MSAP.ai

msap.ai

RAG 도입 실전 가이드

사내 문서에 근거해 답하는 AI의 원리와 실패 원인,
검색 정확도 개선 순서와 운영 조건



TOPIC	DATE
Enterprise RAG	2026.11

CONTENTS

Agenda

01 RAG가 필요한 이유
LLM 단독 한계 · 파인튜닝 · 긴 컨텍스트와 비교

02 RAG 구성 요소와 동작
임베딩 · 검색 방식 · 저장소 · 청킹 · 프롬프트

03 RAG 답변이 틀리는 원인
실패 유형 · 공정별 진단 · 문서 형식과 변경

04 검색 정확도를 높이는 기법
하이브리드 · 질의 변환 · 재순위 · 청크 맥락화

05 GraphRAG와 에이전틱 RAG
관계 질문 · 다단계 질문 · 질문 유형별 패턴

06 평가 · 권한 · 최신성 운영
평가 세트 · 지표 · 권한 통제 · 공격 대응 · 갱신

07 도입 사례와 MSAP.ai 의 RAG 적용
업무별 효과 · 도입 단계 · CogentAI · OpsKnow

근거

공개 기술 발표 자료 33건과 2025–2026 연구·문서 10건,
MSAP.ai 내부 기술 자료를 장마다 출처로 표시

KEY INSIGHT RAG 품질은 모델보다 문서 정비·청킹·검색 설계에서 갈림 — 개선 순서와 운영 조건을 함께 봐야 도입이 성공함

SECTION 01

RAG가 필요한 이유

LLM 단독 답변의 한계와 RAG · 파인튜닝 · 긴 컨텍스트 비교

LLM 단독 답변의 한계 6가지

KEY INSIGHT 최신·사내 정보가 없으면 LLM 은 그럴듯한 오답을 냄 — 업무 질문에는 근거 문서를 붙이는 장치가 필요함

최신 정보

학습 시점 이후의 사실을 모름

예: 이번 달 바뀐 사내 규정

사내 정보

비공개 문서·매뉴얼을 모름

예: 출장용 장비 신청 부서

권한

누가 어떤 문서를 볼 수 있는지

구분하지 못함

출처

답의 근거 문서를 제시하지 못해

검증할 방법이 없음

신선도

구버전 문서와 현행 문서를

가르지 못함

업무 상태

고객별·건별 진행 상황을

알지 못함

- 학습 데이터에 드문 지식일수록 정답률이 급락함 — 키워드 검색 결과만 붙여도 크게 보완됨

Source: 공개 기술 발표 자료 4건 종합

RAG 동작 3단계

KEY INSIGHT 모델을 바꾸지 않고 근거 문서만 바꿔 답을 고침 — 문서를 갱신하면 다음 질문부터 바로 반영됨

사용자 질문
"올해 출장비 상한은?"

→

01 RETRIEVAL
검색
질문과 가장 관련 있는
문서 조각을 사내 저장소에서
찾음

→

02 AUGMENTATION
증강
찾은 조각을 질문과 합쳐
LLM 에 넘길 프롬프트를
만듦

→

03 GENERATION
생성
LLM 이 제공된 조각을 근거로
답을 쓰고 출처 문서를
함께 붙임

RAG 없이
Q. 이번 대회 첫 경기 상대는?
A. 학습 데이터에 있던 지난 대회의 상대를 답함
사내 규정 질문에는 그럴듯한 일반론을 답함

근거 없음 · 출처 없음

RAG 적용
최신 기사·사내 문서를 먼저 찾아 근거로 사용
모델은 그대로, 문서만 바꿔 답을 고침
문서를 갱신하면 다음 질문부터 바로 반영

근거 문서 · 출처 링크 함께 제시

Source: 공개 기술 발표 자료 4건 종합 · MSAP.ai 내부 기술 자료

지식 추가는 RAG, 형식 조정은 파인튜닝

KEY INSIGHT 대화 14만 건으로 학습해도 말투만 바뀜 — 사내 지식은 문서 검색으로 반영해야 재학습 비용이 들지 않음



지식을 파인튜닝으로 넣은 사례

메신저 대화 14만 건으로 학습

→ 업무 지식 대신 말투만 따라 함

"오전 중에 할게요"

같은 답만 늘어남

비용 비교

자체 모델 학습: 막대한 비용

파인튜닝: 지식 변경 때마다 재학습

RAG: 모델 그대로, 검색만 결합

Source: 공개 기술 발표 자료 2건 종합 · MSAP.ai 내부 기술 자료

긴 컨텍스트 방식과 RAG 비교

KEY INSIGHT 입력이 길수록 중간 정보를 놓치고 비용이 커짐 — 문서가 적고 자주 안 바뀔 때만 전부 넣는 방식이 유리함

전부 넣기 방식의 한계

64k 토큰을 넘으면 정확도를 유지하는 모델이 소수

긴 컨텍스트 RAG 성능 연구

입력 중간의 정보를 놓침 (Lost in the middle) 최근 정보 편향도 함께 생김

70% 컨텍스트 한도의 약 70%를 채우면 답 품질 하락

소스코드 질의 연구: 입력을 1/20 로 줄여 품질 유지

항목	전부 넣기	RAG
코드 복잡도	낮음	높음
응답 속도	빠름	보통
유지보수	쉬움	보통
확장성	문서량에 한계	**문서가 늘어도 유지**
비용	매번 전체 토큰	**필요한 토큰만**
정확성	중간 정보 누락	**문서 수정으로 개선**
보안	권한 분리 불가	**권한 없는 문서 차단**

실무 혼합 같은 대화 스레드는 전부 넣고, 다른 자료는 RAG 로 가져옴

확장성 · 비용 · 권한 분리에서 RAG 유리

Source: 공개 기술 발표 자료 3건 종합 · Long Context RAG Performance(arXiv 2411.03538) · MSAP.ai 내부 기술 자료

RAG가 맞지 않는 업무 유형

KEY INSIGHT 즉답이 필요하고 근거가 한곳에 모인 질문에 RAG 가 강함 — 전체 맥락을 빠짐없이 모아야 하는 일은 다른 구성을 씀

사례: 10만 자 고객 인터뷰 기록에서 정보 추출

RAG 가 강한 조건	이 과제의 조건
즉답 필요	배치 처리로 충분
입력 비용 절감	비용보다 품질 우선
근거가 한곳에 모임	근거가 수천 자에 흩어짐
일부 발췌로 답 가능	누락 없이 전체를 모아야 함

RAG 로 시도했을 때 실패 원인

· 한 주제가 수천 자에 걸쳐 청크 하나에 다 들어가지 않음
· 대화라 주어·목적어가 모호해 검색이 근거를 못 찾음
· 쿼리 확장 · HyDE · 부모-자식 청크도 효과 없음

전체 정리는 나눠 읽기가 검색보다 정확함

대안: 겹침 창으로 전체를 읽음

1 1만 자 단위로 겹치게 창 분할
같은 주제로 돌아오는 대화를 한 창에 담는 크기

↓

2 창마다 모든 질문을 LLM 에 넣어 추출
고충 · 현행 운영 · 도입 조건 · 숙제

↓

3 질문별로 모아 요약
결과: 품질과 요청자 만족도 상승 · 대신 느리고 비쌈

다른 예
데이터 전체를 가로지르는 분석은
파일 첨부나 에이전트로 처리

Source: 공개 기술 발표 자료 2건 종합

SECTION 02

RAG 구성 요소와 동작

색인 · 임베딩 · 검색 · 청킹 · 프롬프트 구성 방법

RAG 파이프라인: 색인 흐름과 질의 흐름

KEY INSIGHT 답 품질은 문서를 쪼개고 저장하는 단계에서 먼저 정해짐 — 이 단계의 손실은 뒤에서 복구되지 않음



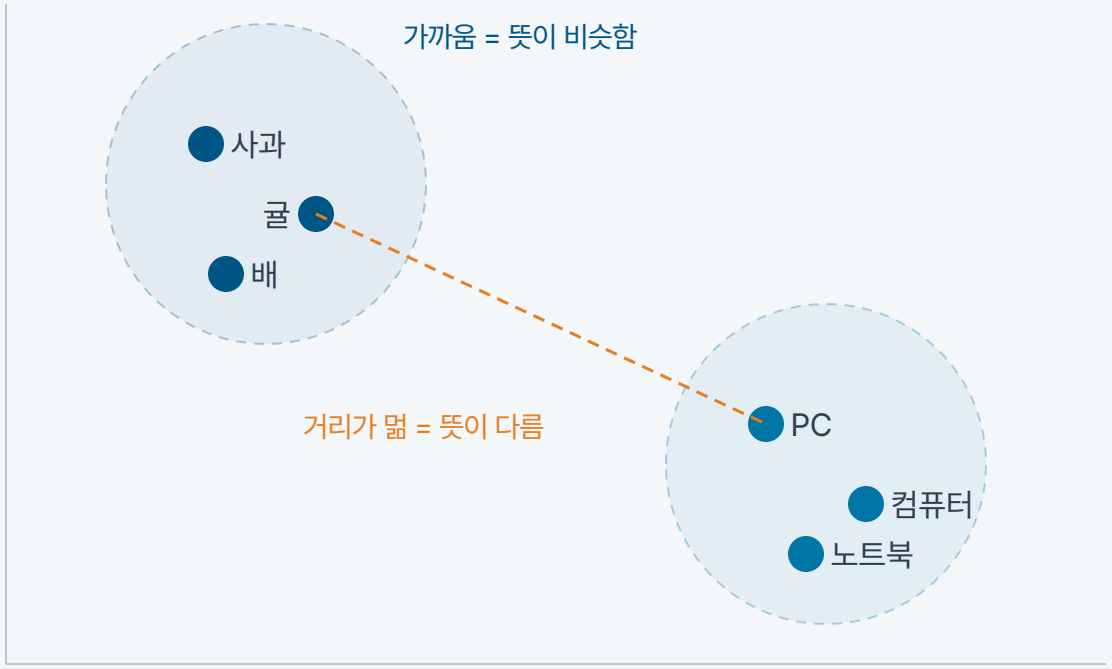
색인 흐름(수집 · 정규화 · 청킹)에서 잃은 정보는 질의 흐름의 어떤 기법으로도 되살릴 수 없음

Source: 공개 기술 발표 자료 5건 종합 · MSAP.ai 내부 기술 자료

임베딩 기반 의미 검색 원리

KEY INSIGHT 키워드가 달라도 같은 뜻이면 찾아냄 — 동의어·표기 흔들림이 많은 사내 문서 검색의 출발점임

의미 공간 (2차원으로 줄여 표시)



임베딩이란

글을 수백 차원의 숫자 벡터로 바꿈

뜻이 비슷한 글은 가까운 자리에 놓임

유사도를 재는 방법 3가지

코사인	의미 검색 · 문서 분류
유클리드 거리	수치 비교 · 추천
내적	협업 필터링

키워드 검색의 약점

"PC" 와 "컴퓨터" 를 다른 말로 봄

표기 흔들림 · 동의어가 많을수록 누락 증가

Source: 공개 기술 발표 자료 2건 종합 · MSAP.ai 내부 기술 자료

키워드 검색과 벡터 검색의 강점 비교

KEY INSIGHT 벡터는 뜻을, BM25 는 제품명·코드를 잘 찾음 — 기업 문서는 두 방식을 함께 써야 누락이 줄어듦

비교 항목	BM25 키워드 검색	벡터 검색
의미 이해	약함	**강함**
정확한 키워드	**강함**	약함
동의어 · 다른 표현	약함	**강함**
제품명 · 모델명 · 코드	**강함**	약함
속도	**빠름**	보통
결과 해석	**쉬움 (일치 단어)**	어려움 (거리 값)

키워드를 정확히 알 때
"ERR-4021 오류 조치"

기억이 흐릿해 다른 말로 찾을 때
"접속이 자꾸 끊기는 문제"

