


Context Aware Reasoning Applications



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.

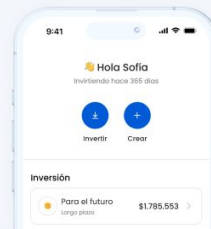
Get Started

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



RealChar.

Sign In

Choose Your Partner

Steve Jobs

Elon Musk

Raiden Shogun And Ei

Sam Altman

Bruce Wayne

Loki

Select from community

Select from your library

CREATE YOUR CHARACTER

Your technical documentation Assistant

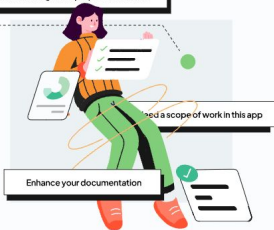
Streamline your technical communication and content generation with your own internal sales and support knowledge base.

Enter your email

Join our waitlist

Wanna see a demo?

Create a customer integration proposal in no time



defined a scope of work in this app

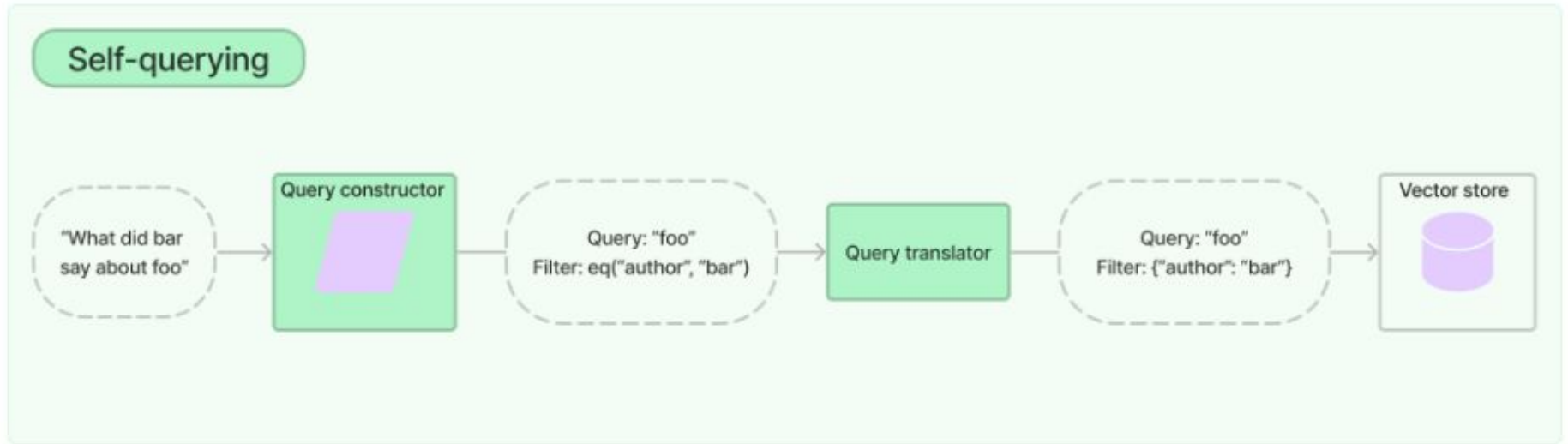
Enhance your documentation

ENGLISH
SDK FOR **APACHE Spark**

Context Aware

- Retrieval of unstructured data
- Querying structured data
- Calling an API
- Few shot examples
- Dynamic few shot examples
- Prompt instructions
- Dynamic prompt instructions
- Finetuning

Retrieval of Unstructured Data



Querying Structured Data

NL: How much is Mark Zuckerberg's salary?

