**TA's 솔루션 해설**

* <https://gist.github.com/ICE0208/a63490ad7163935be3dbfe66552e8237>

1. ChatOpenAI 모델 초기화 및 설정
   * ChatOpenAI 생성자를 이용하여 OpenAI의 GPT 모델을 초기화합니다.
   * 모델 초기화 시 몇몇 설정을 커스텀할 수 있습니다.
   * 해설 코드에서는 temperature 를 추가로 설정하였습니다.
2. 메모리 초기화
   * ConversationBufferMemory를 사용하여 이전 대화 기록을 저장하고 관리합니다.
   * return\_messages=True로 설정하여 채팅 형식으로 기록을 로드할 수 있도록 합니다.
3. 예제 생성
   * 세 가지 영화에 대한 예제 데이터를 생성하였습니다.
   * 각 예제는 영화 제목과 그에 대한 3개의 이모티콘 답변을 포함합니다.
   * LLM이 예제를 효과적으로 학습하도록 동일한 형식으로 작성하였습니다.
4. 프롬프트 생성
   * ChatPromptTemplate.from\_messages를 이용하여 예제 대화 메시지 템플릿을 생성합니다.
   * FewShotChatMessagePromptTemplate를 사용하여 예제 대화 메시지와 예제 데이터를 결합하고, 새로운 영화 제목에 대해 이모티콘으로 답변하는 프롬프트를 생성합니다.
   * 최종 프롬프트는 시스템 메시지, 예제 프롬프트(fewshot\_chat\_prompt), 대화 기록 플레이스홀더(MessagesPlaceholder), 그리고 사용자 입력 메시지로 구성됩니다.
5. 체인 설정
   * RunnablePassthrough.assign을 사용하여 메모리 로드 함수를 체인에 연결합니다.
   * final\_prompt와 llm을 연결하여 최종 체인을 생성합니다.
6. 체인 실행 함수 (invoke\_chain)
   * invoke\_chain 함수를 통해 질문을 입력받아 체인을 실행하고, 결과를 메모리에 저장합니다.
   * 저장된 대화를 기반으로 이전 질문에 대한 답변을 요청할 수 있습니다.
7. 작동 테스트
   * 정상적인 작동을 확인하기 위해 새로운 두 개의 영화에 대해 질문하고, 첫 번째로 물어본 영화에 대한 답변을 확인했습니다.
   * 영화 제목에 대한 답변이 이모티콘 3개로 구성된 것을 보아 예제가 잘 학습되었음을 확인할 수 있습니다.
   * 또한 첫 번째로 질문한 영화가 무엇인지에 대한 정확한 대답을 통해 메모리 적용이 잘 되었음을 알 수 있습니다.
8. 결론

* 이번 챌린지는 메모리가 적용된 LCEL 체인을 구현하여, LLM이 대화 기록에 대한 맥락을 이해하고 그에 맞는 적절한 답변을 할 수 있도록 하는 방법을 연습하는 것이었습니다.
* 또한, 체인에 예시 데이터를 제공함으로써 모델이 동일한 형식의 응답을 생성하도록 하는 연습도 함께 진행되었습니다. 이를 통해 지난 챌린지에서 학습한 내용을 복습할 수 있었습니다.