Module 6:
Prompt Engineering vs Fine-tuning & Agent

Hamza Farooq
Nikka Mofid

Announcement

Announcements for the last class:

- November 11th off (work on your presentations)
- Nov 18th Final Project presentation in class
- This will be the last class where we will teach


THIS IS THE END
imgflip.com

Learning Outcomes Part -1

We will be covering topics on:

- What is Prompt Engineering
- Techniques of Prompt Engineering
- Fine-tuning LLMs
- PEFT
- Validation metrics
- Code Walkthrough in fine-tuning models*
    - ChatGPT

Learning Outcomes for Agents

- **LLM Agents - A Quick Refresher**
  - How AI Agents Work?
  - AI Agents vs Single Prompted LLM
  - Hands-on Colab Notebook on Multi-Agents for Content Creation

- **Autogen — AI Agents Framework**
  - What is AutoGen?
  - Main Features of AutoGen
  - Types of Agents in AutoGen
  - Diverse Applications
  - Hands-on Notebook on Getting Started with AutoGen

00

Self Hosted vs Closed LLMs Discussion

	Open-source LLMs	Closed-source LLMs
Availability	Freely available	Restricted: Paid customers, licence holders
Cost	Typically lower-cost or free	Higher cost, often with subscription fees
Integration	Can be integrated with a variety of applications	Limited to company-provided integrations
Implementation	Likely slower, especially if training is required	Ready-made APIs could make implementation as easy as plug and play
Customisation	Can be modified and adapted to specific needs	Limited customisation options
IP rights	No IP rights, free to use and modify	Company retains IP rights
Collaboration & innovation	Encourages collaboration and shared innovation	Limited to company's resources and vision

Collaboration & innovation	Encourages collaboration and shared innovation	Limited to company's resources and vision
Model training, data access	Open data access, customisable training	Limited data access, pre-trained models
Transparency & explainability	Transparent, explainable model architecture	Often proprietary, less transparent
Security	Vulnerable to exploitation in open community	Company-driven security measures
Quality control	Varies by project and community	Company-driven quality control
Community support	Large developer community, support for popular models	Limited support, usually provided by the company
Updates & maintenance	Community-driven updates and maintenance	Company-driven updates and maintenance

Information compiled in 2023.

L'ATELIER BNP PARIBAS

01

Prompt Engineering ?

Prompt Engineering

Prompt engineering involves designing and refining language model prompts to achieve specific desired outputs. It includes crafting prompts that provide clear instructions, context, or constraints to guide the model's responses.

Prompt Engineering

In other words, it involves taking a *prompt* and manipulating it to get a type of desired output.

Adding context to the prompt, output instructions or response constraints are all examples of prompt engineering.

