

A Short Lecture About AI Agents

Apr 16, 2025

Presented by Shaohui Wang

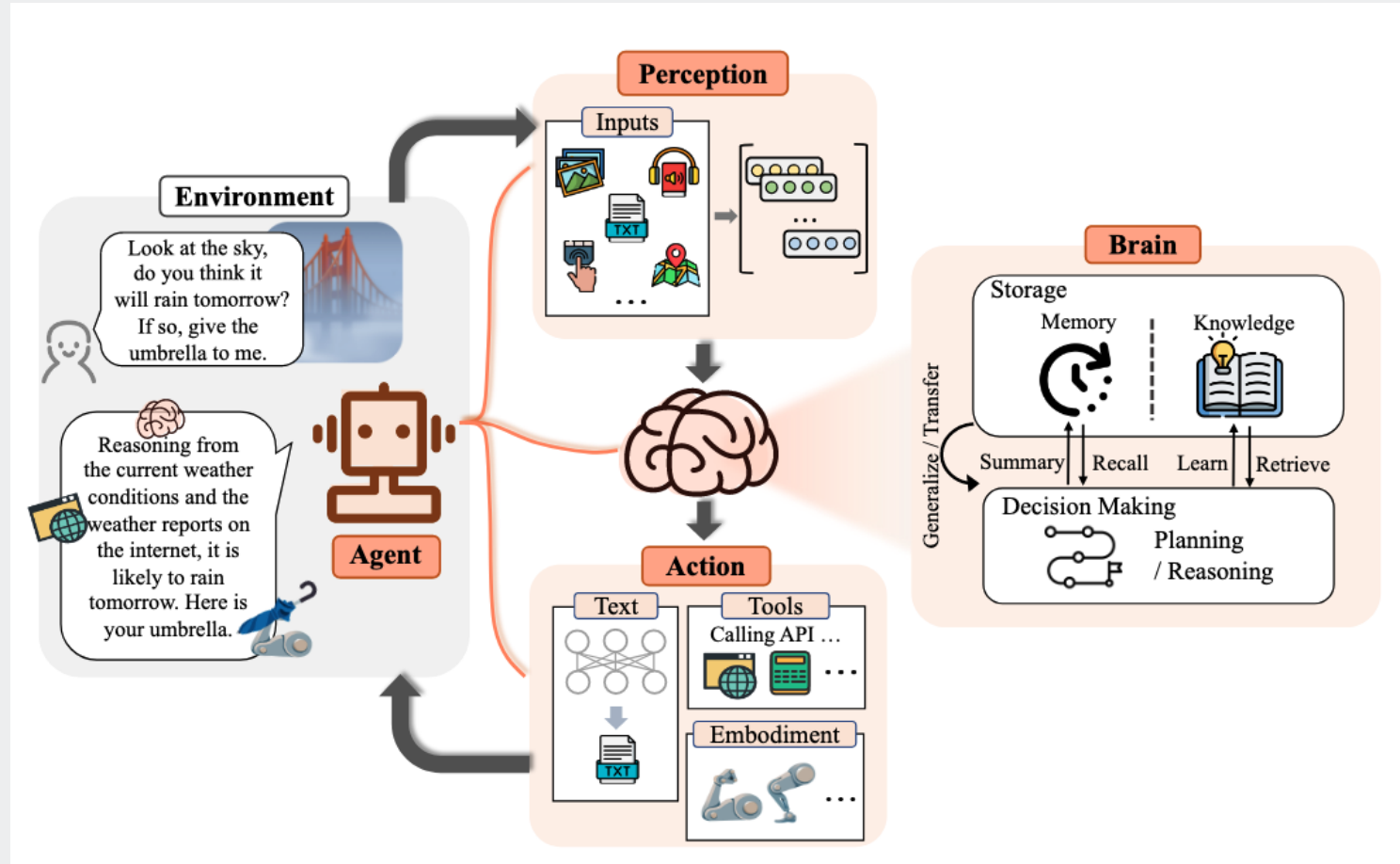
What are AI Agents



```
Welcome to
EEEEEE LL      IIII  ZZZZZZ  AAAAA
EE      LL      II    ZZ      AA  AA
EEEEEE LL      II    ZZ      AAAAAA
EE      LL      II    ZZ      AA  AA
EEEEEE LLLLLL  IIII  ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
```



Xi, Zhiheng, Wenxiang Chen, Xin Guo, Wei He, Yiwon Ding, Boyang Hong, Ming Zhang, et al. "The Rise and Potential of Large Language Model Based Agents: A Survey." arXiv, September 19, 2023.
<https://doi.org/10.48550/arXiv.2309.07864>.

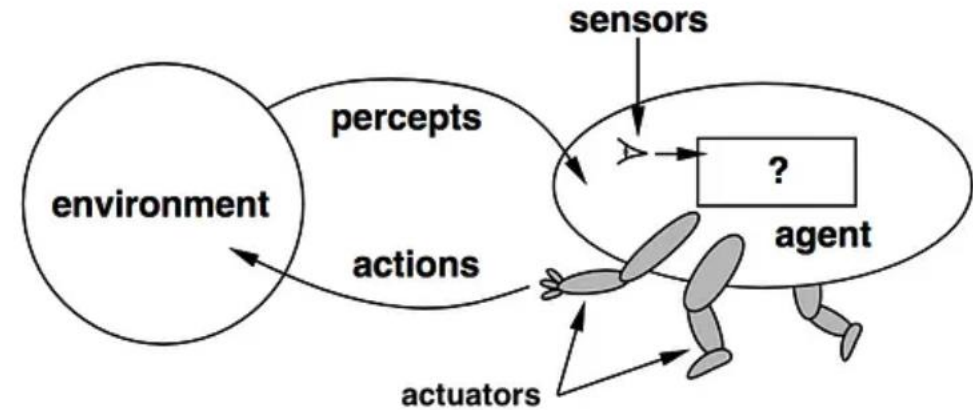
Two Competing Views of LLM Agents

LLM-first view: We make an LLM into an agent!

Implications: scaffold on top of LLMs, prompting-focused, heavy on engineering

Agent-first view: We integrate LLMs into AI agents so they can use language for reasoning and communication!

Implications: All the same challenges faced by previous AI agents (e.g., perception, reasoning, world models, planning) still remain, but we need to reexamine them through the new lens of LLMs and tackle new ones (e.g., synthetic data, self-reflection, internalized search)



An agent interact its environment through its sensors and actuators



Agent: A bomb-disposal robot designed for DARPA

“An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators”

