AI/ML

Graph RAG가 벡터 RAG보다 나은 순간과 아닌 순간

방동걸 (주)웨이스 대표이사

2026년 8월 19일

RAG를 붙였는데 답이 신통치 않을 때 나오는 흔한 처방이 "그래프로 바꿔보자"입니다. 그런데 그래프로 옮기고도 그대로인 경우가 적지 않습니다. 검색 방식이 문제가 아니었기 때문입니다. 웨이스가 VEXPLOR 안에서 제조 데이터에 RAG를 붙이며 그 차이를 직접 확인한 뒤에 적은 글입니다.

두 방식은 성능의 위아래 관계가 아니라 잘하는 질문이 다릅니다. 어떤 질문에서 벡터 검색이 구조적으로 무너지는지, 반대로 그래프가 비용만 늘리는 경우는 언제인지, 그리고 옮기기 전에 무엇을 먼저 확인해야 하는지 정리했습니다.



검색 단위가 다르면 이길 수 있는 질문도 다릅니다.

---

1. 검색 단위가 다르다 — 여기서 모든 차이가 나온다

벡터 RAG는 문서를 조각(청크)으로 잘라 임베딩하고, 질문도 임베딩해서 가까운 조각 몇 개를 가져옵니다. 검색의 단위는 **텍스트 조각**이고, 판단 기준은 **의미적 유사도**입니다.

Graph RAG는 문서에서 개체와 관계를 뽑아 그래프로 색인하고, 질문을 그래프 순회로 바꿔 답을 만듭니다. 검색의 단위는 **노드와 엣지**이고, 판단 기준은 **연결성**입니다.

한 문장으로 줄이면 이렇습니다. **벡터는 비슷한 것을 찾고, 그래프는 이어진 것을 찾습니다.** 비슷한 것과 이어진 것은 자주 일치하지만 항상 그렇지는 않고, 어긋나는 지점이 바로 두 방식의 성능이 갈리는 지점입니다.

---

2. 벡터 검색이 구조적으로 무너지는 질문 4가지

성능 튜닝으로 메워지지 않는, 방식 자체의 한계입니다.

(1) 여러 문서를 건너야 답이 나오는 질문

"이 부품을 납품한 협력사가 공급하는 다른 품목 중 최근 반품이 있었던 것"처럼 A→B→C를 거쳐야 답이 되는 질문입니다. 최종 답이 어느 한 조각에도 통째로 들어 있지 않으므로, 조각 유사도로는 중간 고리를 집어낼 수 없습니다. 질문 문장과 가장 비슷한 조각은 대개 첫 번째 홉 주변에 몰려 있고, 정작 답이 있는 세 번째 홉의 문서는 질문과 어휘가 거의 겹치지 않아 상위에 오르지 못합니다.

(2) 세거나 비교해야 하는 질문

"지난 분기에 두 번 이상 멈춘 설비가 몇 대인가" 같은 질문입니다. 임베딩은 세지 못합니다. 그리고 상위 k개만 가져오는 구조라 애초에 전수를 볼 수 없습니다. k를 키우면 컨텍스트가 넘치고, 넘치지 않게 줄이면 빠진 항목이 생깁니다. 어느 쪽이든 집계의 정답은 나오지 않는데, 더 나쁜 것은 **모델이 가져온 조각만으로 그럴듯한 숫자를 만들어 답한다**는 점입니다. 틀린 것도 문제지만 틀렸다는 신호가 없는 것이 더 문제입니다.