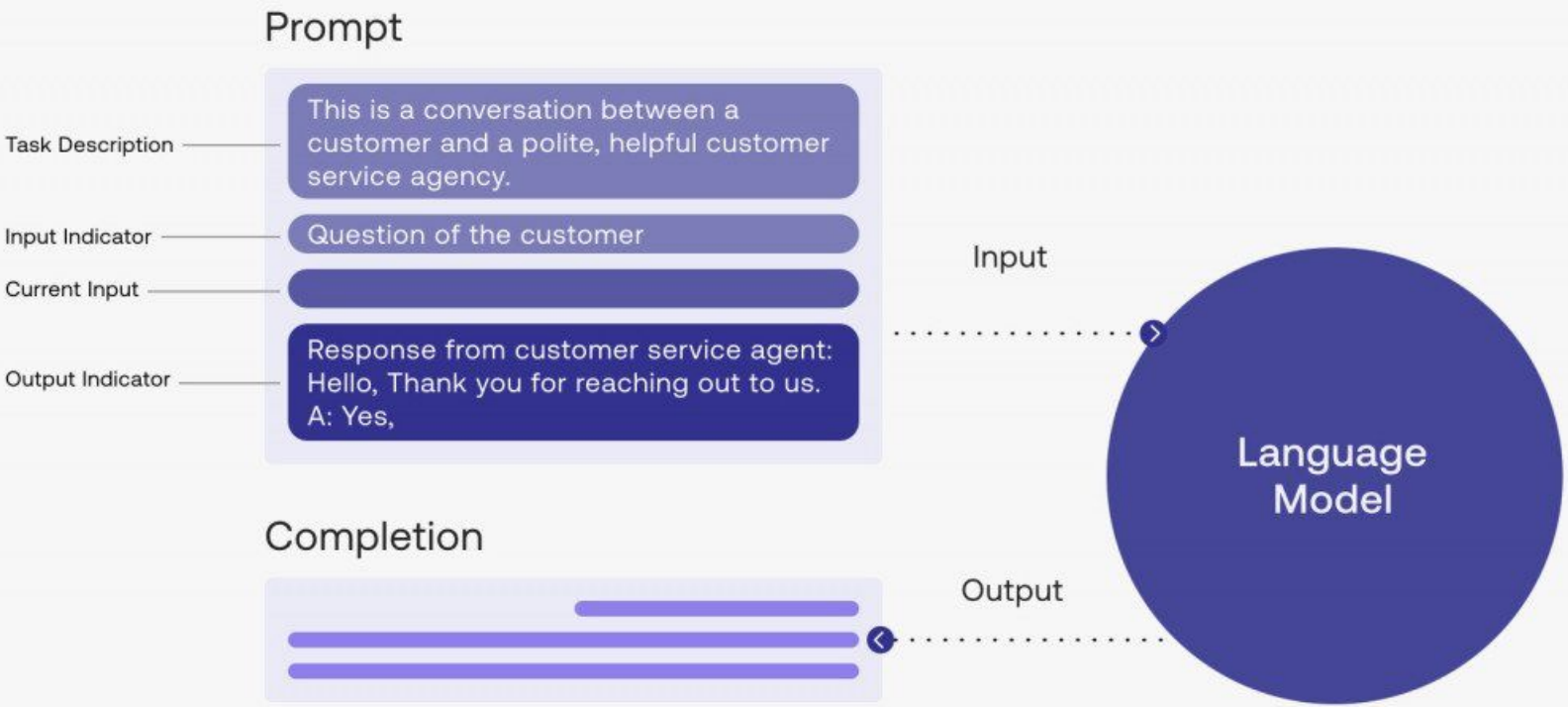
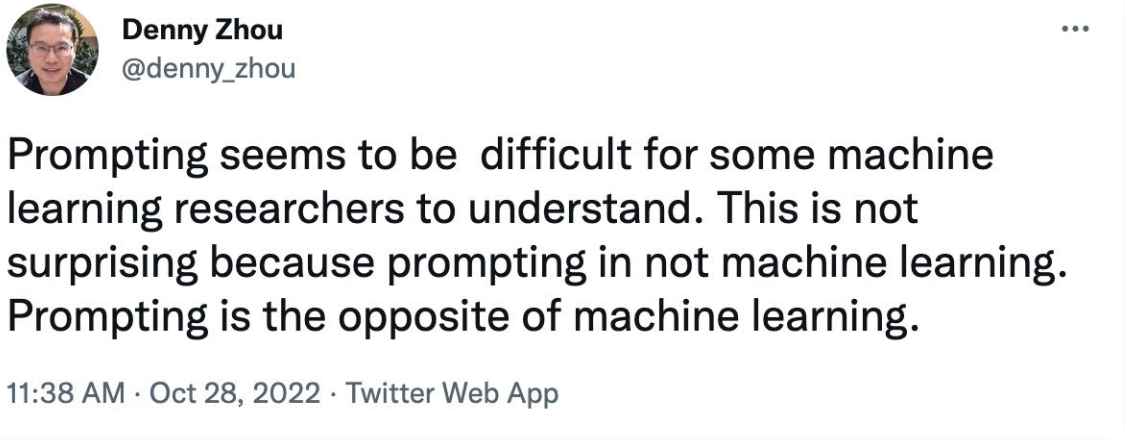


Image: cohere.ai

Prompt engineering is of an more art, than science.