-- Russel & Norvig, AI: A Modern Approach

Agents in Virtual Sandbox Environments

Generative Agents: Interactive Simulacra of Human Behavior

Joon Sung Park
Stanford University
Stanford, USA
joonspk@stanford.edu

Meredith Ringel Morris
Google DeepMind
Seattle, WA, USA
merrie@google.com

Joseph C. O'Brien
Stanford University
Stanford, USA
jobrien3@stanford.edu

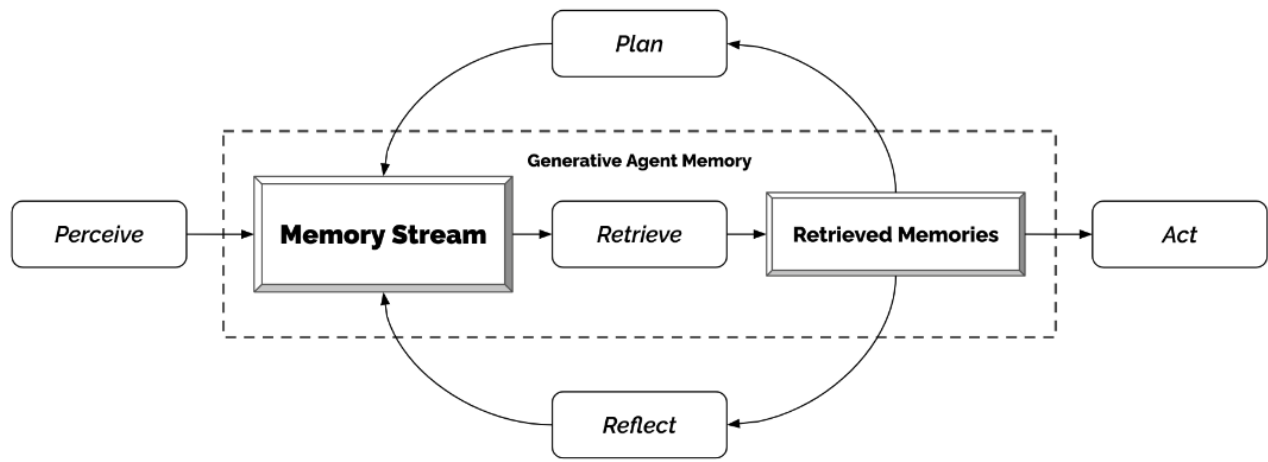
Percy Liang
Stanford University
Stanford, USA
pliang@cs.stanford.edu

Carrie J. Cai
Google Research
Mountain View, CA, USA
cjcai@google.com

Michael S. Bernstein
Stanford University
Stanford, USA
msb@cs.stanford.edu



Figure 4: At the beginning of the simulation, one agent is initialized with an intent to organize a Valentine's Day party. Despite many possible points of failure in the ensuing chain of events—agents might not act on that intent, might forget to tell others, might not remember to show up—the Valentine's Day party does, in fact, occur, with a number of agents gathering and interacting.



Simulator



Video clip 1

... but I'm gonna go around gathering a little bit more wood from these trees.

Aligned text 1



Video clip 2

... I'm also going to gather a little bit of stone from the side of this little hill here.

Aligned text 2



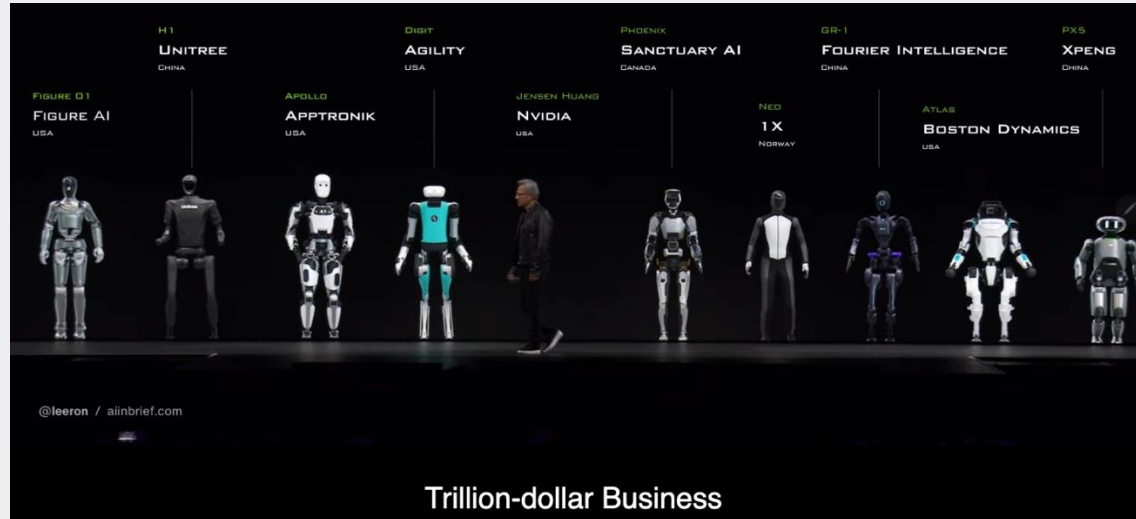
Video clip 3

As I raised my axe in front of this pig, there's only one thing you know is gonna happen...

Aligned text 3

Agents in Physical Environments

Embodied AI



Trillion-dollar Business



Tesla Optimus



Boston Dynamics e-Atlas



Figure F.02

Society Simulation with LLM-based Agents

Blind Judgement: Agent-Based Supreme Court Modelling With GPT

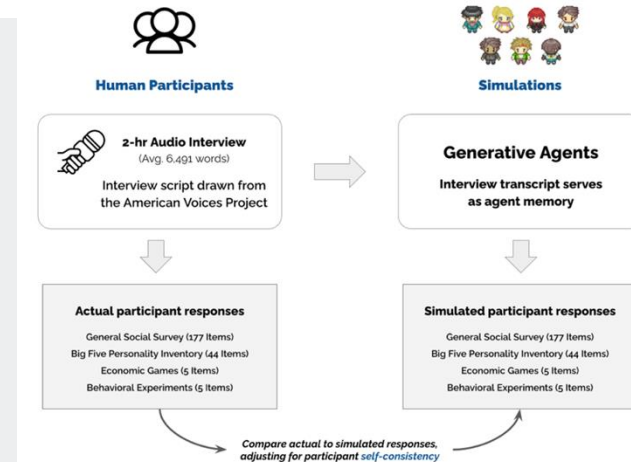
Sil Hamilton

McGill University
sil.hamilton@mcgill.ca

Justice	Accuracy	κ
Samuel Alito	65%	0.30
Ruth Bader Ginsburg	62%	0.21
Clarence Thomas	59%	0.18
Stephen Breyer	58%	0.16
John Roberts	57%	0.13
Elena Kagan	56%	0.12
Anthony Kennedy	54%	0.09
Sonia Sotomayor	51%	0.00
Antonin Scalia	50%	-0.03

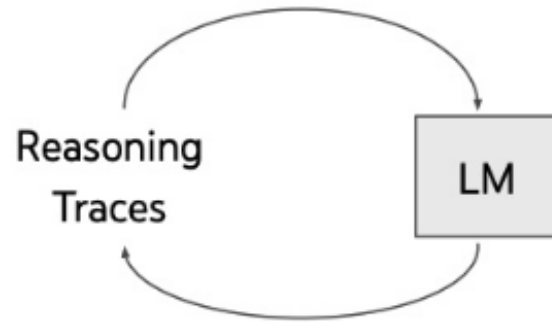
Generative Agent Simulations of 1,000 People

Authors: Joon Sung Park^{1*}, Carolyn Q. Zou^{1,2}, Aaron Shaw², Benjamin Mako Hill³, Carrie Cai⁴, Meredith Ringel Morris⁵, Robb Willer⁶, Percy Liang¹, Michael S. Bernstein¹



Reasoning and Acting

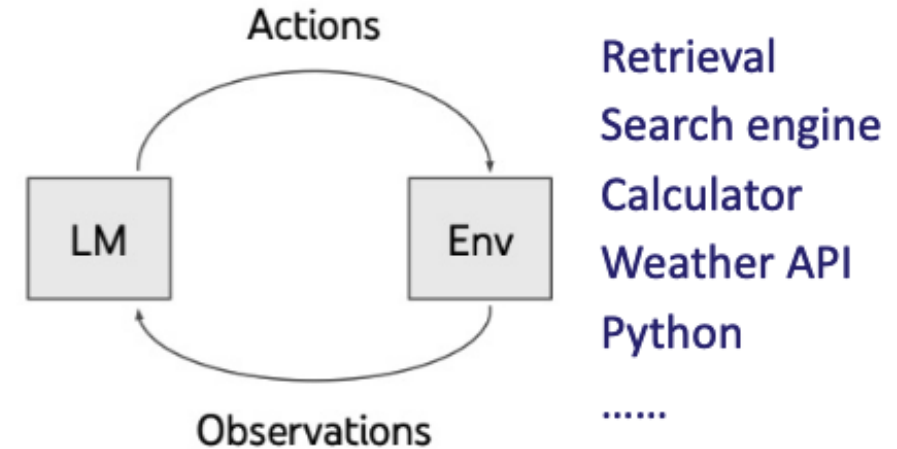
CoT



Flexible and general to
augment test-time compute

Lack of external knowledge and tools

RAG/Retrieval/Code/Tool use



Lack of reasoning

Flexible and general to
augment knowledge,
computation, feedback, etc.

