

WHITEPAPER

MSAP.ai

업무에 RAG 가 필요한 이유

답과 근거를 함께 주는 한국어 업무 지식 인프라

문서로 일하는 조직에서 RAG는 질문에 대한 답과 근거 문서를 함께 돌려줍니다. 한국어 문서에서는 임베딩과 검색 구성이 그 성과를 가릅니다.

2026-11-27 · 작성자: SMART CF Team · msap.ai

목차

업무에 RAG 가 필요한 이유

1장: 일은 문서로 하는데 문서를 찾지 못해 생기는 업무 비용
- 1.1 문서를 찾고 담당자에게 다시 묻는 데 쓰는 업무 시간
- 1.2 공유 드라이브와 메일로 흩어져 검색되지 않는 업무 문서
- 1.3 범용 생성형 AI 에 사내 질문을 할 때 생기는 환각과 근거 부재

2장: 키워드 검색이 답을 찾지 못하는 이유와 RAG 의 차이
- 2.1 같은 낱말이 있어야 찾는 키워드 검색의 구조적 한계
- 2.2 뜻으로 찾아 근거와 함께 답하는 RAG 의 처리 흐름
- 2.3 파인튜닝·긴 문맥 입력과 비교한 RAG 의 쓰임새

3장: 업무별로 보는 RAG 활용과 여덟 개 업무의 도입 효과
- 3.1 마케팅과 영업에서 고객 질문에 근거 자료로 답하는 RAG
- 3.2 회계·재무와 인사에서 규정과 기준 문서를 찾는 반복 질의
- 3.3 개발·운영과 고객지원·CRM 에서 장애 기록과 상담 이력을 쓰는 RAG
- 3.4 ERP·구매와 공공 행정에서 규정과 법령을 근거로 답하는 RAG
- 3.5 표와 도면, 스캔본에 핵심 내용이 담긴 이미지 문서의 검색

4장: 한국어 문서의 검색 성능을 가르는 한국어 임베딩 선택
- 4.1 조사가 붙고 띄어쓰기가 일정하지 않은 한국어 문서의 검색 난점
- 4.2 형태소 분석으로 높이는 한국어 키워드 검색의 재현율
- 4.3 한국어 임베딩 모델 비교표와 예시 실험으로 보는 선택 기준

5장: 키워드와 의미를 함께 찾는 하이브리드 검색과 리랭킹 설계
- 5.1 키워드 검색과 의미 검색이 서로 놓치는 질문 유형
- 5.2 하이브리드 검색에서 두 결과를 합치는 RRF 와 단순 결합의 위험
- 5.3 리랭커로 상위 후보의 순위를 다시 매기는 마지막 단계

6장: RAG 단독 운영에서 드러나는 한계와 보완 기술
- 6.1 문서 준비부터 답변 생성까지 RAG 가 실패하는 일곱 가지 지점
- 6.2 한계마다 맞는 그래프 RAG·문맥 덧붙임·긴 문맥 라우팅 대응표
- 6.3 다른 부서 문서가 검색되는 권한 누수와 악성 문서 주입 대응

7장: 챗봇을 넘어 업무 처리의 근거가 되는 RAG
- 7.1 문서 작성과 대조 점검 단계에 근거를 붙이는 RAG
- 7.2 에이전트가 판단 근거를 찾는 도구로서의 RAG
- 7.3 업무 시스템에 RAG 를 연결할 때 지킬 감사 기록과 승인 원칙

8장: 파일럿 업무 선정부터 확산까지의 도입 순서와 점검표
- 8.1 문서 찾기 비용이 큰 업무를 고르는 파일럿 선정 기준
- 8.2 평가 세트에서 시작해 운영 확산까지 넓히는 네 단계 도입 순서
- 8.3 결재 전에 확인하는 한국어 RAG 도입 점검표

9장: MSAP.ai 가 한국어 RAG 를 구축하고 운영하는 방식

9.1 MSAP.ai 한국어 RAG 의 하이브리드 검색과 리랭커 구성
9.2 조직 안에서 운영하는 MSAP.ai 한국어 RAG 의 모델 서빙과 그래프 검색
9.3 MSAP.ai 와 함께 시작하는 한국어 RAG 파일럿 진행 순서
Appendix
References
함께 읽을 자료
Glossary

업무에 RAG 가 필요한 이유

문서로 일하는 조직에서 RAG는 질문에 대한 답과 그 답의 근거 문서를 함께 돌려주는 업무 지식 인프라이며, 한국어 문서에서는 임베딩과 검색 구성이 그 성과를 가릅니다. 1장은 문서를 찾지 못해 생기는 업무 비용을, 2장과 3장은 키워드 검색과 RAG의 차이, 업무별 활용 사례와 효과를 다룹니다. 이어 4장과 5장은 한국어 임베딩·형태소 분석·하이브리드 검색·리랭커 구성을, 6장과 7장은 RAG의 한계와 보완 기술, 챗봇을 넘어선 업무 처리 쓰임새를 정리합니다. 8장은 파일럿 업무를 고르는 기준과 결재 전 점검표를, 9장은 이 조건을 갖춘 한국어 RAG 구성과 파일럿 진행 순서를 제시합니다. 따라서 의사결정권자와 IT 담당자는 읽는 동안 자사 업무 가운데 문서 찾기 비용이 큰 업무 하나를 골라, 한국어 RAG 파일럿 대상으로 정할 근거를 모을 수 있습니다.

1장: 일은 문서로 하는데 문서를 찾지 못해 생기는 업무 비용

RAG(Retrieval-Augmented Generation)는 질문과 관련된 사내 문서를 먼저 검색한 뒤 그 문서를 근거로 생성형 AI가 답을 만들게 하는 방식입니다. 이 방식이 업무 현장에서 검토 대상에 오르는 이유는 공개된 도입 사례의 수치에서 먼저 드러납니다. Morgan Stanley 어드바이저 팀은 필요한 사내 문서에 접근하는 비율이 20%에서 80%로 올랐고[S16], Twitch 광고 영업 조직은 도입 전에 광고주 질문의 20%에 답하지 못하고 있었습니다[S17]. 규정, 보고서, 계약서, 회의록으로 일하는 조직이라면 같은 비용이 이미 업무 시간 안에 섞여 있습니다. 1장은 문서를 찾지 못해 생기는 비용, 흩어진 저장 위치와 문서 형식, 범용 생성형 AI의 환각 위험을 차례로 짚고, 문서 검색 문제를 도입 검토 대상으로 올릴지 판단할 근거를 정리합니다.

1.1 문서를 찾고 담당자에게 다시 묻는 데 쓰는 업무 시간

문서를 찾는 시간과 담당자에게 다시 묻는 시간은 업무 일지나 원가 장부에 별도 항목으로 남지 않기 때문에, 대부분의 조직은 이 비용의 크기를 모른 채 넘어가고 문서 찾기를 개인의 검색 습관 문제로 여기기 쉽습니다. 하지만 공개된 도입 사례에서 도입 전 상태를 보여 주는 수치를 모으면, 찾지 못하는 비율과 반복 질문의 규모가 개인이 아닌 조직 단위로 나타납니다. 아래 네 사례는 업무 영역이 서로 다르지만 문서와 답 사이의 거리가 비용으로 이어졌다는 점이 같습니다.

사례	업무 영역	공개된 수치	수치가 가리키는 대상
Morgan Stanley	자산관리 어드바이저 지원	문서 접근률 20% → 80%, 어드바이저 팀 98% 이상 사용	어드바이저가 필요한 사내 문서에 접근한 비율(도입 전후)
Twitch	광고 영업	도입 전 광고주 질문 20% 미응답	답하지 못한 고객 질문
IBM AskHR	인사	흔한 질문 94%를 사람 연결 없이 해결, 티켓 75% 감소	인사 담당자가 받던 반복 질문
Uber Genie	엔지니어링 운영 지원	154개 채널, 질문 7만 건 이상, 약 1.3만 엔지니어링	선임 엔지니어가 답하던 반복 질문

사례	업무 영역	공개된 수치	수치가 가리키는 대상
		시간 절감(추정)	

출처: S16, S17, S20, S21이며, 표의 수치는 모두 해당 기업이 공개 발표한 값입니다.

1.1.1 규정과 이전 제안서를 사람마다 반복해서 찾는 검색 작업

같은 문서를 각자 다시 찾는 반복 검색

조직에서 가장 자주 찾는 문서는 새 문서보다 이미 있는 문서입니다. 규정 개정 이력, 지난해 같은 고객에게 낸 이전 제안서, 표준 계약서의 최신본, 지난 분기 보고서의 원본 표가 대표적입니다. 이런 문서는 한 사람이 찾아도 그 결과가 다른 사람에게 공유되지 않기 때문에, 다음 사람이 같은 폴더와 같은 메일함을 처음부터 다시 검색합니다. 제안서 작성 담당자 세 명이 같은 고객의 지난 제안서를 각자 찾으면 검색 비용은 세 번 발생하고, 셋이 서로 다른 판본을 찾아 쓰는 위험도 함께 생깁니다.

문서가 있는데도 닿지 못하는 비율

검색이 실패하는 이유는 대개 문서가 없어서보다 문서에 닿지 못해서입니다. Morgan Stanley 사례가 이 차이를 수치로 보여 줍니다. 어드바이저가 필요한 사내 문서에 접근하는 비율은 도입 전 20%였고, 생성형 AI 기반 문서 검색 도구를 쓴 뒤 80%로 올랐으며 어드바이저 팀의 98% 이상이 이 도구를 썼습니다. 이 수치는 OpenAI가 공개한 고객 사례에 실린 회사 발표 기준이며, 같은 사례는 이 변화로 문서 검색에 드는 시간이 크게 줄었다고 설명합니다[S16]. 다만 측정 방식은 공개되지 않았으므로, 자기 조직에서는 도입 전에 문서 접근률이나 검색 성공률을 먼저 재어 두어야 도입 뒤 효과를 같은 기준으로 비교할 수 있습니다.

1.1.2 담당자에게 다시 묻는 질문이 만드는 숨은 응대 비용

찾지 못한 답이 사람에게 넘어가는 흐름

문서에서 답을 찾지 못한 직원은 결국 그 문서를 아는 사람에게 묻습니다. 그 결과 비용은 묻는 쪽과 답하는 쪽 양쪽에서 생기며, 답하는 쪽은 대개 인사 담당자, 선임 엔지니어, 영업 지원 인력처럼 조직에서 대체하기 어려운 사람입니다. 아래 표는 공개 사례에서 반복 질문을 받던 사람과 그 규모를 정리한 것입니다.

사례	반복 질문을 받던 사람	공개된 규모
Twitch	광고 영업·영업 지원 인력	도입 전 광고주 질문의 20%에 답하지 못함
IBM AskHR	인사 담당자	흔한 인사 질문의 94%를 사람 연결 없이 해결, 2016년 이후 티켓 75% 감소
Uber Genie	사내 지원 채널을 맡은 선임 엔지니어	154개 채널에서 질문 7만 건 이상 응답, 약 1.3만 엔지니어링 시간 절감(추정)

출처: S17, S20, S21.

핵심 인력의 시간으로 계산하는 응대 비용

Twitch 사례의 20% 미응답은 고객 쪽에서 드러난 비용입니다[S17]. 답을 받지 못한 광고주 질문은 계약 지연이나 이탈로 이어질 수 있기 때문입니다. 반면 IBM과 Uber 사례는 조직 내부의 비용을 보여 줍니다. 두 회사가 줄였다고 밝힌 것은 인사 담당자와 선임 엔지니어처럼 대체하기 어려운 사람이 반복 질문에 쓰던 시간이며, 위 표의 수치는 모두 회사 발표 기준입니다[S20][S21]. 도입 뒤의 효과와 측정 조건은 3장 3.2.2절과 3.3.1절에서 업무별로 다룹니다. 의사결정권자는 「반복 질문 수 × 답하는 사람 1인의 시간」을 우리 조직에 대입해 보는 것만으로도 이 비용이 도입 검토 대상인지 판단할 수 있습니다.

1.2 공유 드라이브와 메일로 흩어져 검색되지 않는 업무 문서

업무 문서는 한곳에 모여 있지 않습니다. 같은 사업의 문서라도 계약서는 공유 드라이브에, 협의 내용은 메일 첨부에, 결재 문서는 그룹웨어에, 작업 중인 초안은 개인 PC에 남습니다. 게다가 저장 위치마다 형식이 다르고 검색 기능도 위치마다 따로 있어서, 직원은 한 번의 질문에 답하려고 서너 개의 검색창을 차례로 열어야 합니다. 아래 표는 흔한 저장 위치와 그곳에 주로 남는 문서 형식을 예시로 정리한 것입니다.

저장 위치	주로 남는 문서	흔한 형식
공유 드라이브	규정, 계약서, 제안서, 보고서	HWP, DOCX, PDF, XLSX
메일 첨부	협의 내용, 변경 합의서, 견적 요청서	PDF, HWP, 이미지 캡처
그룹웨어·결재 시스템	결재 문서, 공지, 회의록	게시글 본문, HWP, 스캔 PDF
개인 PC	작업 중인 초안, 내려받은 참고 자료	형식 제한 없음
업무 시스템 첨부	상담 이력, 장애 보고, 구매 증빙	텍스트, 스캔 PDF, 이미지

흩어진 문서는 검색 품질을 따지기 전에 검색 대상에 들어가지 못하는 문제가 먼저 생기므로, RAG 도입의 첫 과제는 검색창을 바꾸는 일보다 흩어진 문서를 한 검색 대상으로 연결하는 일에 가깝습니다.

1.2.1 HWP 와 스캔 PDF 로 남아 글자 검색이 어려운 기관 문서

표와 이미지 안에 담긴 핵심 내용

한국의 공공기관과 기업 문서는 HWP, PDF, 스캔본 비중이 높고, 업무 판단에 쓰이는 핵심 내용이 표와 이미지 안에 들어 있는 경우가 많습니다. 예를 들어 수당 지급 기준은 규정 본문보다 별표의 금액 구간 표에 있고, 공사 사양은 도면 이미지에, 결재 이력은 서명이 찍힌 스캔 PDF에 남습니다. 글자만 읽어 색인하는 검색은 이런 표와 이미지를 건너뛰기 때문에, 문서는 검색 결과에 나와도 답이 담긴 부분은 빠집니다. 문서 페이지를 이미지 그대로 임베딩해 검색하는 ColPali 같은 방식이 대안으로 연구되고 있으며, 글자 추출 단계를 거치지 않고도 표와 그림이 섞인 페이지를 찾을 수 있다는 점을 보였습니다[S15].

문서 형식별로 다시 확인할 항목

형식마다 글자 검색이 가능한 정도와 도입 전에 확인할 항목이 다릅니다. 다음 표는 형식별 일반 특성을 정리한 것으로, 실제 변환 품질은 문서 작성 방식과 변환 도구에 따라 달라집니다.

문서 형식	글자 검색 가능 여부	도입 전에 확인할 항목
텍스트 PDF	가능	표의 칸이 읽는 순서대로 추출되는지

문서 형식	글자 검색 가능 여부	도입 전에 확인할 항목
스캔 PDF	문자 인식(OCR)을 거치기 전에는 불가	인식 정확도, 표 구조 보존 여부
HWP	변환 도구를 거쳐야 가능	표·각주·삽입 개체의 변환 상태
이미지(도면·화면 캡처)	불가	이미지 자체 검색 또는 설명 글 추가 필요

따라서 도입 범위를 정할 때 의사결정권자가 가장 먼저 요청할 자료는 대상 업무 문서의 형식 목록입니다. 형식 목록이 있어야 어떤 문서를 바로 넣고 어떤 문서를 변환 뒤에 넣을지 정할 수 있습니다.

오래된 문서가 자산으로 남는 조건

오래된 업무 기록이 자산인지 짐인지는 보관 기간보다 저장 형식에서 갈립니다. 검색은 본문 글자를 훑고, RAG 가 답의 근거로 인용하는 것도 본문의 문단이므로, 본문이 글자로 살아 있는 문서만 검색과 인용 양쪽에 쓰입니다. 같은 십 년 치 기록이라도 그림 파일로 굳은 문서는 답의 근거가 되지 못하고, 문단 단위로 읽히는 문서는 같은 사고나 민원이 다시 생겼을 때 대응 시간을 줄여 줍니다. 이 차이는 도입 전에 간단히 확인할 수 있습니다. 대상 저장소에서 가장 오래된 문서 한 건을 골라 변환해 보고, 제목과 목록, 표가 구분되어 읽히는지 보면 됩니다. 변환 결과에서 표의 칸 순서가 뒤섞이거나 각주가 본문 중간에 끼어 있으면, 그 형식의 문서는 변환 방식을 바꾸거나 정리 작업을 거친 뒤에 색인 범위에 넣습니다.

1.2.2 인사 이동과 퇴사로 끊기는 업무 맥락과 지식 공백

업무 인수인계에서 사라지는 판단 근거

담당자가 바뀌면 문서는 남아도 문서를 해석하는 맥락이 함께 사라집니다. 어느 판본이 최신인지, 왜 그 조건으로 계약했는지, 지난번 민원에 어떤 근거로 답했는지는 대개 전임자의 기억에만 있습니다. 업무 인수인계 때 흔히 넘겨받는 문서 목록은 다음과 같지만, 목록만으로는 결정 사유까지 넘어오지 않습니다.

- 업무 규정과 개정 이력
- 진행 중인 계약서와 변경 합의서
- 이전 제안서와 평가 결과
- 회의록과 결정 사유
- 거래처·민원 응대 기록

흩어진 지식을 한곳에서 찾게 만든 사례

인력 변동이 잦은 조직일수록 이 지식 공백은 같은 형태로 되풀이됩니다. IskayPet은 여러 시스템에 흩어진 사내 지식을 한 클라우드 사업자의 관리형 지식 기반 서비스로 묶어, 직원이 필요한 지식을 한곳에서 찾을 수 있게 했습니다[S26]. 공공 부문에서는 싱가포르 GovTech의 AIBots가 규모를 보여 줍니다. 2025년 2월 기준 115개 기관의 공무원 4만 명이 기관별 지식 봇을 만들어 쓰고 있으며, 자세한 내용은 3장 3.4.2절에서 다룹니다[S28]. 즉 문서를 사람의 기억 대신 검색 가능한 지식으로 연결해 두면, 담당자가 바뀌어도 후임자가 같은 근거에 닿을 수 있습니다.

같은 제목으로 쌓이는 보고서와 구분 속성

인수인계 뒤 문서를 찾을 때 자주 부딪히는 문제는 같은 서식의 보고서가 사업마다, 고객사마다, 연차마다 같은 제목으로 쌓여 있다는 점입니다. 중복처럼 보이는 문서의 상당수는 실제로는 서로 다른 대상에 대한 같은 종류의 기록이고, 새 저장소로 옮겨도 이 사정은 그대로 따라옵니다. 후임자는 어느 사업의 몇 년 차 보고서가 지금 질문의 근거인지 가려야 하는데, 본문이 거의 같으면 검색도 그 차이를 구분하지 못합니다. 색인할 때는 사업명, 대상, 작성 시기 같은 속성을 본문과 함께 남겨 두어야 합니다. 도구를 검토할 때도 중복이 없는 깨끗한 저장소가 아니라 중복이 있는 실제 저장소에서 문서가 구분되는지를 물어야 합니다.

1.3 범용 생성형 AI 에 사내 질문을 할 때 생기는 환각과 근거 부재

많은 조직이 이미 범용 생성형 AI를 업무에 쓰고 있으므로, 문서 검색 문제도 그 도구로 해결된다고 보는 의견이 나옵니다. 하지만 범용 생성형 AI는 사내 규정과 계약서를 학습하지 않았기 때문에, 사내 질문에 답할 근거가 없는 상태에서 그럴듯한 문장을 만들어 냅니다. 사실과 다르거나 근거 없는 내용을 그럴듯하게 만들어 내는 현상을 환각(hallucination)이라고 부릅니다. 검색으로 근거 문서를 붙이면 환각은 크게 줄지만 0이 되지는 않으며, 법률 질의를 대상으로 한 연구가 그 차이를 보여 줍니다.

도구 구분	법률 질의 환각 비율	비고
범용 챗봇	58%–82%	검색 없이 답을 생성
RAG를 쓰는 법률 AI 전문 도구	17%–34% 이상	판례·법령 검색 뒤 답을 생성

출처: S2. 2024년 프리프린트 기준이며, 2025년 동료심사판은 같은 결과를 정확도 표현으로 바꿔 실었습니다.

1.3.1 사내 문서를 모르는 범용 챗봇이 내놓는 그럴듯한 오답

법률 질의에서 측정한 환각 비율

Stanford 연구진은 실제 법률 업무에서 나올 법한 질의를 여러 AI 도구에 넣고 답의 정확성과 근거를 검증했습니다. 그 결과 범용 챗봇은 질의의 58%에서 82%에 환각이 섞인 답을 냈고, 판례와 법령을 검색해 답하는 RAG 기반 전문 도구도 17% 이상에서 34% 이상의 질의에서 환각을 보였습니다[S2]. 다만 이 연구는 2025년 동료심사를 거친 판에서 「환각」 대신 「정확도」라는 표현으로 결과를 다시 정리했으므로, 수치를 인용할 때는 측정 대상이 법률 질의라는 조건을 함께 밝혀야 합니다.

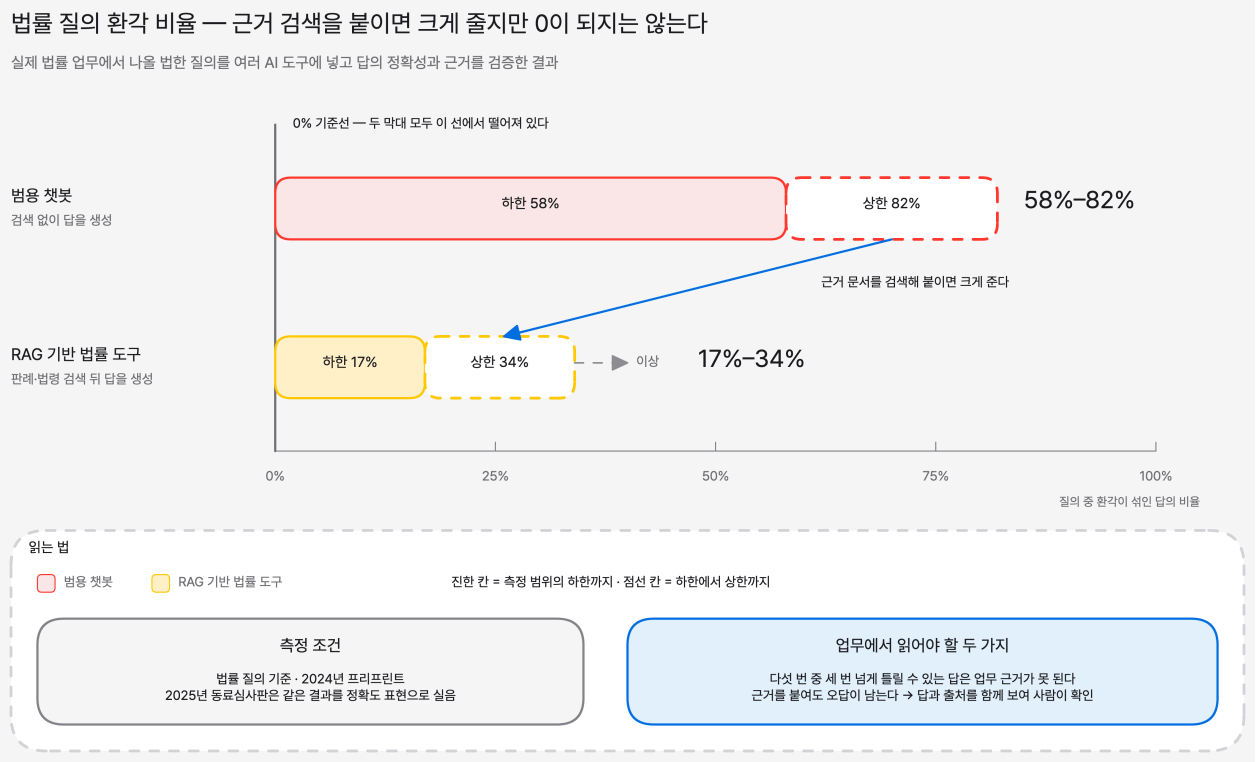

그림. 법률 질의에서 범용 챗봇과 RAG 기반 법률 도구의 환각 비율 비교

업무 근거로 쓸 수 없는 답의 조건

두 수치에서 읽어야 할 사실은 두 가지입니다. 하나는 문서를 모르는 AI의 답을 업무 근거로 쓸 수 없다는 점으로, 규정 해석이나 계약 조건 확인처럼 답이 틀리면 책임이 따르는 업무에서 다섯 번 중 세 번 넘게 틀릴 수 있는 답은 그대로 쓸 수 없습니다. 다른 하나는 근거 문서를 붙여도 오답이 남는다는 점으로, 답과 함께 근거 문서를 보여 주어 사람이 확인할 수 있게 해야 합니다. 생성형 AI를 업무에 쓰려면 사내 문서 연결과 근거 표시가 함께 갖춰져야 합니다.

1.3.2 답과 근거 문서를 함께 돌려주는 RAG 의 출발점

검색을 붙였을 때 줄어든 환각

검색 단계를 붙이는 효과는 법률 분야 밖에서도 확인됐습니다. Meta 연구진은 대화 모델이 답하기 전에 관련 문서를 검색해 참고하게 하자 환각 응답이 60% 넘게 줄었고 대화 능력은 그대로 유지됐다고 보고했습니다[S1]. 모델에 새 지식을 넣는 방법으로 파인튜닝(모델을 추가 학습시키는 방식)과 RAG를 비교한 연구에서도 RAG가 더 정확했으며, 그 결과는 2장 2.3.1절에서 자세히 다룹니다[S3]. 즉 모델이 모르는 사내 지식은 모델 안에 다시 학습시키는 방식보다 필요할 때 검색해 붙이는 방식이 더 안정적으로 작동합니다.

답과 출처를 한 단위로 돌려주는 구조

RAG의 기본 단위는 답과 출처의 묶음입니다. 사용자가 질문하면 시스템은 사내 문서에서 관련 부분을 검색하고, 찾은 근거 문서를 생성형 AI에 함께 넘겨 답을 만들게 한 다음, 답과 근거 문서의 위치를 같이 돌려줍니다. RAG는 대화를 잘하는 챗봇보다 질문에 맞는 근거를 가져오는 장치로 이해하는 편이 정확하며, 사용자는 답이 맞는지 출처를 열어 바로 확인할 수 있습니다.

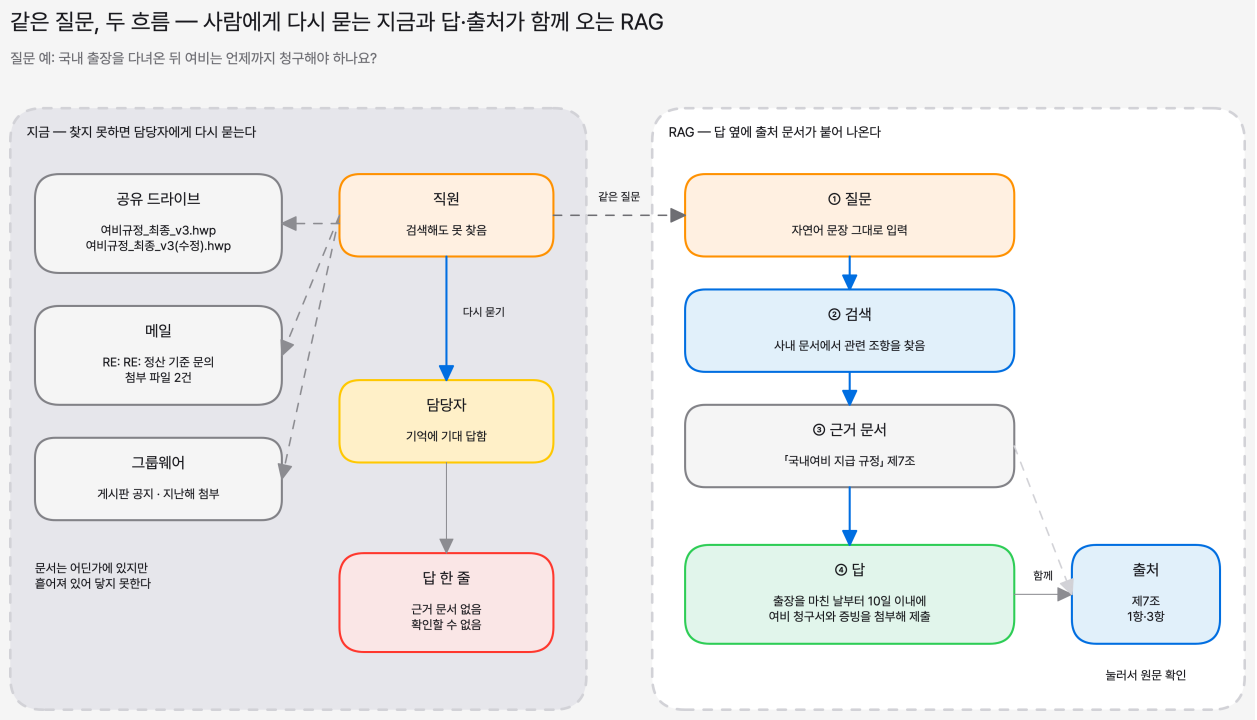

그림. 담당자에게 묻는 현재 흐름과 근거와 함께 답하는 RAG 흐름

다만 이 구조가 제대로 작동하려면 첫 단계인 검색이 질문에 맞는 문서를 찾아내야 합니다. 검색 단계가 기존 키워드 검색과 같은 방식이라면 1.1에서 본 「문서가 있는데도 닿지 못하는」 문제가 그대로 남기 때문에, 키워드 검색이 어떤 질문에서 답을 놓치고 RAG의 검색은 무엇을 바꾸는지가 다음 판단의 출발점이 됩니다.

2장: 키워드 검색이 답을 찾지 못하는 이유와 RAG 의 차이

기업과 공공기관의 사내 검색창은 대부분 키워드 검색으로 동작합니다. 키워드 검색은 질문에 들어 있는 낱말이 문서에 그대로 나오는지를 보고 문서를 고르는 방식이며, 제품 코드나 법령 조항 번호처럼 정확한 문자열을 찾을 때는 지금도 가장 믿을 만한 방법입니다[S4]. 하지만 같은 뜻을 다른 말로 쓴 문서는 찾지 못하고, 맞는 문서를 찾은 뒤에도 문서 목록만 돌려주기 때문에 답을 읽어 내는 일은 여전히 직원에게 남습니다. RAG는 질문과 뜻이 가까운 문서 조각을 찾아 언어 모델에 건네고, 모델이 그 조각만 근거로 답을 쓰게 한 뒤 답마다 출처를 붙이는 구성입니다. RAG는 기존 검색을 걷어 내지 않고 그 위에 의미 검색과 답변 생성을 더하는 기술이며, 키워드 검색은 5장의 하이브리드 검색에서 의미 검색과 짝을 이뤄 다시 쓰입니다. 아래 내용은 조직의 검색 실패가 표현 차이에서 오는지, 찾은 문서를 읽고 엮는 일에서 오는지를 가려 보고, RAG·파인튜닝·긴 문맥 입력 가운데 어느 방식이 문서 양과 변경 주기에 맞는지 판단하는 기준을 제공합니다.

2.1 같은 낱말이 있어야 찾는 키워드 검색의 구조적 한계

키워드 검색 엔진은 문서를 낱말 단위로 쪼개 어느 문서에 어떤 낱말이 몇 번 나오는지 목록으로 만들어 두고, 질문의 낱말이 자주 나오면서 다른 문서에는 드물게 나오는 문서일수록 높은 점수를 줍니다. 이 점수 계산식 가운데 가장 널리 쓰이는 것이 BM25이며, 18개 데이터셋으로 검색 방식을 비교한 BEIR 평가에서 BM25는 별도 학습 없이도 튼튼한 기준선이었고 의미 검색 모델은 학습한 분야 밖의 문서에서 BM25보다 못한 경우가 많았습니다[S4]. 따라서 키워드 검색은 버릴 대상이 아니며, 문제는 낱말 일치라는 원리 자체에서 생기는 두 가지 한계입니다. 하나는 같은 뜻을 다른 낱말로 쓴 문서를 놓치는 표현 차이이고, 다른 하나는 찾은 결과가 답이 아닌 문서 목록이라는 점입니다.

2.1.1 같은 뜻을 다른 말로 쓴 문서를 놓치는 표현 차이 문제

낱말 일치 원리가 만드는 검색 누락

직원이 「출장비 정산 방법」을 검색했는데 규정집에는 「국내여비 지급」과 「여비 청구」라는 표현만 있다면, 두 문장 사이에 공통 낱말이 하나도 없으므로 키워드 검색은 그 규정에 점수를 주지 못합니다. 사람은 출장비와 여비가 같은 대상을 가리킨다는 것을 알지만, 검색 엔진은 글자가 같은지만 비교하기 때문입니다. 규정 문서가 분명히 있는데도 결과 화면에는 「출장」이라는 낱말이 들어간 공지나 행사 안내가 먼저 나옵니다. 이런 실패의 상당 부분은 낱말 일치라는 검색 방식에서 생기며, 직원의 검색 실력만으로는 해결하기 어렵습니다.

동의어·약어·부서별 용어에서 생기는 세 가지 유형

표현 차이는 대체로 세 갈래로 나타납니다. 출장비와 여비처럼 뜻이 같은 동의어가 있고, VPN처럼 약어로 부르는 대상을 문서에서는 「원격 접속 서비스」로 풀어 쓰는 약어 차이가 있으며, 같은 절차를 영업팀은 「품의」로, 구매팀은 「구매 요청 결재」로 부르는 부서별 용어 차이가 있습니다. 공공기관에서는 이 간격이 더 넓어지는데, 규정과 지침은 「복무」나 「계약 체결」 같은 법령 용어로 쓰이고 직원과 민원인은 일상어로 묻기 때문입니다. 아래 표는 행정 업무에서 자주 보이는 질의 네 건을 키워드 검색에 넣었을 때 어떤 결과가 나오는지 정리한 것입니다.

직원이 입력한 질의	문서에 실제로 쓰인 표현	키워드 검색 결과
출장비 정산 방법	국내여비 지급 규정, 여비 청구서	공통 낱말이 없어 규정을 찾지 못함

직원이 입력한 질의	문서에 실제로 쓰인 표현	키워드 검색 결과
휴가 쓰려면 누구 결재	연차 유급휴가 사용 신청, 복무 관리 지침	어절 단위로 나누면 「유급휴가」와 「휴가」가 달라 지침이 뒤로 밀림
VPN 신청	원격 접속 서비스 이용 신청서	약어와 풀어 쓴 이름이 달라 신청서를 찾지 못함
품의 올리는 법	구매 요청 결재 절차(구매팀 문서)	부서별 용어 차이로 절차 문서를 찾지 못함

동의어 사전을 만들어 검색 엔진에 넣으면 일부는 보완할 수 있습니다. 하지만 사전은 부서와 문서가 늘어날 때마다 사람이 계속 고쳐야 하고, 사전에 오르지 않은 새 표현은 여전히 놓칩니다. 한국어에는 「계약서를」과 「계약서의」처럼 조사가 붙어 낱말의 형태가 바뀌는 문제가 하나 더 있으며, 이 부분은 형태소 분석으로 낱말을 원형에 가깝게 나누면 상당히 줄어듭니다(4장). 표현 차이를 근본적으로 줄이려면 글자의 일치보다 뜻의 가까움을 비교하는 검색이 필요합니다.

2.1.2 문서 목록만 돌려주고 읽기는 사람에게 남기는 검색 결과

문서 목록과 답변 사이에 남는 사람의 일

키워드 검색이 맞는 문서를 찾아 주더라도 결과물은 제목과 짧은 발췌가 붙은 문서 목록 열 건 정도입니다. 직원은 그중 몇 건을 열어 해당 조항을 찾아 읽고, 개정 전후 문서의 내용이 다르면 어느 쪽이 최신인지 가려야 하며, 여러 문서에 흩어진 조건을 직접 엮어 답을 만듭니다. 1장 1.1절에서 살펴본 문서 찾기 비용은 대부분 이처럼 「찾기」와 「답하기」 사이에서 사람이 하는 일에서 생깁니다. 검색 품질을 높이는 것만으로는 이 비용이 사라지지 않으며, 찾은 내용을 읽어 답으로 바꾸는 단계까지 함께 다뤄야 합니다.

긴 입력의 중간에 묻힌 정보를 놓치는 현상

그렇다고 찾은 문서를 통째로 언어 모델에 넣는 방식이 해법이 되지는 않습니다. 여러 문서를 이어 붙인 긴 입력에서 질문과 관련된 정보가 처음이나 끝이 아닌 중간에 놓이면 언어 모델의 답변 정확도가 크게 떨어진다는 연구 결과가 있습니다[S14]. 이 현상은 긴 보고서를 훑는 사람에게서도 흔히 나타나는데, 분량이 늘어날수록 중간 내용에 주의가 덜 가기 때문입니다. 이 때문에 RAG는 문서 전체를 넘기는 대신 질문과 관련된 조각만 골라 짧게 건네는 쪽으로 설계됩니다.

