TECHLENT

# 高阶大语言模型课程

Huajun Zeng

12/13/2024 - 1/31/2025
（12月27日和1月3日放假，共计6次课）
每周五 5pm-7pm PT / 8pm-10pm ET

# 课程安排

| Week | Date | Content | Week | Date | Content |
|---|---|---|---|---|---|
| 1 | 2024-12-13 | Retrieval Augmented Generation (RAG) for LLM<br>● Why augmenting LLMs?<br>● Methods for LLM augmentation<br>● Augmenting LLMs with retrieval<br>● Augmenting LLMs with fine tuning | 4 | 2025-01-17 | Pipeline for LLM Applications: From Code to Products<br>● Full stack LLM: tools needed for an LLM application<br>● Case study: build an LLM app from ground |
| 2 | 2024-12-20 | Chatbot Building with LLM APIs<br>● Environment setup<br>● Introduction to LLM APIs<br>● Using chat completion APIs<br>● Using fine tuning APIs | 5 | 2025-01-24 | More LLM Applications and Course Project<br>● Showcase of potential LLM applications for productivity, creativity and more<br>● More advanced LLM applications: AI Agent, Multi-modality, etc.<br>● Introduction to the course project: requirement and discussion |
| 3 | 2025-01-10 | Chatbot Building with LLM Frameworks and Vector Database<br>● Introduction to Langchain and LlamaIndex<br>● Case study: a chatbot from Langchain and vector database | 6 | 2025-01-31 | Project Presentation<br>● Student presentation on the course project |

# 家庭作业回顾

- Homework
  - Add Flask to the Chatbot you developed
  - Successfully run it on your local desktop
  - (Optional) Deploy it to AWS and serve it online

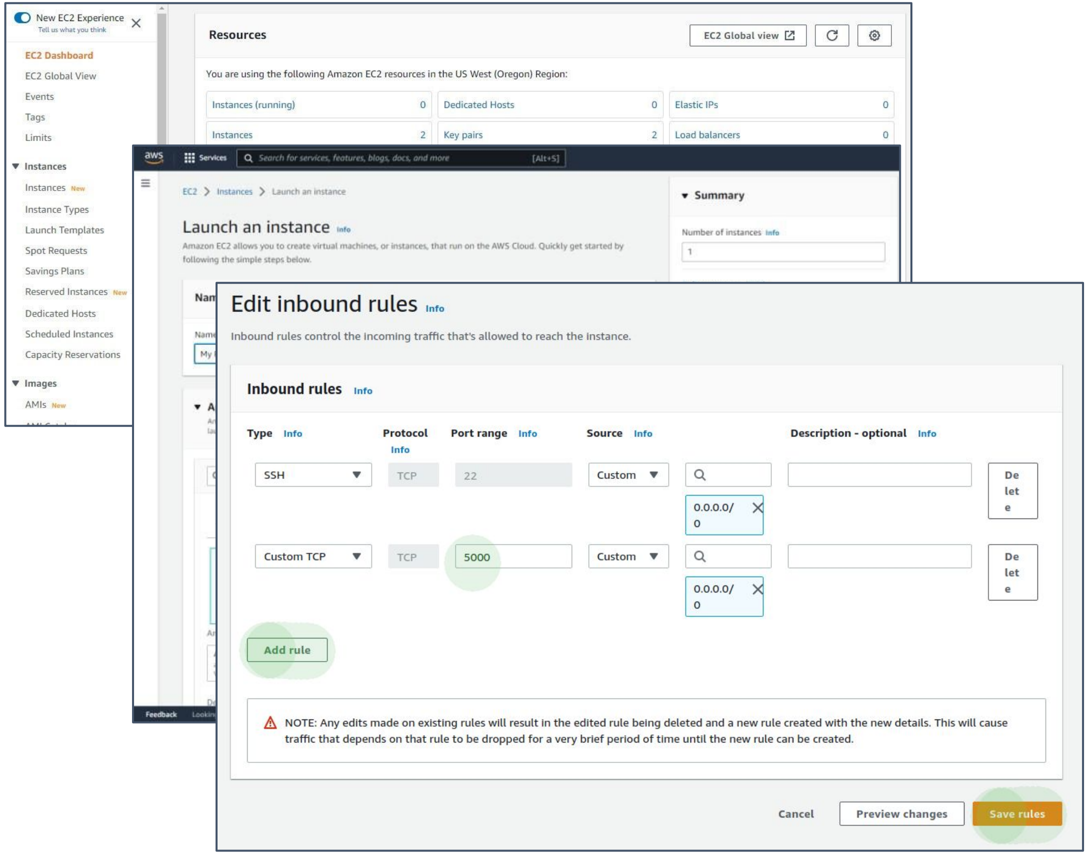
# Homework Presentation from Bingcheng Han