검색 엔진 4단계

1 텍스트 분석
형태소 분석 · 어간 처리

2 인덱스
역색인 · 그래프 · 양자화

3 매칭
단어 일치 · 벡터 거리

4 랭킹
점수 계산 · 순위 융합

Source: 공개 기술 발표 자료 2건 종합 · MSAP.ai 내부 기술 자료

벡터 저장소 선택 기준

KEY INSIGHT 100만 건 이하·기존 DB 운영 조직은 전용 DB 없이 시작 가능 — 규모와 운영 역량에 맞추면 비용이 줄어듦

전용 벡터DB

100만 벡터 이상
실시간 대량 검색

인메모리 구조라
비용이 큼

대규모 · 고성능

RDB 벡터 확장

기존 DB 에 벡터 타입 추가
정합성 · 보안 체계 재사용

DBA 역량을 그대로 활용
디스크 활용으로 저렴

중규모 · 기존 DB 조직

DB 내장 파이프라인

텍스트 → 청크 → 벡터 →
SQL 검색을 DB 안에서 처리

정형 테이블과 문서를
한 질의로 함께 근거화

정형 + 비정형 결합

기존 검색 API 활용

벡터DB 없이 LLM 이
검색어를 여러 개 생성

예: 메신저 검색 API +
유의어 · 날짜 필터

작게 빠르게 시작

직접 구축할 때의 난제 5가지

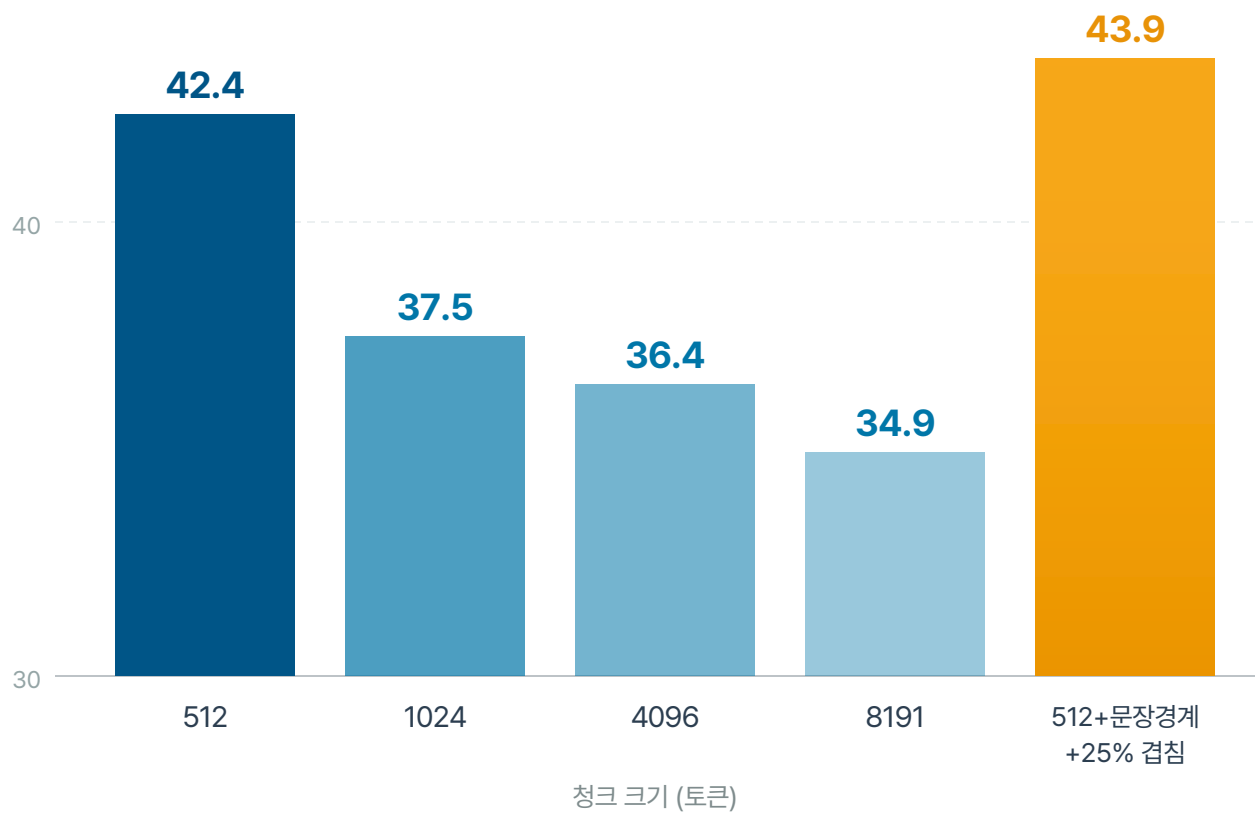
다중 데이터 소스 · 대용량 임베딩 · 증분 갱신 · 검색 확장 · 오케스트레이션 → 관리형 지식베이스 서비스가 대신 처리
공통 확인 항목: 발신자 · 날짜 같은 구조화 필터를 벡터 검색과 함께 쓸 수 있는가

Source: 공개 기술 발표 자료 6건 종합 · MSAP.ai 내부 기술 자료

청크 크기별 검색 재현율

KEY INSIGHT 너무 크면 잡음, 너무 작으면 맥락 손실 — 512 토큰 안팎에서 시작해 사내 평가 세트로 조정해야 함

청크 크기별 Recall@50 (높을수록 정답 문서를 많이 찾음)



현장 청킹 규칙

- 512 토큰에서 시작 (256 은 맥락 손실, 1024 는 잡음)
- 앞뒤 청크와 20–30% 겹침
- 조건과 결론은 같은 청크에 둠
- 조 · 항 · 제목 단위로 자름
- 문장 중간에서 자르지 않음

최종 크기는 사내 평가 세트로 조정

흔한 기본값

100 · 256 · 512 토큰 또는 500–1000자
검색 단위가 크면 관련성↑ 잡음↑,
작으면 정밀도↑ 맥락↓

Source: 공개 기술 발표 자료 4건 종합

청킹 방식 5단계

KEY INSIGHT 단계가 올라갈수록 맥락 보존이 좋아지고 처리 비용도 커짐 — 문서 구조가 분명하면 구조 기반 청킹이 비용 대비 효과가 큼

위로 갈수록 맥락 보존 ↑ · 처리 비용 ↑

청크의 정의
주제 일관성을 유지하는 최대 문장 집합



Source: 공개 기술 발표 자료 3건 종합

청크 보강 방법

KEY INSIGHT 제목·예상 질문·상위 문맥을 붙이면 짧은 조각도 맞게 검색됨 — 모델 교체 없이 정확도를 올리는 방법임

청크 제목 생성

조각마다 무엇에 관한 내용인지 한 줄
"2분기 실적 · A 사업부 매출"

후보 질문 색인

이 청크가 답할 수 있는 질문을
함께 색인 (HyQE)

부모-자식 청크

작은 청크로 찾고 큰 부모 청크를 전달
페이지 검색이 놓친 정답 위치를 찾음
(Small2Big · Auto-Merging)

청크에 덧붙이는 정보

색인 단계에서 한 번만 처리

원래 청크

"매출 3% 증가,
전 분기 대비
원가 감소"

무엇의 매출인지 알 수 없음

요약 인덱스

문서 요약으로 찾고 원문을 반환
(Multi-Vector Retriever)

메타데이터

파일명 · 페이지 · 장 제목 ·
시행일 · 접근 권한(ACL)

중복 제거

같은 내용 청크를 합쳐
상위 결과 자리 낭비를 막음

Source: 공개 기술 발표 자료 7건 종합

데이터 유형별 전처리 규칙

KEY INSIGHT 표·도표·이미지는 평문으로 바꾸면 의미를 잃음 — 형식마다 처리 규칙을 미리 정해야 검색에서 빠지지 않음

비정형 문서

PDF · 워드 · 마크다운

레이아웃 분석 파서 사용

다단 · 머리글 혼입 제거

제목 · 소제목을 청크에 포함

메일 · 메신저

HTML 태그 · 인용문 · 서명 제거
레거시 인코딩 통일
문단 → 줄 → 길이 순으로 분할

반정형 데이터

표 · 엑셀

셀 관계를 HTML · 마크다운

표로 구조화

머리행을 청크마다 유지

병합 셀을 의미 단위로 보존

단순 평문 변환 시
값이 무엇의 값인지 사라짐
→ 합계 · 건수 계산 불가

정형 데이터

DB · CSV

집계 · 계산은 SQL · BI 로

관계는 지식 그래프로

벡터 검색에 넣지 않음

복잡한 집계는 DWH,
다른 데이터와 결합은
별도 가공 후 로드

이미지 · 도표

스캔 · 차트 · 도면

캡션 생성 · OCR 로

글로 바꿔 색인

도표는 파서로

마크다운 텍스트화

오피스 문서를 PDF 로
단순 변환하면
도표 정보가 사라짐

Source: 공개 기술 발표 자료 6건 종합

답변 프롬프트 구성 4칸

KEY INSIGHT 근거가 없을 때 거절하는 규칙이 있어야 환각이 줄어듦 — 답에는 결론·근거·전제·한계를 함께 적게 함

LLM 에 넘기는 프롬프트

1 시스템 지시

역할 · 근거 문서만 쓰라는 규칙 · 근거가 없으면 "확인되지 않음"

2 사용자 질문

원문 그대로 전달

3 검색 근거

청크 본문 + 출처(문서명 · 페이지)

- 관련도 높음 · 맨 앞
- 관련도 낮음 · 가운데
- 관련도 높음 · 맨 뒤

4 출력 계약

답의 형식을 고정: 결론 · 근거 · 전제 · 한계

답변 예시

결론	국내 출장 숙박비 상한은 1박 10만 원
근거	여비 규정 제7조 (2026.03 개정)
전제	일반 직원 기준
한계	해외 출장은 별도 규정 확인 필요

예시 값은 설명용

근거가 없을 때

"확인되지 않음"으로 답하고

담당자 연결로 넘김 — 환각 방지

Source: 공개 기술 발표 자료 5건 종합

SECTION 03

RAG 답변이 틀리는 원인

현장 실패 유형 · 공정별 진단 · 문서 형식과 변경의 함정

현장 RAG 실패 유형 5가지

KEY INSIGHT 권한 누출과 구버전 답변은 검색 튜닝으로 못 고침 — 문서 관리와 권한 연동을 함께 설계해야 함

검색 문제 — 청킹 · 검색 · 재순위로 개선

엉뚱한 문서

질문과 무관한
문서를 가져옴

연차를 물었는데
경영 이념을 답함

맥락 부족

조건 · 예외가
다른 청크에 남음

예외 조항 없이
원칙만 답함

환각 · 불안정

같은 질문에도
답이 매번 다름

상여금 연 2회를
연 3회로 답함

데이터 기반 문제 — 문서 관리 · 권한 연동으로 해결

권한 밖 노출

볼 수 없는 문서가
답에 섞임

타 부서 인사 정보가
근거로 인용됨

구버전 정보

개정 전 문서가
색인에 남음

2년 전 규정으로
답함

권한 누출과 구버전 답변은 검색 알고리즘을 바꿔도 남음 — 문서 수명 관리와 원본 권한 연동을 처음부터 설계에 포함

Source: 공개 기술 발표 자료 2건 종합

5개 공정으로 실패 위치 찾기

KEY INSIGHT 상위 K개 안에 정답이 없으면 어떤 LLM 도 답하지 못함 — 답변을 고치기 전에 검색 결과부터 확인해야 함

1 질의	2 검색	3 순위	4 프롬프트	5 답변
증상				
구어 · 오타 · 모호한 질문	정답 문서가 후보에 없음	정답이 2페이지, 상위 K 밖에 있음	근거는 왔는데 LLM 이 무시함	형식 · 말투가 업무와 맞지 않음
처방				
질의 재작성	청킹 · 하이브리드 검색	재순위 · 메타 필터	근거 배치 · 지시 개선	출력 계약 · 예시

