Lang chain & RAG

2025.10.28

20211955 김선권

LLM의 한계점

1.LLM은 특정 기간동안의 정보로 학습을 하기 때문에 그 이후에 일어난 일에 대한 답변을 불가능

2. LLM모델은 토큰 제한이 있음 → 대용량의 정보를 통한 챗봇 AI생성 어려움

3. Hallucination : 엉뚱한 대답을 하거나 거짓말을 하는 경우가 많음

LLM의 개량

1.Fine-tuning : 기존 모델의 가중치를 조정하여 원하는 용도의 모델로 업데이트

2.N-shot Learning : n개의 출력 예시를 제시하여, 용도에 맞는 출력을 하도록 조정

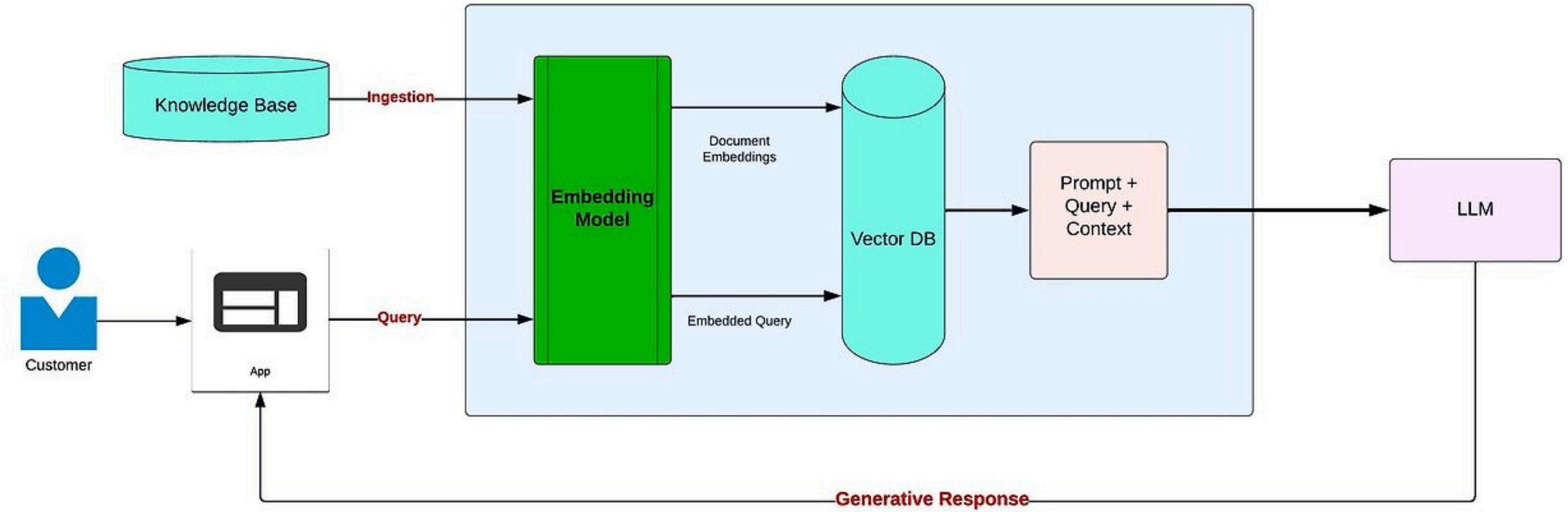
3.In-context Learning : 문맥을 제시하고, 이 문맥 기반으로 모델이 출력하도록 조정

→ lang chain은 In-context learning의 도구

Lang chain : LLM을 활용한 애플리케이션을 쉽게 개발할 수 있도록 도와주는 소프트웨어 프레임워크

RAG(Retrieval Augmented Generation)

