스터디 1주차_rag의 개념

1. 연구 배경 및 문제 정의

최근 대규모 언어 모델(LLM)의 발전으로 자연어 질의응답, 요약, 추천 등 다양한 작업이 가능해졌다. 그러나 LLM을 단독으로 사용할 경우 몇 가지 한계점이 존재한다. LLM은 사전에 학습된 데이터에 기반하여 답변을 생성하기 때문에, 최신 정보나 특정 도메인의 내부 데이터를 정확하게 반영하기 어렵다. 또한 숫자나 사실 기반 질문에 대해 그럴듯하지만 실제와 다른 답변을 생성하는 환각(hallucination) 문제가 발생할 수 있으며, 답변의 출처를 명확하게 제시하지 못한다는 문제도 있다.

예를 들어 “삼성전자 2023년 매출이 얼마인가?”와 같은 질문을 LLM에 단독으로 질의할 경우, 모델은 학습 데이터에 기반하여 대략적인 수치를 추론하거나 잘못된 연도의 정보를 제공할 가능성이 있다. 이 경우 답변이 자연스럽게 보일 수는 있으나, 실제 데이터와 다를 경우 심각한 신뢰성 문제를 야기한다.

이러한 한계를 해결하기 위한 방법으로 Retrieval-Augmented Generation(RAG) 구조가 주목받고 있다.

---

2. RAG 개념 및 기본 구조

RAG는 Retrieval-Augmented Generation의 약자로, 답변을 생성하기 전에 외부 문서에서 관련 정보를 검색(Retrieval)하고, 해당 정보를 근거로 답변을 생성(Generation)하는 구조이다. 즉, LLM이 자신의 내부 지식만으로 답변을 생성하는 것이 아니라, 개발자가 제공한 문서 집합을 참고하여 답변을 생성하도록 유도하는 방식이다.

RAG의 기본적인 동작 과정은 다음과 같다.

1. 사용자가 자연어 질문을 입력한다.
2. 입력된 질문을 임베딩(embedding)하여 벡터 형태로 변환한다.
3. 사전에 구축된 문서 임베딩과 질문 임베딩 간의 유사도를 계산한다.
4. 유사도가 높은 문서 조각(chunk)을 검색한다.
5. 검색된 문서 조각을 LLM의 입력 프롬프트에 포함한다.
6. LLM은 제공된 문서를 근거로 답변을 생성한다.

이러한 구조를 통해 RAG는 최신 정보 반영, 사실 기반 답변 생성, 출처 제시가 가능하다는 장점을 가진다.

3. RAG의 범위와 데이터 편향 문제

RAG의 중요한 특징 중 하나는 **검색 범위가 자동으로 정해지지 않는다는 점**이다. RAG는 인터넷 전체를 검색하는 구조가 아니라, **개발자가 사전에 제공한 데이터셋만을 검색 대상**으로 사용한다. 즉, RAG의 지식 범위는 전적으로 설계자가 어떤 문서를 선택하느냐에 따라 결정된다.

이러한 특성은 데이터 편향 문제와도 직접적으로 연결된다. LLM 단독 사용 시에는 모델이 학습한 데이터의 편향을 그대로 답변에 반영할 수밖에 없지만, RAG에서는 특정 출처나 특정 관점의 데이터만 포함하거나 제외하는 방식으로 편향을 어느 정도 통제할 수 있다. 예를 들어, 기업 분석 RAG 시스템에서 재무제표와 공식 공시 자료만을 포함하도록 설계할 경우, 뉴스나 커뮤니티의 주관적 의견으로 인한 편향을 줄일 수 있다.

또한 RAG는 검색된 문서를 함께 제공하기 때문에, 답변의 출처를 명확히 제시할 수 있다. 이를 통해 사용자는 답변의 근거를 직접 확인할 수 있으며, 결과의 투명성과 신뢰성을 높일 수 있다.