Importance of Prompt Engineering ?

- **Reduces Ambiguity** → Clarifies user intent, leading to more precise and relevant outputs from AI models.

Importance of Prompt Engineering ?

- **Improves User Experience** → Provides users with more accurate and helpful responses, improving satisfaction and engagement.

  Helps users get better answers, boosting their satisfaction and engagement.

Why Prompt Engineering works ?

- LLMs have been trained with extensive data and human annotated prompts.

- Hence when we use prompts to define role, details and intent, it yields better results

- Minimizes misunderstandings and errors by clearly defining the input parameters and expected output.

The Science of Prompt Engineering

Prompt engineering is the art of crafting precise prompts to guide the AI into delivering a desired output.

Importance in Domain Adaptation:

- **Targeted Responses** → Ensures responses are not only accurate but also contextually relevant.
- **Leverage Domain Knowledge** → Incorporates specific terminology and workflows unique to the domain.

The Science of Prompt Engineering

Prompt engineering is the art of crafting precise prompts to guide the AI into delivering a desired output.

Different prompting Strategies:

- **Instruction-based** → Direct and explicit instructions for tasks.
- **Contextual Clues** → Incorporating specific context to guide the AI's focus.

What are some more Advanced Prompt Engineering Techniques?

What are some more Advanced Prompt Engineering Techniques?

- **Zero-shot Prompting** → Utilizing generic prompts without prior examples.

What are some more Advanced Prompt Engineering Techniques?

- **Few-shot Prompting** → Providing a few examples to set a response pattern.

What are some more Advanced Prompt Engineering Techniques?

- **Fine-tuning with Examples** → Intensive example-based training to refine model understanding.

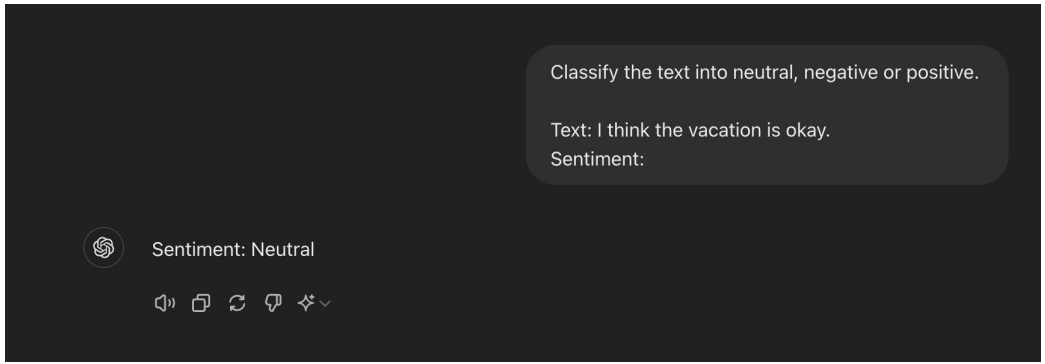
What are some more Advanced Prompt Engineering Techniques?

- **Zero-shot Prompting** → Utilizing generic prompts without prior examples.
- **Few-shot Prompting** → Providing a few examples to set a response pattern.
- **Fine-tuning with Examples** → Intensive example-based training to refine model understanding.

Impact of Prompt Engineering → Enhanced model performance, reliability, and domain-specific accuracy.