Yao, Shunyu, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. "ReAct: Synergizing Reasoning and Acting in Language Models." arXiv, March 10, 2023. <https://doi.org/10.48550/arXiv.2210.03629>.

Chain-of-Thought Prompting

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei Xuezhi Wang Dale Schuurmans Maarten Bosma
Brian Ichter Fei Xia Ed H. Chi Quoc V. Le Denny Zhou

Google Research, Brain Team
{jasonwei,dennyzhou}@google.com

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. ✅

Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Large Language Models are Zero-Shot Reasoners

Takeshi Kojima
The University of Tokyo
t.kojima@weblab.t.u-tokyo.ac.jp

Shixiang Shane Gu
Google Research, Brain Team

Machel Reid
Google Research*

Yutaka Matsuo
The University of Tokyo

Yusuke Iwasawa
The University of Tokyo

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 ❌

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✅

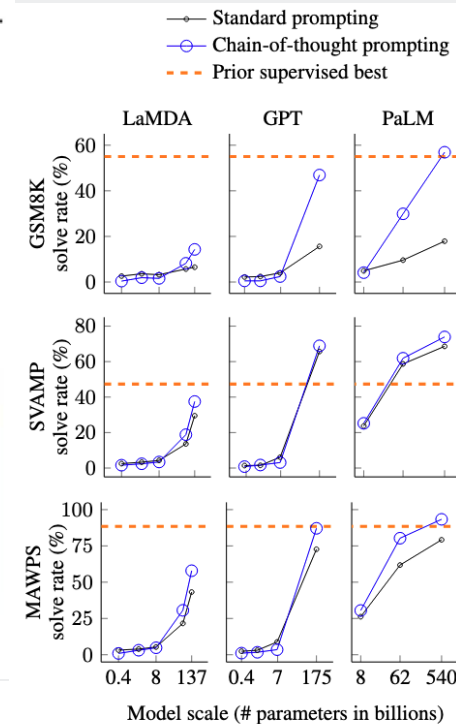
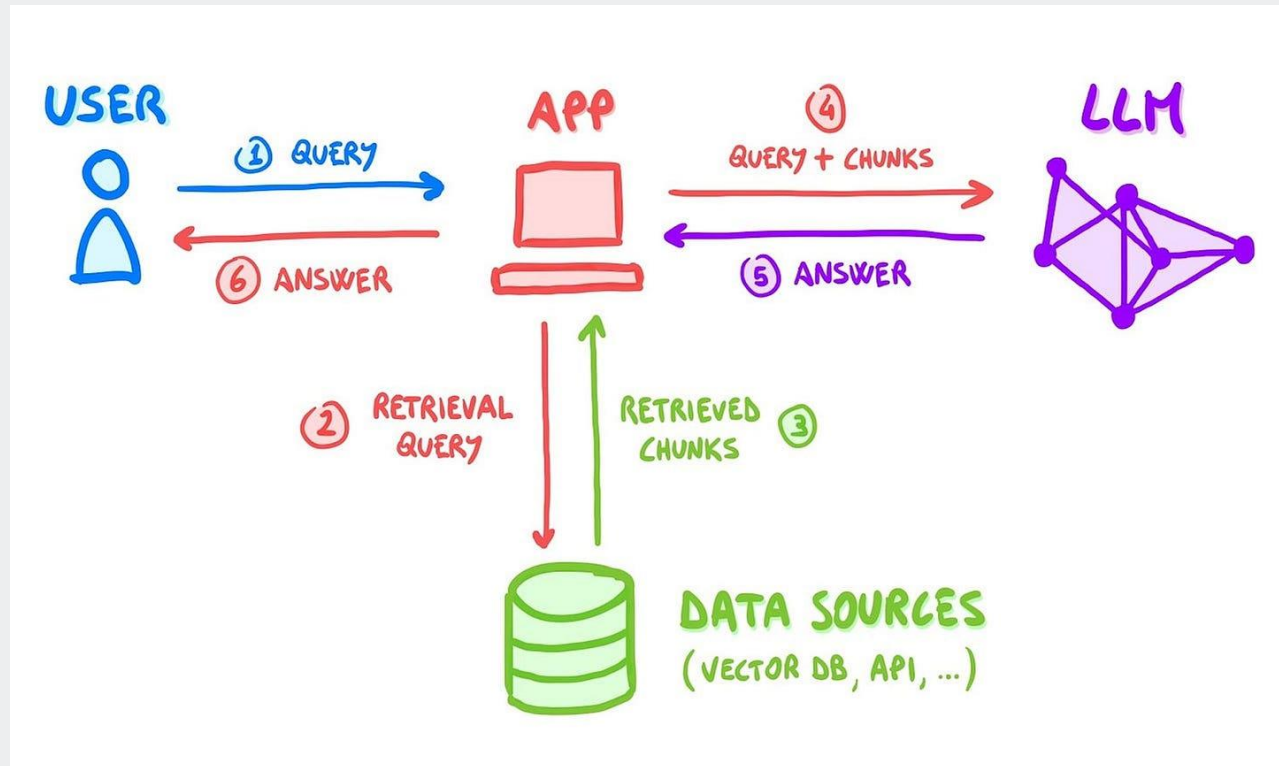


Figure 4: Chain-of-thought prompting enables large language models to solve challenging math problems. Notably, chain-of-thought reasoning is an emergent ability of increasing model scale. Prior best numbers are from Cobbe et al. (2021) for GSM8K, Jie et al. (2022) for SVAMP, and Lan et al. (2021) for MAWPS.

Lab 1: Let's Build a Naive Retrieval-Augmented Generation (RAG) System from Scratch



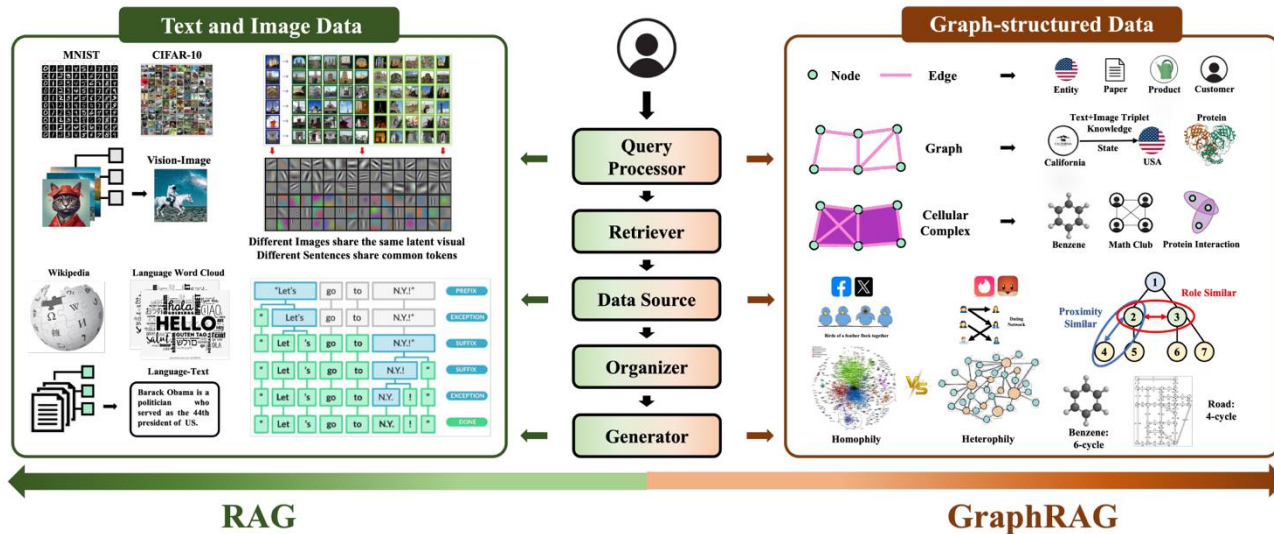
<https://1drv.ms/u/s!Aufi3o6YPJkzrvNDaYHkjMBhvmuq1w?e=A8kvUP>