진단 순서

답변을 고치기 전에 검색 결과 로그부터 확인

상위 K 안에 정답이 있는지가 첫 질문

실험 예: 페이지 단위 단순 검색

핵심 용어가 30번 넘게 나오는 1쪽 대신

12쪽 · 26쪽을 근거로 가져옴

Source: 공개 기술 발표 자료 3건 종합 · MSAP.ai 내부 기술 자료

검수 반려 원인 3분류와 처방

KEY INSIGHT 반려의 대부분은 검색 원인 — 모델 교체 없이 지식 설계만 바꿔 검색 원인 반려가 사라짐

검색 원인

근거를 못 찾거나 엉뚱한 근거를 가져옴

처방: 코퍼스 분리 · 청크 · 하이브리드 · 재순위

최신성 원인

매일 바뀌는 값을 문서에서 찾음

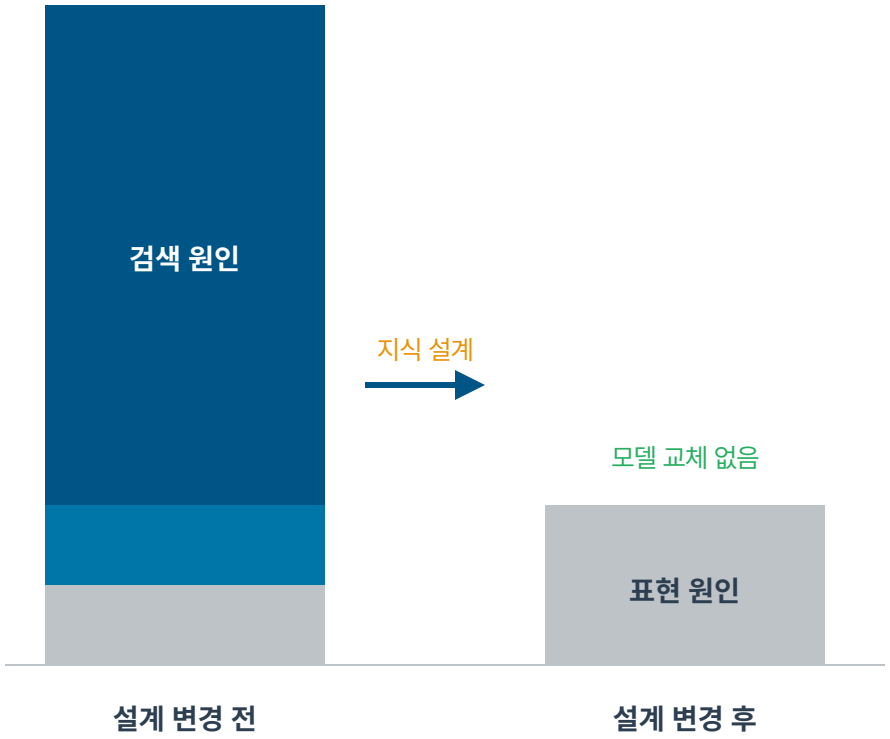
처방: 재고 · 가격 · 상태는 RAG 대신 API 로 조회

표현 원인

내용은 맞지만 말투 · 형식이 취향과 다름

처방: 지시문 · 예시(few-shot)

반려 사유 구성 변화 (개념도)



모델 교체 · 긴 컨텍스트 · 파인튜닝 등 다른 처방 10가지는 모두 한계

Source: 공개 기술 발표 자료 1건 종합

잘못 자른 청크가 만든 오답 3건

KEY INSIGHT 조건·질문·머리행이 다른 청크로 떨어지면 답이 뒤집힘 — 조건과 결론, Q 와 A, 머리행과 값은 같은 청크에 둬야 함

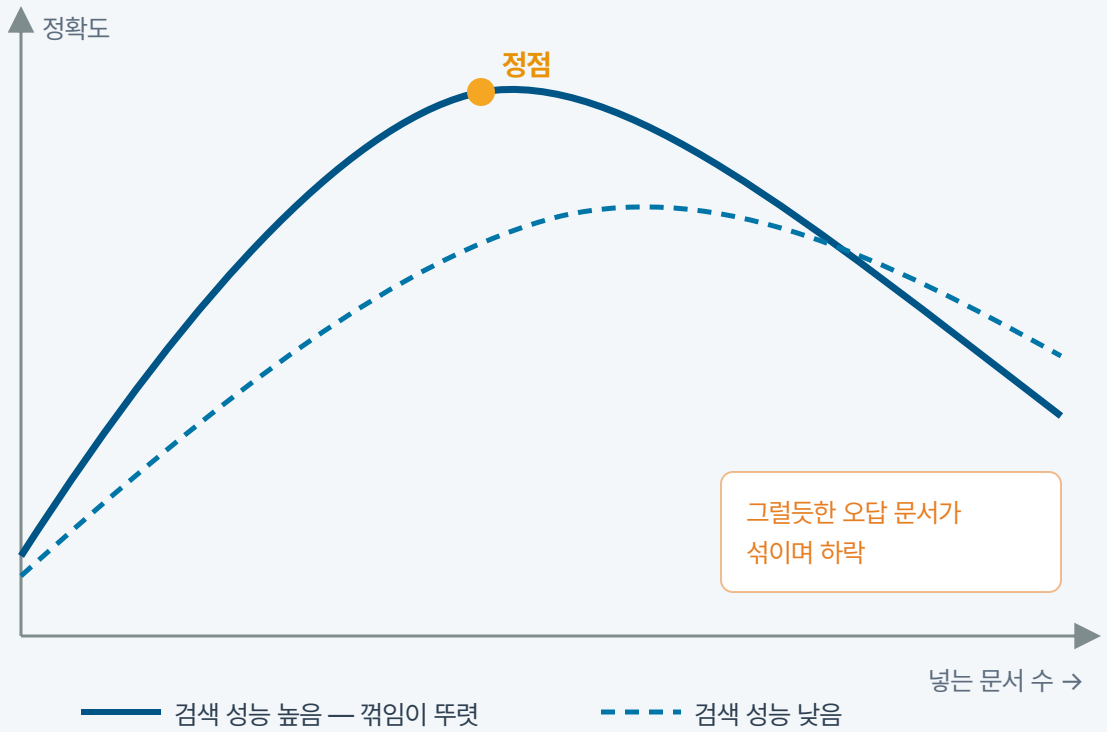
원문	잘못 자른 청크		결과
반품 규정 "구매 후 30일 이내, 미개봉 상품만 반품 가능"	청크 A "구매 후 30일 이내,"	청크 B (검색 안 됨) "미개봉 상품만 …"	개봉한 상품도 30일 안이면 반품 가능하다고 답함 처방: 조건과 결론을 같은 청크에
FAQ 엑셀 Q. 법인카드 분실 신고처? A. 경리부	Q 는 앞 청크에, 다음 행의 A 와 묶임	A "경리부" 는 다음 청크로 밀림	경리부 대신 정보시스템부라고 답함 처방: Q&A 한 쌍씩 분리해 색인
판정 표 머리행: 지표 · 기준값 · 판정 값 행: … 2,800 …	머리행 · 기준값은 앞 조각에 남음	값 "2,800" 만 검색 결과로 옴	2,800 이 무엇의 값인지 모름 합계 · 건수 계산도 불가 처방: 청크마다 머리행 유지

Source: 공개 기술 발표 자료 2건 종합 · MSAP.ai 내부 기술 자료

근거 문서 양과 정확도의 관계

KEY INSIGHT 그럴듯하지만 틀린 문서가 판단을 흐림 — 넣는 양을 늘리기보다 상위 결과의 정밀도를 높여야 함

넣는 근거 문서 양과 답 정확도 (개념도)



하드 네거티브: 그럴듯하지만 틀린 문서

- 개정 전 구버전 규정
- 국내 출장 질문에 섞인 해외 출장 규정
- 다른 부서의 세칙

무관 문서 사례: 엉뚱한 문서로 한 은행의 설립 연도를 틀리게 답함

대응

- 재순위로 상위 결과 정밀도를 높임
- 시행일 · 대상 메타데이터로 필터링
- 관련 낮은 문서는 프롬프트 가운데에 배치
- 외부 지식 + 생성 지식 결합 시
  정답률 18.56 → 50.58 (시각 질의응답 연구)

Source: 공개 기술 발표 자료 4건 종합

엑셀 · 도표 문서의 의미 소실

KEY INSIGHT RAG 품질은 검색보다 로드 단계에서 먼저 정해짐 — 레이아웃에 의미가 있는 문서는 구조를 살려 변환해야 함

원본 엑셀 업무 흐름도



평문 변환

평문으로 바꾼 결과

"영업 구매 A 접수 B 검토 C 발주
1분기 공통 일정"

- × A → B → C 순서가 단어 나열로 바뀜
- × 도형 안 텍스트는 아예 추출되지 않기도 함
- × 색으로 구분한 담당 부서 의미 소실
- × 병합 셀이 어느 행에 걸리는지 소실

업무 흐름을 물으면 엉뚱한 순서로 답함

LLM 에 넘길 때 지켜야 할 최소 의미 4요건

좌표
행 · 열 · 영역의 위치

병합
하나의 의미 단위

색상
역할 · 상태를 나타내는 정보

도형 · 화살표
단계 사이의 관계

Source: 공개 기술 발표 자료 2건 종합

엑셀 구조화 변환 4단계와 효과

KEY INSIGHT 문자 일치는 비슷해도 누락 내성이 좋아짐 — RAG 에 필요한 것은 정보를 빠뜨리지 않는 변환임

1 엑셀 원본
도형 · 차트 · 조건부 서식 접근

2 의미구조 JSON
좌표 · 병합 · 색 · 관계 보존

3 LLM 재구성
Markdown + Mermaid 흐름도

4 지식베이스 로드
흐름 질문에 순서대로 답함

벤치마크: 엑셀 문서 12건

지표	PDF 변환	의미구조 변환
문자열 완전 일치	**0.608**	0.584
누락 내성 (원문 기준)	0.909	**0.934**
누락 내성 (마크다운 기준)	0.776	**0.796**

글자 일치는 비슷하지만 정보 누락은 의미구조 변환이 적음
도형 접근에는 엑셀 실행 환경이 필요함

문서 유형별 처리 선택

단순 문서
PDF 변환으로 충분

이미지가 많은 문서
비전 모델(VLM)로 마크다운화

구조가 핵심인 엑셀
의미구조 변환 — 표를 SQL 조회 대상으로 두는 방법도 있음

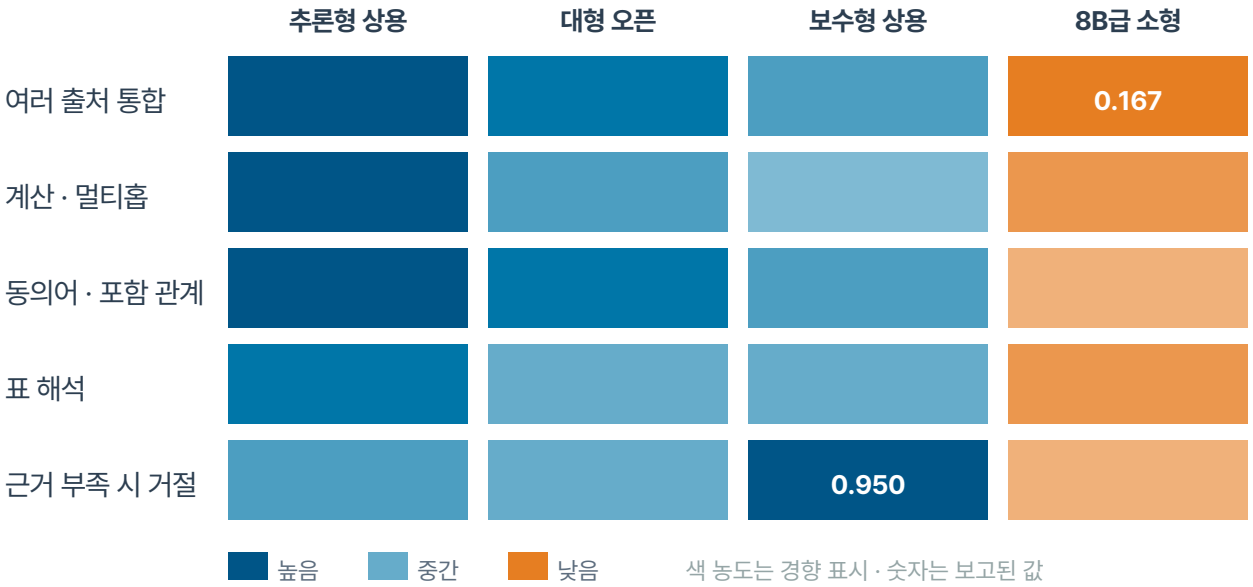
Source: 공개 기술 발표 자료 1건 종합 · MSAP.ai 내부 기술 자료

