

CHAPTER 6. 메모리

1. ConversationBufferMemory

메시지를 저장한 다음 변수에 메시지를 추출할 수 있게 해줍니다.

사람의 입력과 AI의 답변을 pair 하게 만들어서 저장하는 역할.

예제 - ConversationBufferMemory

```
from langchain.memory import ConversationBufferMemory
```

```
memory = ConversationBufferMemory()  
print(memory)
```

save_context(inputs, outputs) 메서드를 사용하여 대화 기록을 저장할 수 있습니다.

- 이 메소드는 inputs와 outputs 두 개의 인자를 받습니다.
- inputs는 사용자의 입력을, outputs는 AI의 출력을 저장합니다.
- 이 메서드를 사용하면 대화 기록이 history 키에 저장됩니다.
- 이후 load_memory_variables 메소드를 사용하여 저장된 대화 기록을 확인할 수 있습니다.

질문과 답변이 한 턴 이 된다.

```
memory.save_context(  
    inputs={'human': '안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?'},  
    outputs={'ai': '안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해  
주시겠어요?'}  
)  
print(memory)
```

결과

```
chat_memory=InMemoryChatMessageHistory(messages=[HumanMessage(content='안녕하세요, 비대면으로 은행 계좌  
를 개설하고 싶습니다. 어떻게 시작해야 하나요?'), AIMessage(content='안녕하세요! 계좌 개설을 원하신다니  
기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?')])
```

memory 의 load_memory_variables({}) 함수는 메시지 히스토리를 반환합니다.

```
print(memory.load_memory_variables({}))
```

결과

```
{'history': 'Human: 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?\nAI:  
안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?'}
```

```
print(memory.load_memory_variables({})['history'])
```

결과

Human: 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?

AI: 안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?

```
# inputs: dictionary(key: "human" or "ai", value: 질문)
```

```
# outputs: dictionary(key: "ai" or "human", value: 답변)
```

```
memory.save_context(
```

```
    inputs={"human": "네, 신분증을 준비했습니다. 이제 무엇을 해야 하나요?"},
```

```
    outputs={
```

```
        "ai": "감사합니다. 신분증 앞뒤를 명확하게 촬영하여 업로드해 주세요. 이후 본인 인증 절차를 진행  
하겠습니다."
```

```
    },
```

```
)
```

```
memory.save_context(
```

```
    inputs={"human": "사진을 업로드했습니다. 본인 인증은 어떻게 진행되나요?"},
```

```
    outputs={
```

```
        "ai": "업로드해 주신 사진을 확인했습니다. 이제 휴대폰을 통한 본인 인증을 진행해 주세요. 문자로  
발송된 인증번호를 입력해 주시면 됩니다."
```

```
    },
```

```
)
```

```
memory.save_context(
```

```
    inputs={"human": "인증번호를 입력했습니다. 계좌 개설은 이제 어떻게 하나요?"},
```

```
    outputs={
```

```
        "ai": "본인 인증이 완료되었습니다. 이제 원하시는 계좌 종류를 선택하고 필요한 정보를 입력해 주세  
요. 예금 종류, 통화 종류 등을 선택할 수 있습니다."
```

```
    },
```

```
)
```

```
print(memory.load_memory_variables({})["history"])
```

결과

Human: 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?

AI: 안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?

Human: 네, 신분증을 준비했습니다. 이제 무엇을 해야 하나요?

AI: 감사합니다. 신분증 앞뒤를 명확하게 촬영하여 업로드해 주세요. 이후 본인 인증 절차를 진행하겠습니다.

Human: 사진을 업로드했습니다. 본인 인증은 어떻게 진행되나요?

AI: 업로드해 주신 사진을 확인했습니다. 이제 휴대폰을 통한 본인 인증을 진행해 주세요. 문자로 발송된 인증
번호를 입력해 주시면 됩니다.

Human: 인증번호를 입력했습니다. 계좌 개설은 이제 어떻게 하나요?