(3) 부정과 배제가 들어간 질문

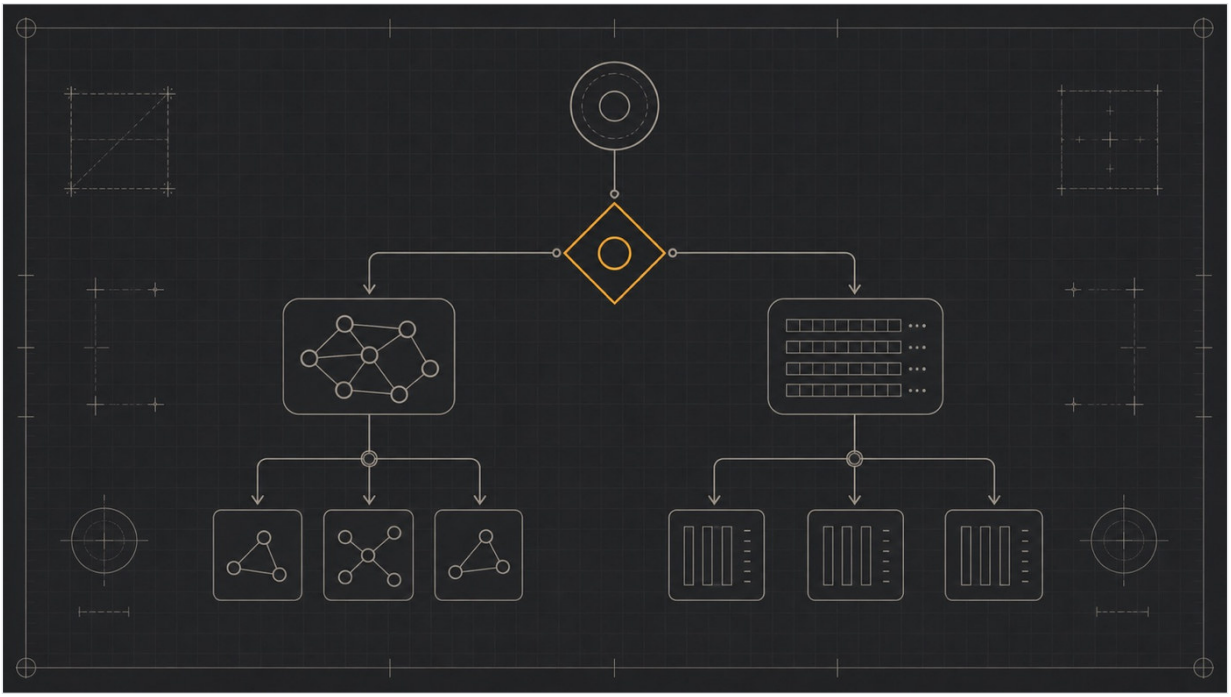
"이 첨가제를 사용하지 **않는** 공정"을 물으면 임베딩 공간에서는 "사용하는 공정"을 설명한 조각이 상위에 올라옵니다. 두 문장은 어휘가 거의 같고, 부정어 하나가 벡터를 그만큼 멀리 밀어내지 못하기 때문입니다. 사람 눈에는 정반대인 두 문장이 검색기에는 이웃입니다.

(4) 전체를 훑어야 하는 질문

"이 문서 더미 전체에서 반복해서 나타나는 쟁점은 무엇인가" 같은 질문입니다. 상위 k 검색은 본질적으로

국소적이라 전역 구조를 보지 못합니다. 마이크로소프트가 GraphRAG를 내놓으며 겨냥한 지점이 정확히 여기였습니다 — 개별 사실 조회가 아니라 말뭉치 전체에 대한 요약형 질문입니다.

이 4가지의 공통점은 **답이 문장 안이 아니라 문장들 사이의 관계에 있다**는 것입니다. 관계를 색인하지 않은 검색기가 관계를 묻는 질문에 약한 것은 튜닝의 문제가 아닙니다.



어느 쪽이 낫냐가 아니라 어떤 질문이냐가 기준입니다.

---

3. 그런데 그래프가 손해인 경우도 분명하다

반대편도 있는 그대로 봐야 합니다. 아래에 해당하면 그래프는 비용만 늘립니다.

답이 한 문단에 들어 있는 질문이 대부분일 때. 매뉴얼에서 규격값 하나를 찾는 질의가 트래픽의 대부분이라면 그래프는 오버헤드입니다. 이런 질의는 벡터에 키워드 검색을 섞고 재순위를 붙이는 것으로 충분히 올라갑니다.