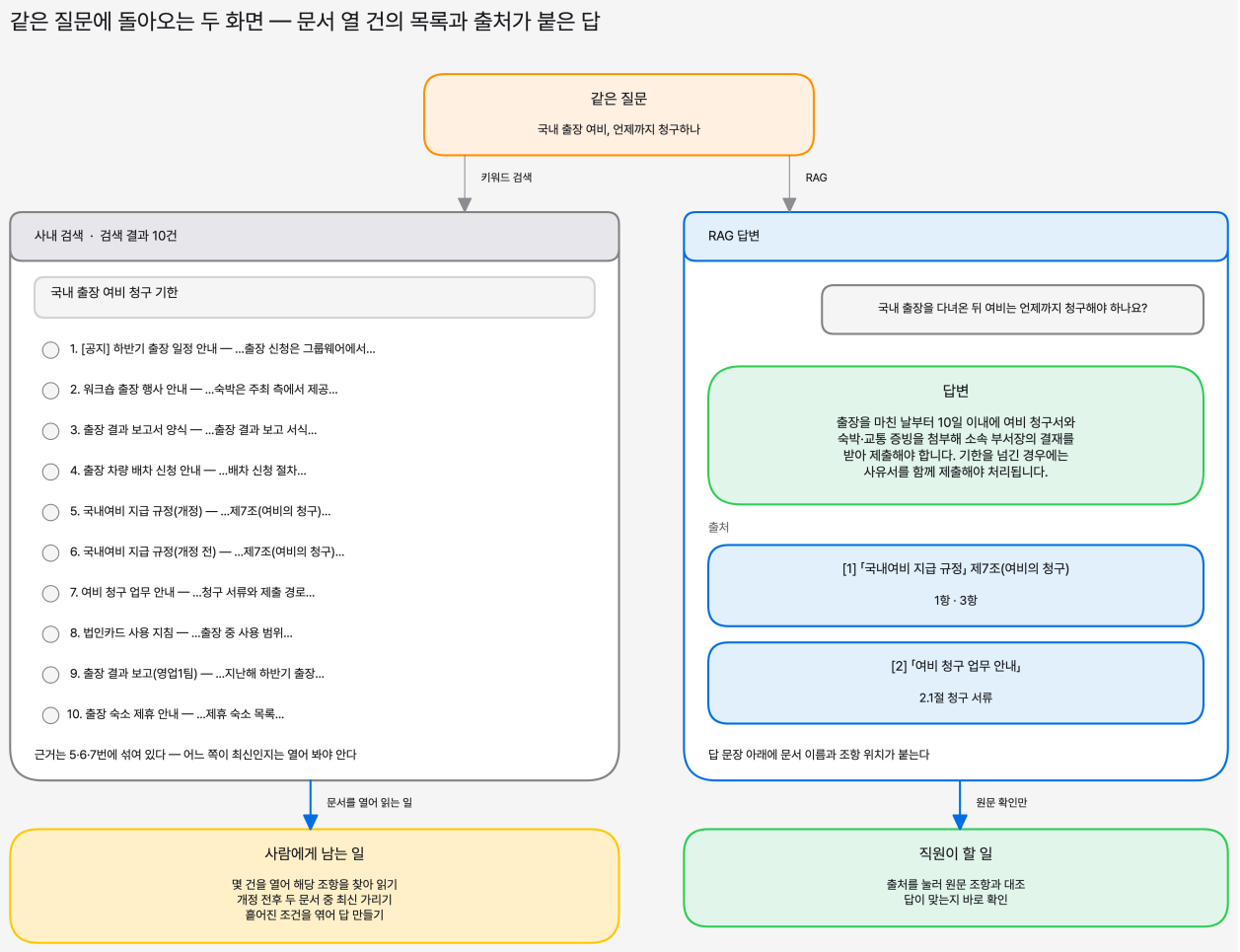

그림. 키워드 검색 결과 화면과 RAG 답변 화면의 비교

두 방식의 차이를 항목별로 정리하면 다음과 같습니다. 찾는 단위와 결과물이 다르다는 점이 업무 시간에 가장 직접적인 영향을 줍니다.

비교 항목	키워드 검색	RAG
입력	낱말 조합	자연어 질문
찾는 기준	낱말 일치와 출현 빈도	뜻의 가까움(필요하면 낱말 일치를 함께 사용)
찾는 단위	문서 전체	문단·조항 크기의 문서 조각
결과물	문서 목록	답변 문장과 근거 조각
근거 표시	문서 제목과 링크	답변마다 출처 문서 이름과 위치
강한 질문	제품 코드, 조항 번호, 고유명	표현이 다른 설명형 질문

2.2 뜻으로 찾아 근거와 함께 답하는 RAG 의 처리 흐름

RAG의 처리는 문서를 미리 손질해 두는 준비 단계와 질문이 들어올 때마다 실행되는 답변 단계로 나뉩니다. 준비 단계는 파싱·청킹·임베딩·색인의 네 과정으로 이뤄지고 문서가 추가되거나 바뀔 때마다 다시 실행되며, 답변

단계는 질문 변환과 조각 검색, 근거 기반 답변 생성의 세 과정으로 이뤄집니다. 각 과정의 입력과 출력이 분명하기 때문에 답이 틀렸을 때 문서를 잘못 읽었는지, 조각을 잘못 찾았는지, 모델이 조각을 잘못 옮겼는지를 단계별로 확인할 수 있습니다. RAG는 IT 담당자가 과정마다 품질을 재고 고칠 수 있는 구성이며, 내부를 들여다볼 수 없는 장치로 볼 필요가 없습니다.

준비 단계와 답변 단계의 구성 요소를 도입 점검 항목과 함께 짧게 정리한 글은 함께 읽을 자료 「RAG란 무엇인가 — 파이프라인 해부와 도입 체크리스트」[R2]에 있습니다.

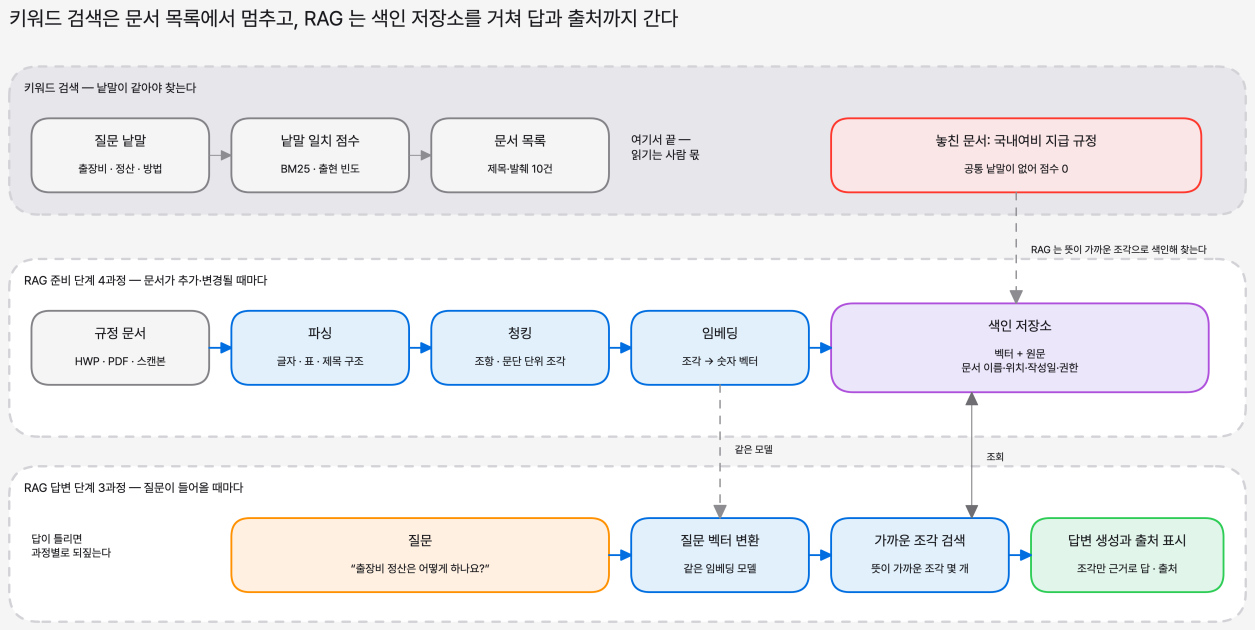

그림. 키워드 검색 흐름과 RAG 처리 흐름(준비 4단계, 답변 3단계)의 비교

2.2.1 문서를 조각으로 나눠 의미 벡터로 색인하는 준비 단계

파싱·청킹·임베딩·색인의 네 과정

준비 단계는 HWP·PDF·스캔본 같은 문서 파일에서 글자와 표를 꺼내는 파싱으로 시작해, 꺼낸 글을 알맞은 길이로 자르는 청킹, 조각의 뜻을 숫자 목록으로 바꾸는 임베딩, 그 결과를 찾기 쉽게 저장하는 색인으로 이어집니다. 임베딩은 문장의 뜻을 좌표로 바꾼 것이라고 이해하면 쉬운데, 뜻이 비슷한 문장은 좌표 공간에서 서로 가까운 위치에 놓이므로 「출장비 정산」과 「여비 청구」처럼 낱말이 달라도 가까운 조각으로 찾을 수 있습니다. 각 과정이 하는 일과 그 과정이 답의 품질에 주는 영향은 아래 표와 같습니다.

과정	하는 일	품질에 주는 영향
파싱(parsing)	문서 파일에서 글자·표·제목 구조를 읽어 냄	표가 깨지거나 스캔본 글자를 읽지 못하면 이후 과정이 그 내용을 알 수 없음
청킹(chunking)	글을 조항·문단 단위의 조각으로 자름	조각이 너무 길면 관련 없는 내용이 섞이고, 너무 짧으면 앞뒤 맥락이 끊김
임베딩(embedding)	조각의 뜻을 수백에서 수천 개의 숫자로 된 벡터로 바꿈	한국어를 잘 이해하는 모델인지에 따라 뜻의 가까움 판단이 크게 달라짐(4

과정	하는 일	품질에 주는 영향
		장)
색인(indexing)	벡터와 원문, 문서 이름·위치·작성일·권한 정보를 함께 저장	출처 정보를 함께 저장해야 답변에 출처를 붙이고 권한별로 걸러 낼 수 있음

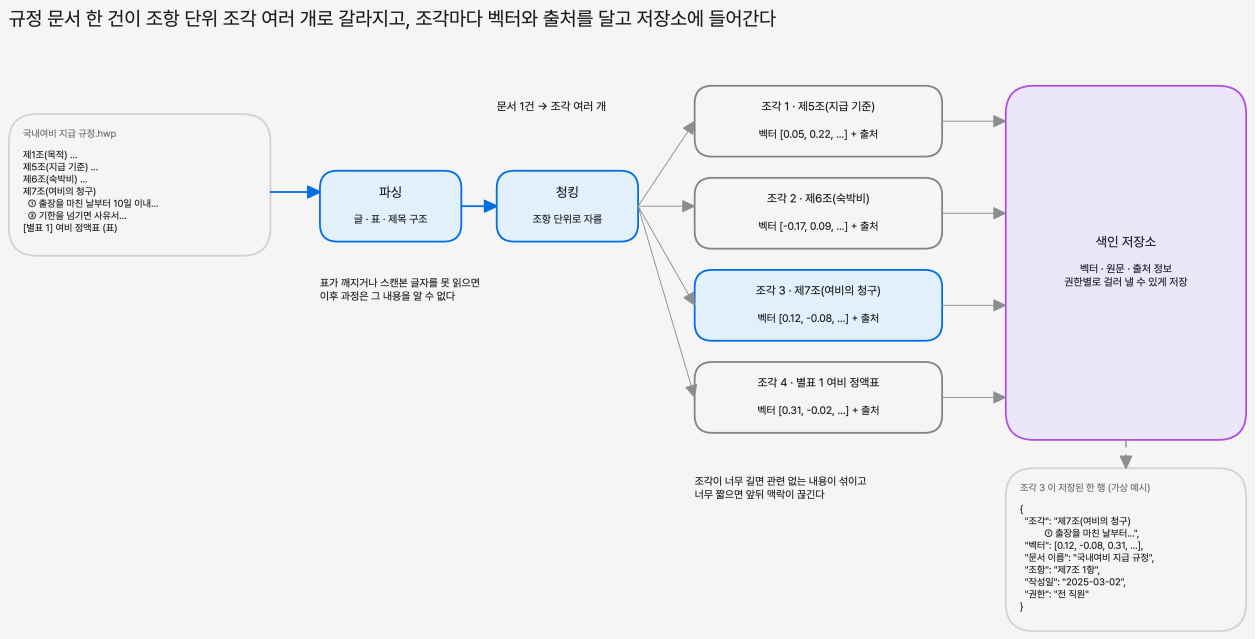

그림. 문서를 조각으로 나눠 벡터로 색인하는 준비 단계

조각 나누는 방식에 따라 달라지는 검색 결과

청킹은 단순한 전처리처럼 보이지만 검색 결과를 직접 바꿉니다. 예를 들어 「제3조(지급 기준)」라는 조각만 떼어 보면 그것이 여비 규정인지 복리후생 규정인지 알 수 없으므로, 질문과 뜻이 맞아도 다른 규정의 같은 조항과 구별되지 않습니다. 각 조각에 원래 문서의 어느 부분인지 설명하는 짧은 문맥을 덧붙여 임베딩했더니 상위 20개 검색 결과에서 정답 조각을 놓치는 비율이 35% 줄었다는 실험 결과가 있으며, 다른 구성 요소를 더했을 때의 결과는 5장 5.3.1절에서 다룹니다[S6]. RAG의 답변 품질은 언어 모델을 고르기 전에 준비 단계에서 먼저 정해지므로, 도입을 검토할 때 문서 정리와 파싱 품질 점검을 모델 선택보다 앞에 두는 편이 효과적입니다.

2.2.2 질문과 가까운 조각을 찾아 출처와 함께 답하는 응답 단계

질문 벡터 변환부터 출처 표시까지의 세 과정

답변 단계에서는 먼저 직원의 질문을 준비 단계와 같은 임베딩 모델로 벡터로 바꾸고, 색인에서 그 벡터와 가장 가까운 조각을 몇 개에서 수십 개 정도 찾아냅니다. 그다음에는 질문과 찾은 조각을 함께 언어 모델에 넣으면서 「아래 자료에 근거해서만 답하고, 자료에 없으면 모른다고 답하라」는 지시를 붙입니다. 마지막으로 모델이 쓴 답변의 문장마다 어느 조각에서 왔는지를 표시해, 직원이 출처 문서를 눌러 원문을 바로 확인할 수 있게 합니다. 이때 같은 임베딩 모델을 써야 질문과 문서 조각이 같은 좌표 공간에 놓여 가까움을 비교할 수 있습니다.

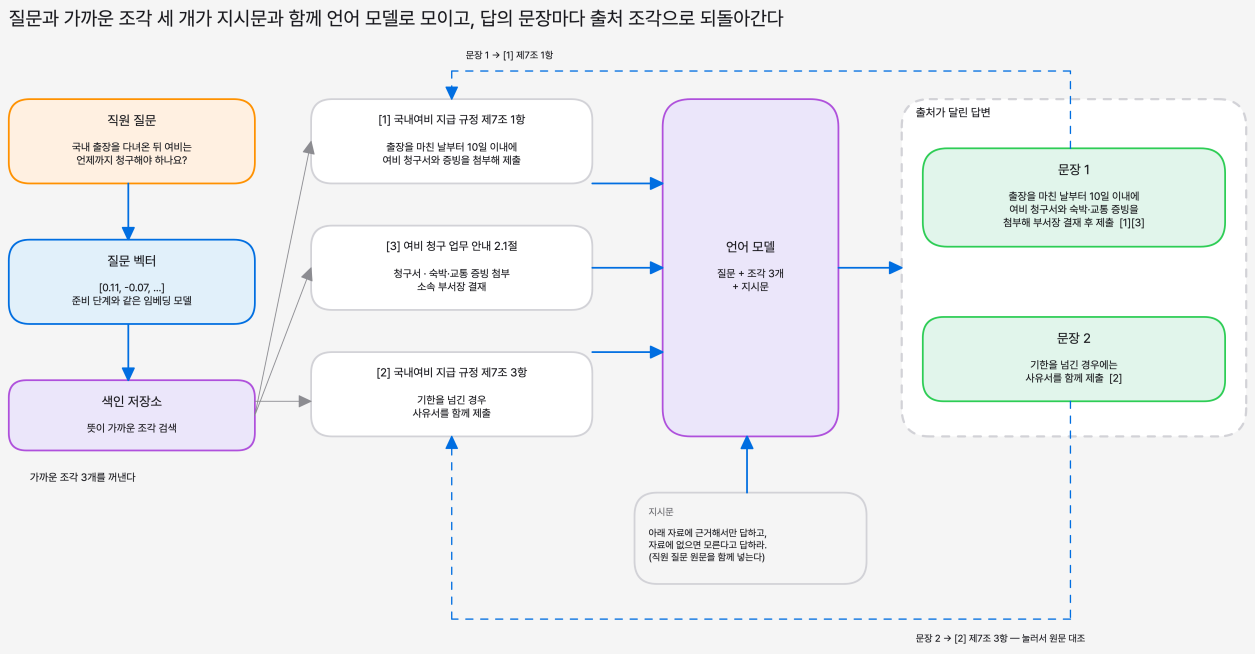

그림. 질문과 가까운 조각을 찾아 출처와 함께 답하는 응답 단계

아래는 설명을 위해 만든 가상의 규정과 답변 예시입니다. 답변 아래에 출처 문서 이름과 조항 위치가 함께 붙는다는 점이 문서 목록만 주는 검색 결과와 가장 다른 부분입니다.

> ***질문**: 국내 출장을 다녀온 뒤 여비는 언제까지 청구해야 하나요?*
>
> ***답변**: 출장을 마친 날부터 10일 이내에 여비 청구서와 숙박·교통 증빙을 첨부해 소속 부서장의 결재를 받아 제출해야 합니다. 기한을 넘긴 경우에는 사유서를 함께 제출해야 처리됩니다.*
>
> ***출처**: 「국내여비 지급 규정」 제7조(여비의 청구) 1항·3항, 「여비 청구 업무 안내」 2.1절*

출처 표시가 업무 신뢰의 조건이 되는 이유

언어 모델은 학습한 내용을 바탕으로 그럴듯한 문장을 만들어 내는 성질이 있어서, 사내 규정처럼 학습 때 본 적 없는 내용을 물으면 사실과 다른 답을 자신 있게 쓰기도 합니다. RAG는 근거 조각을 입력에 넣어 모델이 기억 대신 자료를 옮겨 쓰게 합니다. 다만 1장 1.3절에서 본 것처럼 검색을 붙이면 환각이 크게 줄지만 사라지지는 않으며, 법률 질의 평가에서는 RAG 기반 전문 도구에도 환각이 남았습니다[S1][S2]. 출처 표시는 직원이 답을 원문과 대조해 확인할 수 있게 하는 필수 장치로 설계해야 하며, RAG가 실패하는 경우와 그 보완 기술은 6장에서 정리합니다.

2.3 파인튜닝·긴 문맥 입력과 비교한 RAG 의 쓰임새

RAG를 검토하는 조직에서는 「사내 문서로 모델을 학습시키면 되지 않느냐」와 「요즘 모델은 긴 글을 읽으니 문서를 통째로 넣으면 되지 않느냐」는 두 가지 반론이 자주 나옵니다. 두 방식 모두 실제로 쓰이는 기술이며, 어느 쪽이 맞는지는 유행보다 업무 지식의 양과 문서가 바뀌는 주기, 근거 표시가 필요한지, 운영 조직이 감당할 수 있는 부담으로 판단해야 합니다. 세 방식을 같은 기준으로 비교하면 다음과 같습니다.

판단 기준	RAG	파인튜닝	긴 문맥 입력
알맞은 지식의 양	계속 늘어나는 대규모 문서 모음	학습 데이터로 정리할 수 있는 범위	약 20만 토큰 미만의 작은 문서 묶음[S6]
문서가 바뀔 때	바뀐 문서만 다시 색인	다시 학습하고 다시 평가	질문할 때 최신 문서를 넣으면 반영
근거 표시	답변마다 출처 조각 표시 가능	답이 어느 문서에서 왔는지 가리기 어려움	넣은 문서는 알지만 문서 안 위치는 별도 지시가 필요
운영 부담	파싱·색인·검색 품질 관리	학습 데이터 준비, 재학습, 평가	질문마다 많은 토큰을 처리해 응답 시간과 계산량이 늘어남
잘 맞는 용도	규정·매뉴얼·상담 이력 질의응답	답변 말투, 보고서 형식, 분야 용어 사용 습관	계약서 한 건 검토처럼 작은 문서 묶음 분석

2.3.1 지식 주입 실험에서 비지도 파인튜닝보다 앞선 RAG 의 결과

비지도 파인튜닝과 RAG를 비교한 지식 주입 실험

비지도 파인튜닝은 문서 원문을 모델에 추가로 읽혀 그 내용을 모델 내부의 가중치에 반영하는 방식이고, RAG는 모델은 그대로 두고 질문할 때 관련 조각을 찾아 입력에 넣는 방식입니다. Ovadia 외(Microsoft) 연구진은 언어 모델이 학습 때 보지 못한 새 지식을 넣는 방법으로 두 방식을 여러 주제의 질문으로 비교했고, RAG가 비지도 파인튜닝보다 일관되게 높은 정확도를 보였다고 보고했습니다[S3]. 모델이 모르는 사내 지식을 답하게 하려는 목적이라면 모델을 고치는 것보다 필요할 때 자료를 건네는 편이 더 정확했습니다.

비교 항목	비지도 파인튜닝	RAG
지식을 넣는 방법	문서 원문을 추가 학습해 가중치에 반영	질문 때 관련 조각을 찾아 입력에 넣음
새 지식 질문의 정확도	비교 조건 전반에서 RAG보다 낮음	비교 조건 전반에서 일관되게 높음
문서가 바뀔 때	다시 학습	바뀐 문서만 다시 색인

자주 바뀌는 사내 문서에서 드러나는 운영상 장점

정확도 외에 운영 측면의 차이도 큽니다. 사내 규정은 매년 개정되고 상품 안내와 업무 매뉴얼은 분기마다 바뀌는데, 파인튜닝은 문서가 바뀔 때마다 학습 데이터를 다시 만들고 재학습과 평가를 거쳐야 합니다. 반면 RAG는 개정된 문서만 다시 색인하면 다음 질문부터 바로 반영되고, 폐기된 문서를 색인에서 빼는 것만으로 낡은 답을 막을 수 있으며, 문서 단위로 열람 권한을 걸러 낼 수도 있습니다. 다만 파인튜닝이 맞는 경우도 분명히 있는데, 답변 말투나 보고서 형식, 분야 용어를 쓰는 습관처럼 지식보다 행동 방식을 가르칠 때입니다. 두 방식을 경쟁 관계로 보기보다 지식은 RAG로 건네고 형식은 파인튜닝으로 맞추는 식으로 함께 쓰는 구성도 검토할 만합니다.

2.3.2 문서량 약 20만 토큰 미만에서 통째로 넣는 방식과 라우팅

문서 양에 따라 달라지는 권장 방식

토큰은 언어 모델이 한 번에 읽고 처리하는 글자 묶음이며, 모델이 한 번에 받을 수 있는 입력 길이와 계산량이 모두 토큰 수로 정해집니다. 업무 지식의 양이 약 20만 토큰 미만이라면 검색 구성을 따로 만들기보다 문서 전체를 입력에 넣는 편이 더 간단하고 나은 결과를 낸다는 권고가 있습니다[S6]. 규정집 몇 권이나 제품 설명서 몇 종처럼 범위가 작고 잘 바뀌지 않는 업무라면 RAG가 오히려 과한 구성일 수 있다는 뜻입니다. 한편 한국어는 같은 내용이라도 토크나이저에 따라 영어보다 토큰이 더 많이 나올 수 있으므로, 이 기준을 적용할 때는 실제 문서로 토큰 수를 세어 보는 것이 안전합니다(4장).

RAG와 긴 문맥을 섞어 쓰는 라우팅

두 방식을 하나만 고를 필요도 없습니다. Li 외(Google DeepMind) 연구진은 모델이 먼저 찾은 조각만으로 답할 수 있는지 스스로 판단하게 하고, 답할 수 없을 때만 긴 문맥 입력으로 넘기는 Self-Route 라우팅을 제안했습니다. 이 방식은 질의의 82%를 RAG로 처리하면서 긴 문맥 입력만 쓸 때와 비슷한 성능을 유지했고, 계산 비용은 39%에서 65%까지 줄였습니다[S13]. 대부분의 질문은 조각 몇 개로 충분하고 문서 전체를 엮어야 하는 질문은 일부라는 점을 구성에 반영하면 성능과 비용을 함께 맞출 수 있습니다. 문서 양에 따른 권장 방식을 정리하면 다음과 같습니다.

업무 지식의 양과 성격	권장 방식	근거
약 20만 토큰 미만이고 잘 바뀌지 않음	문서 전체를 넣는 긴 문맥 입력	[S6]
약 20만 토큰 이상이거나 계속 늘어남	RAG	[S6]
대부분은 조각으로 답하되 일부 질문은 문서 전체를 엮어야 함	RAG를 기본으로 하고 긴 문맥 입력으로 넘기는 라우팅	[S13]

이 기준은 8장의 파일럿 범위를 정할 때 그대로 쓰입니다. 파일럿 대상 업무를 고르실 때 먼저 그 업무의 문서 양과 개정 주기를 재 보시면, RAG가 필요한 업무와 문서를 통째로 넣어도 되는 업무가 자연스럽게 갈라집니다. RAG가 필요한 업무로 판단되면 같은 RAG라도 한국어 임베딩 선택과 하이브리드 검색·리랭커 구성에 따라 정답률이 크게 달라지므로 그 설계가 다음 과제가 됩니다(4장, 5장).

3장: 업무별로 보는 RAG 활용과 여덟 개 업무의 도입 효과

2장에서 본 RAG의 구조가 실제 업무에서 어떤 결과를 냈는지는 공개 사례로 확인할 수 있습니다. 2024년 이후 금융 자문, 인사 문의, 장애 대응, 고객 상담, 법령 안내처럼 문서 의존도가 높은 업무에서 기업과 정부 기관이 이름을 밝히고 도입 결과를 공개하기 시작했습니다[S16][S20][S29]. 다만 공개된 수치는 업무마다 측정 기준이 다르고, 마케팅과 구매처럼 이름 있는 기업의 효과 수치가 아직 나오지 않은 업무도 있습니다. 이 장은 여덟 개 업무를 지금의 불편, 색인할 문서와 질문 예시, 기대 효과, 공개 레퍼런스의 네 칸으로 같은 순서에 맞춰 정리했습니다. 업무별로 문서 찾기에 드는 시간을 견주어 보면 8장에서 고를 파일럿 후보를 두세 개로 좁힐 수 있습니다.

3.1 마케팅과 영업에서 고객 질문에 근거 자료로 답하는 RAG

마케팅과 영업은 고객을 직접 상대하는 업무라서 자료를 찾는 데 걸린 시간이 그대로 응답 속도와 기회 손실로 이어집니다. 두 업무 모두 캠페인 보고서, 상품 설명서, 과거 제안서처럼 이미 만들어 둔 자료가 많지만, 필요한 순간에 찾지 못하면 새로 쓰거나 답을 미루게 됩니다. 매출과 가까운 업무에서 RAG 효과를 먼저 확인하려는 조직은 이 두 업무를 함께 살펴볼 만합니다.

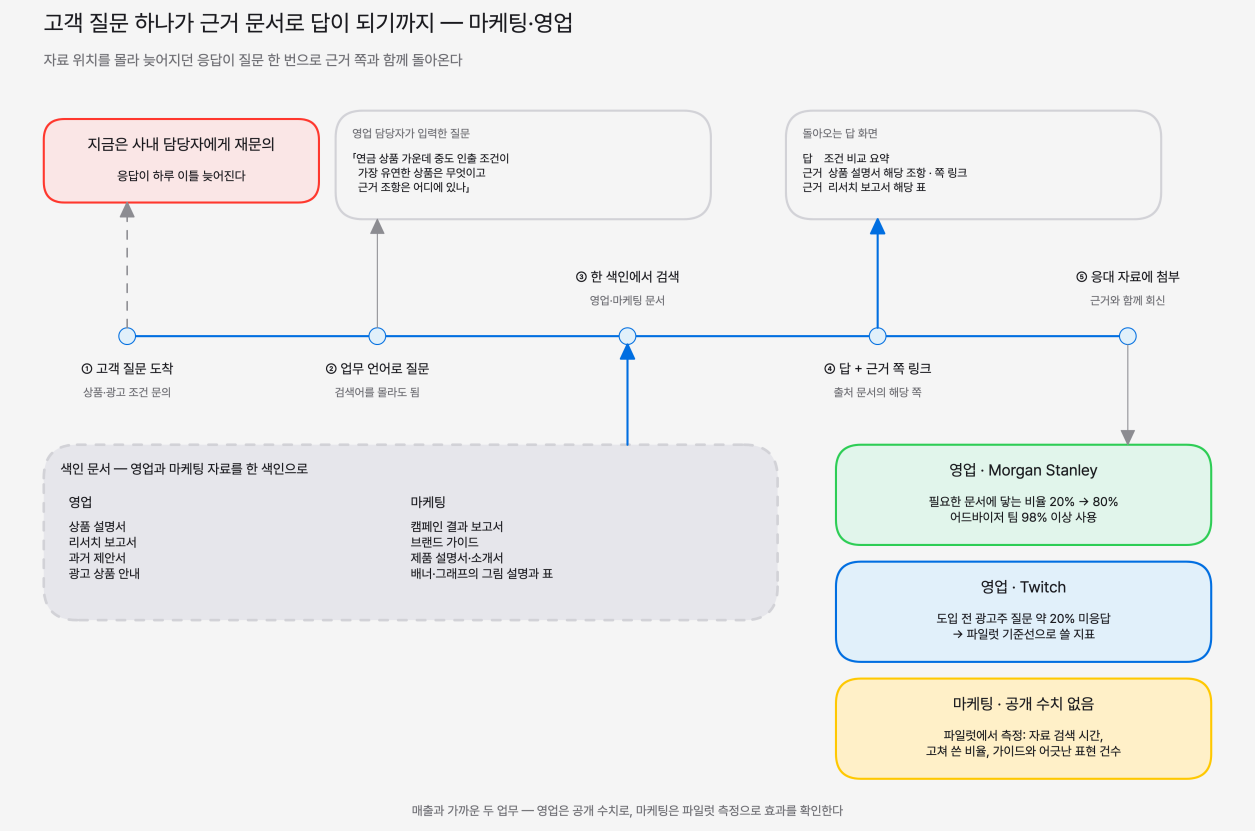

그림. 마케팅·영업 — 고객 질문 하나가 근거 문서로 답이 되기까지의 흐름과 공개 레퍼런스

3.1.1 캠페인 자료와 브랜드 가이드를 다시 쓰게 만드는 마케팅 지식 검색

캠페인 자료 재작성의 불편

마케팅 조직에서 가장 흔한 낭비는 이미 있는 자료를 찾지 못해 비슷한 문서를 다시 만드는 일입니다. 지난 캠페인의 결과 보고서는 공유 드라이브의 연도별 폴더에, 브랜드 문구 규칙은 디자인 팀의 별도 가이드 파일에, 제품 소개 자료는 영업 부서의 메일 첨부에 흩어져 있는 경우가 많습니다. 이때 담당자는 파일 이름으로 키워드 검색을 해 보다가 찾지 못하면 기억에 의존해 문구를 새로 쓰고, 그 결과 같은 제품을 부서마다 다른 표현으로 소개하게 됩니다. 마케팅에서 문서 찾기 비용은 작업 시간뿐 아니라 브랜드 표현의 일관성에도 영향을 줍니다.

마케팅 문서의 색인 범위와 질문 예시

RAG를 적용할 때는 캠페인 결과 보고서, 브랜드 가이드, 제품 설명서와 소개서를 하나의 색인으로 묶습니다. 이 문서들은 배너 시안, 화면 캡처, 성과 그래프처럼 이미지 비중이 높기 때문에 그림 설명과 표 내용까지 함께 색인해야 답의 근거가 빠지지 않습니다(3.5절). 질문은 마케팅 담당자가 업무 언어 그대로 입력하며, 답에는 근거가 된 보고서 페이지 링크가 함께 붙습니다.

- 「작년 하반기 공공기관 대상 웨비나 안내 메일 가운데 반응이 좋았던 제목과 그 결과는 무엇이었나」
- 「브랜드 가이드에서 제품명 뒤에 붙이지 말라고 정한 표현은 무엇인가」
- 「신제품 소개 문구에 인용할 수 있는 고객 사례는 어느 보고서에 있나」

마케팅에서 기대하는 효과와 측정 항목

기대 효과는 이미 만든 캠페인 자료를 다시 쓰는 비율이 늘고 부서마다 달랐던 제품 표현이 하나로 맞춰지는 것입니다. 부서마다 흩어진 사내 지식을 하나의 지식 기반으로 모은 IskayPet 사례처럼, 첫 단계는 자료를 한 색인 안에 모으는 일입니다[S26]. 하지만 마케팅 업무에서 RAG 도입 효과를 수치로 공개한 이름 있는 기업 사례는 아직 확인되지 않으므로, 다른 업무의 수치를 빌려 와 기대치를 정하면 판단이 틀어집니다. 파일럿에서 자료 검색에 걸린 시간, 기존 자료를 고쳐 쓴 비율, 브랜드 가이드와 어긋난 표현 건수를 도입 전후로 재는 방식이 현실적입니다.

3.1.2 어드바이저 문서 접근률 20%→80% 와 광고 영업 질문 대응 사례

영업 응답을 늦추는 자료 탐색

영업 담당자가 고객 질문에 바로 답하지 못하는 이유는 대개 지식 부족보다 자료 위치를 모르는 데 있습니다. 상품 설명서, 리서치 보고서, 이전 제안서, 광고 상품 안내는 분량이 많고 자주 바뀌기 때문에, 고객 앞에서 필요한 조항이나 표를 찾으려면 사내 담당자에게 다시 물어야 합니다. 응답이 하루 이틀 늦어지고, 그사이 경쟁 제안이 먼저 도착하는 경우가 생깁니다.

영업 문서의 색인 범위와 질문 예시

적용 범위는 상품 설명서, 리서치 보고서, 과거 제안서, 광고 상품 안내 문서입니다. 영업 담당자는 고객 질문을 받는 즉시 아래와 같이 묻고, 답과 함께 근거 문서의 해당 쪽을 고객 응대 자료에 붙입니다.

- 「연금 상품 가운데 중도 인출 조건이 가장 유연한 상품은 무엇이고 근거 조항은 어디에 있나」
- 「지난 분기 제조업 고객에게 낸 제안서에서 쓴 도입 일정 표를 찾아 달라」
- 「동영상 광고 상품에서 고를 수 있는 타깃팅 옵션과 노출 위치는 무엇인가」

Morgan Stanley 어드바이저 문서 접근률

Morgan Stanley는 자산관리 어드바이저가 사내 리서치와 상품 문서를 질문으로 찾는 도구를 도입했습니다. 어드바이저 팀의 98% 이상이 이 도구를 쓰고 있으며, 필요한 문서에 닿는 비율이 20%에서 80%로 올랐다고 밝혔습니다[S16]. 이 수치는 언어 모델 공급사가 고객 사례로 낸 회사 발표 기준이므로, 측정 방식은 자기 조직에서 다시 확인해야 합니다.

Twitch 광고 영업의 미응답 질문

Twitch는 광고 영업 과정에서 광고주가 보낸 질문 가운데 약 20%가 답을 받지 못하는 문제를 겪었습니다[S17]. 이 문제를 줄이려고 광고 상품 안내 자료를 검색해 영업 담당자에게 답 초안을 주는 구성을 만들었고, 검색 단계에 에이전트 흐름을 결합했습니다. 다만 공개 자료는 도입 전 문제를 수치로 밝혔을 뿐 도입 후 미응답 비율을 같은 기준으로 제시하지 않으므로, 도입 전 미응답 비율은 파일럿의 기준선으로 쓰기에 알맞은 지표입니다. 두 사례 모두 클라우드 사업자와 모델 공급사의 기술 블로그에 실린 회사 발표입니다.

3.2 회계·재무와 인사에서 규정과 기준 문서를 찾는 반복 질의

회계·재무와 인사는 규정, 기준서, 정책 문서가 많고 같은 질문이 직원 수만큼 반복되는 관리 업무입니다. 질문 대부분은 이미 문서에 답이 있지만 담당자가 문서를 찾아 읽고 다시 설명하는 데 시간을 씁니다. 내부 지원 부서의 반복 응대를 줄이는 RAG는 효과를 가장 빨리 확인할 수 있는 영역에 속합니다.

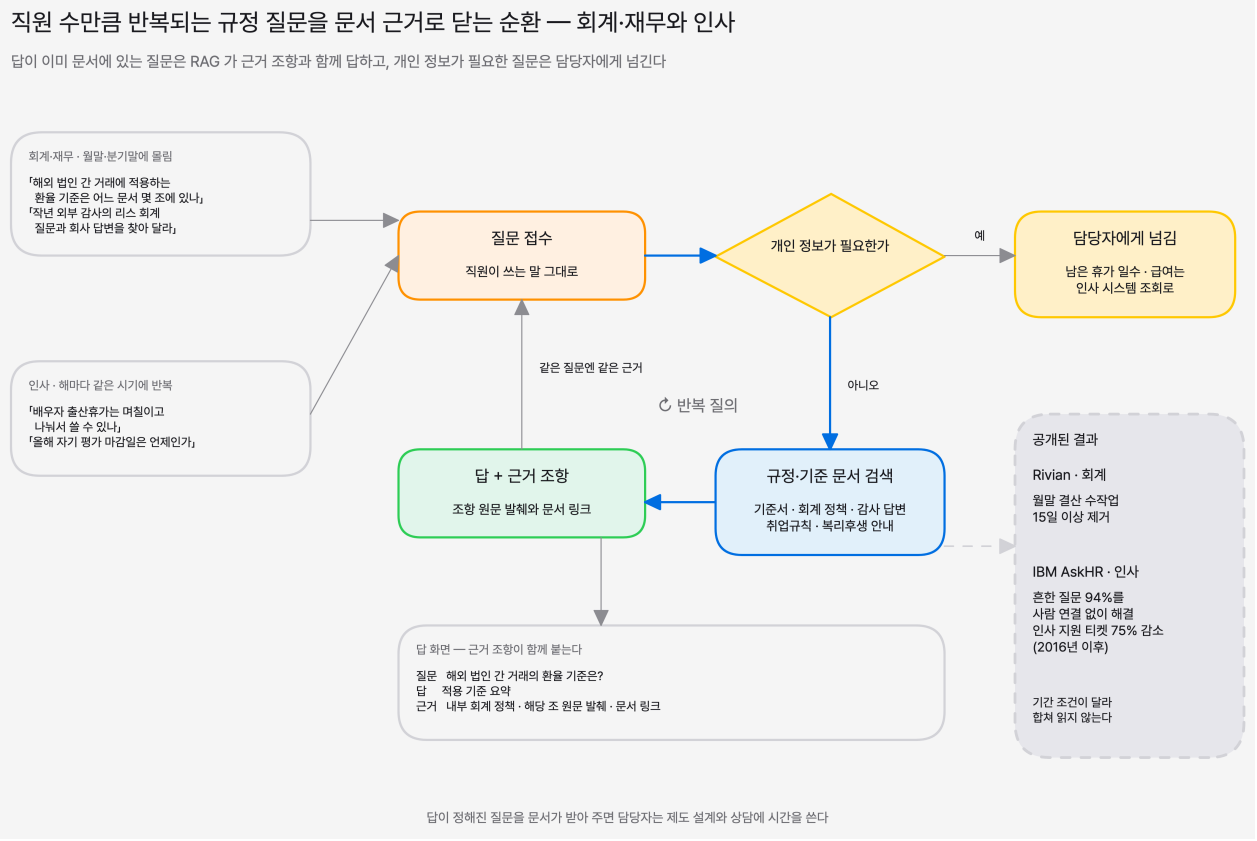

그림. 회계·재무와 인사 — 반복되는 규정 질문을 근거 조항으로 처리하고 개인 정보 질문은 담당자에게 넘기는 순환

3.2.1 월말 결산 15일 이상의 수작업을 없앤 회계·재무 문서 대응

월말에 몰리는 회계 확인 작업

회계·재무 부서는 결산 기준, 회계 정책, 감사 질의 답변 근거를 찾는 작업이 월말과 분기말에 한꺼번에 몰립니다. 결산 기준이 개정 이력과 함께 여러 문서에 나뉘어 있고, 과거 감사에서 어떤 근거로 답했는지는 담당자의 메일함에만 남아 있는 경우가 많기 때문입니다. 마감 일정이 정해진 업무에서 근거 확인이 늦어지면 처리 시간이 늘어날 뿐 아니라 오류를 고칠 여유가 줄어 결산 품질 위험이 커집니다.