외부 데이터를 참조하여 LLM이 답변할 수 있도록 해주는 프레임워크



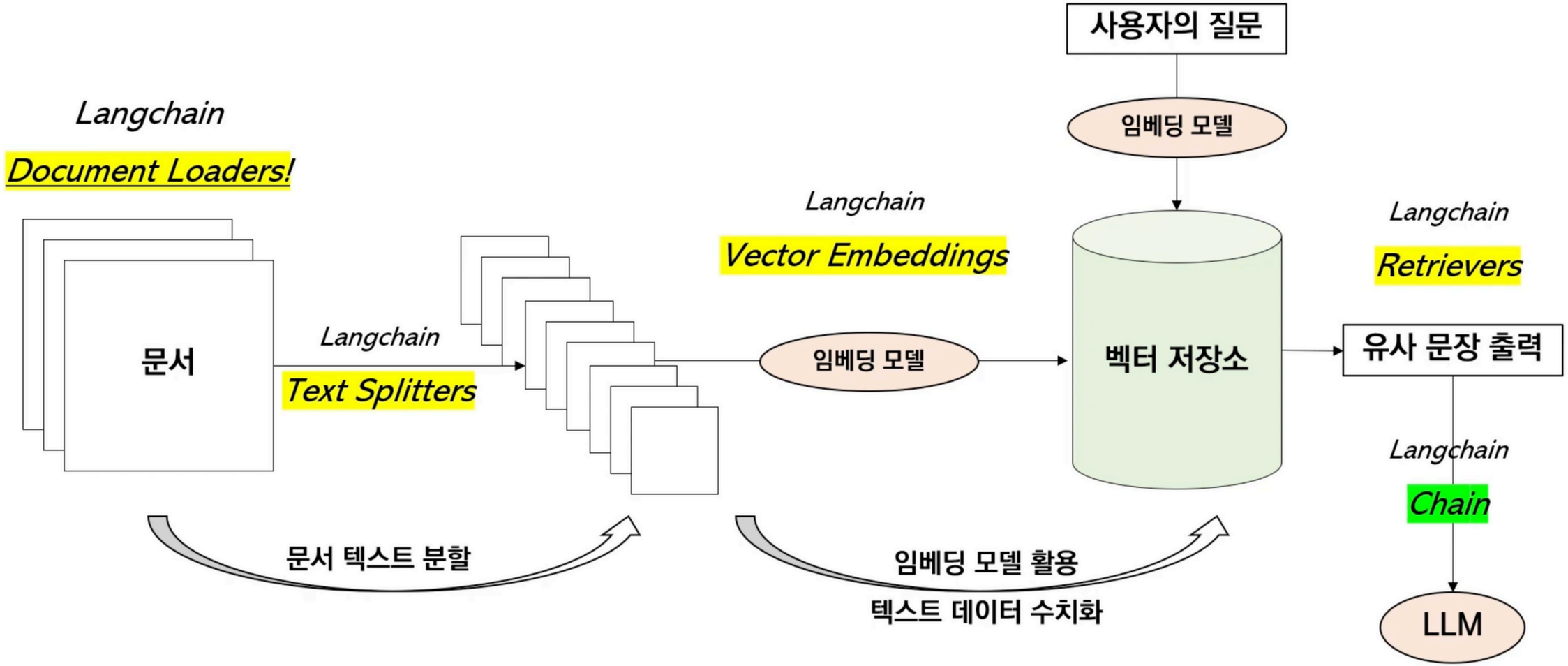
RAG(Retrieval Augmented Generation)

1.준비단계

수집(Load)
분할(Split)
임베딩(Enbed)
저장(Store)