생성 모델의 근거 해석 오류

KEY INSIGHT 표 해석·계산·멀티홉에서 최신 모델도 자주 틀림 — 표 구조화 전처리와 용도별 모델 선택이 함께 필요함

생성 모델 단독 평가 설계

검색 영향을 뺀 가상 시나리오 114문항 · 청크 약 512토큰 · 비슷한 방해 청크 6–8개



종합 정확도 0.9 를 넘는 모델 없음 · 보수형 모델은 거절 1위지만
답할 수 있는 질문도 거절하는 경향

대표 오답 패턴 6가지

1 단위가 섞인 표기를 잘못 읽음

2 주석(주말 할증 등)을 빠뜨림

3 다단계 계산 중 덧셈 실수

4 중간 엔티티를 못 이어 멀티홉 실패

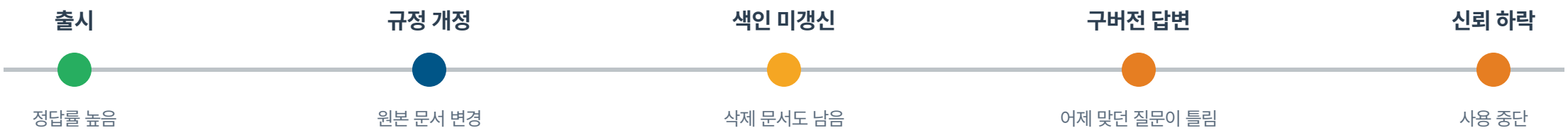
5 병합 셀 · 여러 청크로 나뉜 표 해석 실패

6 근거가 없는데 키워드에 반응해 날조

Source: 공개 기술 발표 자료 1건 종합

문서 변경이 만드는 지식 드리프트

KEY INSIGHT 문구 수정·구조 변경·청크 경계 이동만으로도 회귀가 생김 — 갱신뿐 아니라 삭제 반영까지 소스별로 확인해야 함



회귀를 만드는 변경 4가지

- 문구 소폭 변경
- 노이즈 데이터 혼입
- 문서 구조 변경
- 청크 경계 이동

즉시 재색인만으로는 부족 —
변경 뒤 회귀 평가가 필요

원본 저장소	신규 · 갱신	삭제 반영
객체 스토리지	자동	메타데이터로 논리 삭제
데이터베이스	변경 추적	변경 추적 · High Water Mark
문서 협업 도구	자동	다음 실행 때 감지
외부 저장소 바로가기	자동	**미지원 → Push API**

갱신은 감지해도 삭제는 자동 반영되지 않을 수 있음 — 부모 문서와 자식 청크의 삭제 범위까지 관리

Source: 공개 기술 발표 자료 2건 종합

SECTION 04

검색 정확도를 높이는 기법

하이브리드 검색 · 질의 변환 · 재순위 · 청크 맥락화의 효과와 순서

검색 품질 개선 4단계

KEY INSIGHT 단순한 기본기가 복잡한 기법보다 먼저 효과를 냄 —
평가 세트 없이 기법부터 넣으면 무엇이 좋아졌는지 알 수 없음

아래 단계부터 효과를 확인하며 한 칸씩 올라감

1단계
청크 개선
크기 · 경계 · 메타데이터

2단계
하이브리드 검색
BM25 + 벡터

3단계
재순위
Reranker 로 상위 재정렬

4단계
GraphRAG · 에이전틱 검색
관계 · 다단계 질문에만 선택 적용

기반: 평가 세트 (질문 · 정답 · 정답 문서)

함께 할 일

문서 정비
구조화 · 통합 · 구버전 삭제

질문 가이드 배포
좋은 질문 예시를 사용자에게 공유

순서를 거꾸로 하면

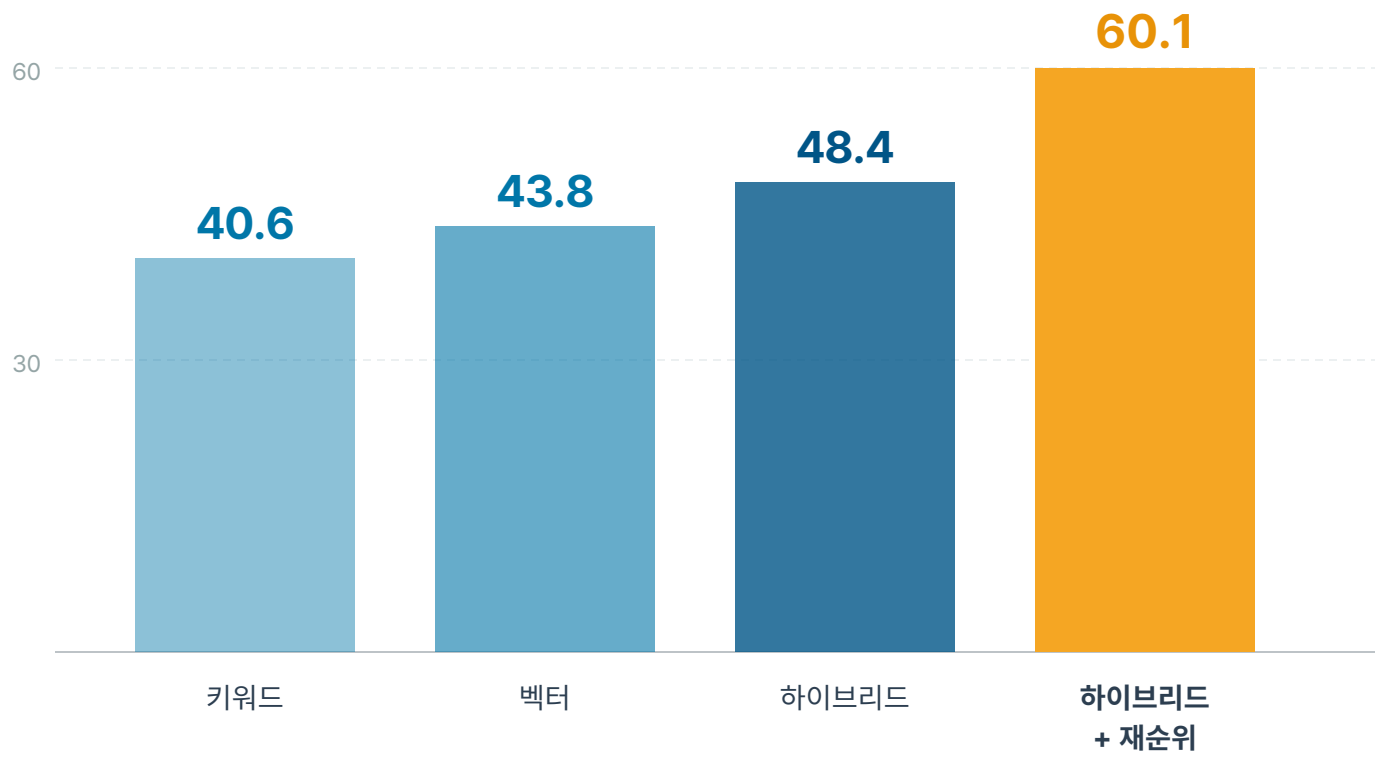
복잡한 기법부터 넣으면
지연 · 비용만 늘고
무엇이 효과였는지
구분할 수 없음

Source: 공개 기술 발표 자료 2건 종합 · MSAP.ai 내부 기술 자료

검색 방식별 정확도 점수

KEY INSIGHT 하이브리드에 재순위를 더한 구성이 모든 질의 유형에서 최고 — 검색 정확도를 올릴 때 먼저 검토할 구성임

검색 방식별 nDCG@3 (고객 데이터 기준)



다국어 벤치마크

49.6 → 72.0

기본 검색 → 하이브리드 + 재순위

오타가 있는 질의

28.8 → 54.6

질의 유형별로도 모든 유형에서 최고

Source: 공개 기술 발표 자료 1건 종합 · MSAP.ai 내부 기술 자료

하이브리드 검색 결합 방식 3가지

KEY INSIGHT 점수 척도가 다른 두 검색은 순위로 합치는 RRF 가 다루기 쉬움 — 두 검색을 모두 돌리므로 비용 증가를 감안해야 함

점수 가중 결합

α × 벡터 + (1 − α) × BM25

α 는 보통 0.5–0.7

두 점수의 척도를 맞추는 정규화가 필요

가중치 조정이 쉬움

RRF 순위 병합

척도 차이에 강함

Σ 1 / (k + 순위), k = 60

점수 대신 순위를 더함

척도가 다른 검색도 정규화 없이 결합

양쪽에 나온 문서가 위로 올라감

단계적 좁히기

BM25 로 거름 → 벡터로 정렬

키워드로 후보를 먼저 줄인 뒤 의미로 정밀하게 순위를 매김

벡터 계산량이 적음

⚠ 가중치와 무관하게 두 검색을 모두 실행하므로 비용 · 지연이 늘어남 — 제품명 · 코드 질의가 많은지로 도입 판단

Source: 공개 기술 발표 자료 2건 종합

RRF 순위 병합 계산 예시

KEY INSIGHT 양쪽 결과에 모두 나온 문서가 위로 올라옴 — 벡터 검색은 항상 무언가를 돌려주므로 거리 임계값으로 잘라야 함

키워드 검색 결과

- 1위 문서 A
- 2위 문서 B
- 3위 문서 C

벡터 검색 결과

- 1위 문서 B
- 2위 문서 D
- 3위 문서 A

문서	계산 (k = 60)	점수
B	1/62 + 1/61	**0.0325**
A	1/61 + 1/63	**0.0323**
D	1/62	0.0161
C	1/63	0.0159

예: 3위와 5위로 나온 문서 = 1/63 + 1/65

최종 순위

1 문서 B

2 문서 A

3 문서 D

4 문서 C

양쪽에 모두 나온 B · A 가 상위로 올라감

주의: 벡터 검색은 항상 무언가를 돌려줌

질문과 상관없는 문서도 상위 K 에 들어옴 —
절대 거리 임계값으로 잘라야 엉뚱한 근거가 섞이지 않음

Source: 공개 기술 발표 자료 2건 종합 · MSAP.ai 내부 기술 자료

질의 변환 기법 4가지

KEY INSIGHT 구어 질문은 문서의 표현과 다름 — 검색용 질의·키워드 질의·필터 조건 세 벌로 바꿔야 누락이 줄어듦

MultiQuery · RAG-Fusion

같은 뜻의 질문을
여러 개 만들어 검색
결과는 RRF 로 통합

"VPN 접속 실패"
→ "원격 접속 오류"
→ "VPN 인증 문제"

Decomposition

복합 질문을
하위 질문으로 나눠
따로 검색

"A 와 B 제품 라이선스 차이"
→ A 라이선스 조건
→ B 라이선스 조건

HyDE · Step-back

HyDE: 가상의 답을 먼저
쓰고 그 답으로 검색

Step-back: 한 단계 일반화
한 질문도 함께 검색

"이 서버 메모리 부족 원인"
→ "메모리 부족이 생기는
일반적 원인"

Self-Query

자연어에서 메타데이터
필터 조건을 뽑아냄

"작년 보안 정책"
→ 질의: "보안 정책"
→ 필터: year == 2025

재작성 결과는 세 벌로 만듦

"호텔비" → 벡터용 "숙박비 상한" · 키워드용 "숙박비" · 필터 "시행일 · 승인 상태"
"어깨 뭉친 게 열흘째 안 풀리네" → "어깨 뭉침 푸는 운동"

Source: 공개 기술 발표 자료 6건 종합

단계별 고급 RAG 기법 지도

KEY INSIGHT 기법마다 고치는 단계가 다름 — 실패 위치를 먼저 진단하고 그 단계의 기법만 골라 넣어야 함

사전 검색 계획: 무엇을 찾을지	검색 · 사후 검색 획득: 어떻게 가져올지	생성 답변: 어떻게 쓸지
질의 확장 HyDE · GAR · MultiQuery	**메타데이터 필터** 시행일 · 부서 · 문서 유형	**근거 배치** Long-Context Reorder
오타 교정 · 재작성 구어를 검색어로 변환	**하이브리드 · 임베딩 튜닝** BM25 + 벡터, 도메인 학습	**출력 형식 고정** 결론 · 근거 · 전제 · 한계
질문 분해 Self-Ask · Decomposition	**재순위 · 필터링** Cross-encoder · LLM 재순위	**자기 검증** Self-RAG · Corrective RAG
일반화 질문 Step-Back	**요약 압축** RECOMP · 불필요 문장 제거	**리더 방식** 추출형 · 생성형
검색 대상 라우터 메타데이터 · 의미 라우터	**반복 · 적응형 검색** ITER-RETGEN · FLARE	**답변 거절 규칙** 근거 없음 → 확인되지 않음

구성 요소를 모듈처럼 바꿔 끼우는 Modular RAG 로 발전 중

Source: 공개 기술 발표 자료 3건 종합

재순위: 넓게 찾고 다시 고르기

KEY INSIGHT 1차 검색은 재현율, 재순위는 정밀도를 맡음 — LLM 에 넘기는 근거가 줄어 비용과 오답이 함께 줄어듦



예: top 4 → top 2 로 압축 · 점수 기준 미달 결과 제외

효과

48.4 → 60.1

하이브리드 → 재순위 추가 (nDCG@3)

