

생성형 AI

Day 19

생성형AI III



목차

1. Large Language Model
2. ChatGPT
3. Retrieve Augmented Generation?
4. RAG 아키텍처
5. RAG 프레임워크
6. Langchain
7. LlamaIndex
8. RAG의 최신 동향
9. 실습과제



Large Language Model (LLM)

LLM?

- 대규모 텍스트 데이터를 학습하여 자연어 이해와 생성을 할 수 있는 인공지능 언어모델
- 텍스트 데이터를 이해하고 생성하여 다양한 자연어 처리(NLP) 작업을 수행
- 최근 LLM은 트랜스포머 아키텍처를 이용

LLM의 학습방법

사전 학습 (Pre-training)

- 대규모 텍스트 코퍼스를 사용하여 모델이 언어의 일반적인 패턴과 구조를 학습하는 과정
- 모델이 언어의 기초적인 이해를 습득하게 함

미세 조정 (Fine-tuning)

- 전 학습된 모델을 특정 작업에 맞게 추가 학습하는 과정
- 특정 작업(예: 감정 분석, 번역 등)에서 모델의 성능을 최적화

Large Language Model (LLM)

언어모델?

- 주어진 텍스트의 다음 단어를 예측하는 모델
- 텍스트 데이터의 확률 분포를 학습하여 다음 단어를 정확하게 예측

언어모델링 방법에 따른 종류

자가회귀 모델 (Autoregressive Model)

- 이전 단어들을 기반으로 다음 단어를 순차적으로 예측하는 모델
- GPT 시리즈 (GPT-1, GPT-2, GPT-3)
- 텍스트 생성, 번역, 대화형 AI 등

마스크드 언어 모델 (Masked Language Model)

- 문장의 일부 단어를 마스킹(masking)하고, 마스킹된 단어를 예측하는 모델
- BERT (Bidirectional Encoder Representations from Transformers)
- 텍스트 분류, 질문 응답, 자연어 추론 등

Large Language Model (LLM)

언어 모델은 어떻게 평가하지?

퍼플렉시티 (Perplexity)

- 모델이 주어진 텍스트를 얼마나 잘 예측하는지를 나타내는 지표
- 예측 확률의 역수의 기하평균

BLEU (Bilingual Evaluation Understudy)

- 생성된 텍스트와 참조 텍스트 간의 유사성을 측정하는 지표
- 기계 번역, 텍스트 요약 등

Large Language Model (LLM)



The Open Ko-LLM Leaderboard objectively evaluates the performance of Korean Large Language Model (LLM).

When you submit a model on the "Submit here!" page, it is automatically evaluated. The GPU used for evaluation is operated with the support of [KT](#). The data used for evaluation consists of datasets to assess reasoning, language understanding, hallucination, and commonsense. The evaluation dataset is exclusively private and only available for evaluation process. More detailed information about the benchmark dataset is provided on the "About" page.

This leaderboard is co-hosted by [Upstage](#), and [NIA](#) that provides various Korean Data Sets through [AI-Hub](#), and operated by [Upstage](#).

The screenshot shows the user interface of the Open Ko-LLM LeaderBoard. At the top, there are tabs for "LLM Benchmark", "Metrics through time", "About", and "Submit here!". Below these are sections for "Model types", "Precision", and "Model sizes (in billions of parameters)". The "Model types" section includes checkboxes for "pretrained" (green), "instruction-tuned" (red), "RL-tuned" (blue), and a question mark. The "Precision" section includes checkboxes for "float16" and a question mark. The "Model sizes" section includes checkboxes for "Unknown", "0~3B", "3~7B", "7~13B", "13~35B", "35~60B", and "60B+". On the left, there is a "Select columns to show" dropdown menu with checkboxes for various metrics like Average, Ko-ARC, Ko-HellaSwag, etc. At the bottom, there are checkboxes for "Show private/deleted models", "Show merges", and "Show flagged models". The main area shows a table header with columns for Model, Average, Ko-ARC, Ko-HellaSwag, Ko-MMLU, Ko-TruthfulQA, Ko-Winogrande, Ko-GSM8k, Ko-CommonGen V2, and Ko-EQ. The "Average" column is currently sorted in descending order.

ChatGPT

ChatGPT?

- OpenAI에서 개발한 대규모 언어 모델 GPT 시리즈를 기반으로 한 대화형 인공지능
- 자연스러운 대화를 통해 사용자의 질문에 답변하고, 다양한 언어 처리 작업을 수행

실무에서도 많이 쓸까?