- Cost-efficient AI implementation and Scalling
- Access to current domain-specific data
- Lower risk of AI Hallucination
- Enhanced developer control and model maintenance
- Greater data security

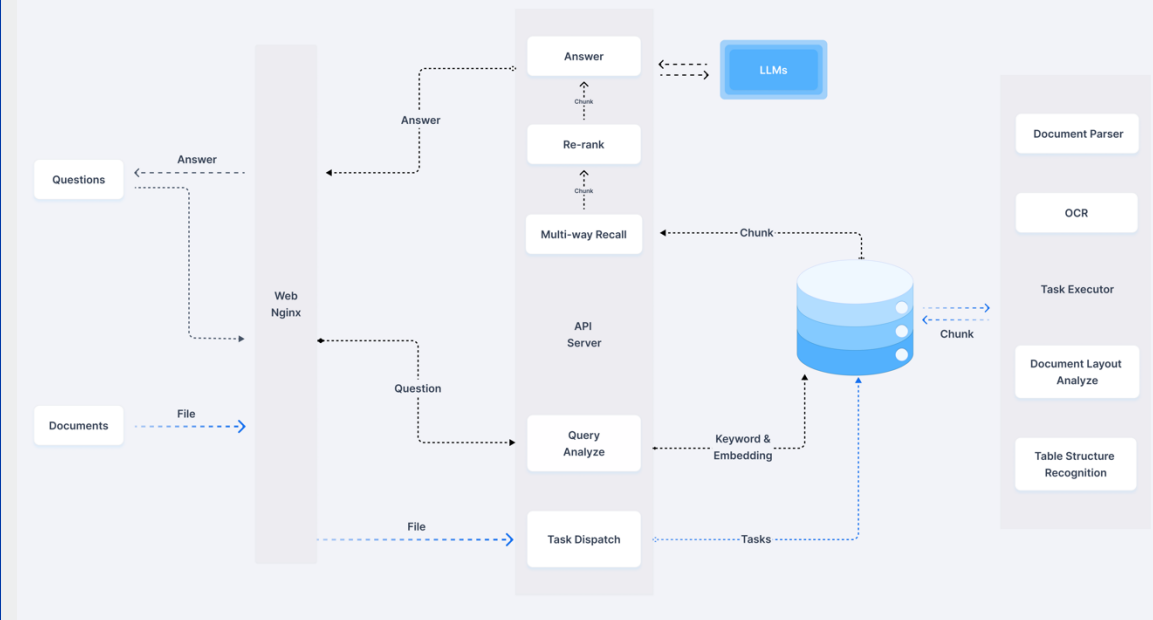
More Advanced RAG

Retrieval-Augmented Generation with Graphs (GraphRAG)

Haoyu Han¹, Yu Wang², Harry Shomer¹, Kai Guo¹, Jiayuan Ding⁵, Yongjia Lei², Mahantesh Halappanavar³, Ryan A. Rossi⁴, Subhabrata Mukherjee⁵, Xianfeng Tang⁶, Qi He⁶, Zhigang Hua⁷, Bo Long⁷, Tong Zhao⁸, Neil Shah⁸, Amin Javari⁹, Yinglong Xia⁷, Jiliang Tang¹
¹Michigan State University, ²University of Oregon, ³Pacific Northwest National Laboratory
⁴Adobe Research, ⁵Hippocratic AI, ⁶Amazon, ⁷Meta, ⁸Snap Inc., ⁹The Home Depot,
{hanhaoy1, shomerha, guokai1, tangjili}@msu.edu,



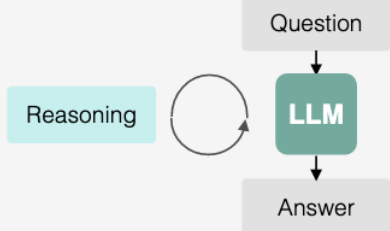
Knowledge Graph RAG



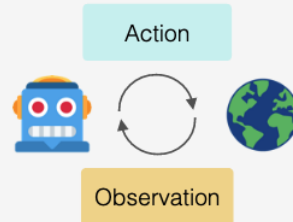
Multi-hop Reasoning RAG

Combining Reasoning and Acting

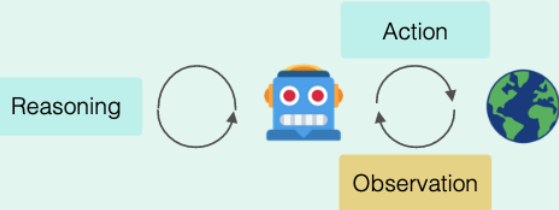
Reasoning (update internal belief)



Acting (obtain external feedback)



ReAct: a new paradigm of agents that **reason and act**



- **Synergy** of reasoning and acting
- **Simple** and intuitive to use
- **General** across domains

Tips:

This is the theory foundation of a most important MCP tool: Sequential Thinking

LLM

(GPT-4)

Thought: I need to find the current market capitalization of Apple, Nvidia, and Microsoft to understand how much it would cost to buy all of them.

Action: search["current market capitalization of Apple, Nvidia, and Microsoft"]

Observation: "Nvidia's market cap stood at \$1.832 trillion as of about 10 a.m. ET Wednesday, while Alphabet's was at \$1.814 trillion, bringing Nvidia behind just Microsoft (MSFT) at \$3.015 trillion and Apple (AAPL) at \$2.836 trillion."



(Mar 2024)

Thought: Now that I have the market cap for all companies, let's add them together and see if \$7 trillion is enough.

Action: search["1.832 trillion + 3.015 trillion + 2.836 trillion"]