초기 색인 비용. 개체와 관계 추출을 LLM으로 돌리면 문서량에 비례해 토큰 비용이 듭니다. 같은 말뭉치에 대해 임베딩만 만드는 것과 비교하면 자릿수가 달라지는 일이 흔합니다. 문서가 자주 갱신되면 이 비용이 한 번이 아니라 계속 발생합니다.

스키마가 흔들리는 영역. 무엇을 노드로 볼지, 어떤 관계를 인정할지가 아직 정해지지 않았다면 그래프는 만들 때마다 모양이 달라집니다. 그래프는 정의가 선행되는 구조물이라, 정의가 유동적이면 유지가 되지 않습니다.

서술형 지식. 왜 그렇게 판단했는지, 어떤 맥락이었는지 같은 내용은 개체-관계로 정규화하는 순간 상당 부분이 깎여 나갑니다. 이런 지식은 원문 조각째로 두고 벡터로 찾는 편이 낫습니다.

그리고 가장 중요한 한 가지. **그래프의 품질이 답의 상한입니다.** 추출이 틀리면 시스템은 벡터일 때보다 더 자신 있게 틀립니다. 그래프에서 나온 답은 "관계에 근거한 답"처럼 보여서 검토자가 덜 의심하기 때문입니다. 근거가 구조화될수록 사람의 경계심이 낮아진다는 것은, 도입할 때 계산에 넣어야 하는 비용입니다.

---

4. 옮겨보고 나서야 알게 된 3가지

문헌에 잘 안 나오는, 실제로 부딪히고 나서 알게 된 것들입니다.

(1) 진짜 비용은 그래프가 아니라 개체 해상도였다

그래프를 그리는 작업 자체는 어렵지 않습니다. 어려운 것은 **"이 둘이 같은 대상인가"를 판정하는 일**입니다. 시스템마다 같은 것을 다르게 부르고, 표기가 미묘하게 다르고, 약어와 정식 명칭이 섞입니다. 이걸 정리하지 않고 그래프를 만들면 같은 대상이 여러 노드로 쪼개지고, 쪼개진 노드는 연결을 끊습니다. **연결이 끊긴 그래프는 그래프를 도입한 이유를 그대로 없앱니다.** 멀티홉을 잘하려고 옮겼는데 홉이 이어지지 않는 상태가 됩니다.

작업량 감각으로 말하면, 그래프 도입 일정의 대부분은 그래프 설계가 아니라 이름 정리에 들어간다고 보는 편이 실제에 가깝습니다.

(2) 병목이 검색이 아니라 질문 이해였던 적이 있다

답이 나쁘길래 검색을 그래프로 바꿨는데 결과가 거의 그대로였던 경우가 있습니다. 뜯어보니 검색은 맞는 근거를 가져오고 있었고, 질문의 의도를 다르게 알아듣고 있었습니다.

그래서 방식을 바꾸기 전에 실패 사례를 먼저 분류하기를 권합니다. 최소 3가지로 갈립니다.

- **검색 실패** — 맞는 근거를 못 가져왔다
- **생성 실패** — 근거는 맞는데 답이 틀렸다
- **이해 실패** — 질문을 다르게 알아들었다

처방이 전부 다릅니다. 그래프가 고쳐주는 것은 첫 번째뿐입니다. 두 번째는 프롬프트와 근거 제시 방식의 문제이고, 세 번째는 질의 재작성과 명확화 질문의 영역입니다. 실패 100건만 손으로 분류해봐도 어디에 투자해야 하는지가 대개 분명해집니다.