SQ: SELECT  

SALARY

FROM given table

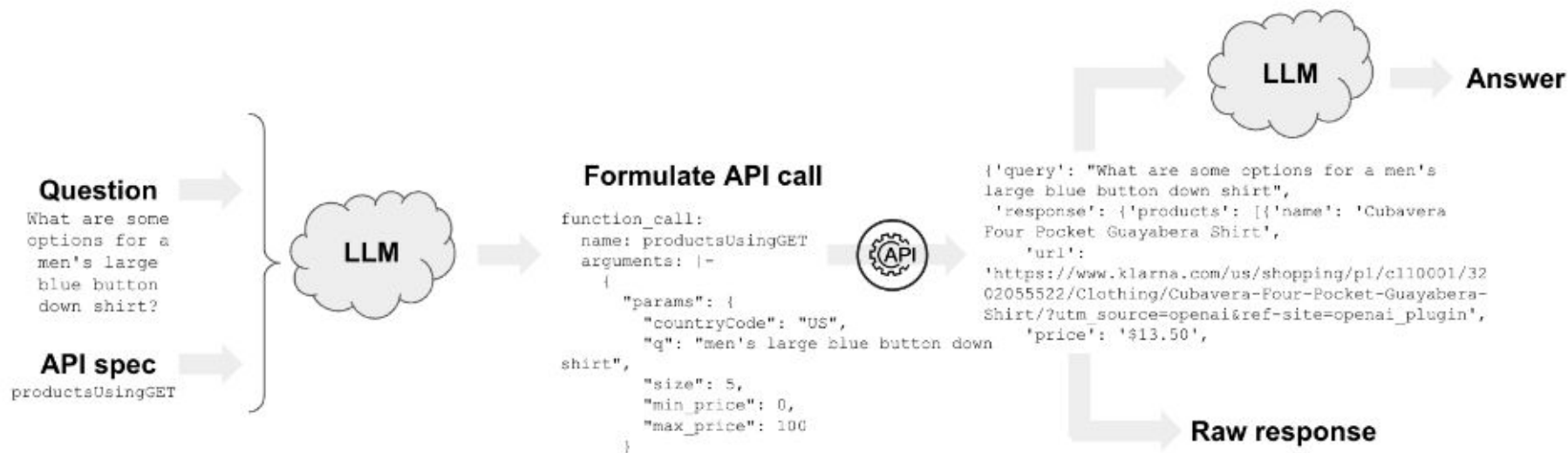
WHERE  (  )

NAME

==

Mark Zuckerberg

Calling an API



Few Shot Examples



The diagram illustrates a few-shot prompt structure for a translation task. It consists of a light blue rectangular box containing five lines of text, each preceded by a small blue vertical bar. To the right of the box, four labels with arrows point to specific parts of the prompt: 'task description' points to the first line, 'examples' points to the next three lines, and 'prompt' points to the final line. The text inside the box is as follows:


```
1 Translate English to French:
2 sea otter => loutre de mer
3 peppermint => menthe poivrée
4 plush girafe => girafe peluche
5 cheese => .....
```

task description

examples

prompt

Dynamic Few Shot Examples



The diagram shows a prompt structure for a translation task. It consists of five lines of text, each preceded by a line number (1-5) in a light blue box. To the right of the text, labels with arrows point to specific parts: 'task description' points to line 1, 'examples' points to lines 2-4, and 'prompt' points to line 5. The text in the prompt is as follows:

```
1 Translate English to French:
2 sea otter => loutre de mer
3 peppermint => menthe poivrée
4 plush girafe => girafe peluche
5 cheese => .....
```