Observation: 7683000000000

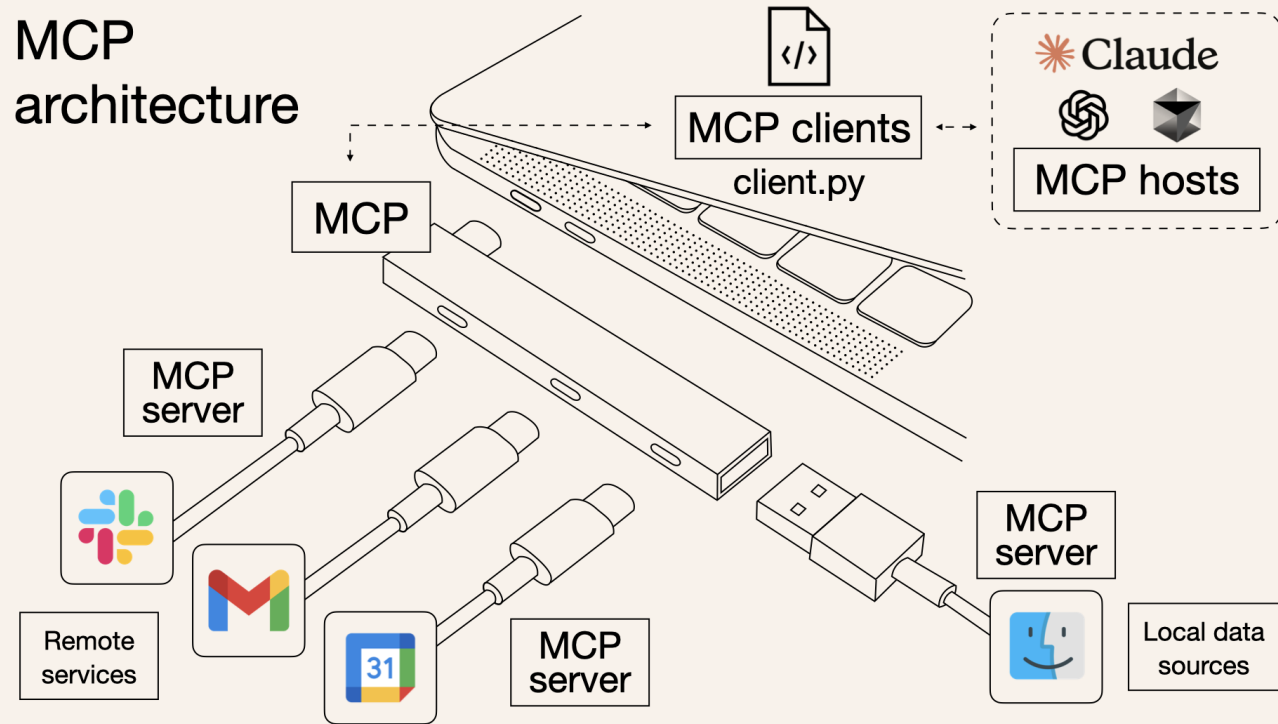
Thought: The amount of money needed in addition to the \$7 trillion is \$683 billion.

Action: finish["You need an additional \$683 billion to be able to buy Apple, Nvidia, and Microsoft."] ✓

Yao, Shunyu, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. "ReAct: Synergizing Reasoning and Acting in Language Models." arXiv, March 10, 2023. <https://doi.org/10.48550/arXiv.2210.03629>.

Model Context Protocol

MCP architecture



Example Servers:

[Filesystem](#) - Secure file operations with configurable access controls

[Git](#) - Tools to read, search, and manipulate Git repositories

[GitHub](#) - Repository management, file operations, and GitHub API integration

[Memory](#) - Knowledge graph-based persistent memory system

[Obsidian Markdown Notes](#) - Read and search through Markdown notes in Obsidian vaults

[Brave Search](#) - Web and local search using Brave's Search API

[Sequential Thinking](#) - Dynamic problem-solving through thought sequences

Complete List of official MCP Servers:

<https://github.com/modelcontextprotocol/servers>

Awesome MCP Servers:

<https://mcpservers.org/>

Introduction of MCP:

<https://modelcontextprotocol.io/introduction>

Installation MCP on Claude Desktop:

<https://modelcontextprotocol.io/quickstart/user>

Lab 2: Use Claude Desktop and MCP

Use Those MPC Servess:

[Filesystem](#) - Secure file operations with configurable access controls

[Brave Search](#) - Web and local search using Brave's Search API (You need to register an account and obtain the API Key)

[Sequential Thinking](#) - Dynamic problem-solving through thought sequences

Introduction of MCP:

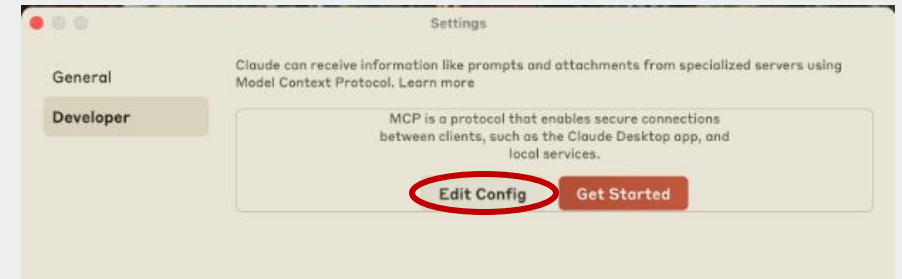
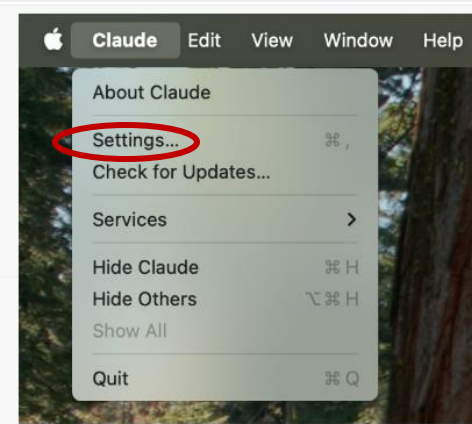
<https://modelcontextprotocol.io/introduction>

Installation MCP on Claude Desktop:

<https://modelcontextprotocol.io/quickstart/user>

Task: Search for the most recent Stock price for Apple, Microsoft, Nvidia, AMD and Intel .Then write a md file to record those prices, put the md file on my desktop. Use sequential Thinking to finish this task.

1. Install Claude Desktop <https://claude.ai/download>
2. Install Node.js `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash`
3. Add Filesystem MCP Server



```
{
  "mcpServers": {
    "filesystem": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-filesystem",
        "/Users/username/Desktop",
        "/Users/username/Downloads"
      ]
    }
  }
}
```

Mac

```
{
  "mcpServers": {
    "filesystem": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-filesystem",
        "C:\\Users\\username\\Desktop",
        "C:\\Users\\username\\Downloads"
      ]
    }
  }
}
```

Windows

<https://modelcontextprotocol.io/quickstart/user>
#windows

4. Install Brave Search <https://github.com/modelcontextprotocol/servers/tree/main/src/brave-search>
5. Install Sequential Thinking <https://github.com/modelcontextprotocol/servers/tree/main/src/sequentialthinking>

Lab 2

A Search for the most recent Stock price for Apple, Microsoft, Nvidia, AMD and Intel .Then write a md file to record those prices, put the md file on my desktop. Use sequential Thinking to finish this task.

I'll help you search for the most recent stock prices for Apple, Microsoft, NVIDIA, AMD, and Intel, then create a markdown file with this information and save it to your desktop. Let me break this down into steps using sequential thinking.

View result from sequentialthinking from sequential-thinking (local) >
View result from brave_web_search from brave-search (local) >
View result from sequentialthinking from sequential-thinking (local) >
View result from brave_web_search from brave-search (local) >
View result from sequentialthinking from sequential-thinking (local) >
View result from brave_web_search from brave-search (local) >
View result from sequentialthinking from sequential-thinking (local) >
View result from brave_web_search from brave-search (local) >
View result from sequentialthinking from sequential-thinking (local) >
View result from brave_web_search from brave-search (local) >
View result from sequentialthinking from sequential-thinking (local) >
View result from list_allowed_directories from filesystem (local) >
View result from sequentialthinking from sequential-thinking (local) >
View result from write_file from filesystem (local) >

```
{
  `path`: `/Users/apple/Desktop/tech_stock_prices.md`,
  `content`: `# Tech Company Stock Prices

**Date: April 11, 2025**
```

View result from sequentialthinking from sequential-thinking (local) >

```
{
  `thought`: `I've successfully created the markdown file "tech_stock_prices.md" or
1. A title
2. The date of the stock prices (April 11, 2025)
3. A formatted table with columns for Company, Ticker, and Price
4. All five requested companies: Apple, Microsoft, NVIDIA, AMD, and Intel
```

I've successfully retrieved the most recent stock prices for the requested tech companies and created a markdown file on your desktop:

Stock Prices - April 11, 2025

This document contains stock closing prices for selected technology companies on April 11, 2025.

