Ch 02. Prompt

상태	진행 중

CH 02. 프롬프트(Prompt)

Introduction

프롬프트 단계란 무엇인가?

검색기에서 검색된 문서들을 바탕으로 **언어 모델이 사용할 질문이나 명령을 생성하는 과정**

→ **검색된 정보를 바탕**으로 최종 사용자의 질문에 가장 잘 대응할 수 있는 응답을 생성하기 위해 필수적인 단계

→ 그렇다면 사전에 문서는 어떻게 준비를 할 수 있는가?

💡 **프롬프트 단계의 전제**

- 프롬프트 단계는 **이미 "검색기(Retriever)"가 관련 문서를 찾아놓은 이후** 수행된다.

→ 즉, 이 시점의 **"문서(document)"**는 **사전에 벡터화되어 저장된 데이터(벡터 스토어)**에서 검색된 결과물 (우리가 흔히 아는 일반적인 "글"이나 "논문"을 뜻하는 게 아니다.)

  - 각 문서는 모델이 참고할 수 있도록 **page_content**와 **metadata**를 포함한 객체로 관리

→ 프롬프트는 새로운 정보를 수집하는 단계가 아니라, **기존에 확보된 정보를 모델이 이해할 수 있도록 구성하는 단계**

- 그렇다면 기존에 확보된 정보는 무엇을 의미하는가?

→ 기존에 정보 확보를 위해 우리는 **도메인 범위나 사용 목적에 따라 필요한 데이터만 선택적으로 구축한다**.
→ 즉, **임베딩(embedding) 생성**과 **벡터 스토어** 저장에는 비용이 발생하므로,
→ "자주 쓰이는 정보만 미리 임베딩 → 나머지는 점진적 추가" 전략

- 그렇다면 우리가 쓰는 ChatGPT는 이러한 데이터 수집을 어떻게 하는가?
  → 애초에 ChatGPT는 RAG 기반이 아닌 모델은 이미 내부 학습 데이터가 존재하고 모델의 지식을 어떻게 활용할 것인가로 간다.

구분	RAG 기반 LangChain	ChatGPT(범용 모델)
정보 원천	외부 벡터 스토어(사전 구축)	내부 학습 데이터
프롬프트 목적	검색된 문서 기반으로 답변 유도	모델의 지식 활용 방향 설정
정보 갱신성	최신 데이터 반영 가능	학습 시점 이후 정보 반영 어려움

→ 그러면 **ChatGPT**와 **DeepSeek의 사전 규모**는 어떨까? (출처: ChatGPT)

모델명	파라미터 수 (Parameters)	학습 토큰 수 (Training tokens)	맥락 길이(Context window)
GPT-4	약 **1.8 조(1.8 trillion)** 개 추정 (Exploding Topics)	약 **13 조(tokens)** 추정 (seifeur.com)	최대 **32K 토큰 이상** 일부 버전에서 (originality.ai)
DeepSeek-V3	약 **671 억(671 billion)** 개 파라미터 (huggingface.co)	약 **14.8 조 (tokens)** 추정 (Epoch AI)	최대 **128K 토큰** 지원 모델 존재 (huggingface.co)

프롬프트는 문맥(Context) 설정 + 정보 통합 + 응답 품질 향상 효과가 있다.

1. **문맥(Context) 설정**: 프롬프트는 언어 모델이 특정 문맥에서 작동하도록 설정하는 역할
2. **정보 통합**: 프롬프트 단계에서 다양한 문서 정보를 통합하고, 모델이 이를 효율적으로 활용할 수 있는 형식으로 조정하는 역할
3. **응답 품질 향상**: 모델이 보다 정확하고 유용한 정보를 제공하게 돕는 역할

RAG (Retrieval-Augmented Generation) 프롬프트 구조

> RAG란 외부 지식 베이스에서 관련 정보를 검색하여 활용하도록 하는 기술

1. 지시사항(Instruction)
2. 질문(사용자 입력 질문)
3. 문맥(검색된 정보)

```
당신은 질문-답변(Question-Answer) Task 를 수행하는 AI 어시스턴트 입니다.
검색된 문맥(context)를 사용하여 질문(question)에 답하세요.
만약, 문맥(context) 으로부터 답을 찾을 수 없다면 '모른다' 고 말하세요.
한국어로 대답하세요.

#Question:
{이곳에 사용자가 입력한 질문이 삽입됩니다}

#Context:
{이곳에 검색된 정보가 삽입됩니다}
```

01. Prompt

Prompt 생성 방식

1. from_template() 메소드를 사용하여 PromptTemplate 객체 생성

2. PromptTemplate 객체 생성과 동시에 prompt 생성

> 추가 유효성 검사를 위해 `input_variables` 를 명시적으로 지정함.