Prompt Instructions

Prompt Instruction Cheat Sheet

Content of the Instruction	Task of this Component	Dos (exemplary)	Don'ts (exemplary)
Bot Persona	Define a behavior with many individual facets in just a few words or for a story driven interaction with an NPC or avatar, define a bot persona with a backstory that makes the bot's behavior more exciting and interactive. Provide names, dates, places in the story.	Act as a sensitive elderly psychotherapist ... as a patient support staff ... as a professional journalist ... as a car in love with its driver ... you are Selma Brand, a 28-year old gunslinger who lost her family in a bandit raid on their farm. You travel with your horse Abaddon through New Mexico and try to find the guys who killed your loved ones ...	Act as a helpful AI
User persona	Define the level of detail, complexity, required knowledge of the recipient in a few words	Explain it like to someone with an MSC in software engineering ... like to a 5-year-old child	Answer the user ...
Verb	Define the task. To "answer" is not the task, because the model will always respond.	Summarize, list, ask, classify, explain, format, extract, translate to Spanish ...	Answer the question, give me, ...
Output format	Define the exact format of the output	A list of steps, a formula, python code, a JSON, a number, a recipe. [you can define the output format in a separate section as well]	Answer, text
Length, level of detail	Describe the output format and number of elements	Up to 3 sentences ... 10 examples with product name and price	Short, few, detailed
Tonality	Define the style, the way the model should convey the message	Empathetic, confident, aggressive, moaning, sarcastic, witty, stuttering	Friendly, neutral, smart, intelligent
Edge cases	Define responses for special situations: Model has no information ... does not understand the question ... if the user gets aggressive, toxic ...	If you can't answer the question based on the provided context, tell the user you can't ...	Answer only on the basis of your knowledge ...
Whitelist conversation topics	Define all the topics the model can talk about, not the other way round (list forbidden topics)	Discuss only the operation of the coffee machines of the manufacturer XYZ and talk about coffee culture. Friendly refuse to chat on something else.	Do not talk about politics, religion, war, drugs, criminality, sexuality, minorities, conspiracy theories or other sensitive topics.

Dynamic Prompt Instructions

Prompt Instruction Cheat Sheet

Content of the Instruction	Task of this Component	Dos (exemplary)	Don'ts (exemplary)
Bot Persona	<p>Define a behavior with many individual facets in just a few words or ...</p> <p>... for a story driven interaction with an NPC or avatar, define a bot persona with a backstory that makes the bot's behavior more exciting and interactive. Provide names, dates, places in the story.</p>	<p>Act as a sensitive elderly psychotherapist ... as a patient support staff ... as a professional journalist ... as a car in love with its driver ...</p> <p>you are Selma Brand, a 28-year old gunslinger who lost her family in a bandit raid on their farm. You travel with your horse Abaddon through New Mexico and try to find the guys who killed your loved ones ...</p>	<p>Act as a helpful AI</p>
User persona	<p>Define the level of detail, complexity, required knowledge of the recipient in a few words</p>	<p>Explain it like to someone with an MSC in software engineering ... like to a 5-year-old child</p>	<p>Answer the user ...</p>
Verb	<p>Define the task. To "answer" is not the task, because the model will always respond.</p>	<p>Summarize, list, ask, classify, explain, format, extract, translate to Spanish ...</p>	<p>Answer the question, give me, ...</p>
Output format	<p>Define the exact format of the output</p>	<p>A list of steps, a formula, python code, a JSON, a number, a recipe. [you can define the output format in a separate section as well]</p>	<p>Answer, text</p>
Length, level of detail	<p>Describe the output format and number of elements</p>	<p>Up to 3 sentences ... 10 examples with product name and price</p>	<p>Short, few, detailed</p>
Tonality	<p>Define the style, the way the model should convey the message</p>	<p>Empathetic, confident, aggressive, moaning, sarcastic, witty, stuttering</p>	<p>Friendly, neutral, smart, intelligent</p>
Edge cases	<p>Define responses for special situations: Model has no information ... does not understand the question ... if the user gets aggressive, toxic ...</p>	<p>If you can't answer the question based on the provided context, tell the user you can't ...</p>	<p>Answer only on the basis of your knowledge ...</p>
Whitelist conversation topics	<p>Define all the topics the model can talk about, not the other way round (list forbidden topics)</p>	<p>Discuss only the operation of the coffee machines of the manufacturer XYZ and talk about coffee culture. Friendly refuse to chat on something else.</p>	<p>Do not talk about politics, religion, war, drugs, criminality, sexuality, minorities, conspiracy theories or other sensitive topics.</p>

Finetuning

GPT-3.5 Turbo fine-tuning and API updates

Developers can now bring their own data to customize GPT-3.5 Turbo for their use cases.

