기술

제조 현장 AI는 왜 자꾸 틀릴까: 검색을 네 갈래로 쪼갠 이야기

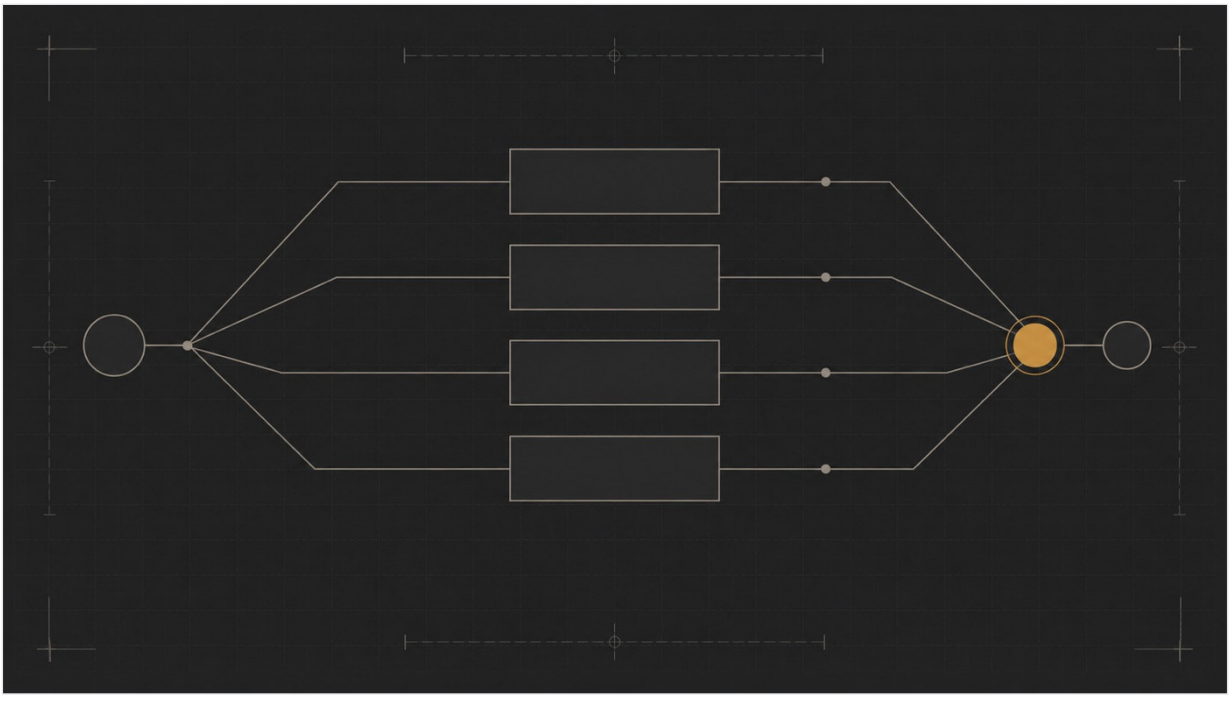
방동걸 (주)웨이스 대표이사

2026년 6월 21일

현장에 데모를 들고 가면 거의 매번 똑같은 장면이 나옵니다. 공장장님이 팔짱을 끼고 한참 보다가 이렇게 말씀하세요. "답은 잘 하네. 근데 이거 믿고 일해도 돼요?" 웨이스가 VEXPLOR를 시연하는 현장에서 직접 들은 질문이라, 그 답을 여기에 적습니다.

솔직히 이 질문이 핵심입니다. 요즘 LLM은 충분히 똑똑합니다. 문제는 똑똑함이 아니라 신뢰예요. 그리고 제조 현장에서 AI가 신뢰를 못 얻는 이유는, 대부분 모델이 모자라서가 아니라 질문의 종류가 일반 챗봇과 완전히 다른데 검색 방식은 챗봇 그대로이기 때문입니다.

오늘은 그 얘기를 좀 해보려고 합니다. 흔한 RAG가 공장에서 왜 미끄러지는지, 그리고 저희가 검색을 네 갈래로 쪼개서 어떻게 풀었는지. 좀 길어질 텐데, 끝까지 보시면 "아 그래서 숫자를 안 틀리는구나" 정도는 가져가실 수 있을 겁니다.



질문이 네 종류면 검색도 네 갈래여야 합니다.

01

일단, 흔한 RAG부터 짚고 가죠

사내 문서를 AI에 붙이는 가장 흔한 방법이 벡터 검색 기반 RAG입니다. 거창해 보이지만 원리는 간단해요. 문서를 잘게 잘라서 "의미"를 숫자 벡터로 바꿔 저장해 둡니다. 그러다 질문이 들어오면, 질문도 똑같이 벡터로 바꿔서 의미가 가까운 조각을 찾아 LLM한테 같이 던져 줍니다. "이 질문이랑 비슷한 내용이 여기 있어, 이거 보고 답해" 하는 식이죠.

