In [1]:

```
import os
from dotenv import load_dotenv
from langchain.chat_models import ChatOpenAI
from langchain.prompts import PromptTemplate
from langchain.chains import LLMChain
from langchain.agents import Tool, initialize_agent, AgentType
from langchain.memory import ConversationBufferMemory

# Load API keys
load_dotenv(".env")
openai_api_key = os.getenv("OPENAI_API_KEY")

# LLM
llm = ChatOpenAI(model="gpt-3.5-turbo", temperature=0)

# Tool: Simple QA
qa_prompt = PromptTemplate.from_template("Answer clearly: {question}")
qa_chain = LLMChain(llm=llm, prompt=qa_prompt)
qa_tool = Tool(
    name="Simple QA",
    func=qa_chain.run,
    description="Answer factual questions clearly"
)
```

```
C:\Users\Ramisha Aravind\AppData\Local\Temp\ipykernel_20056\3986403611.py:14:
LangChainDeprecationWarning: The class `ChatOpenAI` was deprecated in LangCha
in 0.0.10 and will be removed in 1.0. An updated version of the class exists
in the :class:`~langchain-openai package and should be used instead. To use i
t run `pip install -U :class:`~langchain-openai` and import as `from :class:`
~langchain_openai import ChatOpenAI``.
  llm = ChatOpenAI(model="gpt-3.5-turbo", temperature=0)
C:\Users\Ramisha Aravind\AppData\Local\Temp\ipykernel_20056\3986403611.py:18:
LangChainDeprecationWarning: The class `LLMChain` was deprecated in LangChain
0.1.17 and will be removed in 1.0. Use :meth:`~RunnableSequence, e.g., `promp
t | llm`` instead.
  qa_chain = LLMChain(llm=llm, prompt=qa_prompt)
```

Model Diagram

In [ ]:

```
#https://python.langchain.com/api_reference/langchain/memory.html?utm_source
```

In [6]:

```
#🧠 1. ConversationBufferMemory
from langchain.memory import ConversationBufferMemory

# Memory (stores chat history)
memory = ConversationBufferMemory(memory_key="chat_history", return_messages

# Agent with tool + memory
agent = initialize_agent(
    tools=[qa_tool],
    llm=llm,
    agent=AgentType.CONVERSATIONAL_REACT_DESCRIPTION,# ZERO_SHOT_REACT_DESCR
    verbose=True,
    memory=memory,
    handle_parsing_errors=True
)

# Run multiple interactions
print("1️⃣ First Question")
res1 = agent.run("What is LangChain?")
print("\nAnswer:", res1)

print("\n2️⃣ Follow-up Question")
res2 = agent.run("Who created it?")
print("\nAnswer:", res2)

print("\n3️⃣ Ask again about previous topic")
res3 = agent.run("Explain it simply again.")
print("\nAnswer:", res3)

#print(memory.chat_memory.messages)

for msg in memory.chat_memory.messages:
    print(f"{msg.type.upper()}: {msg.content}")
```

```
1️⃣ First Question


> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Simple QA
Action Input: What is LangChain?
Observation: LangChain is a blockchain-based platform that aims to provide la
nguage learning services and opportunities for users to practice and improve
their language skills through various interactive features and tools.
Do I need to use a tool? No
AI: LangChain is a blockchain-based platform that focuses on language learnin
g services and opportunities for users to practice and improve their language
skills through interactive features and tools.

> Finished chain.

Answer: LangChain is a blockchain-based platform that focuses on language lea
rning services and opportunities for users to practice and improve their lang
uage skills through interactive features and tools.

2️⃣ Follow-up Question


> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Simple QA
Action Input: Who created LangChain?
Observation: LangChain was created by a team of developers and language enthu
siasts led by its founder, Dr. Alexey Malashkevich.
Do I need to use a tool? No
AI: LangChain was created by a team of developers and language enthusiasts le
d by its founder, Dr. Alexey Malashkevich.

> Finished chain.

Answer: LangChain was created by a team of developers and language enthusiast
s led by its founder, Dr. Alexey Malashkevich.

3️⃣ Ask again about previous topic


> Entering new AgentExecutor chain...
Could not parse LLM output: `LangChain is a platform that helps people learn
and practice languages through interactive tools and features. It was created
by Dr. Alexey Malashkevich and a team of developers and language enthusiasts.
`
For troubleshooting, visit: https://python.langchain.com/docs/troubleshootin
g/errors/OUTPUT_PARSING_FAILURE
Observation: Invalid or incomplete response
Do I need to use a tool? Yes
Action: Simple QA
Action Input: Explain LangChain simply
Observation: LangChain is a blockchain platform that focuses on language-rela
ted services, such as translation, interpretation, and language learning. It
uses blockchain technology to securely and transparently connect users with l
```

```
anguage service providers, ensuring trust and efficiency in language transact
ions.
Do I need to use a tool? No
AI: LangChain is a blockchain platform that focuses on language-related servi
ces, such as translation, interpretation, and language learning. It uses bloc
kchain technology to securely and transparently connect users with language s
ervice providers, ensuring trust and efficiency in language transactions.

> Finished chain.

Answer: LangChain is a blockchain platform that focuses on language-related s
ervices, such as translation, interpretation, and language learning. It uses
blockchain technology to securely and transparently connect users with langua
ge service providers, ensuring trust and efficiency in language transactions.
HUMAN: What is LangChain?
AI: LangChain is a blockchain-based platform that focuses on language learnin
g services and opportunities for users to practice and improve their language
skills through interactive features and tools.
HUMAN: Who created it?
AI: LangChain was created by a team of developers and language enthusiasts le
d by its founder, Dr. Alexey Malashkevich.
HUMAN: Explain it simply again.
AI: LangChain is a blockchain platform that focuses on language-related servi
ces, such as translation, interpretation, and language learning. It uses bloc
kchain technology to securely and transparently connect users with language s
ervice providers, ensuring trust and efficiency in language transactions.
```