LLM 에 넘기는 근거가 줄어 비용 · 오답 감소

운영 사례: 점수 임계값

재순위 점수 기준 2.5 설정

실제 상위 결과 점수 2.4–2.8

기준 미달은 근거에서 제외

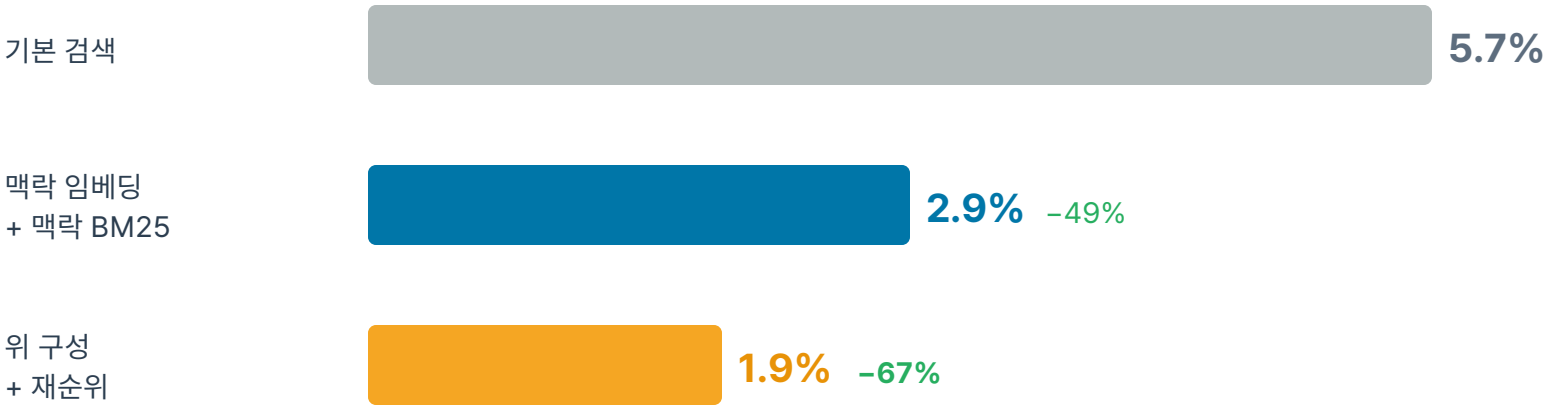
국내 사례: LLM 재순위를 오픈소스 모델로
사내 구축해 공공 데이터로 검증

Source: 공개 기술 발표 자료 5건 종합

청크 맥락화로 검색 실패 67% 감소

KEY INSIGHT 맥락 한 줄이 짧은 청크의 모호함을 없앰 —
재순위와 함께 쓰면 상위 20개 검색 실패율이 5.7% 에서 1.9% 로 낮아짐

상위 20개 검색 실패율 (낮을수록 좋음)



코드 · 소설 · 논문 혼합 데이터, 정답 청크 포함 여부 기준

원래 청크	"매출 3% 증가"
맥락 추가	"2분기 실적 보고서 중 A 사업부 매출에 관한 문단: 매출 3% 증가"

같은 문제를 푸는 방법

Contextual Retrieval
LLM 이 청크마다 문서 맥락
한 줄을 생성해 붙임

Late Chunking
문서 전체를 먼저 임베딩한 뒤
청크 구간별로 나눔

후보 질문 색인
청크가 답할 질문을
함께 색인

Source: 공개 기술 발표 자료 2건 종합 · Anthropic, Contextual Retrieval · Late Chunking(arXiv 2409.04701)

사내 규정 RAG 정답률 개선 사례

KEY INSIGHT 데이터 정비와 청크 크기 조정이 HyDE·재순위보다 효과가 컸음 — 기본기를 먼저 점검하면 적은 비용으로 큰 개선을 얻음

사례 1 · 사내 규정 RAG (통근 수당 · 휴가 등 예외 규정 15문항)



실험 축: 전처리(직원 유형별 문서 분할) · 청킹(표준 · 큰 청크 · 부모-자식 · HyDE) · 후처리(Cross-encoder 재순위) — 데이터 정비와 큰 청크가 가장 효과적

사례 2 · 사내 헬프데스크 RAG



FAQ 지식 정비 · 검색 튜닝 · 프롬프트 개선 / FAQ 자동 생성 · 문의 이력 정제 도구 병행

왜 기본기가 이겼나

큰 청크는 여러 문단을 함께 읽는 것과 같아 예외 조항의 맥락이 유지됨

LLM 은 상위 K 개만 읽으므로 정답이 그 안에 들어오는 것이 가장 큰 개선

다음 단계

모호한 질문에 되묻는 에이전트
규정 유형별 멀티 인덱스 라우팅
의미 단위 동적 청킹

사례 1 은 표본 15문항 · 필수 키워드 자동 판정

Source: 공개 기술 발표 자료 2건 종합

자기 점검형 RAG 기법

KEY INSIGHT 검색 결과를 평가하는 단계를 넣으면 무관 문서로 답하는 일이 줄어듦 — 대신 호출 횟수가 늘어 지연이 커짐

Self-RAG

검색이 필요한지 LLM 이 먼저 판단

여러 답을 만든 뒤 근거와 맞는지

스스로 평가해 가장 좋은 답을 고름

필요 없는 검색을 줄임

Corrective RAG

검색 결과를 정답 · 모호 · 오답으로 분류

오답이면 질문을 바꾸거나

웹 검색으로 다시 찾음

무관 문서로 답하는 일을 막음

RAPTOR

문서를 재귀적으로 요약한 트리로 색인

세부 질문은 아래 층, 큰 질문은

위 층의 요약에서 답함

문서 전체를 묻는 질문에 대응

FLARE

답을 쓰는 도중 다음 토큰의

생성 확률이 임계값 아래로

떨어질 때만 검색

자신 없는 부분만 근거 보강

Source: 공개 기술 발표 자료 3건 종합

SECTION 05

GraphRAG와 에이전틱 RAG

관계 질문과 다단계 질문을 푸는 두 가지 확장 방식

벡터 RAG와 GraphRAG의 차이

KEY INSIGHT 여러 관계를 건너는 질문과 문서 전체 요약은 벡터 검색으로 풀기 어려움 — 그래프는 연결된 근거와 출처 경로를 함께 줌

벡터 RAG — 비슷한 조각을 찾음

조각 1 · 유사도 0.82

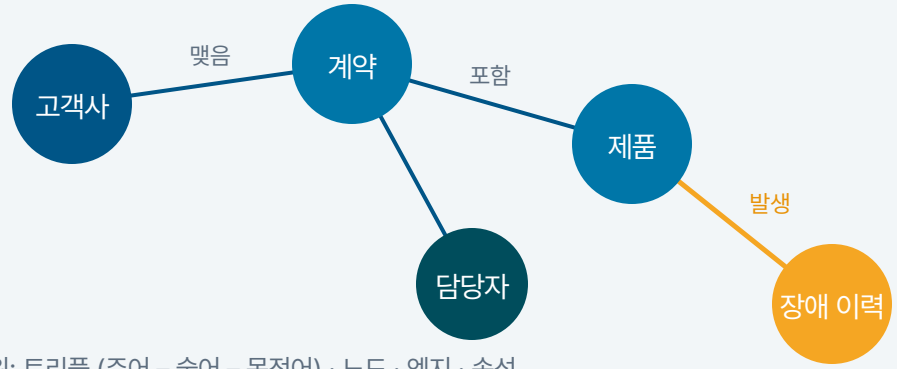
조각 2 · 유사도 0.79

조각 3 · 유사도 0.77

- 상위 K 조각만 가져옴
- 여러 관계를 건너는 질문(친구의 친구)에 약함
- 문서 전체를 요약하는 질문에 답하기 어려움

유사도 값은 설명용

GraphRAG — 관계를 따라감



기본 단위: 트리플 (주어 – 술어 – 목적어) · 노드 · 엣지 · 속성

비교 축	벡터 RAG	GraphRAG
복잡한 조건 질의	조건이 많을수록 약함	조건이 많아도 빠르고 정확
완전성	조각 단위	연결된 답 전체를 반환
출처 투명성	근거 경로가 안 보임	오답 원인을 경로로 추적

그래프의 약점: 집계 · 통계는 RDB 가 적합

Source: 공개 기술 발표 자료 3건 종합

GraphRAG 구축 6단계와 조회 방식

KEY INSIGHT 정형 데이터는 직접, 비정형 문서는 LLM 으로 트리플을 뽑음 — 조회는 고정 템플릿부터 시작해야 신뢰성이 높음

구축 6단계

1. 청크 분할
2. LLM 으로 노드 · 엣지 · 속성 추출
3. 요소 요약
4. 계층형 커뮤니티 탐지
5. 커뮤니티 요약
6. 질의별 답 통합

트리플을 뽑는 방법

정형(CSV · JSON · 코드 트리): 행 · 열 · 계층에서 직접 추출

비정형(PDF · 마크다운 · 워드): LLM 이 청크에서 추출

기존 RDB 를 그래프로 선언하는 방법(SQL/PGQ)도 있음

구축 유형 3가지

도메인 그래프 — 기존 업무 데이터로 구성

어휘 그래프 + 엔티티 확장 — 문서 청크와 엔티티 연결

커뮤니티 요약 추가 — 대규모 전역 요약용

조회 방식 (신뢰성 높은 순)

템플릿 질의

고정 질의 · 신뢰성 높음
여기서 시작

동적 생성

파라미터만 LLM 이 추출

그래프 탐색

벡터로 찾은 노드의 이웃 확장
LLM 호출 없어 빠름

Text-to-Cypher

자유롭지만 신뢰성 낮음
작은 그래프에 적합

Source: 공개 기술 발표 자료 2건 종합 · MSAP.ai 내부 기술 자료

GraphRAG 도입 비용과 계열

KEY INSIGHT 색인 비용을 0.1% 수준으로 줄인 방식이 나와 문턱이 낮아짐 — 관계 질문이 많은 업무부터 골라 적용해야 함

강점

- 여러 단계를 건너는 추론
- 문서 전체를 아우르는 질문
- 근거 경로로 해석 가능

과제

- 구축 중 LLM 대량 호출로 비용 큼
- 문서 성격에 따라 그래프 품질 갈림
- 같은 개념을 한 엔티티로 못 합침
- 데이터가 늘면 커뮤니티 재요약

계열	특징
Microsoft GraphRAG	커뮤니티 요약으로 전역 질문에 답함
LazyGraphRAG	색인 비용은 벡터 RAG 수준 (GraphRAG 의 0.1%) 전역 질의 비용 1/700 · 품질은 비슷
LightRAG	저수준 · 고수준 2단 검색, 증분 갱신
Graphiti	관계에 유효 기간을 붙인 동적 그래프
StructRAG	질문에 맞춰 지식 구조를 동적으로 선택

확장: 멀티모달 GraphRAG (연구 단계)

텍스트 그래프와 이미지 그래프를 합쳐 도면 · 차트 문서에 대응
답이 없는 질문에 "해당 없음"을 정확히 판정 · 구축이 느려 실시간 용도는 부적합

Source: 공개 기술 발표 자료 2건 종합 · Microsoft Research, LazyGraphRAG · MSAP.ai 내부 기술 자료

에이전틱 RAG 동작 방식

KEY INSIGHT 검색이 필요한지부터 에이전트가 판단함 — 복합 질문을 하위 질문으로 나눠 여러 저장소를 병렬로 찾음

일반 RAG

질문 = 검색어

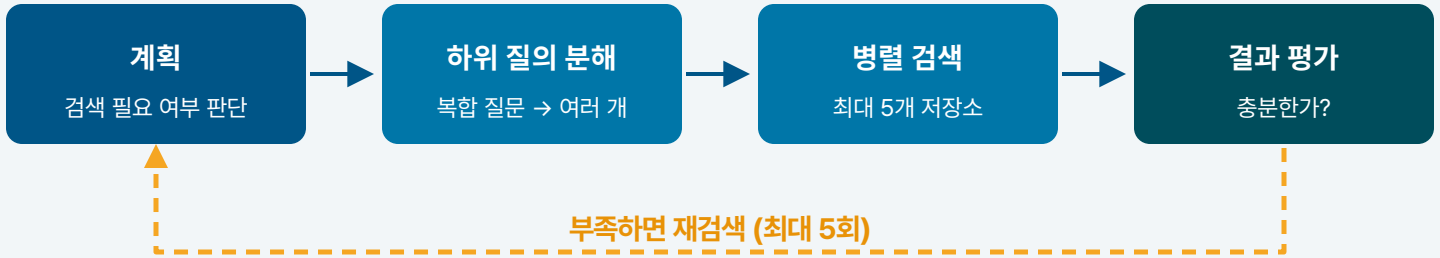
↓

검색 1회 (항상 고정)