매뉴얼 Q&A 같은 데서는 이게 잘 먹힙니다. "휴가 규정 알려줘" 같은 질문은 결국 "비슷한 문장 찾기"니까요.

그런데 공장에서 실제로 들어오는 질문을 며칠만 모아 보면, 이게 영 딴판입니다. 한번 종류별로 줄 세워 볼게요.

- "이 불량이랑 엮인 설비·자재·LOT이 뭐야?" — 점에서 점으로 관계를 타고 가야 답이 나오는 질문
- "지난주 불량률 추이랑 설비별 가동률 좀" — 정확한 숫자를 집계해야 하는 질문
- "이 자재 떨어졌는데 대신 뭐 써?" — 무엇이 무엇을 대체하는지 정해진 규칙이 있어야 하는 질문
- "작업표준서 3.2절에 뭐라고 돼 있었지?" — 문서 원문을 찾아 그대로 인용해야 하는 질문

벡터 검색이 자신 있게 답할 수 있는 건 사실 마지막 하나뿐입니다. 나머지 세 개에서는 어떻게 되냐면, 못 한다고 솔직하게 말하면 차라리 다행인데 그러질 않아요. 티 안 나게 빗나갑니다. 관계는 한두 칸 따라가다 길을 잃고, 집계 숫자는 그럴듯하게 지어내고(이게 그 악명 높은 수치 환각입니다), 대체 품목은 정의가 없으니 결국 "아마 이거일 거예요" 하는 추측이 됩니다.

사무실에서야 추측이 틀려도 오타 수준이지만, 공장은 다릅니다. 여기서 추측이 틀리면 불량이 나고, 라인이 서고, 심하면 리콜로 이어져요. "그럴듯한 답"의 비용이 자릿수가 다른 곳입니다. 그러니 공장장님이 팔짱을 끼는 게 당연하죠.

02

질문이 네 종류면, 검색도 네 갈래로 가자

여기서 저희가 내린 결론은 단순했습니다. 질문이 네 종류면 검색기도 네 개를 두자. 한 가지 검색으로 다 풀려고 욕심내지 말고.

그래서 VEXPLOR는 질문이 들어오면 네 개의 경로를 동시에 돌립니다. 각자 자기가 제일 잘하는 질문 유형 하나씩을 맡고, 한 경로가 비는 곳을 옆 경로가 메우는 구조예요.

경로	잘하는 일	현장 예시
Graph	엔티티 사이 관계 추론	불량 → 설비 → 자재 → LOT 연결 추적
Vector	의미가 비슷한 것 찾기	비슷한 증상·비슷한 작업표준 검색
SQL	정확한 수치·집계	불량률·가동률·재고 같은 사실값
Doc	비정형 문서 원문 근거	작업표준서·매뉴얼 원문 인용

이 중에서 제가 제일 강조하고 싶은 설계 결정이 하나 있습니다. 숫자를 LLM한테 절대 안 맡긴다는 거예요.

"지난주 불량률 몇 퍼센트야?"라고 물으면, 보통의 AI는 학습한 기억을 더듬어 그럴듯한 숫자를 만들어 냅니다. 그게 환각의 출발점이죠. VEXPLOR에서는 이 질문이 들어오면 SQL 경로가 정형 데이터에서 실제 값을 조회해 옵니다. LLM은 그 값을 받아 문장으로 풀어 줄 뿐, 숫자 자체는 손대지 않아요. "조심해서 답하기"로 환각을 막는 게 아니라, 숫자를 만들 권한 자체를 모델한테서 뺏어 버린 겁니다. 같은 식으로 Graph는 연결된 답을, Doc은 근거가 되는 원문을 가져옵니다.

