



# Building Context-Aware Reasoning Applications with LangChain

H

what's the weather like in San Francisco right now?



I'm sorry but I currently do not have the capability to access real-time data, including current weather information. If you want up-to-date weather information for San Francisco, or any other location, I'd recommend checking a reliable weather website or using a weather app on your mobile device.

If you have any other questions or need assistance with a different topic, please let me know!

# Build AI Chat Search For Documentation

The best way to create production-ready chat with your data apps. Ingest, customize, then deploy with one line of code.

Get Started

Talk to Us

BACKED BY  **Combinator**

# Get a Second Brain with Quivr

Quivr is your second brain in the cloud, designed to easily store and retrieve unstructured information.

Get Started

Github 

## The best decision for your money

Organize your savings by investing in simple and regulated alternatives, with low commissions. No paperwork or fine print.

Open your free account



## Your technical documentation Assistant

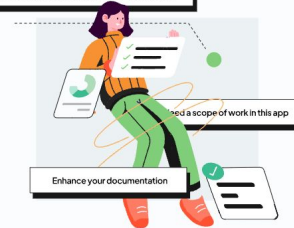
Streamline your technical communication and content generation with your own internal sales and support knowledge base.

Enter your email

Join our waitlist

 Wanna see a demo?

Create a customer integration proposal in no time

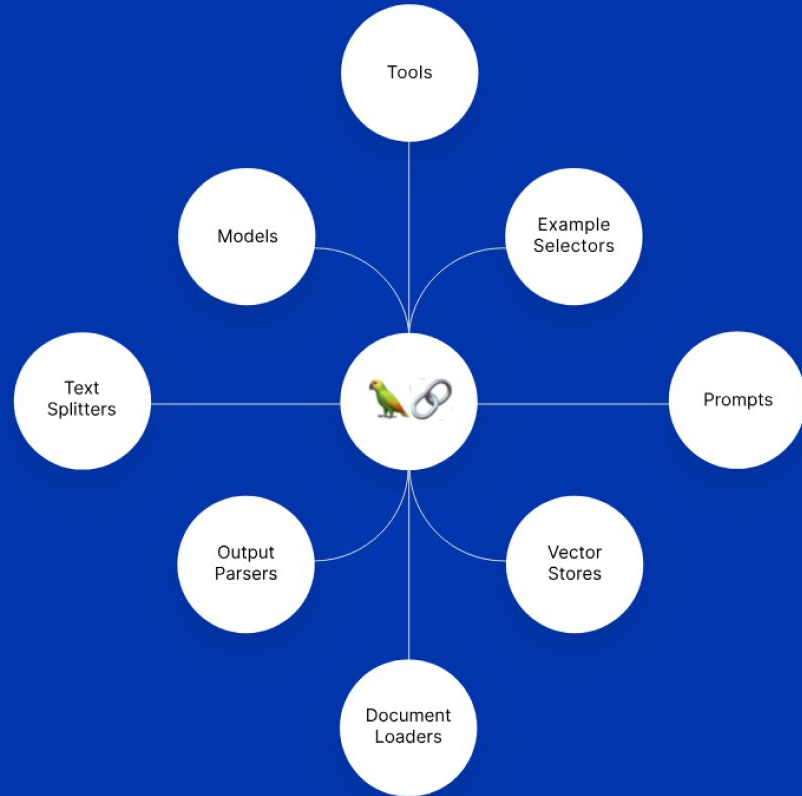


Define a scope of work in this app

Enhance your documentation

ENGLISH  
SDK FOR  **SPARK**

# Context-aware



# 4 Types of Context-Awareness

1. Instruction Prompting
2. Few Shot Examples
3. RAG
4. Fine-tuning

# 1. Instruction Prompting

Be a helpful chatbot  
Respond in a friendly tone  
Only talk about LangChain

## 2. Few Shot Examples

Hi // Hi. I'm a LangChain Chatbot.