AI: 본인 인증이 완료되었습니다. 이제 원하시는 계좌 종류를 선택하고 필요한 정보를 입력해 주세요. 예금 종
류, 통화 종류 등을 선택할 수 있습니다.

```
memory.save_context(
    inputs={"human": "정보를 모두 입력했습니다. 다음 단계는 무엇인가요?"},
    outputs={
        "ai": "입력해 주신 정보를 확인했습니다. 계좌 개설 절차가 거의 끝났습니다. 마지막으로 이용 약관에 동의해 주시고, 계좌 개설을 최종 확인해 주세요."
    },
)
memory.save_context(
    inputs={"human": "모든 절차를 완료했습니다. 계좌가 개설된 건가요?"},
    outputs={
        "ai": "네, 계좌 개설이 완료되었습니다. 고객님의 계좌 번호와 관련 정보는 등록하신 이메일로 발송되었습니다. 추가적인 도움이 필요하시면 언제든지 문의해 주세요. 감사합니다!"
    },
)

print(memory.load_memory_variables({})["history"])
```

결과

Human: 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?

AI: 안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?

Human: 네, 신분증을 준비했습니다. 이제 무엇을 해야 하나요?

AI: 감사합니다. 신분증 앞뒤를 명확하게 촬영하여 업로드해 주세요. 이후 본인 인증 절차를 진행하겠습니다.

Human: 사진을 업로드했습니다. 본인 인증은 어떻게 진행되나요?

AI: 업로드해 주신 사진을 확인했습니다. 이제 휴대폰을 통한 본인 인증을 진행해 주세요. 문자로 발송된 인증 번호를 입력해 주시면 됩니다.

Human: 인증번호를 입력했습니다. 계좌 개설은 이제 어떻게 하나요?

AI: 본인 인증이 완료되었습니다. 이제 원하시는 계좌 종류를 선택하고 필요한 정보를 입력해 주세요. 예금 종류, 통화 종류 등을 선택할 수 있습니다.

Human: 정보를 모두 입력했습니다. 다음 단계는 무엇인가요?

AI: 입력해 주신 정보를 확인했습니다. 계좌 개설 절차가 거의 끝났습니다. 마지막으로 이용 약관에 동의해 주시고, 계좌 개설을 최종 확인해 주세요.

Human: 모든 절차를 완료했습니다. 계좌가 개설된 건가요?

AI: 네, 계좌 개설이 완료되었습니다. 고객님의 계좌 번호와 관련 정보는 등록하신 이메일로 발송되었습니다. 추가적인 도움이 필요하시면 언제든지 문의해 주세요. 감사합니다!

return_messages=True 로 설정하면 HumanMessage 와 AIMessage 객체를 반환합니다.

예제 - ConversationBufferMemory

```
from langchain.memory import ConversationBufferMemory
```

```
# return_messages=True 로 설정하면 HumanMessage 와 AIMessage 객체를 반환합니다.
```

```
memory = ConversationBufferMemory(return_messages=True)
```

```
memory.save_context(
    inputs={
        "human": "안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?"},
    outputs={
        "ai": "안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?"},
)
```

```
memory.save_context(
    inputs={"human": "네, 신분증을 준비했습니다. 이제 무엇을 해야 하나요?"},
    outputs={
        "ai": "감사합니다. 신분증 앞뒤를 명확하게 촬영하여 업로드해 주세요. 이후 본인 인증 절차를 진행하겠습니다."},
)
```

```
memory.save_context(
    inputs={"human": "사진을 업로드했습니다. 본인 인증은 어떻게 진행되나요?"},
    outputs={
        "ai": "업로드해 주신 사진을 확인했습니다. 이제 휴대폰을 통한 본인 인증을 진행해 주세요. 문자로 발송된 인증번호를 입력해 주시면 됩니다."},
)
```

```
# 리스트에 HumanMessage, AIMessage 객체가 들어있습니다.
```

```
# 결과가 객체이면 체인에 대화 내용을 넣어서 멀티 턴으로 구현할 때 입력으로 객체를 바로 넣을 수 있다.
```

```
memory.load_memory_variables({})["history"]
```

결과

```
[HumanMessage(content='안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?'),
 AIMessage(content='안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?'),
 HumanMessage(content='네, 신분증을 준비했습니다. 이제 무엇을 해야 하나요?'),
 AIMessage(content='감사합니다. 신분증 앞뒤를 명확하게 촬영하여 업로드해 주세요. 이후 본인 인증 절차를 진행하겠습니다.'),
 HumanMessage(content='사진을 업로드했습니다. 본인 인증은 어떻게 진행되나요?'),
 AIMessage(content='업로드해 주신 사진을 확인했습니다. 이제 휴대폰을 통한 본인 인증을 진행해 주세요. 문자로 발송된 인증번호를 입력해 주시면 됩니다.')]