> → 인스턴스화 중에 템플릿 문자열에 있는 변수와 비교하여 불일치하는 경우 예외를 발생시킨다.
>
> → 더 안전한 Prompt 생성 가능

`partial_variables` : 함수를 부분적으로 사용하자

→ 항상 공통된 방식으로 가져오고 싶은 변수가 있는 경우 사용한다.

→ 공통된 방식이 무엇일까? (일정한 구조?) 예: **날짜나 시간**

> 항상 현재 날짜가 표시되기를 원하는 프롬프트가 있다고 가정해 보겠습니다. 프롬프트에 하드 코딩할 수도 없고, 다른 입력 변수와 함께 전달하는 것도 번거롭습니다. 이 경우 항상 현재 **날짜를 반환하는 함수** 를 사용하여 프롬프트를 부분적으로 변경할 수 있으면 매우 편리합니다.

`ChatPromptTemplate` : 대화목록을 프롬프트로 주입하자

→ 메시지는 **튜플(tuple) 형식**, (`role` , `message`) 로 구성하여 리스트로 생성가능

role

- `"system"` : 시스템 설정 메시지 입니다. 주로 전역설정과 관련된 프롬프트입니다.
- `"human"` : 사용자 입력 메시지 입니다.
- `"ai"` : AI 의 답변 메시지입니다.

`MessagePlaceholder` : 포맷 과정에서 렌더링할 메세지를 제어하자

- 렌더링할 메세지를 제어한다?

  → 메시지 프롬프트 템플릿에 어떤 역할을 사용해야 할지 확실하지 않거나 서식 지정 중에 메시지 목록을 삽입하려는 경우

```
from langchain_core.output_parsers import StrOutputParser
from langchain_core.prompts import ChatPromptTemplate, MessagesPl
aceholder

chat_prompt = ChatPromptTemplate.from_messages(
    [
        (
            "system",
            "당신은 요약 전문 AI 어시스턴트입니다. 당신의 임무는 주요 키워드로 대
화를 요약하는 것입니다.",
        ),
```

```
        MessagesPlaceholder(variable_name="conversation"),
        ("human", "지금까지의 대화를 {word_count} 단어로 요약합니다."),
    ]
)
chat_prompt
```

02. 퓨샷 프롬프트 (FewShotPromptTemplate)

왜 `FewShotPromptTemplate` 를 사용해야하는가?

→ 예시를 더 쉽게 간단하게 넣을 수 있다

항목	`PromptTemplate`	`FewShotPromptTemplate`
역할	단일 프롬프트 생성	여러 예시를 포함한 프롬프트 자동 생성
입력 구조	`{question}` , `{answer}`	`examples` 리스트 + `example_prompt` + `suffix`
예시 삽입	직접 수동 반복	자동으로 examples를 순회하여 삽입
사용 목적	1개의 입력만 테스트할 때	few-shot 학습 형태로 문맥을 제공할 때

💡 `**` 연산자의 역할
→ 딕셔너리 언패킹 연산자

항목	설명
의미	딕셔너리의 key-value를 자동으로 함수 인자로 전달
사용 목적	`.format()` 이나 함수 호출 시, 변수를 일일이 나열하지 않아도 됨
결과	템플릿 내 `{question}` , `{answer}` 가 자동으로 채워짐

- `examples[0]` : 예시 딕셔너리 구조

```
{
   "question": "스티브 잡스와 아인슈타인 중 누가 더 오래 살았나요?",
   "answer": "이 질문에 추가 질문이 필요한가요: 예. ..."
}
```

- 실제 언패킹 효과

```
example_prompt.format(
   question="스티브 잡스와 아인슈타인 중 누가 더 오래 살았나요?",
   answer="이 질문에 추가 질문이 필요한가요: 예. ..."
)
```