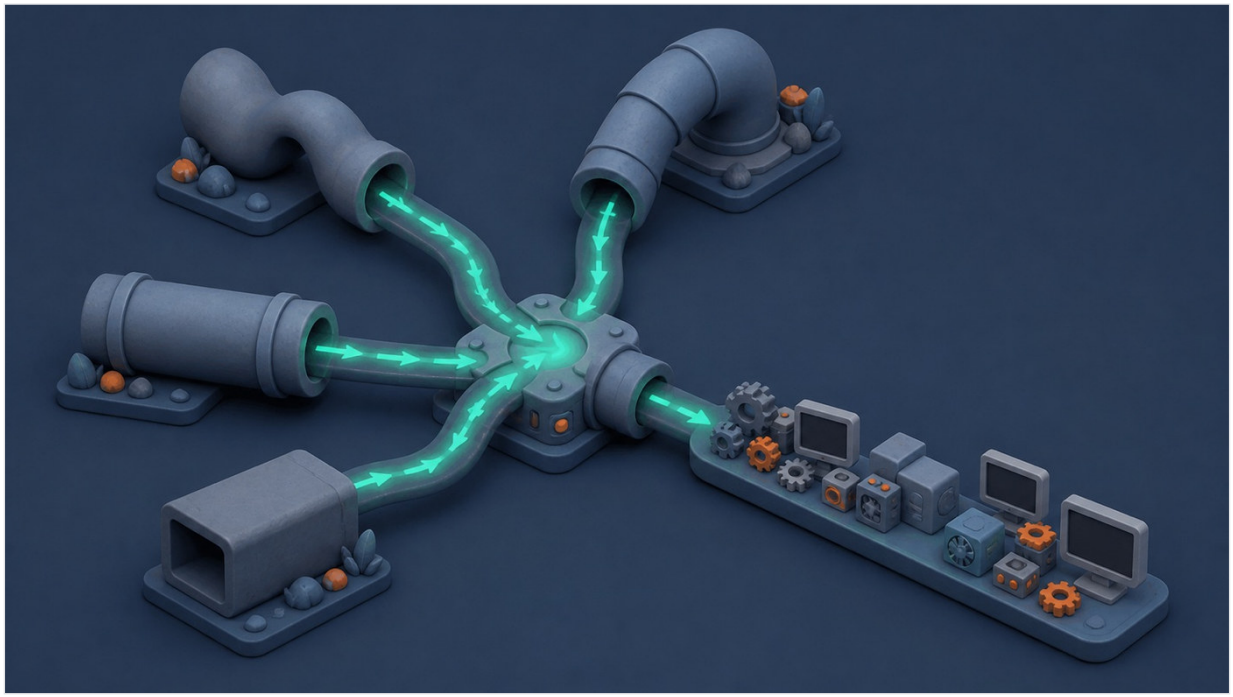
질문 하나가 네 경로를 어떻게 지나는지

말로만 하면 추상적이니, 질문 하나를 실제로 따라가 볼게요.

"어제 3호기에서 난 불량, 어떤 자재가 원인일 거 같아?"

- Graph는 3호기 불량을 출발점으로 잡고, 그 불량이 속한 LOT → 그 LOT에 투입된 자재 → 그 자재의 대체재까지 관계를 쭉 타고 내려갑니다.
- SQL은 어제 3호기 불량이 정확히 몇 건이었는지, 자재별 투입량이 얼마였는지 집계합니다.
- Doc은 그 자재와 공정의 작업표준서를 찾아옵니다.
- Vector는 과거에 비슷한 증상이 있었던 사례를 끌어옵니다.

이 네 개가 한자리에 모인 다음에야 LLM이 답을 씁니다. 만약 벡터 검색 하나만 썼다면 "예전에 비슷한 불량이 있었네요" 정도는 말해도, "정확히 어느 LOT의 어느 자재"는 영영 못 짚었을 거예요. 현장에서 필요한 건 후자인데 말이죠.



네 개의 답을 하나로 합치는 방법이 품질을 가릅니다.

03

네 개의 답, 어떻게 하나로 합치나

자, 경로가 넷이면 결과도 넷입니다. 그런데 여기서 골치 아픈 게 하나 생겨요. 네 검색기의 점수 체계가 제각각이라는 겁니다.

벡터 유사도는 0에서 1 사이 값이고, 그래프는 경로 길이로 따지고, SQL은 매칭이거나 아니거나 둘 중 하나입니다. 단위가 다 다른데 이걸 그냥 더해 버리면? 한 경로가 다른 경로를 숫자로 깔아뭉개 버립니다. 공정한 비교가 안 돼요.

이 문제를 푸는 표준 기법이 RRF(Reciprocal Rank Fusion)입니다. 저희가 발명한 게 아니라 정보검색 분야에서 오래 써 온 공개된 방법이에요. 핵심 아이디어는 점수의 절댓값을 버리고 "각 검색기에서 몇 등이었나"라는 순위만 본다는 겁니다.

```
RRF_score(d) = Σ 1 / (k + rank_i(d)) (보통 k=60)
```

수식이 좀 차갑게 생겼는데, 하는 일은 의외로 단순합니다. 어느 한 경로에서만 반짝 1등인 문서보다, 여러 경로에서 고르게 위쪽에 든 문서한테 더 높은 점수를 줘요. 단위가 전혀 다른 신호들도 "순위"라는 공통 자로 바꿔 놓으니 공평하게 섞입니다. 여러 갈래가 입을 모아 가리키는 근거가 자연스럽게 맨 위로 떠오르는 거죠.