(3) 그래프 질의를 모델에게 짜게 하면 조용히 틀린다

자연어를 그래프 질의로 바꾸는 일을 모델에게 맡기면, 문법도 맞고 결과도 나오는데 의미가 다른 질의가 나옵니다. 관계 방향을 뒤집거나, 조건을 엉뚱한 노드에 걸거나, 스키마에 있는 다른 관계를 대신 사용합니다. **빈 결과는 오류로 잡히지만, 그럴듯한 잘못된 결과는 아무 데도 걸리지 않습니다.**

현실적인 완화책은 자유 생성을 줄이는 것입니다. 검증된 질의 템플릿을 만들어 두고 모델에게는 어떤 템플릿에 어떤 파라미터를 채울지만 고르게 하면, 표현력은 줄지만 조용한 오답이 크게 줄어듭니다. 자유도를 남겨야 하는 구간에는 결과 건수·타입 같은 사후 검사를 붙입니다.

---

5. 판정 체크리스트 6문항

옮길지 말지를 감으로 정하지 않기 위한 최소 점검입니다. 예/아니오로 답합니다.

1. 답을 얻으려면 여러 문서를 건너야 하는 질문이 자주 있는가
2. 세기·비교·부정이 들어간 질문이 자주 있는가
3. 같은 대상을 부르는 이름이 시스템마다 다른가
4. 무엇을 개체로 볼지에 대한 정의가 안정적인가
5. 추출된 관계가 맞는지 검토해줄 사람이 있는가
6. 들어오는 질문의 유형이 반복되는가 (템플릿화가 가능한가)

예가 4개 이상이면 그래프 도입을 검토할 만합니다. **2개 이하**라면 그래프보다 먼저 할 일이 남아 있습니다 — 키워드 검색 병행, 재순위, 청킹 재설계, 질의 재작성 쪽이 투자 대비 회수가 훨씬 빠릅니다.

3번이 예인데 5번이 아니오라면 특히 조심해야 합니다. 이름은 뒤섞여 있는데 검토할 사람이 없다는 뜻이고, 이 조합이 앞서 말한 "끊긴 그래프"로 가는 가장 흔한 경로입니다.

---

6. 그래서 실무에서는 이 순서를 권합니다

① **벡터에 키워드 검색을 섞고 재순위를 붙인 구성으로 기준선을 만듭니다.** 비교 대상 없이 그래프의 효과를 말할 수 없습니다.

② **실패 사례 100건을 손으로 분류합니다.** 검색·생성·이해 중 어디가 다수인지 봅니다.

③ **검색 실패가 다수이고 그 안에서 멀티홉·집계·부정이 큰 비중이면 그때 그래프를 얹습니다.**

④ **전면 교체가 아니라 라우팅으로 붙입니다.** 질문 유형을 먼저 분류해서 관계형 질문만 그래프로 보내고 나머지는 기존 경로로 보냅니다. 두 경로를 함께 두면 그래프 품질이 나쁠 때 시스템 전체가 같이 나빠지지 않습니다.

Graph RAG는 벡터 RAG의 다음 세대가 아니라 **다른 문제를 푸는 다른 도구**입니다. 옮기기 전에 지금 틀리는 답이 어떤 종류인지부터 세어보는 것이, 옮긴 다음에 그대로인 이유를 찾는 것보다 훨씬 저렴합니다.

참고: Microsoft Research, "From Local to Global: A Graph RAG Approach to Query-Focused Summarization"(arXiv 2404.16130) · Edge et al. GraphRAG 오픈소스 구현 · 하이브리드 검색·재순위 관련 통용 실무 문헌 — 2026-08-19 확인 기준

제조 현장의 AI 적용 기록은 wace.me

WACE(웨이스)는 공장이 스스로 판단하고 멈추지 않게 하는 제조 AI 운영체제 VEXPLOR를 만드는 회사다.

원문 blog.vexplor.com/post/when-graph-rag-beats-vector-rag
발행 2026년 8월 19일 · 방동걸 (주)웨이스 대표이사

회사 주식회사 웨이스 (WACE Inc.) · 대표이사 방동걸 · 사업자등록번호 439-87-02852
주소 (18462) 경기 화성시 동탄구 동탄영천로 150, 에이동(A동) 1702-1708호 (영천동, 실리콘앨리동탄)
연락처 전화 070-4730-0629 · contact@wace.me
홈페이지 wace.me · linkedin.com/company/wace-inc

원문 홈페이지 링크드인