Can you tell me a joke? // I only help  
with LangChain

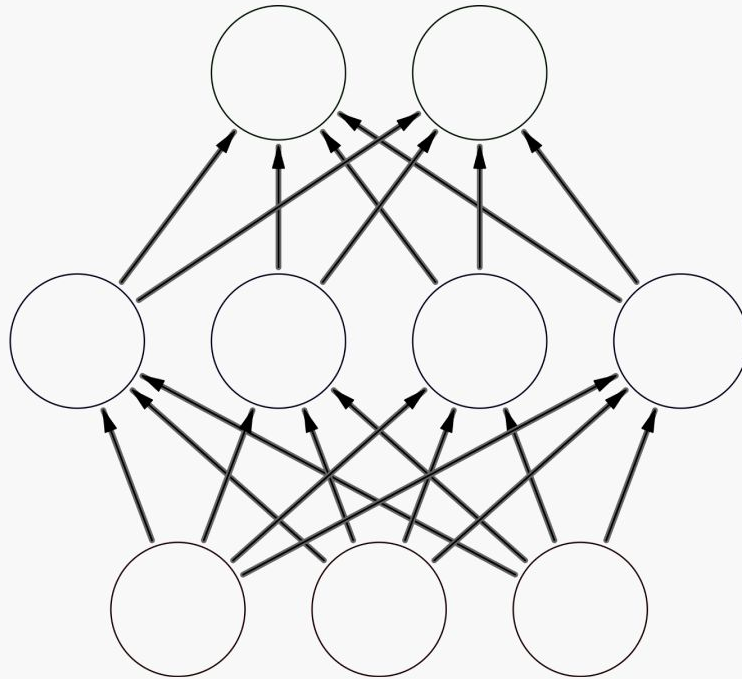
# 3. Retrieval Augmented Generation (RAG)

**answer the user question based only on the context below**

*LangChain is a framework for developing applications powered by large language models. It enables context-aware, reasoning applications. The main value props of LangChain are components and off-the-shelf chains. Components are abstractions for working with large language models, along with a collection of implementations for each abstraction. Components are modular and easy-to-use, whether you are using the rest of the LangChain framework or not. Off-the-shelf chains are structured assemblies of components for accomplishing higher-level tasks.*