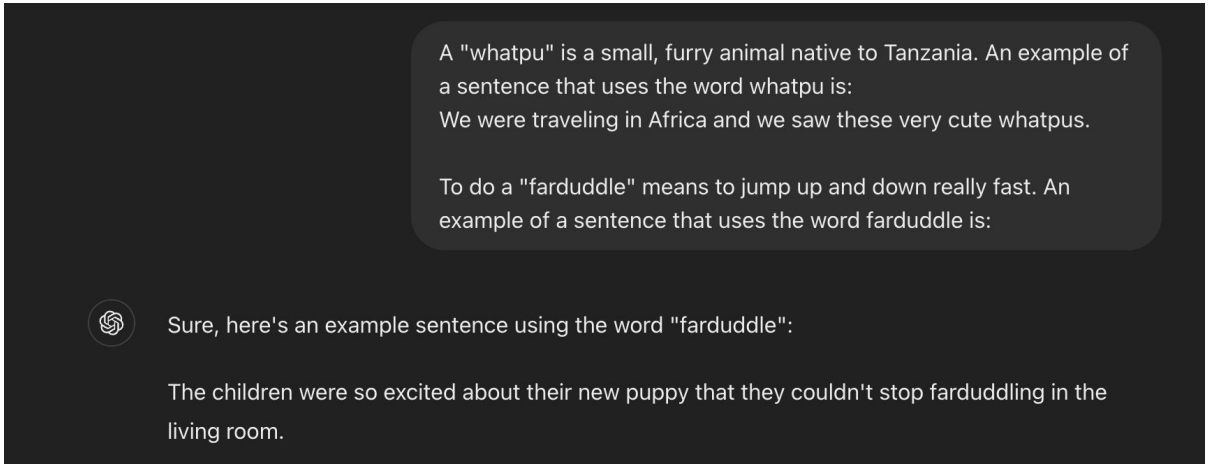
Examples : Zero Shot Prompting

- Large-scale training makes these models capable of performing some tasks in a **"zero-shot"** manner.
- Zero-shot prompting means that the prompt used to interact with the model **won't contain examples** or demonstrations.
- The zero-shot prompt directly instructs the model to perform a task **without any additional examples to steer it**.



Examples: Few Shot Prompting

- Few-shot prompting can be used as a technique to **enable in-context learning** where we provide demonstrations in the prompt to steer the model to better performance.
- The demonstrations serve as **conditioning for subsequent examples** where we would like the model to generate a response.



