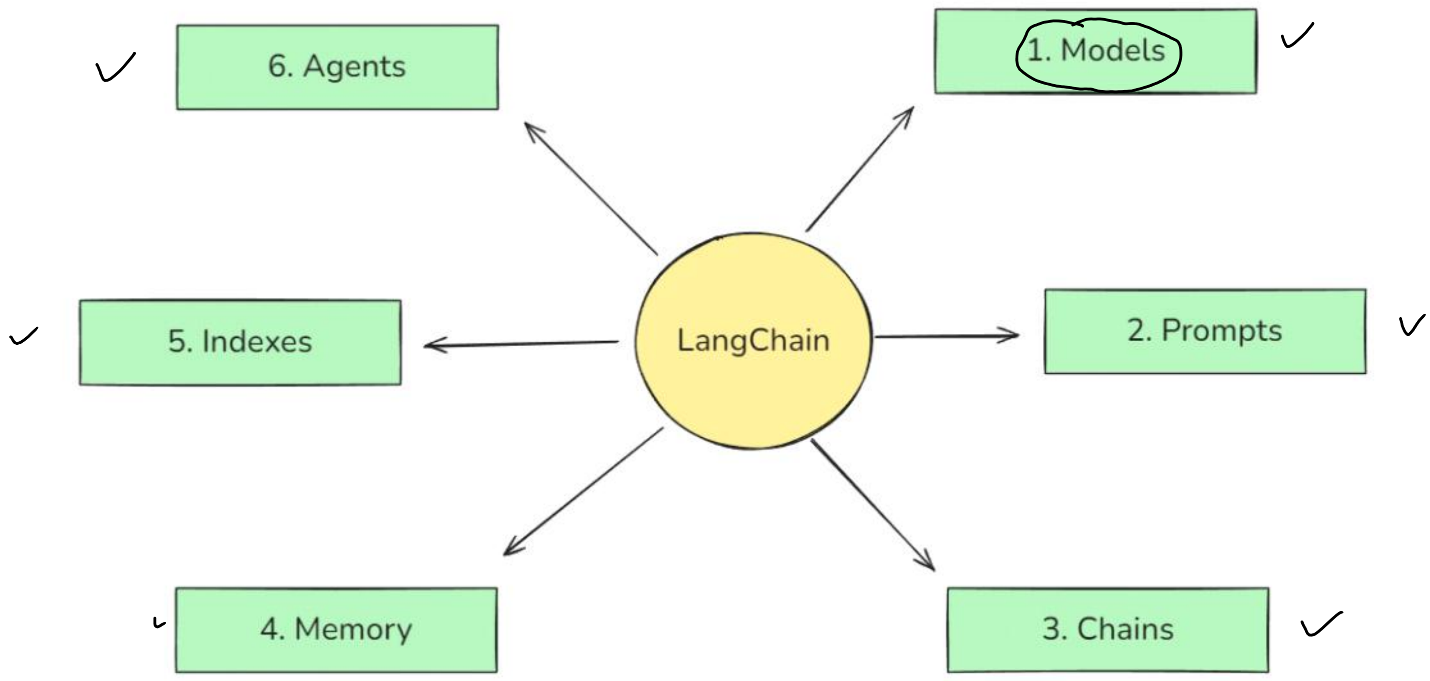
LangChain Components

23 January 2025 10:30



Models

23 January 2025 10:30

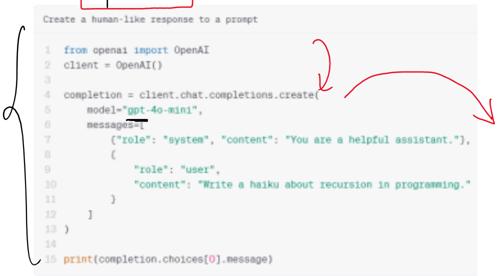
[In LangChain, “models” are the core interfaces through which you interact with AI models.