## 4. Fine-tuning





# 5 Levels of Reasoning

1. LLM Call
2. Chaining
3. Routing
4. Automations / State Machines
5. Autonomous Agents

# 1. LLM Call

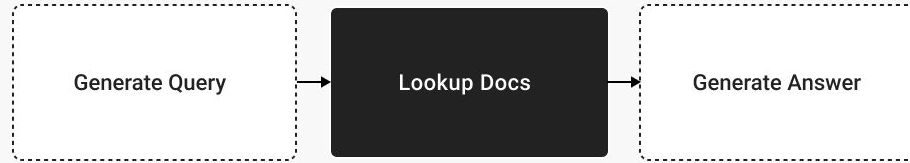


Llama 2

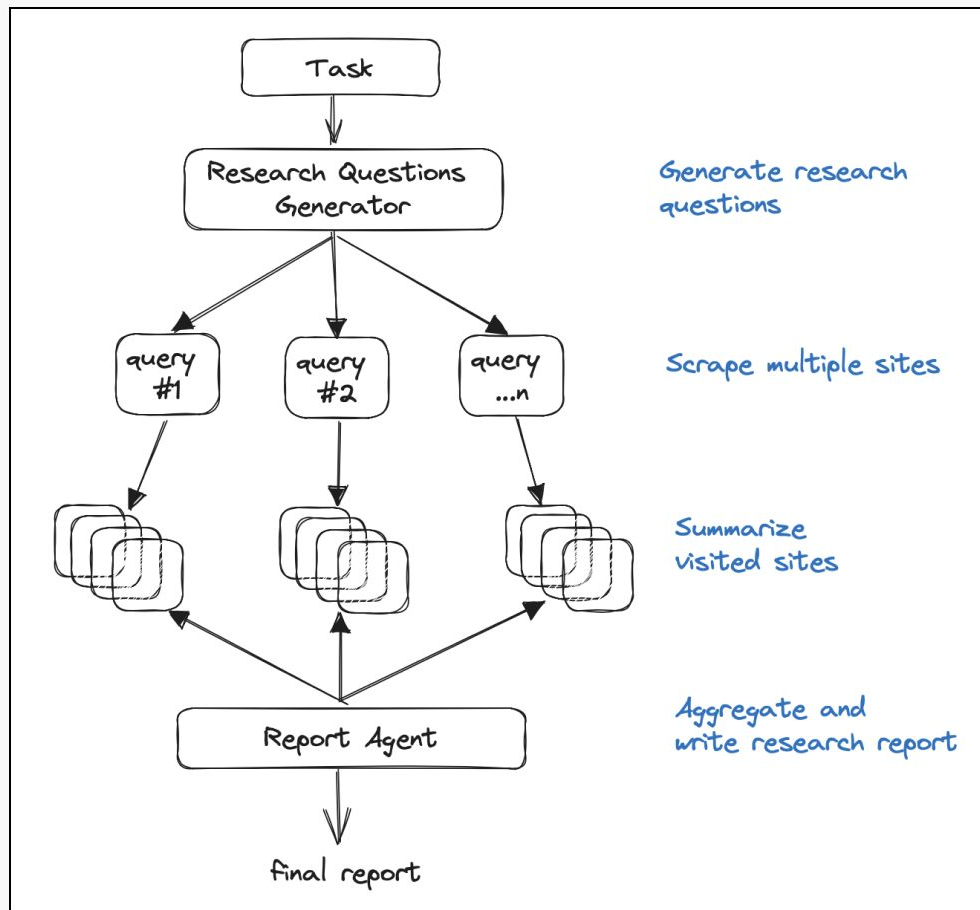


Falcon 180B

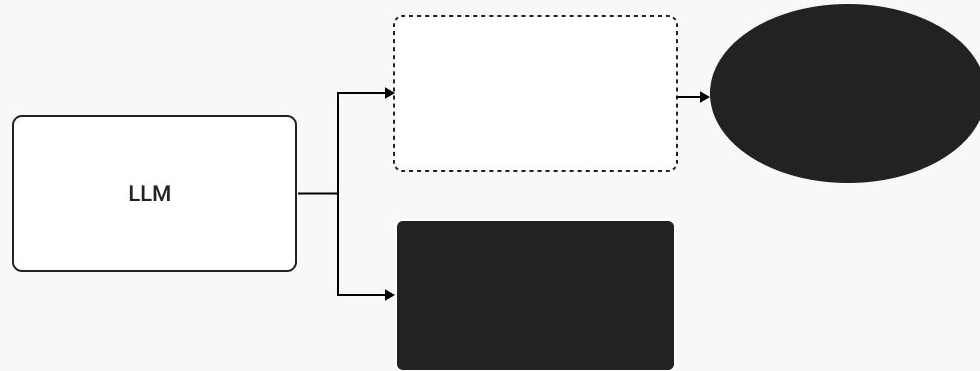
## 2. Chaining



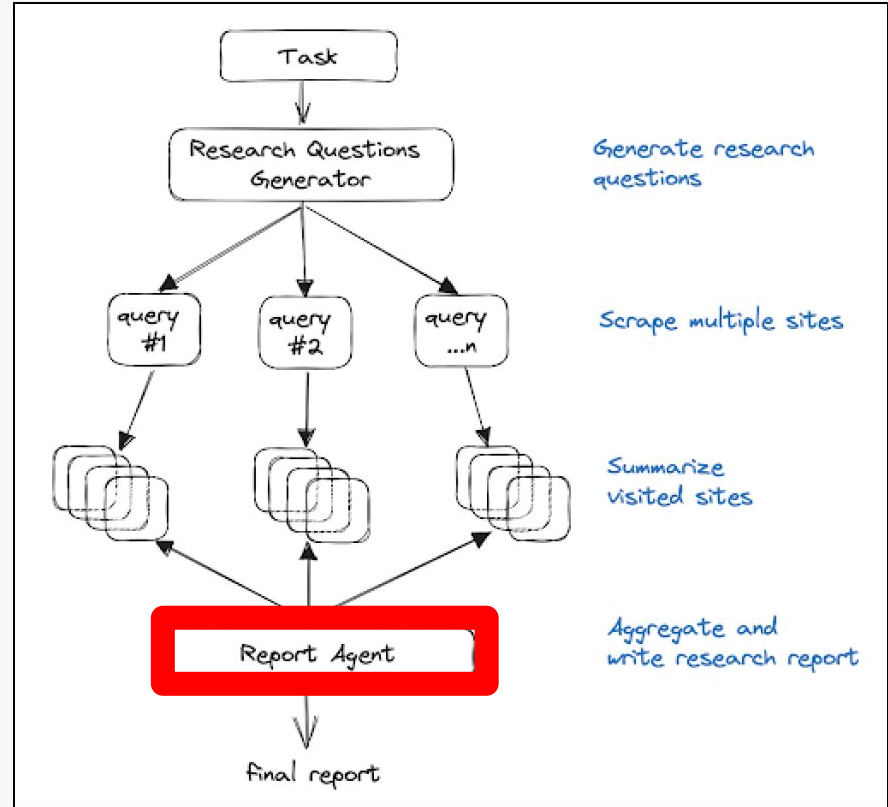