2.실행 단계

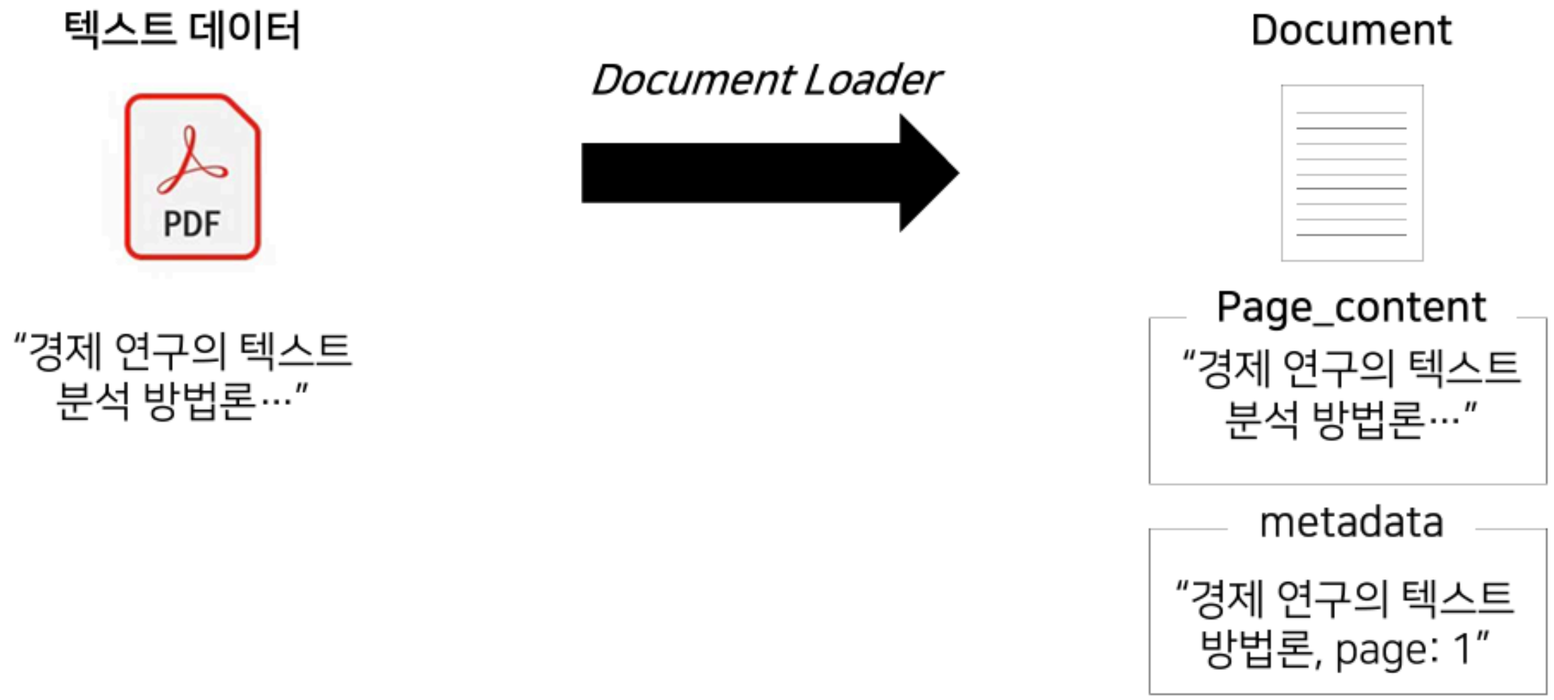
질문(Query)
검색(Retrieve)
증강(Augment)
생성(Generate)


사용자의 질문
임베딩 모델
Langchain
Document Loaders!
문서
Langchain
Text Splitters
Langchain
Vector Embeddings
임베딩 모델
벡터 저장소
Langchain
Retrievers
유사 문장 출력
Langchain
Chain
LLM
문서 텍스트 분할
임베딩 모델 활용
텍스트 데이터 수치화

RAG(Retrieval Augmented Generation)

1.Document Loaders

RAG가 참고할 데이터는 여러 비정형 데이터들이 많기 때문에 이와 같은 다양한 소스에서 데이터를 읽어와 처리가 가능한 표준화된 문서 형식으로 변환하는 과정



RAG(Retrieval Augmented Generation)

