LLM을 이용해 추출한다. 엔터티는 농산물 품목명, 지역, 영양성분, 기업명 등 도메인 특화된 개체들을 포함하며, 관계는 "생산", "수출", "포함", "대체" 등의 의미적 연결을 나타낸다.

추출된 엔터티와 관계 정보에는 해당 관계를 추출한 문서의 메타정보를 추가하여 출처를 명확히 한다. 이후 GraphDB의 쿼리 언어인 Cypher를 이용해 Neo4j 데이터베이스에 삽입함으로써 그래프를 구축한다. 각 노드는 엔터티를 나타내고, 엣지는 엔터티 간의 관계를 표현하며, 엣지의 속성으로 관계 타입과 출처 문서 정보가 저장된다.

문서 내 관계 정보 이외에도 식품산업통계정보 시스템[2]aTFIS에서 수집한 식품원료 원산지 정보와 식약처의 '국가표준식품성분표'를 분석하여 품목의 원산지와 성분간의 관계 그래프를 추가 구성하였다. 원산지 정보는 품목과 지역이 각각 노드가 되고 그것이 원산지 정보임을 나타내는 엣지로 연결되어 있다. 영양소 정보는 품목과 영양성분이 노드를 형성하고 영양소 정보임을 나타내는 엣지가 함량 정보와 함께 연결되어 있다.



[그림 1] 식품산업 분야의 품목, 원산지, 영양성분 등의 관계 그래프 구축 예시

[그림 1]은 실제 구축된 Graph RAG의 시각화 결과를 보여준다. 중앙의 밀집된 노드들은 주요 농산물 품목과 지역 간의 복잡한 관계를 나타내며, 외곽의 노드들은 상대적으로 독립적인 엔터티들을 표현한다. 이러한 그래프 구조를 통해 Multi-Hop 질의에서 필요한 관계 체인을 효과적으로 탐색할 수 있다.

또한, GraphDB의 기본 Cypher 검색은 검색어가 완벽히 일치하는 경우에만 해당 노드가 반환되는 문제가 있다. 이를 해결하기 위해 Lucene 기반의 fulltext index를 생성하여 유사어와 부분 매칭도 가능하도록 했다. 검색 결과로는 해당 노드와 인접 노드, 그리고 둘 사이의 관계를 관련도 순으로 상위 500개까지 반환하도록 설정했다.

3.3. VectorRAG와 Graph RAG의 결합

제안하는 시스템의 처리 흐름은 질의 분석 단계에서 시작된다. 입력된 질의는 우선 도메인 특화 엔터티 추출 과정을 거치며, 예를 들어 식품 도메인에서는 품목, 생산지, 영양 성분, 관련 기업 등이 주요 대상이 된다. 이때 LLM은 단순 키워드 매칭을 넘어 의미적 분석을 수행하여 잠재적 엔터티를 식별한다.

추출된 엔터티는 Graph RAG의 진입점으로 활용된다. 그래프 검색은 노드 탐색을 통해 엔터티 간의 연결 구조를 파악하고, 그 결과를 시작 노드, 도착 노드, 연결 유형, 출처 문서의 네 가지 핵심 정보로 구조화한다. 이를 통해 단순 키워드 집합이 아닌 의미 기반의 관계망이 형성된다.

본 연구의 핵심 기여는 바로 이 과정에서의 쿼리 변환 기법에 있다. 기존 접근이 키워드를 단순 추가하는 수준에 머물렀다면, 제안하는 시스템은 새로운 서술문을 생성한다. 예를 들어, "기상 이변이 농업에 미친 영향"이라는 초기 질의가 주어졌을 때, 그래프에서 *[충청-토마토, 전남-양파]*와 같은 관계가 탐지되면, 시스템은 이를 기반으로 *"충청 지역 토마토 재배와 전남 지역 양파 생산이 기상 이변으로 겪은 구체적 피해 상황과 대응 방안"*과 같이 맥락이 풍부한 질의로 변환한다.

이렇게 강화된 질의는 VectorRAG로 전달되어 문서 검색을 수행한다. 이때 그래프 정보는 세 가지 차원에서 검색 품질을 향상시킨다. 첫째, 의미의 구체화를 통해 모호한 표현이 구체적 엔터티와 관계로 치환되어 정확도가 높아진다. 둘째, 맥락의 확장을 통해 단일 키워드가 아닌 관계망 전체가 검색 범위를 결정하여 관련 문서를 놓칠 가능성을 줄인다. 셋째, 우선순위 조정을 통해 그래프 상에서 강한 연결을 보이는 엔터티 쌍이 더 높은 가중치를 받아 검색 효율을 높인다.