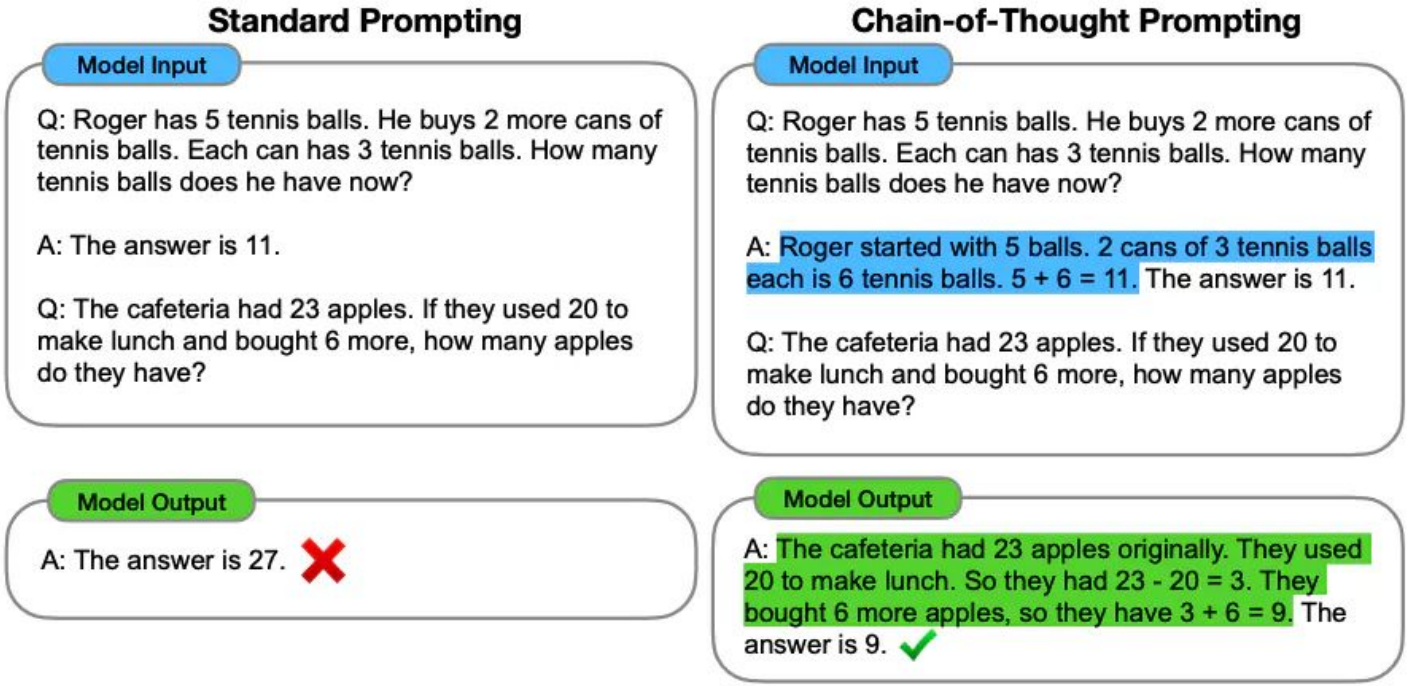
Chain of Thoughts

- chain-of-thought (CoT) prompting enables **complex reasoning capabilities through intermediate reasoning steps**.

- You can **combine it with few-shot prompting** to get better results on more complex tasks that require reasoning before responding.

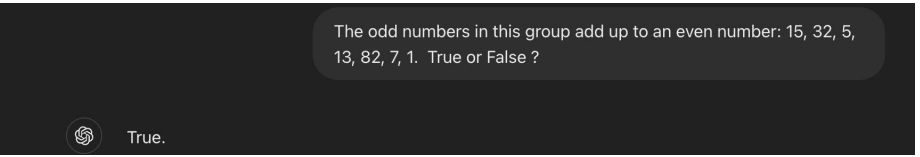
Source: https://www.promptingguide.ai/techniques/cot