# Homework Comments

- Great discussion on WeChat
- Good to see many real world questions
- app.db doesn’t need to be commit to github

# 部署细节

- AWS EC2
  - Choose an image
  - Setup python
  - Elastic IP
  - Add inbound rule for your port



Resources

You are using the following Amazon EC2 resources in the US West (Oregon) Region:

| | | | | | |
|---|---|---|---|---|---|
| Instances (running) | 0 | Dedicated Hosts | 0 | Elastic IPs | 0 |
| Instances | 2 | Key pairs | 2 | Load balancers | 0 |

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Summary

Number of instances: 1

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

| Type | Protocol | Port range | Source | | Description - optional |
|---|---|---|---|---|---|
| SSH | TCP | 22 | Custom | 0.0.0.0/0 | |
| Custom TCP | TCP | 5000 | Custom | 0.0.0.0/0 | |

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

# 产品级App的考虑

- Domain name
  - Choose a Domain Registrar
  - Buy a domain name
  - Configure DNS
- Security
  - Https port
  - Authentication
- Scalability
  - Flask app scale up / load balancer
  - Database sharding
  - Docker based deployment

# 第五课: 更多LLM应用

- More LLM functionalities
  - Multi-modality
  - Function call
- Showcase of potential LLM applications
- Introduction to the course project: requirement and discussion

# OpenAI Multi-Modal APIs

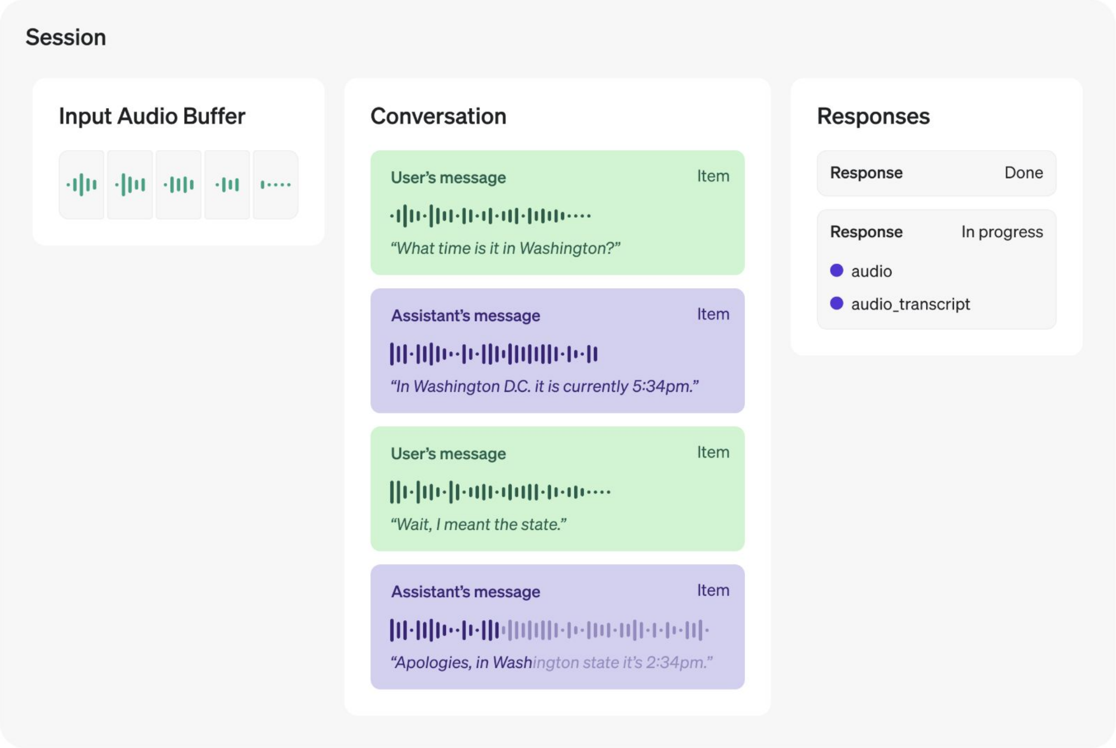
- Vision: Using gpt-4o or gpt-4o-mini to understand images
- DALL-E: generate images from natural language prompts
- Sora: AI model which generates videos from text descriptions
- Whisper: transcribe and translate speech to text
- Text to speech: convert text data to speech
- Realtime API: low-latency, "speech in, speech out" conversational interactions