이와 같은 통합적 접근은 특히 복잡한 도메인 질의에서 강점을 보인다. 단순 사실 확인을 넘어 여러 개체 간의 관계와 영향을 파악해야 하는 상황에서 그래프는 구조적 관계를, 벡터는 세부적 내용을 제공함으로써 상호 보완적 시너지를 극대화한다.

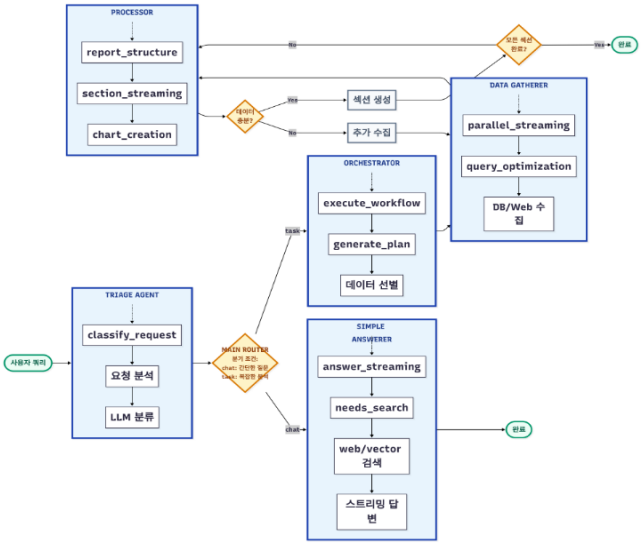
3.4 Vector/Graph RAG 결합 기반 적응형 멀티에이

2) https://www.atfis.or.kr/home/index.do

전트 검색 시스템

본 연구에서 제안하는 시스템은 AFLOW(Adaptive Flow) 기반 멀티에이전트 아키텍처[9]를 통해 Multi-Hop 질의를 효율적으로 처리한다. 시스템은 질의 복잡도와 특성을 분석하여 Vector RAG와 Graph RAG를 동적으로 선택하고 결합한다. 전체 시스템 구성도는 [그림 2]와 같고, 시스템은 다음과 같은 역할별 에이전트로 구성된다:

- TriageAgent: 질의 복잡도를 판단하여 단순/복잡 질의를 분류하는 초기 라우팅 담당
- OrchestratorAgent: Multi-Hop 질의를 단계별로 분해하고 실행 계획을 수립하는 중앙 조정자
- DataGathererAgent: 각 도구별 쿼리 최적화 및 병렬 검색을 실행하는 데이터 수집 담당
- ProcessorAgent: 수집된 데이터를 통합하여 최종 응답을 생성하는 후처리 담당



[그림 2]. 시스템 구조도

OrchestratorAgent의 동적 실행 계획 수립 및 사전점검(Pre-Check) 메커니즘

OrchestratorAgent는 Multi-Hop 질의를 처리 가능한 단계로 분해하고 각 단계별 최적 도구를 선택한다. 질의 분해는 먼저 LLM을 통해 사용자의 질의 의도를 파악하여 복잡한 질의를 단일 태스크로 수행 가능한 질의들로 분해한 후, 각 태스크 간 의존성을 분석하여 순차적 처리가 필요한 부분과 독립적으로 실행 가능한 부분을 식별한다.

사전점검(Pre-Check) 메커니즘은 본격적인 계획 수립 전에 GraphDB에서 관련 데이터의 존재 여부를 빠르게 확인하는 과정이다. 그래프 검색을 통해 반환된 텍스트 보고서를 패턴 분석하여 원산지 관계, 영양성분 관계, 문서 관계의 존재 여부를 각각 불린 값으로 판단한다. 이 사전점검 결과는 OrchestratorAgent의 generate_plan 메서드에서 프롬프트에 직접 주입되어, 계획 수립 과정에서 실제 데이터 존재 여부를 바탕으로 최적화된 실행 계획을 생성할 수 있도록 한다. 예를 들어 원산지 관계가 존재한다면 관련 질의에 GraphDB 검색을 우선 활용하고, 영양성분 관계가 존재한다면 정량적 수치는 RDB(Relational Database)에서, 영양소 연결관계는 GraphDB에서 처리하도록 도구 선택을 최적화한다. 이를 통해 불필요한 검색 단계를 미리 제거하고 효율적인 멀티에이전트 워크플로우를 구성한다.

DataGatherer와 교차참조

DataGathererAgent는 Graph RAG 검색 결과에서 추출된 엔터티 관계 정보를 Vector RAG의 쿼리 정제에 활용하는 교차 참조 메커니즘으로 구현된다. Graph RAG의 텍스트 보고서 메타정보 중 "문서 관계 정보" 필드를 파싱하여 <소스 엔터티, 타겟 엔터티, 관계 타입, 관계가 추출된 문서명>과 같은 구조화된 정보를 추출한다.