- cs: 자동화된 고객 응대, 상담원 지원
- content: 마케팅 콘텐츠, 블로그 및 기사 작성
- edu: 외국어 번역, 문법 교정
- business: 회의록 작성, 실시간 회의 요약, 리포트 작성
- computer: 코드 생성, 버그 수정

정말?

- 데이터 프라이버시 및 보안 문제
- 비용 문제
- 정확성 및 신뢰성 부족

ChatGPT

데이터 프라이버시 및 보안 문제

- 대화의 맥락을 기억하는 ChatGPT
- 이전 고객의 정보를 기억하는 경우
- 해당 고객의 정보를 Context상에 갖고 있어서 바로 처리 해버린 경우

비용 문제

- BM에 비해 ChatGPT 토큰이 과하게 소비되는 경우
- 어뷰징으로 인한 비용 상승

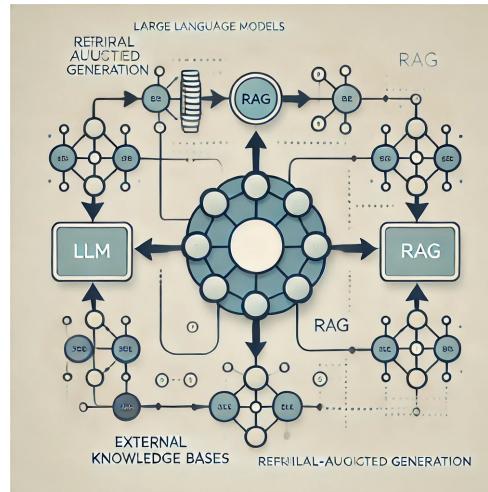
정확성 및 신뢰성 부족

- 의료 분야에서 잘못된 조언을 내리는 경우
- 온라인 커머스 분야에서 다양한 상품과 매겨져있는 가격 설정을 혼동하는 경우

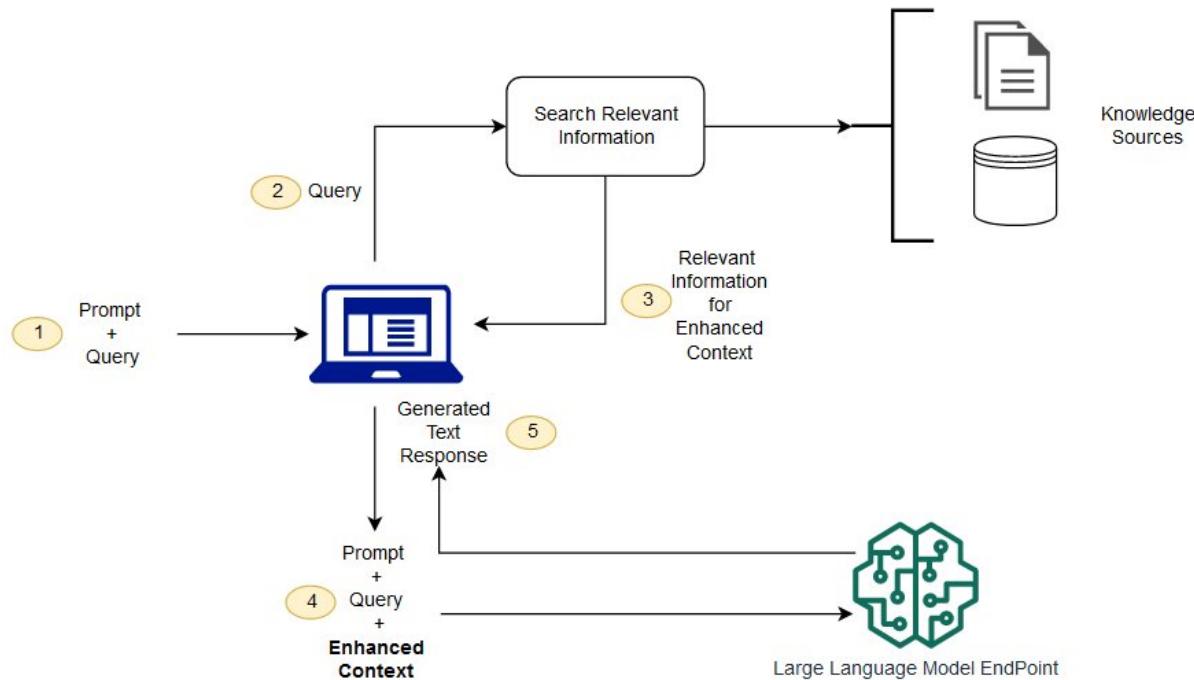
Retrieve Augmented Generation (RAG)

RAG?

