

AI Agent Building

We will be using LangChain and LangGraph to build Agent for following examples. Both are open source frameworks designed to help developers build applications with LLMs.

LangChain : Building LLM powered applications by executing a sequence of functions in a chain.

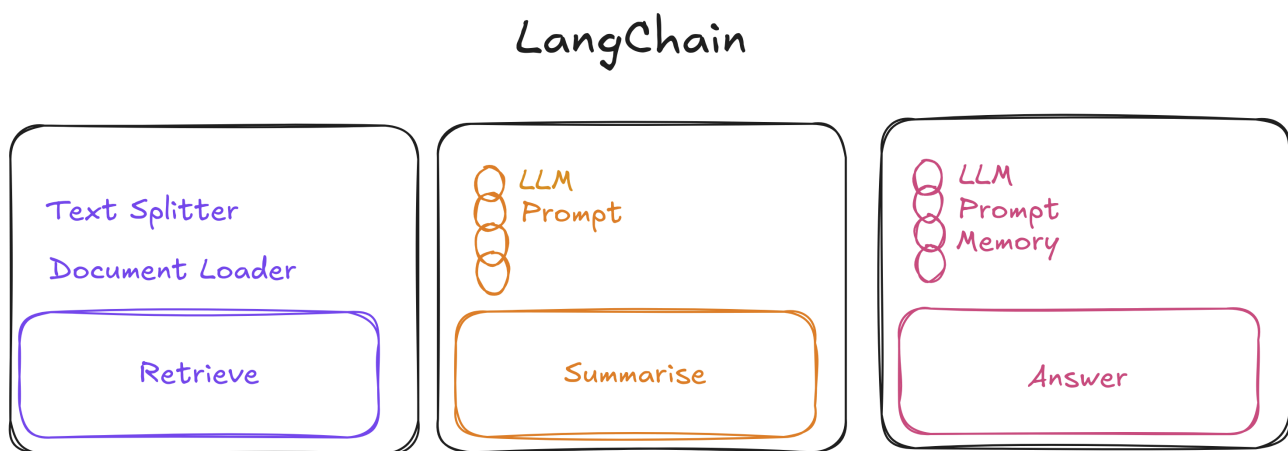
1. Retrieve : Data from a website.
2. Summarise : That data that we retrieved and summary the data to do something with it.
3. Answer : Then we have it answer user questions.

Example of a retrieve is a **Document Loader** . Fetch/Load content from different data sources. Very Large Documents we can use a Text Splitter which is a another LangChain Component.

For the summarise we would use a Chain. The Chain will include a **prompt component** and also contain an LLM component to pass request's to an LLM.

For the answer we will have new chain that may include a memory component that would store conversation, history and context. We would also have another prompt and LLM in this chain to generate the answer.

We may use different LLMs in the summarise step then in the answer step.



LangGraph : is an extension of LangChain designed for building stateful multi-agents systems that can handle complex nonlinear workflows.

1. Process User Input (LLM Component to understand and route to the appropriate action node.)
2. Ad Tasks
3. Completed Tasks
4. Summarise Tasks
5. State (Maintain the task list across all the interaction)

LangGraph helps us create this as a graph structure. Each step above is considered as a node. The flow between each step is known as edges.

LangChain is to provide an abstraction layer for chaining LLM operations into a LLM application.

LangGraph is to create and manage multi agent systems and workflows.

- LangGraph is an extension of LangChain that supports graphs
- Single and Multi-agents flows are described and represented as graphs.
- Built-in persistence allows for human-in-the-loop-workflows

LangGraph / LangChain / LangFlow and LangSmith

First we will be connecting to Ollama using ChatOpenAI :

ChatOpenAI is the primary class used for chatting with LLM models. This represents LangChain's interface for interacting with OpenAI's or Ollama API.

Internally, this is a subclass of *BaseChatModel*, which is a generic class that implements subclasses for individual API services.

We will first setup and adding a prompt template

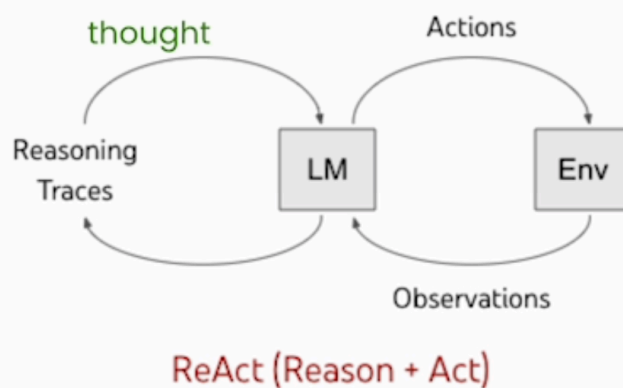
A prompt template is **a reproducible way to generate prompts**. It allows us to create prompts based on existing statements that have placeholders for queries or field names.