이렇게 추출된 관계 정보는 딕셔너리 형태로 구조화되어 Vector RAG 쿼리 최적화에 활용된다. 쿼리 정제 과정은 LLM을 통해 수행되며, 원본 사용자 질의와 Graph RAG에서 추출한 관계 정보를 결합하여 보다 구체적이고 맥락이 풍부한 검색 쿼리를 생성한다.

플레이스홀더 메커니즘 기반 컨텍스트 전파

Multi-Hop 처리의 핵심은 이전 단계 결과를 다음 단계 입력으로 활용하는 것이다. 시스템은 [step-X의 결과] 플레이스홀더를 실제 컨텍스트로 치환하여 의존성을 처리한다.

- Step1: 질의 Q_1 실행 → 결과 R_1
- Step2: 질의 Q_2 + 컨텍스트 $C_1(R_1)$ → 결과 R_2
- Step3: 질의 Q_3 + 컨텍스트 $C_2(R_1, R_2)$ → 결과 R_3
- Step 4: 질의 Q_4 + 컨텍스트 $C_3(R_1, R_2, R_3)$ → 결과 R_4

예를 들어, "2025년 7-8월 집중호우가 피해지역 농산물 가격에 미친 영향" 질의는 다음과 같이 처리된다.

- Step 1: [Web Search]
  Q_1("2025년 7-8월 집중호우 피해지역") → R_1(강원, 호남)
- Step 2: [Graph RAG]
  Q_2("[step-1 결과] 지역의 주요 농산물") + C_1(강원, 호남) → R_2(강원-배, 호남-포도)
- Step 3: [Vector RAG]
  Q_3("[step-2 결과] 품목의 생산 피해 현황") + C_2(강원-배, 호남-포도) → R_3(피해 현황 문서)
- Step 4: [RDB]
  Q_4("[step-3 결과] 품목의 가격 변동") + C_3(배, 포도 피해 데이터) → R_4(시계열 가격 데이터)

독립적인 Multi-Hop 경로를 식별하여 병렬로 처리함으로써 시스템의 성능을 최적화한다. 의존성이 없는 작업들은 병렬 실행 그룹으로 구성하고, 의존성이 있는 작업들은 순차 실행되도록 Step별로 나누어 계획한다.

4. Multi-Hop 질의 처리 사례 분석

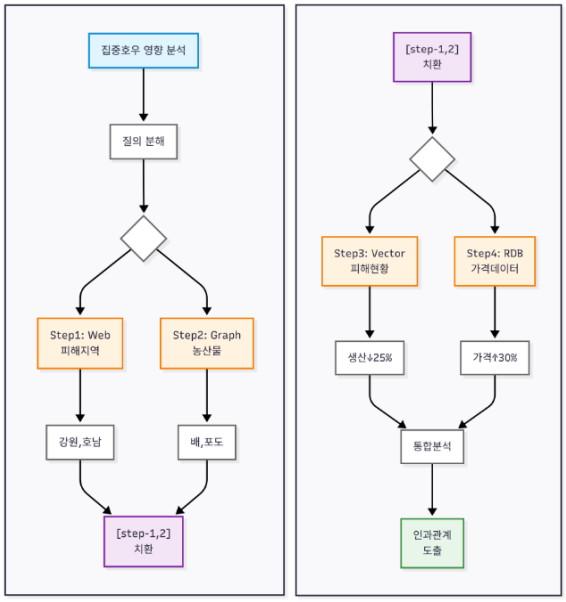
본 절에서는 제안한 시스템이 실제 Multi-Hop 질의를 어떻게 처리하는지 구체적인 사례를 통해 설명한다.

4.1 2-Hop 질의 - 관계 추적

[그림 3] 2-Hop 질의 처리 흐름 예: 제주 특산품 수출국 탐색

[그림 3]은 2-Hop 질의 처리 과정을 보여준다. 첫 번째 Hop에서 GraphRAG를 통해 제주도 특산품인 감귤의 생산 관계를 파악한다. 두 번째 Hop에서는 감귤의 수출 관계를 추적하고, 이를 통해 최종적으로 중국, 일본, 미국이라는 정보를 얻게 된다. 이로서, 최종적인 보고서 답변을 생성하게 된다. 각 단계에서 추출된 관계 정보가 다음 단계의 쿼리 생성에 활용되는 것을 알 수 있다.