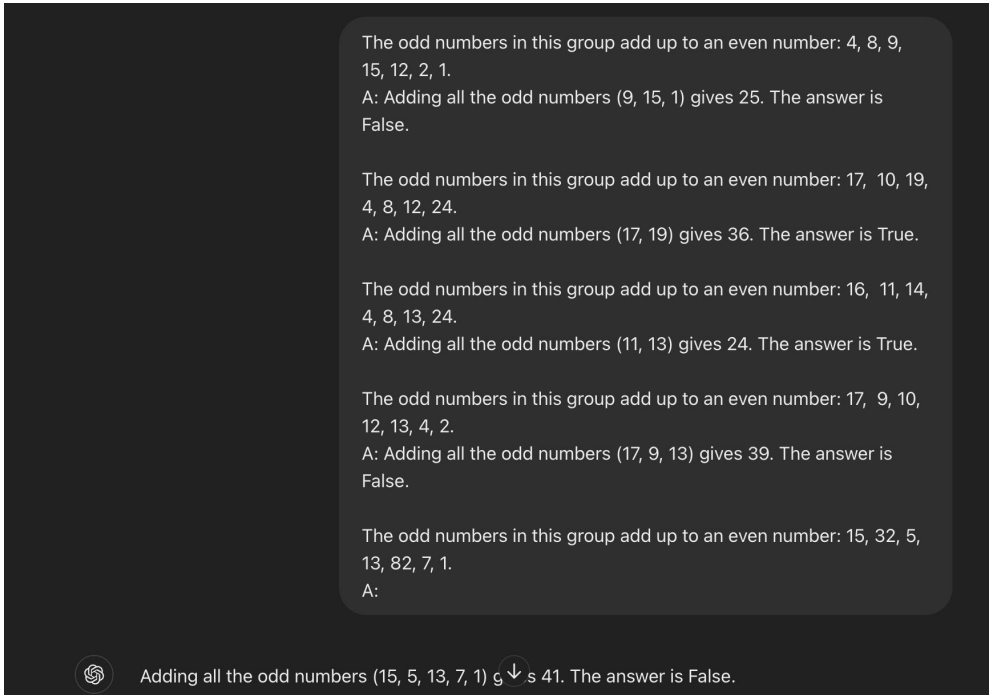
Chain of Thoughts



Source: https://www.promptingguide.ai/techniques/cot

Chain of Thoughts





Source: https://www.promptingguide.ai/techniques/cot

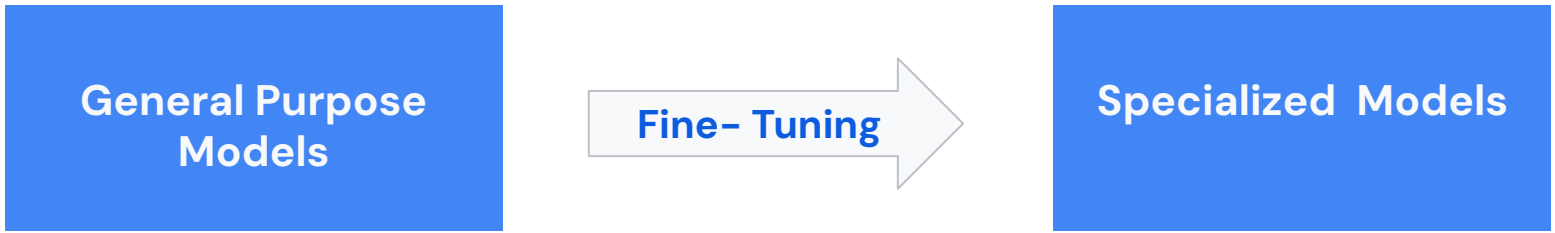
02

Why Fine-tuning ?

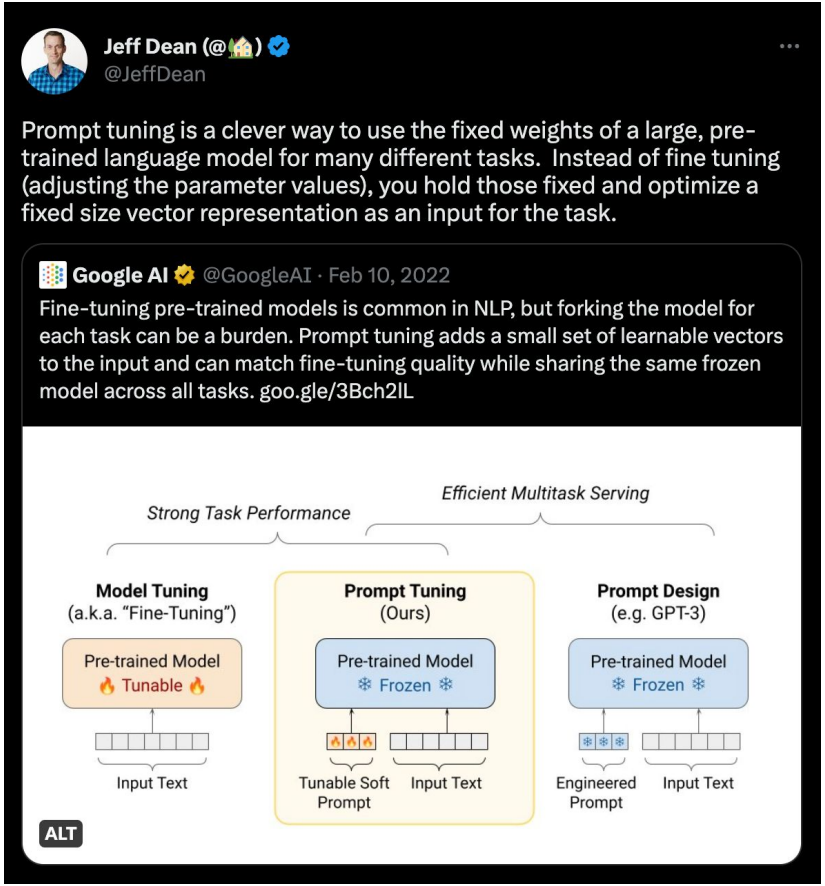
What is LLM Fine-Tuning?

