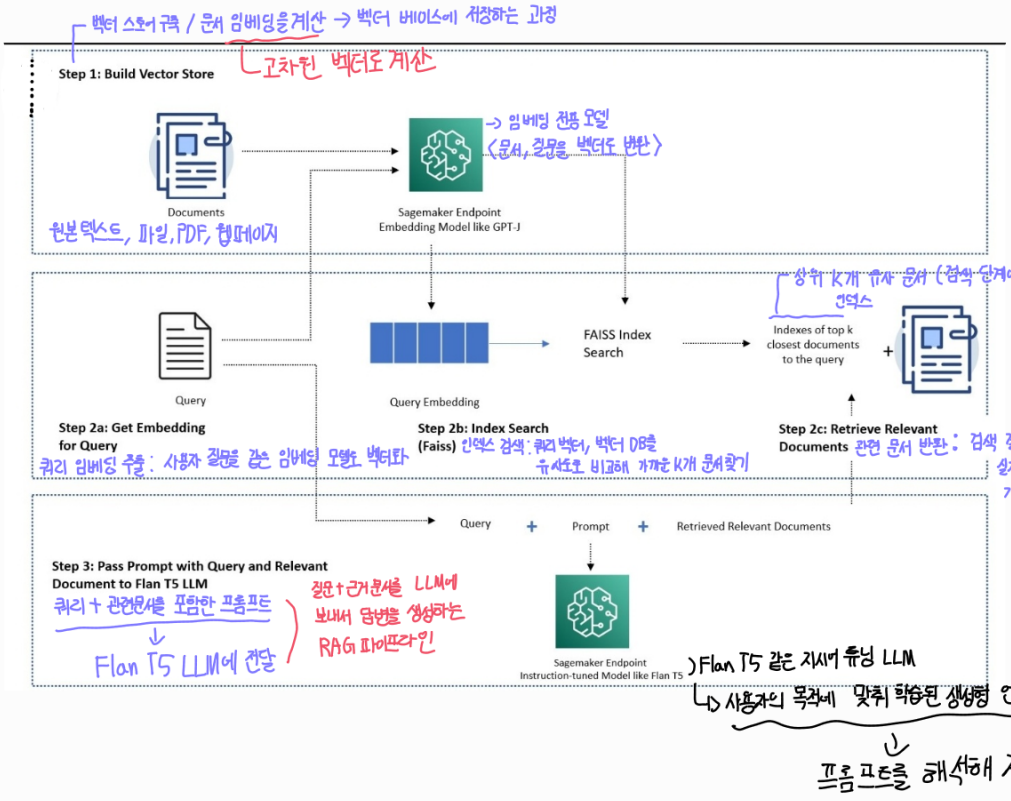
RAP (Retrieval - Augmented Generation)

① 문서 즉석 검색 → 주입 팩트 기반 답변생성

└ 파라미터에 없는 최신, 논문 기술 활용 가능



〈과정〉

질문 임베딩 → 문서검색 → 컨텍스트 구축 → 답변 생성 → 선택적 재귀

↓ 사용자 프롬프트를 벡터로 변환

↓ 벡터 DB, 하이브리드 검색, 지식그래프 질의 } Top-k문서추출

└ 문서조각을 필터, 재정렬 후 프롬프트에 삽입

(프롬프트+문서) 읽고 최종텍스트 생성

└ 응답을 검토, 추가

확률 = 검색확률 , 생성확률

$P(y|x) = \Sigma_z P(y|x,z) , P(z|x)$

* Prompt

: 전체 입력 문자열

사용이유 : 문맥을 이해하게 하고, 역할, 목적, 형식을 따르도록 지시하기 위해서

* Embedding

: 문장, 단어, 문서를 고차원의 실수 벡터로 변환

사용이유 : 의미적 거리를 수학적으로 계산해 비슷한 내용끼리 가깝게, 다른 것은 멀리두기 위해

* Hybrid search

: 키워드 서치 + 벡터검색, BM25 점수 + 벡터 점수 가중합

장점 : 두 방식을 장점 동시에, 노이즈 상쇄

- ① 속도가 빠르고 정확하며 키워드에 강하다. (의미 변형에 약함)
- ② 동의어·유사 문장에 강하다. (연산량이 많고 잡음, 문서가 많음)

* filter

① 메타 데이터
- 검색전에 WHERE 처럼 필터

② 규칙 기반
- 검색 후 if 문처럼 버리기

③ 토픽/엔터티
- 인덱스 태그 + 조건부 탐색

메타 / 규칙 필터

* re-ranking (재정렬)

: 검색한 후 상위 N개 후보중 정말 쓸만한 순서 다시 매기기

모델 종류

Cross-Encoder : 문서, 쿼리를 하나의 입력으로 넣어 정확한 점수 산출
- 품질 상승, 속도 저하

LLM 기반 채점 : LLM에게 문서가 질문답에 얼마나 적절한지 채점
- 품질상승, 비용 시간 절약

Rule + Heuristic : 제목 매칭 여부, 최신 날짜 가산점
- 빠르지만 품질제한

〈지식 그래프와 융합〉

① 엔티티 랭킹 기반 : 프롬프트에서 추출한 엔티티 → KG ID에 매핑
노드, 트리플 검색

② 경로 추론 기반 : KG에서 질의응답 경로 찾아 텍스트로 변환

③ Hybird RAG : 텍스트 + KG 벡터 = 하나의 인덱스
단계별 두 retriever, Cascad
└ 정밀도와 설명 가능성 동시에 높이기