회계 문서의 색인 범위와 질문 예시

색인 대상은 회계 기준서, 내부 회계 정책, 과거 감사 질의와 답변, 전표에 붙은 증빙 이미지입니다. 증빙은 영수증이나 세금계산서를 스캔한 이미지라서 글자 인식과 표 구조 보존이 함께 필요합니다(3.5절).

- 「해외 법인 간 거래에 적용하는 환율 기준은 어느 정책 문서 몇 조에 있나」
- 「작년 외부 감사에서 리스 회계 처리에 대해 받은 질문과 회사가 낸 답변을 찾아 달라」
- 「이 출장비 전표에 첨부된 영수증이 증빙 기준을 충족하는가」

Rivian과 Amazon Finance의 재무 사례

Rivian은 회계 절차서를 지식 기반에 넣고, AI 에이전트가 그 문서를 근거로 ERP(Enterprise Resource Planning, 전사적 자원 관리)와 연동된 결산 작업을 처리하도록 구성했습니다. 클라우드 사업자의 기술 블로그에 실린 회사 발표에 따르면 이 구성으로 월말 결산에서 15일 이상의 수작업이 없어졌습니다[S18]. 즉 Rivian 사례에서 RAG는 에이전트가 판단 근거를 찾는 도구로 쓰였으며, 처리 구조는 7장 7.2.1절에서 자세히 다룹니다. Amazon Finance는 규제 기관의 질의에 답하는 과정에 생성형 AI 검색을 붙여 답변 작성 절차를 간소화했습니다[S19]. 두 사례는 마감이 있는 재무 업무에서 시간 단축이 곧 결산 위험을 줄이는 효과로 이어진다는 점을 보여 줍니다.

3.2.2 흔한 인사 질문 94% 를 사람 연결 없이 해결한 인사 문의 대응

반복 인사 질문에 묶이는 담당자

인사 부서에는 휴가 일수, 복리후생 신청 방법, 평가 일정처럼 답이 정해진 질문이 해마다 같은 시기에 반복해서 들어옵니다. 담당자는 취업규칙과 공지문을 찾아 같은 설명을 되풀이하느라 제도 설계나 상담처럼 사람이 해야 하는 일에 쓸 시간을 잃습니다. 직원이 쓰는 말과 규정의 용어가 달라서 사내 포털의 키워드 검색으로는 답을 찾지 못하는 경우가 많고, 이 차이를 메우려면 한국어 임베딩 선택이 중요해집니다(4장).

인사 문서의 색인 범위와 질문 예시

색인 대상은 취업규칙, 복리후생 안내, 인사 제도 문서, 평가 일정 공지입니다. 남은 휴가 일수나 급여처럼 개인 정보가 필요한 질문은 RAG가 답하지 않고 인사 시스템 조회나 담당자 연결로 넘기는 분기를 처음부터 설계합니다.

- 「배우자 출산휴가는 며칠이고 나눠서 쓸 수 있나」
- 「올해 성과 평가의 자기 평가 마감일은 언제인가」
- 「건강검진 지원 대상에 가족도 포함되나」

IBM AskHR의 인사 문의 처리 수치

IBM은 사내 인사 문의 창구인 AskHR에서 흔한 질문의 94%를 사람 담당자 연결 없이 해결한다고 밝혔습니다. 같은 발표에서 인사 지원 티켓은 2016년 이후 75% 줄었고, 이 도구가 4년 동안 인사 운영비 40% 절감에 기여

했다고 설명했습니다[S20]. 이 수치는 자사 발표이며 기간 조건이 서로 다르므로, 세 숫자를 하나의 효과로 합쳐 읽지 않아야 합니다. 그래도 인사 문의 자동 응대가 직원이 답을 기다리는 시간과 담당자의 응대 시간을 함께 줄인다는 판단의 근거로는 충분합니다.

3.3 개발·운영과 고객지원·CRM 에서 장애 기록과 상담 이력을 쓰는 RAG

개발·운영과 고객지원은 장애 보고서, 운영 절차서, 상담 이력처럼 매일 쌓이지만 다시 읽히지 않는 기록을 가장 많이 가진 업무입니다. 이 기록을 답의 근거로 쓰면 선임 엔지니어와 숙련 상담원에게 몰리던 질문이 줄어듭니다. 기술 부서와 고객 접점 부서 모두에서 과거 기록이 그대로 업무 자산이 됩니다.

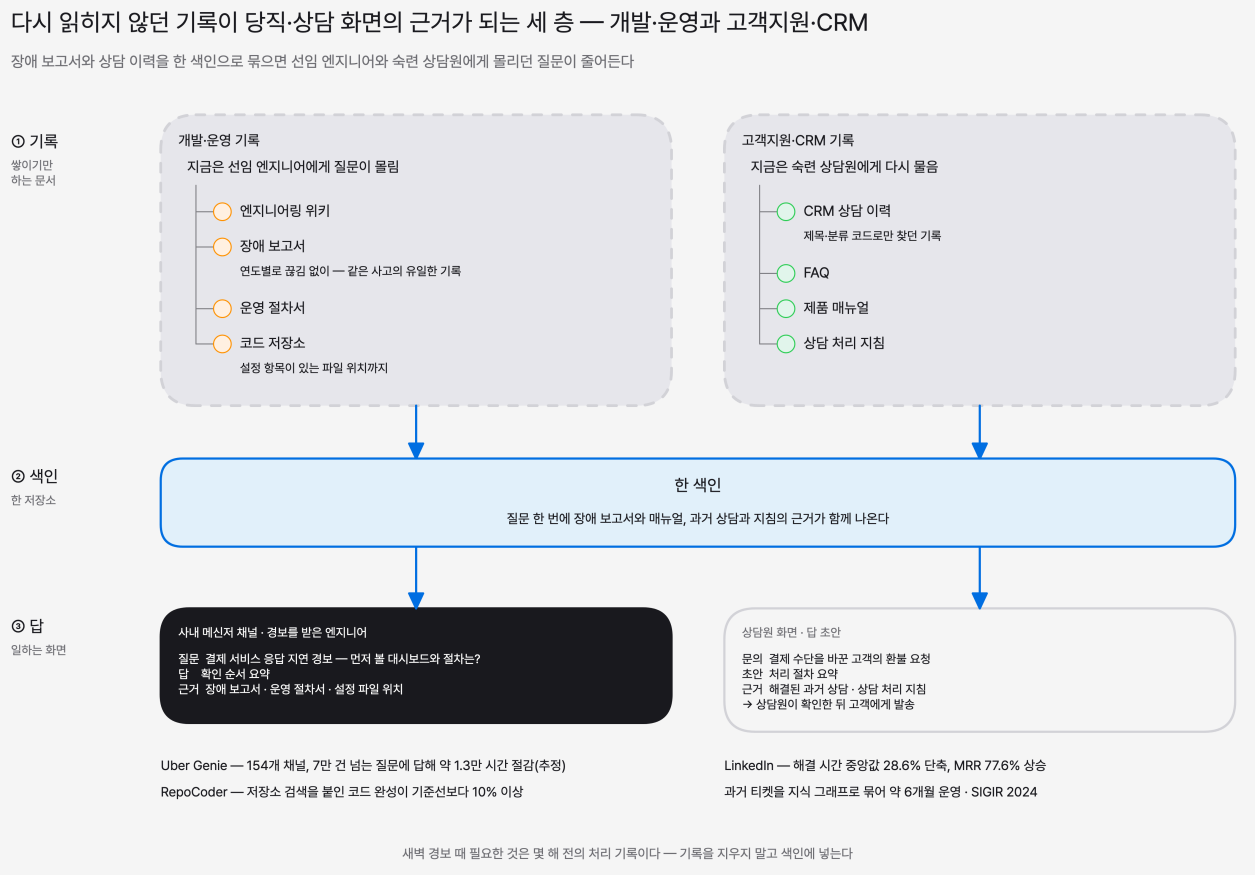

그림. 개발·운영과 고객지원·CRM — 쌓인 기록이 한 색인을 거쳐 당직·상담 화면의 답 근거가 되는 세 층

3.3.1 엔지니어링 시간 약 1.3만 시간을 줄인 개발·운영 질의 응답

선임 엔지니어에게 몰리는 운영 질문

개발 조직에서는 운영 문서, 장애 처리 기록, 코드 설명을 찾으려는 질문이 시스템을 오래 다룬 선임 엔지니어에게 몰립니다. 문서는 위키와 장애 보고서, 코드 저장소에 나뉘어 있고, 검색어를 정확히 알아야 찾을 수 있기 때문입니다. 선임 인력의 시간이 반복 답변에 쓰이고, 장애가 나면 답을 기다리는 동안 복구가 늦어집니다.

개발·운영 문서의 색인 범위와 질문 예시

색인 대상은 엔지니어링 위키, 장애 보고서, 운영 절차서, 코드 저장소입니다. 엔지니어는 평소 쓰는 사내 메신저 채널에서 질문하고, 답에는 근거가 된 문서와 코드 파일 위치가 붙습니다.

- 「결제 서비스에서 응답 지연 경보가 울리면 먼저 확인할 대시보드와 절차는 무엇인가」

- 「지난달 배포 실패의 원인과 되돌리기 절차가 적힌 장애 보고서를 찾아 달라」
- 「주문 서비스의 재시도 횟수를 정하는 설정은 어느 파일에 있나」

장애 대응에서는 색인 범위를 한 저장소로 묶는 일이 특히 중요합니다. 새벽에 경보를 받은 담당자는 지난번 같은 증상 때 무엇을 했는지 적힌 장애 보고서와, 해당 설정 항목의 뜻이 적힌 운영 매뉴얼을 번갈아 봅니다. 두 문서가 다른 시스템에 있으면 창을 바꾸고 다시 로그인한 뒤 검색을 두 번 해야 합니다. 같은 색인에 들어 있으면 질문 한 번에 양쪽 근거가 함께 나오므로, 파일럿 범위를 정할 때 장애 보고서와 매뉴얼을 함께 넣는 편이 효과를 재기 쉽습니다.

오래된 장애 기록도 같은 이유로 색인 범위에 넣을 만합니다. 같은 사고가 다시 났을 때 대응 시간을 가르는 것은 대개 몇 해 전의 처리 기록이고, 담당자가 바뀐 뒤에는 그 기록이 조직이 그 사고를 겪었다는 유일한 증거가 됩니다. 파일럿 문서를 고를 때는 최근 1년 치만 넣기보다 연도별 장애 보고서를 끊김 없이 넣을 수 있는지 먼저 확인하고, 가장 오래된 보고서가 문단 단위로 읽히는지 표본으로 열어 봅니다.

Uber Genie와 RepoCoder의 결과

Uber는 장애 대응 질문에 답하는 사내 도우미 Genie를 154개 채널에 배치해 7만 건이 넘는 질문에 답했고, 이로써 엔지니어링 시간을 약 1.3만 시간 줄였다고 추정했습니다[S21]. 1.3만 시간은 Uber가 밝힌 추정치이므로 실제로 잰 노동 시간과는 구분해서 읽어야 합니다. 한편 연구 쪽에서는 RepoCoder가 코드 저장소 안의 관련 파일을 반복해서 검색해 코드 완성 모델에 넣었을 때 검색 없이 완성하는 기준선보다 10% 이상 좋은 결과를 냈습니다[S23]. 운영 문서와 코드 모두 검색 근거를 붙이면 답의 품질이 오르며, 선임 엔지니어의 시간을 되찾는 효과는 개발 조직이 생산성을 판단하는 근거로 쓸 수 있습니다.

3.3.2 해결 시간 중앙값 28.6% 단축의 고객지원 상담 이력 검색

사람마다 다른 상담 답변

상담원이 비슷한 과거 문의와 처리 방법을 찾지 못하면 해결이 늦어지고, 같은 문의에도 상담원마다 다른 답을 주게 됩니다. 과거 상담 이력은 CRM(Customer Relationship Management, 고객 관계 관리) 시스템에 쌓여 있지만, 제목과 분류 코드로만 검색할 수 있어 문의 내용이 조금만 달라도 찾지 못합니다. 신입 상담원일수록 숙련 상담원에게 처리 방법을 다시 묻는 일이 잦아지고, 그만큼 숙련 상담원이 고객 응대에 쓸 시간이 줄어듭니다.

상담 문서의 색인 범위와 질문 예시

색인 대상은 CRM에 쌓인 상담 이력, 자주 묻는 질문을 모은 FAQ, 제품 매뉴얼, 상담 처리 지침입니다. 이 구성에서 RAG는 상담원 화면에 답 초안과 근거 문서를 띄우고, 상담원은 그 내용을 확인한 뒤 고객에게 답을 보냅니다.

- 「결제 수단을 바꾼 고객이 환불을 요청할 때 처리 절차는 무엇인가」
- 「이 오류 코드로 들어온 과거 문의 가운데 해결된 건의 처리 방법을 보여 달라」

LinkedIn과 Klarna의 상담 수치

LinkedIn은 과거 상담 티켓을 지식 그래프로 묶고 그 위에서 RAG 검색을 수행하는 구성을 약 6개월 운영했습니다. 문의 해결 시간의 중앙값이 28.6% 줄었고, 검색 정확도를 나타내는 MRR(Mean Reciprocal Rank, 평균 역순위)이 77.6% 올랐습니다[S24]. 이 결과는 정보 검색 학회 SIGIR 2024에 실린 논문이라서 이 장의 기

업 사례 가운데 근거 수준이 높은 편입니다. 반면 Klarna는 상담 도우미가 첫 달에 230만 건을 처리하고 해결 시간을 11분에서 2분 미만으로 줄였으며 재문의가 25% 감소했다고 발표했습니다[S25]. 다만 이 수치는 회사 발표 기준이고 이후 그 효과를 두고 논란이 있었으므로 보조 근거로만 참고하는 편이 안전합니다. 고객지원은 해결 시간과 재문의율처럼 도입 전후를 비교하기 쉬운 지표가 이미 있어서 파일럿 후보에 자주 오릅니다.

3.4 ERP·구매와 공공 행정에서 규정과 법령을 근거로 답하는 RAG

ERP·구매와 공공 행정은 답에 반드시 규정, 계약, 법령 같은 근거 조항이 따라와야 하는 업무입니다. 근거 없는 답은 쓸 수 없기 때문에 답과 출처를 함께 돌려주는 RAG의 구조가 그대로 들어맞습니다. 공공 영역은 국내외에서 시범을 넘어 확대 단계에 들어선 사례가 공개되어 있어 도입 가능성을 판단하기 쉽습니다.

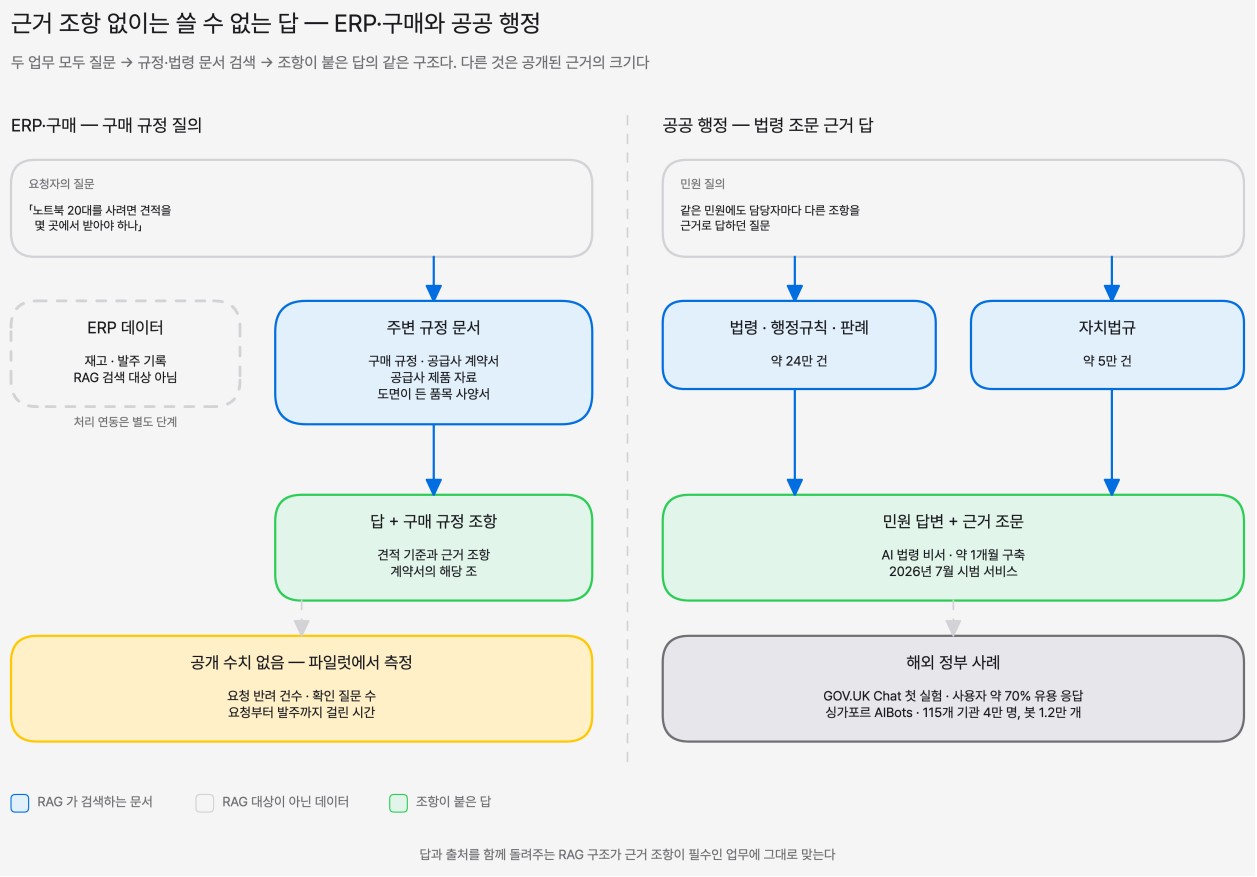

그림. ERP·구매와 공공 행정 — 구매 규정 질의와 법령 조문 근거 답의 비교

3.4.1 구매 규정과 계약서를 근거로 답하는 ERP·구매 요청 지원

구매 요청 단계의 확인 질문

구매 규정, 계약 조건, 품목 기준이 여러 문서에 나뉘어 있으면 요청자와 구매 담당 사이에 확인 질문이 여러 번 오갑니다. 요청자는 어떤 서류를 붙여야 하는지 모른 채 요청을 올리고, 구매 담당은 규정과 맞지 않는 요청을 반려하면서 같은 설명을 반복합니다. 요청부터 발주까지 걸리는 시간 가운데 상당 부분이 실제 구매 처리가 아닌 규정 확인에 쓰입니다.

구매 문서의 색인 범위와 질문 예시

색인 대상은 구매 규정, 공급사와 맺은 계약서, 공급사 제품 자료, 도면 이미지가 들어 있는 품목 사양서입니다. 여기서 RAG의 대상은 재고나 발주 기록 같은 ERP 데이터를 둘러싼 규정 문서이며, ERP 시스템과 직접 연동해 처리까지 잇는 구성은 7장에서 다룹니다.

- 「노트북 20대를 사려면 견적을 몇 곳에서 받아야 하나」
- 「이 공급사와 맺은 기본 계약에서 납기 지연 시 적용 조건은 몇 조에 있나」
- 「대체품으로 등록된 이 부품이 사양서의 치수 기준을 충족하나」

구매 업무에서 기대하는 효과와 측정 항목

흩어진 사내 지식을 하나로 묶은 IskayPet 사례처럼, 구매 부서도 규정과 계약 문서를 한 지식 기반에 모으는 작업에서 시작합니다[S26]. 다만 구매 분야에서 RAG 도입 효과를 수치로 공개한 사례는 아직 확인되지 않으므로 기대 효과는 흐름으로만 정리할 수 있습니다. 요청 단계에서 규정 근거를 바로 보여 주면 반려 건수와 확인 질문이 줄어들 것으로 기대하며, 파일럿에서는 요청 반려 건수, 요청자와 구매 담당이 주고받은 확인 질문 수, 요청부터 발주까지 걸린 시간을 재면 됩니다.

3.4.2 법령 약 24만 건을 연결한 AI 법령 비서와 국내외 공공 사례

민원 답변이 담당자마다 다른 행정 현장

민원 응대와 행정 처리에는 법령, 행정규칙, 판례, 자치법규 확인이 반드시 따르며, 같은 민원에도 담당자마다 다른 조항을 근거로 답하는 일이 생깁니다. 법령은 개정이 잦고 자치법규는 지방자치단체마다 달라서 담당자가 모든 근거를 기억하기 어렵기 때문입니다. 공공 영역의 RAG는 답의 근거 조항을 함께 보여 주어 답변의 일관성을 높이는 데 초점을 둡니다.

국내 공공기관의 도입 사례

법제처 등은 법령, 행정규칙, 판례 약 24만 건과 자치법규 약 5만 건을 근거로 답하는 AI 법령 비서를 약 1개월 만에 구축해 2026년 7월 시범 서비스를 시작했다고 뉴시스가 보도했습니다[S29]. 국민권익위원회는 민원 창구인 국민신문고에 생성형 AI를 적용한다고 밝혔다고 디지털투데이가 전했고[S30], 국세청은 생성형 AI 상담 챗봇의 적용 범위를 넓힌다고 헤럴드경제가 2026년 4월 보도했습니다[S31]. 세 사례는 법령과 민원 답변 사례처럼 근거가 분명한 문서에서 국내 공공기관이 이미 RAG 계열 서비스를 운영 단계로 옮기고 있음을 보여 줍니다.

해외 정부의 도입 사례

영국 정부디지털서비스(GDS)는 정부 안내 웹사이트 GOV.UK의 내용을 근거로 답하는 GOV.UK Chat을 처음 실험했고, 참여한 사용자의 약 70%가 답이 유용했다고 응답했습니다[S27]. 싱가포르 GovTech는 공무원이 자기 업무 문서로 챗봇을 만드는 AIBots를 운영하며, 2025년 2월 기준으로 115개 기관의 4만 명이 1.2만 개의 봇을 만들었다고 밝혔습니다[S28]. 아래 표는 국내외 공공 사례를 기관, 대상 문서, 규모, 결과로 나란히 놓은 것입니다.

기관	대상 문서	규모	결과
법제처 등 (국내)	법령·행정규칙·판례, 자치법규	약 24만 건 + 약 5만 건	약 1개월 구축, 2026년 7월 시범 서비스 (뉴시스 보도)

기관	대상 문서	규모	결과
국민권익위원회 (국내)	국민신문고 민원	보도에 규모 미기재	생성형 AI 적용 (디지털투데이 보도)
국세청 (국내)	세무 상담 안내	보도에 규모 미기재	생성형 AI 챗봇 확대 (헤럴드경제 보도)
영국 GDS (해외)	GOV.UK 정부 안내	첫 실험	사용자 약 70% 유용 응답
싱가포르 GovTech (해외)	기관별 업무 문서	115개 기관, 4만 명, 봇 1.2만 개	2025년 2월 기준 운영

여덟 개 업무의 레퍼런스 종합

여덟 개 업무의 공개 레퍼런스를 한 표로 모으면 근거의 성격이 업무마다 크게 다릅니다. 학회 논문과 정부 발표처럼 검증 수준이 높은 근거가 있는 업무가 있는 반면, 회사 발표에만 기대거나 공개 수치가 없는 업무도 있습니다. 파일럿 업무를 고를 때는 효과 수치의 크기와 함께 그 수치의 근거 성격을 같이 따져야 합니다.

업무	대표 레퍼런스	공개 수치	근거 성격
마케팅	IskayPet 지식 통합	없음	파일럿에서 측정
영업	Morgan Stanley, Twitch	문서 접근률 20%→80%, 도입 전 미응답 20%	회사 발표
회계·재무	Rivian, Amazon Finance	월말 결산 수작업 15일 이상 제거	회사 발표
인사	IBM AskHR	흔한 질문 94% 자동 해결, 티켓 75% 감소	자사 발표
개발·운영	Uber Genie, RepoCoder	약 1.3만 시간 절감(추정), 기준선 대비 10% 이상	자사 발표, 논문
고객지원·CRM	LinkedIn, Klarna	해결 시간 중앙값 28.6% 단축	학회 논문, 자사 발표(논란)
ERP·구매	IskayPet 지식 통합	없음	파일럿에서 측정
공공	법제처 등, GOV.UK, AIBots	법령 약 24만 건, 유용 응답 약 70%	정부 발표, 언론 보도

3.5 표와 도면, 스캔본에 핵심 내용이 담긴 이미지 문서의 검색

업무 문서의 핵심 정보는 문장보다 표와 그림에 들어 있는 경우가 많습니다. 사양표의 수치, 조직도의 보고 관계, 도면의 치수는 글자만 뽑아내면 구조가 깨지고, 스캔본은 글자 인식 단계를 거치지 않으면 아예 검색되지 않습니다. 파일럿 대상 업무의 문서 형식이 준비 기간과 난이도를 바꾸며, 의사결정권자는 이 점을 업무 선정 전에 알고 있어야 합니다.

3.5.1 글자 추출로는 놓치는 표와 도면 안의 업무 정보

글자 추출에서 깨지는 표 구조

견적표, 사양표, 조직도, 설비 도면은 칸의 위치가 곧 의미이기 때문에 글자만 순서대로 뽑으면 어느 값이 어느 항목에 속하는지 사라집니다. 예를 들어 품목별 정격 출력과 무게를 적은 사양표에서 글자만 뽑으면 머리글과 값이 한 줄로 이어지고, 「A-200의 무게는 얼마인가」라는 질문에 엉뚱한 값이 답으로 나올 수 있습니다. 반면 표 구조를 보존하는 파싱은 행마다 머리글을 붙여 조각을 만들기 때문에 같은 질문에 정확한 근거를 돌려줍니다.

> [글자만 추출한 결과]
> 품목 정격 출력 무게 A-100 5kW 12kg A-200 7.5kW 15kg
>
> [표 구조를 보존한 결과]
> 품목: A-100 | 정격 출력: 5kW | 무게: 12kg
> 품목: A-200 | 정격 출력: 7.5kW | 무게: 15kg

HWP 문서의 확인 항목

국내 공공기관과 기업 문서의 상당수는 HWP(한글 워드프로세서 문서) 형식이며, 병합된 표 칸, 각주, 그림 캡션이 많아 파싱 결과가 도구마다 달라집니다. 하지만 HWP 파싱 품질을 비교한 공개 수치는 아직 없으므로, 도입 전에 대표 문서를 표본으로 골라 병합 칸의 값, 각주와 본문의 연결, 그림 캡션, 개정 이력 표가 제대로 남는지 직접 확인해야 합니다. 파싱을 마친 한국어 문장은 형태소 분석과 한국어 임베딩 단계를 거쳐 색인되므로, 파싱 품질은 4장의 검색 품질과도 이어집니다. 아래 표는 문서 형식별로 적합한 처리 방식을 정리한 것입니다.

문서 형식	글자 추출	표 구조 보존 파싱	페이지 이미지 검색
글 중심 PDF·워드 문서	충분함	필요 적음	필요 적음
표가 많은 보고서·사양서	값과 머리글 연결이 깨짐	적합	보조로 사용
스캔본·영수증	글자 인식 후에만 가능	글자 인식 품질에 좌우	적합
도면·조직도	대부분 정보 손실	제한적	적합
HWP 문서	도구마다 결과가 다름	표본 문서로 확인 필요	변환 후 사용

3.5.2 문서 페이지 이미지를 그대로 검색하는 멀티모달 검색 방식

ColPali의 페이지 이미지 임베딩

ColPali는 문서 페이지를 글자로 바꾸지 않고 페이지 이미지 그대로 비전 언어 모델에 넣어 임베딩하는 검색 방식입니다[S15]. 즉 글자 인식, 레이아웃 분석, 조각 나누기 같은 전처리 단계를 거치지 않고 페이지 한 장을 여러 개의 작은 벡터로 나타낸 뒤, 질문의 낱말 벡터와 하나씩 비교해 관련 페이지를 찾습니다. 연구진이 만든 문서 이미지 검색 벤치마크에서 이 방식은 글자를 추출해 검색하는 기존 파이프라인보다 높은 성능을 냈고 색인 준비 시간도 짧았습니다[S15]. 도면, 차트, 스캔본처럼 글자 추출로 정보가 사라지는 문서에서 특히 유리합니다.

연구 단계 기술과 운영 기술의 구분

다만 ColPali는 2024년에 발표된 연구 단계 기술이며, 페이지마다 벡터를 여러 개 저장하므로 글자 기반 색인보다 저장 공간이 많이 듭니다. 논문의 평가도 영어와 프랑스어 문서 중심이라서 한국어 문서에서의 성능은 도입 전에 자기 조직 문서로 다시 재야 합니다. 대부분의 조직에는 글자 추출과 표 구조 보존 파싱으로 시작하고, 도면이나 스캔본 비중이 큰 업무에서만 페이지 이미지 검색을 덧붙이는 순서가 현실적입니다.

비교 항목	글자 추출 방식	페이지 이미지 검색 방식
색인 준비	글자 인식·파싱·조각 나누기 필요	페이지 이미지를 바로 임베딩
유리한 문서	글 중심 문서, 구조가 단순한 표	도면, 차트, 스캔본
불리한 문서	도면, 복잡한 표, 저품질 스캔본	글이 긴 문서의 세부 문장
저장 공간	조각당 벡터 1개 수준	페이지당 여러 벡터로 더 큼
근거 표시	문장 단위로 인용	페이지 이미지 단위로 제시
성숙도	운영 환경에서 널리 쓰임	연구 단계, 한국어 성능은 표본 문서로 측정

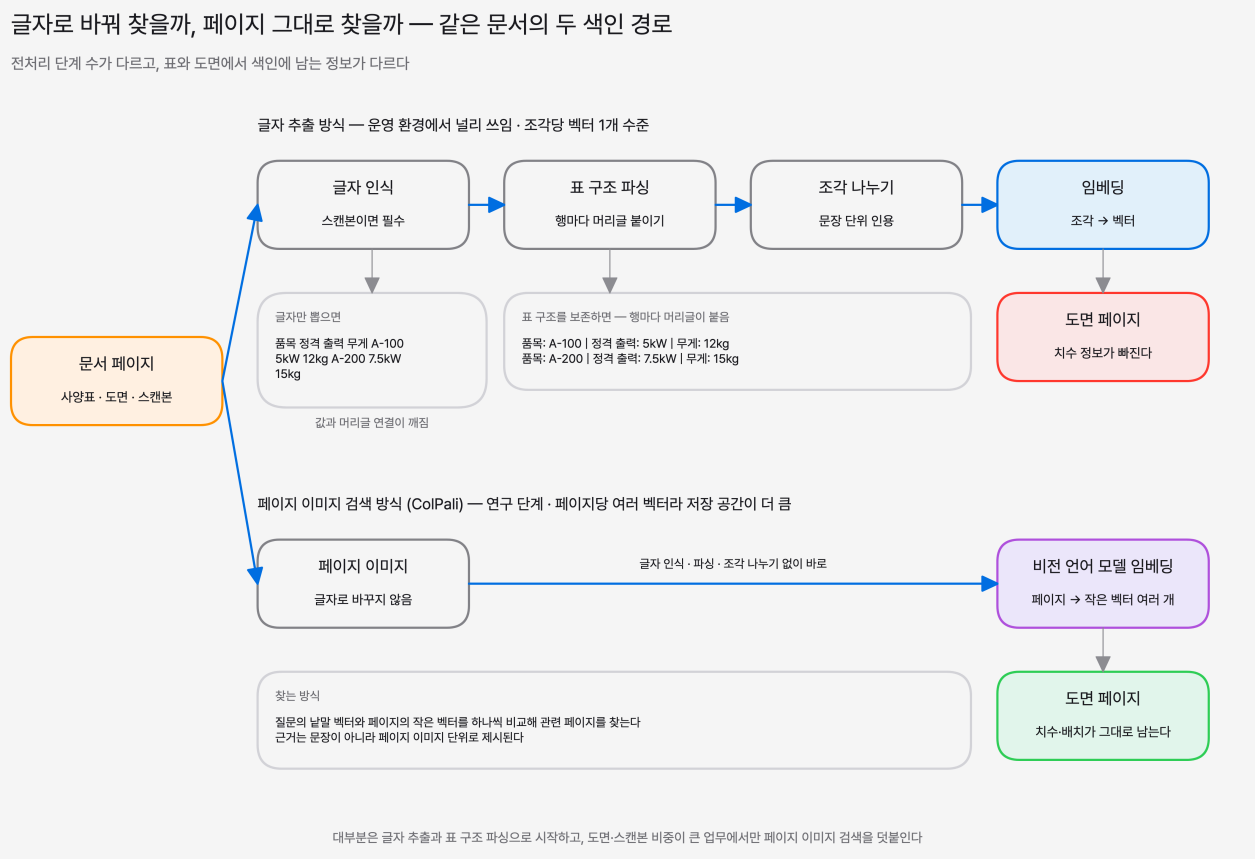

그림. 글자 추출 방식과 페이지 이미지 검색 방식의 처리 흐름 비교

4장: 한국어 문서의 검색 성능을 가르는 한국어 임베딩 선택

3장의 사례가 보여 준 효과는 검색이 질문에 맞는 문서를 찾아낸다는 전제 위에 있습니다. 한국어 업무 문서를 대상으로 RAG를 구성할 때 검색 품질은 두 구성 요소에서 크게 달라집니다. 키워드 검색에서는 문장을 낱말 단위로 나누는 형태소 분석이, 의미 검색에서는 문장을 숫자 벡터로 바꾸는 한국어 임베딩 모델이 성능을 가릅니다. 영어 문서를 기준으로 만든 검색 구성을 한국어 문서에 그대로 옮기면 찾아야 할 문서를 놓치는 일이 잦아지는데, 공개 연구에서는 낱말을 나누는 방식만 바꿔도 BM25 재현율이 0.396에서 0.619로 올랐습니다[S37]. 이 장에서는 한국어가 검색에 까다로운 이유와 형태소 분석기 선택 기준, 공개 비교표와 예시 실험으로 본 한국어 임베딩 모델의 차이를 정리하고, 자기 조직 문서로 다시 재 볼 임베딩 후보 2–3개를 고르는 평가 절차를 제시합니다.

4.1 조사가 붙고 띄어쓰기가 일정하지 않은 한국어 문서의 검색 난점

한국어 문서 검색이 영어보다 어려운 이유는 낱말의 형태가 문장 안에서 계속 바뀌는 교착어 구조와, 같은 내용을 담는 데 필요한 토큰 수가 많다는 점에 있습니다. 앞의 것은 키워드 검색이 문서를 놓치게 만들고, 뒤의 것은 임베딩 모델과 생성 모델이 한 번에 처리할 수 있는 문서 양을 줄입니다. 따라서 한국어 RAG의 첫 설계 결정은 문장을 어떤 단위로 쪼개어 색인할지를 정하는 일입니다.

4.1.1 「계약서를」「계약서의」처럼 조사에 따라 형태가 바뀌는 교착어의 검색 문제

교착어는 낱말 뒤에 조사와 어미가 붙어 문법 관계를 나타내는 언어입니다. 한국어에서는 「계약서」라는 낱말 하나가 문장에 따라 「계약서를」, 「계약서의」, 「계약서에는」처럼 서로 다른 문자열로 나타납니다. 공백을 기준으로 낱말을 자르는 키워드 검색은 이 문자열들을 서로 다른 낱말로 색인하기 때문에, 사용자가 「계약서」라고 입력해도 「계약서에는」만 들어 있는 문서를 찾지 못합니다. 영어도 복수형이나 시제 변화가 있지만 낱말 경계가 공백으로 분명하고 변화의 가짓수가 적어 같은 문제가 훨씬 덜 일어납니다.

기본 낱말	문서 안에 실제로 나타나는 형태	공백 분리 키워드 검색에서 일어나는 일
계약서	계약서를 · 계약서의 · 계약서에는 · 계약서로서	「계약서」 질의와 문자열이 달라 일치하지 않음
결재	결재가 · 결재를 · 결재받은 · 결재해야	「결재 절차」 질의로 「결재받은」 문서를 놓침
정산	정산은 · 정산할 · 정산했으며 · 정산서	활용형과 복합어가 모두 다른 낱말로 색인됨

띄어쓰기가 사람마다 다르다는 점이 이 문제를 키웁니다. 같은 부서 문서 안에서도 「업무 협약서」와 「업무협약서」, 「출장 비용 정산」과 「출장비용정산」이 함께 쓰이며, 공백 분리 방식은 두 표기를 서로 다른 낱말로 취급합니다. 그래서 한국어 키워드 검색은 공백 대신 형태소 분석기로 낱말의 뿌리를 떼어 내거나, 글자를 일정 길이로 잘라 색인하는 방식을 써야 제대로 작동합니다[S37].

4.1.2 토크나이저에 따라 영어보다 최대 약 3배 늘어나는 한국어 토큰 수

토크나이저는 글을 모델이 읽는 단위인 토큰으로 자르는 도구입니다. 대부분의 생성 모델과 임베딩 모델은 영어 문서를 많이 학습한 토크나이저를 쓰기 때문에, 한국어 문장은 영어보다 더 잘게 쪼개집니다. 고려대 연구진은 같은 뜻의 문장을 여러 토크나이저로 나눠 본 결과 한국어 토큰 수가 영어와 비슷한 경우에서 최대 약 3배까지 늘어난다는 점을 보였습니다[S38]. 이 차이는 토크나이저가 한국어 낱말을 얼마나 많이 어휘 목록에 담고 있는지에 따라 달라집니다.

토큰 수가 늘면 운영 부담이 두 방향으로 커집니다. 같은 분량의 문서를 임베딩하고 답변을 생성하는 데 드는 처리 시간이 늘어나고, 모델마다 정해진 최대 입력 토큰 수 안에 넣을 수 있는 한국어 문서 양이 영어보다 적어지므로, 문서 조각을 더 작게 나누거나 검색 결과 중 일부만 생성 모델에 넘겨야 합니다. 예를 들어 최대 입력이 512 토큰인 임베딩 모델은 영어 문단 하나를 통째로 담을 수 있어도 같은 길이의 한국어 문단은 잘라 내야 할 수 있습니다. 한국어 RAG의 처리량과 입력 한도를 예측할 때는 문자 수가 아니라 실제로 쓰려는 토크나이저로 잰 토큰 수를 기준으로 삼아야 합니다.