2. Text Splitters

토큰 제한이 있는 LLM이 여러 문장을 참고해 답변할 수 있도록 문서를 분할하는 역할

1. CharacterTextSplitter

   구분자 1개(/n/n) 기준으로 분할하므로 max token을 넘어가는 경우 발생

2. RecursiveCharacterTextSplitter

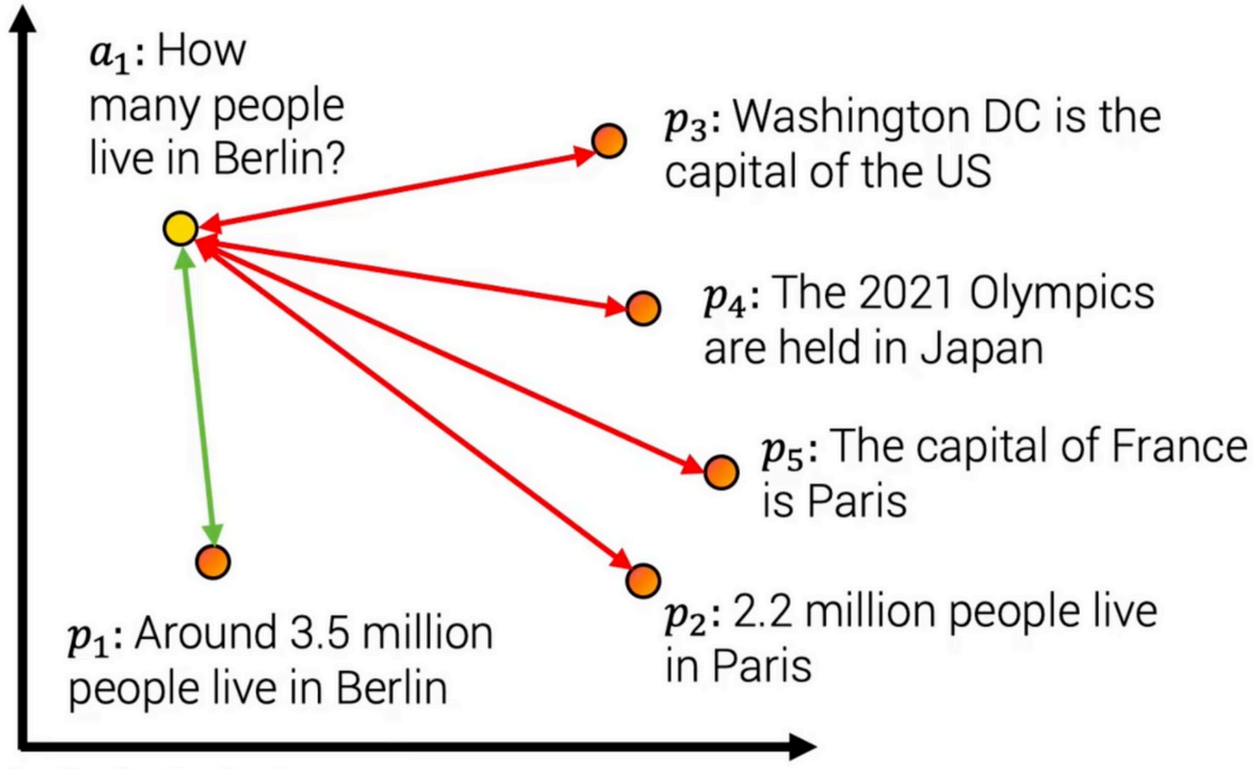
   줄바꿈, 마침표, 쉼표 순으로 재귀적으로 분할해서 max token을 넘지않음

RAG(Retrieval Augmented Generation)

3.Text Embedding

text splitter가 청킹한 텍스트를 숫자로 변환하여 문장 간의 유사성을 비교할 수 있게 함.