TECHLENT

# OpenAI Realtime API

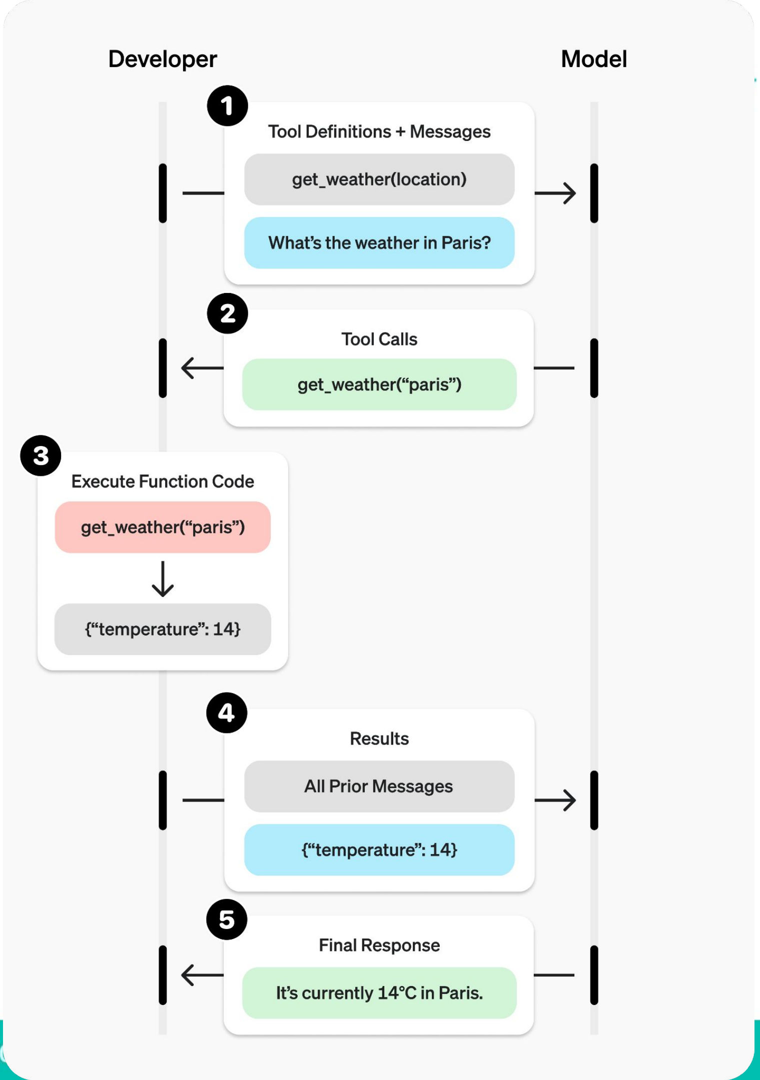
- Text input/output
- Audio input/output
- Voice Activity Detection (VAD)
- Function Calling

[https://openai.com/index/introducing-the-realtime-api/](https://openai.com/index/introducing-the-realtime-api/)



# Function Call (Tool Use)

- Extend LLM's capabilities
- Integrate with your existing systems
- Automate complex tasks



# Function Call Code

Refer to

- backend_api/test_function_call.py
- backend_langchain/test_12_tool_call.py

# Best Practices for Tool Definitions

- Provide very detailed descriptions for the tool, including:
  - What the tool does
  - When it should be used (and when it shouldn’t)
  - What each parameter means and how it affects the tool’s behavior
- Add examples:
  - When you don’t know how to describe
  - Add 1 or multiple examples on which function to call in which case

