/*elice*/ Enterprise

RAG 파이프라인 구축

WEEK 3 | DAY 3

RAG 시스템 아키텍처 구축

Day 1 ✅	Day 2 ✅	Day 3 ★	Day 4
텍스트 → 벡터	**PDF → 벡터DB**	**벡터DB + LLM**	**할루시네이션 대응**
임베딩 + ChromaDB	로딩 + 청킹 + 저장	RAG + 출처 표시	검증 + 최적화

Day1~2에서 만든 벡터DB 위에, 오늘 LLM을 연결하면 RAG가 완성됩니다.

오늘의 학습 목표

RAG 파이프라인 구축

PDF 문서 기반 RAG를
end-to-end로 구축할 수 있다

출처 표시 구현

파일명 + 페이지를 포함한
신뢰할 수 있는 응답 생성

프롬프트 최적화

커스텀 프롬프트로
답변 품질을 개선할 수 있다

데일리 로드맵

오전 1	오전 2	오전 3	점심	오후 1	오후 2	오후 3
체험	이해	심화		구축	개선	정리

오전에 이해하고, 오후에 직접 만듭니다

PREVIEW

시스템 시연

화면을 보면서 따라와 주세요
궁금한 점은 메모해두세요
(Jupyter Notebook 시연)

실습 데이터: 우리가 질문할 문서

동일 제품(RF9000)의 서로 다른 문서 유형으로 RAG를 체험합니다

제품사양서

㈜스마트홈테크 | 4페이지

p.1 표지 (문서번호, 개정이력, 승인란)

p.2 주요 사양표 (용량 868L, 소비전력, 치수…)

p.3 핵심 기능 (FoodAI™ , 멀티에어플로우)

p.4 에러코드표 (E1~E9) + 유지보수

질의 예시: "소비전력은?", "E3 에러코드는?"

시험성적서

㈜한국전자시험원 | 3페이지

p.1 표지 (성적서 번호, 의뢰업체)

p.2 시험 항목 테이블 (전기안전, EMC,
에너지효율, 소음 ~ 기준값/측정값)

p.3 종합 판정 + 특기사항

질의 예시: "에너지효율 등급은?", "소음 결과는?"

이 PDF 2개에 질문하면 답변 + 출처가 나옵니다

데모 ① 사양서에 질문하기

▶ 지금 실행합니다 — 결과를 주목하세요

1

Q1: "RF9000의 총 용량은 얼마인가요?"

사양표에서 용량 정보 검색

2

Q2: "소비전력은?"

기술 수치 정확도 확인

3

Q3: "E3 에러코드는 무엇인가요?"

에러코드표에서 검색

데모 ① 결과: 답변 + 출처가 함께 나온다

Q: "RF9000의 총 용량은 얼마인가요?"

A: RF9000의 총 용량은 868L입니다.
(냉장 524L / 냉동 344L)

출처:
[1] 제품사양서_스마트냉장고_RF9000.pdf (p.2)
[2] 제품사양서_스마트냉장고_RF9000.pdf (p.2)

✅ **숫자가 정확하다**
868L, 524L, 344L

✅ **어디서 찾았는지 나온다**
사양서 p.2

✅ **원본을 열어서 확인 가능**
PDF 대조 가능

데모 ② 시험성적서에도 질문해보자

```
Q: "에너지효율 등급은?"

A: RF9000의 에너지소비효율 등급은 1등급입니다.

출처:
  [1] 시험성적서_RF9000.pdf (p.2) — 기준: 1등급 | 측정값: 1등급 | 적합
  [2] 제품사양서_RF9000.pdf (p.2) — 에너지효율등급: 1등급
```

두 문서에서 교차 확인!

시험성적서에서 기준값/측정값을 찾고, 사양서에서도 같은 정보를 확인했습니다.
RAG가 벡터DB에 들어있는 두 문서를 모두 검색하기 때문에 가능합니다.

데모 ③ 문서에 없는 질문을 하면?

Q: "RF9000의 가격은 얼마인가요?"

A: 해당 정보를 제공된 문서에서 찾을 수 없습니다.
제품사양서와 시험성적서에는 가격 정보가 포함되어 있지 않습니다.

LLM만 사용하면?

"대략 200만원 정도 합니다"
→ 근거 없이 지어냄
→ 할루시네이션!

RAG를 사용하면?

"해당 정보를 찾을 수 없습니다"
→ 문서에 없으면 솔직하게 답변
→ 할루시네이션 방지!