4.2 4-Hop 복합 질의 - 인과관계 체인



[그림 4] 4-Hop 복합질의처리 및 플레이스홀더 기반 컨텍스트 전파
질의: "2025년 7-8월 집중호우가 피해지역 농산물 가격에 미친 영향"

[그림 4]는 복잡한 4-Hop 질의가 어떻게 단계별로 분해되고, 각 단계의 결과가 플레이스홀더 메커니즘을 통해 다음 단계로 전파되는 과정을 시각화한 것이다. Web Search, Graph Search, Vector Search, RDB가 순차적으로 활용되며, [step-X의 결과] 플레이스홀더가 실제 컨텍스트로 치환되는 과정을 통해 정보 손실 없는 최종답변을 얻을 수 있다.

5. 실험 및 평가

본 연구의 실험은 식품산업 종사자를 위한 리서치 에이전트 시스템을 Docker 기반의 Multi-Hop RAG 시스템으로 구축하고 성능을 평가하였다.

5.1 실험 설정

시스템은 FastAPI, Elasticsearch 8.11.0, Neo4j 5.0로 구성되었으며, LLM으로는 Gemini API(2.5 pro/flash/flash-lite)를 사용하였다. 성능 측정은 HTTP API를 통한 실시간 로그 분석과 모니터링으로 수행되어 실제 운영 환경과 동일한 조건에서의 평가가 가능했다.

5.2 실험 설계 및 데이터셋

총 200개의 테스트 질의를 구성하여 실험을 진행하였다. Graph RAG를 적용한 시스템의 성능을 평가하기 위해, Graph-to-Vector(활성/비활성), Precheck(활성/비활성) 총 4가지의 경우로 테스트 질의를 50개씩 나누고 진행하였다. 또한, 50개 질의는 2-Hop 17개, 3-Hop 17개, 4-Hop 16개로 구성되어 Hop 복잡도별 성능을 균형있게 평가할 수 있도록 하였다.

5.3 실험결과

지표는 평균 응답 시간(구조생성시간), 평균 재검색 횟수, 평균 생성 계획 Step, 검색 정확도로 구성되어 있으며, 이를 통해 시스템의 처리 속도와 검색 품질을 4가지 Graph 기능 조합에 따라 테스트하여 각각의 효과를 자세히 분석하였다.

Method	Test Count	Avg Response Time (s)	Avg Search Count	Avg Plan Steps
Graph-to-Vector ON + Pre-check OFF	50	115.94	7.2	2.3
Graph-to-Vector OFF + Pre-check OFF	50	116.66	7.2	2.3
Graph-to-Vector ON + Pre-check ON	50	113.63	7.1	2.4
Graph-to-Vector OFF + Pre-check ON	50	123.66	7.3	2.4

[표 1] Graph-to-Vector 및 Pre-check 조합별 시스템 성능 비교 결과
각 조건에서 평균 응답 시간, 평균 검색 횟수, 평균 계획 단계 수를 비교한 결과

Method	Re-search Rate (%)	Avg Re-search Count	Total Re-search Events
Graph-to-Vector ON + Pre-check OFF	54.0	0.54	27
Graph-to-Vector OFF + Pre-check OFF	40.0	0.40	20
Graph-to-Vector ON + Pre-check ON	38.0	0.38	19
Graph-to-Vector OFF + Pre-check ON	44.0	0.44	22

[표 2] Graph-to-Vector 및 Pre-check 조합별 재검색(Re-search) 지표 비교
각 조건별 재검색률, 평균 재검색 횟수, 총 재검색 발생 건수

5.4 결과 분석

Graph-to-Vector 메커니즘 활성화 시 평균 응답 시간이 114.78초로 비활성화 시 120.16초 대비 4.5% 개선되었다. 재검색 발생률은 Graph-to-Vector 단독 활성화 시 54.0%로 높았으나, Pre-check와 함께 활성화 시 38.0%로 크게 감소했다. 이는 초기 검색 정확도 향상으로 추가 데이터 수집 필요성이 줄어들었음을 의미한다. 다만 쿼리 재생성 품질 평가에서는 한계가 드러났다. 2점 척도 평가 결과 대부분 1점 수준에 머물렀으며, 재생성 메커니즘을 일부 보완할 필요성이 있음을 확인할 수 있다.

Pre-check 메커니즘은 평균 응답 시간이 118.65초로 비활성화 대비 2.0% 증가하는 트레이드오프를 보였다. 그러나 평균 계획 단계 수치가 2.3에서 2.4로 증가하여 체계적 질의 분해가 이루어졌고, 재검색 횟수는 0.47회에서 0.41회로 감소했음을 알 수 있다. 문서 검색 성능은 Pre-check OFF 0.2035, ON 0.2168로 개선되었으며, 특히 Graph-to-Vector OFF + Pre-check ON 조합이 0.2372로 최고 성능을 기록했다.

Hop 수준별 분석에서는 2-Hop 질의가 속도 향상이 두드러지는 반면, 3-Hop 이상 질의의 경우 속도 향상보다 품질 개선이 뚜렷했다. 특히 4-Hop 질의의 재검색 발생률이 60% 이상으로 높은 것을 보면, 복잡한 질의일수록 초기 검색만으로는 충분한 정보 수집이 어려움을 알 수 있고, Hop 수가 증가할수록 많은 구조적 정보를 획득하기 때문에 처리 시간이 증가하는 트레이드오프가 확인되었다.

최적 조합은 Graph-to-Vector와 Pre-check 모두 활성화된 경우로, 113.63초 응답 시간, 100% 성공률, 38.0% 재검색 발생률로 최고 성능을 기록했다. 이는 Pre-check의 오버헤드를 Graph-to-Vector의 최적화가 상쇄하면서도 재검색 발생률이 줄어들어 성능 향상이 이루어진 것으로 해석된다.

6. 결론

본 연구에서는 Multi-Hop 질의를 처리하기 위해 Vector RAG와 Graph RAG의 결합 시스템을 제안했다. 이와 더불어, 플레이스홀더 기반의 컨텍스트 전파 메커니즘을 적용하여 긴 추론에서도 정보의 손실을 최소화하였으며, Graph-to-Vector와 Pre-check 메커니즘 조합이 113.63초 응답 시간, 100% 성공률, 38.0% 재검색 발생률로 최적 성능을 달성했다.

또한, 실험을 통해 Graph RAG의 관계 탐색과 Vector RAG 검색을 결합한 시스템의 효과를 확인하였다. 다만, 재생성된 쿼리의 품질에서 일부 개선 사항이 확인되었고, 향후 2단계 쿼리 재생성 시스템의 도입이 필요하다. 또한, 낮은 Hop에서는 속도를, 높은 Hop에서는 품질을 향상시킬 수 있도록 추가적인 최적화가 필요하다.

본 연구는 복잡한 Multi-Hop 질의가 빈번한 식품 리서치 도메인에서 Graph RAG를 적용한 새로운 시스템을 제시하였으며, 타 도메인으로의 확장가능성을 시사한다. 향후 자동화된 평가 메트릭을 보완하고, 강화학습 기반의 적응 메커니즘을 도입하여 시스템 성능을 더욱 효과적으로 빠르게 향상시킬 수 있을 것으로 기대한다.

참고문헌

[1] Y. Gao, Y. Xiong, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang and H. Wang, “Retrieval-Augmented Generation for Large Language Models: A Survey”, arXiv preprint arXiv:2312.109972.1, 2023.

[2] A. Singh, A. Ehtesham, S. Kumar and T. T. Khoei, “ AGENTIC RETRIEVAL-AUGMENTED GENERATION: A SURVEY ON AGENTIC RAG”, arXiv preprint arXiv:2501.09136, 2025.

[3] Y. Tang and Y. Yang, “MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries”, arXiv preprint arXiv:2401.15391, 2024.

[4] N. Zhang, Z. Shang, Y. Sun, W. Liang, Z. Wei and Z. Zhang, “Credible Plan-Driven RAG Method For Multi-Hop Question Answering”, arXiv preprint arXiv:2504.16787, 2025.

[5] H. Liu, Y. Zhu, K. Zhang, Y. Zhou, H. Fan, L. Zhao and Q. Chen, “HopRAG: Multi-Hop Reasoning for Logic-Aware Retrieval-Augmented Generation”, arXiv preprint arXiv:2502.12442, 2025.

[6] M. Poliakov and N. Shvai, “Multi-Meta-RAG: Improving RAG for Multi-Hop Queries using Database Filtering with LLM-Extracted Metadata, In International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications. Cham: Springer Nature Switzerland, pp.334-342, 2024.

[7] https://huggingface.co/dragonkue/BGE-m3-ko

[8] https://huggingface.co/dragonkue/bge-reranker-v2-m3-ko

[9] J. Zhang, J. Xiang, Z. Yu, F. Teng, X. Chen, J. Chen, M. Zhuge, X. Cheng, S. Hong, J. Wang, B. Zheng, B. Liu, Y. Luo and C. Wu, “AFLOW: AUTOMATING AGENTIC WORKFLOW GENERATION”, The Thirteenth Interantional Conference on Learning Representations(ICLR 2025), 2025.