[NLP]
↳ chatbot → NLU
reply
context aware text gen
→ [LLMs] → internet use → Billions
↓
size (>100GB)

(APIs) ← server) API ← query
→ user

[implement]

OpenAI



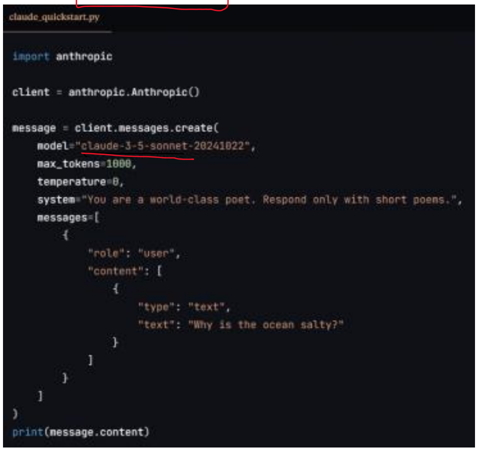
```
Create a human-like response to a prompt

from openai import OpenAI
client = OpenAI()

completion = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {
            "role": "user",
            "content": "Write a haiku about recursion in programming."
        }
    ]
)

print(completion.choices[0].message)
```

claude api



```
claude_quickstart.py

import anthropic

client = anthropic.Anthropic()

message = client.messages.create(
    model="claude-3-5-sonnet-20241022",
    max_tokens=1000,
    temperature=0,
    system="You are a world-class poet. Respond only with short poems.",
    messages=[
        {
            "role": "user",
            "content": [
                {
                    "type": "text",
                    "text": "Why is the ocean salty?"
                }
            ]
        }
    ]
)
print(message.content)
```

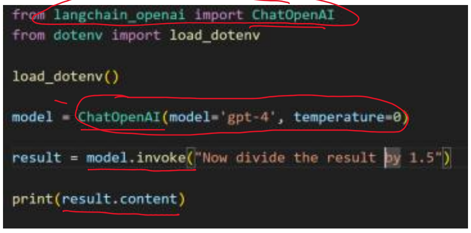
Langchain

interface → API
→

openAI

Langchain

Claude



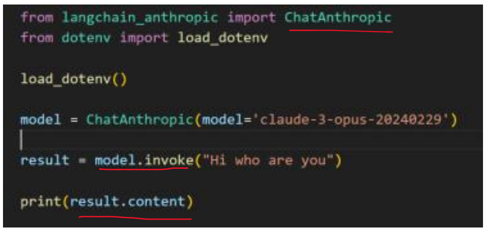
```
from langchain_openai import ChatOpenAI
from dotenv import load_dotenv

load_dotenv()

model = ChatOpenAI(model='gpt-4', temperature=0)

result = model.invoke("Now divide the result by 1.5")

print(result.content)
```



```
from langchain_anthropic import ChatAnthropic
from dotenv import load_dotenv

load_dotenv()

model = ChatAnthropic(model='claude-3-opus-20240229')

result = model.invoke("Hi who are you")

print(result.content)
```

langchains

[language model]

LLMs

text → LLMs → text

[Embedding model]

text → vector (output)

+ semantic search

Prompts

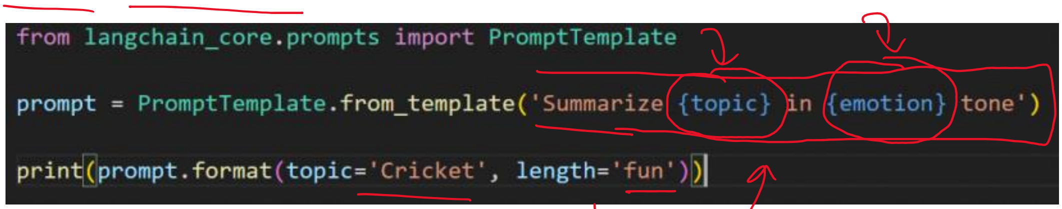
23 January 2025 10:30

LLM → input → prompt

Chatgpt → [What is Campusx]