4.2 형태소 분석으로 높이는 한국어 키워드 검색의 재현율

한국어 키워드 검색은 낱말을 한국어 문법에 맞게 나누기만 해도 성능이 크게 좋아집니다. 키워드 검색은 제품 코드나 조항 번호처럼 정확한 문자열을 찾는 데 강하므로, 의미 검색과 함께 쓰는 하이브리드 검색(5장)의 한쪽 축으로 남겨 두어야 합니다. 이 축을 한국어에 맞게 준비하는 방법이 형태소 분석이고, 분석기를 고르는 기준은 사용자 사전, 복합명사 처리 방식, 연동 환경으로 정리할 수 있습니다.

4.2.1 공백 분리 0.396 에서 형태소 분석 0.619 로 오른 BM25 재현율

BM25는 질의에 나온 낱말이 문서에 얼마나 자주, 얼마나 드물게 나타나는지로 점수를 매기는 키워드 검색 방식입니다. 낱말 단위로 점수를 계산하므로 문장을 어떤 단위로 나누느냐가 결과를 직접 좌우합니다. 재현율은 찾아야 할 문서를 얼마나 빠짐없이 찾았는지를 나타내는 지표이고, Recall@100은 상위 100건 안에 정답 문서가 들어온 비율입니다.

한국어 BM25 토큰화 방식	Recall@100
다국어 서브워드 토크나이저	0.367
공백 분리	0.396
형태소 분석기	0.619

출처: 한국어 정보 검색 연구의 BM25 비교 결과[S37]

한국어 검색 연구에서 BM25 Recall@100은 공백 분리 0.396, 다국어 서브워드 토크나이저 0.367, 형태소 분석기 0.619였습니다. 즉 형태소 분석기를 쓰면 공백 분리보다 찾아낸 정답 문서가 약 1.6배 많아졌습니다. 다국어 서브워드 토크나이저가 공백 분리보다도 낮았다는 점도 눈여겨볼 만한데, 여러 언어에 두루 쓰도록 만든 분할 방식이 한국어 낱말의 뿌리와 조사를 제대로 구분하지 못했기 때문입니다. 키워드 검색은 버리기보다 한국어 형태소 분석을 붙여 고쳐 쓰는 편이 하이브리드 검색 전체의 성능을 끌어올리는 출발점이 됩니다.

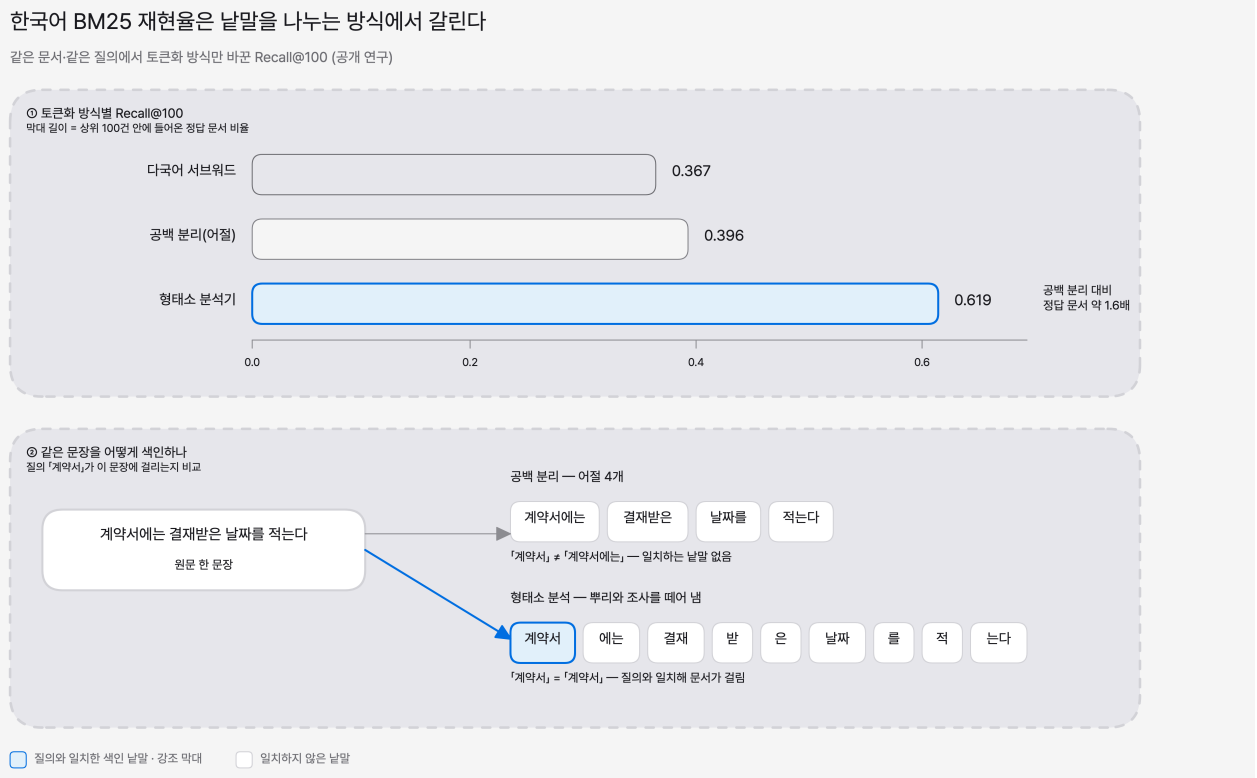

그림. 한국어 BM25 토큰화 방식별 Recall@100 비교

4.2.2 Nori·Kiwi 같은 한국어 형태소 분석기의 선택 기준

검색 엔진에서 바로 쓸 수 있는 대표적인 한국어 형태소 분석기는 Nori입니다. Nori는 Apache Lucene에 포함된 한국어 분석기로, Lucene 기반 검색 엔진에서 플러그인으로 설치해 색인과 질의에 같은 분석 규칙을 적용합니다[S39]. 사용자 사전으로 사내 용어와 제품명을 등록할 수 있고, `decompound_mode` 설정으로 「업무협약서」 같은 복합명사를 나누지 않을지, 나눈 조각만 남길지, 원형과 조각을 함께 남길지를 정할 수 있습니다. 품사 필터로 조사와 어미를 색인에서 빼면 「계약서를」과 「계약서의」가 같은 「계약서」로 모입니다.

파이썬으로 RAG 파이프라인을 구성한다면 Kiwi 같은 라이브러리형 분석기를 BM25에 직접 연결하는 방식이 흔합니다. 한국어 RAG 평가 도구인 AutoRAG는 BM25 토크나이저로 공백 분리와 함께 Kiwi, Okt, Kkma 같은 한국어 형태소 분석기를 선택지로 제공합니다[S40]. 이때 분석기마다 사전 구성과 처리 속도가 달라서 같은 문서라도 낱말이 조금씩 다르게 나뉘므로, 색인할 때와 질의할 때 반드시 같은 분석기와 같은 사용자 사전을 써야 합니다.

선택 기준	확인할 내용	업무 문서에서 중요한 이유
사용자 사전	사내 용어·제품명·약어를 등록하고 갱신하는 방법	「PG-101」 같은 코드나 부서 고유 용어가 잘못 쪼개지는 일을 막음
복합명사 처리	나누지 않기 · 조각만 남기기 · 원형과 조각을 함께 남기기	「출장비용정산」과 「출장 비용 정산」을 같은 문서로 찾게 함
처리 속도	문서 전체 색인 시간과 질의당 분석 시간	문서가 많을수록 재색인 시간이 길어짐

선택 기준	확인할 내용	업무 문서에서 중요한 이유
연동 환경	검색 엔진 플러그인인지 파이썬 라이브러리인지	기존 검색 엔진을 쓸지 새 파이프라인을 만들지에 따라 고를 수 있는 분석기가 달라짐

기존에 Lucene 기반 검색 엔진을 운영하고 있다면 Nori를 켜고 사용자 사전을 채우는 것이 가장 짧은 경로입니다. 새로 파이썬 기반 RAG를 만든다면 Kiwi처럼 라이브러리로 부를 수 있는 분석기가 평가와 교체에 더 편합니다.

4.3 한국어 임베딩 모델 비교표와 예시 실험으로 보는 선택 기준

의미 검색에서는 임베딩 모델이 성능을 가릅니다. 임베딩 모델은 문장을 수백에서 수천 개의 숫자로 이루어진 벡터로 바꾸고, 뜻이 가까운 문장은 가까운 벡터가 되도록 학습된 모델입니다. 한국어 문장을 충분히 학습하지 않은 모델은 「육아휴직」과 「애 낳고 쉴 수 있는 기간」이 같은 뜻이라는 점을 벡터에 담지 못합니다. 공개 비교표로 후보를 좁히고, 예시 실험으로 차이의 크기를 확인한 다음, 자기 조직 문서로 다시 재어 최종 모델을 정하는 순서가 필요합니다.

4.3.1 한국어 검색 7종 평균 NDCG@10 으로 본 한국어 임베딩 모델 비교

NDCG@10은 상위 10건 안에 정답 문서가 얼마나 위쪽에 놓였는지를 0에서 1 사이 값으로 나타내는 지표입니다. 아래 표는 한국어 검색 데이터셋 7종의 평균 NDCG@10을 한 모델 카드에서 같은 조건으로 잰 값입니다[S33].

임베딩 모델	한국어 추가 학습	한국어 검색 7종 평균 NDCG@10
snowflake-arctic-embed-l-v2.0-ko	예	0.7404
BGE-m3-ko	예	0.7300
KURE-v1	예	0.7277
bge-m3	아니요(다국어)	0.7242
multilingual-e5-large	아니요(다국어)	0.7216
Qwen3-Embedding-0.6B	아니요(다국어)	0.6762
상용 영어 중심 API	아니요	0.6622
ko-sroberta	예(소형 구형 모델)	0.5263

출처: snowflake-arctic-embed-l-v2.0-ko 모델 카드의 비교표[S33]. 모델 제작자가 잰 값이며, 문서 종류와 질의 형태에 따라 순위가 바뀔 수 있습니다.

상위 세 모델은 모두 다국어 모델을 한국어 데이터로 다시 학습한 모델입니다. BGE-m3-ko와 KURE-v1은 bge-m3를 바탕으로 한국어 검색 데이터를 더 학습했고, 두 모델 모두 원래 모델보다 높은 점수를 냈습니다[S34][S35]. 반면 상용 영어 중심 API는 0.6622로 다국어 공개 모델보다도 낮았고, 한국어로 학습했더라도

크기가 작고 오래된 ko-sroberta는 0.5263에 그쳤습니다. 후보를 고를 때는 「한국어로 추가 학습했는가」와 「모델 크기가 충분한가」 두 기준을 함께 보아야 합니다.

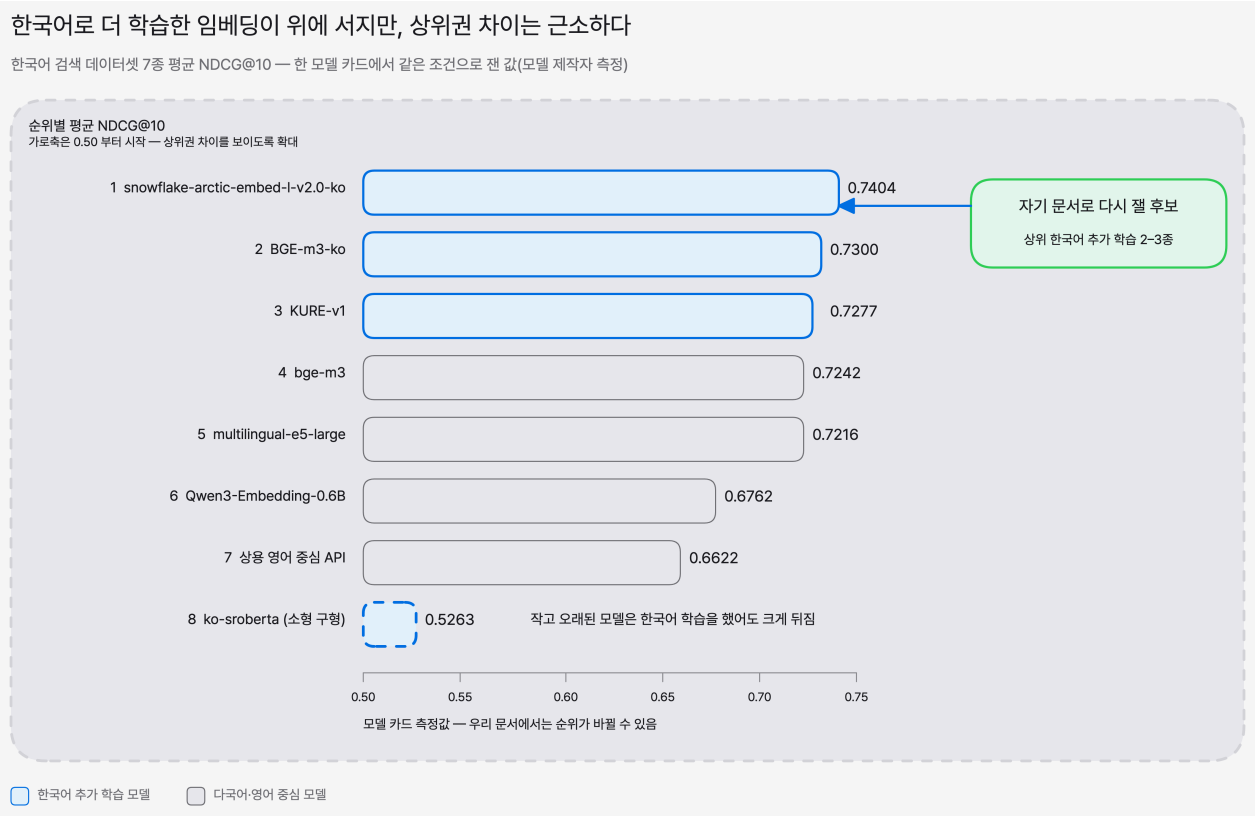

그림. 한국어 검색 7종 평균 NDCG@10으로 본 임베딩 모델 8종 비교(모델 제작자 측정)

다른 비교표도 같은 경향을 보입니다. BGE-M3 논문의 MIRACL 한국어 결과에서도 의미 검색이 BM25를 크게 앞섰는데, 이 값은 100점 만점 기준이고 데이터셋도 달라서 위 표와 직접 비교할 수 없으며 구체적인 수치는 5장 5.1.2절과 5.2.1절에서 다룹니다[S32]. 더 큰 모델을 넣은 한국어 9종 비교에서는 Qwen3-Embedding-8B가 평균 0.7825, 4B가 0.7718로 상위에 올랐는데, 이 값은 경쟁 모델 제작자가 잰 것이라 신뢰도를 낮게 보아야 합니다[S41]. 큰 모델은 점수가 높은 대신 벡터 차원이 크고 처리 시간이 길어지므로, 문서량과 응답 시간 목표를 함께 놓고 판단해야 합니다.

모델을 고를 때는 점수 외에 세 가지 사양을 함께 확인합니다.

- **벡터 차원 수**: BGE-M3 계열은 1024차원 밀집 벡터를 만듭니다. 차원이 클수록 벡터 저장 공간과 검색 계산량이 늘어납니다[S32].
- **최대 입력 토큰 수**: BGE-M3, BGE-m3-ko, KURE-v1은 최대 8,192토큰까지 한 번에 임베딩합니다[S34][S35]. 반면 최대 입력이 512토큰 이하인 모델은 4.1.2의 토큰 증가를 고려해 문서 조각을 더 작게 나누어야 합니다.
- **모델 교체 시 재색인 부담**: 임베딩 모델을 바꾸면 기존 벡터와 새 벡터가 서로 다른 공간에 놓이므로 모든 문서 조각을 다시 임베딩하고 색인을 새로 만들어야 합니다. 차원 수까지 달라지면 벡터 데이터베이스의 컬렉션 구성도 바꿔야 하므로, 처음 고를 때 평가를 충분히 하고 교체 기간에는 새 색인을 따로 만들어 비교한 뒤 전환하는 편이 안전합니다.

4.3.2 영어 중심 임베딩 0.20 과 한국어 임베딩 1.00 의 정답률 차이를 보인 예시 실험

공개 비교표의 점수 차이가 실제 업무 질문에서 어느 정도로 나타나는지 보기 위해, 업무 문서 27건과 질의 15건으로 만든 예시 실험을 소개합니다. 문서는 휴가·근무 규정, 장애 대응 기준, 제품 코드 설명처럼 사내에서 자주 찾는 내용으로 구성했고, 질의는 직원이 실제로 물을 법한 구어 표현으로 썼습니다. 정답률@1은 첫 번째 검색 결과가 정답 문서인 비율이고, 정답률@3은 상위 3건 안에 정답 문서가 들어온 비율입니다.

검색 방식(예시 실험: 업무 문서 27건 · 질의 15건)	정답률@1	정답률@3
어절 단위 BM25	0.47	0.53
한글 2글자 분할 BM25	0.67	0.73
영어 중심 임베딩	0.20	0.47
한국어 임베딩(BGE-m3-ko)	1.00	1.00

한국어 임베딩은 15개 질의 모두에서 첫 번째 결과로 정답 문서를 돌려주었고, 영어 중심 임베딩은 5개 중 1개꼴로만 맞혔습니다. 키워드 검색 두 방식이 모두 놓친 질의는 다음 세 건으로, 질문에 쓴 낱말이 문서에 한 글자도 나오지 않는다는 공통점이 있습니다.

질의	찾아야 할 문서	키워드 검색	한국어 임베딩
애 낳고 쉴 수 있는 기간	육아휴직	놓침	찾음
집에서 일하는 건 일주일에 몇 번 가능해?	재택근무	놓침	찾음
서비스가 멈췄을 때 고객 응대는 얼마나 빨리	장애 문의 응답	놓침	찾음

반대로 「PG-101」, 「PG-102」, 「B-2270」처럼 제품 코드가 들어간 질의는 키워드 검색도 정답을 맞혔습니다. 정리하면 한국어 임베딩은 표현이 다른 질문에서, 키워드 검색은 정확한 문자열이 있는 질문에서 강하며, 이 차이가 두 방식을 합치는 하이브리드 검색(5장)의 근거가 됩니다. 다만 이 실험은 문서 27건과 질의 15건으로 만든 소규모 합성 문장 실험이라 경향을 보여 줄 뿐이고, 정답률 1.00이 실제 업무 환경에서도 그대로 나온다는 뜻으로 읽어서는 안 됩니다. 문서가 수만 건으로 늘고 비슷한 규정이 여러 판으로 쌓이면 정답률은 달라지므로, 같은 방식의 평가를 자기 조직 문서로 다시 해 보아야 합니다.

4.3.3 모델 카드 점수를 자기 조직 문서로 다시 재는 임베딩 평가 절차

모델 카드의 점수는 후보를 좁히는 데 쓰고, 최종 선택은 자기 조직 문서로 잰 결과를 따릅니다. 공개 데이터셋은 뉴스·백과사전 문장이 많아서 결재 규정이나 장애 보고서처럼 형식이 정해진 업무 문서와 성격이 다르기 때문입니다. 따라서 공개 점수와 자기 문서 결과가 다르면 자기 문서 결과를 기준으로 삼는다는 원칙을 도입 계획에 미리 적어 두는 것이 좋습니다. 검색 품질을 답변 품질과 따로 재는 방법은 RAG 평가 체계인 RAGAS의 문맥 관련성 지표를 참고할 수 있으며, 지표 구성은 6장 6.3.2절에서 정리합니다[S12].

단계	할 일	점검 기준
평가 세트 만들기	실제 직원 질문을 수십 개에서 100개 안팎 모으고, 질문마다 정답 문서를 짝지음	구어 질문·코드 질문·다른 표현 질문이 고루 섞였는가
후보 고르기	공개 비교표에서 한국어 추가 학습 모델 2–3개와 비교용 다국어 모델 1개를 고름	최대 입력 토큰과 벡터 차원이 운영 환경에 맞는가
같은 조건으로 실행	같은 문서 조각, 같은 형태소 분석 설정으로 색인하고 질의를 돌림	조각 크기·전처리가 모든 후보에 같게 적용됐는가
지표 비교	정답률@1·정답률@3과 질의당 응답 시간을 함께 기록	정답률 차이가 응답 시간 증가를 감수할 만큼 큰가
판정과 기록	놓친 질의를 유형별로 나누고 선택 이유를 남김	다음 모델 교체 때 같은 평가 세트로 다시 잴 수 있는가

이 절차를 거치면 이 장의 판단인 「우리 문서로 다시 재 볼 임베딩 후보 2–3개」가 구체적인 평가 결과로 바뀝니다. 평가 세트를 한 번 만들어 두면 모델을 바꾸거나 문서가 늘어날 때마다 같은 질문으로 다시 잴 수 있어, 8장의 파일럿 점검표에서 검색 품질 항목의 기준 자료로 그대로 쓸 수 있습니다.

5장: 키워드와 의미를 함께 찾는 하이브리드 검색과 리랭킹 설계

4장에서 고른 형태소 분석과 한국어 임베딩은 각각 키워드 검색과 의미 검색의 품질을 정합니다. 하이브리드 검색은 이 두 검색, 곧 키워드 검색(BM25)과 의미 검색(임베딩 벡터 검색)을 함께 돌린 뒤 두 결과를 하나의 순위로 합치는 검색 구성입니다. 업무 질문에는 제품 코드처럼 글자가 정확히 맞아야 하는 질문과 일상 표현으로 던지는 질문이 섞여 있어서, 한쪽 검색만으로는 질문 전체를 처리하지 못합니다. RAG를 운영 단계로 올리는 조직은 대부분 두 검색을 합치고, 마지막에 리랭커(reranker, 후보 문서를 질문과 함께 다시 읽어 순위를 매기는 모델)로 상위 순위를 다듬습니다. 다만 두 검색을 같은 무게로 합치기만 하면 한국어 임베딩 하나만 쓸 때보다 정답률이 떨어질 수도 있습니다. 결합 방식과 가중치는 기본값으로 두지 말고 자기 조직의 질문으로 재서 정해야 하며, 자기 조직의 질문 유형에 맞는 결합 구성을 고르는 판단 기준을 아래에 차례로 정리했습니다.

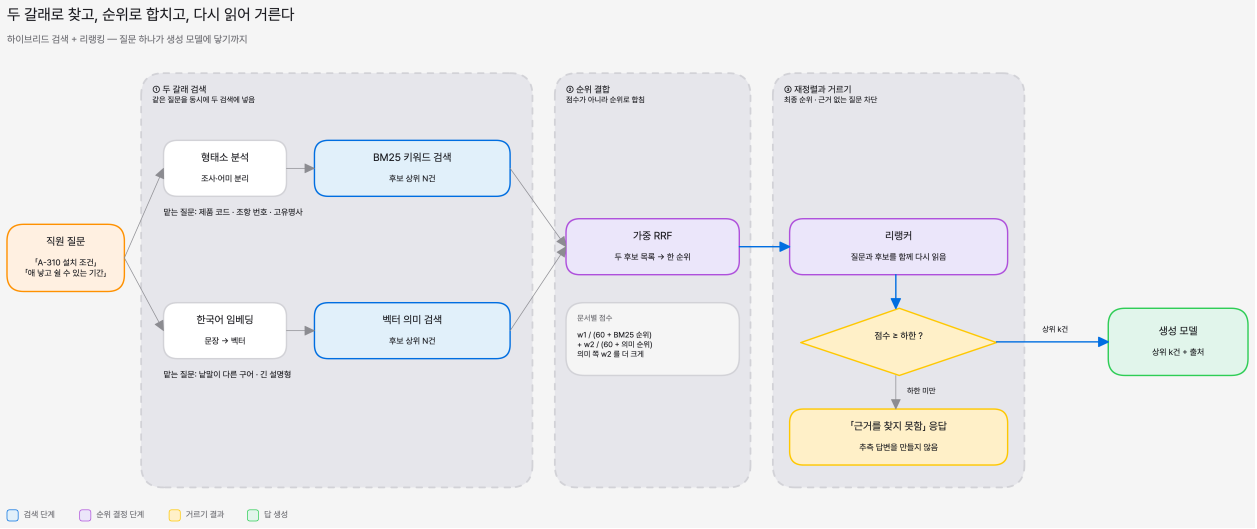

그림. 키워드 검색과 의미 검색의 결과를 순위로 합친 뒤 리랭커로 다시 정렬하는 하이브리드 검색 구조

5.1 키워드 검색과 의미 검색이 서로 놓치는 질문 유형

키워드 검색과 의미 검색은 약한 질문이 서로 다릅니다. 키워드 검색은 질문과 문서에 같은 낱말이 있어야 찾고, 의미 검색은 뜻이 가까운 문장을 찾는 대신 글자 하나 차이로 갈리는 코드나 번호를 구분하는 힘이 약합니다. 따라서 도입을 검토하는 의사결정권자가 먼저 확인할 사항은 자기 조직 업무 질문이 어느 쪽에 더 많이 몰려 있는지입니다. 질의 유형별 강약을 정리하면 다음 표와 같습니다.

질의 유형	예	키워드 검색(BM25)	의미 검색(임베딩)
코드·번호	「A-310 모델 설치 조건」	강함 — 같은 문자열이 있는 문서를 바로 찾음	약함 — 번호가 다른 비슷한 문서를 함께 가져옴
조항·고유명사	「제17조 2항의 적용 대상」	강함	보통 — 조항 번호를 구분하지 못할 때가 있음
수치 조건	「3년 이상 근속자 포상 기준」	보통 — 숫자 표기가 같아야 맞음	약함 — 숫자의 크기 차이를 잘 반영하지 못함

질의 유형	예	키워드 검색(BM25)	의미 검색(임베딩)
동의어·구어	「애 낳고 쉴 수 있는 기간」	약함 — 문서의 「육아휴직」과 낱말이 겹치지 않음	강함 — 뜻이 같은 조항을 찾음
긴 설명형 질문	「서비스가 멈췄을 때 고객 응대 기준」	약함 — 흔한 낱말에 점수가 흩어짐	강함

표의 예시 문자열은 설명을 위해 만든 가상의 형식입니다.

낱말 대신 사람과 시기로 기억되는 질문

표의 다섯 유형 밖에서 자주 나오는 질문이 하나 더 있습니다. 실무자의 기억은 낱말 하나로 오지 않고 「누가 썼던, 어느 종류의, 언제쯤 문서」라는 흐릿한 묶음으로 옵니다. 「그건 앞 담당자가 정리해 뒀을 텐데」라는 기억이 문서 제목보다 오래 남기 때문입니다. 이런 질문은 키워드 검색도 의미 검색도 본문만으로는 잘 찾지 못하므로, 작성자·문서 종류·작성 시기를 색인 속성으로 두고 검색 전에 범위를 좁히는 필터를 함께 씁니다. 이 필터는 검색 범위를 좁히는 데만 쓰고, 누가 무엇을 얼마나 썼는지 집계하는 인사 조회 용도로 넓히지 않도록 정해 두는 편이 현업의 거부감을 줄입니다.

5.1.1 제품 코드·법령 조항 번호처럼 정확한 문자열이 필요한 질문

정확 일치 질문에서 생기는 의미 검색의 오답

「A-310 모델의 설치 조건」을 물었는데 의미 검색이 A-301이나 A-320 설명서를 먼저 내놓는 경우가 정확 일치 질문의 대표적인 실패입니다. 임베딩은 문장 전체의 뜻을 벡터 하나로 압축하므로, 번호만 다르고 나머지 문장이 거의 같은 두 문서를 아주 가까운 벡터로 만듭니다. 「제17조 2항」과 「제17조 3항」은 사람에게는 전혀 다른 조항이지만, 같은 장 안에서 비슷한 문체로 쓰여 있으면 의미 검색에서는 순위가 뒤섞이기 쉽습니다. 반면 BM25는 「A-310」이라는 문자열을 하나의 검색어로 다루기 때문에, 그 문자열이 들어 있는 문서를 정확히 위로 올립니다. 제품 코드·조항 번호·계약 번호로 묻는 질문이 많은 구매, 법무, 기술지원 업무에서 키워드 검색을 빼면 안 되는 이유가 여기에 있습니다.

학습 분야 밖에서 튼튼했던 BM25 기준선

키워드 검색의 이런 강점은 공개 평가에서도 확인됩니다. 18개 데이터셋으로 검색 모델을 별도 학습 없이 비교한 BEIR 평가에서 BM25는 분야가 바뀌어도 성능이 크게 흔들리지 않는 튼튼한 기준선이었고, 의미 검색 모델은 학습한 분야 밖의 문서에서 BM25보다 못한 경우가 많았습니다[S4]. 사내 문서는 범용 임베딩 모델이 학습한 웹 문서와 어휘·문체가 다르므로, 이 결과는 기업 문서 검색에도 그대로 적용해 볼 만합니다. 4장 4.3.2절의 예시 실험(업무 문서 27건, 질의 15건)에서도 PG-101, PG-102, B-2270 같은 코드 질의는 키워드 검색이 모두 맞혔습니다. 따라서 RAG를 도입하더라도 키워드 검색은 버릴 구성 요소가 아니며, 정확한 문자열을 맡는 한 축으로 계속 두어야 합니다.

5.1.2 표현이 달라도 뜻이 같은 문장을 찾아야 하는 질문

낱말이 겹치지 않는 질문을 찾는 의미 검색

2장 2.1.1에서 본 표현 차이 문제는 의미 검색이 맡습니다. 직원은 「애 낳고 쉴 수 있는 기간」이라고 묻지만 인사 규정에는 「육아휴직」이라고 적혀 있고, 「집에서 일하는 건 일주일에 몇 번 가능해?」라는 질문의 정답 조항은 「재택근무」라는 낱말로 시작합니다. 질문과 정답 문서 사이에 겹치는 낱말이 거의 없으므로 BM25는 이런 질문에

서 점수를 매길 근거를 찾지 못합니다. 4장 4.3.2절의 예시 실험에서도 이런 구어 질의 세 건은 키워드 검색이 모두 놓쳤고 한국어 임베딩은 모두 찾았습니다. 의미 검색은 이처럼 키워드 검색이 구조적으로 처리하지 못하는 질문을 맡는 축입니다.

MIRACL 한국어 평가로 본 두 검색의 격차

차이의 크기는 공개 평가에서 더 분명하게 드러납니다. 다국어 검색 평가셋 MIRACL의 한국어 부분에서 BGE-M3 논문이 보고한 nDCG@10(상위 10건 안에서 정답이 얼마나 위에 놓였는지 재는 지표)은 BM25가 37.1이었고, BGE-M3의 의미 검색(dense)은 69.9였습니다[S32]. BGE-M3가 낱말 가중치로 찾는 sparse 방식은 61.5로 그 중간에 놓였습니다.

검색 방식(MIRACL 한국어)	nDCG@10
BM25	37.1
sparse(모델이 매긴 낱말 가중치)	61.5
dense(의미 검색)	69.9

출처: BGE-M3 논문의 MIRACL 한국어 결과[S32]

다만 이 평가셋은 위키백과 문단 검색이라 코드·번호 질의가 적습니다[S32]. 일상 표현이 많은 사내 질문에서는 의미 검색이 주력이 되고, 코드 질문 비중이 큰 업무에서는 키워드 검색의 몫이 커진다고 보는 편이 맞습니다.

5.2 하이브리드 검색에서 두 결과를 합치는 RRF 와 단순 결합의 위험

두 검색 결과를 하나로 합치는 가장 널리 쓰이는 방법은 RRF(Reciprocal Rank Fusion, 역순위 결합)입니다. RRF는 점수 대신 순위만 쓰므로 설정이 간단하고, 두 검색이 고르게 좋을 때는 정확도를 확실히 더합니다[S5]. 하지만 한쪽 검색이 특정 질문에서 크게 약하면, 같은 무게로 합친 결과가 좋은 쪽 검색 하나보다 나빠지기도 합니다. 결합 방식은 하이브리드 검색을 켜는 순간 정해지는 기본값으로 다루면 안 되고, 결합 전후 정답률을 재서 가중치까지 정해야 하는 설계 항목입니다.

5.2.1 점수 대신 순위로 두 검색 결과를 합치는 RRF 의 원리

점수 척도가 다른 두 검색을 순위로 맞추는 방식

BM25 점수는 0부터 수십까지 퍼지고 임베딩 유사도는 0과 1 사이에 몰려 있어서, 두 점수를 그대로 더하면 한쪽 점수가 결과를 좌우합니다. RRF는 점수를 버리고 각 검색에서 문서가 몇 위였는지만 봅니다. 계산식은 한 줄입니다. 문서마다 「1 ÷ (k + 그 검색에서의 순위)」를 검색별로 구해 더하고, 합이 큰 순서로 다시 줄을 세웁니다. 원 논문은 k를 60으로 두었고, 이 값이 여러 평가에서 고르게 좋은 결과를 냈습니다[S5]. 예를 들어 키워드 검색 1위, 의미 검색 3위인 문서는 1/61과 1/63을 더해 약 0.032를 얻고, 의미 검색에서만 2위인 문서는 1/62, 약 0.016을 얻습니다. 결과적으로 두 검색이 함께 위로 올린 문서가 앞으로 나오는 구조입니다.

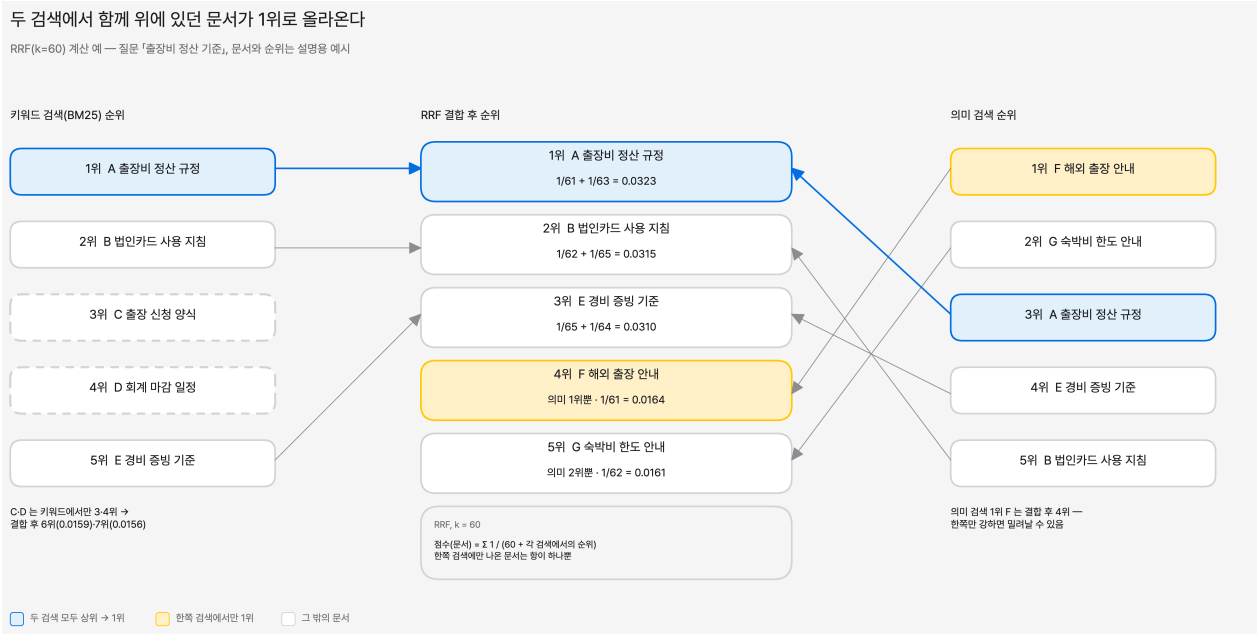

그림. 키워드 검색과 의미 검색의 순위를 RRF(k=60)로 합쳐 새 순위를 만드는 예

두 검색이 고르게 좋을 때 더해지는 결합 효과

두 검색이 각자 다른 질문을 맞히면서 전반적인 수준도 비슷하다면 결합은 성능을 더합니다. MIRACL 한국어 결과를 보면 dense 하나일 때 69.9였던 nDCG@10이 sparse를 합치자 71.2로, 문서 조각의 여러 벡터를 함께 비교하는 방식까지 모두 합치자 72.1로 올랐습니다[S32].

결합 구성(MIRACL 한국어)	nDCG@10
dense 단독	69.9
dense + sparse	71.2
전체 결합(dense + sparse + 다중 벡터)	72.1

출처: BGE-M3 논문[S32]

이 결과에서 눈여겨볼 조건은 sparse의 단독 성능(61.5)이 dense와 크게 벌어지지 않았다는 점입니다. 결합할 두 검색의 수준이 비슷할수록 RRF가 더하는 몫이 커집니다.

5.2.2 한국어 임베딩 1.00 이 단순 RRF 결합으로 0.67 이 된 예시 실험

업무 문서 27건 예시 실험의 결합 전후 정답률

업무 문서 27건과 질의 15건으로 만든 예시 실험에서는 결합이 오히려 정답률을 낮췄습니다. 정답 문서가 1위에 오른 비율(정답률@1)은 한국어 임베딩(BGE-m3-ko) 단독이 1.00이었는데, 한글 2글자 분할 BM25와 같은 무게의 RRF로 합치자 0.67로 떨어졌습니다. 정답이 3위 안에 든 비율(정답률@3)도 1.00에서 0.80으로 낮아졌습니다.

검색 구성	정답률@1	정답률@3
한글 2글자 분할 BM25	0.67	0.73

검색 구성	정답률@1	정답률@3
한국어 임베딩(BGE-m3-ko)	1.00	1.00
BM25(2글자) + 한국어 임베딩, 같은 무게 RRF	0.67	0.80

이 실험은 소규모 합성 문장으로 만든 것이라 수치 자체를 일반화할 수는 없고, 결합이 정답을 밀어낼 수 있다는 경향을 보여 주는 데 의미가 있으며, 나머지 검색 방식의 결과는 4장 4.3.2절의 표에 있습니다.

한쪽 검색이 약할 때 결합이 정답을 밀어내는 원리

원인은 RRF 계산 방식에 있습니다. 「애 낳고 쉴 수 있는 기간」처럼 BM25가 정답을 아예 찾지 못한 질의에서, 정답 문서는 의미 검색 1위 점수인 1/61만 얻습니다. 두 검색 모두에서 2위에 오른 다른 문서는 1/62를 두 번 더해 두 배 가까운 점수를 받으므로 정답보다 앞에 섭니다. 같은 무게의 RRF는 「두 검색이 함께 고른 문서」를 우대하기 때문에, 한쪽 검색이 놓친 정답은 다른 쪽에서 1위여도 밀려날 수 있습니다. 따라서 의미 검색이 훨씬 강한 한국어 문서 환경에서는 의미 검색 쪽 항에 더 큰 가중치를 곱하거나, 키워드 검색은 코드·번호 질의에만 반영하도록 질의 유형별로 나누는 설계가 필요합니다.