# LLM Applications - OpenAI Assistant

- Customizable AI Assistants: Developers can create assistants tailored to specific tasks or domains by providing instructions and selecting appropriate models.
- Tool Integration: Assistants can utilize multiple tools simultaneously, including:
  - Code Interpreter: Allows writing and executing Python code for data analysis and manipulation.
  - File Search (Retrieval): Enables processing and searching through uploaded documents, creating a knowledge base for the assistant.
  - Function Calling: Permits integration with external APIs and custom functions, extending the assistant's capabilities.
- Persistent Threads: Assistants maintain conversation history through threads, simplifying state management and allowing for continuous interactions.
- https://platform.openai.com/playground/assistants

# LLM Applications - OpenAI Canvas

- Writing Assistance:
  - Suggest Edits: Offers real-time suggestions and feedback on writing
  - Adjust Length: Allows users to expand or condense content easily
  - Change Reading Level: Adapts text complexity to suit different audiences
  - Add Final Polish: Provides a final check for grammar, clarity, and consistency
- Coding Support:
  - Review Code: Provides inline suggestions for code improvement
  - Add Logs: Inserts print statements for easier debugging
  - Add Comments: Generates explanatory comments for code sections
  - Fix Bugs: Detects and rewrites problematic code
  - Port to Another Language: Translates code between different programming languages
- https://openai.com/index/introducing-canvas/

# LLM Application - OpenAI Operator

- Advanced AI agent designed to autonomously perform web-based tasks on behalf of users
  - Booking travel arrangements
  - Purchasing groceries
  - Filing expense reports
  - Making restaurant reservations
- This allows it to interpret screenshots and perform actions such as typing, clicking, and scrolling, effectively navigating web interfaces like a human user.
- 
- https://openai.com/index/introducing-operator/