```

예제 - chain

```
from langchain_openai import ChatOpenAI
from langchain.chains import ConversationChain
from langchain.memory import ConversationBufferMemory

from dotenv import load_dotenv
import os

load_dotenv(verbose=True)
key = os.getenv('OPENAI_API_KEY')

# LLM 모델을 생성합니다.
llm = ChatOpenAI(temperature=0, model_name="gpt-4o-mini")

# ConversationChain을 생성합니다.
conversation = ConversationChain(
    llm=llm,
    memory=ConversationBufferMemory(), # ConversationBufferMemory를 사용합니다.
)

# 대화를 시작합니다.
response = conversation.predict(
    input="안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?"
)

print(response)

# 이전 대화 내용을 불렛포인트로 정리해 달라는 요청을 보냅니다.
response = conversation.predict(
    input="이전 답변을 불렛포인트 형식으로 정리하여 알려주세요."
)

print(response)
```

실행결과

안녕하세요! 비대면으로 은행 계좌를 개설하는 것은 요즘 매우 편리한 방법입니다. 일반적으로 다음과 같은 단계로 진행할 수 있습니다:

1. ****은행 선택****: 먼저, 비대면 계좌 개설을 지원하는 은행을 선택해야 합니다. 많은 은행들이 모바일 앱이나 웹사이트를 통해 비대면 계좌 개설 서비스를 제공하고 있습니다.
2. ****필요 서류 준비****: 신분증(주민등록증, 운전면허증 등)과 같은 기본적인 신분 확인 서류가 필요합니다. 일부 은행에서는 추가적인 서류를 요구할 수 있으니, 미리 확인해보세요.
3. ****온라인 신청****: 선택한 은행의 웹사이트나 모바일 앱에 접속하여 계좌 개설 신청을 시작합니다. 일반적으로 개인 정보 입력, 신분증 사진 업로드, 계좌 종류 선택 등의 과정이 포함됩니다.
4. ****본인 인증****: 은행에서 요구하는 본인 인증 절차를 진행합니다. 이는 보통 SMS 인증, 영상 통화, 또는 생체 인식(지문, 얼굴 인식 등)을 통해 이루어집니다.
5. ****계좌 개설 완료****: 모든 절차가 완료되면, 계좌 개설이 완료되었다는 확인 메시지를 받게 됩니다. 이후에는 계좌 번호와 관련된 정보를 이메일이나 문자로 받을 수 있습니다.
6. ****카드 발급 및 인터넷 뱅킹 설정****: 필요하다면, 카드 발급 신청을 하거나 인터넷 뱅킹 서비스를 설정할 수 있습니다.

각 은행마다 세부 절차가 다를 수 있으니, 선택한 은행의 공식 웹사이트에서 구체적인 안내를 확인하는 것이 좋습니다. 추가로 궁금한 점이 있으면 언제든지 물어보세요!

물론입니다! 비대면으로 은행 계좌를 개설하는 단계는 다음과 같이 정리할 수 있습니다:

- ****은행 선택****: 비대면 계좌 개설을 지원하는 은행 선택
- ****필요 서류 준비****: 신분증(주민등록증, 운전면허증 등) 준비
- ****온라인 신청****: 은행 웹사이트나 모바일 앱에서 계좌 개설 신청
 - 개인 정보 입력
 - 신분증 사진 업로드
 - 계좌 종류 선택
- ****본인 인증****: SMS 인증, 영상 통화, 생체 인식 등을 통한 본인 인증
- ****계좌 개설 완료****: 계좌 개설 완료 확인 메시지 수신
- ****카드 발급 및 인터넷 뱅킹 설정****: 필요 시 카드 발급 신청 및 인터넷 뱅킹 서비스 설정

각 은행마다 세부 절차가 다를 수 있으니, 공식 웹사이트에서 구체적인 안내를 확인하세요! 추가 질문이 있으면 언제든지 말씀해 주세요!