키워드 쪽 품질을 높이는 형태소 분석

가중치를 조정하기 전에 키워드 검색 자체를 한국어에 맞게 고치는 일도 함께 해야 합니다. 예시 실험에서 어절 단위 BM25는 0.47, 한글 2글자 분할 BM25는 0.67로, 같은 BM25라도 낱말을 자르는 방식만 바꿔 정답률이 크게 달라졌습니다. 「계약서를」「계약서의」를 모두 「계약서」로 맞추는 형태소 분석(4장 4.2)을 키워드 쪽에 적용하면, 결합에 들어가는 BM25 순위의 품질이 올라가 결합 손실도 줄어듭니다. 결합 순위를 한 번 더 다듬는 리랭커가 남은 오차를 맡습니다.

5.3 리랭커로 상위 후보의 순위를 다시 매기는 마지막 단계

리랭커는 1차 검색이 고른 후보 수십 건을 질문과 한 쌍씩 다시 읽고 관련도를 새로 매기는 모델입니다. 임베딩 검색은 질문과 문서를 따로 벡터로 만든 뒤 거리만 비교하지만, 리랭커는 질문과 문서를 함께 읽으므로 「육아휴직 기간」과 「육아휴직 신청 절차」처럼 주제는 같고 답은 다른 문서를 더 잘 가려냅니다. 하지만 후보마다 모델을 한 번씩 실행하므로 후보가 많을수록 응답 시간이 늘어납니다. 리랭커를 넣을지, 몇 건에 적용할지는 정확도와 응답 시간 목표를 함께 놓고 판단할 사항입니다.

5.3.1 문맥 임베딩에 BM25 와 리랭커를 더해 검색 실패율을 67% 줄인 결과

구성 요소를 더할 때마다 줄어든 검색 실패율

검색 구성 요소를 하나씩 더할 때 효과가 어떻게 쌓이는지 보여 주는 공개 결과가 있습니다. 한 AI 모델 개발사가 공개한 문맥 검색(Contextual Retrieval) 실험에서, 문서 조각마다 그 조각이 문서 전체에서 어떤 내용인지 설명하는 문장을 덧붙여 임베딩하자 상위 20건 안에서 정답 조각을 찾지 못하는 비율이 35% 줄었습니다. 같은 방식으로 문맥을 덧붙인 BM25를 함께 쓰자 감소 폭이 49%로 커졌고, 리랭커까지 더하자 67%가 되었습니다 [S6]. 다만 이 수치는 제품을 만든 회사가 직접 잰 결과이므로, 방향을 참고하는 자료로 쓰고 크기는 자기 조직 문서로 다시 확인해야 합니다.

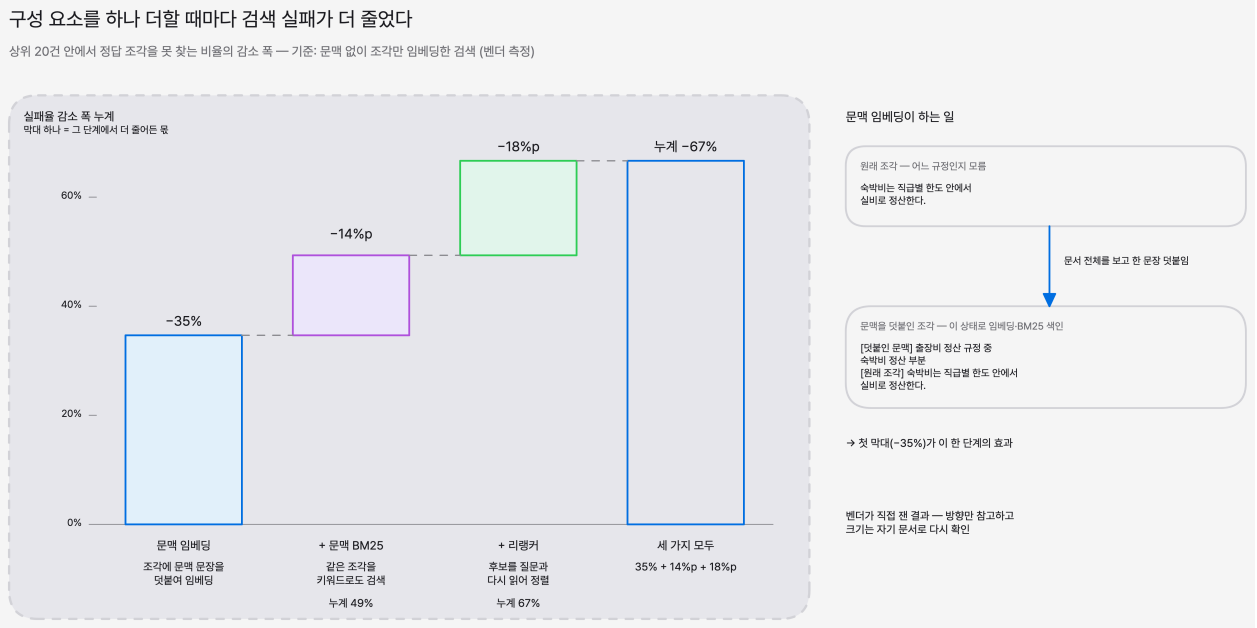

그림. 문맥 임베딩, BM25, 리랭커를 차례로 더할 때 상위 20건 검색 실패율이 줄어든 폭(벤더 측정)

단계적 도입과 단계별 측정

이 결과에서 얻을 수 있는 실무 원칙은 구성 요소를 한꺼번에 넣지 말고 하나씩 더하면서 재라는 것입니다. 앞의 예시 실험처럼 BM25를 더한 결과가 나빠질 수도 있으므로, 단계마다 정답률@1과 정답률@3을 기록해 두면 어느 구성 요소가 효과를 냈고 어느 것이 손실을 만들었는지 바로 가려낼 수 있습니다. 파일럿 단계에서는 한국어 임베딩 단독을 기준으로 두고, 형태소 분석 BM25 결합, 가중치 조정, 리랭커 추가 순서로 한 단계씩 비교하는 방식이 무난합니다. 업무별 질문 묶음을 따로 두고 재면, 코드 질의가 많은 업무와 구어 질의가 많은 업무에서 구성이 달라져야 하는지도 함께 확인할 수 있습니다.

5.3.2 한국어판 리랭커 Top-1 F1 0.9123 과 응답 시간 사이의 균형

한국어로 다시 학습한 리랭커의 정확도

리랭커도 임베딩과 마찬가지로 한국어 데이터로 다시 학습한 모델을 고르는 편이 낫습니다. 공개 다국어 리랭커 bge-reranker-v2-m3를 한국어로 추가 학습한 bge-reranker-v2-m3-ko는 모델 카드에 공개한 평가에서 1위 후보의 정확도를 재는 Top-1 F1이 0.9123으로, 원본 모델의 0.8772보다 높았습니다[S36]. 한국어 질문과 문서의 조사·어미 변화를 더 많이 학습했기 때문에, 주제는 같고 답이 다른 문서를 구분하는 힘이 커진 것으로 볼 수 있습니다.

리랭커	Top-1 F1
bge-reranker-v2-m3(원본)	0.8772
bge-reranker-v2-m3-ko(한국어 추가 학습)	0.9123

출처: bge-reranker-v2-m3-ko 모델 카드[S36]

적용 후보 수와 응답 시간을 함께 정하는 기준

재순위 방식은 BEIR 평가에서 가장 정확한 구성이었지만 계산 비용도 가장 컸습니다[S4]. 리랭커의 응답 시간은 후보 수에 비례해 늘어나므로, 적용 범위는 먼저 응답 시간 목표를 정한 뒤 그 안에 들어오는 후보 수로 정합니다. 리랭커 점수에 하한을 두면 관련도가 낮은 후보를 생성 모델에 넘기지 않게 되고, 하한을 넘는 후보가 하나도 없을 때는 「근거 문서를 찾지 못했다」고 답하게 만들 수 있습니다. 판단 기준을 정리하면 다음과 같습니다.

판단 항목	기준
리랭커 적용 여부	결합 검색의 정답률@3이 정답률@1보다 크게 높으면 적용 효과가 큼(정답이 상위권에 있으나 1위가 아닌 경우가 많다는 뜻)
적용 후보 수	응답 시간 목표 안에서 처리할 수 있는 최대 건수로 시작하고, 후보 수를 줄여도 정답률이 유지되는 최소값을 찾음
점수 하한	정답이 없는 질문 묶음을 따로 두고, 그 질문들이 하한에 걸려 「근거 없음」으로 처리되는 값을 고름
모델 선택	원본과 한국어판을 같은 질문 묶음으로 비교하고, 한국어판의 정확도 이점이 응답 시간 증가를 상쇄하는지 확인

결국 하이브리드 검색과 리랭커의 구성은 문서와 질문의 성격에 따라 정답이 달라지는 설계 문제입니다. 파일럿 대상 업무의 실제 질문으로 결합 전후와 리랭커 전후를 재고, 그 수치로 가중치·후보 수·점수 하한을 정하는 순서를 지키면 「하이브리드면 항상 낫다」는 가정으로 생기는 손실을 피할 수 있습니다.

6장: RAG 단독 운영에서 드러나는 한계와 보완 기술

4장과 5장의 한국어 검색 구성을 갖추더라도, RAG가 처음 구축한 상태 그대로 모든 업무 질문에 정확히 답하지는 못합니다. 실제 RAG 시스템 세 곳을 분석한 연구는 실패가 문서 준비, 검색, 답변 생성 단계에서 고르게 생기며 RAG의 견고성은 설계 단계보다 운영 과정에서 다듬어진다고 결론지었습니다[S7]. 1장 1.3절에서 본 법률 질의 연구처럼 근거 문서를 붙여도 환각은 남기 때문에[S2], RAG가 오답을 크게 줄이더라도 0으로 만들지는 못한다는 점을 전제로 두고 파일럿을 계획할 때 어떤 실패가 어느 단계에서 생기고 무엇으로 보완하는지를 미리 정해 두어야 합니다. 이 장에서는 RAG의 실패 유형 일곱 가지, 한계별 보완 기술 대응표, 권한 누수와 악성 문서 주입 대책, 품질 평가 지표를 차례로 정리했습니다.

6.1 문서 준비부터 답변 생성까지 RAG 가 실패하는 일곱 가지 지점

RAG가 틀린 답을 내면 언어 모델의 문제로 보기 쉽지만, 원인은 그보다 앞선 문서 준비나 검색 단계에 있는 경우가 적지 않습니다. Barnett 등은 RAG의 실패를 일곱 가지 유형으로 나눴고, 각 유형은 문서 색인, 검색 순위, 문맥 구성, 답 생성 가운데 한 단계에 속합니다[S7]. 따라서 틀린 답이 나왔을 때 답 생성 단계부터 문서 준비 단계까지 거슬러 올라가며 점검하는 절차가 있어야 원인을 빨리 찾을 수 있습니다. 아래 표는 일곱 가지 실패 유형을 단계, 증상, 점검 방법으로 정리한 것입니다.

단계	실패 유형	증상	점검 방법
문서 준비	문서 누락	답할 문서가 색인에 없는데 그럴듯한 답이 나옴	질문별 정답 문서가 색인에 있는지 대조
검색	상위 후보 탈락	정답 문서가 상위 후보 밖으로 밀림	질의별 정답 문서의 검색 순위 기록
문맥 구성	문맥 제외	검색은 됐으나 모델 입력 한도 때문에 빠짐	모델에 실제로 넣은 조각 목록 확인
답 생성	답 추출 실패	입력에 정답이 있으나 다른 내용에 묻혀 놓침	입력 조각과 답을 나란히 비교
답 생성	형식 오류	표·목록 같은 요구 형식을 지키지 않음	형식 지시와 출력 대조
답 생성	구체성 부적합	답이 지나치게 일반적이거나 지나치게 세부적	질문 유형별 기대 답 수준 정의
답 생성	불완전한 답	여러 항목을 묻는 질문에 일부만 답함	복수 항목 질문의 누락 항목 수 집계

(출처: Barnett 외 2024의 일곱 가지 실패 유형을 업무 점검 항목으로 재구성[S7])

6.1.1 문서가 없거나 검색에서 빠져 틀린 답이 나오는 경우

색인에 없는 문서와 추측 답변

첫 번째 실패 유형은 답이 담긴 문서가 처음부터 색인에 없는 경우입니다. 「올해 바뀐 출장 정산 기준」을 묻는데 개정 공문이 아직 등록되지 않았다면, 모델은 이전 규정 조각이나 일반 상식을 바탕으로 그럴듯한 답을 만들어 냅니다. 답에 출처가 붙어 있으니 사용자는 그 내용을 믿기 쉽고, 결국 개정 전 규정에 따라 업무를 처리하게 됩니다. 그래서 질문에 맞는 문서가 없을 때 「제공된 문서에서 근거를 찾지 못했습니다」라고 답하도록 지시하고, 검색 점수가 정한 하한보다 낮으면 답을 만들지 않도록 설정해야 합니다[S7]. 색인 갱신 주기가 길어도 같은 문제가 생기는데, 규정이 바뀐 뒤 색인에 반영되기 전까지는 문서가 있어도 낡은 판이 답의 근거로 쓰이기 때문입니다.

상위 후보에서 밀린 정답 문서

두 번째 유형은 정답 문서가 색인에는 있지만 검색 결과의 상위 후보에 들지 못하는 경우입니다. RAG는 보통 상위 몇 개 조각만 모델에 넣기 때문에, 정답 조각이 6위에 있고 상위 5개만 쓴다면 모델 입장에서는 문서가 없는 것과 같으므로 언어 모델을 더 큰 모델로 바꿔도 문제는 풀리지 않습니다. 질의마다 정답 문서의 검색 순위를 기록해 두면, 틀린 답이 검색 단계에서 생겼는지 생성 단계에서 생겼는지 가를 수 있습니다. 한국어 문서에서는 임베딩 모델 선택과 형태소 분석이 이 순위를 크게 바꾸고(4장), 하이브리드 검색과 리랭커가 밀린 후보를 다시 위로 올립니다(5장).

운영 과정에서 다듬어지는 견고성

일곱 가지 실패 유형 대부분은 구축 전에 미리 알아내기 어렵고, 실제 사용자 질문이 들어와야 드러납니다. Barnett 등이 RAG의 견고성은 처음부터 설계로 완성되기보다 운영하면서 다듬어진다고 정리한 것도 같은 이유입니다[S7]. 파일럿 계획에는 답변 품질 목표와 함께, 틀린 답을 모으고 위 표의 단계별로 분류하는 운영 절차가 들어가야 합니다. 틀린 답을 단계별로 나눠 보면 문서 보강, 검색 설정 조정, 모델 지시 수정 가운데 무엇을 먼저 해야 하는지가 정해집니다.

6.1.2 여러 문서를 엮어야 하는 질문과 긴 입력 중간에서 약해지는 RAG

여러 문서를 엮어야 하는 질문

단일 문서 질문은 조각 하나만 찾으면 답이 나오며, 「연차휴가 이월 한도는 며칠입니까」는 취업규칙의 한 조항으로 답할 수 있습니다. 반면 「육아휴직 중인 직원이 사내 교육비 지원을 받을 수 있습니까」는 휴직 규정, 교육비 지원 지침, 복무 규정을 함께 읽어야 답이 나옵니다. MultiHop-RAG 연구는 이처럼 여러 문서의 근거를 이어 붙여야 답할 수 있는 질문으로 데이터셋을 만들었고, 기존 RAG 방식은 검색과 답변 양쪽에서 모두 만족스러운 수준에 이르지 못했습니다[S8]. 검색은 질문 전체와 가장 비슷한 조각을 찾는 구조라서, 질문의 조건 일부만 담은 문서는 순위가 밀리기 쉽습니다. 규정 여러 개를 함께 봐야 하는 질문이 많은 업무라면 6.2의 보완 기술을 처음부터 검토 대상에 넣어야 합니다.

긴 입력의 중간에 놓인 정보

검색 조각을 많이 넣으면 정답 조각이 포함될 확률은 오르지만, 모델이 그 조각을 제대로 쓰는지는 별개의 문제입니다. Liu 등은 관련 정보가 입력의 처음이나 끝에 있을 때보다 중간에 있을 때 언어 모델의 성능이 크게 떨어진다는 결과를 보고했습니다[S14]. 조각 20개를 넣었는데 정답 조각이 10번째에 있으면, 모델이 그 내용을 충분히 반영하지 못할 수 있습니다. 그래서 조각 수를 무작정 늘리기보다 리랭커로 상위 몇 개를 정확히 고르고, 관련도가 가장 높은 조각을 입력 앞쪽에 두는 배치가 필요합니다. 이 현상은 문서를 통째로 넣는 긴 문맥 입력에서도 똑같이 나타나므로, 어느 방식을 쓰든 같은 점검이 필요합니다.

6.2 한계마다 맞는 그래프 RAG·문맥 덧붙임·긴 문맥 라우팅 대응표

6.1의 실패 유형은 원인이 서로 다르므로 보완 기술도 유형마다 다릅니다. 모든 보완 기술을 한꺼번에 넣으면 구축 부담과 응답 시간이 함께 늘고, 어떤 기술이 효과를 냈는지 가리기도 어려워집니다. 파일럿에서는 실패 사례로 한계가 확인된 곳에만 보완 기술을 하나씩 더하는 순서가 바람직합니다. 아래 대응표는 RAG 단독 운영에서 자주 드러나는 한계를 원인, 보완 기술, 적용 조건과 짝지은 것입니다.

한계	원인	보완 기술	적용 조건	근거
근거 없는 추측 답변	문서 누락, 낮은 검색 점수에도 답 생성	「찾지 못함」 응답 지시, 검색 점수 하한, 출처 표시 의무화	모든 업무의 기본 설정	[S2][S7]
정답 문서 순위 밀림	표현 차이, 한국어 형태 변화	한국어 임베딩, 하이브리드 검색, 리랭커	검색 순위 기록에서 탈락이 확인될 때	[S6]
조각만으로 뜻이 모호함	조각이 문서 제목과 앞뒤 문맥을 잃음	문맥 덧붙임 임베딩	비슷한 문장이 여러 문서에 반복될 때	[S6]
다중 문서 질문 실패	조건 일부만 담은 문서의 순위가 밀림	질문 분해와 반복 검색	규정 여러 개를 엮는 질문이 많을 때	[S8]
전체 요약형 질문 실패	상위 조각 몇 개로는 문서 묶음 전체를 볼 수 없음	그래프 기반 RAG(GraphRAG)	「전체 쟁점」류 질문이 반복될 때	[S9]
입력 중간 정보 누락	긴 입력의 중간 정보 활용 저하	리랭커로 조각 수 축소, 관련 조각 앞쪽 배치	투입 조각이 많을 때	[S14]
응답 시간과 처리량 증가	긴 문맥 일괄 입력, 보완 단계 누적	RAG·긴 문맥 라우팅	질문 난이도의 편차가 클 때	[S13]
개정 전 내용으로 답함	색인 갱신 지연	원본 변경 감지 재색인, 답에 문서 판 표시	규정 개정이 잦은 업무	[S7]

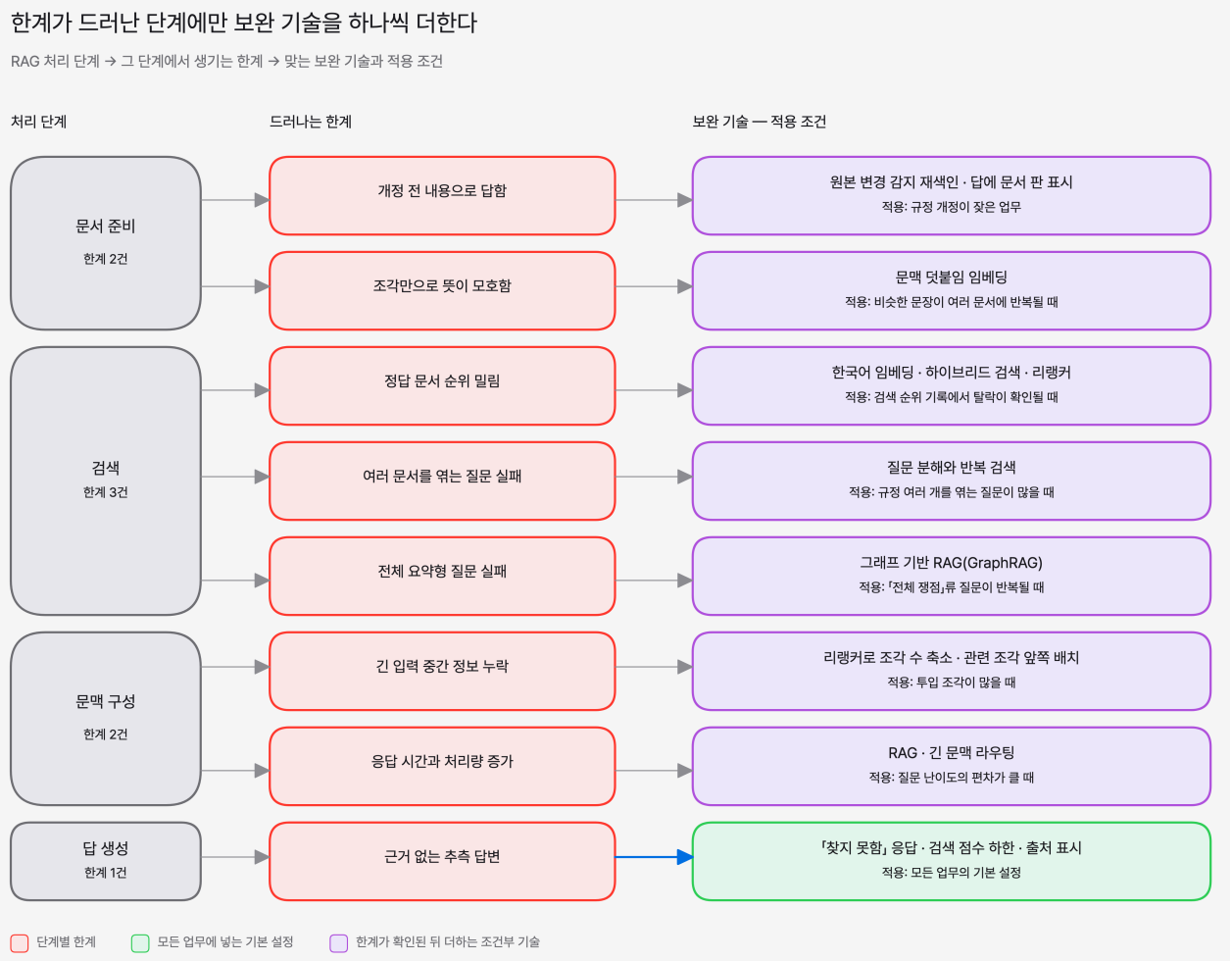

그림. RAG 처리 단계별 한계와 보완 기술의 대응

6.2.1 전체 요약형 질문과 다중 문서 질문을 위한 그래프 기반 RAG 와 반복 검색

전체 요약형 질문용 그래프 RAG

일반 RAG는 질문과 비슷한 조각 몇 개를 찾아 답하기 때문에, 「지난 2년간 고객 불만 보고서에서 반복된 주요 쟁점은 무엇입니까」 같은 질문에 약합니다. 이런 질문의 답은 특정 조각 하나에 있지 않고 문서 묶음 전체에 흩어져 있기 때문입니다. GraphRAG 연구는 문서에서 개체와 관계를 뽑아 그래프를 만들고, 서로 밀접한 개체 묶음마다 요약을 미리 만들어 두는 방식으로 이 문제를 다뤘습니다. 전체 요약형 질문에서 이 방식은 일반 RAG와 비교해 답의 포괄성과 다양성 승률이 약 70%–80%였습니다[S9]. 다만 이 수치는 LLM이 두 답을 나란히 놓고 심사한 결과이므로, 자기 업무 질문으로 사람이 다시 확인할 필요가 있습니다.

그래프 구축 부담과 도입 시점

그래프 기반 RAG는 색인 단계에서 문서 전체를 LLM으로 읽어 개체와 관계를 뽑으므로, 일반 RAG보다 색인 시간과 연산량이 크게 늘어납니다. 문서가 바뀔 때마다 그래프와 요약을 다시 만들어야 하는 운영 부담도 따라옵니다. 그래서 그래프 기반 RAG는 처음부터 넣기보다, 파일럿 질문 기록에서 전체 요약형 질문이 반복된다는 점이 확인된 뒤에 넣는 편이 합리적입니다. 규정 조항 하나를 찾는 질문이 대부분인 인사·총무 문의라면, 한국어 임베딩과 하이브리드 검색만으로도 충분할 수 있습니다.

질문을 나눠 찾는 반복 검색

다중 문서 질문에는 그래프보다 가벼운 보완책으로 반복 검색이 있으며, 질문을 하위 질문으로 나누고 앞선 검색 결과를 보고 다음 검색어를 정하는 과정을 몇 차례 되풀이하는 방식입니다. 앞의 육아휴직자 교육비 질문이라면 「육아휴직 중 복무 상태」와 「교육비 지원 대상 요건」으로 나눠 각각 검색한 뒤 두 결과를 합쳐 답합니다. 이 방식은 그래프를 따로 만들 필요가 없어 기존 색인을 그대로 쓸 수 있습니다. 하지만 검색과 모델 호출이 여러 번 일어나 응답 시간이 늘고, 질문 분해가 틀리면 관련 없는 문서를 모으게 됩니다. 반복 횟수 상한과 단계별 검색 기록을 함께 두어야 실패했을 때 원인을 추적할 수 있습니다.

6.2.2 조각에 문맥을 덧붙이는 임베딩과 RAG·긴 문맥 라우팅

조각에 문맥을 덧붙이는 임베딩

문서를 조각으로 나누면 각 조각은 원래 문서의 제목과 앞뒤 맥락을 잃습니다. 「지급 기간은 최대 12개월로 한다」라는 조각만 보면 어느 규정의 어떤 지원 항목인지 알 수 없고, 비슷한 문장이 여러 규정에 있으면 검색이 엉뚱한 조각을 고르게 됩니다. 문맥 덧붙임(Contextual Retrieval) 방식은 색인 전에 LLM이 각 조각에 「이 조각은 교육비 지원 지침의 지급 조항에 속한다」 같은 짧은 설명을 붙인 뒤 임베딩합니다. 이 방식이 상위 20개 검색 실패율을 줄인 폭은 5장 5.3.1절에서 보았듯이 제작사가 직접 잰 결과이고 데이터셋에 따라 효과가 달라지므로, 자기 문서로 다시 재 보아야 합니다[S6]. 조각마다 LLM을 한 번씩 호출하므로 문서량이 많을수록 색인 시간과 연산량이 함께 늘어납니다.

아래 그림은 문서 글자 추출부터 답변까지 다섯 단계마다 언어 모델을 써서 검색 품질을 높이는 구성 예입니다. 셋째 단계에서 조각에 앞뒤 문맥 정보와 예상 질문을 덧붙이는 부분이 이 절의 문맥 덧붙임에 해당합니다.

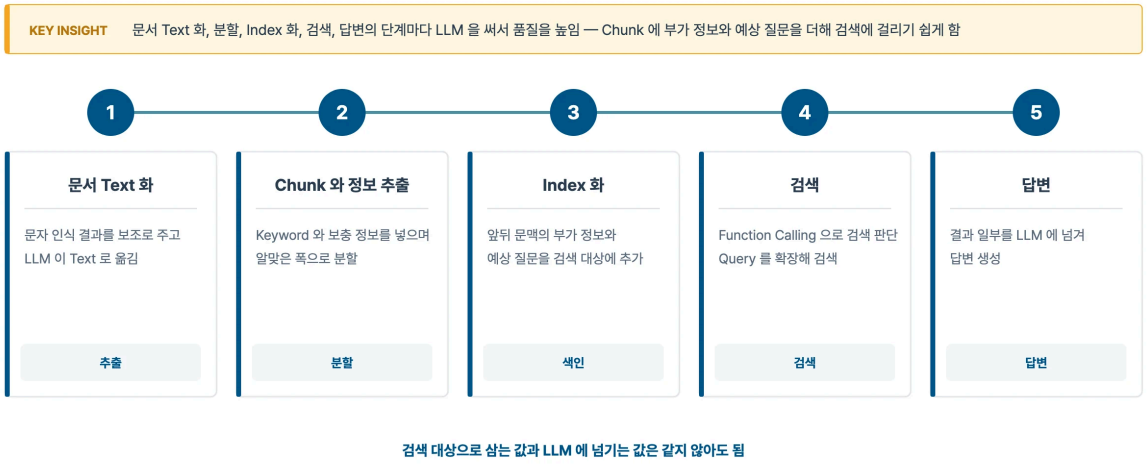

그림. 단계마다 언어 모델을 써서 검색 품질을 높이는 RAG 구현 다섯 단계

그림 아래쪽 문장처럼 검색 대상으로 삼는 값과 언어 모델에 넘기는 값은 같지 않아도 됩니다. 덧붙인 설명과 예상 질문은 검색에 걸리게 하는 데 쓰고, 답을 만들 때는 원문 조각을 넘기면 답의 근거가 원문에서 벗어나지 않습니다. 이렇게 나누면 덧붙인 설명이 틀렸더라도 그 오류가 답 문장으로 옮겨 가지 않고 검색 순위에만 영향을 줍니다.

RAG와 긴 문맥 라우팅

최근 언어 모델은 수십만 토큰을 한 번에 읽을 수 있어, 문서를 통째로 넣는 긴 문맥 입력이 RAG의 대안으로 거론됩니다. Li 등은 자원을 충분히 쓰면 긴 문맥 입력의 평균 성능이 RAG보다 높지만 처리 비용이 크게 늘어난다는 점을 확인하고, 두 방식을 섞는 Self-Route를 제안했습니다[S13]. Self-Route는 먼저 RAG로 답을 시도하고, 모델이 검색 결과만으로는 답할 수 없다고 판단한 질문만 긴 문맥 입력으로 넘깁니다. RAG로 처리한 질의 비율과 비용 절감 폭은 2장 2.3.2절에서 정리했으며, 그 절감 폭은 연구 조건에서 긴 문맥 입력만 쓸 때와 비교한 상대 비율이므로 실제 효과는 문서량과 질문 분포에 따라 달라집니다. 지식 기반 전체가 작다면 검색 단계 없이 통째로 넣는 편이 나을 수 있다는 기준도 2.3.2절과 같습니다.

문맥 덧붙임과 라우팅의 적용 조건

두 기술은 효과와 운영 부담이 서로 다른 단계에서 생기므로, 도입 판단도 따로 해야 합니다. 문맥 덧붙임은 색인 단계에 부담이 몰리고 질의 응답 시간에는 영향이 적은 반면, 라우팅은 색인 부담이 없는 대신 긴 문맥으로 넘어간 질문의 응답 시간이 길어집니다. 아래 표는 두 기술의 기대 효과, 운영 부담, 맞는 조건을 같은 기준으로 비교한 것입니다.

보완 기술	기대 효과	운영 부담	맞는 조건
문맥 덧붙임 임베딩	상위 20개 검색 실패율 35%–67% 감소(조합에 따라)	조각마다 LLM 호출, 재색인 시간 증가	비슷한 문장이 여러 규정·서식에 반복되는 문서
RAG·긴 문맥 라우팅	질의 82% RAG 처리, 비용 39%–65% 감소(연구 조건)	긴 문맥으로 넘어간 질문의 응답 지연	대부분은 단순 질문이고 일부만 문서 전체 이해가 필요한 업무

(출처: 문맥 덧붙임은 [S6]의 제작사 측정, 라우팅은 [S13]의 연구 조건 수치)

6.3 다른 부서 문서가 검색되는 권한 누수와 악성 문서 주입 대응

RAG를 한 부서에서 전사로 넓히면 답변 품질 문제와 함께 보안 문제가 새로 생깁니다. 여러 부서의 문서가 한 벡터 저장소에 모이면 검색 결과에 권한 밖 문서가 섞일 수 있고, 누구나 문서를 올릴 수 있으면 의도적으로 조작한 문서가 답을 바꿀 수 있습니다. 운영 중 품질을 재는 기준이 없으면 설정을 바꾼 뒤 품질이 나아졌는지 판단할 근거가 없습니다. 따라서 보안 검토와 평가 체계는 확산 단계가 아닌 파일럿 단계부터 설계에 들어가야 합니다.

6.3.1 권한 인식 저장소와 데이터 분할로 막는 문서 권한 누수

다른 부서 문서의 권한 누수

OWASP는 LLM 애플리케이션 10대 위험 2025년판에서 벡터·임베딩 취약점을 LLM08로 분류했습니다[S10]. 여러 사용자 그룹이 같은 벡터 데이터베이스를 쓰면, 한 그룹의 질문에 다른 그룹만 볼 수 있는 문서가 검색될 수 있습니다. 일반 직원이 「올해 성과 평가 기준」을 물었는데 인사팀 전용 평가 자료 조각이 답의 근거로 나오는 경우가 그 예입니다. 원본 문서 시스템에 권한이 걸려 있어도 색인 과정에서 그 권한 정보를 함께 옮기지 않으면, 검색 단계에서는 권한이 사라집니다. OWASP가 권고하는 대책은 사용자 권한을 반영해 검색 결과를 거르는 권한 인식 저장소와, 부서·보안 등급별로 데이터를 논리적으로 나누는 데이터 분할입니다[S10].

임베딩 벡터의 원문 복원 공격

임베딩은 문장을 숫자 벡터로 바꾼 값이므로 원문이 그대로 드러나지 않는다고 여기기 쉽습니다. 하지만 OWASP는 임베딩에서 원문 정보를 상당 부분 되살리는 임베딩 역변환 공격을 같은 항목의 위험으로 들었습니다[S10]. 따라서 벡터 저장소도 원본 문서와 같은 보안 등급으로 관리해야 하며, 접근 통제·저장 암호화·백업 관리·접근 기록을 원본 문서 저장소의 기준에 맞춥니다. 외부 임베딩 서비스를 쓴다면 문서 조각이 조직 밖으로 전송된다는 점도 보안 검토 항목에 넣어야 합니다.

악성 문서 주입과 등록 승인

PoisonedRAG 연구는 공격자가 특정 질문을 노려 지식 기반에 조작 문서 5건만 넣어도 공격 성공률이 약 90%에 이른다는 결과를 보였습니다[S11]. 조작 문서는 노린 질문과 비슷한 표현으로 써서 검색 상위에 오르게 하고, 그 안에 공격자가 원하는 답을 담는 방식입니다. 연구진이 시험한 질문 바꿔 쓰기, 문장 이상도 기반 탐지 같은 기존 방어 기법은 이 공격을 충분히 막지 못했습니다[S11]. 그러므로 탐지 기술에만 기대기보다 문서가 지식 기반에 들어오는 경로를 통제해야 합니다. 등록 권한을 문서 담당 부서로 제한하고, 새 문서는 승인을 거친 뒤 색인하며, 답마다 출처 문서와 등록자를 남겨 문제가 생기면 해당 문서를 바로 찾아 내릴 수 있게 합니다.

권한 누수, 임베딩 원문 복원, 악성 문서 주입 세 가지 위험의 발생 방식과 대책을 정리하면 다음과 같습니다.

위험	발생 방식	대책
권한 누수	공용 벡터 저장소에서 권한 정보 없이 검색	권한 인식 저장소, 부서·등급별 데이터 분할, 검색 시 사용자 권한 필터
임베딩 원문 복원	임베딩 역변환 공격	벡터 저장소를 원본과 같은 등급으로 관리, 접근 통제와 저장 암호화
악성 문서 주입	노린 질문과 비슷하게 쓴 조작 문서 등록	등록 권한 제한, 승인 후 색인, 답마다 출처와 등록자 기록

(출처: 권한 누수와 원문 복원은 OWASP LLM08[S10], 악성 문서 주입은 PoisonedRAG[S11])

6.3.2 검색 품질과 답변 품질을 따로 재는 RAGAS 평가 체계

검색 품질과 답변 품질의 분리 측정

RAG의 답이 틀렸을 때 원인이 검색인지 생성인지 가르려면 두 단계를 따로 재야 합니다. RAGAS는 사람이 정답을 일일이 달지 않아도 LLM을 이용해 RAG 품질을 자동으로 재는 평가 체계로, 근거 충실도, 답변 관련성, 문맥 관련성 세 가지를 핵심 지표로 제안했습니다[S12]. 근거 충실도는 답에 담긴 주장이 가져온 문서에 실제로 들어 있는지를, 문맥 관련성은 검색한 조각이 질문에 필요한 내용을 중심으로 담고 있는지를 봅니다. 근거 충실도가 높은데 답이 틀렸다면 검색이 잘못된 문서를 가져온 것이고, 문맥 관련성이 높은데 근거 충실도가 낮다면 모델이 문서 밖의 내용을 지어낸 것입니다. 지표를 이렇게 나누면 6.1의 실패 유형 가운데 어느 단계를 고쳐야 할지가 드러납니다. 아래 표는 RAGAS의 세 지표와 보조 지표인 정답률@k를 뜻과 보는 단계로 정리한 것입니다.

지표	뜻	보는 단계
근거 충실도(faithfulness)	답이 가져온 문서에 근거했는가	답 생성
답변 관련성(answer relevance)	답이 질문에 맞는 내용을 말했는가	답 생성

지표	뜻	보는 단계
문맥 관련성(context relevance)	검색한 조각이 질문에 필요한 내용 위주인가	검색
정답률@k(보조 지표)	정답 문서가 상위 k개 안에 들었는가	검색

(출처: 앞의 세 지표는 RAGAS[S12], 정답률@k는 4장·5장 예시 실험과 같은 방식)

정기 평가와 실패 사례 수집

평가는 도입 전에 한 번 하고 끝내면 의미가 줄어드는데, 문서가 추가되고 질문 유형이 바뀌면 품질도 함께 바뀌기 때문입니다. 따라서 업무 담당자가 확인한 질문과 정답 문서 묶음을 평가 세트로 만들어 두고, 임베딩 모델이나 검색 설정을 바꿀 때마다, 그리고 정해 둔 주기마다 같은 세트로 다시 재는 절차가 필요합니다. 운영 화면에서는 사용자가 「도움 안 됨」을 누른 답과 「근거를 찾지 못했습니다」로 끝난 질문을 모아 6.1의 일곱 유형으로 분류하고, 그 결과를 평가 세트에 더합니다. 다만 RAGAS 같은 자동 지표도 결국 LLM이 판단한 값이므로, 결재·계약처럼 중요한 업무는 표본을 뽑아 사람이 함께 확인해야 합니다.

