

/* elice */

Enterprise

직접 만들기

실습: 기본 RAG 구축

13:30 ~ 14:20

오전 내용 정리

오전 1 - 체험

라이브 데모 체험

답변 + 출처 확인

기본 vs 커스텀 차이 확인

오전 2 - 이해

RAG 5단계 코드 확인
(RetrievalQA 5줄)

코드의 역할 이해

커스텀 프롬프트 효과

오전 3 - 심화

출처 표시의 중요성

metadata 구조

ask_with_sources() 함수

이제 직접 만들 차례입니다. 오전에 본 그 시스템을 지금부터 구축합니다.

실습 안내

50분 안에 RetrievalQA 체인을 구축하고 테스트합니다

10분

Section 0~2

환경설정 + 벡터DB 구축 + 검색 테스트

Day2 복습

25분

Section 3

RetrievalQA 체인 구축 + 테스트

핵심

10분

체크포인트

4개 항목 통과 확인

확인



막히면 바로 손 들어주세요.

환경 설정

2분

점심에 미리 실행하셨다면, 바로 다음 슬라이드로 넘어가세요!

```
# Cell 1: 패키지 설치  
!pip install langchain langchain-openai langchain-community chromadb  
pypdf
```

```
# Cell 2: API 키 설정  
import os  
os.environ["OPENAI_API_KEY"] = "sk-..."
```

설치 완료 확인

에러 없이 셀 실행 완료되면 OK

API 키 주의

제공한 키를 그대로 입력

벡터 DB 구축

Day2 복습

3분

이미 학습된 내용으로 셀 실행 진행

1

PDF 로딩

```
PyPDFLoader("data/제품사양서_...pdf")
```

→ all_docs = 7개 (4+3페이지)

2

청킹

```
RecursiveCharacterTextSplitter(500, 50)
```

→ chunks ≈ 15~20개

3

벡터DB 저장

```
Chroma.from_documents(chunks, embeddings)
```

→ ChromaDB에 임베딩 저장 완료



"셀 3개를 순서대로 실행하세요. 결과 숫자만 확인하면 됩니다."

RetrievalQA 체인 구축

오전 2에서 배운 코드를 직접 타이핑합니다

핵심 슬라이드

```
from langchain.chains import RetrievalQA
from langchain_openai import ChatOpenAI

qa = RetrievalQA.from_chain_type(
    llm=ChatOpenAI(model="gpt-4o-mini", temperature=0),           ← Step 4 - LLM 선택
    chain_type="stuff",                                         ← Step 3 - 조립 방식
    retriever=vectorstore.as_retriever(search_kwargs={"k": 3}),   ← Step 2 - 검색기 설정
    return_source_documents=True                                ← Step 5 - 출처 반환!
)
result = qa.invoke({"query": "RF9000의 총 용량은?"})
```

첫 질의 실행!

코드를 실행하고, 첫 번째 질문을 던져보세요

```
result = qa.invoke({"query": "RF9000의 총 용량은?"})  
  
print(result["result"])  
print(result["source_documents"])
```

예상 결과

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

출처: 제품사양서_스마트냉장고_RF9000.pdf (p.2)

답변이 나왔다면, 축하합니다! 여러분의 RAG가 동작합니다!

질의 5개 테스트

아래 질문을 하나씩 실행하고 답변을 확인하세요

사양서 질문 (3개)

Q1 "RF9000의 총 용량은?"

Q2 "소비전력은 얼마인가요?"

Q3 "E3 에러코드는 무엇인가요?"

시험성적서 질문 (2개)

Q4 "에너지효율 등급은?"

Q5 "소음 측정값은 얼마인가요?"

각 질문마다 확인할 것

- ① 답변이 정확한가?
- ② 출처(source_documents)가 나오는가?
- ③ 어떤 PDF의 몇 페이지인가?

답변을 원본 PDF와 대조하기

RAG가 답한 내용이 실제 문서에 있는지 직접 확인해보세요

1

답변 확인

result["result"]에서 답변 내용을 읽습니다

2

출처 확인

result["source_documents"]에서 파일명 + 페이지를 확인합니다

3

PDF 열기

data/ 폴더에서 해당 PDF를 열고 페이지로 이동합니다

4

대조하기

RAG 답변의 숫자·단위가 PDF 원본과 일치하는지 비교합니다

왜 대조하나요? 현업에서는 RAG 답변을 그대로 믿으면 안 됩니다. 출처로 돌아가서 확인하는 습관이 핵심입니다.

오후1 체크포인트 - 4개 항목 통과하기

1

- 패키지 설치 완료

pip install 에러 없이 완료

2

- **all_docs = 7개, chunks ≈ 15~20개**

사양서 4p + 시험성적서 3p = 7개 문서

3

- **RetrievalQA 답변 + source_documents 출력**

질문에 답변이 나오고 출처도 표시됨

4

- **질의 3개 이상 테스트 완료**

사양서 2개 + 시험성적서 1개 이상

트러블슈팅: 흔한 에러 3가지

막혔을 때 여기를 먼저 확인하세요

① ModuleNotFoundError

→ Section 0의 pip install 셀을 다시 실행하세요

커널 재시작 후 다시 시도

② AuthenticationError (API 키)

→ API 키를 다시 복사해서 붙여넣기. 앞뒤 공백 확인

따옴표 안에 정확히 입력

③ FileNotFoundError (PDF 경로)

→ data/ 폴더에 PDF 2개가 있는지 확인. 파일명 오타 주의

경로: data/제품사양서_...



막히면 바로 손 들어주세요.