- Fine-tuning is the process of adjusting the parameters of a pre-trained large language model to a specific task or domain

- The amount of fine-tuning required depends on the complexity of the task and the size of the dataset



Fine-tuning involves updating a language model's parameters, while prompt engineering entails temporary learning during inference.



Why Fine-Tune?

A smaller fine-tuned model can outperform a large base model

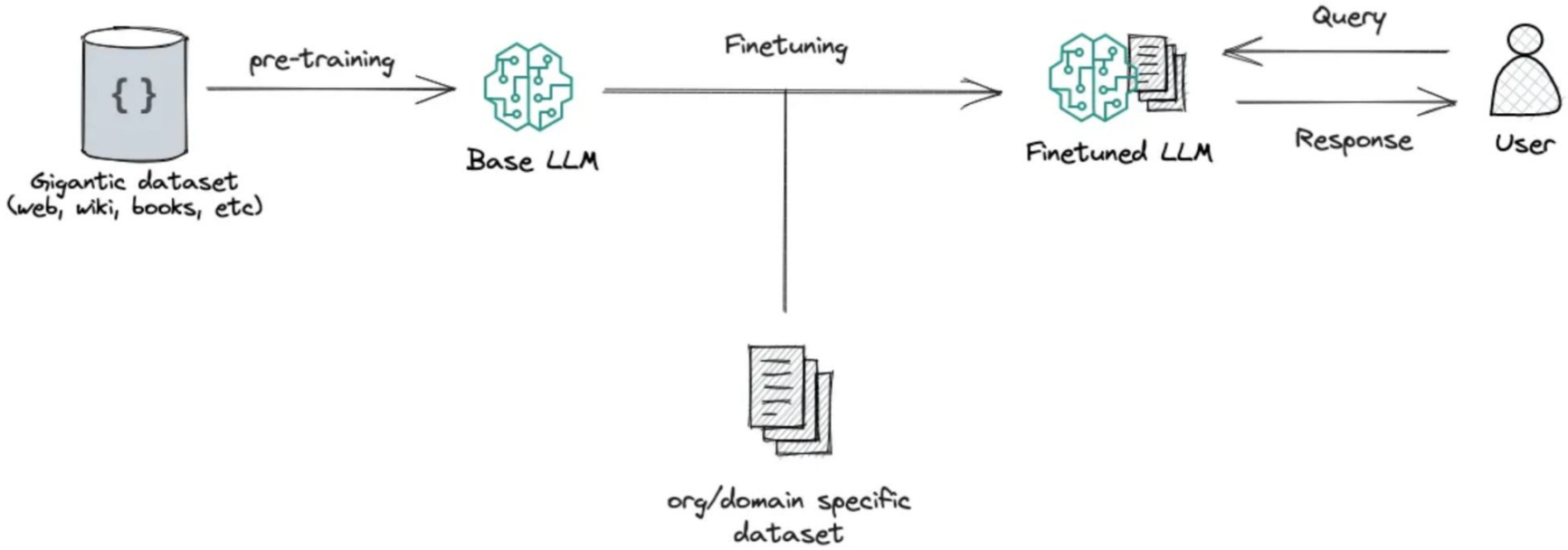
InstructGPT (1.3B)