품질 보고 항목 합의

도입 뒤 품질을 무엇으로 보고받을지는 구축 전에 의사결정권자와 IT 담당자가 정해 두어야 합니다. 답변 정확도 하나만 보고받으면, 품질이 나아졌을 때 그 이유가 검색 조정인지 문서 보강인지 알 수 없습니다. 보고 항목은 근거 충실도, 정답률@k, 「찾지 못함」 응답 비율, 실패 유형별 건수처럼 단계별로 나누는 편이 좋습니다. 보고 항목을 미리 합의해 두면 파일럿이 끝났을 때 확산 여부를 처음 정한 기준 그대로 판단할 수 있고, 6.2의 보완 기술 가운데 무엇을 다음에 더할지도 같은 숫자로 고를 수 있습니다.

7장: 챗봇을 넘어 업무 처리의 근거가 되는 RAG

6장의 보완 기술까지 갖춘 RAG는 보통 사내 질문에 답하는 챗봇 형태로 처음 도입됩니다. 공개된 도입 사례를 보면 효과가 크게 나타난 업무는 질문 창 바깥에 있는 경우가 많습니다. 규제 기관의 질의에 회신서를 쓰고, 민원에 답변 초안을 만들고, 결산 전표를 준비하는 처리 단계에서 RAG가 기준 문서를 찾아 근거로 제시합니다. 예를 들어 국민권익위원회는 2026년 2월부터 국민신문고에서 법령과 기존 민원 사례를 찾아 민원 답변 초안을 만드는 서비스를 시작했고[S30], Rivian은 AI 에이전트가 재무 절차서를 검색해 판단하게 하는 방식으로 월말 결산 작업을 바꾸었습니다[S18]. RAG를 챗봇 하나에 대한 투자로 보기보다 여러 업무 처리 단계가 함께 쓰는 지식 기반으로 보는 편이 도입 효과를 더 정확하게 설명합니다. 파일럿 다음에 넓힐 업무를 고를 때 비교할 쓰임새 네 가지와, 업무 시스템에 연결할 때 지킬 원칙 두 가지를 차례로 정리했습니다.

쓰임새는 RAG가 만든 결과를 누가 어떻게 쓰는지에 따라 네 가지로 나눌 수 있습니다. 챗봇형은 사람이 답을 읽고 판단하며, 작성 지원형과 점검형은 사람이 고칠 초안이나 점검 결과를 받습니다. 반면 에이전트형은 여러 단계 업무를 처리하는 프로그램이 단계마다 RAG를 불러 근거를 찾으므로, 결과가 업무 시스템에 반영되기 전에 승인 단계를 두어야 합니다.

쓰임새	입력	RAG가 하는 일	결과물	최종 확인	공개 사례
챗봇형	직원·고객의 질문	관련 문서 조각을 찾아 출처와 함께 답	답변과 출처 목록	질문한 사람	Uber Genie[S21]
작성 지원형	회신·보고서 작성 요청	규정·과거 회신을 찾아 근거 조각과 초안 작성	근거가 표시된 초안	작성 담당자	Amazon Finance 규제 질의[S19]
점검형	새 계약서·신청서	기준 규정과 조항별로 대조	누락·차이 목록	검토 담당자	공개 정량 사례 없음
에이전트형	업무 이벤트·처리 요청	단계마다 절차서와 처리 기록 검색	보류 상태의 처리 결과	결재권자	Rivian 재무 에이전트[S18]

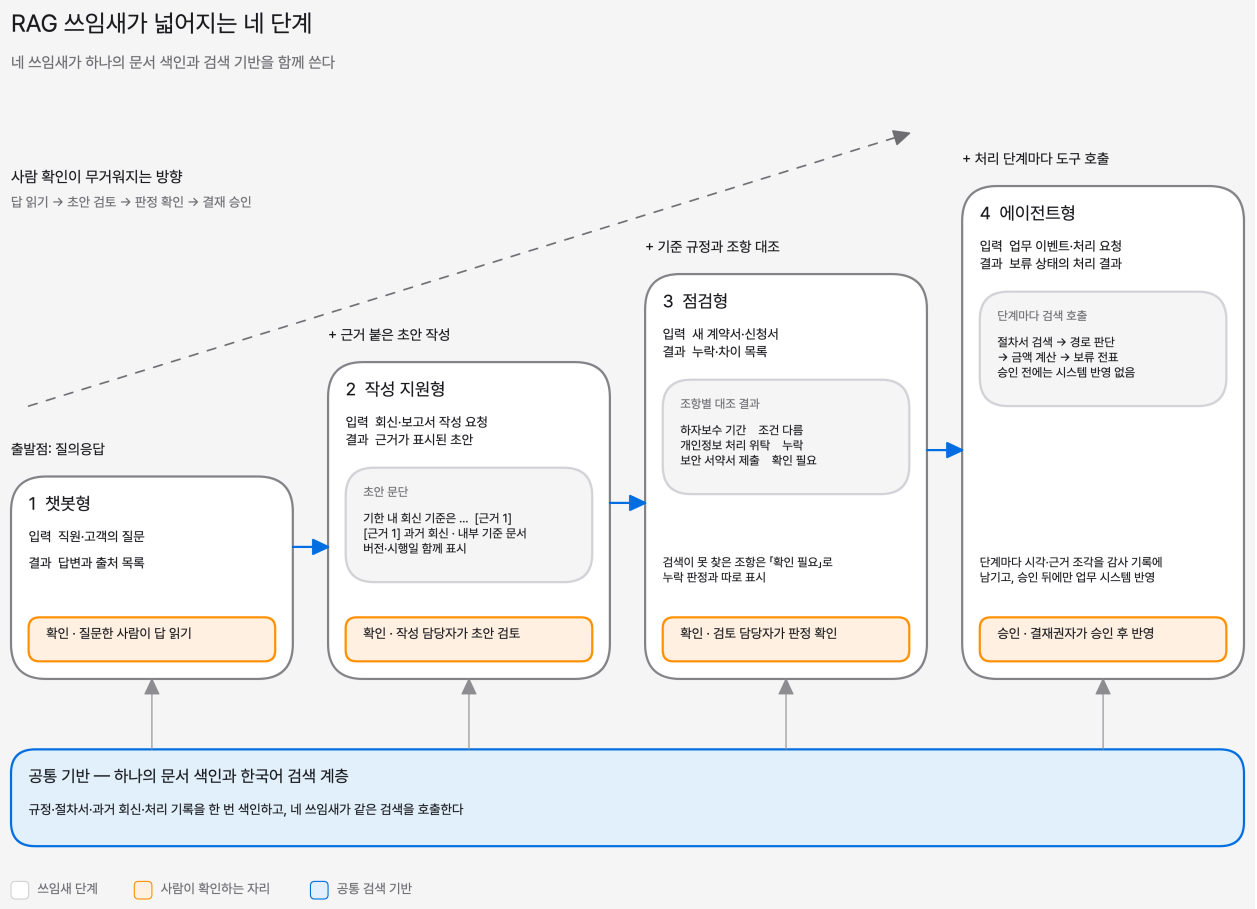

그림. 질의응답에서 작성 지원, 대조 점검, 에이전트 연계로 넓어지는 RAG 쓰임새 단계

7.1 문서 작성과 대조 점검 단계에 근거를 붙이는 RAG

문서 작성과 대조 점검은 조직에서 문서 찾기 비용이 가장 많이 드는 처리 단계입니다. 회신서 한 건을 쓰려면 관련 규정, 과거 회신, 업무 매뉴얼을 모두 찾아 읽어야 하고, 계약서 한 건을 검토하려면 구매 규정의 조항을 하나씩 맞춰 보아야 합니다. 질문 하나에 답하는 경우보다 한 건을 처리하는 데 찾아야 할 문서가 훨씬 많기 때문에, 근거 검색을 처리 단계 안에 넣으면 줄어드는 시간도 그만큼 커집니다. 아래 표는 공개 사례와 연구 결과로 확인된 적용 예시를 업무 처리 단계별로 모은 것입니다.

업무 처리 단계	찾는 근거 문서	산출물	공개 근거
규제 질의 회신	과거 회신, 내부 기준 문서 수천 건	출처가 인용된 회신 초안	Amazon Finance[S19]
민원 답변	관련 법령, 기존 민원 사례, 업무 매뉴얼	담당자에게 제시하는 답변 초안	국민권익위원회 국민신문고[S30]
법령 근거 확인	법령·행정규칙·판례 약 24만 건, 자치법규 약 5만 건	조문이 연결된 답변	AI 법령 비서[S29]
기록 분석 요약	상담·민원 기록 전체	주제별 요약 보고서	GraphRAG 연구[S9]
계약서·신청서 점검	구매 규정, 표준 계약서, 신청 요건	누락·차이 목록	적용 유형(정량 사례 없음)

기록 분석 요약은 개별 질문이 아니라 문서 전체를 묶어 보는 질문이므로 일반 RAG보다 그래프 기반 RAG가 더 잘 맞습니다. 그래프 기반 요약이 전체 요약형 질문에서 보인 결과와 그 측정 조건은 6장 6.2.1절에서 정리했습니다[S9]. 작성 지원 단계를 넓힐 때는 회신·답변처럼 근거 몇 건으로 쓰는 문서와, 분석 보고서처럼 기록 전체를 읽어야 하는 문서를 나누어 검색 구성을 정해야 합니다.

7.1.1 보고서와 답변서 초안에 근거 문서 조각을 붙이는 작성 지원

Amazon Finance 규제 질의 회신의 근거 검색 구성

Amazon의 재무 기술 조직은 여러 관할 당국에서 오는 규제 질의에 정해진 기한 안에 회신해야 합니다. 질의마다 요구 사항과 문서 형식이 다르고, 회신을 쓰려면 PDF·발표 자료·워드 문서·CSV로 흩어진 과거 문서 수천 건에서 선례와 근거 자료를 찾아야 했습니다. 그래서 팀마다 자기 문서로 지식 기반을 따로 만들고, 담당자가 질문하면 약어가 많은 질문을 최대 다섯 가지 변형 질의로 넓혀 동시에 검색하도록 구성했습니다. 이 병렬 검색으로 검색 지연은 순차 처리 때의 10초에서 2초 미만으로 줄었고, 답변에는 근거가 된 원문 문서가 인용 표시로 함께 나옵니다. 회신은 한 번에 끝나지 않고 담당자가 여러 차례 질문을 이어 가며 다듬으므로, 대화 이력을 세션 단위로 보관해 앞선 문맥을 이어 씁니다[S19].

국민신문고 민원 답변 초안의 공공 적용

국내 공공 부문에서도 같은 방식이 민원 처리 단계에 들어갔습니다. 국민권익위원회는 국민신문고에 생성형 AI 서비스 세 가지를 시범 도입했고, 그중 식품의약품안전처에 적용한 서비스는 관련 법령, 기존 민원 사례, 업무 매뉴얼을 분석해 답변 초안을 만들어 담당자에게 제시합니다. 같은 시기에 도입한 빈발·중복 민원 일괄 처리 서비스는 담당 공무원이 답변 내용을 검토할 시간을 확보하는 것을 목적으로 밝혔습니다. 서비스는 데이터 품질을 보완한 뒤 2026년 2월 5일부터 시작됐습니다[S30]. 민원 답변처럼 대외로 나가는 문서에서도 초안 작성과 근거 검색은 RAG가 맡고, 답변의 책임은 담당자에게 남는 구조가 공공 기관에서 받아들여지고 있습니다.

초안 작성자와 RAG의 역할 구분

작성 지원형 RAG에서 문서의 작성자는 여전히 사람입니다. RAG는 요청에 맞는 규정 조각과 과거 회신을 찾아 초안 문단 옆에 근거로 표시하고, 담당자는 그 근거를 열어 보고 문장을 고치거나 지웁니다. 의사결정권자가 눈여겨볼 변화는 작성 시간 단축보다 초안의 품질 쪽에 있습니다. 근거 없이 쓴 초안은 검토자가 처음부터 규정을 다시 찾아야 하지만, 문장마다 출처가 표시된 초안은 검토자가 그 출처가 맞는지만 확인하면 됩니다. 다만 출처가 표시되었다고 해서 내용이 맞다는 보장은 없으므로, 근거 문서의 버전과 시행일을 초안에 함께 보여 주어야 낡은 규정이 인용되는 경우를 검토 단계에서 걸러낼 수 있습니다.

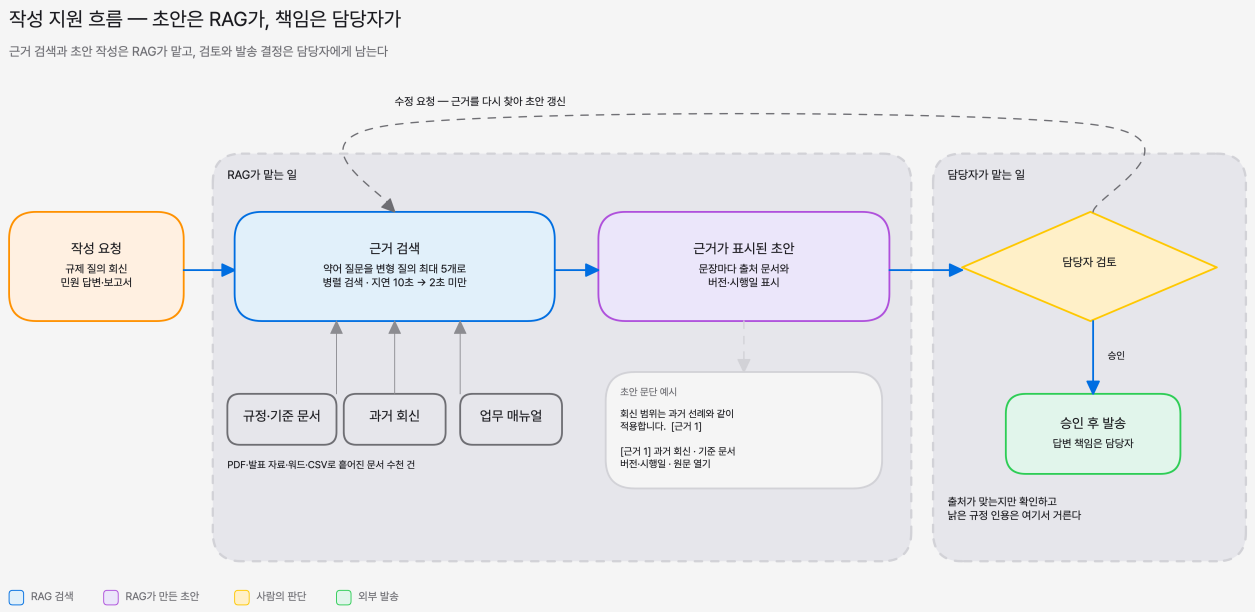

그림. 작성 요청에서 근거 검색, 초안 작성, 담당자 검토로 이어지는 작성 지원 흐름

7.1.2 계약서·규정 대조로 빠진 항목을 찾는 점검 업무

계약서와 신청서 대조 점검의 처리 방식

점검형 RAG는 새로 들어온 계약서나 신청서를 기준 규정과 조항 단위로 맞춰 보고, 빠진 항목과 조건이 다른 항목을 표시합니다. 처리 순서는 대상 문서를 조항으로 나누고, 조항마다 기준 문서에서 대응하는 조각을 검색한 뒤, 두 조각을 나란히 놓고 차이를 판정하는 방식입니다. 이 쓰임새는 공개된 정량 사례가 아직 드물기 때문에 효과 크기를 수치로 말하기는 어렵습니다. 하지만 점검 업무는 한 항목을 놓쳤을 때 치르는 비용이 크기 때문에, 판정마다 기준 문서의 어느 조각과 비교했는지를 보여 주는 RAG의 근거 표시가 특히 쓸모 있습니다. 아래 표는 구매 계약서 한 건을 점검한 결과를 어떤 형태로 돌려주는지 보여 주는 예시입니다.

점검 항목	기준 문서 조각	대상 문서 내용	판정
하자보수 기간	구매 규정 하자보수 조항: 검수 후 2년	계약서 제9조: 검수 후 1년	조건 다름
개인정보 처리 위탁	표준 계약서 위탁 조항: 위탁 범위와 재위탁 제한 명시	해당 조항 없음	누락
검수 기한	구매 규정 검수 조항: 납품 후 14일 이내	계약서 제6조: 납품 후 14일 이내	일치
보안 서약서 제출	정보보안 지침 외부 인력 조항	대응 조각을 찾지 못함	확인 필요

점검 결과에서 누락과 검색 실패의 구분

점검 결과를 설계할 때는 「기준에 없음」과 「검색이 찾지 못함」을 다른 판정으로 나누어야 합니다. RAG의 일곱 가지 실패 유형 가운데에는 답이 문서에 있는데도 검색 상위 후보에 들지 못하는 경우와, 후보에는 들었지만 답변 생성 단계에서 빠지는 경우가 포함되어 있습니다[S7]. 표의 마지막 행처럼 대응 조각을 찾지 못한 항목을 「일치」

나 「누락」으로 처리하면 검색 실패가 점검 결과로 둔갑합니다. 이런 항목은 「확인 필요」로 따로 표시하고, 검토 담당자가 기준 문서를 직접 열어 보도록 처리 흐름을 정해 두는 편이 안전합니다.

검토 담당자의 최종 확인 원칙

점검형 RAG의 판정은 검토 담당자가 최종 확인할 대상입니다. 조건이 다르다는 판정이 나와도 협상 결과로 예외를 허용한 계약일 수 있고, 일치 판정이 나와도 같은 조항의 다른 문장에 단서가 붙어 있을 수 있기 때문입니다. 그래서 점검 결과 화면에는 판정만 두지 않고 기준 조각과 대상 조각을 함께 보여 주며, 담당자가 판정을 고친 내역을 남겨 다음 점검의 평가 자료로 씁니다. 쌓인 수정 내역은 8장 8.2.1절에서 다루는 평가 세트의 재료가 되므로, 점검 업무는 파일럿 이후 검색 품질을 계속 재는 데에도 쓸모가 있습니다.

7.2 에이전트가 판단 근거를 찾는 도구로서의 RAG

AI 에이전트는 업무 이벤트를 받아 여러 단계를 차례로 처리하면서, 단계마다 필요한 도구를 호출하는 프로그램입니다. RAG는 에이전트가 판단하기 전에 절차서·과거 처리 기록·정책 문서를 찾아보는 도구 역할을 합니다. 에이전트가 어떤 경로를 고르고 어떤 값을 계산할지는 검색으로 가져온 근거에 따라 정해지므로, 지금 구축하는 문서 색인과 검색 구성이 이후 에이전트 도입의 기반이 됩니다. 공개된 사례에서 RAG를 도구나 근거 검색 단계로 쓴 결과는 다음과 같습니다.

사례	업무	RAG가 맡은 역할	공개된 결과
Rivian 재무 에이전트 [S18]	구매 주문 발생 비용 회계 처리	절차서 검색 후 처리 경로 판단	개념 증명 5주 뒤 운영 전환, 결산 수작업 제거(수치는 3.2.1)
Uber 에이전트형 RAG[S22]	보안·개인정보 정책 질의	질문 분해, 출처 문서 선별, 검색 결합	잘못된 조언 60% 감소, 허용 가능한 답변 27% 증가
Uber Genie[S21]	사내 온콜 지원 질의	사내 문서 검색 후 답변	엔지니어링 시간 절감(추정, 수치는 3.3.1)
LinkedIn 고객 서비스 [S24]	상담 티켓 처리	과거 티켓을 지식 그래프로 묶어 검색	해결 시간 단축(수치는 3.3.2)
RepoCoder[S23]	코드 작성 지원	코드 저장소를 반복 검색해 생성에 반영	코드 완성 품질 향상(수치는 3.3.1)

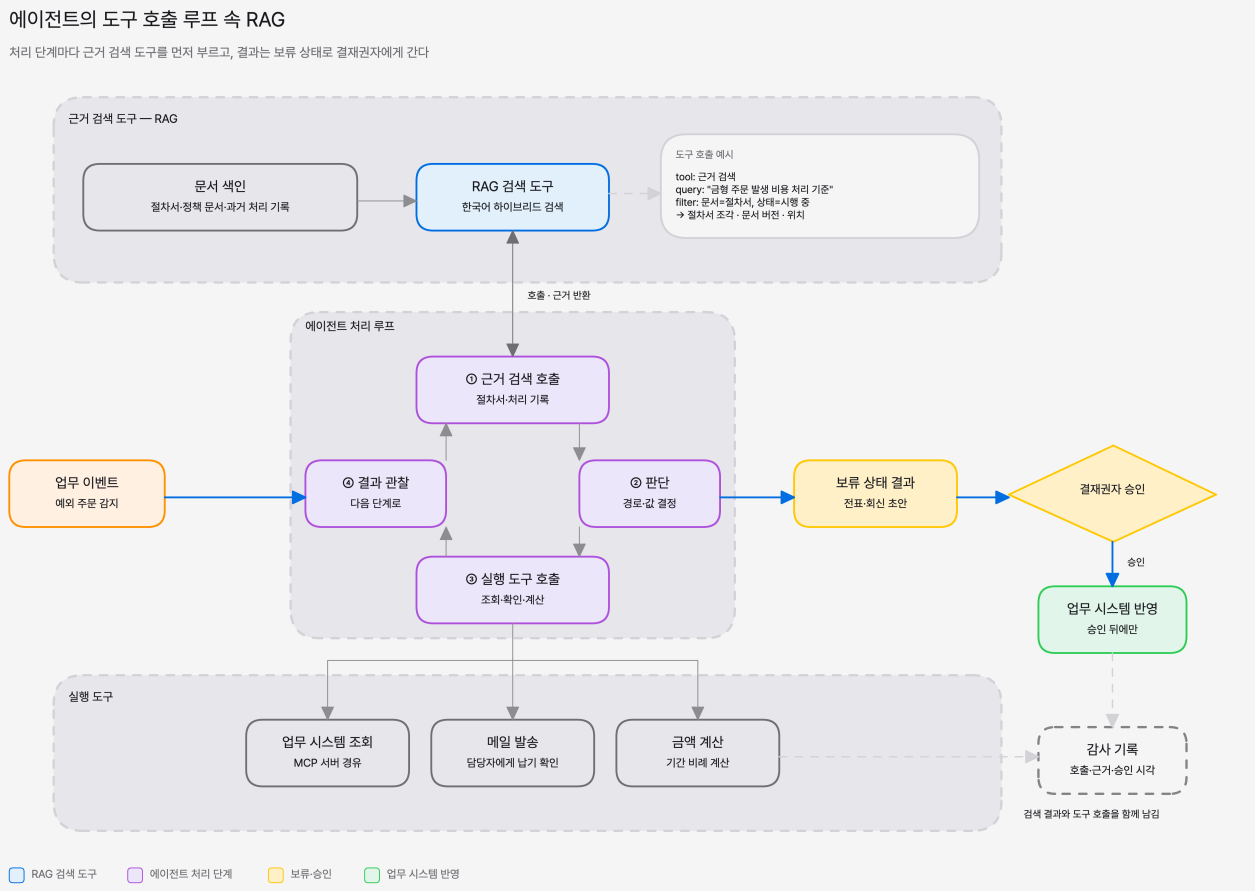

그림. 에이전트가 처리 단계마다 RAG 검색 도구를 호출하는 구성

RAG와 에이전트는 설계의 출발점이 다릅니다. 아래 그림처럼 RAG는 질문과 문서 데이터에 맞춰 구성하고, 에이전트는 사람이 일하는 순서에 맞춰 구성합니다.



그림. RAG의 구성 요소가 에이전트의 처리 단계로 옮겨 가는 대응 관계

오른쪽 대응표에서 보듯 질문에 맞는 문서군을 고르는 일과 질문을 바꿔 다시 검색하는 일은 에이전트에서 계획과 도구 사용 단계가 됩니다. 따라서 지금 만드는 문서 색인과 검색 구성은 에이전트 단계에서 버려지지 않고 도

구 하나로 그대로 쓰입니다. 다만 에이전트 단계에서는 검색 결과가 곧바로 업무 처리의 입력이 되므로, 7.3절의 감사 기록과 승인 원칙을 검색 단계부터 함께 설계해야 합니다.

7.2.1 월말 결산 수작업을 줄인 Rivian 재무 에이전트의 근거 검색

Rivian 금형 구매 주문 회계 처리의 업무 조건

Rivian이 처음 자동화한 업무는 맞춤 금형 구매 주문의 발생 비용 회계 처리였습니다. 프레스 금형이나 사출 금형은 개발에 12개월에서 24개월까지 걸리고 청구서는 주문 후 18개월이 넘어서야 오는 경우가 많은데, 미국 회계 기준(GAAP)은 비용을 청구서 도착 시점이 아니라 개발 기간에 걸쳐 나누어 인식하도록 요구합니다. 재무 담당자는 이 기준에 맞추려고 수백 건의 주문을 동시에 추적하며 납기를 확인하고, 기간 비례로 금액을 계산해 ERP에 반영하고, 외부 감사인을 위해 처리 기록을 남겨야 했습니다. 금액 기준, 주문 특성, 납기에 따라 갈래가 달라지는 이 규칙은 코드로 옮기면 조건문이 수천 줄로 늘어나고 규칙이 바뀔 때마다 개발자가 고쳐야 하는 구조였습니다[S18].

절차서를 검색해 판단하는 에이전트 구성

Rivian은 처리 절차를 코드에 넣는 대신 상세한 표준 운영 절차서(SOP, Standard Operating Procedure)로 문서화하고 지식 기반에 저장했습니다. 에이전트는 ERP 이벤트로 예외 주문을 감지하면 상태 저장소와 지식 기반에서 맥락을 모으고, 주문 특성에 맞는 절차서 조각을 검색한 뒤 금액 기준에 따라 검증 경로를 고릅니다. 그 뒤 MCP(Model Context Protocol, 모델 도구 연결 규약) 서버로 ERP를 조회하고 주문 담당자에게 메일로 납기를 확인한 다음, 정액 방식으로 금액을 계산해 보류 전표를 만듭니다. 전표는 재무 관리자가 승인한 뒤에만 ERP에 전기되며, 처리 단계마다 시각이 찍힌 기록이 남습니다[S18]. 그래서 재무 관리자는 코드 변경을 요청하지 않고 절차서 문서를 고쳐 업무 규칙을 바꿀 수 있습니다.

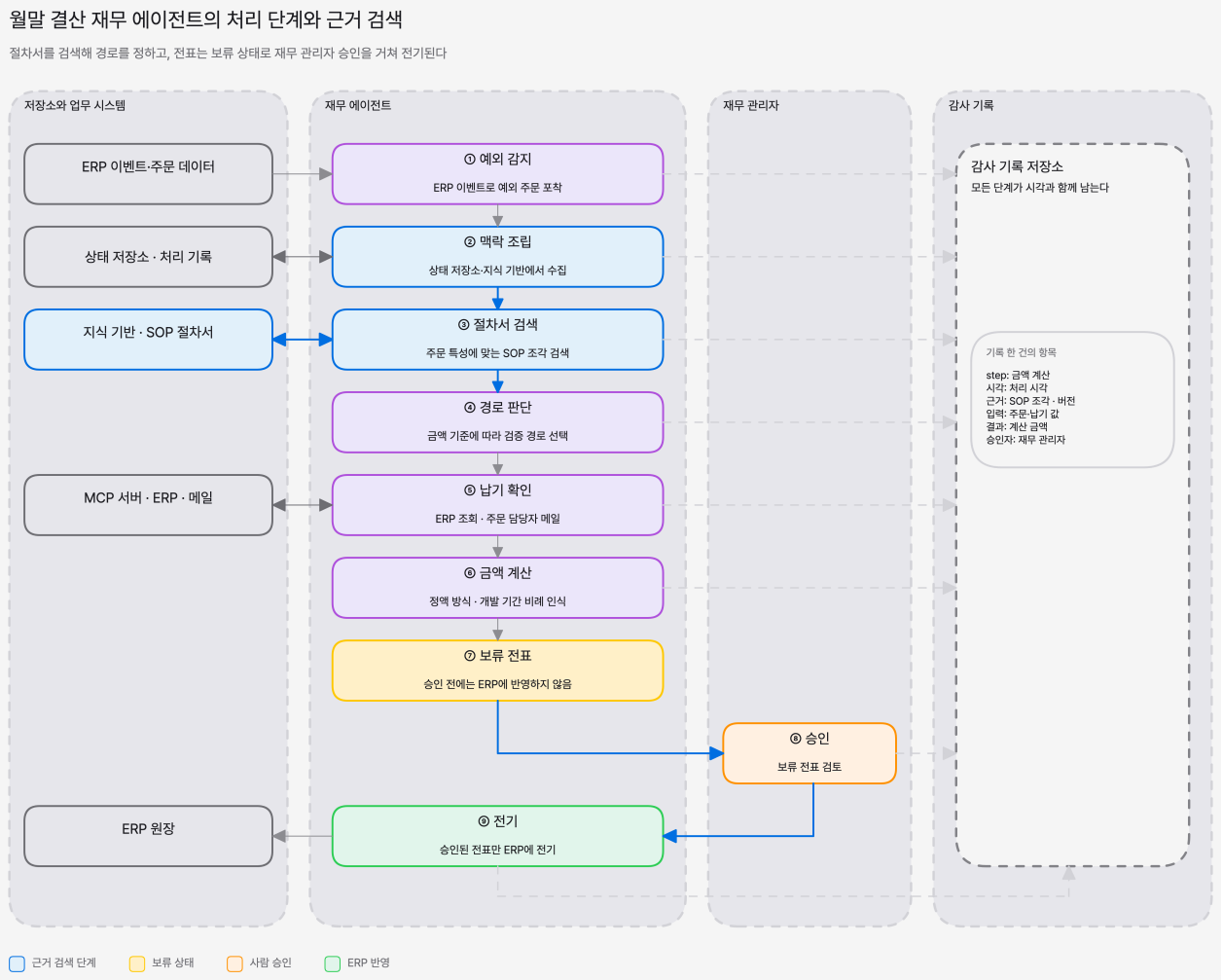

그림. 재무 에이전트가 결산 처리 단계마다 절차서와 처리 기록을 검색하는 구성

에이전트 판단 품질과 검색 품질의 관계

이 구성에서 에이전트의 판단 품질은 결국 절차서 검색의 품질에 달려 있습니다. 금액 기준이 다른 절차서 조각이 걸리면 검증 경로가 바뀌고, 그 뒤의 계산과 전표가 모두 틀린 근거 위에서 만들어지기 때문입니다. 반면 맞는 조각이 걸리면 같은 절차서가 주문마다 일관되게 적용되므로, Rivian은 정확도와 내부 통제가 함께 강화되었고 외부 감사인이 스프레드시트 계산 때보다 처리 과정의 투명성을 높게 평가했다고 밝혔습니다[S18]. 에이전트 도입을 검토하는 조직은 에이전트 기능보다 먼저 절차서의 한국어 검색 품질을 4장과 5장의 기준으로 재 보아야 합니다.

7.2.2 잘못된 조언을 60% 줄인 에이전트형 RAG 의 검색 계획

Uber 정책 질의 코파일럿의 단순 RAG 한계

Uber는 사내 온콜 지원 코파일럿 Genie로 엔지니어의 보안·개인정보 정책 질문에 사내 문서를 인용해 답했습니다. 정책 문서 40개 이상을 넣고 분야 전문가가 만든 100개 이상의 평가 질의로 점검해 보니, 단순 RAG의 답은 자주 불완전하거나 부정확했습니다. 원인은 두 가지였습니다. 첫째, PDF를 읽는 과정에서 표 구조가 깨져 셀의 행·열 맥락이 사라졌고, 문서 조각을 나눌 때 표가 잘못 잘렸습니다. 둘째, 모호한 질문과 정책 사이의 미묘한 차이 때문에 관련 없는 문맥이 검색되어 환각으로 이어졌습니다[S22].

질문을 나누고 출처를 고르는 검색 계획

에이전트형 RAG는 검색 전에 질문을 다루는 단계를 더했습니다. 질의 최적화 단계가 모호한 질문을 명확하게 고치고 복합 질문을 단순한 질문 여러 개로 나누면, 출처 식별 단계가 그 질문에 해당할 가능성이 높은 정책 문서로 범위를 좁힙니다. 이렇게 좁힌 범위에서 BM25 키워드 검색과 벡터 검색의 결과를 합치고, 후처리 단계가 중복 조각을 없앤 뒤 원문 순서대로 정렬해 답변 생성에 넘깁니다. 평가에서는 분야 전문가가 기준 답변을 한 번 작성해 두고, LLM 심사가 답변마다 0점에서 5점까지 점수와 이유를 매기게 해 평가 시간을 몇 주에서 몇 분으로 줄였습니다. 그 결과 잘못된 조언은 상대 기준으로 60% 줄었고 허용 가능한 답변은 27% 늘었습니다[S22].

비교 항목	단순 RAG	에이전트형 RAG
질문 처리	원 질문을 그대로 검색	모호한 질문 명확화, 복합 질문 분해
검색 범위	전체 문서	해당 가능성이 높은 정책 문서로 축소
검색 방식	벡터 검색	BM25 키워드 검색과 벡터 검색 결합
표 문서 처리	PDF에서 표 구조 손실	표를 구조 보존 형식으로 변환, 조각 단위 유지
결과(상대 변화)	기준	잘못된 조언 60% 감소, 허용 가능한 답변 27% 증가

표 출처: Uber Engineering[S22]

단순 RAG 다음 단계로 보는 에이전트형 RAG 로드맵

이 사례에서 개선 폭을 만든 구성 요소는 대부분 검색 단계에 있습니다. 질문 분해와 출처 선별은 에이전트가 검색을 계획하는 기능이고, 키워드 검색과 벡터 검색의 결합은 5장의 하이브리드 검색과 같은 구성입니다. 따라서 파일럿에서 한국어 하이브리드 검색과 평가 세트를 먼저 갖춰 두면, 다음 단계에서 질문 분해와 출처 선별을 더해 에이전트형 RAG로 넓히는 순서를 로드맵에 올릴 수 있습니다. 에이전트와 챗봇이 업무 처리 방식에서 어떻게 다른지는 함께 읽을 자료 「AI 에이전트와 챗봇의 차이」[R1]에 정리되어 있습니다.

7.3 업무 시스템에 RAG 를 연결할 때 지킬 감사 기록과 승인 원칙

업무 시스템과 이어진 RAG는 답이 틀렸을 때 그 결과가 회신서, 민원 답변, 전표 같은 업무 기록으로 남습니다. RAG를 업무 처리 단계에 넣는 조직은 답의 출처를 되짚을 수 있는 감사 기록과, 사람이 최종 확인하는 승인 단계를 도입 전에 설계 원칙으로 정해야 합니다. RAG의 견고성은 처음부터 설계로 완성되지 않고 운영하면서 다듬어지므로[S7], 무엇을 검색했고 무엇으로 답했는지 남긴 기록이 있어야 운영 중에 품질을 고칠 수 있습니다. 감사 기록은 사후 책임 추적과 품질 개선에 함께 쓰이고, 승인 단계는 업무 책임의 경계를 정합니다.

7.3.1 답마다 출처 문서와 검색 경로를 남기는 감사 기록

감사 기록에 남길 항목

감사 기록은 질문 한 건이 답이나 초안으로 바뀌기까지의 경로를 나중에 그대로 다시 볼 수 있게 남기는 기록입니다. Amazon Finance는 질의 확장, 지식 검색, 모델 입력과 응답, 지연 시간, 인용된 원문 문서를 요청마다 추적 기록으로 남기고, 대화 기록을 변경할 수 없는 감사 기록으로 보관해 준법 검토에 씁니다[S19]. Rivian도 처리 단계마다 시각이 찍힌 기록을 보관해 내부 통제 요건을 충족했습니다[S18]. 아래 표는 두 사례에 공통된 항목을 IT 담당자가 요건서에 바로 옮길 수 있게 정리한 것입니다.

기록 항목	남길 내용	쓰임
요청 정보	요청 시각, 요청자, 요청자의 문서 접근 권한 범위	권한 밖 문서 인용 여부 확인
질문	원 질문, 확장·분해된 변형 질의	검색 실패 원인 분석
검색 결과	후보 문서 ID, 문서 버전, 순위, 점수	검색 품질 평가, 낡은 문서 검색 여부 확인
인용 문서	답에 실제로 쓴 문서 조각과 위치	답의 근거 재확인
답·초안	생성된 답변 또는 초안 원문	사람 수정 전후 비교
생성 조건	모델 버전, 프롬프트 버전, 검색 구성 버전	품질 변화 시점 추적
사람 검토	승인·수정·반려 결과, 확인자, 확인 시각	업무 책임 경계 확인
업무 반영	반영한 업무 시스템, 반영 건 번호	전표·회신과 답의 연결

문서 버전 기록과 낡은 규정 인용의 추적

감사 기록에서 가장 자주 빠지는 항목은 인용 문서의 버전입니다. 규정이 개정된 뒤에도 개정 전 규정이 색인에 남아 있으면 검색은 정상으로 보이는데 답은 폐지된 기준을 인용하게 됩니다. Amazon Finance도 원문에 없는 내용을 지어내는 경우와 낡은 준법 지침이 검색되는 경우를 운영 중에 잡아내야 할 대표적인 문제로 꼽았습니다[S19]. 따라서 문서 ID와 함께 버전과 시행일을 기록해 두면, 민원이나 감사에서 특정 답의 근거를 물을 때 그 시점에 어느 판의 규정을 인용했는지 바로 답할 수 있습니다.

찾지 못한 범위까지 남기는 답변 기록

답을 찾지 못한 경우의 응답도 기록할 가치가 있습니다. 「해당 기간 보고서에는 관련 문서가 없고, 날짜별로 다시 검색해도 결과가 없었다」처럼 어디를 뒤졌는지 밝힌 응답은 결론만 남긴 응답과 달리 나중에 다시 검증할 수 있습니다. 검색 범위와 조건이 함께 남아 있으면 몇 달 뒤 같은 조건으로 다시 검색해 결론이 그대로인지 확인하고, 달라졌다면 어느 조건이 바뀌었는지 짚을 수 있습니다. 반면 범위 없이 결론만 남은 기록은 참고 자료로만 쓰입니다.

한 사람이 받은 답이 대화 화면에서 끝나지 않고 검색 가능한 기록으로 남으면 쓰임이 하나 더 생깁니다. 같은 문제를 만난 다른 담당자가 누가 언제 물었는지 몰라도 그 분석을 찾아 쓸 수 있습니다. 이 점에서 RAG 답변 기록은 감사 자료이면서 조직의 업무 기록이 됩니다.