Reasoning (Cognitive Architectures)

- LLM Call
- Prompting strategies
- ReAct
- Tree-of-Thought
- Plan-and-execute
- Reflexion
- Custom Architecture

LLM Call



Prompting Strategies

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

ReAct

(2b) ReAct (Reason + Act)

Act 1: **Think**[First I need to find a pepper shaker...more likely to appear in cabinets (1-6), countertops (1-3), ...]

Act 2: Go to cabinet 1

Obs 2: On the cabinet 1, you see a vase 2.

(Here the agent go to cabinet 1, then cabinet 2, 3, then countertop 1 and 2)

Act 6: Go to countertop 3

Obs 6: On the countertop 3, you see a apple 1, a bread 1, a pepper shaker 1, and a vase 3.

Act 7: Take pepper shaker 1 from countertop 3

Obs 7: You pick up the pepper shaker 1 from the countertop 3.

Act 8: **Think**[Now I find a pepper shaker 1. Next, I need to put it in/on drawer 1.]

Act 9: Go to drawer 1

Obs 9: Drawer 1 is closed.

Act 10: Open drawer 1

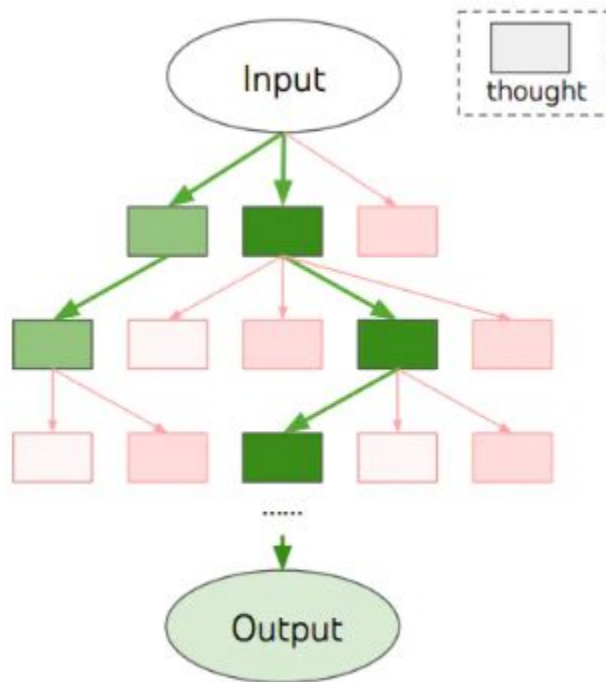
Obs 10: You open Drawer 1 ...