In [7]:

```python
#2)ConversationBufferWindowMemory

from langchain.memory import ConversationBufferWindowMemory
from langchain.agents import initialize_agent, AgentType

memory = ConversationBufferWindowMemory(k=3,memory_key="chat_history")

agent = initialize_agent(
    tools=[qa_tool],
    llm=llm,
    agent=AgentType.CONVERSATIONAL_REACT_DESCRIPTION,
    verbose=True,
    memory=memory
)

# Run multiple interactions
print("1️⃣ First Question")
res1 = agent.run("What is LangChain?")
print("\nAnswer:", res1)

print("\n2️⃣ Follow-up Question")
res2 = agent.run("Who created it?")
print("\nAnswer:", res2)

print("\n3️⃣ Ask again about previous topic")
res3 = agent.run("Explain it simply again.")
print("\nAnswer:", res3)
```

```
for msg in memory.chat_memory.messages:
    print(f"{msg.type.upper()}: {msg.content}")
```

```
C:\Users\Ramisha Aravind\AppData\Local\Temp\ipykernel_20056\935981996.py:6: L
angChainDeprecationWarning: Please see the migration guide at: https://pytho
n.langchain.com/docs/versions/migrating_memory/
  memory = ConversationBufferWindowMemory(k=3,memory_key="chat_history")
```

```
1️⃣ First Question


> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Simple QA
Action Input: What is LangChain?
Observation: LangChain is a blockchain-based platform that aims to provide la
nguage learning services and tools to users around the world. It utilizes blo
ckchain technology to create a decentralized and secure ecosystem for languag
e learners and educators to connect and exchange resources.
Do I need to use a tool? No
AI: LangChain is a blockchain-based platform that focuses on providing langua
ge learning services and tools to users globally. It uses blockchain technolo
gy to create a secure and decentralized environment for language learners and
educators to interact and share resources.

> Finished chain.

Answer: LangChain is a blockchain-based platform that focuses on providing la
nguage learning services and tools to users globally. It uses blockchain tech
nology to create a secure and decentralized environment for language learners
and educators to interact and share resources.

2️⃣ Follow-up Question


> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Simple QA
Action Input: Who created LangChain?
Observation: LangChain was created by a team of developers and language enthu
siasts led by its founder, Dr. Alexei Antonov.
Do I need to use a tool? No
AI: LangChain was created by a team of developers and language enthusiasts le
d by its founder, Dr. Alexei Antonov.

> Finished chain.

Answer: LangChain was created by a team of developers and language enthusiast
s led by its founder, Dr. Alexei Antonov.

3️⃣ Ask again about previous topic


> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Simple QA
Action Input: What is LangChain?
Observation: LangChain is a blockchain-based platform that aims to provide la
nguage learning services and opportunities for users to practice and improve
their language skills through various interactive features and tools.
Do I need to use a tool? No
AI: LangChain is a platform that uses blockchain technology to help people le
arn languages and improve their language skills through interactive features
and tools.
```

```
> Finished chain.

Answer: LangChain is a platform that uses blockchain technology to help peopl
e learn languages and improve their language skills through interactive featu
res and tools.
HUMAN: What is LangChain?
AI: LangChain is a blockchain-based platform that focuses on providing langua
ge learning services and tools to users globally. It uses blockchain technolo
gy to create a secure and decentralized environment for language learners and
educators to interact and share resources.
HUMAN: Who created it?
AI: LangChain was created by a team of developers and language enthusiasts le
d by its founder, Dr. Alexei Antonov.
HUMAN: Explain it simply again.
AI: LangChain is a platform that uses blockchain technology to help people le
arn languages and improve their language skills through interactive features
and tools.
```