코사인 유사도로 유사성 비교



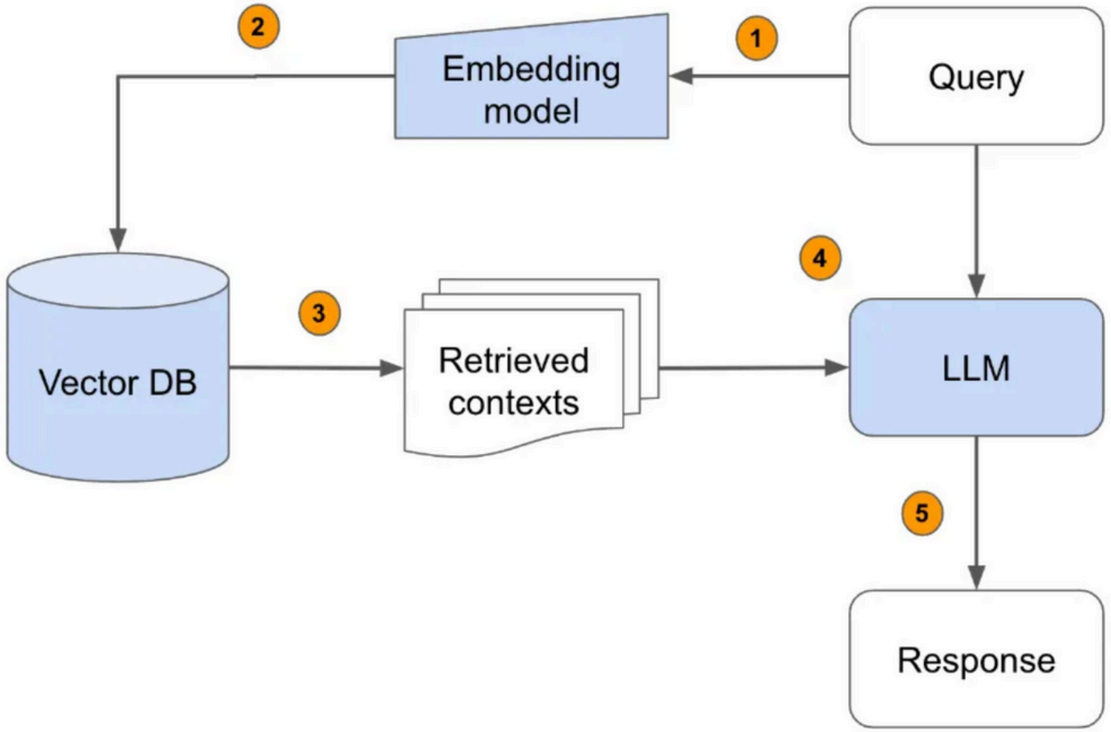
대용량의 말뭉치로 임베딩 모델을 훈련하여 사전 학습된 모델을 통해 쉽게 임베딩해서 텍스트를 수치로 변환가능

RAG(Retrieval Augmented Generation)

