

# CH02 서기

프롬프트 엔지니어링이란?

: GPT3 때부터 연구되어온 대상, 학생들에게도 인기 많아짐

개발 중심에 서게 된 이유

- 같은 모델이더라도 프롬프트를 어떻게 만들게 되느냐에 따라 시스템 성능과 만족도에 차이가 커짐
- 잘못된 프롬프트는 모델을 비효율적이게 만들어줌 → 더욱 중요해짐

프롬프트: LLM 기반 성능과 동작을 완벽하게 통제하기 위한 도구

프롬프트 필요성

- 특정 문맥에서 작동하도록 해줌

RAG 프롬프트 구조

- 지시사항
- 질문
- 문맥

## I. PromptTemplate 기본과 응용

실습 → colab을 통해...

대표적인 프롬프트 사례: RAG 프롬프트

- .from\_template() 메소드 → template 형태로 만들어줌
- PromptTemplate() 생성자 (input\_variables 명시) → prompt 객체 생성
  - ChatPromptTemplate: 일반 프롬프트랑은 다르게 역할을 부여해 메시지 리스트를 구성함 / 메시지는 튜플 형식으로 구성됨; (role, message)
  - MessagePlaceholder: 긴 대화 목록이나 컨텍스트를 변수 형태로 받아와 프롬프트의 원하는 위치에 통째로 삽입 가능
- partial\_variables → 프롬프트의 유연성을 높이는 기능 / 일부 변수에 미리 값을 할당해 줌
- partial → 함수를 부분적으로 사용하고 싶을 때

## **II. FewShotPromptTemplate**

: 모델이 예시를 보고 학습하게 함

- Example Selector:

- 예시를 선정하는 기준:

- 사용자의 현재 질문과 가장 유사한 질문만 포함함 (토큰 제한 때문)
    - 임베딩에 저장
    - Chroma에서 지정한 예시 수만큼 유사 질문을 뽑아냄

## **III. LangChain Hub 활용법**

: LangChain Hub에서 프롬프트를 받아서 실행하는 예제

- repo id, commit id로 받아올 수 있음

Hub에 푸시할 때 LangSmith가 필요한 이유

: LangSmith는 LangChain Hub 기반 인프라 역할을 하기 때문

개인화된 프롬프트(Hub에 업로드)

: hub.push때 '/'가 없을 때 업로드가 됨

Stuff Documents

- 내용을 한 프롬프트 안에 뭉텅이로 넣음 (Chunk 형식으로)
- 토큰 수 제한 때문에 너무 길면 제한됨

Map Prompt

- 문서를 작은 덩어리로 나눠서, 각 프롬프트 실행

Reduce Prompt (축소 프롬프트)

- Map 단계에서 나온 결과물을 하나의 프롬프트로 만들고, 최종 통합 작업을 요청함
- 토큰 제한에 부딪힐 수 있음

Metadata Tagger

- 텍스트에서 메타데이터를 추출

- 형식을 정리해서, 어떤 걸 추출해달라 → 거기에 포함된 내용을 추출해줌 (비정형 → 정형 변환)

### Chain of Density

- 기본 요약
- 밀도 증가
- 한 번에 완성된 요약이 아닌, 점진적으로 정보를 추가해 가며 완성함

### RAG 문서 프롬프트

- 찾은 정보를 바탕으로 완성해라!