



UNLEASHING THE LLM POTENTIAL WITH MEMORY

**7 TYPES OF MEMORY TO
USE WITH LLM IN
LANGCHAIN**

WHY MAKE IT REMEMBER?

MEMORY PROVIDES THE CONCEPT OF STATE IN INTERACTION

- 1) EXTRACTING MULTIPLE PIECES OF INFO**
- 2) CONTROLLING HOW MUCH MEMORY IS STORED**
- 3) SUMMARIZING THE ENTIRE CHAT**
- 4) USING MEMORY INSIDE CHAIN**
- 5) USING MEMORY INSIDE AGENT**

CHALLENGE SOLVED : PROVIDES STATE

HOW TO MAKE IT REMEMBER?

LANGCHAIN PROVIDES **7 DIFFERENT MEMORIES**

- 1) CONVERSATION BUFFER MEMORY : ENTIRE HISTORY**
- 2) CONVERSATION BUFFER WINDOW MEMORY : K PAST INTERACTION**
- 3) CONVERSATION TOKEN BUFFER MEMORY : N TOKENS OF MEMORY**
- 4) CONVERSATION ENTITY MEMORY: EXTRACTS ENTITY**
- 5) CONVERSATION KNOWLEDGE GRAPH MEMORY: RETURN GRAPH**
- 6) CONVERSATION SUMMARY MEMORY : SUMMARIZES**
- 7) CONVERSATION SUMMARY BUFFER MEMORY : SUMMARIZES THE K INTERACTION**

BEST UNDERSTOOD IN ACTION

LEARN IMPORTANT METHODS

1) LANGCHAIN MEMORY CLASSES UNDER LANGCHAIN.MEMORY MODULE

2) EACH MEMORY CLASS HAS METHODS

A) SAVE_CONTEXT

HUMANMESSAGE(CONTENT='HI',
ADDITIONAL_KWARGS={}),

B) LOAD_MEMORY_VARIABLES

AIMESSAGE(CONTENT='WHATS UP',
ADDITIONAL_KWARGS={})

C) MEMORY.STORE

D) GET_CURRENT_ENTITIES (KNOWLEDGE GRAPH MEMORY)

E) PREDICT_NEW_SUMMARY

ADVANCED:

F) CHAT_MEMORY.ADD_USER_MSG

- CUSTOMISING AI PREFIX

G) CHAT_MEMORY.ADD_AI_MSG

- CUSTOMISING HUMAN PREFIX

- CUSTOM MEMORY CLASSES INHERITING

MEMORIES TO CHAIN & AGENT

BEST WAYS TO ACCESS THE MEMORY IS THROUGH THE PROMPTS

```
PREFIX = """HAVE A CONVERSATION WITH A HUMAN, ANSWERING THE  
FOLLOWING QUESTIONS AS BEST YOU CAN. YOU HAVE ACCESS TO THE  
FOLLOWING TOOLS:"""
```

```
SUFFIX = """BEGIN!"""
```

```
{CHAT_HISTORY}  
QUESTION: {INPUT}  
{AGENT_SCRATCHPAD}"""
```

```
PROMPT = ZEROSHOTAGENT.CREATE_PROMPT(  
    TOOLS,  
    PREFIX=PREFIX,  
    SUFFIX=SUFFIX,  
    INPUT_VARIABLES=["INPUT", "CHAT_HISTORY",  
    "AGENT_SCRATCHPAD"]  
)
```

```
MEMORY = CONVERSATIONBUFFERMEMORY(MEMORY_KEY="CHAT_HISTORY")
```

```
LLM_CHAIN = LLMCHAIN(LLM=OPENAI(TEMPERATURE=0),  
    PROMPT=PROMPT)
```

```
AGENT = ZEROSHOTAGENT(LLM_CHAIN=LLM_CHAIN,  
    TOOLS=TOOLS, VERBOSE=TRUE)
```

```
AGENT_CHAIN =  
AGENTEXECUTOR.FROM_AGENT_AND_TOOLS(AGENT=AGENT,  
    TOOLS=TOOLS, VERBOSE=TRUE, MEMORY=MEMORY)
```

```
TEMPLATE = """YOU ARE A CHATBOT HAVING A  
CONVERSATION WITH A HUMAN.
```

```
{CHAT_HISTORY}  
HUMAN: {HUMAN_INPUT}  
CHATBOT: """
```

```
PROMPT = PROMPTTEMPLATE(  
    INPUT_VARIABLES=["CHAT_HISTORY",  
    "HUMAN_INPUT"],  
    TEMPLATE=TEMPLATE  
)  
MEMORY =  
CONVERSATIONBUFFERMEMORY(MEMORY_KEY="CHAT_HISTORY")
```

```
LLM_CHAIN = LLMCHAIN(  
    LLM=OPENAI(),  
    PROMPT=PROMPT,  
    VERBOSE=TRUE,  
    MEMORY=MEMORY,  
)
```

LETS HEAD TO COLAB

NOW WE ARE TALKING!!! PRACTICE

THANKS FOR WATCHING

 **LIKE**

 **SHARE**

 **SUBSCRIBE**