Price Summary

Table4 × 6			
Company	Ticker	Price (USD)	Date
Apple	AAPL	\$197.84	April 11, 2025
Microsoft	MSFT	\$388.45	April 11, 2025
NVIDIA	NVDA	\$110.93	April 11, 2025
AMD	AMD	\$83.64	April 7, 2025*
ORCK	ORCK	Data not available	-

*Note: Most recent available data for AMD is from April 7, 2025

Notes

- All prices are closing prices for the respective dates
- ORCK ticker symbol could not be found or verified in financial databases
- Data sourced from MacroTrends financial data

Generated on: April 13, 2025

The Counterparty of MCP: OpenAI's Response API

TOOL	DESCRIPTION
Function calling	Interact with developer-defined code.
Web search	Fetch up-to-date information from the web.
File search	Perform semantic search across your documents.
Computer use	Understand and control a computer or browser.

Web search tool example

javascript ↕

```
1 import OpenAI from "openai";
2 const client = new OpenAI();
3
4 const response = await client.responses.create({
5   model: "gpt-4o",
6   tools: [ { type: "web_search_preview" } ],
7   input: "What was a positive news story from today?",
8 });
9
10 console.log(response.output_text);
```

```
def chat_with_gpt(messages):
```

```
    try:
        response = client.chat.completions.create(
            model="gpt-4-turbo",
            messages=messages,
            tools=[{"type": "web_search_preview"}]
        )

        return response.choices[0].message.content

    except Exception as e:
        return f"error: {str(e)}"
```

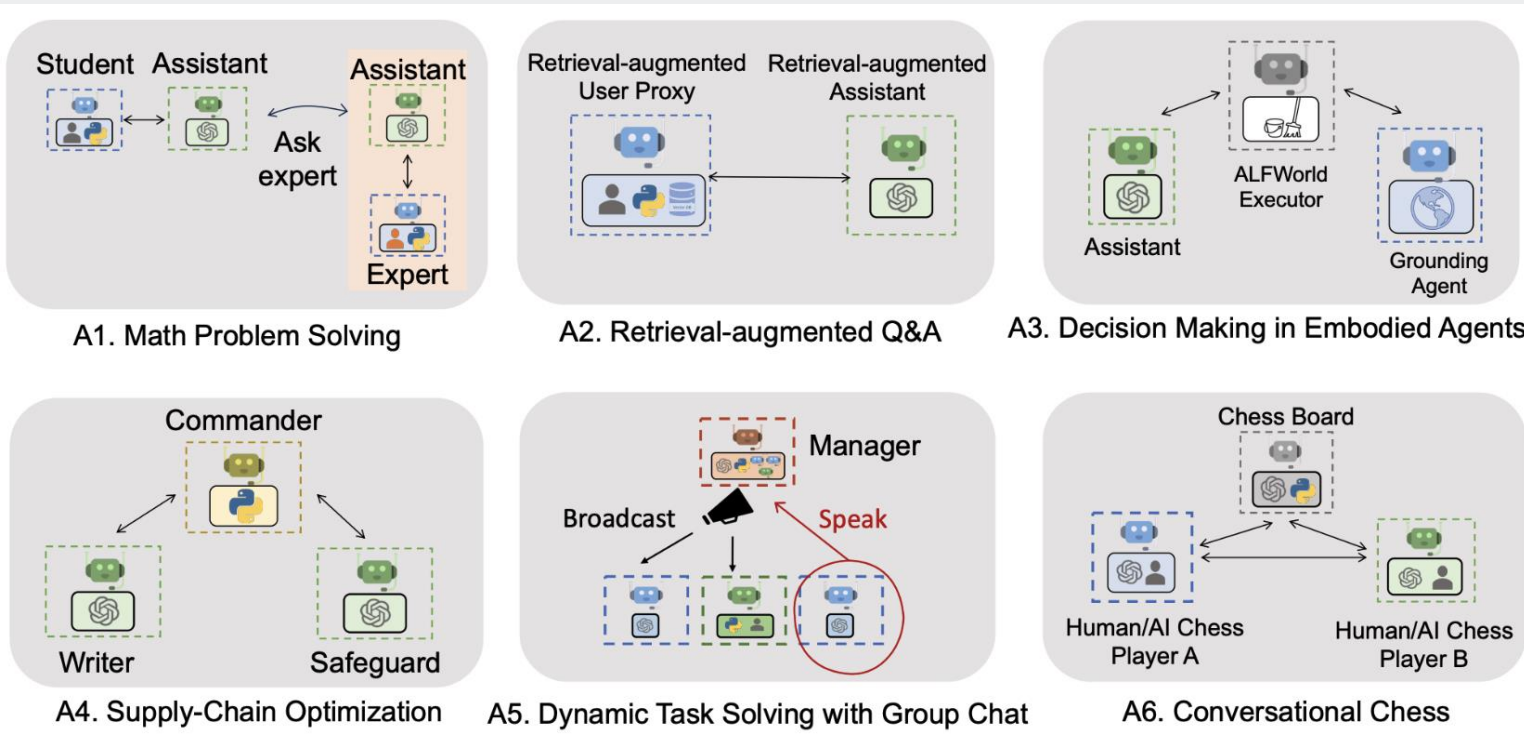
```
def chat_with_openai_tools(messages):
```

```
    try:
        response = client.chat.completions.create(
            model="gpt-4-turbo",
            messages=messages,
            tools=[
                {"type": "web_search"},
                {"type": "retrieval"},
                {"type": "code_interpreter"}
            ]
        )

        return response.choices[0].message.content

    except Exception as e:
        return f"error: {str(e)}"
```

From one Agent to Multiply Agents: AutoGen Framework



```
# Create an OpenAI model client.
model_client = OpenAIChatCompletionClient(
    model="gpt-4o-2024-08-06",
    # api_key="sk-...", # Optional if you have an OPENAI_API_KEY env variable set.
)

# Create the primary agent.
primary_agent = AssistantAgent(
    "primary",
    model_client=model_client,
    system_message="You are a helpful AI assistant.",
)

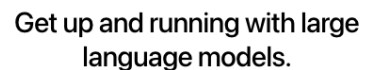
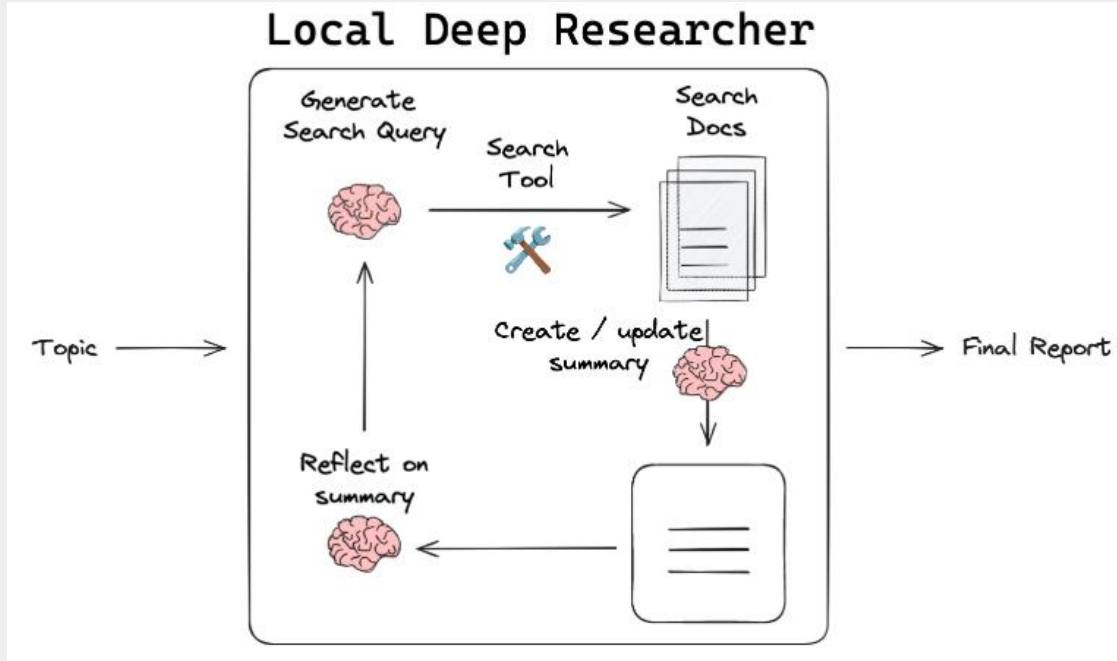
# Create the critic agent.
critic_agent = AssistantAgent(
    "critic",
    model_client=model_client,
    system_message="Provide constructive feedback. Respond with 'APPROVE' to when y

# Define a termination condition that stops the task if the critic approves.
text_termination = TextMentionTermination("APPROVE")

# Create a team with the primary and critic agents.
team = RoundRobinGroupChat([primary_agent, critic_agent], termination_condition=ter
```