↓

생성

에이전틱 RAG — 필요할 때 검색하고 부족하면 다시



충분하면 답 합성 + 출처

넓은 의미의 RAG

벡터 검색에 더해 웹 검색 ·
API 호출 · SQL 을 도구로 사용
RAG 를 에이전트의 검색 도구로 붙이거나
MCP 서버로 공유

실행 예

여러 언어가 섞인 복합 질문을
하위 질의 3개로 분해 → 재순위 →
답 합성, 출처 · 재순위 점수 함께 반환

구현 선택

관리형 서비스: 인덱스 → 인덱서 →
검색 에이전트 설정만으로 구성
직접 구현: 개발량 · 운영비 큼,
대신 맞춤 범위가 넓음

Source: 공개 기술 발표 자료 4건 종합

에이전틱 검색의 정확도와 지연

KEY INSIGHT 다단계 질문 정답률이 세 배로 오르지만 응답이 수십 초로 늘어남 — 검색 담당을 분리해 주 작업 흐름을 보호해야 함

여러 쪽 절차 질문 정답률

20→60%

ReAct 형 에이전틱 RAG
종료 판단은 LLM 이 함

반복 검색 도입 효과

5.9배

개선 요인 (논문 보고)

응답 지연 (관리형 사례)

약 26초

질의 계획 약 11초
+ 답 합성 약 15초

컨텍스트 누적

검색 이력 · 실패 결과가 주 작업 흐름에 쌓임

답 품질 · 비용에 영향

대응

검색 담당 에이전트를 분리해
결과 요약만 주 에이전트에 돌려줌

단순 질문은 일반 RAG 로, 다단계 질문만
에이전틱 검색으로 라우팅

구성 선택 교훈

문서마다 에이전트를 둔 구성은
호출이 폭증하고 결과도 부정확

순서가 정해진 업무는 사전 정의형 워크플로가
슈퍼바이저형과 같은 결과를 더 단순하게 냄

Source: 공개 기술 발표 자료 4건 종합 · MSAP.ai 내부 기술 자료

질문 유형별 RAG 패턴

KEY INSIGHT 벡터 검색만으로는 관계·집계·시점 질문에 답하지 못함 — 라우터가 질문마다 맞는 저장소를 골라야 누락이 없음

Naive

일반 문서 질의응답

"휴가 신청 절차는?"

Document

문서 전체를 읽는 요약 · 추출

"이 계약서의 위약 조건 정리"

SQL

수치 · 집계 질문 (NL2SQL)

"지난달 지역별 매출 합계"

저장소 분리 예 (top_k / 거리 임계값)

제품 사양 5 / 0.5

FAQ 3 / 0.4

규정 3 / 0.3 — 위험이 클수록 엄격

저장소마다 독립 갱신

라우터

질문 유형 판별

검색 엔진 조합

조건은 SQL

관계는 그래프

내용은 벡터

최신 정보는 인터넷 검색

Agentic

다단계 · 복합 질문

"규정 확인 후 남은 연차로 신청"

Memory

사용자 개인 맥락

조직 지식과 섞지 않음

Graph

관계 · 연결 질문

"이 장애와 연결된 고객사 목록"

결정 트리로 패턴 선택 · 저장소는 용도별 분리

Source: 공개 기술 발표 자료 2건 종합 · MSAP.ai 내부 기술 자료

SECTION 06

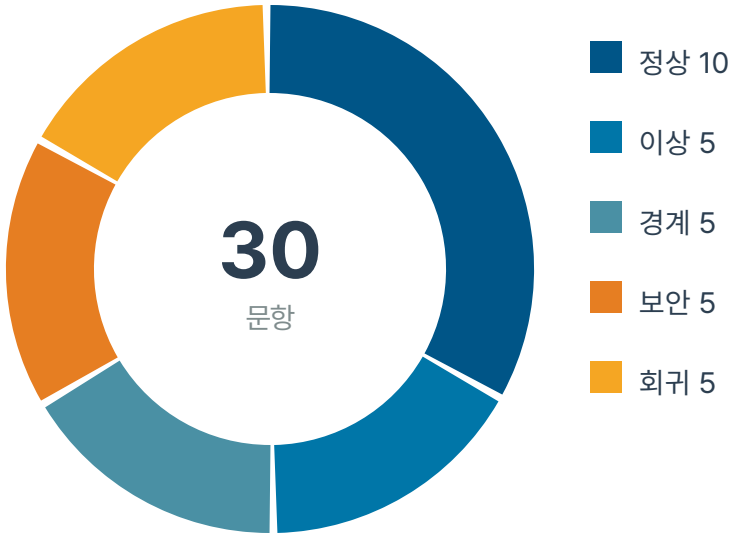
평가 · 권한 · 최신성 운영

운영 단계에서 RAG 품질과 보안을 유지하는 조건

평가 세트 구성

KEY INSIGHT 개선 효과는 평가 데이터로만 확인 가능 — 프로젝트 시작 때 만든 데이터가 이후 모든 변경의 기준이 됨

운영 전 평가 30건 구성



평가 단위: 질문 · 이상적인 답 · 정답 문서 세 가지를 한 묶음으로 구성

시작

코드보다 평가 문항 5건을 먼저 작성

무엇을 맞혀야 성공인지부터 합의

운영 전

30건으로 확장 · 변경 때마다 다시 채점

실제 사용자 질문으로 계속 보충

2층 평가

1층 검색 단독
정답 문서를 찾았는가

2층 종단 평가
답과 도구 호출 기록까지 확인

업무 관점

ROI · 정확도로
유스케이스 선정

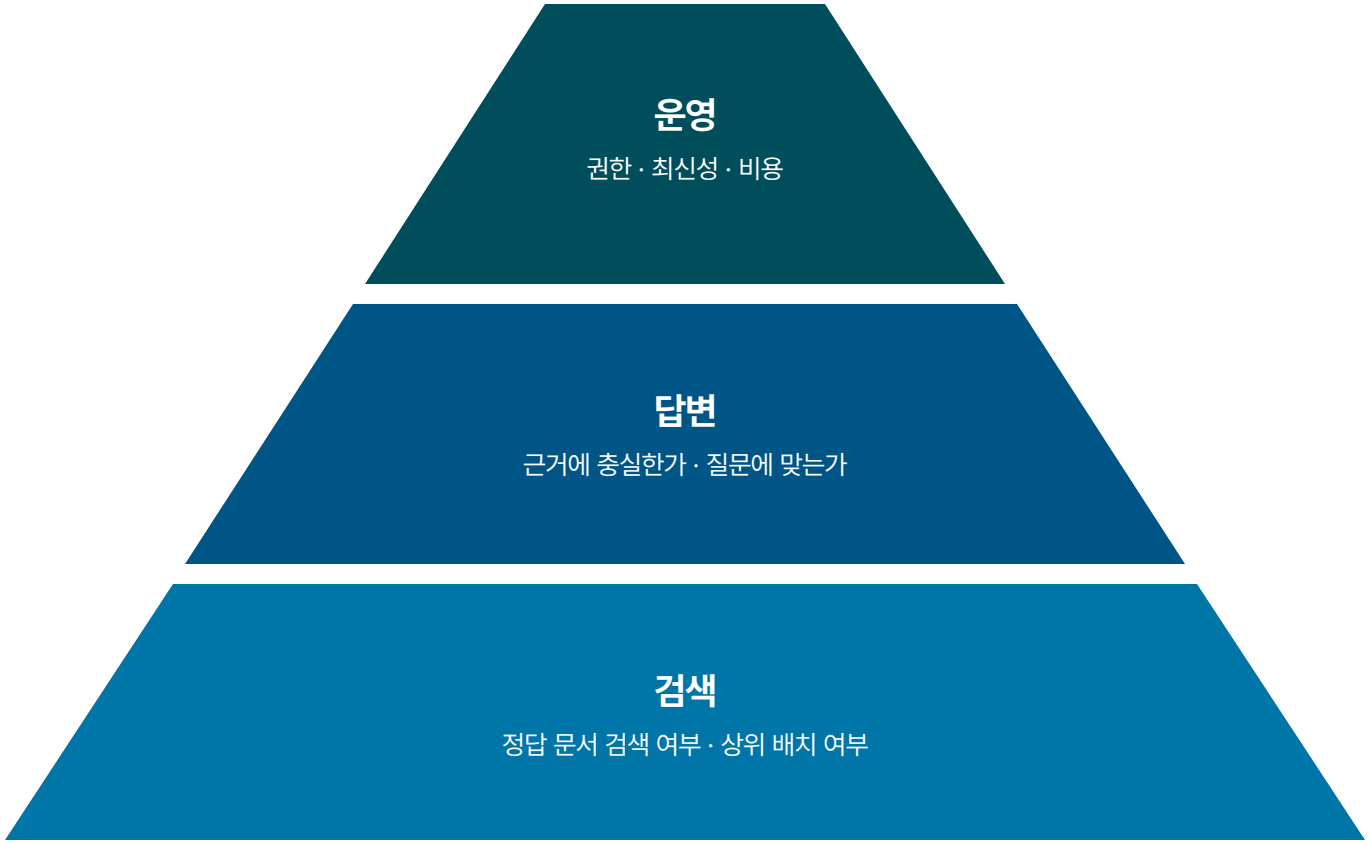
업무에 필요한 정확도를
먼저 정의

정기 평가가 핵심이자 가장 어려움

Source: 공개 기술 발표 자료 3건 종합 · MSAP.ai 내부 기술 자료

RAG 평가 지표 3층 구조

KEY INSIGHT 답변 점수만 보면 원인을 못 찾음 — 검색·답변·운영 지표를 따로 재야 개선 대상을 고를 수 있음



운영 지표

권한 위반 0건 · 구버전 참조율
응답 지연 · 질문당 비용
답할 수 없을 때 거절하는 비율

답변 지표 (RAGAS)

Faithfulness 근거 충실성
Answer Relevancy 답변 적합성
Context Recall · 인용 적중률

검색 지표

Precision · Recall@k · MRR · nDCG
검색 적중률 (평가 세트 정답 문서 일치)
BLEU · ROUGE 같은 문장 지표는 보조로만

Source: 공개 기술 발표 자료 3건 종합 · Ragas 문서 · MSAP.ai 내부 기술 자료

LLM 자동 채점의 한계와 보완

KEY INSIGHT 자동 점수와 현장 감각이 다를 수 있음 — 사람 평가와의 상관을 확인한 뒤에 자동 채점을 넓혀야 함

한계

위치 편향
먼저 본 답에 점수를 더 줌

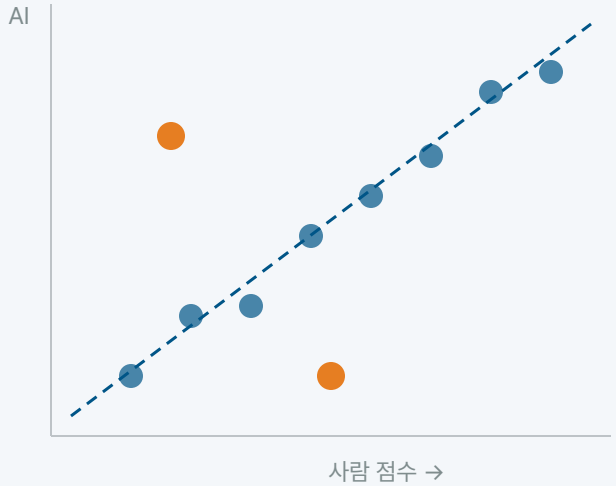
프롬프트 의존
채점 지시 문구에 따라 점수 변동

평가자 자신의 환각
채점 근거를 지어냄

보완

- 답의 순서를 바꿔 반복 평가
- 여러 LLM 의 다수결
- temperature 0 · seed 고정
- 정확성 · 망라성 1–5점 + 사유

사람 점수 vs AI 점수 (개념도)