# GPT Researcher



# 3. Routing

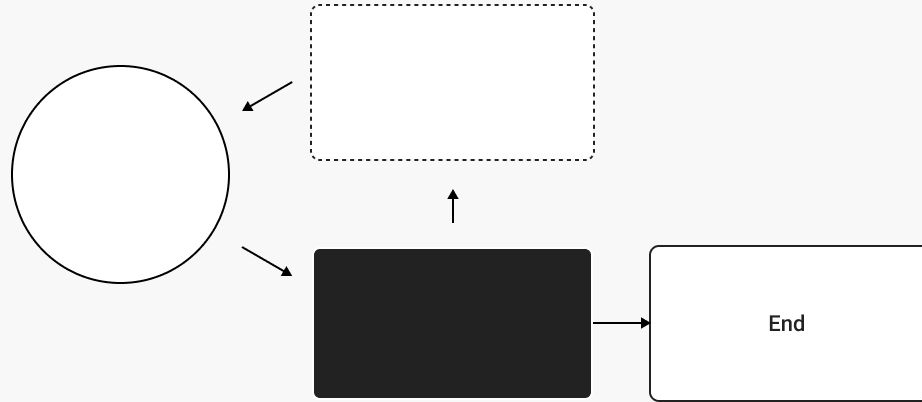


# GPT Researcher

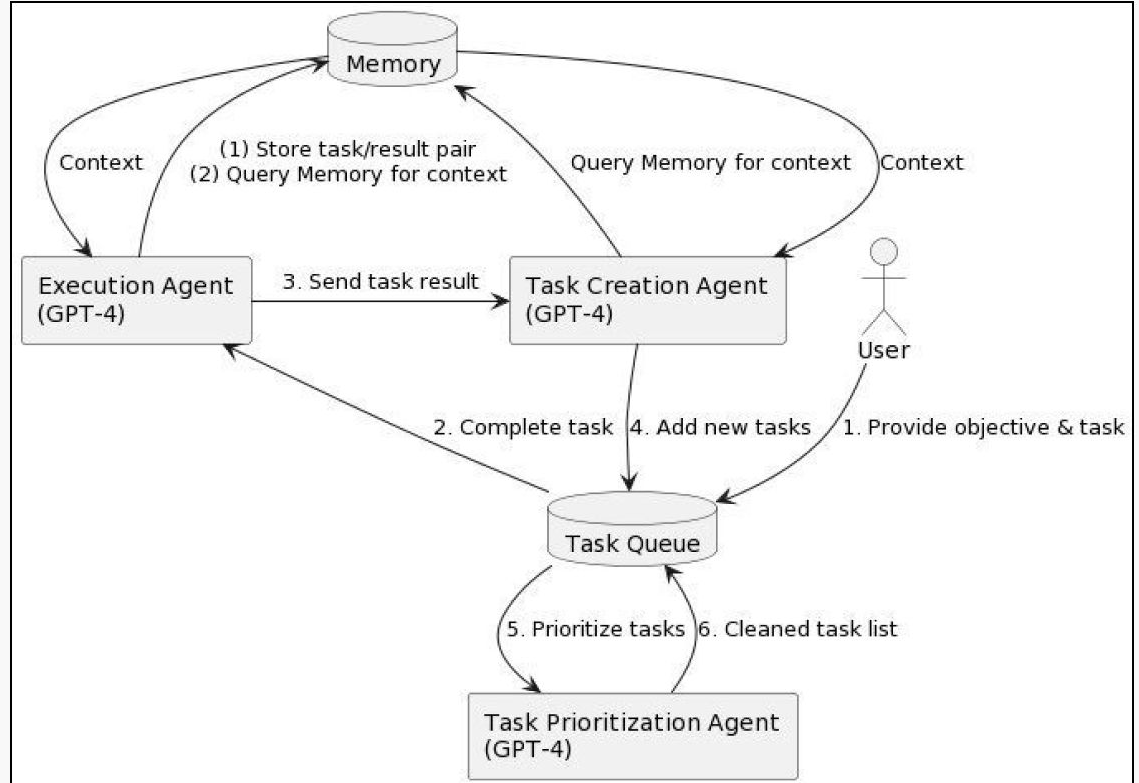




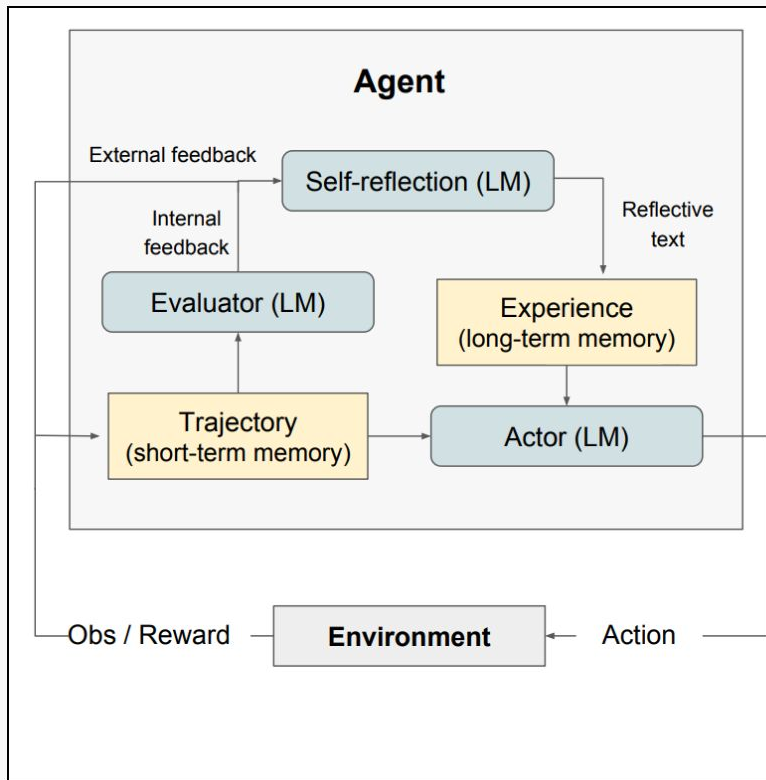
# 4. Automaton / State Machines



# Plan-and-Execute



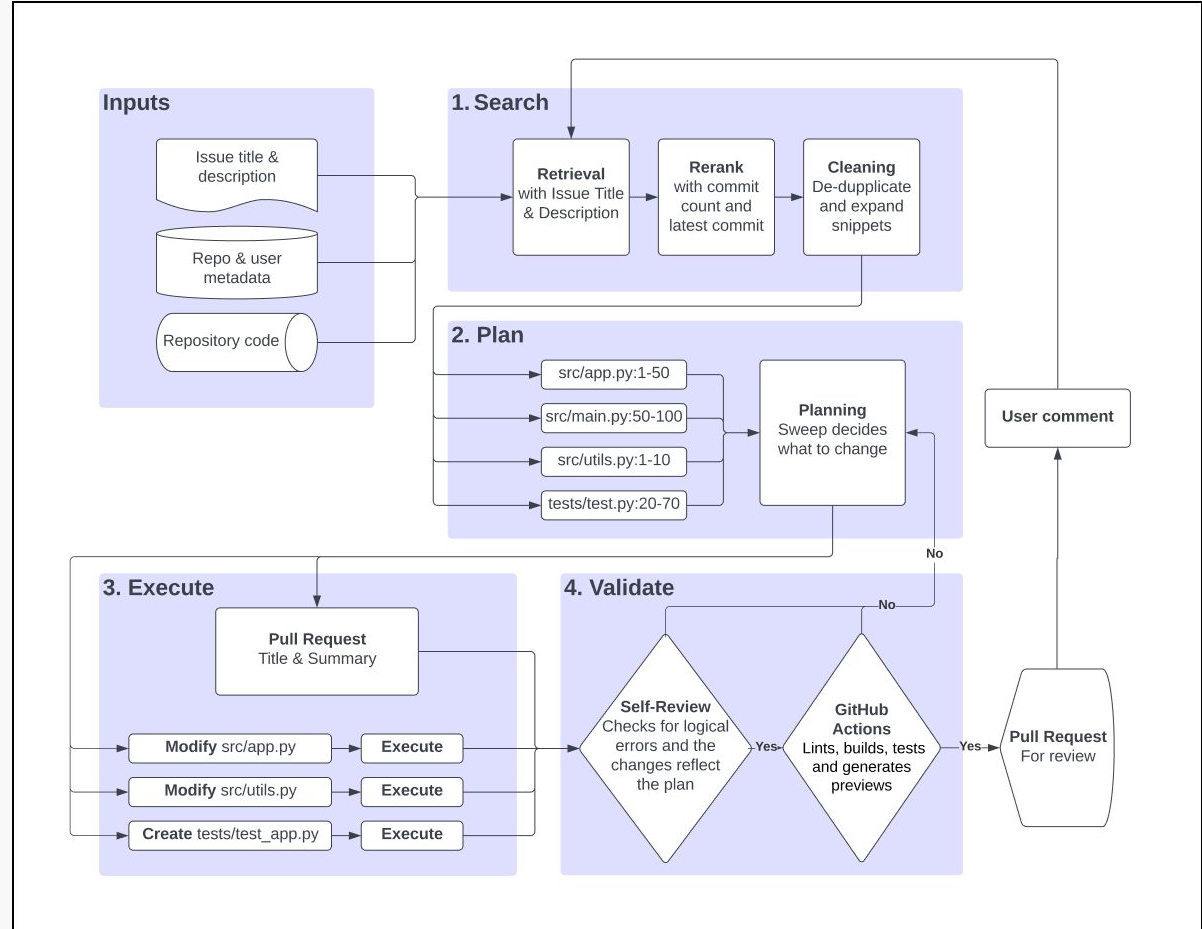
# Reflexion



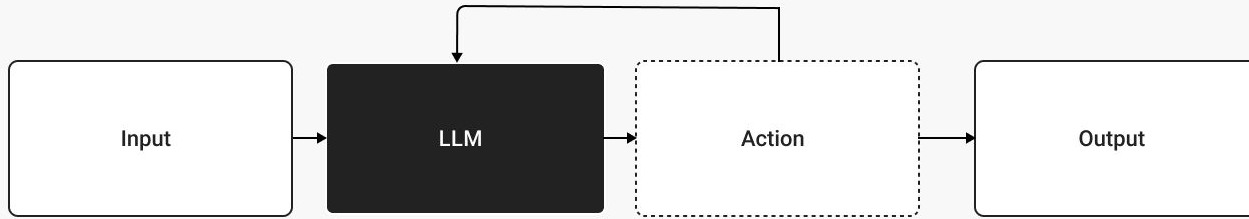
## Algorithm 1 Reinforcement via self-reflection

Initialize Actor, Evaluator, Self-Reflection:  
 $M_a, M_e, M_{sr}$   
Initialize policy  $\pi_\theta(a_i|s_i), \theta = \{M_a, mem\}$   
Generate initial trajectory using  $\pi_\theta$   
Evaluate  $\tau_0$  using  $M_e$   
Generate initial self-reflection  $sr_0$  using  $M_{sr}$   
Set  $mem \leftarrow [sr_0]$   
Set  $t = 0$   
**while**  $M_e$  not pass or  $t < \text{max trials}$  **do**  
    Generate  $\tau_t = [a_0, o_0, \dots, a_i, o_i]$  using  $\pi_\theta$   
    Evaluate  $\tau_t$  using  $M_e$   
    Generate self-reflection  $sr_t$  using  $M_{sr}$   
    Append  $sr_t$  to  $mem$   
    Increment  $t$   
**end while**  
**return**