---

4. 청킹(Chunking)

RAG 시스템에서 문서를 그대로 사용하는 것은 효율적이지 않다. 대부분의 문서는 길이가 길기 때문에, 검색 정확도를 높이기 위해 문서를 적절한 크기의 조각으로 분할하는 과정이 필요하다. 이 과정을 청킹이라고 한다.

청킹 방식에는 다음과 같은 방법들이 있다.

첫째, **고정 길이 청킹**은 문서를 일정한 토큰 또는 문자 수 단위로 분할하는 방식이다. 구현이 간단하고 처리 속도가 빠르다는 장점이 있지만, 문장의 의미나 문맥이 중간에 끊어질 수 있다는 단점이 있다. 이를 보완하기 위해 일반적으로 overlap(겹침)을 적용하여 일부 내용을 중복시킨다.

둘째, **구조 기반 청킹**은 문서의 제목, 소제목, 문단, 표와 같은 구조를 기준으로 분할하는 방식이다. 재무제표, 보고서, 공시 문서와 같이 명확한 구조를 가진 문서에 특히 적합하며, 의미 단위가 잘 유지된다는 장점이 있다.

셋째, **세맨틱 청킹**은 문장의 의미가 변화하는 지점을 기준으로 문서를 분할하는 방식이다. 가장 자연스럽고 검색 품질이 높지만, 구현 난도가 높고 추가적인 계산 비용이 발생한다는 단점이 있다.

---

5. 임베딩과 벡터 검색

임베딩은 텍스트 데이터를 고정 차원의 벡터로 변환하는 과정이다. 임베딩된 벡터는 텍스트의 의미를 수치적으로 표현하며, 의미가 유사한 문장이나 문서는 벡터 공간 상에서 가까운 위치에 배치된다.

RAG 시스템에서는 문서 청킹 이후 각 chunk를 임베딩하여 벡터 데이터베이스에 저장한다. 이후 사용자의 질문도 동일한 임베딩 모델을 통해 벡터로 변환한 뒤, 질문 벡터와 문서 벡터 간의 유사도를 계산하여 관련성이 높은 문서를 검색한다.

이러한 벡터 검색 방식은 단순 키워드 검색과 달리 의미 기반 검색이 가능하다는 장점이 있으며, 자연어 질문에 보다 정확하게 대응할 수 있다.

---

6. NLP 기술과 RAG의 관계

자연어 처리(NLP)는 컴퓨터가 인간의 언어를 이해하고 처리하도록 하는 기술 전반을 의미한다. 텍스트 전처리, 임베딩, 문서 분류, 요약, 질의응답, 언어 생성 등 다양한 기술이 NLP 범주에 포함된다.

RAG는 이러한 NLP 기술을 조합하여 구성된 시스템 아키텍처이다. 즉, RAG 자체가 새로운 NLP 기술이라기보다는, 기존의 NLP 기술(임베딩, 의미 검색, 언어 생성)을 결합하여 외부 문서를 기반으로 답변을 생성하는 응용 구조라고 볼 수 있다.

---

7. 취업 도메인 데이터셋 구성

본 프로젝트에서는 취업 도메인을 대상으로 RAG 기반 챗봇을 구현한다. 이를 위해 다음과 같은 데이터 소스를 고려하였다.

우선, **사람인 API**는 공식적으로 제공되는 채용 공고 데이터로, 비교적 안정적이고 구조화된 데이터를 제공한다. API를 통해 직무, 요구 기술, 근무 조건 등의 정보를 수집할 수 있다.

또한 **Kaggle의 Job Description 데이터셋**은 다양한 직무 설명 데이터를 포함하고 있어 초기 실험 및 모델 테스트에 활용하기 적합하다.

---

8. 참고 자료

- RAG 개념 참고 블로그: https://dwin.tistory.com/172
- 데이터셋 : https://www.kaggle.com/search?q=job+descriptions

- 사람인 api : https://oapi.saramin.co.kr/