2050.02.26

김태훈

Advanced RAG Part 1

Query Translation: Multi-Query

목차

01 **프로젝트 설명**

기존 프로젝트: Langmsith를 통한 Document Check
현 프로젝트 방향성: Document 중복 제거 및 커버리지 확대

02 **아키텍처 및 작동 원리**

워크 플로우1: Multi-Query 생성
워크플로우 2: 병렬 RAG 실행
워크플로우 3: 중복 제거 (Unique Union)
워크플로우 4: 최종 답변 생성

03 **구현 방법**

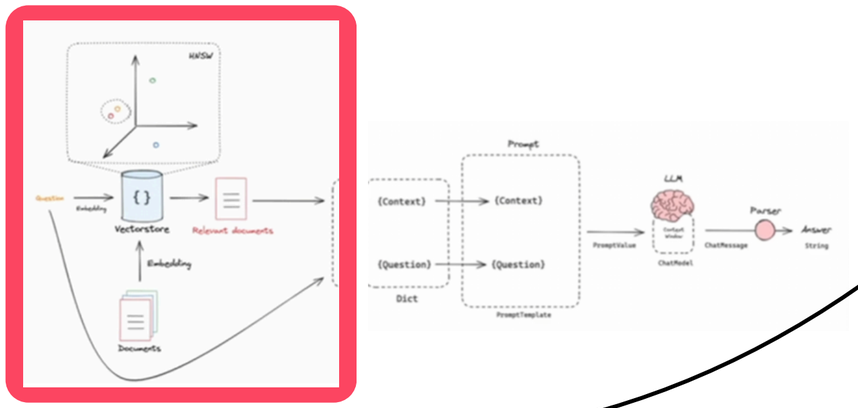
환경 설정: VS Code와 Jupyter Extension을 활용한 개발 환경
코드 구현: Python 기반의 단계별 라이브러리 호출 빛 프로토타입 소스 코드 가이드

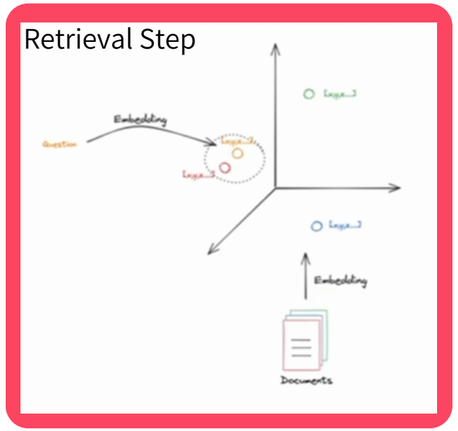
04 **개선 방향**

장단점 분석: 코드 제한 분선을 통한 추후 개선 방향 제시

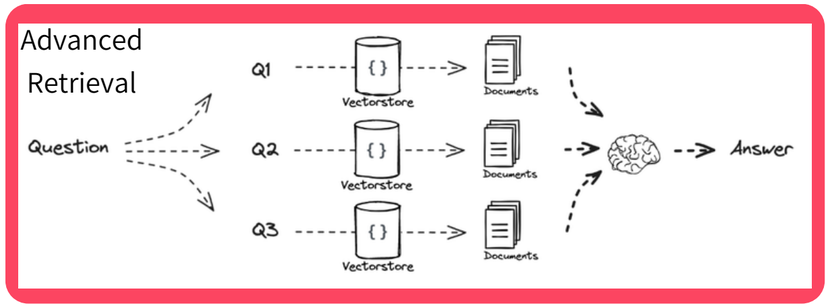
01 프로젝트 설명

AS-IS: RAG 기반 요약 구조의 한계





TO-BE: Multy-Query 생성: 커버리지 확대



02 아키텍처 및 작동 원리

워크 플로우1: Multi-Query 생성

사용자 질문 1개
↓
LLM이 동일한 질문을 5가지 다른 관점으로 재작성
↓
["질문 버전1", "질문 버전2", "질문 버전3", "질문 버전4", "질문 버전5"]

워크 플로우2: 병렬 RAG 실행

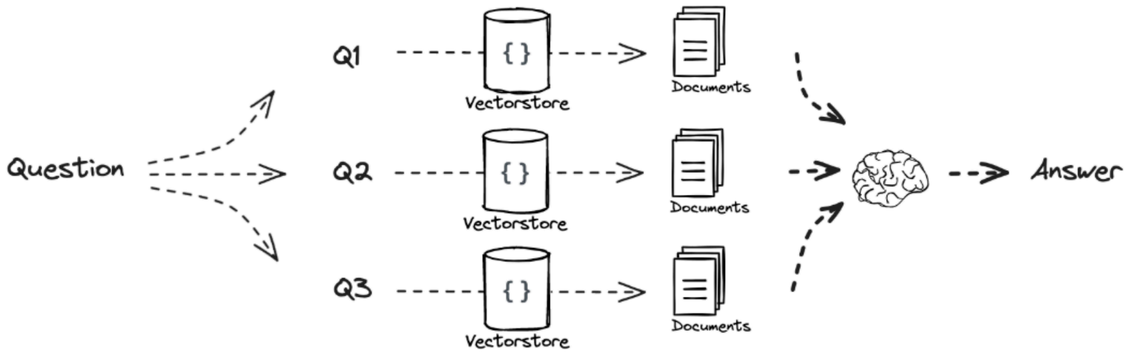
5개의 질문 각각에 대해 retriever를 병렬로 실행 (retriever.map())
↓
[[doc1, doc2, ...], [doc3, doc4, ...], [doc1, doc5, ...], ...]
총 5개의 문서 리스트 반환 (각 리스트당 k개 문서)

워크 플로우 3: 중복 제거 (Unique Union)

5개의 리스트를 하나로 flatten
↓
dumps()로 Document 객체를 string으로 변환 후 set()으로 중복 제거
↓
loads()로 다시 Document 객체로 복원
↓
최종: 중복 없는 unique 문서들

워크 플로우4: 최종 답변 생성

unique 문서들을 context로 사용
↓
LLM이 context + 원본 질문을 바탕으로 최종 답변 생성



03 코드 구현

https://github.com/KimHoon98/RAG-Implementation

0.) README를 통해 페이지에 대한 설명, 프로젝트 순서를 확인할 수 있습니다.
1.) 각 파일마다 README 파일이 첨부 되어있으니 참조하시면 됩니다.
2.) 해당 파워포인트에 대한 설명은 Note explanation Section 참조하세요.

04 개선 방향

RAG 불일치 답변 생성

문제 분석:

- 원인 1. PDF 파싱 과정에서 테이블 구조 붕괴
  - PyPDFLoader에서 연결한 재무 테이블의 구조가 무너짐
- 원인 2. HuggingFace 소형 모델의 한계
  - gemma-2-9b-it는 9B 파라미터 규모의 오픈소스 모델로, GPT-4o 계열 대형 모델 대비 수치 추론 및 표 해석 능력에 한계가 존재

해결 방안:

- 원인 2에 집중하여, 모델 변경 “gemma-2-9b-it” → “gpt-4o-mini” 이후 결과 확인