데모 ④ 프롬프트 하나로 답변이 달라진다

같은 질문: "냉장 용량과 냉동 용량은?"

기본 체인

"냉장 용량은 524리터이고
냉동 용량은 344리터입니다."

→ "리터"로 표기 (단위 불명확)

→ 총 용량 누락

→ 출처 페이지 없음

커스텀 체인

"냉장 524L, 냉동 344L,
총 용량은 868L입니다."

→ "L" 단위로 정확

→ 총 용량 자동 포함

→ 프롬프트 규칙 덕분!

어떻게 이런 차이가 나는지는 잠시 후 자세히 다룹니다

QUESTION

질문 시간

사양서나 시험성적서에 궁금한 것을 물어보세요

예 시

› "필터 교체 주기는?"
› "소음 측정 결과가 기준 이내인가?"
› "절연저항 시험 기준은?"

토론: 이 시스템을 어디에 쓸 수 있을까?

방금 본 시스템이 우리 회사에 있다면?

1

어떤 문서를 넣고 싶나요?

사양서, 시험성적서, 매뉴얼, 규격서, SOP…

2

누가 질문하면 좋을까요?

기술지원팀, 품질관리팀, 신입사원, 고객…

3

가장 먼저 해결하고 싶은 문제는?

조별 2분 토론 → 대표가 한 가지씩 공유

제조업 RAG 활용 시나리오

A/S 기술지원

기술지원팀

고객: "E3 에러코드요"

→ 사양서에서 즉시 조치방법 안내

품질검증 조회

품질관리팀

"소음 시험 결과가 기준 이내?"

→ 시험성적서에서 기준/측정값 확인

신입사원 교육

R&D 신규인력

"이 제품의 핵심 기술이 뭐야?"

→ 사양서 기반으로 정확한 설명

안전규격 확인

인증팀

"절연저항 시험 기준은?"

→ IEC 60335 기준 즉시 확인

여러분의 아이디어와 비슷한 게 있나요?

팀별 RAG 활용 아이디어

1조 _____

2조 _____

3조 _____

4조 _____

📌 이 아이디어는 오후 블록6에서 다시 꺼냅니다. 직접 만들어본 후에 생각이 달라질 수 있습니다.

RAG

Retrieval-Augmented Generation = 검색 증강 생성

LLM이 "자기가 아는 것"이 아니라
"우리가 준 문서"를 근거로 답변하게 만드는 기술

LLM만 쓰면

시험에서 기억에만 의존
→ 틀릴 수 있음

RAG를 쓰면

교과서를 펼쳐보고 답함
→ 근거가 있음

RAG 안에서 일어나는 5단계

앞에서 질문했을 때, 이 5가지가 순서대로 일어났습니다

Step 1	Step 2	Step 3	Step 4	Step 5
질의 임베딩	**벡터DB 검색**	**프롬프트 조립**	**LLM 답변**	**답변 + 출처**
"용량이 얼마야?" → 벡터로 변환	비슷한 문서 조각 3개 찾기	"이 문서를 참고해서 답변하세요"	GPT-4o-mini가 문서 기반 답변 생성	답변 + 파일명 + 페이지 반환
		★ 오늘 핵심 1		★ 오늘 핵심 2

학습한 Step 1, 2 에 이어서 오늘 Step 3, 4, 5 를 진행합니다.

오늘 새로 배울 것은 4가지뿐입니다

학습한 내용 (Day 1~2)	오늘 새로 배울 것
✅ 텍스트 → 벡터 변환 (임베딩)	☐ **RetrievalQA 체인 (5줄)**
✅ PDF 로딩 + 청킹	☐ **커스텀 프롬프트 설계**
✅ ChromaDB 저장 + 유사도 검색	☐ **출처 표시 (파일명 + 페이지)**
✅ PromptTemplate (2주차)	☐ **{context} + {question} 연결**

왼쪽은 이미 할 줄 압니다. 오른쪽 4개만 추가하면 RAG가 완성됩니다.

정리

1

문서에 질문하면 정확한 답변이 나온다

숫자, 단위까지 정확하게

2

답변과 함께 출처가 표시된다

파일명 + 페이지 번호 → 원본 확인 가능

3

문서에 없으면 "모른다"고 한다

지어내지 않음 = 할루시네이션 방지

이 시스템을 오늘 여러분이 직접 만듭니다