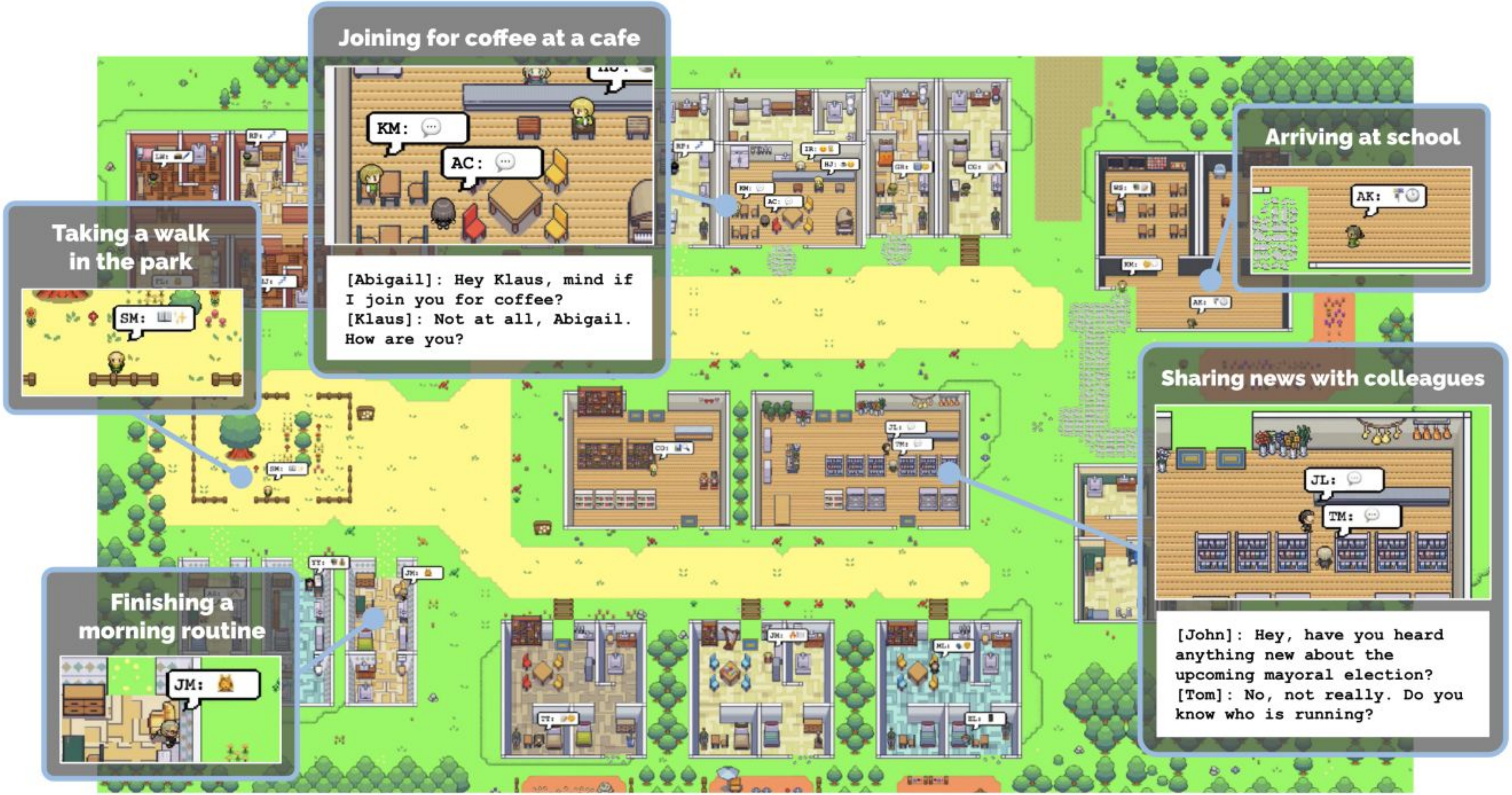
# NotebookLM

- AI-powered tool developed by Google that revolutionizes note-taking and information management
- https://notebooklm.google/

# Agent（智能体）

- An Agent is an AI system capable of acting more or less **independently** to complete tasks with minimal human guidance.
- It can initiate actions on its own, chain together commands, and dynamically adapt to new information in pursuit of a goal.