# Sweep.dev



# 5. Autonomous Agents



# Levels of autonomy in LLM applications



code



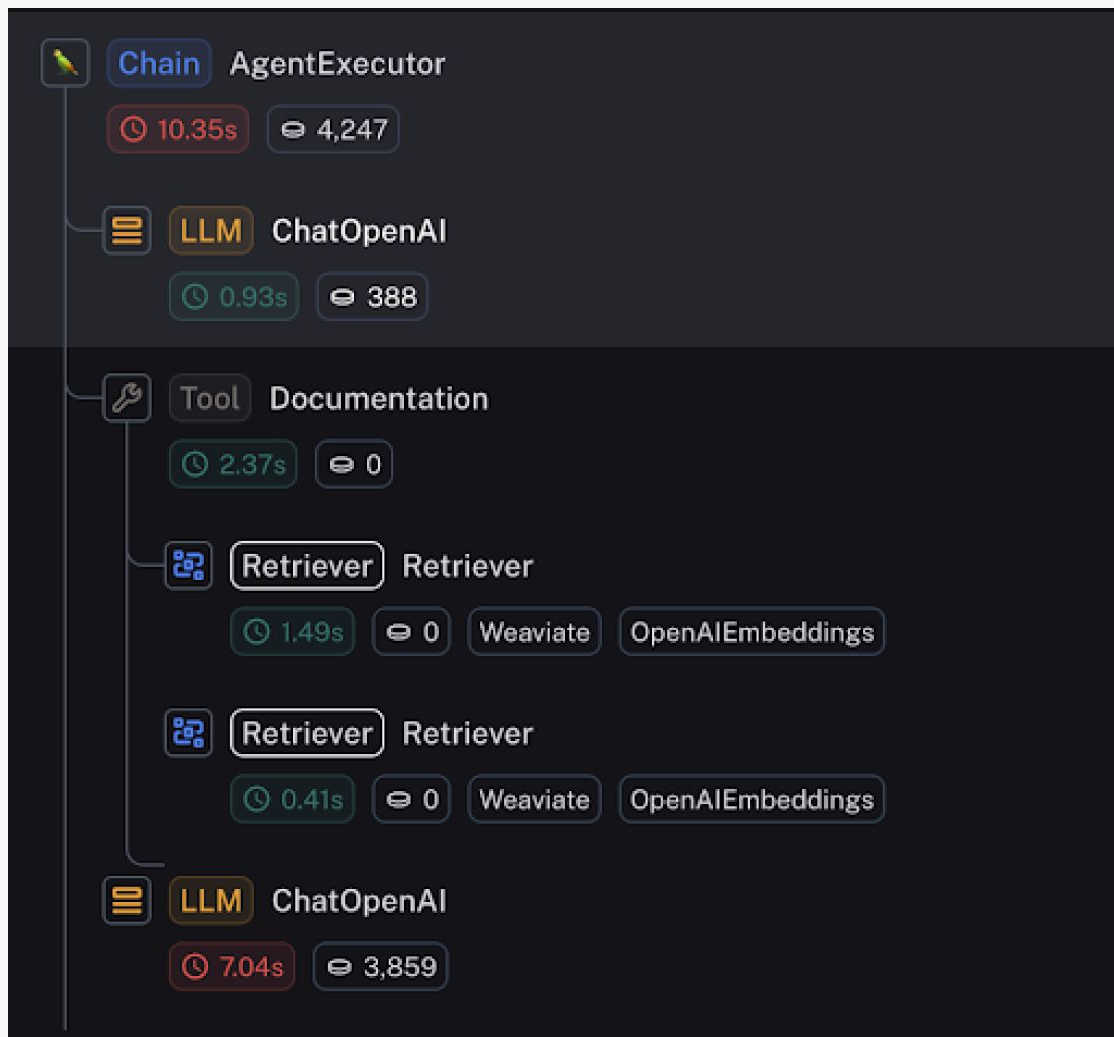
LLMs

		Decide What Output Is	Decide What Steps to Take	Define State Transitions	
HUMAN-DRIVEN	1	Code			
	2	LLM Call		 <i>one step only</i>	
	3	Chain		 <i>multiple steps</i>	
	4	Router			 <i>no cycles</i>
AGENT-EXECUTED	5	State Machine			 <i>cycles</i>
	6	Agent			

# Challenges

- Orchestration
- Data Engineering
- Prompt Engineering
- Debugging
- Evaluation
- Collaboration