7.3.2 사람이 최종 확인하는 승인 단계와 업무 책임 경계

공개 사례에 나타난 사람 승인 구조

RAG가 만든 초안이나 판단은 담당자가 확인하고 승인해야 업무 처리로 이어집니다. Rivian의 재무 에이전트는 자율적으로 단계를 처리하지만, 전표가 ERP에 전기되기 전에는 반드시 재무 관리자가 최종 승인합니다. 이 승인 단계는 내부 통제와 외부 감사 요건을 함께 충족하고, 관리자가 에이전트 결과를 검토하며 절차서를 보완할 예외 사례를 찾아내는 개선 경로도 됩니다[S18]. 국민신문고의 민원 답변 초안도 서비스가 직접 답하지 않고 민원 담

당자에게 초안으로 제시됩니다[S30]. 공개된 사례는 모두 RAG가 근거 검색과 초안 작성을 맡고, 업무 기록으로 남는 최종 처리는 담당자의 승인으로 확정되는 구조를 따릅니다.

업무 영향도별 승인 단계의 배치 기준

승인 단계의 위치는 처리 결과가 업무에 미치는 영향에 따라 정합니다. 영향이 작은 사내 안내 답변은 사후 표본 점검으로도 충분하지만, 대외로 나가는 문서나 업무 시스템의 데이터를 바꾸는 처리는 반영 전에 승인을 받도록 앞에 배치해야 합니다. 에이전트가 쓰는 검색도 요청자의 권한 안에서만 문서를 찾도록 권한 인식 저장소와 데이터 분할을 적용해야, 승인자가 권한 밖 문서를 근거로 한 결과를 모르고 승인하는 경우를 막을 수 있습니다[S10].

업무 영향도	처리 예시	승인 단계 위치	확인자
낮음	사내 규정 안내 질의응답	사후 표본 점검	운영 담당자
중간	내부 보고서 초안, 문서 분류	사용 전 작성자 확인	작성 담당자
높음	대외 회신, 민원 답변, 계약서 점검	발송·확정 전 승인	업무 담당자와 부서장
매우 높음	전표 전기, 업무 시스템 데이터 변경	보류 상태로 생성, 반영 전 결재	결재권자

결재 문서에 함께 정할 자동화 범위와 책임자

승인 구조는 도입을 결재할 때 자동화 범위와 함께 정해야 합니다. 어느 처리 단계까지를 RAG와 에이전트가 맡고 어느 단계부터 사람이 확인하는지, 그리고 각 단계의 확인자가 누구인지를 결재 문서에 적어 두면, 결과에 문제가 생겼을 때 책임 경계를 두고 다툴 일이 줄어듭니다. 예를 들어 위 표의 「높음」 단계에 속한 민원 답변을 파일럿 대상으로 정했다면, 초안 작성은 RAG가 맡고 발송 승인은 담당 부서장이 맡는다는 문장을 결재 문서에 그대로 넣을 수 있습니다. 결재 문서에 정한 범위와 확인자는 8장의 파일럿 성공 기준과 확산 결정 책임자를 정하는 근거가 됩니다.

8장: 파일럿 업무 선정부터 확산까지의 도입 순서와 점검표

한국어 RAG 도입은 문서 찾기 비용이 큰 업무 하나를 골라 파일럿 대상으로 정하는 일에서 시작합니다. 전사 문서를 한꺼번에 색인하면 어떤 업무에서 효과가 났는지 가려낼 수 없고, 검색 구성 가운데 무엇이 성능을 좌우했는지도 확인하기 어렵기 때문입니다. 따라서 이 장에서는 3장의 업무별 사례와 4–6장의 기술 조건을 한데 모아 파일럿 업무 선정 기준, 평가 세트에서 출발하는 네 단계 도입 순서, 결재 전에 확인할 점검표를 차례로 정리합니다. 점검표의 마지막 양식을 채우면 파일럿 대상 업무 1개와 그 성공 기준이 정해지도록 구성했습니다.

8.1 문서 찾기 비용이 큰 업무를 고르는 파일럿 선정 기준

파일럿 업무는 효과와 난이도라는 두 기준으로 고릅니다. 효과는 그 업무에서 같은 질문이 얼마나 반복되고 문서를 찾는 데 시간이 얼마나 드는지로 재고, 난이도는 대상 문서의 양·형식·접근 권한이 얼마나 복잡한지로 잽니다. 첫 파일럿으로 알맞은 업무는 효과가 크면서 난이도가 낮은 업무이고, 효과는 크지만 난이도가 높은 업무는 두 번째 파일럿 후보로 따로 적어 둡니다.

8.1.1 반복 질문 수와 문서 찾기 시간으로 보는 파일럿 효과 크기

효과 크기는 현업 담당자가 바로 답할 수 있는 세 가지 질문으로 매깁니다. 같은 질문이 한 달에 몇 번 반복되는지, 문서 하나를 찾아 답을 확인하는 데 몇 분이 걸리는지, 답이 늦거나 틀렸을 때 어떤 손실이 생기는지를 묻는 것입니다. 세 질문 모두 담당자가 기억이나 업무 기록으로 대략의 숫자를 낼 수 있어서, 별도 조사 없이 첫 회의에서 후보 업무를 비교할 수 있습니다.

후보 목록으로는 3장에서 살펴본 마케팅·영업·회계·인사·개발·고객지원·구매·공공 업무 카드를 그대로 쓰면 되고, 업무마다 세 질문의 답을 숫자로 적어 비교합니다.

반복 질문이 많은 업무일수록 효과가 크게 나타난다는 점은 공개 사례에서도 확인할 수 있습니다. 예를 들어 3장에서 본 Uber의 사내 질의응답 도우미(3.3.1절), IBM의 인사 문의 도우미(3.2.2절), Morgan Stanley의 어드바이저 문서 검색(3.1.2절)은 모두 반복 질문이나 문서 접근이 많은 업무에 적용됐습니다[S21][S20][S16]. 세 사례가 모두 효과를 문의 건수, 절감 시간, 접근률 같은 숫자로 보고했다는 점이 중요합니다. 효과를 잴 지표가 처음부터 정해져 있어야 파일럿이 끝난 뒤 확산 여부를 근거를 갖고 결정할 수 있기 때문입니다.

다음 점수표는 효과 크기를 5점 척도로 매기는 예시입니다. 경계값은 조직 규모에 따라 달라지므로 첫 회의에서 조정해 씁니다.

효과 질문	1점	3점	5점
같은 질문이 한 달에 몇 번 반복되는가	50건 미만	50–300건	300건 이상
문서를 찾아 답을 확인하는 데 몇 분이 걸리는가	5분 미만	5–30분	30분 이상
답이 늦거나 틀리면 어떤 손실이 생기는가	담당자 불편에 그침	내부 처리 지연	고객 응대 지연, 규정 위반 위험

효과 질문	1점	3점	5점
효과를 잴 지표가 이미 기록되고 있는가	지표 없음	수작업으로 집계 가능	헬프데스크·업무 시스템에 기록됨

점수표의 네 번째 질문은 다른 세 질문과 성격이 조금 다릅니다. 앞의 세 질문이 효과의 크기를 묻는다면, 네 번째 질문은 그 효과를 파일럿이 끝난 뒤 숫자로 보여 줄 수 있는지를 묻기 때문입니다. 그래서 반복 질문이 많더라도 문의 건수나 처리 시간이 어디에도 기록되지 않는 업무라면, 파일럿을 시작하기 전에 몇 주 동안 수작업으로라도 기준값을 모아 두는 편이 좋습니다. 그렇게 해야 도입 뒤 줄어든 문의 건수와 확인 시간을 같은 기준으로 비교할 수 있고, 확산을 결정하는 회의에서도 효과를 추정이 아닌 기록으로 설명할 수 있습니다.

8.1.2 문서 양과 형식, 접근 권한 범위로 보는 파일럿 난이도

난이도는 문서 규모와 갱신 주기, 문서 형식, 권한 구조, 정답 확인 가능성 네 가지로 봅니다. 문서 규모가 작은 업무는 RAG 파일럿 대상으로 알맞지 않을 수 있습니다. 대상 지식이 약 20만 토큰 미만이면 검색 단계 없이 문서 전체를 모델 입력에 넣는 방식이 더 나은 결과를 낸다는 보고가 있고[S6], 이런 업무는 2.3.2에서 설명한 긴 문맥 입력으로 처리하는 편이 간단합니다. 반면 문서가 많고 자주 바뀌는 업무는 색인 갱신 절차를 함께 설계해야 하므로 난이도가 올라갑니다.

문서 형식과 권한 구조는 일정에 가장 크게 영향을 줍니다. 스캔 PDF, 표, 도면, HWP 문서의 비중이 높으면 글자 추출 품질부터 확인해야 하고, 페이지 이미지를 그대로 검색하는 방식[S15]을 쓰면 파일럿 범위가 그만큼 넓어집니다. 부서나 직급에 따라 열람 범위가 나뉜 문서가 많으면 검색 결과에 권한을 적용하는 필터를 먼저 만들어야 합니다. 현업 담당자가 질문마다 정답 문서를 지목할 수 있어야 평가 세트를 만들 수 있으므로, 정답 확인이 어려운 업무는 첫 파일럿에서 뺍니다.

난이도 기준	1점(쉬움)	3점	5점(어려움)
문서 규모와 갱신 주기	수백 건, 분기 단위 갱신	수천 건, 월 단위 갱신	수만 건 이상, 매일 갱신
문서 형식	텍스트 위주의 문서	HWP·표가 섞인 문서	스캔본·도면이 다수
접근 권한 구조	전사 공개	부서 단위로 구분	개인·사건 단위로 구분
정답 확인 가능성	정답 문서가 명확함	담당자 판단이 필요함	정답 문서가 없음

두 점수표의 합계를 나란히 놓고 보면 후보 업무가 네 묶음으로 나뉩니다. 예를 들어 전사에 공개된 인사 규정 문의는 반복 질문이 많고 문서가 텍스트 위주라 첫 파일럿 후보의 전형에 속합니다. 법령과 민원 기록을 다루는 공공 업무는 효과가 크지만 HWP와 스캔본 비중이 높아 두 번째 파일럿 후보로 분류하는 경우가 많습니다.

네 묶음의 처리 방식은 다음과 같이 정리됩니다. 효과가 높고 난이도가 낮은 업무는 첫 파일럿 대상으로 정합니다. 효과는 높지만 난이도가 높은 업무는 두 번째 파일럿 후보로 기록해 두고, 그 업무에 필요한 문서 파싱과 권한 필터 준비를 첫 파일럿과 나란히 먼저 시작합니다. 효과가 낮고 난이도도 낮은 업무는 별도 파일럿 없이 확산 단계에서 같은 검색 구성을 재사용해 추가하면 되고, 효과가 낮은데 난이도가 높은 업무는 보류합니다. 이렇게 묶음별 처리 방식을 미리 정해 두면, 후보 업무가 여러 부서에서 동시에 올라와도 어느 업무를 먼저 할지를 점수로 설명할 수 있어 부서 간 조정이 쉬워집니다.

망 분리 환경의 공공기관이 고를 만한 첫 후보

망 분리가 남아 있는 공공기관이라면 다른 기관에서 받은 HWP·HWPX 공문과 정해진 공공 서식을 다루는 업무가 첫 파일럿 후보가 될 수 있습니다. 공문은 매일 들어와 반복량이 많고, 사람이 확인할 지점이 저장 전 한 번으로 단순하며, 연결해야 할 시스템이 문서 변환 단계 정도로 적습니다. 다만 스캔본 문자 인식과 언어 모델 호출까지 모두 조직 안에서 끝나도록 구성해야 외부 서비스 이용 검토 부담 없이 시작할 수 있습니다. 반복량은 효과를, 사람 확인 단계 수와 연결 시스템 수는 난이도를 가늠하는 항목이므로 8.1.1절과 이 절의 두 점수표에 그대로 옮겨 볼 수 있습니다.

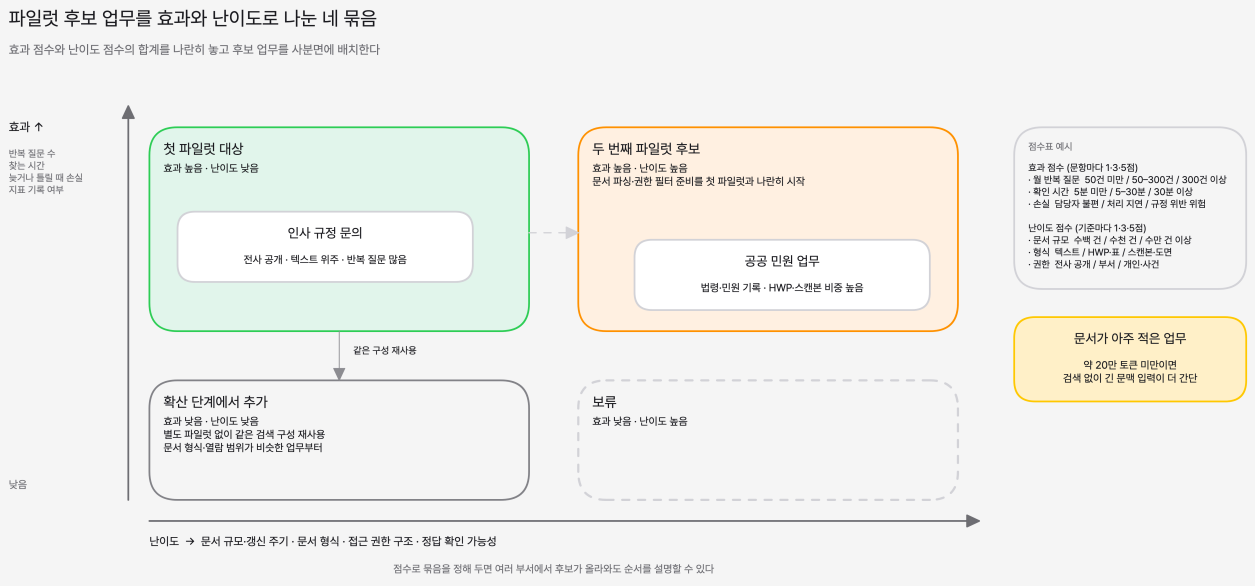

그림. 효과와 난이도 두 가지 기준으로 본 파일럿 후보 업무 배치

8.2 평가 세트에서 시작해 운영 확산까지 넓히는 네 단계 도입 순서

도입은 평가 질의 세트 구축, 파일럿, 확대, 운영의 네 단계로 진행합니다. 단계마다 다음 단계로 넘어가는 완료 기준을 미리 정해 두어야 파일럿이 기한 없이 길어지지 않고, 의사결정권자는 각 단계에서 어떤 숫자를 보고받을지 알 수 있습니다. 그래서 아래 표의 완료 기준은 파일럿 결재 문서에 함께 적어 두는 것을 권합니다.

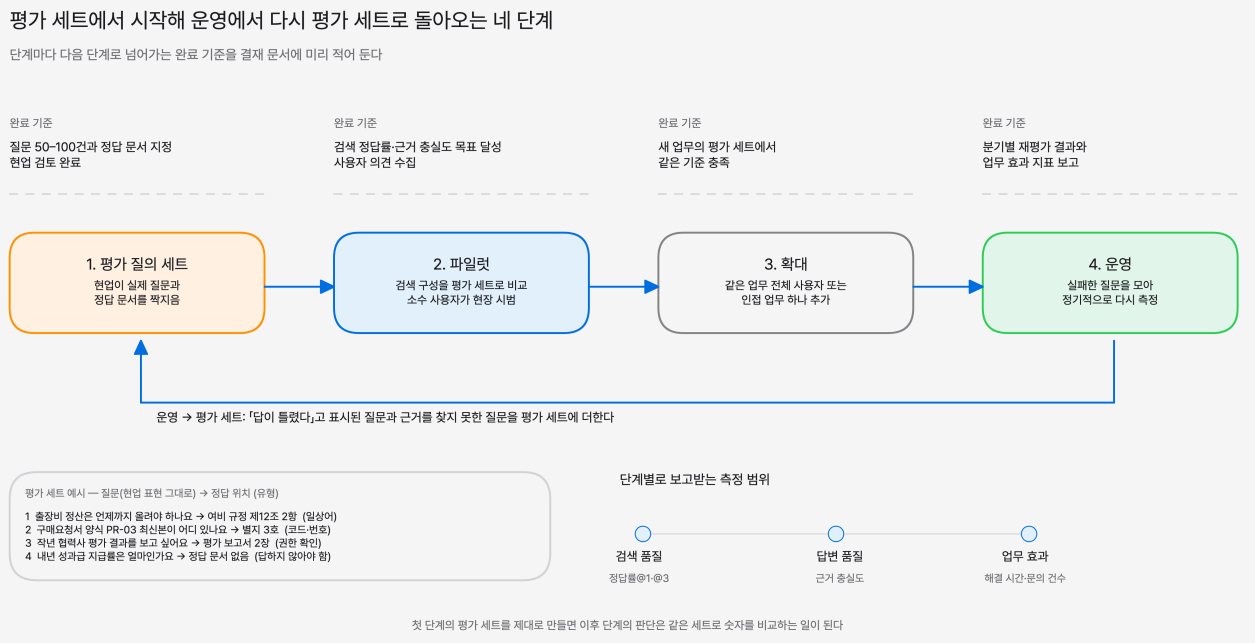

그림. 평가 질의 세트에서 운영까지 이어지는 한국어 RAG 도입 네 단계

단계	하는 일	다음 단계로 넘어가는 완료 기준
1. 평가 질의 세트 구축	현업의 실제 질문과 정답 문서를 짝지어 기준 세트를 만듭니다	질문 50–100건과 정답 문서 지정, 현업 검토 완료
2. 파일럿	검색 구성을 평가 세트로 비교해 고른 뒤 소수 사용자가 현장에서 씁니다	검색 정답률·근거 충실도 목표 달성, 사용자 의견 수집
3. 확대	같은 업무의 전체 사용자로 넓히거나 인접 업무 하나를 더합니다	새 업무의 평가 세트에서 같은 기준 충족
4. 운영	실패한 질문을 모아 평가 세트를 보강하고 정기적으로 다시 잽니다	분기별 재평가 결과와 업무 효과 지표 보고

네 단계 가운데 일정이 가장 자주 밀리는 단계는 첫 단계인 평가 질의 세트 구축입니다. 현업 담당자가 평소 업무를 하면서 질문을 모으고 정답 문서를 지목해야 하므로, 이 작업을 누가 언제까지 맡는지 결재 문서에 이름과 기한으로 적어 두는 것이 좋습니다. 두 번째 단계부터는 같은 평가 세트를 반복해서 쓰기 때문에, 첫 단계를 제대로 마치면 이후 단계의 판단은 숫자를 비교하는 일로 단순해집니다.

8.2.1 실제 질문과 정답 문서로 평가 세트를 먼저 만드는 첫 단계

평가 세트는 IT 담당자가 아니라 현업 담당자가 만들어야 합니다. 직원이 실제로 쓰는 표현과 정답 문서를 아는 사람이 현업이기 때문이고, IT 담당자가 문서 제목을 보고 질문을 지어내면 키워드 검색에 유리한 질문만 모이기 쉽습니다. 질문은 메일, 메신저, 헬프데스크 티켓에서 실제로 오간 문장을 그대로 옮기고, 질문마다 정답 문서와 그 안의 위치를 함께 적습니다.

질문 유형도 고르게 섞어야 구성 비교가 의미를 갖습니다. 4.3.2의 예시 실험처럼 문서와 다른 일상어로 묻는 질문, 5.1.1에서 다룬 제품 코드나 조항 번호처럼 정확한 문자열이 필요한 질문, 권한이 있는 사람만 답을 받아야 하

는 질문을 각각 넣습니다. 정답 문서가 없는 질문을 몇 건 포함하면 시스템이 근거 없이 답을 지어내는지 확인할 수 있습니다.

번호	질문(현업 표현 그대로)	정답 문서	정답 위치	질문 유형
1	출장비 정산은 언제까지 올려야 하나요	여비 규정	제12조 2항	일상어 질문
2	구매요청서 양식 PR-03 최신본이 어디 있나요	구매 업무 지침	별지 3호	코드·번호 질문
3	작년 협력사 평가 결과를 보고 싶어요	협력사 평가 보고서	2장 표 4	권한 확인 질문
4	내년 성과급 지급률은 얼마인가요	(정답 문서 없음)	—	답하지 않아야 하는 질문

예시 표의 네 행은 각각 다른 실패를 확인하는 용도입니다. 첫 행은 문서와 다른 말로 물었을 때 의미 검색이 정답 조항을 찾는지, 둘째 행은 양식 번호 같은 정확한 문자열을 키워드 검색이 놓치지 않는지를 봅니다. 셋째 행은 열람 권한이 없는 사용자에게 보고서가 검색되지 않는지를, 넷째 행은 정답 문서가 없을 때 시스템이 「근거를 찾지 못했다」고 답하는지를 확인합니다. 유형별로 몇 건씩 넣었는지 함께 적어 두면 구성 비교 결과를 유형별로 나눠 읽을 수 있습니다.

평가 세트가 준비되면 4장과 5장에서 설명한 검색 구성을 같은 세트로 비교합니다. 4장에서 보았듯이 한국어 BM25는 형태소 분석기 적용 여부에 따라 재현율이 크게 달라졌지만(4.2.1절), 공개 비교표의 상위 임베딩 모델 세 개는 점수 차이가 0.013 이내로 작았습니다(4.3.1절)[S37][S33]. 즉 형태소 분석 적용 여부는 큰 차이를 만들고, 상위 임베딩 모델 가운데 무엇을 고를지는 자기 조직 문서로 다시 재야 결정할 수 있습니다. 평가 세트는 이처럼 구성 선택과 확산 결정 모두에 근거를 주는 기준 자료입니다.

8.2.2 검색 품질에서 답변 품질과 업무 효과로 넓히는 측정 범위

측정 범위는 검색 품질, 답변 품질, 업무 효과 순으로 넓힙니다. 검색 단계에서 정답 문서를 찾지 못하면 답변 품질을 재는 의미가 없고, 답변이 근거에 충실하지 않으면 업무 시간이 줄어도 그 결과를 믿기 어렵기 때문입니다. 답변 품질은 RAGAS처럼 답이 검색된 문맥에 근거하는지(근거 충실도), 질문에 맞는 답인지(답변 관련성), 검색된 문맥이 질문과 관련 있는지(문맥 관련성)를 나눠 재는 방법을 쓰면 사람이 모든 정답을 작성하지 않아도 일부를 자동으로 평가할 수 있습니다[S12].

측정 범위	주로 보는 단계	지표	참고 사례
검색 품질	평가 세트 구축, 파일럿	정답률@1, 정답률@3, 재현율	4장·5장의 비교 방식
답변 품질	파일럿, 확대	근거 충실도, 답변 관련성, 문맥 관련성	RAGAS 평가 체계[S12]

측정 범위	주로 보는 단계	지표	참고 사례
업무 효과	확대, 운영	해결 시간, 문의·티켓 건수, 문서 접근률	LinkedIn 해결 시간 (3.3.2)[S24], IBM 인사 티켓 건수(3.2.2)[S20]

운영 단계에서는 이 세 지표를 정기적으로 다시 재야 합니다. RAG 시스템이 실패하는 일곱 가지 유형을 분석한 연구는 RAG의 견고성이 처음부터 설계로 완성되기보다 운영하면서 다듬어진다고 결론지었습니다[S7]. 그래서 운영 단계에서는 사용자가 「답이 틀렸다」고 표시한 질문과 근거를 찾지 못한 질문을 모아 평가 세트에 더하고, 분기마다 같은 세트로 검색 구성과 답변 품질을 다시 측정합니다. 의사결정권자는 파일럿 단계에서는 검색 정답률과 근거 충실도를, 확대와 운영 단계에서는 업무 효과 지표를 보고받는 것으로 정해 두면 됩니다.

8.3 결재 전에 확인하는 한국어 RAG 도입 점검표

파일럿 결재 전에는 데이터·권한, 한국어 처리·검색 구성, 효과 지표·책임의 세 묶음을 점검합니다. 앞의 두 묶음은 파일럿 일정과 구성을 정하고, 마지막 묶음은 파일럿이 끝난 뒤 무엇을 근거로 누가 확산을 결정할지를 정합니다. 다음 점검표는 회의 자료로 그대로 쓸 수 있도록 확인 칸을 두었습니다.

8.3.1 문서 상태와 접근 권한, 보안 요건을 묻는 데이터 점검 항목

데이터 준비 상태는 파일럿 일정을 가장 크게 좌우합니다. 벡터 저장소를 여러 부서가 함께 쓰면 다른 부서 문서가 검색 결과에 섞이거나 임베딩에서 원문이 복원될 수 있으므로, 권한 인식 저장소와 데이터 분할을 먼저 설계해야 합니다[S10]. 적은 수의 조작 문서로도 답을 바꿀 수 있다는 연구 결과(6.3.1절)가 있으므로[S11], 누가 어떤 경로로 문서를 등록하는지도 결재 전에 정해 두어야 합니다.

확인	점검 항목	누가 답하는가	관련 절
☐	대상 문서의 저장 위치(공유 드라이브·메일·업무 시스템)와 건수를 파악했는가	현업, IT 담당	1.2
☐	HWP, 스캔 PDF, 표·도면 문서의 비중을 세고 글자 추출 결과를 표본으로 확인했는가	IT 담당	1.2.1, 3.5
☐	최신본과 폐기본을 구분하는 기준과 문서 갱신 주기를 정했는가	현업	6.1.1
☐	부서·직급별 열람 권한을 검색 결과에도 그대로 적용하는 권한 필터를 설계했는가	보안 담당, IT 담당	6.3.1
☐	외부 반출이 제한된 문서가 있는지, 모델 실행을 조직	보안 담당	6.3

확인	점검 항목	누가 답하는가	관련 절
	내부에 두어야 하는지 확인했는가		
☐	문서 등록 경로와 등록 권한을 통제해 악성 문서 주입을 막았는가	보안 담당, IT 담당	6.3
☐	질문, 답변, 출처 문서를 감사 기록으로 남기는가	보안 담당	7.3.1

점검표의 앞쪽 세 항목은 현업과 IT 담당자가 문서 목록을 함께 들여다보아야 답할 수 있습니다. 저장 위치와 건수, 형식별 비중, 최신본을 가리는 기준은 대상 업무 문서를 직접 세어 보는 과정에서 드러나며, 이 숫자가 있어야 파일럿 일정과 색인 범위를 현실적으로 잡을 수 있습니다. 뒤쪽 네 항목은 보안 담당이 결재 전에 답해야 하는 항목입니다. 권한 필터와 문서 등록 경로, 감사 기록은 파일럿 사용자가 적을 때 설계해 두어야 확산 단계에서 사용자가 늘어도 같은 원칙을 그대로 적용할 수 있습니다.

사람이 쓰는 문서와 시스템이 넣는 문서의 구분

권한 설계에 앞서 확인할 질문이 하나 더 있습니다. 대상 업무에서 사람이 직접 작성하는 문서 종류와 시스템이 자동으로 만들어 넣는 문서 종류를 지금 구분할 수 있는지입니다. 어떤 조직은 장애 보고서를 담당자가 직접 쓰게 하고, 어떤 조직은 점검 결과 보고서를 모두 자동 생성한 뒤 수정할 수 없게 잠가 둡니다. 이 구분이 서 있으면 문서 종류별로 등록 권한과 수정 권한을 나눠 악성 문서가 들어올 경로를 좁힐 수 있고, 서 있지 않으면 그 구분을 정하는 일이 색인 작업보다 앞섭니다. 권한을 몇 단계로 나눌 수 있는지보다 이 구분이 실제 저장소에 적용되어 있는지를 먼저 묻는 편이 파일럿 일정을 덜 흔듭니다.

8.3.2 형태소 분석·한국어 임베딩·리랭커 구성을 묻는 검색 점검 항목

검색 점검의 목적은 한국어 검색 구성을 기본값으로 두지 않고 평가 세트로 정했는지 확인하는 데 있습니다. 제로샷 평가에서 BM25는 여러 분야에 걸쳐 튼튼한 기준선이었고 의미 검색은 학습 분야 밖에서 BM25보다 낮은 경우가 많았으므로[S4], 키워드 검색과 의미 검색 중 하나만 고르기 전에 둘을 모두 재야 합니다. 한국어판 리랭커는 모델 카드에서 원본보다 높은 정확도를 보였지만(5.3.2절)[S36], 응답 시간이 늘어나므로 적용 전후를 함께 비교합니다.

확인	점검 항목	근거 장·절
☐	키워드 검색에 한국어 형태소 분석기(Nori, Kiwi 등)를 적용하고 공백 분리 결과와 비교했는가	4.2
☐	한국어 임베딩 후보 2–3개를 자기 조직의 평가 세트로 비교했는가	4.3.3
☐	하이브리드 검색은 결합 전과 결합 후의 정답률을 각각 재서 결합 방식과 가중치를 정했는가	5.2

확인	점검 항목	근거 장·절
☐	한국어판 리랭커 적용 전후의 정답률과 응답 시간을 함께 쟀는가	5.3.2
☐	문서 조각 크기와 조각에 문맥을 덧붙이는 방식을 정했는가	6.2.2
☐	문서량이 작은 업무는 긴 문맥 입력 방식과 비교했는가	2.3.2
☐	표·도면·스캔본이 많은 경우 페이지 이미지 검색이 필요한지 판단했는가	3.5.2
☐	RAGAS 같은 평가 체계로 근거 충실도를 재는 절차를 마련했는가	6.3.2

검색 점검표의 항목은 모두 「무엇을 골랐는가」보다 「무엇과 비교해서 골랐는가」를 묻습니다. 형태소 분석기는 공백 분리와, 한국어 임베딩은 다른 후보와, 하이브리드 검색은 결합 전과, 리랭커는 적용 전과 비교한 기록이 있어야 이 점검표를 통과한 것으로 봅니다. 비교 기록이 남아 있으면 운영 중에 모델을 바꾸거나 문서가 크게 늘었을 때도 같은 평가 세트로 다시 재어 이전 결과와 나란히 놓을 수 있습니다.

8.3.3 파일럿 성공 기준과 확산 결정 책임자를 정하는 효과 점검 항목

마지막 묶음은 파일럿 대상 업무 1개와 그 성공 기준을 적는 결정 양식입니다. 앞의 두 점검표가 파일럿을 시작할 준비가 되었는지를 확인한다면, 이 양식은 파일럿이 끝난 뒤 무엇을 보고 확산을 결정할지를 미리 정합니다. 따라서 빈칸은 결재 회의에서 현업 부서장, IT 담당, 보안 담당이 함께 채우고, 작성 예의 숫자는 조직의 현재 기록을 보고 바꿔 씁니다.

항목	기입란	작성 예
파일럿 대상 업무	_____	인사 규정 문의 응대
대상 사용자와 문서 범위	_____	본사 직원, 인사 규정·지침 문서
검색 품질 기준	_____	평가 세트 정답률@3 목표치
답변 품질 기준	_____	근거 충실도 목표치(RAGAS)
업무 효과 지표와 목표	_____	인사팀 문의 건수, 답 확인 시간의 감소율
평가 기간	_____	8주
확산 여부 결정 책임자	_____	인사 부서장, IT 부서장
확산 시 다음 업무 후보	_____	8.1.2의 두 번째 파일럿 후보

양식에서 가장 자주 비어 있는 칸은 업무 효과 지표와 확산 여부 결정 책임자입니다. 검색 품질과 답변 품질 기준은 평가 세트로 잴 수 있지만, 업무 효과는 현업 부서가 도입 전 기록을 갖고 있어야 비교할 수 있고, 결정 책임자

가 정해져 있지 않으면 파일럿 결과가 좋아도 확산 결정이 미뤄지기 쉽습니다. 두 칸을 결재 회의에서 먼저 채우면 나머지 칸은 IT 담당과 현업이 실무 협의로 채울 수 있습니다.

이 양식의 칸이 모두 채워지면 파일럿 대상 업무 1개와 성공 기준이 정해지고, 도입은 8.2의 첫 단계인 평가 질의 세트 구축으로 넘어갑니다. 다만 평가 기간이 끝났을 때 목표에 못 미친 지표가 있으면 바로 확산하지 말고, 실패한 질문을 분석해 검색 구성이나 문서 준비 상태 가운데 어디를 고칠지 정한 뒤 같은 평가 세트로 다시 잽니다.

9장: MSAP.ai 가 한국어 RAG 를 구축하고 운영하는 방식

MSAP.ai는 한국어 업무 문서를 색인해 질문에 맞는 근거 문서를 찾고, 답과 출처를 함께 돌려주는 한국어 RAG 를 조직 안에 설치해 운영하는 구성을 제공합니다. 4장부터 8장까지 정리한 조건인 형태소 분석, 한국어 임베딩, 하이브리드 검색, 리랭커, 권한과 보안, 평가 세트는 어느 하나만 빠져도 한국어 문서에서 정답률이 떨어지므로, 제품을 고를 때는 기능 목록보다 이 조건이 구성 안에서 어떻게 채워지는지를 먼저 확인해야 합니다. 아래에서는 앞 장의 조건마다 MSAP.ai 구성을 짝지어 보이고, 8장 점검표로 고른 파일럿 업무 하나를 MSAP.ai 측과 진행하는 순서, 확산 판단 기준을 차례로 정리합니다.

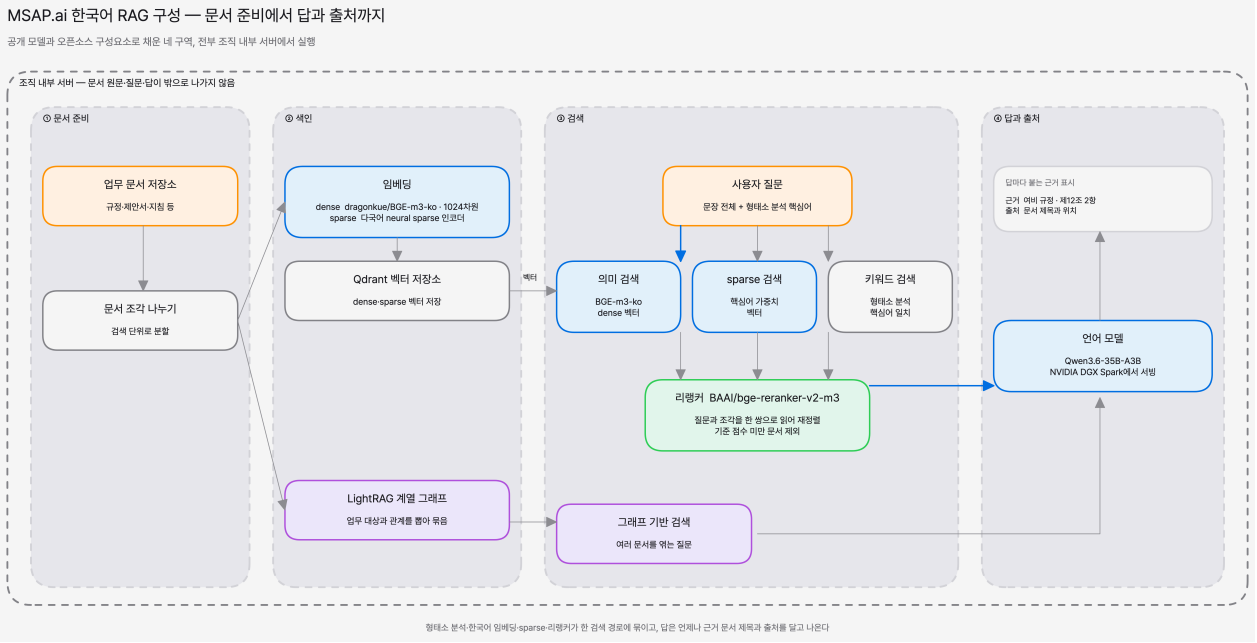

그림. MSAP.ai 한국어 RAG 구성도 — 문서 준비에서 하이브리드 검색과 리랭커를 거쳐 답과 출처까지

9.1 MSAP.ai 한국어 RAG 의 하이브리드 검색과 리랭커 구성

한국어 문서 검색의 정답률은 서로 다른 검색 방식을 어떻게 묶는지와 마지막에 순위를 어떻게 보정하는지에서 크게 달라집니다. 5장 5.1.2절과 5.2.1절에서 본 MIRACL 한국어 평가에서도 의미 검색이 BM25를 크게 앞섰고, 의미 검색에 sparse 검색을 더하면 점수가 더 올랐습니다[S32]. MSAP.ai는 이 결과가 가리키는 방향대로 의미 검색, 핵심어 가중치 검색, 키워드 검색을 한 검색 경로에 묶고 그 뒤에 리랭커를 두는 구성을 씁니다.

MSAP.ai가 제공하는 RAG 구성의 개요는 함께 읽을 자료 「MSAP.ai RAG 서비스 소개」[R3]에서 볼 수 있습니다.

9.1.1 한국어 임베딩과 형태소 분석기를 함께 쓰는 MSAP.ai 하이브리드 검색

한국어 임베딩 BGE-m3-ko 기반 의미 검색

MSAP.ai의 의미 검색은 공개 모델 dragonkue/BGE-m3-ko를 dense 임베딩으로 쓰며, 문서 조각 하나를 1024차원 벡터로 바꿔 질문과의 의미 거리를 잽니다. 이 모델은 다국어 모델 BGE-M3를 한국어 검색 데이터로 다시 학습한 것이고, 4장 4.3.1절의 한국어 검색 7종 비교에서 원본 bge-m3와 상용 영어 중심 임베딩 API보다 높은 점수를 받았습니다[S33][S34]. 따라서 4장에서 영어 중심 임베딩과 한국어 임베딩 사이에 드러난 정답률

차이를 구성 단계에서 먼저 줄여 둔 선택이라고 볼 수 있습니다. MSAP.ai는 공개 사전학습 모델을 별도 튜닝 없이 그대로 쓰므로, 도입을 검토하는 조직이 모델 카드와 공개 평가 결과를 직접 대조해 볼 수 있습니다. 다만 모델 카드 점수는 공개 데이터셋 기준이므로 파일럿에서는 고객 문서로 만든 평가 세트로 KURE-v1 같은 다른 한국어 임베딩 후보와 함께 다시 재야 합니다(4.3.3).

형태소 분석기로 핵심어를 뽑는 질의 처리

MSAP.ai는 질문 문장 전체를 의미 검색에 넣는 것과 함께, 한국어 형태소 분석기로 질문에서 뽑은 핵심어를 키워드 검색과 핵심어 가중치 검색에 따로 넣습니다. 「계약서를」「계약서의」처럼 조사가 붙어 형태가 바뀐 어절을 「계약서」로 줄여 두면 키워드 검색이 어형 변화 때문에 문서를 놓치는 일이 줄어듭니다. 4장 4.2.1절에서 본 것처럼 형태소 분석만으로도 한국어 BM25의 재현율이 크게 올랐습니다[S37]. 질문 하나가 의미 벡터 검색과 핵심어 검색 두 갈래로 동시에 들어가므로, 제품 코드나 조항 번호처럼 문자열이 정확해야 하는 질문과 표현은 달라도 뜻이 같은 문서를 찾아야 하는 질문을 한 구성 안에서 함께 다룰 수 있습니다.