↳ prompt → Prompt enginerin

1. Dynamic & Reusable Prompts

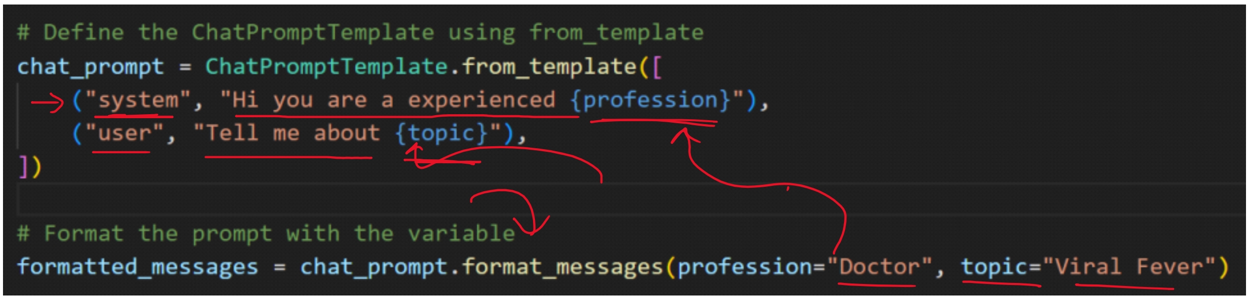


```
from langchain_core.prompts import PromptTemplate

prompt = PromptTemplate.from_template('Summarize {topic} in {emotion} tone')

print(prompt.format(topic='Cricket', length='fun'))
```

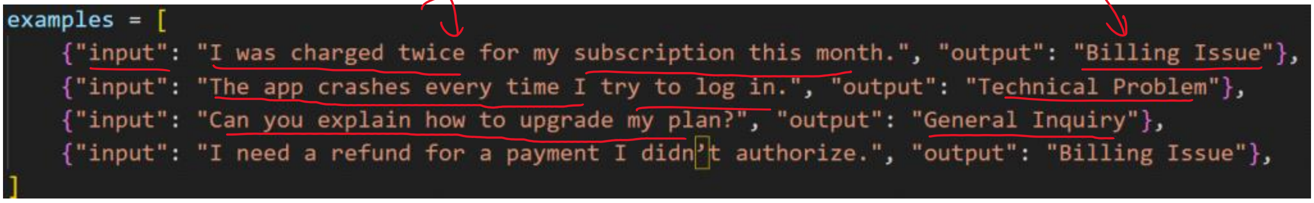
2. Role-Based Prompts



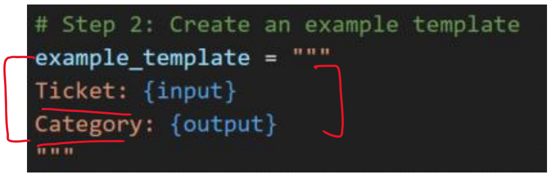
```
# Define the ChatPromptTemplate using from_template
chat_prompt = ChatPromptTemplate.from_template([
    ("system", "Hi you are a experienced {profession}"),
    ("user", "Tell me about {topic}"),
])

# Format the prompt with the variable
formatted_messages = chat_prompt.format_messages(profession="Doctor", topic="Viral Fever")
```