- Humans may provide overall goals or constraints but do not typically intervene in step-by-step tasks once the Agent is running.

# Generative Agents in The Sims



Joining for coffee at a cafe

[Abigail]: Hey Klaus, mind if I join you for coffee?
[Klaus]: Not at all, Abigail. How are you?

Arriving at school

Taking a walk in the park

Sharing news with colleagues

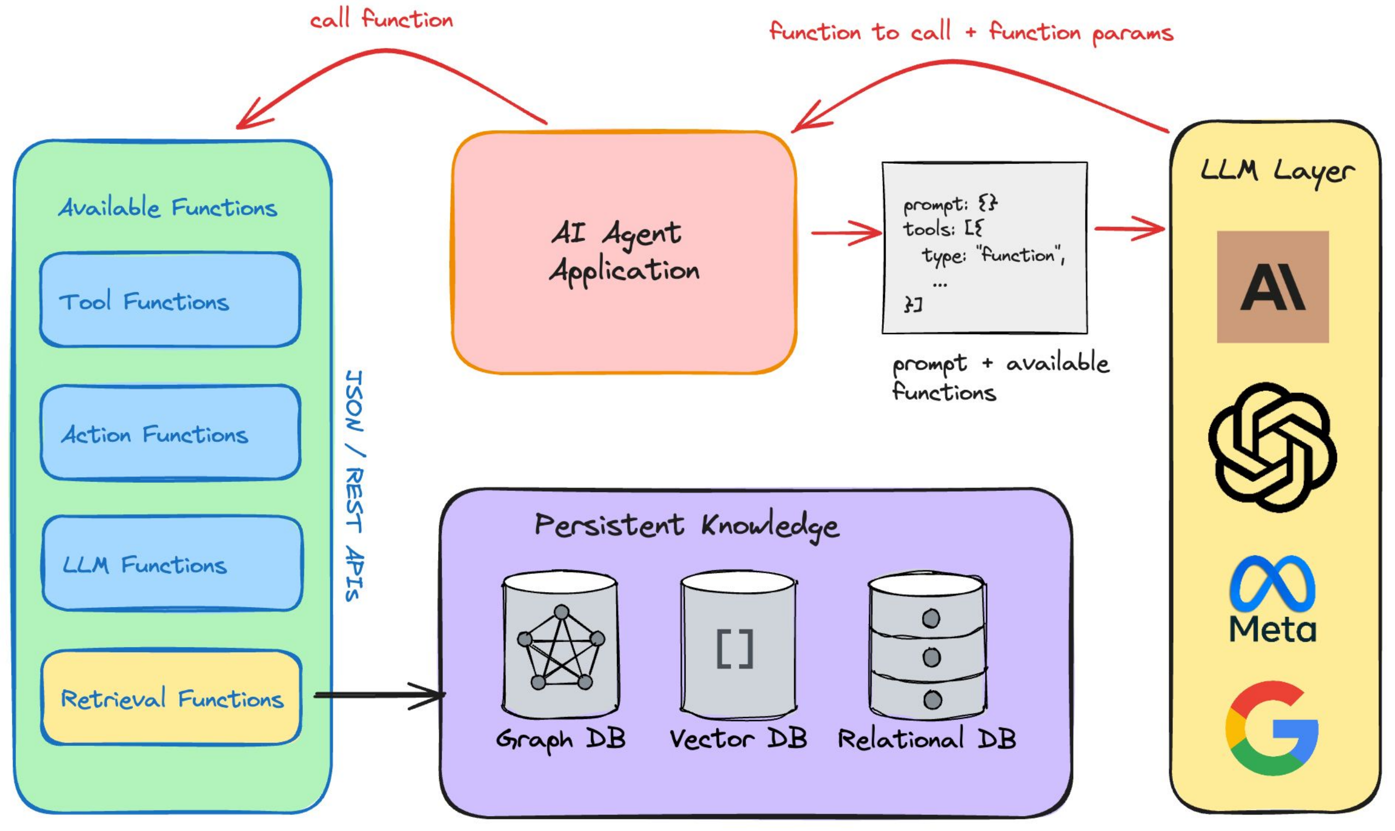
[John]: Hey, have you heard anything new about the upcoming mayoral election?
[Tom]: No, not really. Do you know who is running?