도입 순서

사람 평가와 AI 평가의 상관 확인 → 차이 나는 문항은 전문가 육안 검수 → 일치가 확인된 범위만 자동 채점으로 확대

환각 · 유해성은 자동 평가만으로 보장되지 않음

Source: 공개 기술 발표 자료 4건 종합

오프라인 평가와 온라인 평가 병행

KEY INSIGHT 평가 세트만으로는 실제 질문의 분포를 못 담음 — 사용자 피드백을 다음 평가 세트로 되돌려야 품질이 유지됨

오프라인 평가 · 배포 전

RAGAS · 관리형 평가 도구로
검색 정확도 자동 채점

질문 · 정답 쌍을 LLM 으로 자동 생성,
출처 URL 을 메타데이터로 부착

변경할 때마다 회귀 확인

강점: 빠르고 반복 가능 · 약점: 실제 질문 분포를 못 담음

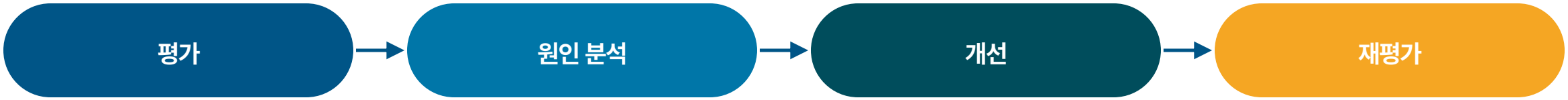
온라인 평가 · 배포 후

담당 부서 시범 → 전 직원 베타

답변마다 Good / Bad 버튼과 코멘트

Bad 답변은 원인 분류 후
평가 세트에 추가

강점: 실제 사용 반영 · 약점: 피드백 양이 들쭉날쭉



사용자 피드백을 다음 평가 세트로 되돌리는 순환

Source: 공개 기술 발표 자료 3건 종합

임베딩 모델은 사내 골든셋으로 선정

KEY INSIGHT 공개 벤치마크 순위와 사내 문서 성능은 다름 — 모델을 바꾸면 전체 재임베딩 비용이 들어 처음 선택이 중요함

선정 절차

1 사내 골든셋 만들기
실제 질문 50–200건 + 정답 근거 문서

2 후보 2–4개 A/B 비교
Recall@k · MRR 로 같은 조건에서 비교

3 비용 · 속도까지 보고 확정
모델을 바꾸면 전체 재임베딩 비용 발생

공개 순위만 보고 고르면
범용 임베딩은 도메인 검색 품질이 부족할 수 있음
검색 로그를 학습 신호로 쓰는 체계가 차이를 만듦

후보 예시

모델	특징
BGE-M3	Dense · Sparse · Multi-vector, 8192 토큰
multilingual E5	경량 기준선
한국어 특화 모델	국내 문서 · 용어에 맞춘 학습

한국어 처리
형태소 분석 · 복합어 분할
표기 흔들림 · 동의어 사전
같은 정규화 규칙을 색인과
검색 양쪽에 적용
요약 프롬프트 언어도 맞춤

필수 메타데이터
source 원본 위치
section 장 · 절
문자 위치 (인용용)
ACL 접근 권한
timestamp 시행 · 갱신일

Source: 공개 기술 발표 자료 2건 종합 · MSAP.ai 내부 기술 자료

공유 에이전트의 문서 권한 통제

KEY INSIGHT 암호화만으로는 정당한 호출의 범위 밖 조회를 못 막음 — 검색 시점에 사용자별 범위를 적용해야 유출이 없음

위험 시나리오: 에이전트 하나를 여러 사람이 공유



검색 시점에 적용하는 통제 방식

전역 ACL

폴더 단위 접근 목록

가장 단순 · 세밀도 낮음

문서별 ACL

메타데이터에 허용 · 거부 목록

고보안 영역은 청크 단위까지 적용

속성 필터

부서 · 직급 같은 속성으로 필터

역할 · 속성 기반 통제(RBAC · ABAC)

원본 시스템의 권한을 그대로 연동하고 검색 때 실시간 재검증

폴더 접근 권한이 그 안의 모든 문서 권한을 뜻하지 않음

RAG 성숙도

Naive → Advanced → Enterprise(권한 · 신선도 · 감사)

Source: 공개 기술 발표 자료 3건 종합 · OWASP Top 10 for LLM Apps 2025

권한 사칭을 막는 2중 방어

KEY INSIGHT 권한 값은 LLM 생성값을 버리고 인증 토큰에서 결정적으로 주입함 —
한 겹이 실패해도 열리지 않도록 도구·리소스 계층을 모두 막아야 함

핵심 위험: 사용자 ID 를 LLM 이 도구 인자로 만들게 두면

프롬프트 인젝션이나 환각으로 다른 사용자를 사칭해 그 사람의 문서를 검색할 수 있음

계층	1차 방어	2차 방어
도구 계층 에이전트 → 게이트웨이	**요청 가로채기 (Interceptor)** 검증된 인증 토큰에서 사용자 정보를 꺼내 도구 호출 인자를 결정적으로 덮어씀 LLM 이 쓴 값은 버림	**정책 엔진 검사** 인자의 사용자와 토큰 클레임이 일치하는지 확인 다르면 호출 거부
리소스 계층 지식베이스	**ACL 을 아는 검색** 문서 권한에 없는 사용자에게는 검색 결과 자체를 돌려주지 않음	**리소스 정책** 게이트웨이 전용 실행 역할 외의 직접 검색 호출을 명시적으로 거부 우회 경로 차단

가로채기 단계의 버그 · 설정 오류는 그대로 열림(fail-open) — 한 겹에 의존하지 않음

Source: 공개 기술 발표 자료 1건 종합

RAG를 노리는 공격 3가지와 대응

KEY INSIGHT 공격자는 모델 대신 검색되는 문서를 조작함 — 로드 전 검증과 검색마다 감사 로그가 기본 방어선임

문서 오염

로드 단계에 악성 · 변조 문서를
넣어 답을 조작

→

대응

출처 검증 · 로드 승인 절차
허용된 원본 저장소에서만 수집

간접 프롬프트 인젝션

정상 문서 안에 숨긴 지시로
검색해 온 에이전트의 행동을 탈취
예: "이전 지시를 무시하고 ..."

→

대응

권한 밖 문서는 내용과 상관없이 아예 검색하지 않음
입력 필터로 지시문 패턴 차단 · 검색 결과는 데이터로만 취급

벡터 · 임베딩 취약점

테넌트 간 유출 · 임베딩 역추적
(OWASP LLM08)

→

대응

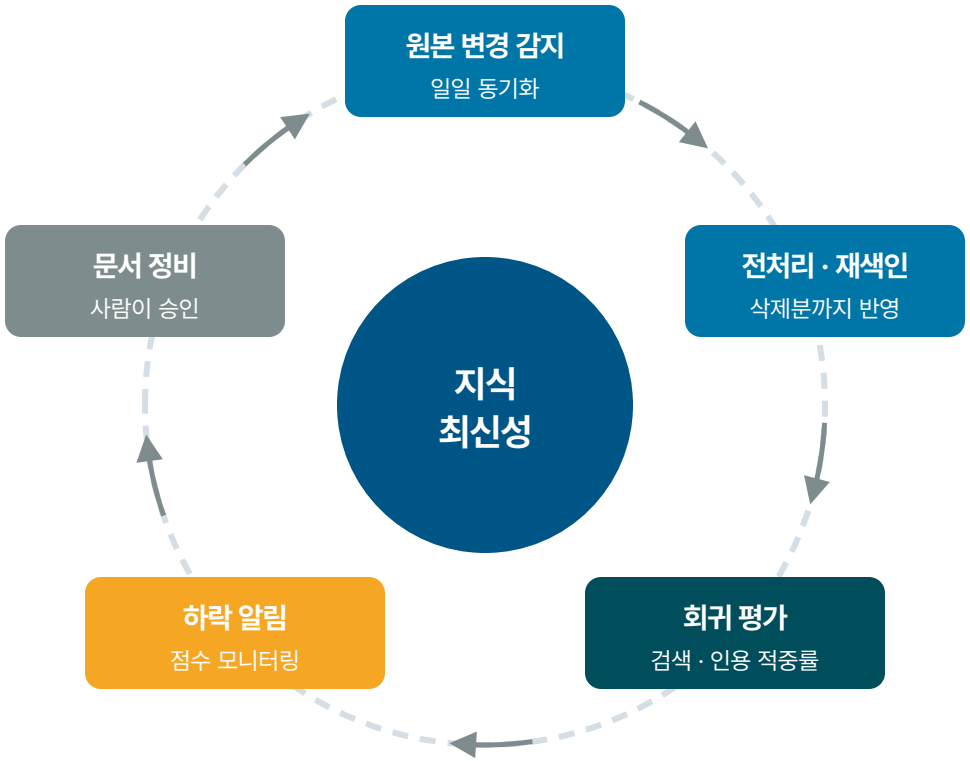
테넌트별 인덱스 분리 · 임베딩 전 개인정보 제거
검색마다 사용자 · 에이전트 신원과 함께 감사 로그

공격자는 검색되는 문서를 조작함 — 들어오는 길(로드)과 나가는 길(검색)을 모두 지킴

Source: 공개 기술 발표 자료 2건 종합 · OWASP Top 10 for LLM Apps 2025

지식 최신성을 유지하는 운영 순환

KEY INSIGHT 정확도 하락은 문서 변경 직후에 생김 — 동기화와 회귀 평가를 자동화하고 노후 문서 판정만 사람이 맡아야 함



로드 방식 선택 — 독자 가공이 필요한가

사전 가공 후 로드

업무 규칙으로 청크 범위 · 메타데이터 제어

이미지를 텍스트로 미리 변환

직접 로드

인덱서가 OCR → 청킹 → 벡터화를 표준 처리

빠르게 시작 · 맞춤 범위는 좁음

운영 분업

노후 문서 식별 · 동기화 · 회귀 채점은 자동화

유효성 판단과 승인은 사람

서버리스 평가로 고정비 없이 운영한 사례

Source: 공개 기술 발표 자료 3건 종합 · MSAP.ai 내부 기술 자료

운영 중 반복되는 실패 패턴

KEY INSIGHT 검색한 척하는 응답과 평가 생략이 가장 발견하기 어려움 — 도구 호출 기록과 정기 평가로만 드러남

01 Prompt Spaghetti

지시문에 규칙을 계속 덧붙여 서로 충돌

→ 200 토큰 넘으면 구조화, 500 넘으면 외부 파일로 분리

02 God Agent

에이전트 하나에 모든 일을 맡김

→ 역할별로 나누고 검색 담당을 분리

03 Memory Amnesia

대화 맥락을 잃어 같은 질문을 반복

→ 개인 기억과 조직 지식을 나눠 저장

04 Phantom Context

검색했다고 답하지만 도구 호출 기록이 비어 있음

→ 도구 호출 궤적까지 평가

05 Eval Neglect

평가 없이 배포 · 변경

→ 변경마다 30건 회귀 평가

06 Token Burn

필요 없는 문서까지 넣어 비용 낭비

→ 재순위 · 거리 임계값으로 근거 수 제한

Source: 공개 기술 발표 자료 1건 종합

지식 배치 위치 선택

KEY INSIGHT 매일 바뀌는 값은 RAG 대신 API 로 조회함 — 양과 변경 빈도로 나눠야 최신성과 비용을 함께 지킴



배치 원칙

조직 지식(RAG)과 개인 기억(Memory)은 섞지 않음

둘이 충돌하면 규정을 우선하고 사용자에게 알림

RAG 를 만들지 않아도 될 때

지식이 적으면 컨텍스트에 직접 넣는 쪽이 간단

서비스 내장 검색 기능으로 충분한 경우도 많음

Source: 공개 기술 발표 자료 1건 종합 · MSAP.ai 내부 기술 자료

SECTION 07

도입 사례와 MSAP.ai 의 RAG 적용

업무별 도입 효과 · 도입 단계 · CogentAI 와 OpsKnow 적용 구조

업무별 RAG 도입 효과

KEY INSIGHT 문서를 찾고 정리하는 시간이 큰 업무일수록 효과가 큼 — 정답형보다 자문형 업무에서 먼저 성과가 남

보안관제 연간 보고서

2개월
→ 2주

10여 명 → 2–3명 (목표)
로그 분석 쿼리 자동 생성
폐쇄망 내부 구축

사내 헬프데스크

약 50%
→ 88%

답변 정확도
자가 해결률 목표 55%
AI 답변 → 사람 상담 전환

조사 보고서 작성

2–3시간
→ 약 5분

경량 모델로 여러 출처 수집
고성능 모델로 통합 · 추론

공공 클라우드 운영

"90%는 아는데 확인이 필요한" 질문 응답 단축

출처 링크 · 파일 경로 표시
감사 로그 알림 요약 + 대응 방법
신규 인력 자가 검색

공통점

문서를 찾고 정리하는 시간이 큰 업무일수록 효과가 큼 — 정답이 하나인 판정보다 사람을 돕는 자문형 업무에서 먼저 성과

각 수치는 공개 발표 사례 기준이며 조직 · 데이터에 따라 다름

Source: 공개 기술 발표 자료 4건 종합

RAG 도입 3단계: PoC에서 전사 확대까지

KEY INSIGHT 실데이터 연동 조율이 개발보다 오래 걸림 —
필요한 데이터만 API 로 받아 모의 데이터로 먼저 개발해야 일정이 지켜짐

1단계 · 약 3개월 PoC

나중에 조정할 수 있는 구조 선정

- 사내 문서 · FAQ 단순 RAG 로 시작
- 출처가 달린 즉답 · 24시간 응대
- 오프라인 평가 세트 준비
- 실데이터는 API 로 일부만 받고 모의 데이터로 먼저 개발