4.Vector stores

자연어 → 숫자 처리 후 이들을 저장하는 벡터 저장소

임베딩 된 데이터를 인덱싱하여 input으로 받는 query와의 유사도를 빠르게 출력



사용자의 질문이 들어왔을 때 이 수백만 개와 일일이 비교하는 것은 매우 비효율적이므로

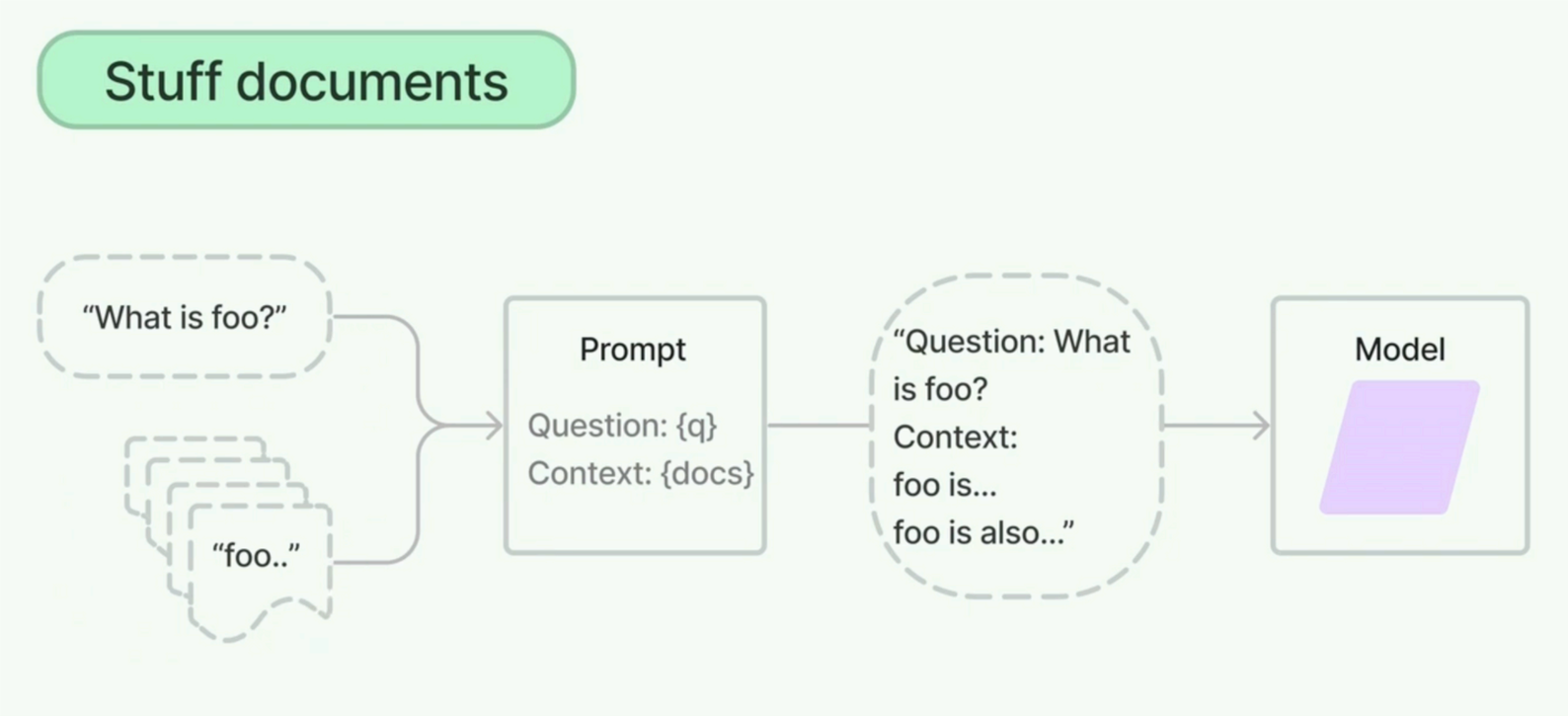
벡터 스토어는 이 벡터들을 저장하고, 특정 벡터와 가장 가까운 벡터들을 초고속으로 찾아주는 DB

RAG(Retrieval Augmented Generation)

4.Retrievers 사용자의 질문을 받아서 벡터스토어에서 가장 관련성 높은 텍스트 청크를 찾아옴

1.Stuff

찾아온 모든 문서 청크를 하나의 프롬프트에 전부 집어넣고 질문과 함께 LLM에 한 번에 보냄



RAG(Retrieval Augmented Generation)