Please review notebook : [Simple_LLM_Ollama_1](#)

Next we are going to add build a ReAct Agent using this flow below :

Please review notebook : [ReAct_Agent_2](#)

Let's build an agent from scratch



Published as a conference paper at ICLR 2023

REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

Shunyu Yao^{*1}, Jeffrey Zhao², Dian Yu², Nan Du², Izhak Shafran², Karthik Narasimhan¹, Yuan Cao²

¹Department of Computer Science, Princeton University

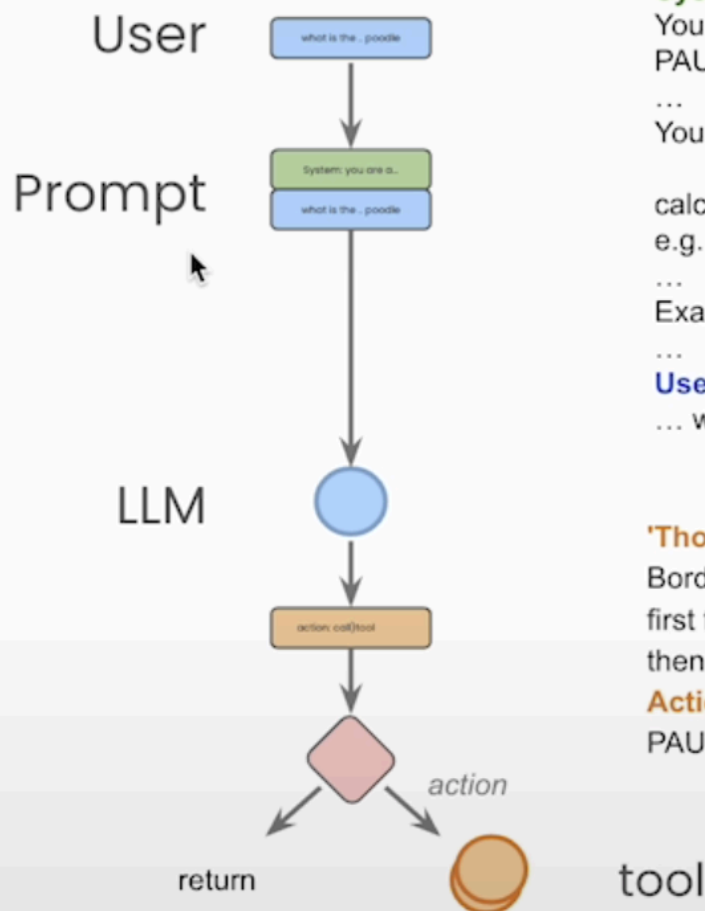
²Google Research, Brain team

¹{shunyuy, karthikh}@princeton.edu

²{jeffreyzhao, dianyu, dunan, izhak, yuancao}@google.com

<https://arxiv.org/pdf/2210.03629.pdf>

Break Down



System:

You run in a loop of Thought, Action, PAUSE, Observation.

...

Your available actions are:

calculate:

e.g. calculate: $4 * 7 / 3$

...

Example session:

...

User:

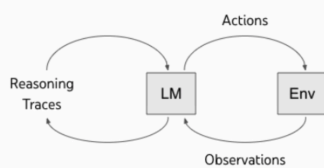
... weight of collie...

'Thought: To find the combined weight of a Border Collie and a Scottish Terrier, I need to first find the average weight of each breed and then add those weights together.

Action: average_dog_weight: Border Collie\n PAUSE'

Different Flows of Agents

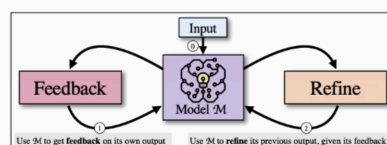
LangGraph supports Cyclic Graphs



ReAct (Reason + Act)

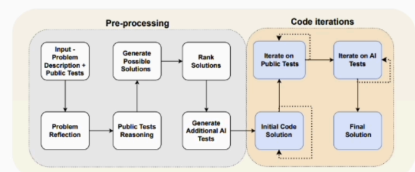
REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

<https://arxiv.org/pdf/2210.03629.pdf>



SELF-REFINE:
Iterative Refinement with Self-Feedback

<https://arxiv.org/pdf/2303.17651.pdf>



(a) The proposed AlphaCodium flow.

Code Generation with AlphaCodium: From Prompt Engineering to Flow Engineering

<https://arxiv.org/pdf/2401.08500.pdf>