For more examples: <https://autogen-ai.github.io/autogen/docs/notebooks>
<https://microsoft.github.io/autogen/stable//index.html>
<https://www.microsoft.com/en-us/research/project/autogen/>

Two Agents Deep Research: Ollama Deep Research



Run Llama 3.3, DeepSeek-R1, Phi-4, Mistral, Gemma 3, and other models, locally.

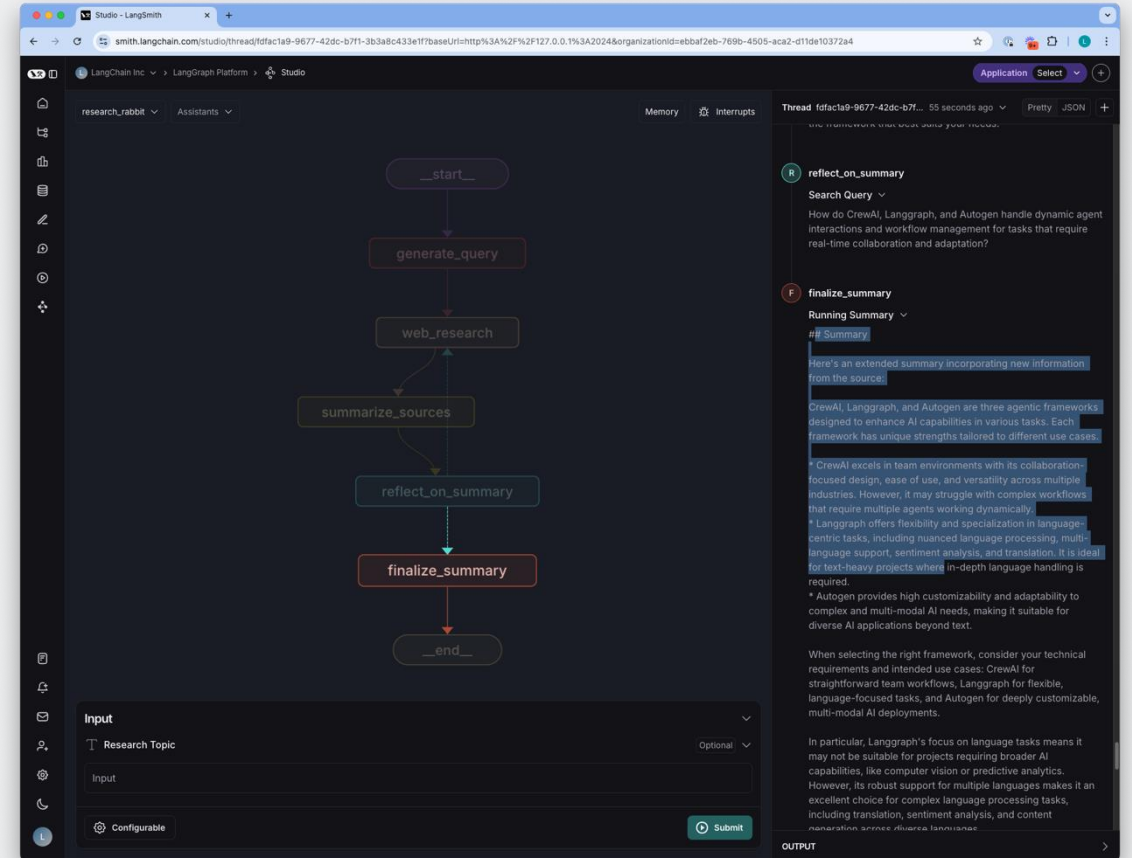
Download ↓

Available for macOS,
Linux, and Windows



```
(base) Mac-Studio:~ apple$ ollama pull gemma3:27b
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling e796792eba26... 10% | 1.8 GB/ 17 GB 52 MB/s 4m56s
(base) Mac-Studio:~ apple$ ollama run gemma3:27b
>>> hello
Hithere! 🙌 Whatcan I helayou with today? 😊 Anything
at all—I canchat,answerquestions,writestories, help wi
th
code,brainstorm ideas...justletme know!

>>> [end a message (/? for help)]
```



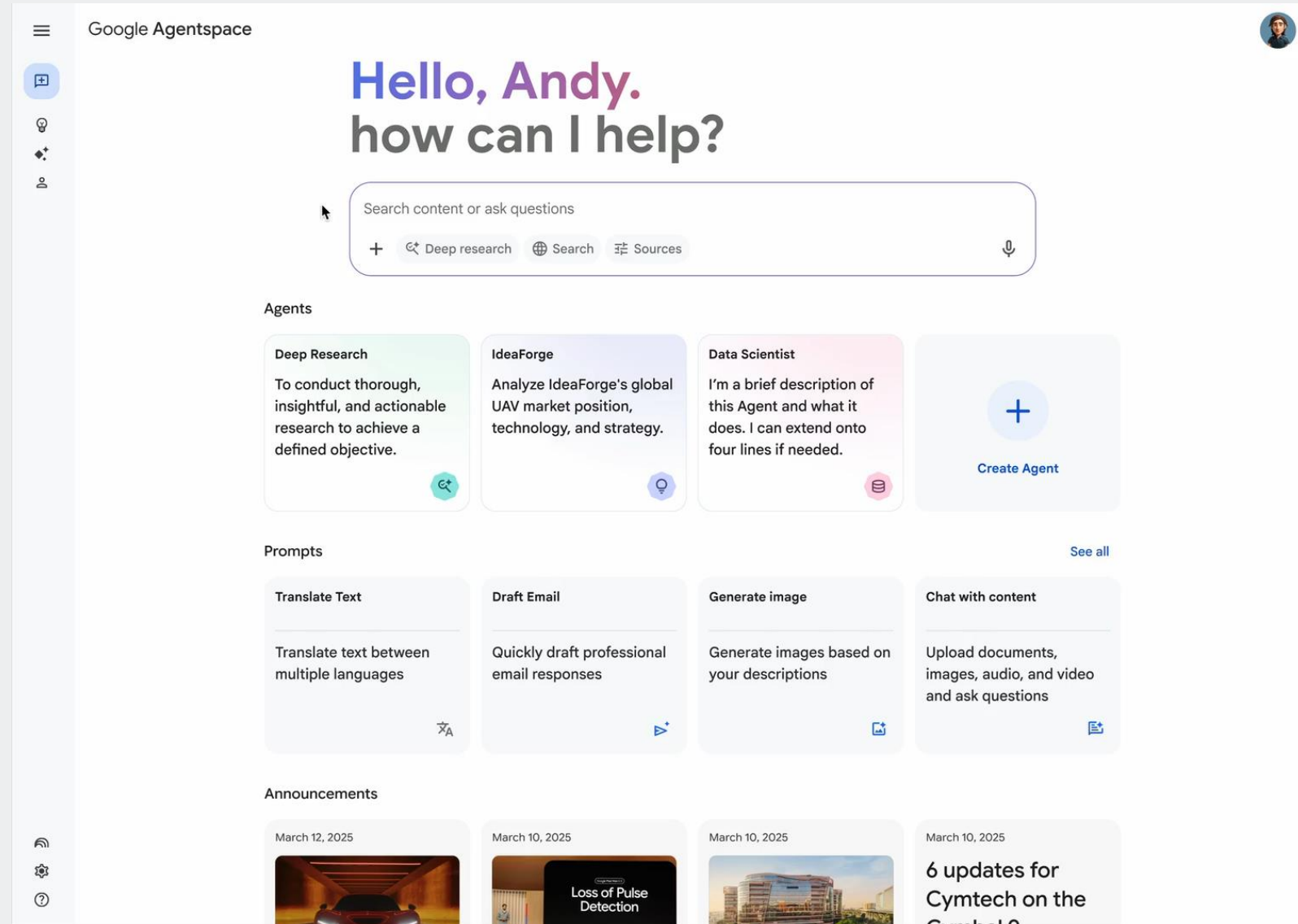
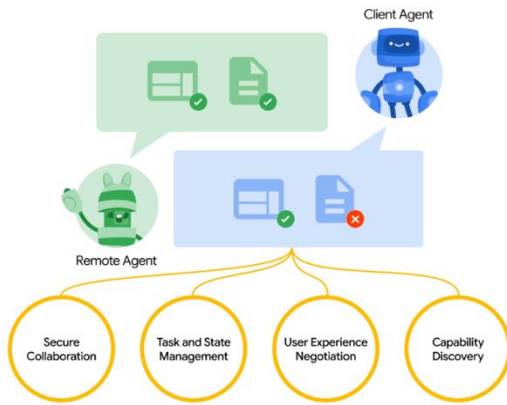
Using: LangChain and LangGraph
Those frameworks are a little bit obsolete, so no need to learn