Finishing a morning routine

# 什么时候应该用Agent

- Opt for simple solutions when building applications with LLMs, and only increase complexity when necessary; sometimes, an agent system may not be needed at all.
- Agent systems involve trade-offs between latency and cost for better task performance, so evaluate whether these trade-offs are justified before use.
- Workflows are ideal for providing stability and consistency in complex tasks, while agents are preferable for scenarios requiring flexibility and large-scale, model-driven decision-making.
- In most cases, optimizing a single LLM invocation with retrieval and contextual examples is sufficient to meet application needs.

# Agentic RAG



```
prompt: {}
tools: [{
  type: "function",
    ...
}]
```

# Agent Framework

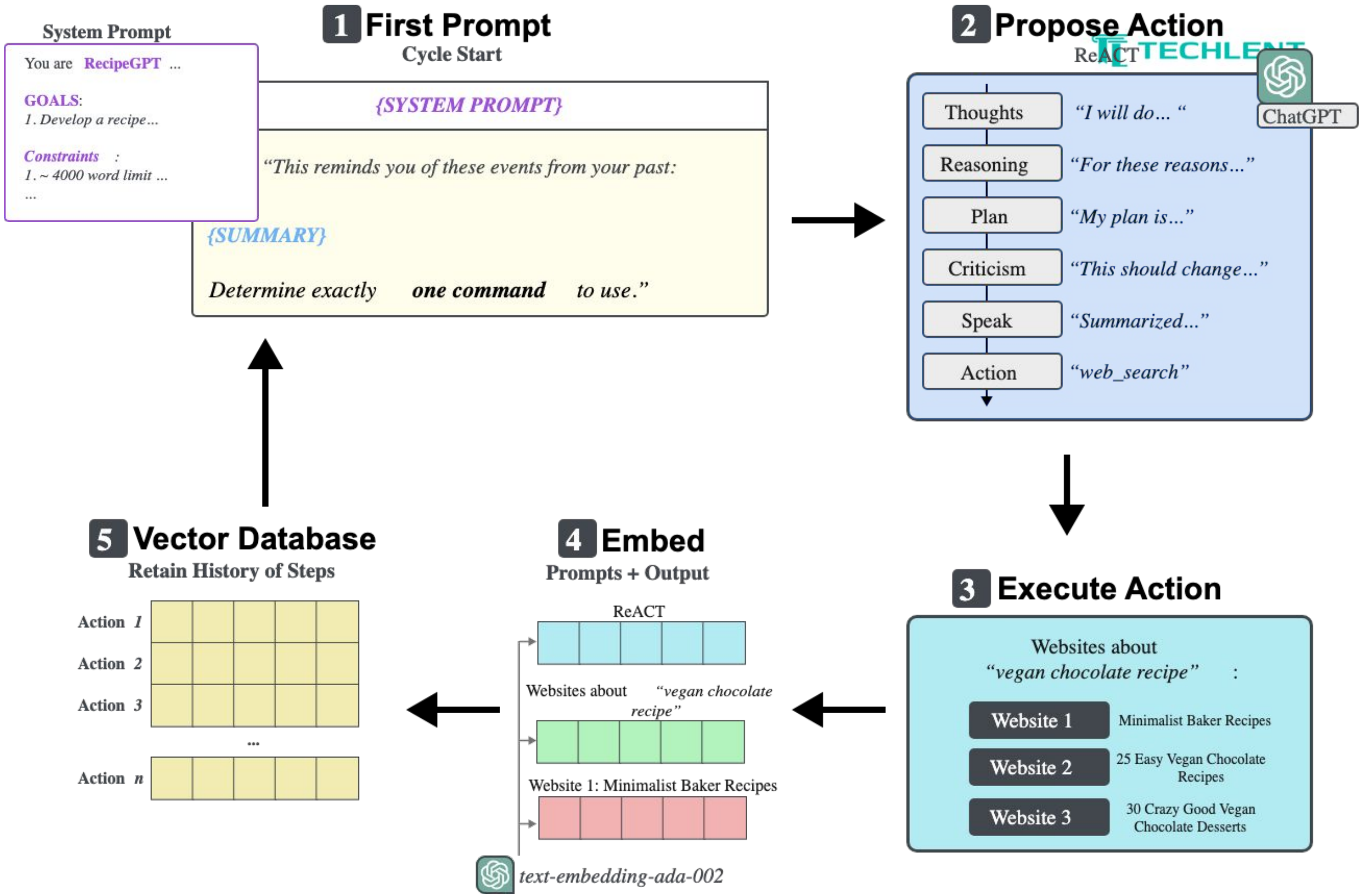
- OpenAI Swarm
- LangGraph
- Microsoft Autogen
- AutoGPT
- CrewAI
- Dify