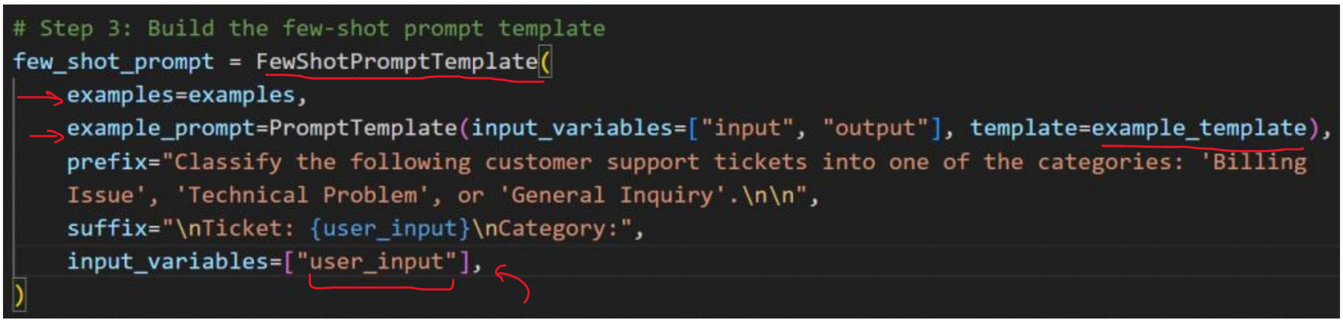
3. Few Shot Prompting



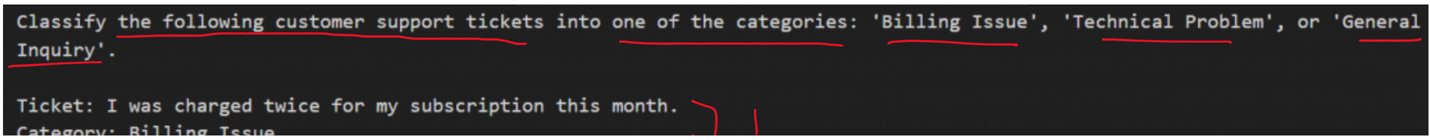
```
examples = [
    {"input": "I was charged twice for my subscription this month.", "output": "Billing Issue"},
    {"input": "The app crashes every time I try to log in.", "output": "Technical Problem"},
    {"input": "Can you explain how to upgrade my plan?", "output": "General Inquiry"},
    {"input": "I need a refund for a payment I didn't authorize.", "output": "Billing Issue"},
]
```



```
# Step 2: Create an example template
example_template = """
Ticket: {input}
Category: {output}
"""
```

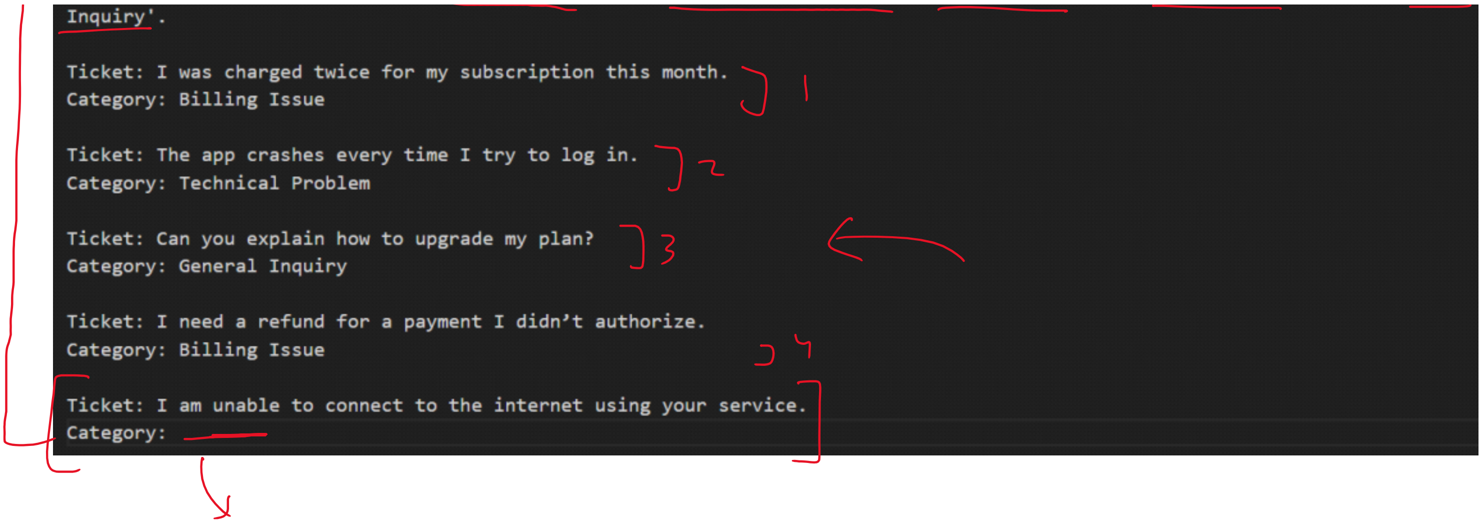


```
# Step 3: Build the few-shot prompt template
few_shot_prompt = FewShotPromptTemplate(
    examples=examples,
    example_prompt=PromptTemplate(input_variables=["input", "output"], template=example_template),
    prefix="Classify the following customer support tickets into one of the categories: 'Billing
    Issue', 'Technical Problem', or 'General Inquiry'.\n\n",
    suffix="\nTicket: {user_input}\nCategory:",
    input_variables=["user_input"],
)
```



```
Classify the following customer support tickets into one of the categories: 'Billing Issue', 'Technical Problem', or 'General
Inquiry'.

Ticket: I was charged twice for my subscription this month.
Category: Billing Issue
```



```
Inquiry'.

Ticket: I was charged twice for my subscription this month.
Category: Billing Issue

Ticket: The app crashes every time I try to log in.
Category: Technical Problem

Ticket: Can you explain how to upgrade my plan?
Category: General Inquiry

Ticket: I need a refund for a payment I didn’t authorize.
Category: Billing Issue

Ticket: I am unable to connect to the internet using your service.
Category:
```