- Retrieve-Augmented Generation (RAG)는 정보 검색(Retrieve)과 텍스트 생성(Generate)을 결합한 하이브리드 언어 모델 아키텍처
- 특정 질의에 대해 외부 지식 베이스에서 관련 정보를 검색한 후, 이 정보를 기반으로 텍스트를 생성하는 방식으로 작동
- 특히 정보가 풍부한 응답을 생성하는 데 유리



RAG 아키텍처



RAG 아키텍처

RAG의 주요 구조

Retrieve 단계

- 입력 질의(query)와 관련된 문서를 외부 지식 베이스에서 검색
- 쿼리와 문서의 임베딩을 생성하고, 임베딩 간의 유사성을 계산하여 관련 문서를 검색
- 임베딩 알고리즘을 통해 수행 가능

Augmented 단계

- 검색된 문서를 기반으로 입력 질의(query) 강화

Generate 단계

- 입력 질의(query) 또는 검색된 문서등을 기반으로 자연스러운 텍스트를 생성

RAG 아키텍처

RAG의 작동 원리

1. 쿼리 임베딩 생성

- 입력된 질의를 임베딩 벡터로 변환

2. 문서 검색

- 쿼리 임베딩을 사용하여 외부 지식 베이스에서 관련 문서들을 검색

3. 문서 임베딩 생성

- 검색된 각 문서를 임베딩 벡터로 변환

4. 문서-쿼리 결합

- 검색된 문서와 쿼리 임베딩을 결합하여 텍스트 생성을 위한 입력으로 사용

5. 텍스트 생성

- 결합된 임베딩을 입력으로 받아 디코더가 새로운 텍스트를 생성

RAG 아키텍처

RAG의 장점

- 외부 지식 베이스를 활용하여 더 풍부하고 정확한 정보를 제공
- 다양한 데이터 소스를 결합하여 더 광범위한 지식을 응답에 포함할
- 단순한 생성 모델보다 더 정확하고 정보가 풍부한 응답을 생성
- 내부 LLM모델로 구성하는 경우 보안, 프라이버시 위협에서 어느정도 벗어날 수 있음
- 파인튜닝보다 효율적으로 특화 도메인에 대응 가능

RAG의 단점

- 검색과 생성을 결합하는 과정이 복잡하며, 모델 학습 및 튜닝이 어려울 수 있
- 임베딩 생성, 문서 검색, 텍스트 생성 등 모든 단계에서 높은 연산 자원이 필요
- 외부 지식 베이스의 품질과 최신성에 크게 의존

RAG 아키텍처

RAG 예시

-질의 응답 시스템 (Question Answering)

: 사용자가 "2022년 FIFA 월드컵 우승 팀은?"이라는 질문을 입력하면, RAG 모델은 최신 뉴스 기사나 공식 발표 자료를 검색하여 정확한 답변을 제공

-헬스케어 챗봇

: 사용자가 "고혈압의 일반적인 증상은 무엇인가요?"라고 질문하면, 챗봇은 최신 의학 논문과 건강 정보 웹사이트에서 관련 정보를 검색하여 정확하고 신뢰할 수 있는 답변을 제공합니다. 이는 환자들이 더 신속하고 정확한 건강 정보를 얻는데 도움이 됩니다

- 학술 논문 검색

: 연구자가 "최근 딥러닝을 활용한 암 진단 연구"에 대해 검색하면, RAG 모델은 최신 학술 논문을 검색하고 주요 내용을 요약하여 제공함으로써 연구자가 필요한 정보를 신속하게 파악할 수 있도록 돕습니다

-법률 상담 챗봇

: 사용자가 "임대 계약서 작성 시 주의사항은?"이라고 질문하면, 챗봇은 관련 법률 문서와 판례를 검색하여 사용자가 주의해야 할 사항들을 요약하여 제공합니다. 이는 초기 상담 비용을 절감하고, 변호사들이 더 복잡한 문제에 집중할 수 있도록 합니다

RAG 프레임워크

직접 구현

임베딩

- TF-IDF, 임베딩 모델

정보저장 및 검색

- RDBMS, NoSQL, VectorDB, ...

LLM

- Llama, ChatGPT, Phi, ...