In [10]:

```python
#🧠 3. ConversationSummaryMemory
from langchain.memory import ConversationSummaryMemory
from langchain.chat_models import ChatOpenAI

llm = ChatOpenAI()
memory = ConversationSummaryMemory(llm=llm,memory_key="chat_history")
# Agent with tool + memory
agent = initialize_agent(
    tools=[qa_tool],
    llm=llm,
    agent=AgentType.CONVERSATIONAL_REACT_DESCRIPTION,
    verbose=True,
    memory=memory
)


# Run multiple interactions
print("1️⃣ First Question")
res1 = agent.run("What is LangChain?")
print("\nAnswer:", res1)

print("\n2️⃣ Follow-up Question")
res2 = agent.run("Who created it?")
print("\nAnswer:", res2)

print("\n3️⃣ Ask again about previous topic")
res3 = agent.run("Explain it simply again.")
print("\nAnswer:", res3)

for msg in memory.chat_memory.messages:
    print(f"{msg.type.upper()}: {msg.content}")
```

```
1️⃣ First Question


> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Simple QA
Action Input: What is LangChain?
Observation: LangChain is a blockchain-based platform that aims to provide la
nguage learning services and opportunities for users to practice and improve
their language skills through various interactive features and tools.
Do I need to use a tool? No
AI: LangChain is a blockchain-based platform that aims to provide language le
arning services and opportunities for users to practice and improve their lan
guage skills through various interactive features and tools.

> Finished chain.

Answer: LangChain is a blockchain-based platform that aims to provide languag
e learning services and opportunities for users to practice and improve their
language skills through various interactive features and tools.

2️⃣ Follow-up Question


> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Simple QA
Action Input: Who created LangChain?
Observation: LangChain was created by a team of developers and language enthu
siasts led by its founder, Dr. Alexey Malashkevich.
Do I need to use a tool? No
AI: LangChain was created by a team of developers and language enthusiasts le
d by its founder, Dr. Alexey Malashkevich.

> Finished chain.

Answer: LangChain was created by a team of developers and language enthusiast
s led by its founder, Dr. Alexey Malashkevich.

3️⃣ Ask again about previous topic


> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Simple QA
Action Input: What is LangChain?
Observation: LangChain is a blockchain-based platform that aims to provide la
nguage learning services and opportunities for users to practice and improve
their language skills through various interactive features and tools.
Do I need to use a tool? No
AI: LangChain is a blockchain-based platform that offers language learning se
rvices and tools for users to practice and improve their language skills. It
was created by a team of developers and language enthusiasts led by its found
er, Dr. Alexey Malashkevich.

> Finished chain.
```

```
Answer: LangChain is a blockchain-based platform that offers language learnin
g services and tools for users to practice and improve their language skills.
It was created by a team of developers and language enthusiasts led by its fo
under, Dr. Alexey Malashkevich.
HUMAN: What is LangChain?
AI: LangChain is a blockchain-based platform that aims to provide language le
arning services and opportunities for users to practice and improve their lan
guage skills through various interactive features and tools.
HUMAN: Who created it?
AI: LangChain was created by a team of developers and language enthusiasts le
d by its founder, Dr. Alexey Malashkevich.
HUMAN: Explain it simply again.
AI: LangChain is a blockchain-based platform that offers language learning se
rvices and tools for users to practice and improve their language skills. It
was created by a team of developers and language enthusiasts led by its found
er, Dr. Alexey Malashkevich.
```