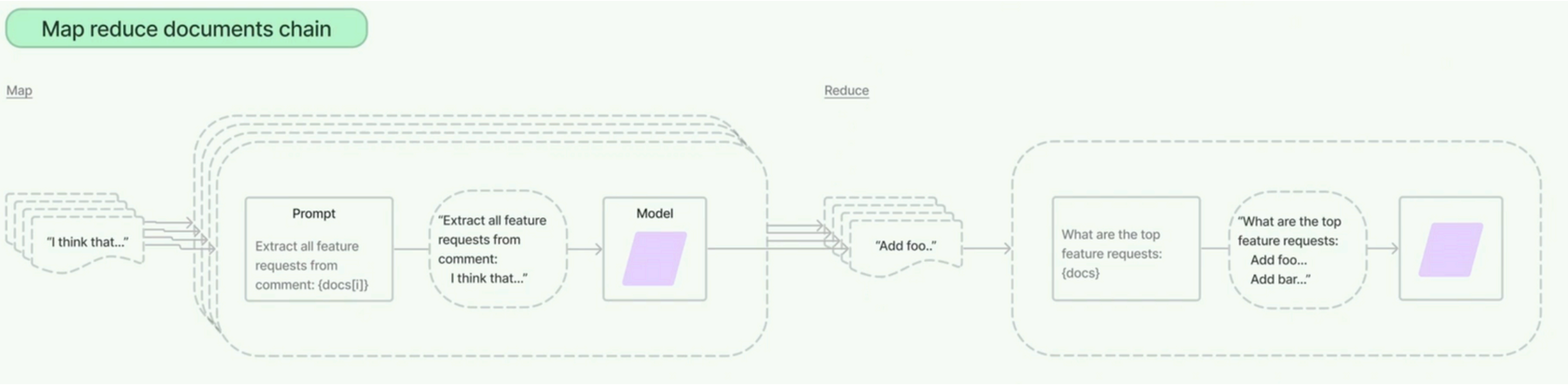
4.Retrievers

사용자의 질문을 받아서 벡터스토어에서 가장 관련성 높은 텍스트 청크를 찾아옴

2. Map-reduce

분할된 텍스트 청크마다 요약을 생성하고(Map), 이를 합친 최종 요약을 생성함(reduce).

다수 호출 필요(속도 저하)