물론 이게 끝은 아닙니다. VEXPLOR는 이 위에 제조 도메인 가중치랑 필터를 한 겹 더 얹어요. 다만 그 토대는 어디까지나 이 검증된 융합 원리입니다. 바퀴를 다시 발명하진 않았다는 얘기예요.

04

검색을 늘려도 안 풀리는 게 있다: 관계가 깔려 있어야 한다

여기까지 읽고 "그럼 검색기만 잔뜩 늘리면 장땡이네?" 싶으셨다면, 절반만 맞습니다. 사실 이게 제일 중요한 대목이에요.

Graph 경로는 따라갈 길이 미리 깔려 있어야 동작합니다. 데이터끼리 "이 불량은 이 LOT에 속한다", "이 자재는 저 자재로 대체된다" 같은 관계가 정의돼 있지 않으면, 그래프 검색은 출발점에서 한 발짝도 못 나가요. 길 없는 데서 내비게이션 켜 봐야 소용없는 것과 같습니다.

그래서 진짜 근본 레버는 검색이 아니라 온톨로지, 그러니까 관계의 표준화입니다. VEXPLOR는 제조 도메인에 311개가 넘는 비즈니스 관계를 미리 정의해 뒀고, 품목 마스터 하나가 14개 솔루션을 관통하도록 설계했습니다. 그래서 "이 불량과 연결된 모든 것"을 찍어서 추측하는 게 아니라, 정의된 관계를 따라 또박또박 답할 수 있어요.

저희가 입버릇처럼 "표준화 없이 AI 없다"고 말하는데, 그 기술적 실체가 정확히 이겁니다. 관계가 표준으로 깔려 있어야 AI가 환각 없이 연결된 추론을 한다는 것.

이 관계망은 9개 도메인 온톨로지로 짜여 있고, CDC(Change Data Capture) 파이프라인을 통해 Neo4j에 실시간으로 반영됩니다. 쉽게 말해 현장에서 데이터가 바뀌면 그래프도 곧바로 따라 바뀐다는 거예요. 덕분에 AI가 어제의 공장이 아니라 지금 이 순간의 공장을 기준으로 답합니다. 이게 은근히 중요해요. 공장은 분 단위로 바뀌니까요.

05

그래서 뭐가 어려운데? (그리고 왜 베끼기 힘든데?)

4-Way RAG의 진짜 난도는 LLM 고르는 데 있지 않습니다. 그건 솔직히 제일 쉬운 부분이에요. 어려운 건 전부 그 아래 기반에 있습니다.

834개가 넘는 표준 테이블과 311개가 넘는 관계 위에서 검색기 네 개를 일관되게 돌리고, 그래프를 실시간으로 살아 있게 유지하고, 제조 맥락에 맞게 결과를 융합하고 걸러 내는 일. 이건 한 번의 PoC로 뚝딱 끝나는 게 아닙니다. 그리고 바로 이 표준화 자산 자체가 남들이 쉽게 못 따라오는 진입장벽이 돼요. 모델은 사 오면 되지만, 이 기반은 사 올 데가 없거든요.

그리고 솔직히 아직 못 푼 재밌는 문제도 잔뜩 남아 있습니다. 경로별 신뢰도를 학습으로 가중하는 법, 온톨로지 일관성을 자동으로 검증하는 법, 그래프랑 SQL 조인을 빠르게 묶는 법, 도메인에 알아서 적응하는

검색을 만드는 법 같은 것들이요. 제조 데이터라는 까다롭고 지저분하고 그래서 진짜인 무대에서 검색·그래프·LLM을 엮어 "현장에서 안 틀리는 AI"를 만드는 일이라, 풀고 나면 보람이 꽤 쏠쏠합니다.

06

같이 만들 사람을 찾습니다

VEXPLOR는 제조 AI OS를 만드는 (주)웨이스의 제품입니다. 위에 늘어놓은 문제들을 같이 붙잡고 풀 분을 찾고 있어요. 위 얘기 읽으면서 "어 이거 재밌겠는데" 싶으셨다면, 아마 잘 맞으실 겁니다.

- 관심 분야: Hybrid RAG / 지식그래프(Neo4j) / 제조 온톨로지 / NL-to-SQL / MLOps
- 더 알아보기: wace.me · 기술 문의 contact@wace.me

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

원문 blog.vexplor.com/post/manufacturing-4way-rag
발행 2026년 6월 21일 · 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