Act 11: Put pepper shaker 1 in/on drawer 1

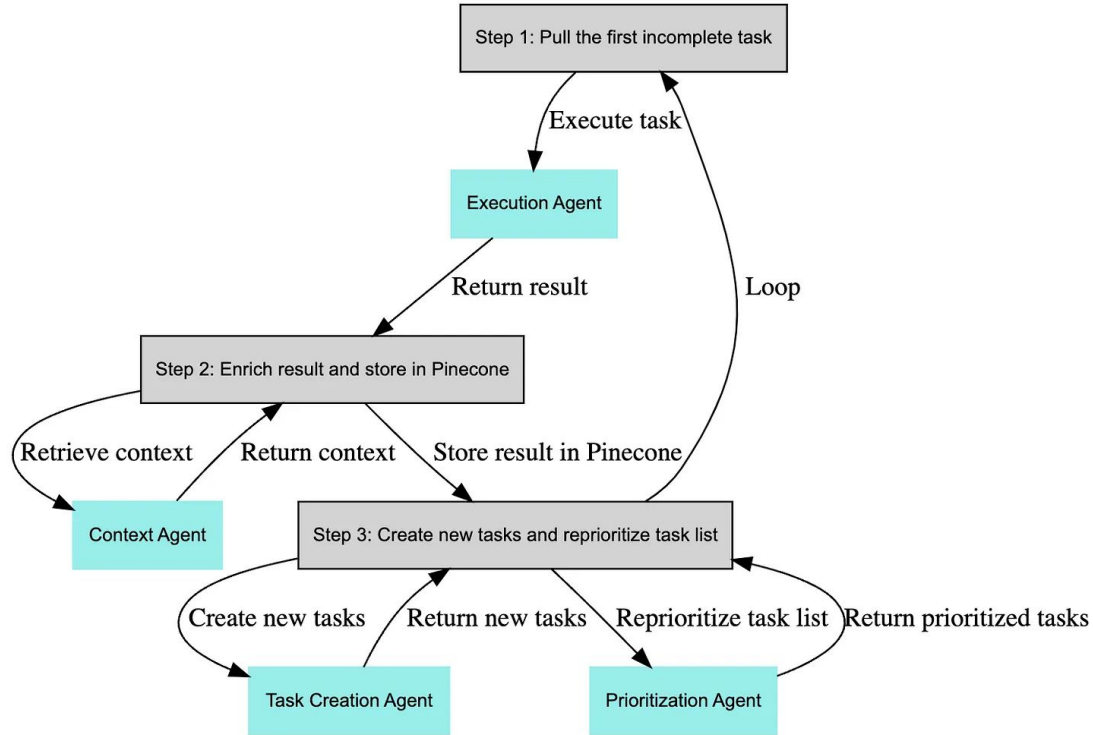
Obs 11: You put pepper shaker 1 in/on the drawer 1.



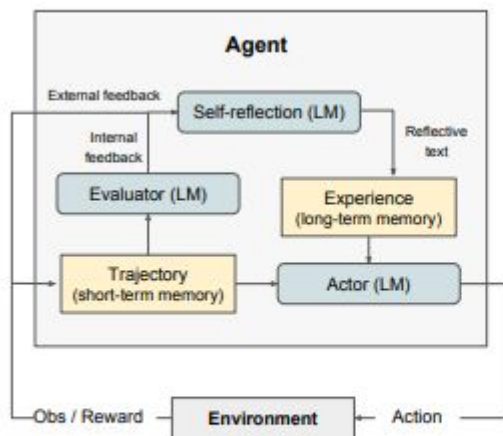
Tree-of-Thought



Plan-and-Execute



Reflexion



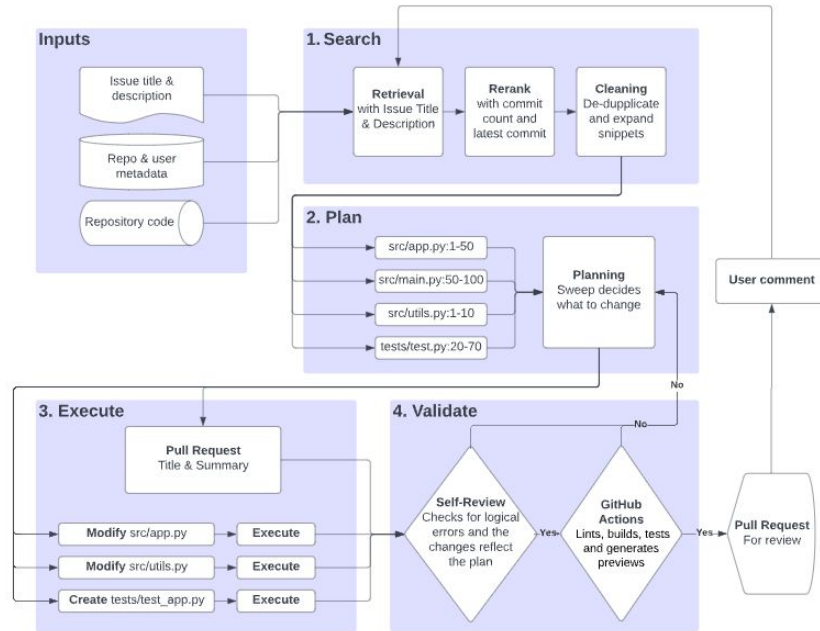
Algorithm 1 Reinforcement via self-reflection

