

retrival

2023년 10월 30일 월요일 오전 7:18

RAG : 나의 knowledge base에서 search하는것이다

process

document loader(Textloader,directorloader...) -> document transformers(Recursive,character..) -> embedding(BGE,openaiembedding..)-> vector store db(chroma.FAISS)-> retrival(similar , contextual pipeline,cohere reranker,,,)



keyword retrival : bm25

embedding retrival : LLMEmbedding

hybrid retrival

reranking : retrive diverse document

diversity ranker : 유사도로 랭킹하고 top k만 가져오면 정보의 다양성이 제한 될수있기 때문에 이거 사용

Rerank

[RAG But Better: Rerankers with Cohere AI](#)



[산학 리뷰할때 나와 비슷한 사람 찾을때 사용해보는거 고려 해보기]

top k로 모두 버리면 정말 관련있는 doc버리게 되잖아 이걸 rerank해서 해결할수있다는 이야기이다

[Cohere Reranker](#) | [Langchain](#)

multi query in RAG

[LangChain Multi-Query Retriever for RAG](#)



[MultiQueryRetriever](#) | [Langchain](#)

user의 query에 대한 여러 질문을 만들어준다

generateQuery로 query질문과 비슷한 질문 3개 뽑을수있는데 callback을 써야 한다 id도 임의로 정해야 한다
또는 log정보를 가져와서 사용할수도 있다

Contextual compression : 프로젝트에서 사용했던 pipeline이다

Ensemble retriever (hybrid search)

multivector Retriever : multiple vectors per document

parentDocumentRetriever : doc을 작게 쪼개고 parent id 참고 하게 한다

[Advanced RAG 02 - Parent Document Retriever](#)



large doc content를 다 넣고 싶지 않다 관련있는 내용만 찾기를 원한다 : split doc from origine

each split doc -> embedding 한다

이러고 사용자가 query를 하게 되면 query는 -> embedding -> child doc -> parent doc 점점 context가 커지는 환경으로 단계가 진행된다

2가지 방법이 있는데 그중 하나는 parent의 chunk size를 크게 잡고 child는 작게 잡고 vectorstore만들기

다양한 doc이 있고 많은양의 결과를 얻고 싶을때 유용하다

self query

[Advanced RAG 01 - Self Querying Retrieval](#)



사용자의 query를 semantic similarity comparison하는 것 뿐 아니라
extract filter from user query on the metadata of stored doc

doc에 저장되어있는 여러 속성들을 필터 통해 추출하는 것도 포함한다는 것이다

Graph LLM

[Graph_RAG_LlamaIndex_Workshop.ipynb - Colaboratory \(google.com\)](#)

cypher

[LlamaIndex Workshop: Building RAG with Knowledge Graphs](#)



[LangChain: Introduction and Getting Started | Pinecone](#)

embedded chain이라는 langchain비슷한 틀있는데 이거가 rag중심으로 나왔나봐 혹시 나중에 또 업데이트하는 일 생기면
이거 사용해봐도 좋을것 같다