From one Agent to Multiple Agents

The Announcing of Agent2Agent Protocol (A2A) On Apr 09, 25



How A2A works



Lab 3: Use n8n to replicate OpenAI's Deep Research feature.

What is n8n? <https://n8n.io/>

n8n DeepResearcher

This template attempts to replicate OpenAI's DeepResearch feature which, at time of writing, is only available to their pro subscribers.

Though the inner workings of DeepResearch have not been made public, it is presumed the feature relies on the ability to deep search the web, scrape web content and invoking reasoning models to generate reports. All of which n8n is really good at!

How it works

- A form is used to first capture the user's research query and how deep they'd like the researcher to go.
- Once submitted, a blank Notion page is created which will later hold the final report and the researcher gets to work.
- The user's query goes through a recursive series of web searches and web scraping to collect data on the research topic to generate partial learnings.
- Once complete, all learnings are combined and given to a reasoning LLM to generate the final report.
- The report is then written to the placeholder Notion page created earlier.

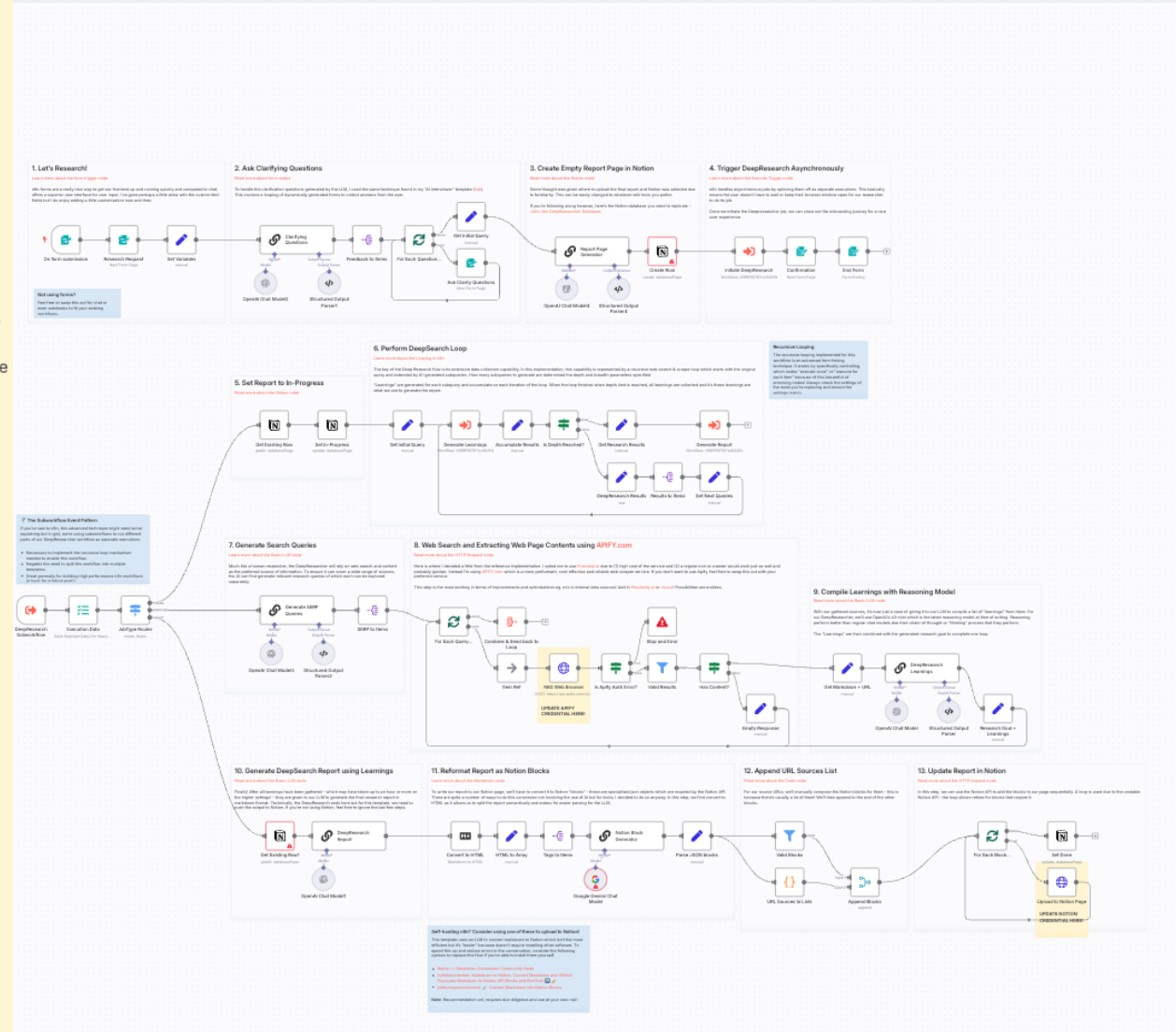
How to use

- Duplicate this Notion database to use with this template:
<https://jimleuk.notion.site/19486dd60c0c80da9cb7eb1468ea9afd?v=19486dd60c0c805c8e0c000ce8c87acf>
- Sign-up for [APIFY.com](https://apiify.com) API Key for web search and scraping services.
- Ensure you have access to OpenAI's o3-mini model. Alternatively, switch this out for o1 series.
- You must publish this workflow and ensure the form url is publically accessible.

On Depth & Breadth Configuration

For more detailed reports, increase depth and breadth but be warned the workflow will take a exponentially more time and money to complete. The defaults are usually good enough.

Depth=1 & Breadth=2 - will take about 5 - 10mins.
Depth=1 & Breadth=3 - will take about 15 - 20mins.
Depth=3 & Breadth=5 - will take about 2+ hours!



Thank You!

Q & A

Contact: Shaohui Wang,
shaohui@student.gsu.edu