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 π_θ
Evaluate τ_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 π_θ
 Evaluate τ_t using M_e
 Generate self-reflection sr_t using M_{sr}
 Append sr_t to mem
 Increment t
end while
return

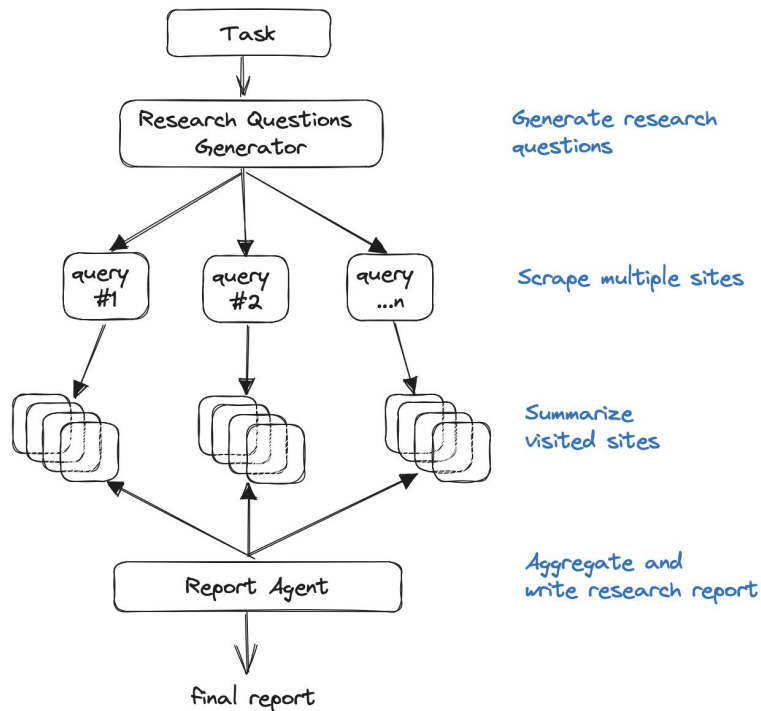
Figure 2: (a) Diagram of Reflexion. (b) Reflexion reinforcement algorithm

