랭체인 기본 개념 정리

202000826 김연범

1. 랭체인(Langchain)이란?

RAG 시스템을 구축하기 위해 설계된 다양한 툴 중 하나. 즉, LLM을 사용해 애플리케이션 생성을 단순화하도록 설계된 프레임워크의 일종이다.
일반적으로 랭체인은 사용자의 프롬프트를 곧바로 LLM에 전달하지 않고, 하나의 프롬프트 탬플릿을 거쳐 전달하도록 추가적인 연결고리를 만들어내는 프롬프트 엔지니어링을 수행한다.
이외에도 다양한 모듈을 제공하는데, 불러온 데이터들을 여러 청크로 분할하는 Text Splitter이나, 분할된 청크들을 저장하는 Vector stores 등 광할한 LLM 서비스 개발을 위한 도구들이 있다.

2. RAG 기본 원리

RAG(Retrieval Augmented Generation)는 LLM의 환각 현상을 해결하는 방법 중 하나로, 검색-증강-생성의 3단계를 거쳐 LLM이 사실에 근거한 답변을 하도록 만든다.

a) Document Loader를 통해 Document 객체 형태로 문서를 로드
b) Text Splitter를 통해 여러 청크(Chunk)로 분할하고 이를 벡터 DB에 저장
c) 사용자의 질문에 답할 근거를 벡터 DB에 검색한 후, 이를 LLM에 전달 및 답변 취득

3. 임베딩

텍스트를 수치로 변환하는 작업으로써, 다량의 텍스트로 사전 학습된 모델이 활용된다. 이 임베딩 작업은 RAG 과정 중 다음과 같은 부분에 사용된다.
a) 문서 청크를 벡터 DB에 저장할 때
b) 사용자의 질문에 답할 근거를 벡터 DB에서 검색할 때

※ 참고로 (b)의 과정에서는 벡터 간 거리를 측정함으로써 서로 다른 벡터 간에 얼마나 유사한지를 알 수 있기 때문에, 일반적으로 코사인 유사도를 활용해 벡터 간 거리를 측정해 유사도를 파악한다.

4. BERT(Bidirectional Encoder Representations from Transformers)

트랜스포머의 인코더만으로 구성된 언어모델로, 레이블링 없이 비지도학습으로 사전 훈련된다. 임베딩 과정은 바로 이, 문장 임베딩을 위해 문장 임베딩에 특화된, Sentence-BERT를 활용한다.

5. Sentence-BERT

a) 레이블링 되지 않은 대량의 텍스트 문서를 학습했기 때문에, 단어나 문장의 맥락 정보를 세부적으로 파악하는 능력을 갖춤
b) 동음이의어와 같은 맥락적 정보를 수치로 표현할 수 있음

6. Retriever

RAG의 문서 검색기로, 사용자의 질문과 답변 근거가 될 문서의 연결을 해주는 가장 중요한 임무를 도맡는다. Retriever가 고려해야 될 사항은 다음과 같다.
a) 사용자의 질문을 어떻게 해석할 것인가? -> 사용자의 질문을 다양한 방식으로 해석해 답변하도록 만들어야 함
b) 답변 근거가 될 문서를 어떻게, 얼마나 가져올 것인가? -> 유사도 순위 기준 N개를 선정할지, 다양한 근거를 포함하도록 N개를 선정할지.

7. 벡터 DB

여러 문서를 벡터 형태로 저장한 DataBase, 내지는 여러 문서의 청크를 임베딩하여 얻어낸 임베딩 벡터의 DataBase를 말한다. 벡터 DB는 다음과 같은 특징을 가진다.

a) 기존 RDB와 달리 비정형 데이터를 저장하는 데 특화된 데이터베이스. 즉, 고차원의 벡터 데이터를 효율적으로 저장하고 검색할 수 있도록 설계.
b) RDB와 데이터 조회 방식이 다름. 즉, 벡터 유사도 계산을 통해 데이터를 조회하는데, 일반적으로 ANN 알고리즘을 바탕으로 임베딩 벡터 간 유사도를 계산, 검색에 활용한다.

※ 벡터 DB의 종류에는 다음과 같은 것들이 있다.
a) 순수 벡터 데이터베이스 ex) Chroma, Weaviate, Qdrant, Pinecone, zilz 등
장)
- 임베딩 벡터를 저장하고 검색하는 것에 초점을 두어 효율적인 유사성 검색이 가능하다.
- 기본적으로 코사인 유사도를 포함한 벡터 연산을 지원하고 확장성이 뛰어남
단)
- 벡터 검색에 특화돼 있어 SQL 기반의 기존 DB와 결합이 어려움
- CRUD가 부실해 시스템 유지보수가 원활하지 못함.
- 벡터 데이터 인덱싱을 하는 계산 과정이 무거워 느리며, 비용이 많이 듦.

b) 텍스트 전용 데이터베이스 ex) Elasticsearch, OpenSearch, Apache Lucene, Solr 등
장)
- 텍스트 데이터를 저장하고 이를 조회하기 위한 기능이 가장 크게 발전됨
- 다국어 검색 지원, 커스터마이징 가능한 토크나이저, stemmer, 불용어 목록, N-gram 지원

단)
- 벡터 검색을 위한 유사도 계산에 최적화되지 않아 RAG 시스템에 최적의 솔루션을 제공하지 못함

c) 벡터 라이브러리 ex) FAISS, Annoy, Hnswlib 등
장)
- 코사인 유사도와 같은 알고리즘 뿐만 아니라, 벡터 압축을 통한 빠른 검색 기능도 지원
- 텍스트 임베딩이나 이미지 임베딩과 같은 고차원 데이터를 효율적으로 검색할 수 있어 AI영역에서 주로 활용됨

단)
- 데이터를 조회하는 방법으로 벡터 유사도 외 메타데이터 필터링이 중요한데, 이 필터링 기능을 지원하지 않음
- 유지보수에 어려움이 있음

d) 벡터 기능이 추가된 NoSQL과 벡터 저장 및 검색 가능한 SQL DB : 논외

※ Chroma DB
- RAG 구축 시 가장 많이 활용되는 오픈 소스 벡터 DB, 즉 순수 벡터 데이터베이스이다.
- 단순한 사용성과 유사도 순위 기능 제공, 빠르다는 장점이 있다.

8. ANN(Approximate Nearest Neighbor) 알고리즘

a) HNSW(Hierarchical Navigable SmallWorld)
계층적 구조를 이용해 빠른 검색이 가능하며, 대규모 데이터셋에서도 매우 빠른 검색 속도를 제공한다.

b) SPTAG(Space Partition Tree And Graph)
공간 분할 트리와 그래프를 결합한 방식으로, 데이터를 트리 구조로 분류하고, 그래프를 통해 세부 검색을 진행한다.

9. LECL(Langchain Expression Language)

여러 모듈을 엮어 하나의 파이프라인으로 만들어주는 역할, 즉 chain을 만드는 것을 간단하게 구현할 수 있도록 해준다.
ex) 프롬프트 탬플릿 - LLM(모델) - 출력 파서(Output Parser)