If you choose to use a framework, make sure to understand the underlying implementation logic, as incorrect assumptions about the underlying mechanisms are often one of the main issues in development.

# AutoGPT

☆ Star 171k

Powerful platform that allows you to create, deploy, and manage continuous AI agents that automate complex workflows.



System Prompt

You are RecipeGPT ...

GOALS:
*1. Develop a recipe...*

*Constraints* :
*1. ~ 4000 word limit ...*
...

**1 First Prompt**
Cycle Start

*{SYSTEM PROMPT}*

*"This reminds you of these events from your past:*

*{SUMMARY}*

*Determine exactly* ***one command*** *to use."*

**2 Propose Action**
ReACT

ChatGPT

| | |
|---|---|
| Thoughts | *"I will do... "* |
| Reasoning | *"For these reasons..."* |
| Plan | *"My plan is..."* |
| Criticism | *"This should change..."* |
| Speak | *"Summarized..."* |
| Action | *"web_search"* |

**3 Execute Action**

Websites about *"vegan chocolate recipe"* :

| | |
|---|---|
| Website 1 | Minimalist Baker Recipes |
| Website 2 | 25 Easy Vegan Chocolate Recipes |
| Website 3 | 30 Crazy Good Vegan Chocolate Desserts |

**4 Embed**
Prompts + Output

ReACT

Websites about *"vegan chocolate recipe"*

Website 1: Minimalist Baker Recipes

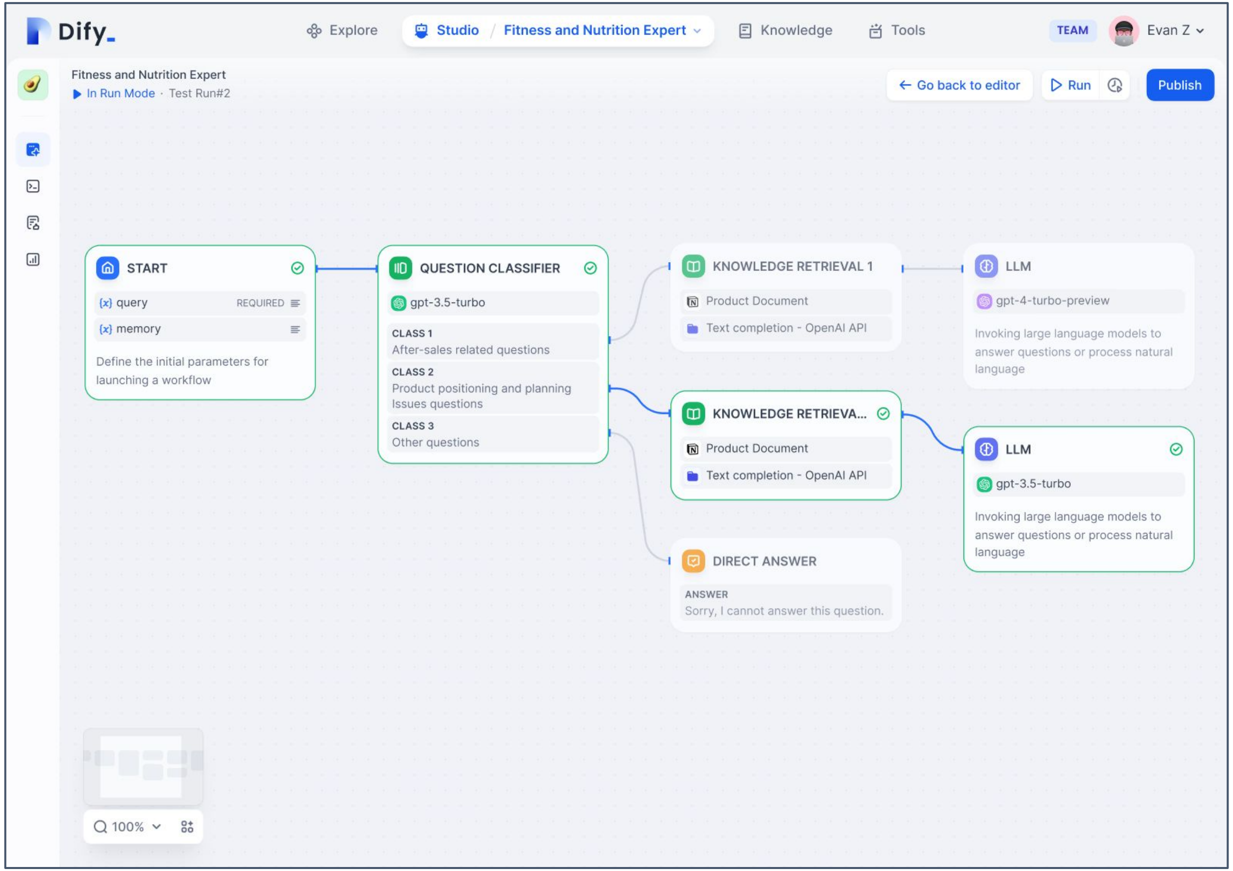
*text-embedding-ada-002*

**5 Vector Database**
Retain History of Steps

Action *1*
Action *2*
Action *3*
...
Action *n*

# Dify

- Visual Prompt Orchestration
- RAG Engine
- Prompt IDE
- Agents and Plugins


Dify
Explore
Studio / Fitness and Nutrition Expert
Knowledge
Tools
TEAM
Evan Z
Fitness and Nutrition Expert
In Run Mode · Test Run#2
Go back to editor
Run
Publish
START
{x} query REQUIRED
{x} memory
Define the initial parameters for launching a workflow
QUESTION CLASSIFIER
gpt-3.5-turbo
CLASS 1
After-sales related questions
CLASS 2
Product positioning and planning Issues questions
CLASS 3
Other questions
KNOWLEDGE RETRIEVAL 1
Product Document
Text completion - OpenAI API
KNOWLEDGE RETRIEVA...
Product Document
Text completion - OpenAI API
DIRECT ANSWER
ANSWER
Sorry, I cannot answer this question.
LLM
gpt-4-turbo-preview
Invoking large language models to answer questions or process natural language
LLM
gpt-3.5-turbo
Invoking large language models to answer questions or process natural language
100%

# Student Project

- Presentation and code
  - Presentation: describe the use case
  - Code: from homework
- Content: LLM or AI related
  - Solving real world problems
  - Forward looking, conceptual prototyping
- Point to make
  - Why this is important problem
  - What make your solution unique

# Some Project Examples from Previous Sessions

- Reference Management and Chatbot Application
- Nvidia 10-K 2020-2024 RAG Chatbot
- AutoInk
- Advancing Chemical Research through NLP and LLM Integration
- Solving Math Problems with Langchain Agents
- Mental Health Assistant
- Code Base Interpretation with LLM
- …

# Questions?