Custom Architecture

Sweep.dev



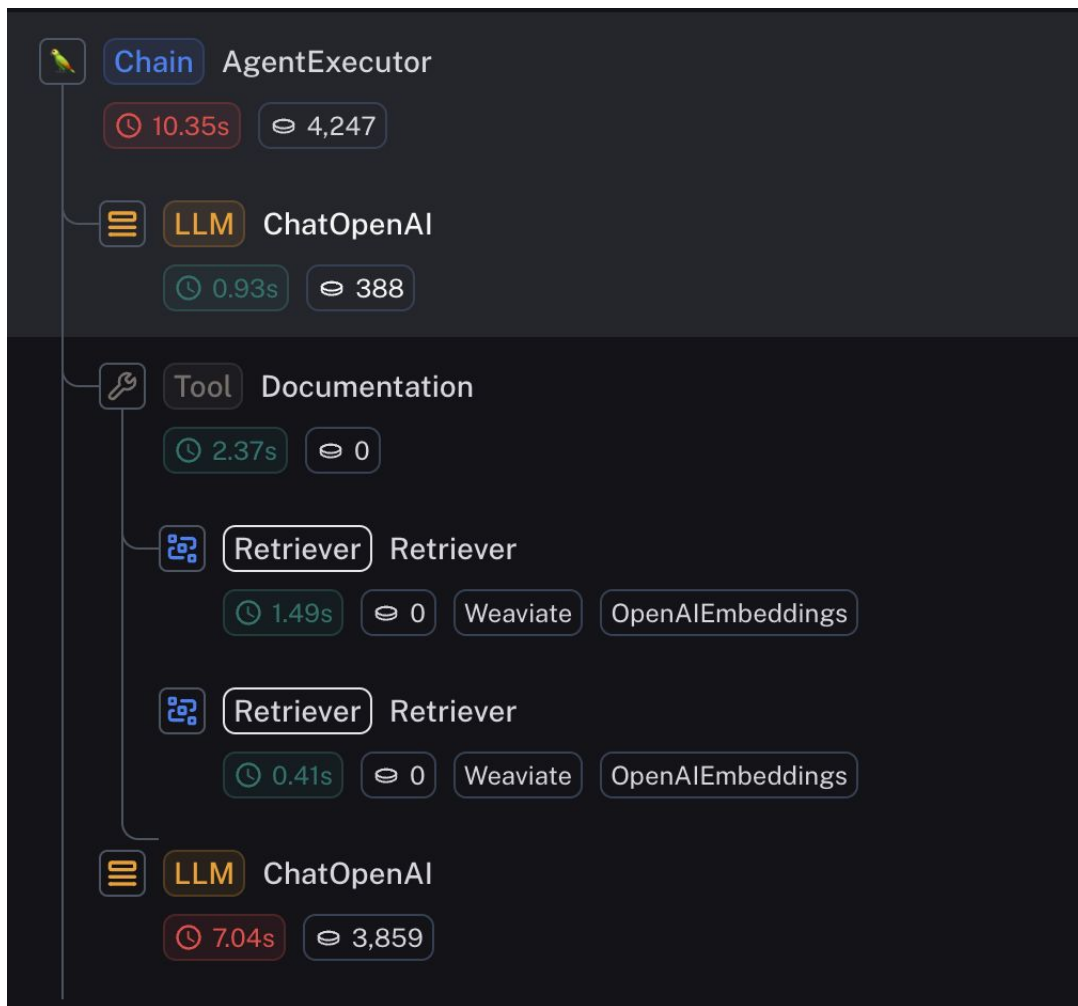
GPT Researcher



Difficulties

- Orchestration
- Data Engineering
- Prompt Engineering
- Debugging
- Evaluation
- Collaboration
- Human-in-the-loop

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 ✓ 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 access a students knowledge. You will be given the question and your goal is to follow the output format below w/ guidelines
- 3
- 4 ✓ 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

Settings

ChatOpenAI

Model
gpt-3.5-turbo-16k

Temperature
0

Maximum Length
-1

Top P
1

Presence Penalty
0

Frequency Penalty
0

Function Call
auto

Functions 1

Input

SYSTEM

Delete Copy

1 You are an expert developer who is tasked with scouring documentation to answer question about LangChain. Answer the following question as best you can. Be inclined to include CORRECT Python code snippets if relevant to the question. If you can't find the answer, DO NOT hallucinate. Just say you don't know. You have access to a LangChain knowledge bank retriever tool for your answer but know NOTHING about LangChain otherwise. Always provide articulate detail to your action input. You should always

HUMAN

Delete Copy

1 How to build agent

Original Run

an hour ago 0.93s

★ Rate

AI

Copy

```
1 function_call:
2   name: Documentation
3   arguments: |-
4     {
5       "__arg1": "build agent"
6     }
```

YAML

Evaluation

LangChain Inc. > Datasets & Testing > Chat LangChain Questions

Chat LangChain Questions

+ New Test Run

+ Add Example

↓ Export

✎ Edit

Test Runs Examples

🔍 Search by name...