LangChain



LlamaIndex

Langchain

- 대규모 언어 모델(LLM)을 활용한 복잡한 언어 처리 파이프라인을 구축하기 위한 프레임워크
- 다양한 언어 모델과 검색 기법을 결합하여 효율적이고 확장 가능한 자연어 처리(NLP) 시스템을 구축하는 데 중점

LlamaIndex

- 대규모 언어 모델을 기반으로 한 검색 및 생성 작업을 지원하는 프레임워크
- 프레임워크는 대규모 데이터셋을 효율적으로 인덱싱하고, 고성능 검색 기능을 제공하는 데 중점

RAG 프레임워크

Langchain

- 검색 및 생성을 독립적인 모듈로 구현할 수 있는 구조를 제공
- 따라서 각 구성 요소를 독립적으로 개발, 테스트, 배포 가능
- 다양한 데이터 소스와 검색 방법을 쉽게 통합하는 기능 제공
- 데이터를 처리하고 변환하는 파이프라인을 유연하게 구성 가능
- 사용자가 특정 요구 사항에 맞게 데이터를 전처리하고, 검색 및 생성 단계를 조정 가능
- 통합할 수 있는 환경을 제공합니다. 이는 언어 모델의 학습, 평가, 배포를 위한 도구와 기능을 포함

구조

1. 데이터 소스: 외부 데이터베이스, 웹사이트, 문서 저장소 등 다양한 데이터 소스에서 정보를 검색
2. 검색 모듈: TF-IDF, BM25 등의 검색 알고리즘을 사용해 입력 쿼리와 관련된 문서를 검색하는 기능을 담당
3. 생성 모듈: 언어모델을 사용해 검색된 문서를 바탕으로 자연스러운 텍스트를 생성하는 기능을 담당
4. 파이프라인 관리: 데이터를 전처리하고, 검색 및 생성 단계를 조정하여 최종 응답을 생성

RAG 프레임워크

LlamaIndex

- The leading data framework for building LLM applications
- 대규모 데이터셋을 효율적으로 인덱싱하고 고성능 검색 기능을 제공하는 데 중점
- 다양한 검색 알고리즘과 언어 모델을 쉽게 통합
- 문서와 쿼리의 임베딩을 생성하고, 유사성을 계산하여 관련 문서를 검색

구조

1. 데이터 인덱싱: 대규모 데이터셋을 효율적으로 인덱싱하여 빠른 검색을 지원하고 그 과정에서 데이터의 구조와 내용을 분석하여 효율적인 검색 가능
2. 쿼리 처리 및 검색: 사용자가 입력한 쿼리를 임베딩 벡터로 변환하고 인덱싱 된 문서를 임베딩 벡터로 변환한 다음 유사도 계산
3. 텍스트 생성: 검색된 문서를 바탕으로 생성에 필요한 입력을 준비
4. 파이프라인 관리: 인덱싱, 검색, 생성을 통합하여 효율적인 작업 흐름을 관리

RAG 프레임워크

Langchain VS LlamaIndex

- Langchain

- 모듈화된 구성 요소: 검색 및 생성 단계를 독립적으로 구현하고 조합할 수 있음
- 확장성: 다양한 데이터 소스와 검색 방법을 쉽게 통합 가능
- 유연한 파이프라인: 데이터를 처리하고 변환하는 파이프라인을 유연하게 구성

- LlamaIndex

- 효율적인 인덱싱: 대규모 데이터셋을 구조화된 형태로 저장하여 빠른 검색을 지원
- 강력한 검색 기능: Dense Retrieval 기법을 사용하여 높은 정확도의 검색 결과 제공
- 통합된 생성 기능: 검색된 정보를 바탕으로 자연스러운 텍스트 생성

→ Langchain은 다양한 응용 분야에서 유연하게 활용할 수 있고, LlamaIndex는 대규모 데이터셋 기반의 고성능 검색 작업에 적합

RAG 최신동향

SELF-RAG

- RAG 시스템의 생성 품질과 사실성을 개선하기 위해 SELF-RAG 프레임워크를 제안
- 필요할 때마다 검색하고, 생성한 내용을 자체 반성하여 평가하는 reflection 토큰을 사용
- reflection 토큰을 통해 검색 빈도를 조정하고 사용자 선호도에 맞게 모델 행동 학습

Adaptive RAG

- 쿼리 복잡도에 따라 가장 적절한 전략을 동적으로 선택하는 Adaptive-RAG 프레임워크 개발
- 다양한 복잡도의 쿼리를 처리하기 위해 적응형 RAG 시스템을 제안
- 쿼리 복잡도에 따라 효율성과 정확성을 균형 있게 향상

->그외에 새로운 연구/논문들이 계속 나오는 중

이론 코드

colab: [https://colab.research.google.com/drive/1ljn4_dpaLQc7kBTtkQAdYtvDdN4MHMo6?
usp=sharing](https://colab.research.google.com/drive/1ljn4_dpaLQc7kBTtkQAdYtvDdN4MHMo6?usp=sharing)

실습 과제

1. Langchain을 통해 RAG 구성해보기
2. LlamaIndex를 통해 RAG 구성해보기
3. 프레임워크를 사용하지 않고 RAG를 구성해보기

실습 진행