2단계 전사 베타

SSO 연동 · 일일 동기화 · 피드백

- 담당 부서 시범 → 전 직원 베타
- 문서 저장소 일일 자동 동기화
- Good / Bad 피드백 수집
- 평가 · 개선 순환 정착

3단계 고도화

여러 출처 수집 · 에이전틱 RAG

- 예: 휴직 신청 에이전트
- 규정 검색 + FAQ + 인사 정보
- 조회를 계획 → 실행 → 평가 → 재시도로 자율 처리

효과를 재는 지표 4가지

1차 응답 시간	자기 해결률	담당자 공수	답변 정확도

도입을 망설이는 이유와 대응

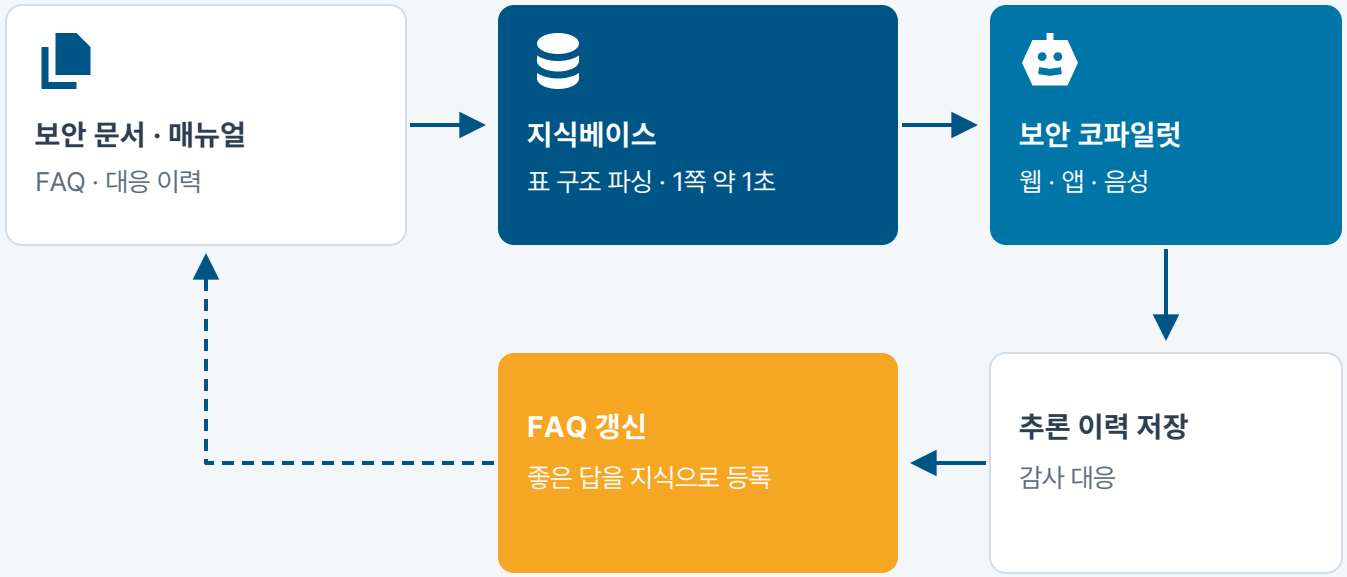
- 보안 → 입력 학습 미사용 · 암호화 · 가드레일
- 활용법 → 질문 가이드 · 예시 배포
- 사내 정보 → RAG 로 근거 연결

Source: 공개 기술 발표 자료 3건 종합

폐쇄망 보안관제 RAG 구성

KEY INSIGHT 정답이 하나인 판정보다 분석가를 돕는 자문형 업무에 맞음 — 폐쇄망 안에서도 지식 축적과 감사가 함께 돌아감

폐쇄망 내부 구성



활용: 자연어로 로그 분석 쿼리 생성 · 실행, 장애 보고서 · 기술 Q&A 정형 생성
관리: 문서 버전 · 주제별 관리, 업무별 프롬프트 관리 · 우수 프롬프트 공유

구축 전 점검 항목

목적	정답형보다 자문형
데이터 양	100만 단어 이상 권장
품질	다양성 · 정확성 확인
토큰화	한국어 처리 확인

1쪽 약 500단어 기준

피싱 클릭 후 내부 침투까지 평균 72분
분석가를 대체하지 않고 보조하며,
순환 배치로 끊기는 지식을 이어 줌

Source: 공개 기술 발표 자료 1건 종합

CogentAI: 운영 데이터와 사내 문서 결합

KEY INSIGHT LLM 은 시스템의 현재 상태를 모름 — 실시간 지표와 매뉴얼을 함께 찾아야 답이 우리 환경에 맞음

운영 데이터

실시간 지표 · 트랜잭션
APM · 관측 데이터
MCP 로 조회

사내 문서

장애 대응 매뉴얼
운영 문서 · 과거 분석 보고서
RAG 로 검색

CogentAI

자연어 질의 → 근거 결합

3계층 구조로 환각 억제
근거가 된 지표 · 문서를 함께 제시

답변

- 원인 트랜잭션 · 근거 지목
- 실행 가능한 조치안
- 참고한 매뉴얼 출처
- 우리 환경 기준의 해석

매뉴얼을 올릴수록
조직의 운영 노하우가
시스템에 축적

LLM 단독은 지금 시스템 상태를 모름 — 운영 데이터와 사내 문서가 함께 검색돼야 답이 우리 환경에 맞음

Source: MSAP.ai 내부 기술 자료

CogentAI 하이브리드 검색 5단계

KEY INSIGHT 용어가 분명한 매뉴얼·FAQ 검색에 효과적 — 데이터가 늘어 유사 문서가 많아지면 관계 기반 검색 보강이 필요함



잘 되는 질문

- 용어가 분명한 매뉴얼 검색
- FAQ · 문서 요약
- 약어와 정식 명칭이 섞인 질문
- 한 · 영 문서가 섞인 저장소

보강이 필요한 질문

- 데이터가 많아 비슷한 문서가 경합할 때
- 질문이 길거나 맥락을 묻는 질문
- 여러 문서 조각을 엮어야 하는 추론

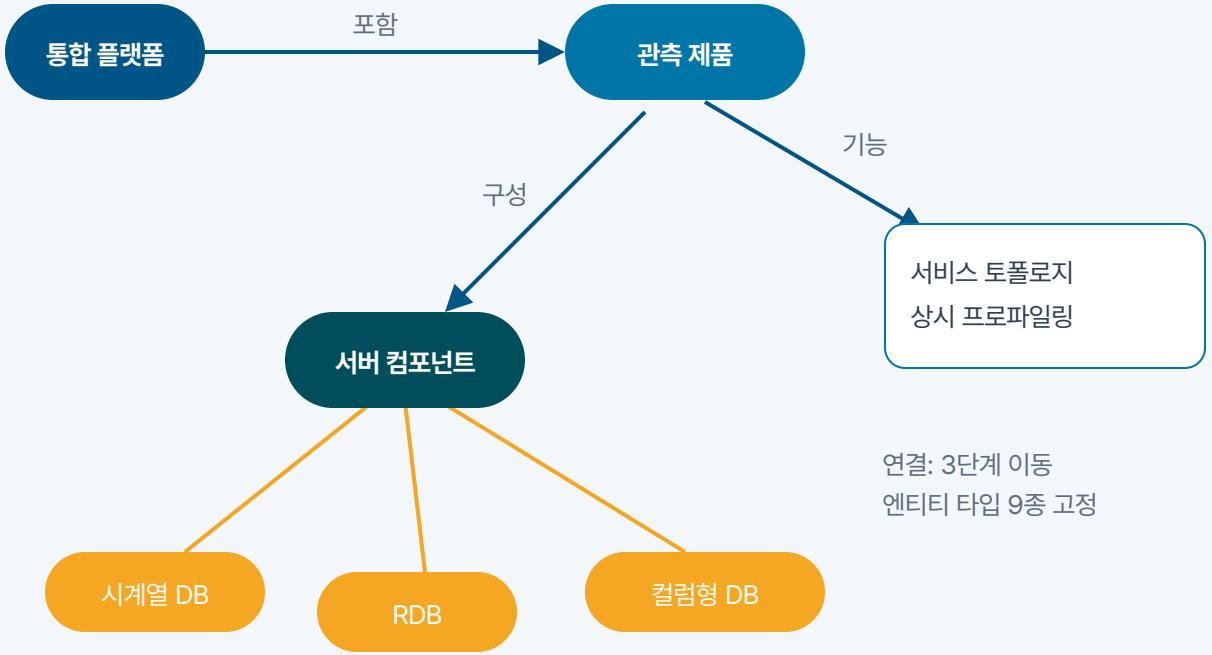
→ 관계 기반 검색(지식 그래프)으로 보강 검증 중

Source: 공개 기술 발표 자료 1건 종합 · MSAP.ai 내부 기술 자료

제품 지식 그래프 검증 결과

KEY INSIGHT 흩어진 문서 조각을 엮어야 하는 질문에 지식 그래프가 유리함 — 고정된 스키마와 정제된 데이터가 먼저 갖춰져야 함

제품 지식 그래프 예시



비교 검증 대상

GraphRAG	커뮤니티 요약, 갱신 비용 큼
LightRAG	증분 병합, 중복 개념 통합
Graphiti	관계에 시간 부여, 비용 큼
자체 그래프	엔티티 + 청크 연결

동적 그래프 구축: 데이터 114건에 약 7만 원

검증 결과

잘 정리된 문서에서는 벡터 방식과 큰 차이 없음

추론형 질문에서는 벡터 방식의 환각이 많음

문서 조각을 엮어야 할 때 그래프가 유리

전제: 고정된 스키마 · 정제된 데이터 · LLM 비용 관리

Source: MSAP.ai 내부 기술 자료

OpsKnow 운영 지식 저장소 구조

KEY INSIGHT 대화 기록은 그 대화를 한 사람만 되짚음 — 문서로 쌓아야 대화를 몰랐던 사람도 검색으로 찾음

운영 지식이 사라지는 곳

메신저
대화 스크롤에 묻힘

티켓
해결 과정이 남지 않음

담당자의 머릿속
순환 배치 · 퇴사 때 사라짐



저장
제품 계층 4단 + 날짜 트리 3단
계층 경로가 문서의 신원
노후 문서 식별은 기계, 승인은 사람

접근
표 머리행을 보존하는 청킹
절 단위 인용 표기
청킹 · 임베딩 손실 관리

활용
CogentAI 분석 결과를 시각 단위
문서로 남겨 다음 담당자가 검색

Source: MSAP.ai 내부 기술 자료

도입 전 점검 항목 10가지

KEY INSIGHT 열 가지 중 평가 세트 · 권한 연동 · 삭제 반영이 빠지기 쉬움 — 이 셋은 PoC 단계에서 먼저 정해야 함

항목	확인 질문	관련 내용
1 목적	자문형 업무인가, 정답이 하나인 판정인가	도입 효과 · 맞지 않는 업무
2 평가 세트	실제 질문 30건 이상과 정답 문서가 있는가	평가 세트 구성 · 지표
3 문서 정비	구버전 정리 · 표 구조화가 되었는가	데이터 유형별 전처리
4 청킹 규칙	조건과 결론, Q 와 A 가 같은 청크에 있는가	청크 크기 · 오답 사례
5 하이브리드 검색	제품명 · 코드 질의도 찾는가	결합 방식 · RRF
6 권한 연동	원본 시스템 권한이 검색 시점에 적용되는가	권한 통제 · 2중 방어
7 갱신 · 삭제 반영	삭제된 문서가 색인에서도 빠지는가	지식 드리프트 · 운영 순환
8 감사 로그	누가 무엇을 검색했는지 남는가	RAG 를 노리는 공격
9 배치 · 모델	폐쇄망 여부와 모델 · 임베딩 선택 기준이 있는가	임베딩 선정 · 생성 모델 오류
10 운영 담당	노후 문서 판정과 승인을 누가 하는가	운영 분업

강조 3항목은 PoC 단계에서 먼저 결정

Source: 본 자료 각 파트 종합

THANK YOU

도입 상담 연락처

품질은 지식 설계에서 갈림 · 평가 세트부터 시작 · 권한과 최신성까지 설계

MSAP.ai

msap.ai

https://www.msap.ai/

02-6953-5427

hello@msap.ai

© MSAP.ai. All rights reserved.