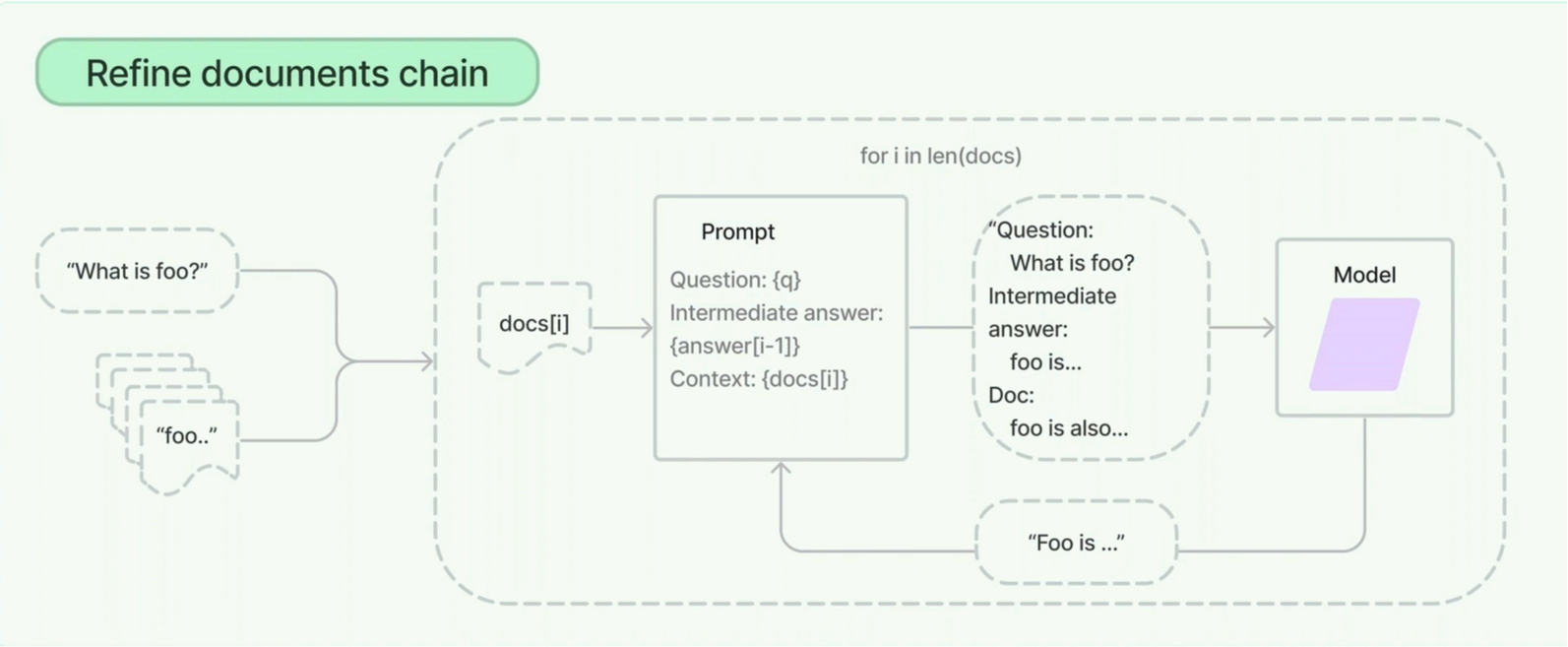
청크가 N개 → N+1번 LLM호출

RAG(Retrieval Augmented Generation)

4.Retrievers 사용자의 질문을 받아서 벡터스토어에서 가장 관련성 높은 텍스트 청크를 찾아옴

3. refine

분할된 텍스트 청크를 순회하면서 누적 답변(Intermediate answer) 생성



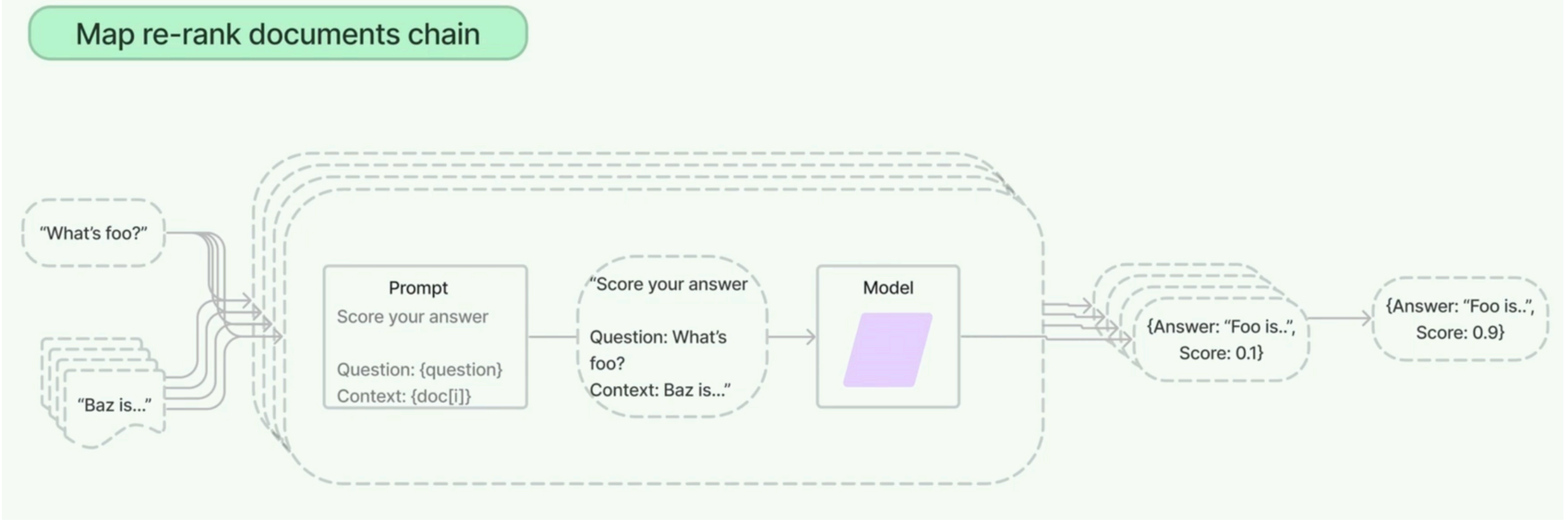
청크가 N개 → LLM N번 호출

RAG(Retrieval Augmented Generation)

4.Retrievers 사용자의 질문을 받아서 벡터스토어에서 가장 관련성 높은 텍스트 청크를 찾아옴

4. Map-Rerank

Map-reduce방식처럼 병렬로 청크를 보냄.
청크에 대한 답변에 score를 매김(정확도, 유사도)



Lang chain & RAG

End

2025.10.28

20211955 김선권