# Orchestration





# Data Engineering

INPUT

Copy

```
1 query: build agent
```

YAML ↕

OUTPUT


Copy

```
1 documents:
2   - metadata:
3     source:
4       https://python.langchain.com/docs/use_cases/more/agents/agents/
5     page_content: |
6       Modify Agent•The built-in LangChain agent
7       types are designed to work well in
8       generic situations, but you may be able to
9       improve performance by modifying
10      the agent implementation. There are several
11      ways you could do this:Modify the
12      base prompt. This can be used to give the
13      agent more context on how it should
14      behave, etc Modify the output parser. This is
```

# Prompt Engineering

## Prompt

TEMPLATE

 Copy

- 1 v Your goal is to create a well crafted set of answers for a test for a specific question.
- 2 Your answers will be used on a test to assess a student's knowledge. You will be given the question and your goal is to follow the output format below w/ guidelines
- 3
- 4 v Answer Choice Guidelines
- 5 Position the correct option so that it appears approximately the same number of times in each possible position for a quiz.
- 6 Answer choices should be written clearly and similarly to each other in content, length, and grammar; avoid giving clues through the use of faulty grammatical construction.
- 7 Make all distractors plausible; they should be common misconceptions that learners may have.
- 8 In answer choices, avoid "all of the above" and "none of the above," which can lead to artificially higher levels of

# Debugging

**Prompt Playground** Secrets & API Keys History Save as

**SYSTEM** 🗑️ 📄

# Prompt

**Objective:**  
Your objective is to create a sequential workflow based on the users query.

Create a plan represented in JSON by only using the tools listed below. The workflow should be a JSON array containing only the sequence index, function name and input. A step in the workflow can receive the output from a previous step as input.

**HUMAN** 🗑️

{input}

+ Add Message

**Inputs**

{output\_format}

Enter variable value...

{tools}

Enter variable value...

{input}

Enter variable value...

**Output** ChatOpenAI gpt-3.5-turbo 🗑️

Click Start to run generation...

**Settings** ✕

**Provider**  
ChatOpenAI 📄

**Model**  
gpt-3.5-turbo 📄

**Temperature** 1 📄

**Maximum Length** -1 📄

**Top P** 1 📄

**Presence Penalty** 0 📄

**Frequency Penalty** 0 📄

**Function Call**  
auto 📄 Functions

**Stop Sequences**  
+ Add stop sequence

▶ Start

# Evaluation

Test Runs Examples

# Collaboration

The image shows a dark-themed user interface for a collaboration platform. On the left is a sidebar with navigation icons and a list of filters. The main area on the right features a search bar and a grid of content cards. Each card represents a different use case or model, with a title, a brief description, and metadata like update time and download counts. A 'Try it' button is present on each card.

**Use cases**

- QA over documents 3,224
- Interacting with APIs 2,342
- Code understanding 2,111
- Agent simulations 2,058
- Agents 1,279
- Chatbots 1,229
- Agents 1,112
- Code writing 1,010
- Autonomous agents 1001
- Extraction 989
- Multi-modal 923
- Extraction 912
- Self-checking 909
- Tagging 902
- Summarization 901
- SQL 850

**Model**

- Anthropic: Claude-2 242
- Meta: Llama-2-13b-chat 489
- OpenAI: gpt-3.5-turbo 1093

**Type**

- ChatPromptTemplate 5,234
- StringPromptTemplate 6,732

**Search** Search prompts, use cases, models...

**Recently Updated** **Top Favored** **Top Downloaded** **Top Viewed**

**Summarization** **Try it**

**hardkothari/prompt-maker**  
Convert your small and lazy prompt into a detailed and better prom...  
(-) ChatPrompt · Updated 3 days ago · ❤️ 26k · ↓ 52k

**Code Understanding** **Try it**

**jacob/langchain-tsdoc-comment-generation**  
Outputs TSDoc comments according to a schema, given some inpu...  
(\*) Prompt · Updated 3 days ago · ❤️ 126k · ↓ 52k

**Extraction** **Try it**

**langchain/entity-memory-extractor**  
Outputs TSDoc comments according to a schema, given some inpu...  
(\*) Prompt · Updated 3 days ago · ❤️ 126k · ↓ 52k

**Summarization** **Try it**

**hardkothari/tweet-from-text**  
Generate twitter threads from a text block.  
(\*) Prompt · Updated 3 days ago · ❤️ 126k · ↓ 52k

**Summarization** **Try it**

**hardkothari/tweet-from-text**  
Generate twitter threads from a text block.



**Thank you!**