1

2

3

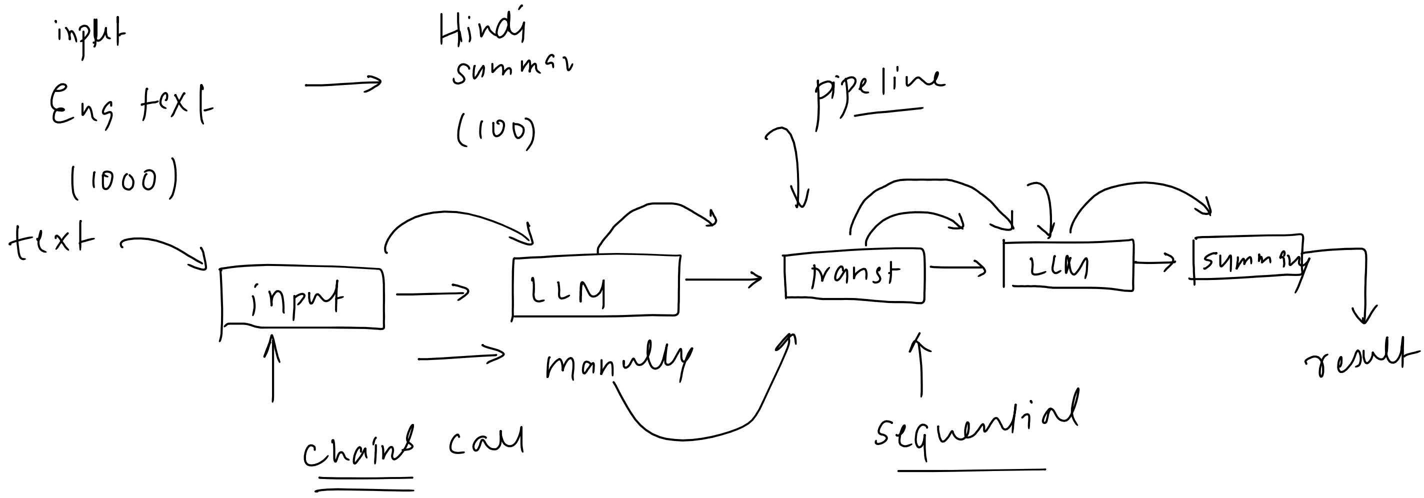
4

Chains

23 January 2025 10:31

Pipelines → LLM

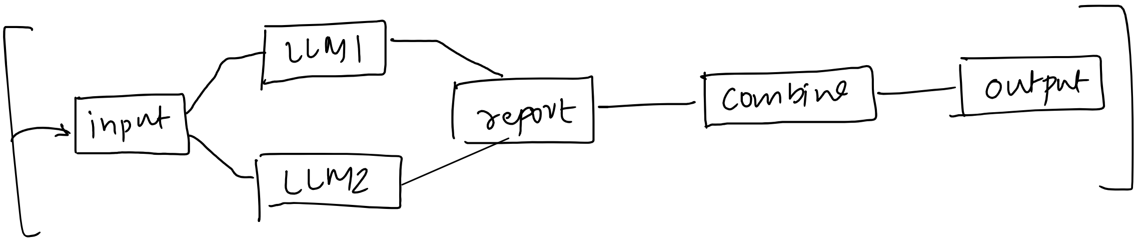
+ Pipeline



Complex pipel

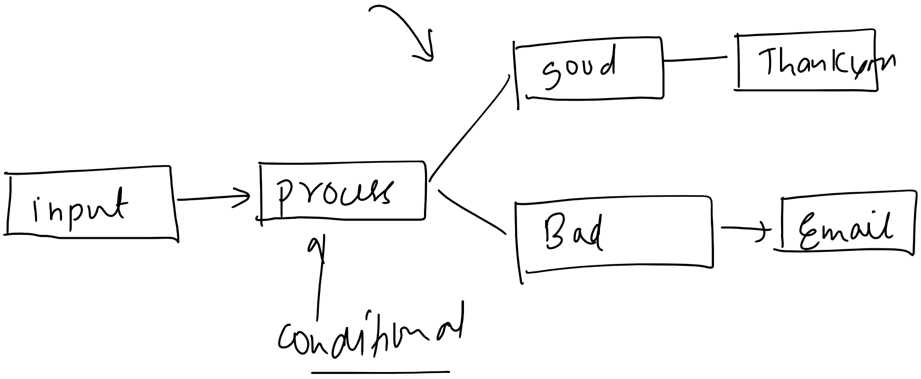
Parallel chain

Conditonal chains