앞 장에서 정리한 한국어 RAG 조건과 MSAP.ai 구성의 대응은 다음과 같습니다. 점검표(8.3.2)의 검색 항목을 하나씩 채울 때 이 표의 오른쪽 열이 확인 대상입니다.

앞 장의 조건	다룬 절	MSAP.ai 구성
한국어 형태소 분석	4.2	질문에서 형태소 분석으로 뽑은 핵심어를 검색에 함께 사용
한국어 임베딩	4.3	dense 임베딩 dragonkue/BGE-m3-ko(1024차원)
키워드·의미 결합	5.1, 5.2	의미 벡터·핵심어 가중치 벡터·키워드 검색으로 후보를 함께 수집
리랭커	5.3	cross-encoder 리랭커 BAAI/bge-reranker-v2-m3, 기준 점수 미만 문서 제외
외부 반출 제한	6.3	임베딩·벡터 저장소·언어 모델을 조직 내부 서버에서 실행
여러 문서를 엮는 질문	6.1, 6.2	LightRAG 계열 그래프 기반 검색
답의 근거 제시	1.3, 7.3	답마다 근거 문서 제목과 출처 표시

9.1.2 리랭커와 sparse 검색을 더해 순위를 보정하는 MSAP.ai 검색 단계

다국어 neural sparse 인코더의 핵심어 가중치

MSAP.ai의 sparse 검색은 공개 모델 opensearch-project/opensearch-neural-sparse-encoding-multilingual-v1을 써서 문서와 질문을 낱말별 가중치 목록으로 바꿉니다. BM25가 낱말의 출현 빈도만 세는 데 비해 학습된 sparse 인코더는 문맥에 따라 낱말마다 중요도를 다르게 매기므로, 핵심어 일치의 정확성은 살리면서 흔한 낱말에 점수가 몰리는 문제를 줄입니다. 같은 경향은 5장 5.1.2절의 MIRACL 한국어 평가에서도 확인되는데, 학습된 sparse 검색이 BM25를 크게 앞섰습니다[S32]. sparse 검색은 의미 검색을 대신하지 않고 의미 검색이 놓친 문자열 일치를 보완하는 경로로 쓰입니다.

cross-encoder 리랭커의 상위 후보 재정렬

세 경로에서 모은 후보는 cross-encoder 리랭커 BAAI/bge-reranker-v2-m3로 넘어가고, 리랭커는 질문과 문서 조각을 한 쌍으로 함께 읽어 관련도 점수를 다시 매긴 뒤 순서를 정합니다. 설정한 기준 점수에 못 미치는 문서는 답변 생성 단계로 넘기지 않으므로, 관련이 약한 조각이 언어 모델의 입력을 차지해 엉뚱한 답을 만드는 일을 줄입니다. 5장 5.3절에서 본 대로 리랭커를 더하면 검색 실패가 더 줄었지만 재순위 방식은 계산 비용이 컸습니다[S4][S6]. 한편 5.3.2절의 한국어판 리랭커는 원본보다 높은 정확도를 보였는데[S36], MSAP.ai의 기본 구성은 원본 공개 모델을 쓰므로 고객 평가 세트에서 원본 모델의 순위 보정이 충분한지를 확인 항목으로 남겨 둡니다.

평가 세트로 정하는 결합 가중과 후보 수

1차 검색에서 가져올 후보 수, 리랭커에 넘길 후보의 배수, 키워드 검색이 돌려줄 최대 조각 수는 MSAP.ai에서 설정값으로 조정합니다. 5.2에서 본 것처럼 두 검색 결과를 같은 무게로 순위 결합(RRF)하면 한국어 임베딩 단독보다 정답률이 낮아질 수 있으므로, 결합 무게와 후보 수는 고객 문서로 만든 평가 세트에서 결합 전후 정답률을 비교해 정해야 합니다[S5]. 다시 말해 리랭커를 켤지, 후보를 몇 건까지 넘길지는 응답 시간과 정답률의 균형으로 판단하는 운영 결정이며 처음 설치할 때 고정되는 값이 아닙니다.

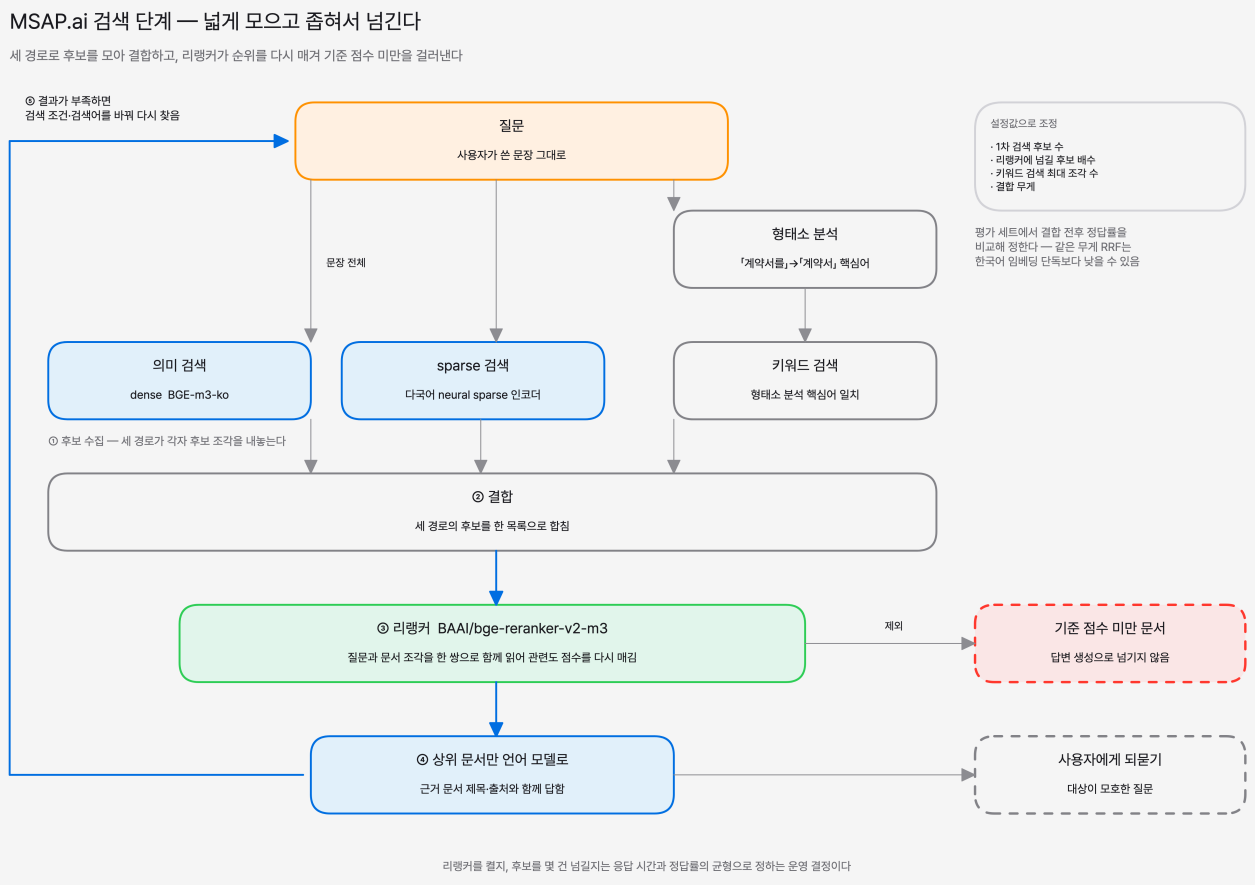

그림. MSAP.ai 검색 단계 — 키워드·의미·sparse 검색의 후보 수집, 결합, 리랭커 재정렬

9.2 조직 안에서 운영하는 MSAP.ai 한국어 RAG 의 모델 서빙과 그래프 검색

업무 문서를 외부 서비스로 보낼 수 없는 기관·기업은 검색 품질보다 먼저 모델과 저장소가 어디에서 실행되는지를 도입 조건으로 묻습니다. 여러 부서가 함께 쓰는 벡터 저장소에서는 다른 그룹 문서가 검색되거나 임베딩에서

원문이 복원될 위험이 있고, 권한 인식 저장소와 데이터 분할이 권고됩니다[S10]. MSAP.ai는 임베딩, 벡터 저장소, 언어 모델을 모두 조직 내부 서버에서 실행하고, 여러 문서를 엮어야 답이 나오는 질문에는 그래프 기반 검색을 더하는 구성을 씁니다.

9.2.1 조직 내부 서버에서 실행하는 임베딩과 언어 모델 서빙 구성

조직 내부 GPU 서버의 언어 모델 서빙

MSAP.ai는 답을 만드는 언어 모델로 오픈 모델 Qwen3.6-35B-A3B를 쓰고, 이 모델을 조직 안에 둔 GPU 서버 NVIDIA DGX Spark에서 서빙합니다. 임베딩 모델과 리랭커도 같은 내부 환경에서 실행되므로 문서 원문, 사용자의 질문, 생성된 답이 조직 밖의 서비스로 전송되지 않습니다. 따라서 망 분리 환경이나 외부 반출 승인이 필요한 문서를 다루는 조직도 기존 반출 제한 규정을 지키면서 파일럿을 진행할 수 있습니다. 다만 내부 서버에서 실행하는 구성은 GPU 자원을 조직이 직접 준비하고 운영해야 하므로, 대상 문서 양과 동시 사용자 수에 맞춰 서버 규모를 파일럿 준비 단계에서 함께 정해야 합니다.

Qdrant 벡터 저장소와 임베딩 재색인

문서 조각의 dense 벡터와 sparse 벡터는 오픈소스 벡터 데이터베이스 Qdrant에 저장되며, 이 저장소도 조직 내부 서버에 설치합니다. 운영에서 가장 주의할 결정은 임베딩 모델 선택인데, 임베딩 모델을 바꾸면 이미 색인한 문서 전체를 새 모델로 다시 벡터화해야 하기 때문입니다. 한국어 비중이 높은 문서에서는 다국어 토크나이저를 쓰는 BGE-M3 계열을 1차 후보로 두고, 파일럿의 구성 비교 단계에서 평가 세트로 모델을 먼저 정한 뒤 대상 문서 전체를 색인하는 순서를 따릅니다. 저장소 안에서 부서별 열람 범위를 어떻게 나눌지는 6.3.1의 권한 누수 대응과 8.3.1의 데이터 점검 항목을 기준으로 파일럿 대상 문서를 고를 때 함께 정합니다.

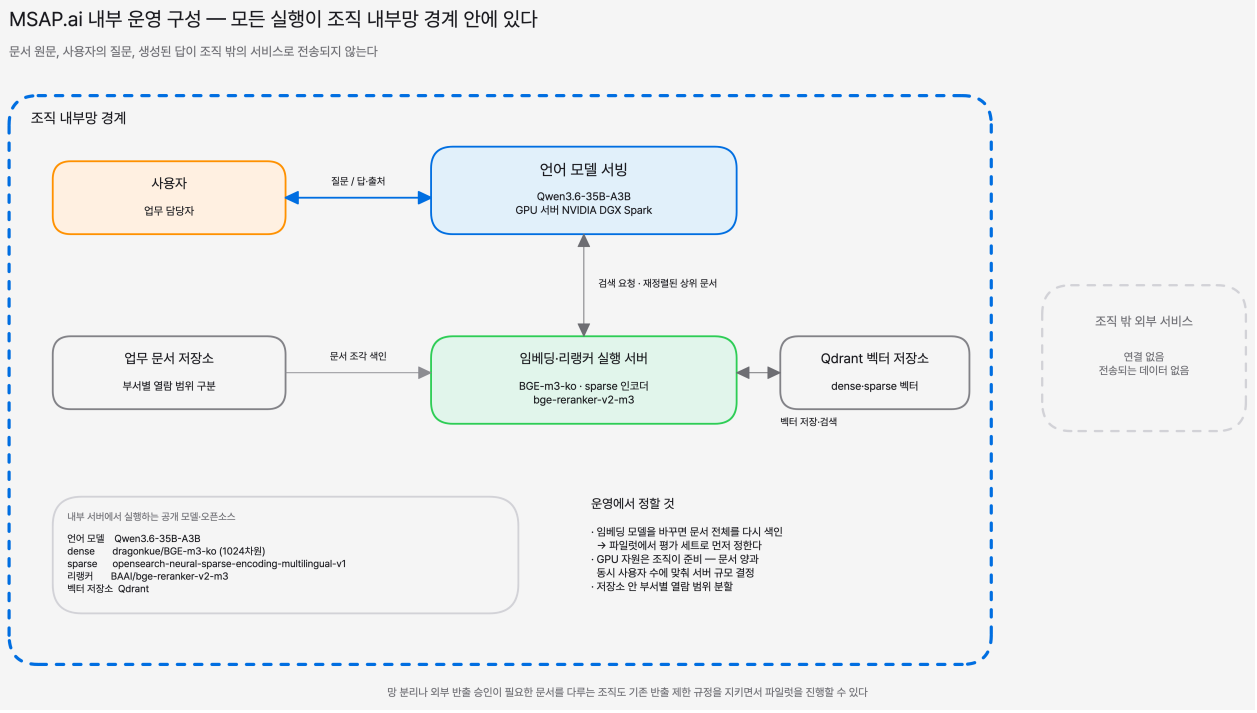

그림. MSAP.ai 내부 운영 구성도 — 문서 저장소, 임베딩, Qdrant 벡터 저장소, 모델 서빙

9.2.2 여러 문서를 엮는 질문을 위한 그래프 기반 검색 구성

LightRAG 계열 그래프 기반 검색

MSAP.ai는 일반 검색과 함께 LightRAG 계열의 그래프 기반 검색 파이프라인을 제공하며, 이 방식은 문서에서 업무 대상과 그 관계를 뽑아 그래프로 묶은 뒤 질문과 관련된 관계를 따라 여러 문서를 함께 찾습니다. 6장 6.1.2절과 6.2.1절에서 본 것처럼 여러 문서를 이어 읽어야 하는 질문에서 기존 RAG는 약했고, 그래프 기반 요약은 전체 요약형 질문에서 일반 RAG보다 나은 결과를 보였습니다[S8][S9]. 다만 그 결과는 언어 모델이 심사한 값이고, 그래프를 만들 때 언어 모델이 문서마다 관계를 추출하므로 색인 비용이 일반 검색보다 큽니다. 6장 기준대로 파일럿 평가 세트에서 여러 문서를 엮는 질문의 비중을 먼저 확인하고, 그 비중이 높은 업무에만 그래프 기반 검색을 켜는 편이 합리적입니다.

검색 결과가 부족할 때의 반복 검색과 에이전트 도구 선택

검색 결과가 질문에 답하기에 부족하면 MSAP.ai의 언어 모델이 검색 조건과 검색어를 바꿔 다시 찾고, 대상이 모호한 질문에는 사용자에게 되묻습니다. 운영 지원 업무에서는 에이전트가 APM·데이터베이스 같은 운영 데이터 조회 도구, 문서 검색 도구 가운데 무엇을 부를지 스스로 정하고, 조회한 수치에 문서 근거를 더해 원인과 조치 방법을 답합니다. 이 구성은 7.2에서 정리한 에이전트의 판단 근거를 찾는 도구로서의 RAG에 해당하며, 답마다 근거 문서 제목과 출처가 붙으므로 7.3의 감사 기록 원칙과도 연결됩니다.

6장에서 정리한 RAG 단독 운영의 한계와 MSAP.ai 구성, 파일럿 평가 항목은 다음과 같습니다. 근거 충실도처럼 답변 품질을 재는 항목은 검색 정답률과 따로 측정합니다[S12].

6장의 한계	MSAP.ai 구성	파일럿 평가 항목
문서가 검색에서 빠짐(6.1.1)	키워드·의미·sparse 하이브리드 검색, 반복 검색	평가 세트 정답률@1·@3
정답 문서의 순위 밀림	리랭커 재정렬, 기준 점수 미만 제외	리랭커 적용 전후 정답률과 응답 시간
여러 문서를 엮는 질문(6.1.2)	LightRAG 계열 그래프 기반 검색	복합 질문 비중과 그래프 적용 전후 비교
근거 없는 답	답마다 근거 문서 제목·출처 표시	근거 충실도(RAGAS faithfulness)
외부 반출과 권한 누수(6.3)	임베딩·저장소·모델의 내부 서버 실행	열람 범위별 색인 대상 문서 구분

9.3 MSAP.ai 와 함께 시작하는 한국어 RAG 파일럿 진행 순서

한국어 RAG 파일럿은 업무 하나와 그 업무의 실제 질문으로 시작해야 결과를 판단할 수 있고, 구성 선택이 고객 문서에서 검증된 뒤에야 다른 업무로 넓힐 근거가 생깁니다. RAG의 견고성은 처음부터 설계되기보다 운영하면서 다듬어지므로[S7], 파일럿의 목적은 완성된 시스템을 한 번에 들이는 일보다 평가 세트와 측정 방식을 먼저 갖추는 데 있습니다. MSAP.ai와의 파일럿은 8.2의 네 단계를 그대로 따르고, 단계마다 고객 조직과 MSAP.ai 측이 맡는 일을 나눠 둡니다.

9.3.1 파일럿 업무 하나를 정해 평가 세트로 구성을 고르는 진행 순서

고객 조직이 먼저 준비하는 질문과 문서

파일럿의 첫 준비물은 8.1 기준으로 고른 업무 하나와, 그 업무 담당자가 실제로 받거나 묻는 질문 수십 건, 그리고 질문마다 정답이 들어 있는 문서입니다. 시스템 설치보다 평가 세트가 먼저 있어야 하는데, 평가 세트가 있어야 형태소 분석·한국어 임베딩·결합 무게·리랭커 조합을 고객 문서 기준으로 비교할 수 있기 때문입니다. 대상 문

서의 위치와 형식, 부서별 열람 범위, 외부 반출 제한 같은 보안 요건을 8.3.1 점검표로 정리해 두면 구성 비교 단계에서 색인 범위를 바로 정할 수 있습니다.

단계별로 고객 조직과 MSAP.ai 측이 맡는 일은 다음과 같습니다. 일정·규모는 대상 문서 양과 질문 수에 따라 달라지므로 첫 협의에서 함께 정합니다.

단계	고객 조직이 맡는 일	MSAP.ai 측이 맡는 일	단계 완료 기준
① 평가 세트	실제 질문 수집, 정답 문서 지정, 현업 담당자 지정	평가 세트 양식 제공, 문서 형식별 파싱 범위 검토	질문–정답 문서 쌍 완성
② 구성 비교	대상 문서 사본과 보안 요건 제공	내부 서버 설치, 임베딩·결합·리랭커 조합별 정답률 측정	구성 하나 선택, 정답률 기준선 기록
③ 현장 시범	시범 사용자 지정, 답변 오류 보고	운영 지원, 근거 충실도와 검색 실패 원인 분석	8.3.3에서 정한 성공 기준 달성 여부 판정
④ 운영 확산	확산 결정 책임자의 판단, 다음 업무 선정	다음 업무 문서 색인, 평가 세트 재측정 체계 운영	확산 대상 업무와 평가 주기 결정

구성 비교 단계에서 임베딩을 먼저 정하는 이유

② 구성 비교 단계에서는 임베딩 모델을 가장 먼저 정하고, 그다음에 결합 무게와 리랭커 적용 범위를 조정합니다. 임베딩 모델을 바꾸면 대상 문서 전체를 다시 색인해야 하므로 첫 선택의 비용이 가장 크고, 결합 무게나 후보 수는 색인을 다시 하지 않고도 설정만으로 바꿀 수 있기 때문입니다. 따라서 평가 세트에서 정답률@1과 정답률@3을 함께 기록해 두면, 리랭커를 켰을 때 늘어나는 응답 시간을 정답률 향상과 맞바꿀 가치가 있는지를 숫자로 판단할 수 있습니다.

9.3.2 파일럿 결과를 다른 업무로 넓히는 확산 판단 기준

검색·답변·업무 세 층의 확산 지표

파일럿 결과는 8.2.2의 순서대로 검색 품질, 답변 품질, 업무 효과의 세 층으로 나눠 판단합니다. 검색 정답률이 기준을 넘었더라도 답이 근거 문서에 충실하지 않으면 답변 품질 단계에서 다시 원인을 찾아야 하며, 이때 검색이 맞는 문서를 가져왔는지와 답이 그 문서에 근거했는지를 RAGAS 지표로 따로 재면 원인을 검색 쪽과 생성 쪽으로 나눌 수 있습니다[S12]. 확산 판단에 쓰는 지표·기준은 다음과 같고, 목표값은 8.3.3 양식에서 고객 조직이 정한 값을 그대로 씁니다.

지표	확산 판단 기준	다음 업무 후보
평가 세트 정답률@3	8.3.3에서 정한 목표 이상, 재측정에서도 유지	같은 문서 형식을 쓰는 인접 업무
근거 충실도	근거 문서 밖의 내용을 답한 비율이 목표 이하	규정·기준 문서 중심 업무(3.2 인사·회계)
문서 찾기 시간·반복 문의 건수	파일럿 전 기록 대비 감소 확인	반복 질문이 많은 업무(3.3 고객지원)

지표	확산 판단 기준	다음 업무 후보
여러 문서를 엮는 질문의 정답률	그래프 기반 검색 적용 뒤 목표 이상	계약·구매처럼 문서 간 대조가 많은 업무(3.4)

같은 구성을 다음 업무로 넓히는 순서

확산은 파일럿과 문서 형식이나 열람 범위가 비슷한 업무부터 시작하는 것이 안전한데, 같은 임베딩 모델과 검색 구성을 다시 색인 없이 그대로 쓸 수 있고 평가 세트 양식도 재사용할 수 있기 때문입니다. 반면 스캔본이나 표 비중이 높은 업무, 여러 부서의 열람 범위가 섞인 업무는 3.5와 6.3에서 본 대로 문서 준비와 권한 분할을 다시 점검한 뒤에 넓혀야 합니다. 운영 중에도 새 문서와 새 질문을 평가 세트에 계속 더해 정해진 주기로 다시 재면, 문서가 바뀌거나 질문 유형이 늘어날 때 정답률이 떨어지는 것을 일찍 발견할 수 있습니다.

MSAP.ai 한국어 RAG 파일럿의 첫 협의 안건

결국 파일럿은 한 번의 시험으로 끝나는 일보다 업무 지식 기반을 쌓는 첫 단계에 가깝고, 그 출발점은 자사 업무 가운데 문서 찾기 비용이 큰 업무 하나를 한국어 RAG 파일럿 대상으로 정하는 결정입니다. MSAP.ai와의 첫 협의에는 다음 네 가지를 준비해 오면 평가 세트 작성부터 바로 시작할 수 있습니다.

- 8.1 기준으로 고른 파일럿 대상 업무 1개와 선정 이유
- 그 업무 담당자가 실제로 받는 질문 목록 초안과 정답이 들어 있는 문서의 위치
- 대상 문서의 형식, 부서별 열람 범위, 외부 반출 제한 같은 보안 요건
- 8.3.3 양식의 성공 기준 지표·목표·평가 기간과 확산 여부를 정할 책임자

Appendix

본 부록은 백서 본문에 인용된 참고문헌과 본문에 등장하는 주요 기술 용어의 정의를 모은 참조 자료입니다.

References

- [S1] Shuster 외, Meta (2021). Retrieval Augmentation Reduces Hallucination in Conversation. https://arxiv.org/abs/2104.07567
- [S2] Magesh 외, Stanford University (2024). Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools. https://arxiv.org/abs/2405.20362
- [S3] Ovadia 외, Microsoft (2024). Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs. https://arxiv.org/abs/2312.05934
- [S4] Thakur 외 (2021). BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. https://arxiv.org/abs/2104.08663
- [S5] Cormack 외 (2009). Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. https://plg.uwaterloo.ca/~gvcormac/cormacksigir09-rrf.pdf
- [S6] Anthropic (2024). Introducing Contextual Retrieval. https://www.anthropic.com/news/contextual-retrieval
- [S7] Barnett 외 (2024). Seven Failure Points When Engineering a Retrieval Augmented Generation System. https://arxiv.org/abs/2401.05856
- [S8] Tang·Yang (2024). MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries. https://arxiv.org/abs/2401.15391
- [S9] Edge 외, Microsoft (2024). From Local to Global: A Graph RAG Approach to Query-Focused Summarization. https://arxiv.org/abs/2404.16130
- [S10] OWASP (2025). LLM08:2025 Vector and Embedding Weaknesses. https://genai.owasp.org/llmrisk/llm082025-vector-and-embedding-weaknesses/
- [S11] Zou 외 (2025). PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models. https://arxiv.org/abs/2402.07867
- [S12] Es 외 (2024). RAGAS: Automated Evaluation of Retrieval Augmented Generation. https://arxiv.org/abs/2309.15217
- [S13] Li 외, Google DeepMind (2024). Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach. https://arxiv.org/abs/2407.16833
- [S14] Liu 외 (2024). Lost in the Middle: How Language Models Use Long Contexts. https://arxiv.org/abs/2307.03172
- [S15] Faysse 외 (2024). ColPali: Efficient Document Retrieval with Vision Language Models. https://arxiv.org/abs/2407.01449
- [S16] OpenAI (2024). Morgan Stanley 고객 사례. https://openai.com/index/morgan-stanley/
- [S17] AWS Machine Learning Blog (2024). How Twitch used agentic workflow with RAG on Amazon Bedrock to supercharge ad sales.

https://aws.amazon.com/blogs/machine-learning/how-twitch-used-agentic-workflow-with-rag-on-amazon-bedrock-to-supercharge-ad-sales

- [S18] AWS for SAP Blog (2025). How Rivian accelerated finance operations with AI agents on Amazon Bedrock. https://aws.amazon.com/blogs/awsforsap/how-rivian-accelerated-finance-operations-with-ai-agents-on-amazon-bedrock
- [S19] AWS Machine Learning Blog (2025). How Amazon Finance streamlines regulatory inquiries by using generative AI on AWS. https://aws.amazon.com/blogs/machine-learning/how-amazon-finance-streamlines-regulatory-inquiries-by-using-generative-ai-on-aws/
- [S20] IBM (2025). IBM AskHR 고객 사례. https://www.ibm.com/case-studies/ibm-askhr
- [S21] Uber Engineering (2024). Genie: Uber's Gen AI On-Call Copilot. https://www.uber.com/blog/genie-ubers-gen-ai-on-call-copilot/
- [S22] Uber Engineering (2025). Enhanced Agentic-RAG. https://www.uber.com/blog/enhanced-agentic-rag/
- [S23] Zhang 외 (2023). RepoCoder: Repository-Level Code Completion Through Iterative Retrieval and Generation. https://arxiv.org/abs/2303.12570
- [S24] Xu 외, LinkedIn (2024). Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering. SIGIR 2024. https://arxiv.org/abs/2404.17723
- [S25] Klarna (2024). Klarna AI assistant handles two-thirds of customer service chats in its first month. https://www.klarna.com/international/press/klarna-ai-assistant-handles-two-thirds-of-customer-service-chats-in-its-first-month/
- [S26] AWS Machine Learning Blog (2025). How IskayPet unified corporate knowledge with Amazon Bedrock managed knowledge bases. https://aws.amazon.com/blogs/machine-learning/how-iskaypet-unified-corporate-knowledge-with-amazon-bedrock-managed-knowledge-bases/
- [S27] 영국 정부디지털서비스(GDS) (2024). The findings of our first generative AI experiment: GOV.UK Chat. https://insidegovuk.blog.gov.uk/2024/01/18/the-findings-of-our-first-generative-ai-experiment-gov-uk-chat/
- [S28] 싱가포르 GovTech (2025). AIBots. https://www.tech.gov.sg/products-and-services/for-government-agencies/productivity-and-marketing/aibots/
- [S29] 뉴시스 (2026). AI 법령 비서 시범 서비스 보도. https://mobile.newsis.com/view/NISX20260713_0003706202
- [S30] 디지털투데이 (2026). 국민권익위원회 국민신문고 생성형 AI 서비스 보도. https://ditoday.com/?p=140053
- [S31] 헤럴드경제 (2026). 국세청 생성형 AI 상담 챗봇 확대 보도. https://www.heraldk.com/article/2026042220455753231
- [S32] Chen 외, BAAI (2024). BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation.

https://arxiv.org/abs/2402.03216

- [S33] dragonkue (2025). snowflake-arctic-embed-l-v2.0-ko 모델 카드. https://huggingface.co/dragonkue/snowflake-arctic-embed-l-v2.0-ko
- [S34] dragonkue (2025). BGE-m3-ko 모델 카드. https://huggingface.co/dragonkue/BGE-m3-ko
- [S35] 고려대학교 NLP&AI 연구실 (2024). KURE-v1 모델 카드. https://huggingface.co/nlpai-lab/KURE-v1
- [S36] dragonkue (2025). bge-reranker-v2-m3-ko 모델 카드. https://huggingface.co/dragonkue/bge-reranker-v2-m3-ko
- [S37] Lee 외 (2022). Better Than Whitespace: Information Retrieval for Korean. https://arxiv.org/pdf/2210.05481
- [S38] 고려대학교 (2024). Korean Penalty of Large Language Models Derived by the Tokenizer. ICICPE 2024. https://hiai.korea.ac.kr/wp-content/uploads/2024/12/Korean-Penalty-of-Large-Language-Models-Derived-by-the-Tokenizer_ICICPE_2024.pdf
- [S39] Elastic (8.18). Korean (nori) analysis plugin. https://www.elastic.co/guide/en/elasticsearch/plugins/8.18/analysis-nori-analyzer.html
- [S40] AutoRAG (2025). BM25 retrieval node 문서. https://marker-inc-korea.github.io/AutoRAG/nodes/retrieval/bm25.html
- [S41] Sionic AI (2026). comsat-embed-ko-8b-preview 모델 카드. https://huggingface.co/sionic-ai/comsat-embed-ko-8b-preview

함께 읽을 자료

- [R1] MSAP.ai 블로그. AI 에이전트와 챗봇의 차이. https://www.msap.ai/blog-home/ai-blog/ai-agent-vs-chatbot/
- [R2] MSAP.ai 블로그. RAG란 무엇인가 — 파이프라인 해부와 도입 체크리스트. https://www.msap.ai/llm/what-is-rag/
- [R3] MSAP.ai. RAG 서비스 소개. https://www.msap.ai/product/msap-ai/ai-as-a-service/retrieval-augmented-generation/

Glossary

아래 표는 본문에 등장하는 주요 기술 용어를 가나다순과 알파벳순으로 정리한 것입니다.

용어	정의
감사 기록	질문 한 건이 답이나 처리 결과로 바뀌기까지 검색한 문서, 인용한 조각, 사람의 승인 내역을 나중에 다시 볼 수 있게 남긴 기록
교착어	낱말 뒤에 조사와 어미가 붙어 문법 관계를 나타내는 언어로, 한국어에서 같은 낱말이 여러 형태로 나타나는 원인
권한 인식 저장소	검색할 때 질문한 사용자의 열람 권한을 반영해 권한 밖 문서를 결과에서 걸러 내는 벡터 저장소 구성
그래프 기반 RAG(GraphRAG)	문서에서 개체와 관계를 뽑아 그래프로 묶고 묶음별 요약을 미리 만들어, 문서 묶음 전체를 보는 질문에 답하는 RAG 방식
근거 충실도(faithfulness)	답에 담긴 주장이 검색해 가져온 문서에 실제로 들어 있는 정도를 재는 답변 품질 지표
긴 문맥 입력	검색 단계 없이 문서 전체를 언어 모델 입력에 한꺼번에 넣는 방식
데이터 분할	부서나 보안 등급에 따라 색인과 저장소를 논리적으로 나눠 다른 그룹의 문서가 검색되지 않게 하는 방법
리랭커(reranker)	1차 검색이 고른 후보를 질문과 한 쌍씩 함께 읽어 관련도를 다시 매기고 순위를 정하는 모델
문맥 덧붙임 임베딩(Contextual Retrieval)	문서 조각마다 원래 문서의 어느 부분인지 설명하는 짧은 문장을 붙인 뒤 임베딩하는 방식
반복 검색	질문을 하위 질문으로 나누고 앞선 검색 결과를 보며 다음 검색어를 정하는 과정을 여러 번 되풀이하는 방식
벡터 데이터베이스	임베딩 벡터와 원문, 출처 정보를 함께 저장하고 가까운 벡터를 빠르게 찾아 주는 저장소
색인(indexing)	검색할 수 있도록 문서 조각과 벡터, 문서 이름·위치·작성일·권한 정보를 함께 저장하는 과정
악성 문서 주입	특정 질문을 노린 조작 문서를 지식 기반에 넣어 RAG의 답을 공격자가 원하는 내용으로 바꾸는 공격
에이전트	업무 이벤트를 받아 여러 단계를 차례로 처리하면서 단계마다 검색·조회 같은 도구를 호출하는 프로그램

용어	정의
의미 검색	낱말의 일치 대신 문장의 뜻이 가까운지를 임베딩 벡터의 거리로 비교해 문서를 찾는 검색
임베딩(embedding)	문장의 뜻을 수백에서 수천 개의 숫자로 된 벡터로 바꾼 값 또는 그 변환 과정
임베딩 역변환 공격	임베딩 벡터에서 원문 정보를 상당 부분 되살려 내는 공격
재현율(Recall)	찾아야 할 문서를 얼마나 빠짐없이 찾았는지 나타내는 지표로, Recall@100은 상위 100건 기준
정답률@k	정답 문서가 검색 결과 상위 k건 안에 들어온 질의의 비율
청킹(chunking)	문서에서 꺼낸 글을 조항이나 문단 단위의 알맞은 길이로 자르는 과정
키워드 검색	질문에 들어 있는 낱말이 문서에 그대로 나오는지를 보고 점수를 매겨 문서를 고르는 검색
토큰·토크나이저	토큰은 언어 모델이 읽고 처리하는 글자 묶음이고, 토크나이저는 글을 토큰으로 자르는 도구
파싱(parsing)	HWP·PDF·스캔본 같은 문서 파일에서 글자·표·제목 구조를 읽어 내는 과정
파인튜닝	문서나 예시 데이터로 모델을 추가 학습시켜 가중치에 지식이나 말투를 반영하는 방식
파일럿	업무 하나와 그 업무의 실제 질문으로 검색 구성과 효과를 먼저 검증하는 시범 도입 단계
평가 세트	현업의 실제 질문과 질문마다의 정답 문서를 짝지어 검색과 답변 품질을 반복해서 재는 기준 자료
하이브리드 검색	키워드 검색과 의미 검색을 함께 돌린 뒤 두 결과를 하나의 순위로 합치는 검색 구성
형태소 분석	한국어 문장을 뜻을 가진 가장 작은 단위로 나누고 조사·어미를 구분해 낱말의 원형을 찾는 처리
환각(hallucination)	언어 모델이 사실과 다르거나 근거 없는 내용을 그럴듯하게 만들어 내는 현상
BEIR	18개 데이터셋으로 검색 모델을 별도 학습 없이 비교하는 정보 검색 평가 묶음
BM25	질의 낱말이 문서에 얼마나 자주, 다른 문서에는 얼마나 드물게 나오는지로 점수를 매기는 대표적인 키워드 검색 점수 계산식

용어	정의
ColPali	문서 페이지를 글자로 바꾸지 않고 페이지 이미지 그대로 임베딩해 검색하는 연구 단계의 멀티모달 검색 방식
CRM(Customer Relationship Management)	고객 정보와 상담 이력을 관리하는 고객 관계 관리 시스템
cross-encoder	질문과 문서를 한 입력으로 함께 읽어 관련도 점수를 내는 모델 구조로, 리랭커에 주로 쓰임
dense 검색	문장 전체의 뜻을 밀집 벡터 하나로 나타내 비교하는 의미 검색
ERP(Enterprise Resource Planning)	재무·구매·재고 같은 기업 자원 정보를 통합 관리하는 전사적 자원 관리 시스템
HWP	국내 공공기관과 기업에서 많이 쓰는 한글 워드프로세서 문서 형식
LightRAG	그래프 기반 검색과 일반 벡터 검색을 함께 쓰는 오픈소스 RAG 파이프라인 계열
LLM(Large Language Model)	대규모 문서로 학습해 문장을 이해하고 생성하는 대규모 언어 모델
MCP(Model Context Protocol)	언어 모델이나 에이전트가 외부 도구와 데이터에 연결하도록 정한 모델 도구 연결 규약
MIRACL	여러 언어의 위키백과 문단을 대상으로 한 다국어 검색 평가 데이터셋
MRR(Mean Reciprocal Rank)	첫 번째 정답이 몇 위에 놓였는지의 역수를 평균한 검색 정확도 지표
NDCG@10	상위 10건 안에 정답 문서가 얼마나 위쪽에 놓였는지를 0에서 1 사이 값으로 나타내는 순위 품질 지표
OCR(문자 인식)	스캔본이나 이미지 안의 글자를 읽어 텍스트로 바꾸는 기술
RAG(Retrieval-Augmented Generation)	질문과 관련된 문서를 먼저 검색하고 그 문서를 근거로 생성형 AI가 답을 만들어 출처와 함께 돌려주는 방식
RAGAS	근거 충실도·답변 관련성·문맥 관련성을 나눠 RAG 품질을 자동으로 재는 평가 체계
RRF(Reciprocal Rank Fusion)	여러 검색 결과를 점수 대신 순위의 역수 합으로 합쳐 새 순위를 만드는 결합 방식
Self-Route	먼저 RAG로 답을 시도하고 답할 수 없다고 판단한 질문만 긴 문맥 입력으로 넘기는 라우팅 방식

용어	정의
SOP(Standard Operating Procedure)	업무 처리 순서와 판단 기준을 문서로 정한 표준 운영 절차서
sparse 검색	모델이 낱말마다 중요도를 매긴 희소 벡터로 문서와 질문을 비교하는 검색
Top-1 F1	1위 후보의 정확도를 정밀도와 재현율의 조화 평균으로 나타낸 리랭커 평가 지표



업무에 RAG 가 필요한 이유

CONTACT

WEB
msap.ai
www.msap.ai/

EMAIL
hello@msap.ai

TEL
02-6953-5427
0269535427

YOUTUBE
@msaptv
www.youtube.com/@msaptv

LINKEDIN
linkedin.com/showcas...
www.linkedin.com/showcase/msap-ai/

FACEBOOK
facebook.com/opennaru
www.facebook.com/opennaru

SCAN

© 2026 SMART CF, MSAP.ai. All rights reserved.