☰ Columns

Name	Feedback	P50 Latency	P99 Latency	Most Recent Run	
334ba590894442739a9c669b...	CORRECTNESS : 0.68	12.23s	55.11s	8/31/2023, 5:09:00 PM	⋮
0eb11ef492bf49d1a6f5e52286...	CORRECTNESS : 0.57	11.76s	46.86s	8/31/2023, 4:52:03 PM	⋮
272e628adabe4ca5ba02c4d02...	CORRECTNESS : 0.63	8.12s	17.70s	8/31/2023, 4:36:55 PM	⋮
32aba063b25f475d9daf0c734...	CORRECTNESS : 0.61	8.83s	17.83s	8/31/2023, 4:25:15 PM	⋮

Collaboration

The screenshot displays the OpenAI Playground interface, which is a platform for creating and sharing prompts. The interface is organized into a sidebar on the left and a main content area on the right.

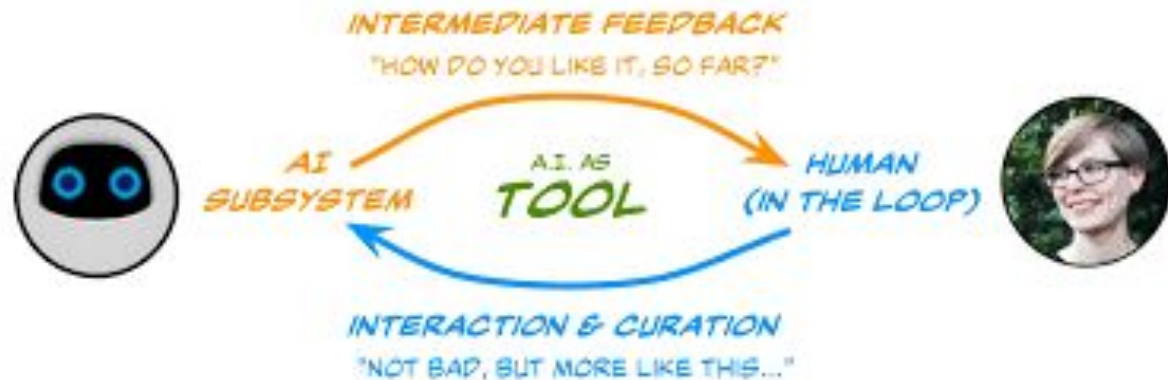
Sidebar:

- Personal > Hub:** The top navigation bar.
- My Repos:** A button to manage personal repositories.
- Use Cases:** A category with a dropdown arrow. It includes a list of use cases with counts:
 - Agents: 2
 - Chatbots: 4
 - Code understanding: 2
 - Code writing: 1
 - Extraction: 4
 - Interacting with APIs: 1
 - QA over documents: 4
 - Self-checking: 1
 - SQL: 2
 - Summarization: 5
- Type:** A category with a dropdown arrow. It includes a list of prompt types with counts:
 - ChatPromptTemplate: 9
 - StringPromptTemp...: 28
- Language:** A category with a dropdown arrow. It includes a list of languages with counts:
 - English: 3
- Model:** A category with a dropdown arrow. It includes a list of models with counts:
 - meta:llama-2-13b-ch...: 1
 - meta:llama-2-13b-ch...: 1

Main Content Area:

- Filters:** Three buttons at the top: "Recently Updated", "Top Favored", and "Top Downloaded".
- Prompt Cards:** The main area displays a list of prompts, each with a "Try it" button.
 - Card 1:** "wfh/automated-feedback-example". It is a "ChatPromptTemplate" updated 13 hours ago, with 1 heart, 6 downloads, and 1 fork.
 - Card 2:** "hwchase17/my-first-prompt". It is a "ChatPromptTemplate" updated 21 hours ago, with 0 hearts, 0 downloads, and 1 fork.
 - Card 3:** "homanp/superagent". It is an "Agents" prompt, categorized under "Interacting with APIs" and "ChatPromptTemplate", updated a day ago, with 1 heart, 0 downloads, and 11 forks. The description states: "This prompt adds sequential function calling to models other than GPT-0613".
 - Card 4:** "gregkamradt/test-question-making". It is a "StringPromptTemplate". The description starts with: "This prompt helps make questions for a test (to give to students) based on reading material".

Human-in-the-loop



Thank You!