In [14]:

```python
from langchain.memory import VectorStoreRetrieverMemory
from langchain.vectorstores import FAISS
from langchain.embeddings import OpenAIEmbeddings

embedding = OpenAIEmbeddings()
retriever = FAISS.from_texts(["initial memory"], embedding).as_retriever()

memory = VectorStoreRetrieverMemory(retriever=retriever,memory_key="chat_his
# Agent with tool + memory
agent = initialize_agent(
    tools=[qa_tool],
    llm=llm,
    agent=AgentType.CONVERSATIONAL_REACT_DESCRIPTION,
    verbose=True,
    memory=memory
)

# Run multiple interactions
print("1️⃣ First Question")
res1 = agent.run("What is LangChain?")
print("\nAnswer:", res1)

print("\n2️⃣ Follow-up Question")
res2 = agent.run("Who created it?")
print("\nAnswer:", res2)

print("\n3️⃣ Ask again about previous topic")
res3 = agent.run("Explain it simply again.")
print("\nAnswer:", res3)
```

```
1️⃣ First Question


> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Simple QA
Action Input: What is LangChain?
Observation: LangChain is a blockchain-based platform that aims to provide la
nguage learning services and opportunities for users to practice and improve
their language skills through various interactive features and tools.
Do I need to use a tool? No
AI: LangChain is a blockchain-based platform that focuses on language learnin
g services and opportunities for users to practice and improve their language
skills through interactive features and tools.

> Finished chain.

Answer: LangChain is a blockchain-based platform that focuses on language lea
rning services and opportunities for users to practice and improve their lang
uage skills through interactive features and tools.

2️⃣ Follow-up Question


> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Simple QA
Action Input: Who created LangChain?
Observation: LangChain was created by a team of developers and language enthu
siasts led by its founder, Dr. Alexey Malashkevich.
Do I need to use a tool? No
AI: LangChain was created by a team of developers and language enthusiasts le
d by its founder, Dr. Alexey Malashkevich.

> Finished chain.

Answer: LangChain was created by a team of developers and language enthusiast
s led by its founder, Dr. Alexey Malashkevich.

3️⃣ Ask again about previous topic


> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Simple QA
Action Input: What is LangChain?
Observation: LangChain is a blockchain-based platform that aims to provide la
nguage learning services and opportunities for users to practice and improve
their language skills through various interactive features and tools.
Do I need to use a tool? No
AI: LangChain is a platform that helps users practice and improve their langu
age skills through interactive features and tools.

> Finished chain.
```

```
Answer: LangChain is a platform that helps users practice and improve their l
anguage skills through interactive features and tools.
```

In [11]:

```python
# Check what's stored in FAISS retriever
docs = retriever.vectorstore.similarity_search("memory")
for i, doc in enumerate(docs):
    print(f"🔹 Document {i+1}: {doc.page_content}")
```

```
🔹 Document 1: initial memory
🔹 Document 2: input: Explain it simply again.
output: LangChain is a blockchain-based platform that aims to provide languag
e learning services and opportunities for users to practice and improve their
language skills through various interactive features and tools.
🔹 Document 3: input: What is LangChain?
output: LangChain is a blockchain-based platform that focuses on providing la
nguage learning services and opportunities for users to practice and improve
their language skills.
🔹 Document 4: input: Who created it?
output: LangChain was created by a team of developers and language enthusiast
s led by its founder, Dr. Alexei Antonov.
```

In [12]:

```python
# See all texts in FAISS index
print("All indexed texts:")
for i, doc in enumerate(retriever.vectorstore.docstore._dict.values()):
    print(f"📄 {i+1}: {doc.page_content}")
```

```
All indexed texts:
📄 1: initial memory
📄 2: input: What is LangChain?
output: LangChain is a blockchain-based platform that focuses on providing la
nguage learning services and opportunities for users to practice and improve
their language skills.
📄 3: input: Who created it?
output: LangChain was created by a team of developers and language enthusiast
s led by its founder, Dr. Alexei Antonov.
📄 4: input: Explain it simply again.
output: LangChain is a blockchain-based platform that aims to provide languag
e learning services and opportunities for users to practice and improve their
language skills through various interactive features and tools.
```

In [15]:

```python
query = "LangChain founder"
results = memory.retriever.get_relevant_documents(query)

print(f"\n🧠 Retrieval for: '{query}'")
for i, doc in enumerate(results):
    print(f"🔹 {i+1}: {doc.page_content}")
```

```
C:\Users\Ramisha Aravind\AppData\Local\Temp\ipykernel_20056\1323272930.py:2:
LangChainDeprecationWarning: The method `BaseRetriever.get_relevant_documents
` was deprecated in langchain-core 0.1.46 and will be removed in 1.0. Use :me
th:`~invoke` instead.
  results = memory.retriever.get_relevant_documents(query)
```

```
🧠 Retrieval for: 'LangChain founder'
🔹 1: input: Who created it?
output: LangChain was created by a team of developers and language enthusiast
s led by its founder, Dr. Alexey Malashkevich.
🔹 2: input: What is LangChain?
output: LangChain is a blockchain-based platform that focuses on language lea
rning services and opportunities for users to practice and improve their lang
uage skills through interactive features and tools.
🔹 3: input: Explain it simply again.
output: LangChain is a platform that helps users practice and improve their l
anguage skills through interactive features and tools.
🔹 4: initial memory
```

In [ ]:

```python
memory.save_context(
    {"input": "Who is the founder of LangChain?"},
    {"output": "Harrison Chase is the founder of LangChain."}
)
```

In [ ]:

```python
#🧠 6. PostgresChatMessageHistory, RedisChatMessageHistory, etc.

from langchain.memory import ConversationBufferMemory
from langchain.storage import PostgresChatMessageHistory

history = PostgresChatMessageHistory(session_id="abc123", connection_string=
memory = ConversationBufferMemory(chat_memory=history)
```

In [ ]: