Analytics Vidhya

LangChain Legacy Syntax

Instructor

Dipanjan Sarkar

Head of Community & Principal AI Scientist at Analytics Vidhya

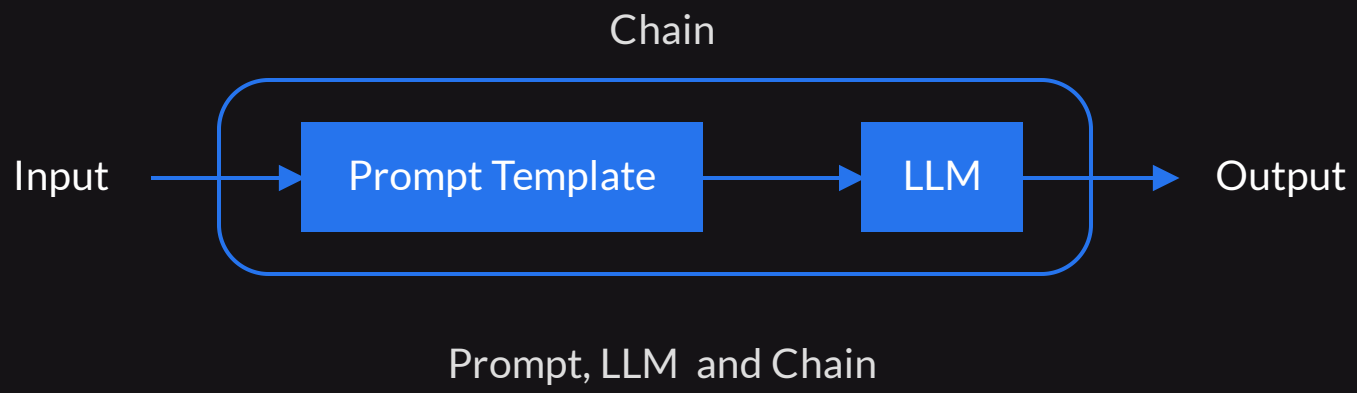
Google Developer Expert - ML & Cloud Champion Innovator

Published Author

Outline

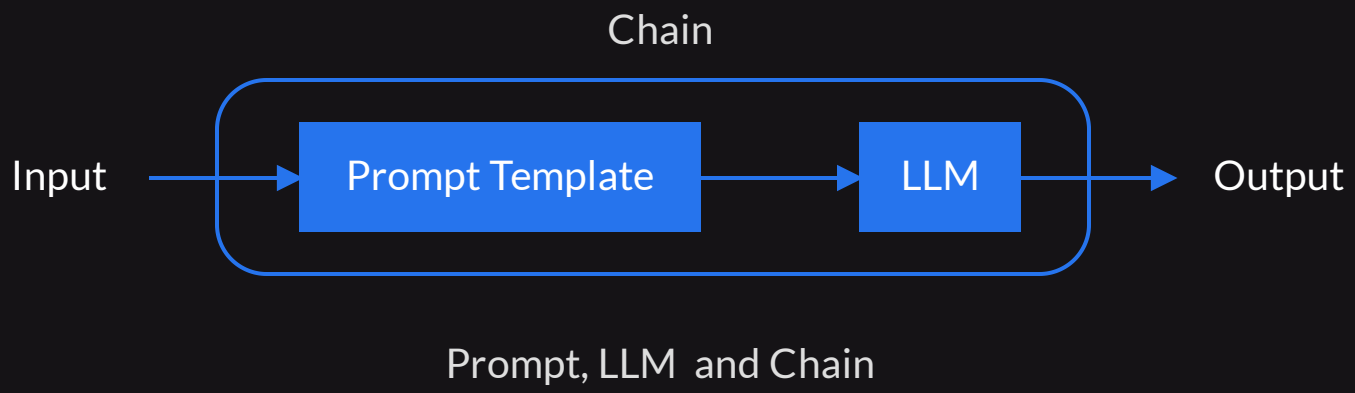
- What is LangChain’s Legacy Syntax?
- What is a LangChain Chain?
- What is a Legacy Chain in LangChain?
- Popular Legacy Chains

What is LangChain's Legacy Syntax?



- Before August 2023, we had a different syntax for creating LLM Chains
- LangChain now calls them as legacy chains
- Newer syntax is built using LangChain Expression Language (LCEL)
- Not all legacy chains have been converted into LCEL variants

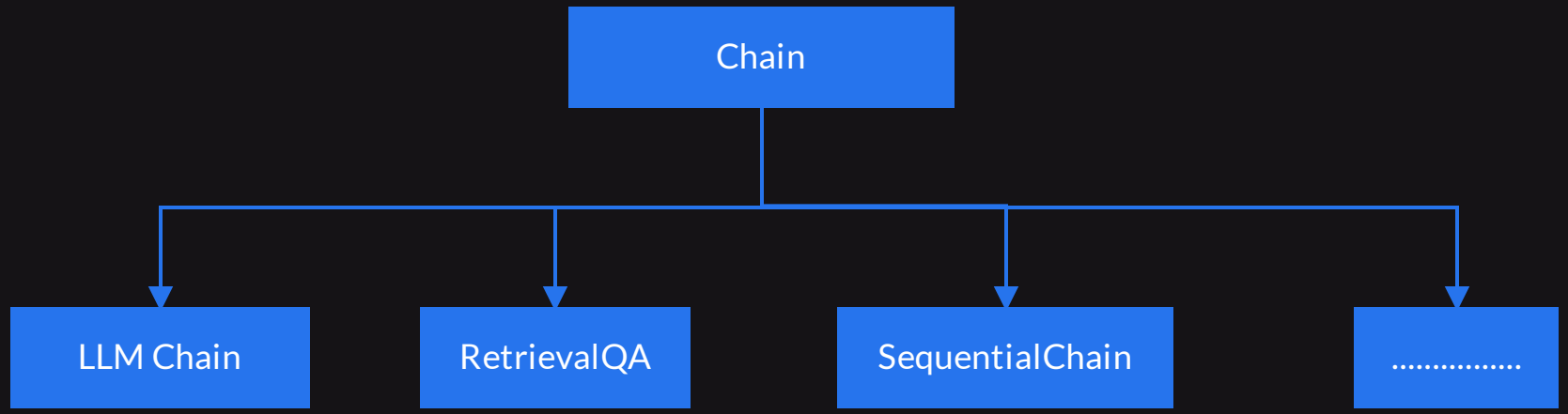
What is LangChain's Legacy Syntax?



Prompt, LLM and Chain

- A Chain is a sequence or a pipeline of steps
- These steps are typically calls in LangChain
  - Calling an LLM
  - Calling a tool
  - Calling a retriever
  - Calling a preprocessing operation

What is a Legacy Chain in LangChain?



- Legacy Chains are constructed by subclassing from a legacy Chain class
- These do not use LCEL but have their own implementation logic
- Many of them are being deprecated so need to keep a check on LangChain new releases and docs
- Some of them are still used as there is no LCEL variant yet

Popular Legacy Chains

Chain Type	Description
LLMChain	Simplest chain to pass queries or prompts to LLMs and get responses
SequentialChain	Connect multiple chains sequentially, where the outputs of one chain feed directly into next
LLMRouterChain	A router chain that uses an LLM chain to perform routing prompts to specific chains based on the prompt and certain conditions
RetrievalQA	This chain is used to connect a vector database retriever to an LLM to build a RAG chain to get responses based on custom data
ConversationChain	Enables the chain to store past conversations in memory and have full conversation with the LLM
ConversationalRetrievalChain	Combination of conversational and RAG chains, enables you to get answers based on custom data and also have a conversation with the LLM

Legacy Chain Example

```python
from langchain_openai import ChatOpenAI
from langchain.chains import LLMChain
from langchain_core.prompts import ChatPromptTemplate

# connect to ChatGPT
chatgpt = ChatOpenAI(model_name="gpt-3.5-turbo", temperature=0)

# create a prompt template to accept user queries
prompt_txt = "{query}"
prompt_template = ChatPromptTemplate.from_template(prompt_txt)

# chain prompt with LLM
llmchain = LLMChain(llm=chatgpt, prompt=prompt_template)

# invoke the chain
response = llmchain.invoke({'query' : 'Explain Generative AI in 1 line'})
print(response['text'])
```

Thank You

Analytics Vidhya