GraphRAG(그래프 검색증강생성) 실무 심층 가이드

> 유사도가 아니라 **관계**로 검색하는 RAG

---

1. GraphRAG가 풀려는 문제

표준 RAG는 "질의와 의미적으로 가까운 청크 k개"를 찾습니다. 이 구조는 **국소적 사실 조회(local factoid)** 에는 탁월하지만, 두 유형의 질문 앞에서 원리적으로 무력합니다.

① **다중 홉 추론** "이 인물과 저 사건을 연결하는 고리는 무엇인가?" — 연결 고리 자체는 어느 청크에도 명시되어 있지 않습니다. 두 개 이상의 사실을 이어붙여야 답이 나오지만, 유사도 검색은 각 사실을 독립적으로만 평가합니다.

② **전역 요약(Global Sensemaking)** "이 코퍼스 전체를 관통하는 핵심 주제 5가지는?" — 이 질문에 대응하는 특정 청크는 **존재하지 않습니다.** 답은 문서 전체에 분산되어 있으므로, top-k 검색은 정의상 도달할 수 없습니다.

Microsoft Research가 2024년 제안한 **GraphRAG**는 이 지점을 겨냥합니다. GraphRAG는 LLM을 사용해 비정형 텍스트 문서로부터 지식그래프와 요약을 생성하고, 이를 활용해 비공개 데이터셋에 대한 검색증강생성을 개선하는 기법입니다. 대규모 비공개 비정형 문서군에 대한 포괄적 전역 개관을 제공하는 동시에 세부적·국소적 정보 탐색도 가능하게 합니다.

핵심 발상은 단순합니다. **검색 전에 문서를 읽어 "누가 무엇과 어떻게 연결되는지"를 미리 구조화해 둔다.**

---

2. 인덱싱 파이프라인

GraphRAG의 비용과 능력은 대부분 인덱싱 단계에 집중되어 있습니다.

```
문서 → 텍스트 유닛 분할 → LLM 엔티티/관계 추출 → 그래프 구축
     → 커뮤니티 탐지(Leiden) → 계층적 커뮤니티 요약 생성 → 임베딩/저장
```

2.1 텍스트 유닛 분할

문서를 300~1,200 토큰 단위로 자릅니다. 이 크기가 이후 추출 밀도를 결정합니다. 작을수록 엔티티 재현율이 높아지지만 LLM 호출 횟수와 비용이 비례해 증가합니다.

2.2 엔티티·관계 추출

각 텍스트 유닛에 대해 LLM이 다음을 추출합니다.

- **엔티티**: 이름, 유형(인물/조직/장소/개념/사건 등), 설명문
- **관계**: 출발 엔티티, 도착 엔티티, 관계 설명, 강도 점수
- **주장(Claim)**: 선택적으로, 시점·주체가 붙은 사실 진술

여기서 **도메인 프롬프트 튜닝**이 결정적입니다. 기본 프롬프트는 범용 엔티티 유형을 가정하므로, 법률 코퍼스라면 `조항/판례/법인/의무`, 부동산 코퍼스라면 `필지/용도지역/거래/규제` 같은 도메인 스키마를 명시해야 그래프가 쓸모 있어집니다. Microsoft도 이 요약 생성 과정에 사람이 개입(human in the loop)하는 경우가 많으며, 어떤 종류의 엔티티와 관계를 추출할지에 대한 사용자 입력이 요약 방식을 좌우한다고 설명합니다. 인덱싱 단계에서 정의되는 사용자 페르소나에 대한 명확한 기술이 노드·엣지·커뮤니티 리포트 구조에 영향을 미칩니다.

2.3 엔티티 해소(Entity Resolution)

"서울대", "서울대학교", "SNU"는 하나의 노드여야 합니다. 문자열 정규화, 임베딩 유사도 병합, LLM 판정을 조합해 중복 노드를 통합합니다. **이 단계를 생략하면 그래프가 파편화되어 GraphRAG의 이점이 사라집니다.** 실무에서 가장 자주 발생하는 실패 지점입니다.

2.4 커뮤니티 탐지

구축된 그래프에 **Leiden 알고리즘**을 적용해 밀접하게 연결된 노드 군집(커뮤니티)을 계층적으로 찾아냅니다. 레벨 0은 큰 덩어리, 레벨이 깊어질수록 세분화된 하위 주제가 됩니다.

2.5 커뮤니티 리포트 생성

각 커뮤니티에 대해 LLM이 요약 보고서를 작성합니다. 하위 커뮤니티 요약을 재료로 상위 커뮤니티 요약을 만드는 **상향식 계층 요약**입니다. 이 리포트 집합이 곧 "코퍼스 전체에 대한 다층 지도"이며, 전역 질의에 답하는 근거가 됩니다.

이 단계가 GraphRAG의 결정적 차별점입니다. 표준 RAG에는 코퍼스 전체를 조망하는 산출물이 아예 존재하지 않습니다.

---

3. 질의 모드

GraphRAG의 쿼리 엔진은 완성된 인덱스 위에서 동작하는 검색 모듈입니다.

3.1 Local Search (국소 검색)국소 검색은 AI가 추출한 지식그래프의 관련 데이터와 원본 문서의 텍스트 청크를 결합해 답변을 생성합니다. 문서에 언급된 특정 엔티티에 대한 이해를 요구하는 질문에 적합합니다.

동작: 질의에서 엔티티를 식별 → 그래프에서 해당 노드의 이웃(관계·주장·소속 커뮤니티·원문 청크)을 수집 → 우선순위에 따라 컨텍스트 윈도우에 채워 생성.

적합 질의: "X 프로젝트의 담당자와 예산은?", "이 물질의 효능은?"

3.2 Global Search (전역 검색)전역 검색은 AI가 생성한 모든 커뮤니티 리포트를 map-reduce 방식으로 탐색해 답변을 생성합니다. 자원 소모가 큰 방법이지만, 데이터셋 전체에 대한 이해를 요구하는 질문에는 좋은 응답을 제공합니다.

동작: - **Map 단계**: 각 커뮤니티 리포트를 병렬로 읽고, 질의와 관련된 부분 답변과 중요도 점수를 생성 - **Reduce 단계**: 점수 상위 부분 답변들을 종합해 최종 답변 작성

적합 질의: "이 자료의 주요 갈등 축은?", "전체 문서에서 반복되는 리스크 유형은?"

비용 주의: 커뮤니티 수만큼 LLM 호출이 발생합니다. 커뮤니티 레벨을 낮게(상위 수준) 제한해 호출 수를 통제하는 것이 실무 표준입니다.

3.3 DRIFT SearchDRIFT(Dynamic Reasoning and Inference with Flexible Traversal) 검색은 Microsoft의 GraphRAG 기법을 확장해 전역 검색과 국소 검색의 특성을 결합하며, 계산 비용과 품질 결과의 균형을 맞추면서 상세한 응답을 생성합니다. DRIFT는 커뮤니티 정보를 검색 과정에 포함시키는 방식으로 국소 검색 질의에 접근합니다. 이는 질의 출발점의 폭을 크게 확장하여, 최종 답변에 훨씬 다양한 사실이 검색·활용되도록 합니다. 커뮤니티 통찰을 활용해 질의를 상세한 후속 질문으로 정제하는, 보다 포괄적인 국소 검색 옵션을 제공합니다.

동작 순서는 먼저 벡터 검색으로 커뮤니티 정보를 활용해 질의의 넓은 출발점을 잡고, 그 커뮤니티 통찰로 원 질의를 상세한 후속 질의들로 정제한 뒤, 지식그래프를 동적으로 순회하며 엔티티·관계·국소 정보를 검색하는 방식입니다.

실무 위치: **국소 검색의 재현율 부족과 전역 검색의 비용 과다 사이의 중간해**입니다. 대부분의 실전 질의에 대해 1순위로 검토할 모드입니다.

3.4 Basic SearchGraphRAG는 질문 유형별로 검색 결과를 비교하기 쉽도록 기본 벡터 RAG의 간단한 구현체도 포함하고 있습니다. 비교 기준선(baseline)으로 유용합니다.

---

4. 표준 RAG vs GraphRAG

항목	표준 RAG	GraphRAG
검색 단위	텍스트 청크	엔티티·관계·커뮤니티 리포트
검색 원리	벡터 유사도	그래프 순회 + 계층 요약
인덱싱 비용	낮음 (임베딩만)	**매우 높음** (문서 전량 LLM 처리)
질의 비용	낮음	중~높음 (전역 검색은 특히 높음)
국소 사실 조회	우수	우수
다중 홉 추론	취약	**강함**
전역 요약	불가	**가능**
증분 갱신	용이	어려움 (그래프·커뮤니티 재계산)
근거 추적성	청크 단위	엔티티·관계 단위로 더 정밀
구축 난이도	낮음	높음 (스키마 설계·엔티티 해소 필요)

핵심 판단 기준: 질의 유형의 분포입니다. 사내 FAQ처럼 국소 조회가 90%인 시스템에 GraphRAG를 도입하면 인덱싱 비용만 수십 배로 늘고 체감 품질은 거의 개선되지 않습니다. 반대로 조사·분석·감사·리서치처럼 "종합적 파악"이 목적인 업무에서는 표준 RAG가 원리적으로 답할 수 없는 영역을 GraphRAG가 열어줍니다.

---

5. 비용 구조와 통제 전략

GraphRAG의 최대 진입장벽은 **인덱싱 비용**입니다. 문서 전량을 LLM으로 읽어 엔티티를 추출하고, 커뮤니티마다 요약을 생성하므로 토큰 소모가 표준 RAG 대비 수십~수백 배에 달할 수 있습니다.

실무 통제 방안

1. **추출 모델 분리** — 엔티티 추출은 저비용·고속 모델, 커뮤니티 요약과 최종 생성은 고성능 모델로 이원화
2. **커뮤니티 레벨 제한** — 전역 검색 시 상위 1~2개 레벨만 사용
3. **선별적 그래프화** — 코퍼스 전체가 아니라 핵심 문서군만 그래프화하고, 나머지는 표준 RAG로 처리
4. **파일럿 스코프** — 전체 구축 전 문서 100~300건으로 프롬프트·스키마를 먼저 수렴시킬 것
5. **경량 구현 검토** — LightRAG, nano-graphrag 등 경량 대안은 인덱싱 비용이 현저히 낮고 증분 갱신을 지원합니다

---

6. 생태계 지형

구현체	특징
microsoft/graphrag	레퍼런스 구현. Local/Global/DRIFT/Basic 4개 모드 제공
LightRAG	이중 수준(저수준 키워드 + 고수준 개념) 검색, 증분 갱신 지원, 저비용
nano-graphrag	최소 구현. 학습·프로토타이핑용
Neo4j + LLM	성숙한 그래프 DB에 Cypher 질의 결합. 운영 안정성·시각화 우수
HippoRAG	해마 기억 이론 착안, Personalized PageRank 기반 다중 홉
LlamaIndex / LangChain 그래프 모듈	기존 RAG 스택에 그래프 계층을 부분 도입

의사결정 팁: 이미 Neo4j 등 그래프 자산이 있고 스키마가 명확한 정형 도메인이라면 Neo4j + Cypher + LLM 조합이 microsoft/graphrag보다 실

용적인 경우가 많습니다. 반대로 스키마 없이 비정형 텍스트 더미에서 출발해야 한다면 GraphRAG의 자동 추출이 유일한 현실적 경로입니다.

---

7. 하이브리드 전략 (실무 권장안)

양자택일이 아니라 **질의 라우팅**이 정답에 가깝습니다.

```
질의 입력
  ↓
[질의 분류기]
  ├─ 단일 사실 조회       → 표준 RAG (저비용·저지연)
  ├─ 엔티티 중심 상세     → GraphRAG Local
  ├─ 다중 홉 연결         → GraphRAG DRIFT
  └─ 전역 주제·요약       → GraphRAG Global
```

이 구조는 비용의 대부분을 저비용 경로로 흡수하면서, 표준 RAG가 실패하는 소수 질의만 고비용 경로로 보냅니다. 실측 기준 전체 질의의 70~85%가 첫 경로로 처리되는 경우가 일반적입니다.

---

8. 도입 체크리스트

- [ ] 질의 로그 분석 — 다중 홉·전역 질의 비중이 20% 이상인가
- [ ] 도메인 엔티티/관계 스키마 정의 (인덱싱 프롬프트 튜닝)
- [ ] 파일럿 코퍼스 100~300건으로 추출 품질 육안 검수
- [ ] 엔티티 해소 규칙 수립 (동의어·약칭 사전)
- [ ] 인덱싱 토큰 비용 실측 후 전체 코퍼스 비용 추정
- [ ] 문서 갱신 주기와 재인덱싱 전략 결정
- [ ] 표준 RAG 대비 A/B 평가셋 구축 (질의 유형별 층화)
- [ ] 질의 라우팅 계층 설계
- [ ] 커뮤니티 리포트의 사실 오류 검수 절차

---

9. 정리

GraphRAG는 표준 RAG의 상위 호환이 아니라 **다른 문제를 푸는 도구**입니다.

- 표준 RAG의 질문: "이 질의와 가장 비슷한 문서는?"
- GraphRAG의 질문: "이 질의와 관련된 개체들은 서로 어떻게 연결되어 있으며, 그 연결망 전체는 무엇을 말하는가?"

전자가 **검색**이라면 후자는 **이해(sensemaking)** 입니다. 비용은 정직하게 비쌉니다. 따라서 도입 판단의 출발점은 기술 사양이 아니라 **"우리 조직의 질의 중 몇 퍼센트가 관계형 질문인가"** 라는 실측입니다. 이 수치 없이 시작하는 GraphRAG 프로젝트는 대체로 비싼 실험으로 끝납니다.

작성: 주식회사 별의문 (Stargate Corporation) · 2026-07-25