

출력파서

: LLM의 응답을 원하는 형식으로 구조화하거나 가공하는 역할을 하는 컴포넌트

- 모델이 생성한 텍스트를 우리가 원하는 데이터 형태(JSON, list, SQL, HTML 등) 로 해석하고 변환하는 단계

1. PydanticOutputParser

: 언어 모델의 출력을 구조화된 정보로 변환(대부분의 파서)

➔ 단순 텍스트 대신 사용자가 필요로 하는 정보를 체계적으로 전달

Get_format_instructions() : 모델이 출력할 정보 형식 정의

Parse() : 출력을 받아들이며 특정 구조로 분석, 변환

2개의 메소드 필수

Description -> 프롬프트 정의 -> chain 생성, 실행 -> parser로 결과 파싱, 객체 변환

2. CommaSeparatedListOutputParser

: 쉼표로 구분된 항목 목록 변환시 사용

-> csv형식을 출력할때 사용하는가?

3. StructuredOutputParser

: 답변을 dict형식으로 구성, key value 쌍으로 갖는 필드 반환시

Pydantic파서 동작하지 않을 경우 사용

4. JsonOutputParser

: 사용자가 원하는 Json 스키마 지정, 조회, 출력 도움

출력구조 정의 후 메소드 사용해서 파서 설정, 지시사항 주입 -> json형태로 출력

➔ pydantic없이 사용하기 사례가 나와있는데 이때 3번과 4번의 차이

5. PandasDataFrameOutputParser

: python프로그래밍 언어에서 자주 사용, 구조화된 데이터 핸들을 위한 도구세트 제공, 데이터 작업에 활용

- 사용자가 임의의 Pandas DataFrame을 지정하고 해당 DataFrame에서 데이터를 추출하여 형식화된 사전 형태로 데이터를 조회할 수 있는 LLM을 요청

Format_parser_output 파서 출력을 사전 형식으로 변환 -> titanic.csv dataframe로드, 변수 할당

6. DatetimeOutputParser

: LLM의 출력을 datetime 형식으로 파싱

*형식 코드 참고

7. EnumOutputParser

: Enum 출력 파서

Enum이란?

- 서로 연관된 상수값들을 하나의 이름 아래에 모아두는 자료형입니다.
"의미 있는 이름을 가진 상수들의 집합"

Langchain.output_parsers.enum import EnumOutputParser

프롬프트에 사람의 정보와 파싱 지침 포함(예시) -> 핵심 정보를 모아둔다.

8. OutputFixingParser

: 출력 파싱과정에서 발생할 수 있는 오류 자동 수정

위의 7까지 파싱 기능들을 래핑하고, 오류 점검 및 수정