AI Agent

feedback



Indexes

23 January 2025 10:31

Indexes connect your application to external knowledge—such as PDFs, websites or databases

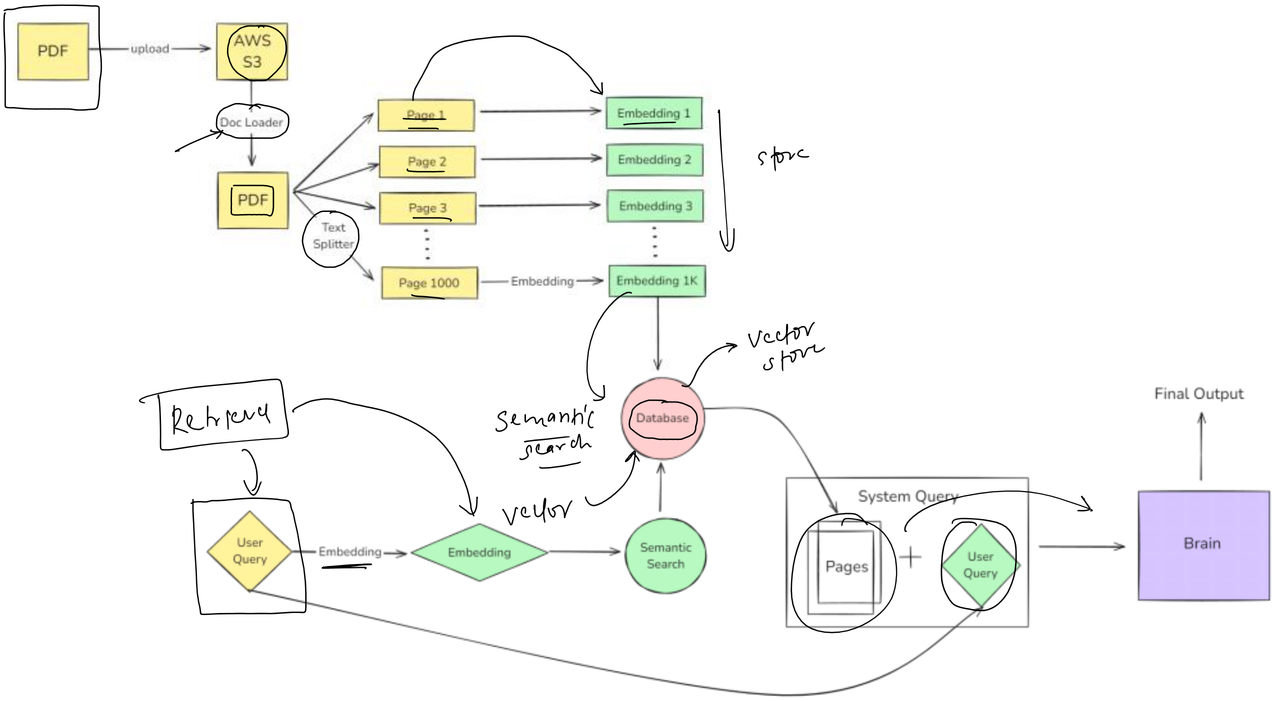
- Doc loader
- text splitter
- vector store
- Retrievers

Chatgpt → xyz → Rules normal

leave policy xyz
notice perm xyz

LLM + external knowledge

Rulebook 1000pages



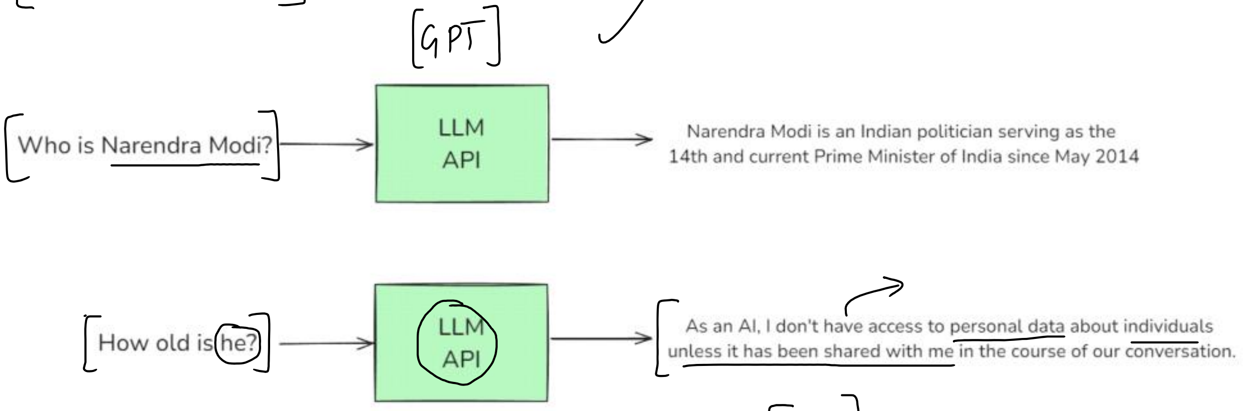
LLM
API

Memory

23 January 2025 10:31

LLM API calls are stateless

Chatbot

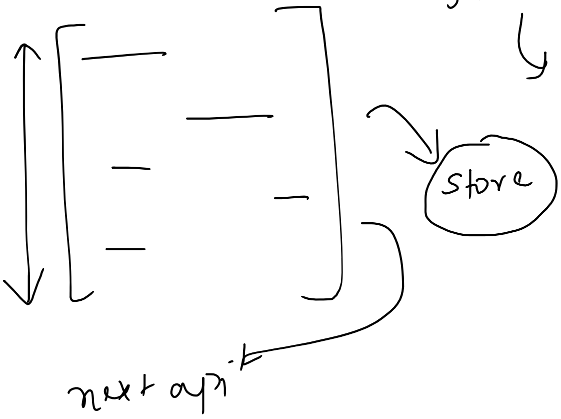


Narendra Modi is an Indian politician serving as the 14th and current Prime Minister of India since May 2014

As an AI, I don't have access to personal data about individuals unless it has been shared with me in the course of our conversation.

last n inters [100]

summary
↳ api call



- ConversationBufferMemory: Stores a transcript of recent messages. Great for short chats but can grow large quickly.
- ConversationBufferWindowMemory: Only keeps the last N interactions to avoid excessive token usage.
- Summarizer-Based Memory: Periodically summarizes older chat segments to keep a condensed memory footprint.
- Custom Memory: For advanced use cases, you can store specialized state (e.g., the user's preferences or key facts about them) in a custom memory class.

Agents

23 January 2025 10:31

AI agent

AI Agents

- LLMs (NLU + Text gen) → chatbot

Chatbot

AI agent
↓
Chatbot with super

travel website

- Shimla
- ~~Manali~~

24th Jan

API → Ingido

→ Book the flight

AI agent

- Reasoning capabili
- Tools

Tools → Calculator, Weather API

User → AI agent → Can you multi today's temp of Delhi with 3

- Chain Of Thought

Delhi temp

→ 25°C

→ 25, 3, *

75 → 75