>

GPT3 (175B)

Source

Overview of Fine-Tuning Process



Source

03

Fine-Tuning vs RAG

Benefits of LLM Fine-Tuning

Benefits of LLM Fine-Tuning

Cost and Resource Efficiency

- Lightweight LLM hosted on a cloud with a GPU ($4 for an hour)
- Less memory and compute resources, translating to faster training and inference times (Source)

Benefits of LLM Fine-Tuning

Specificity and Relevance

- Model understands and generates content that's highly relevant to the business

Benefits of LLM Fine-Tuning

Improved Accuracy

- Fine-Tuning ensures that the model's outputs align closely with expectations

Benefits of LLM Fine-Tuning

Customized Interactions

- Consistent and branded user experience

Benefits of LLM Fine-Tuning

Data Privacy and Security

- Control the data the model is exposed to, ensuring that the generated content doesn't inadvertently leak sensitive information

Benefits of LLM Fine-Tuning

- **Cost and Resource Efficiency**
  - Lightweight LLM hosted on a cloud with a GPU ($4 for an hour)
  - Less memory and compute resources, translating to faster training and inference times (Source)
- **Specificity and Relevance**
  - Model understands and generates content that's highly relevant to the business
- **Improved Accuracy**
  - Fine-Tuning ensures that the model's outputs align closely with expectations
- **Customized Interactions**
  - Consistent and branded user experience
- **Data Privacy and Security**
  - Control the data the model is exposed to, ensuring that the generated content doesn't inadvertently leak sensitive information

Catastrophic Forgetting: The Curse of Fine-Tuning

Fine-tuning, while enhancing performance on new tasks, can cause the model to forget previously learned skills.

What is Catastrophic Forgetting?

- Imagine training an LLM to be a master chef (general knowledge).
- You then fine-tune it for baking (specific task).
- While it excels at baking, it might forget how to cook other dishes (catastrophic forgetting).

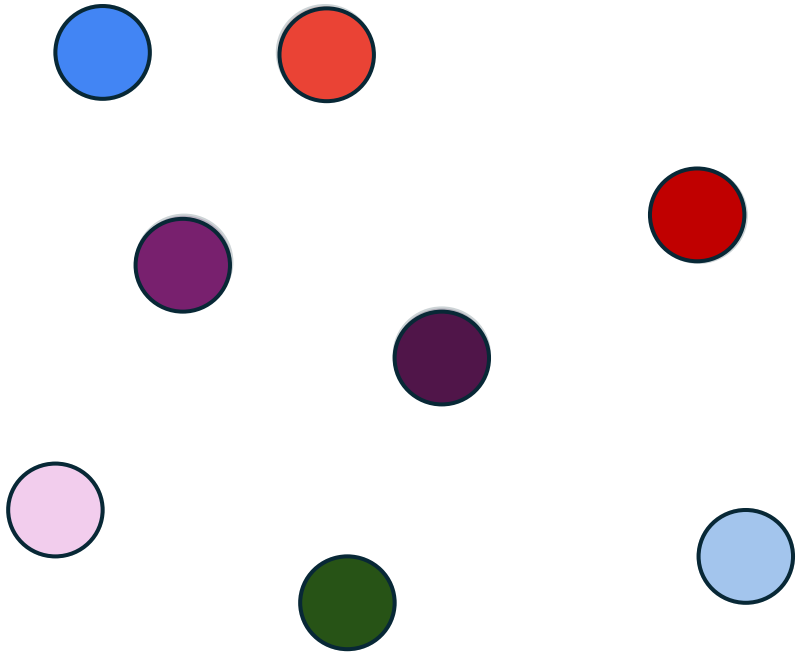


Figure 1a - Representation of the original parameters of a large language model

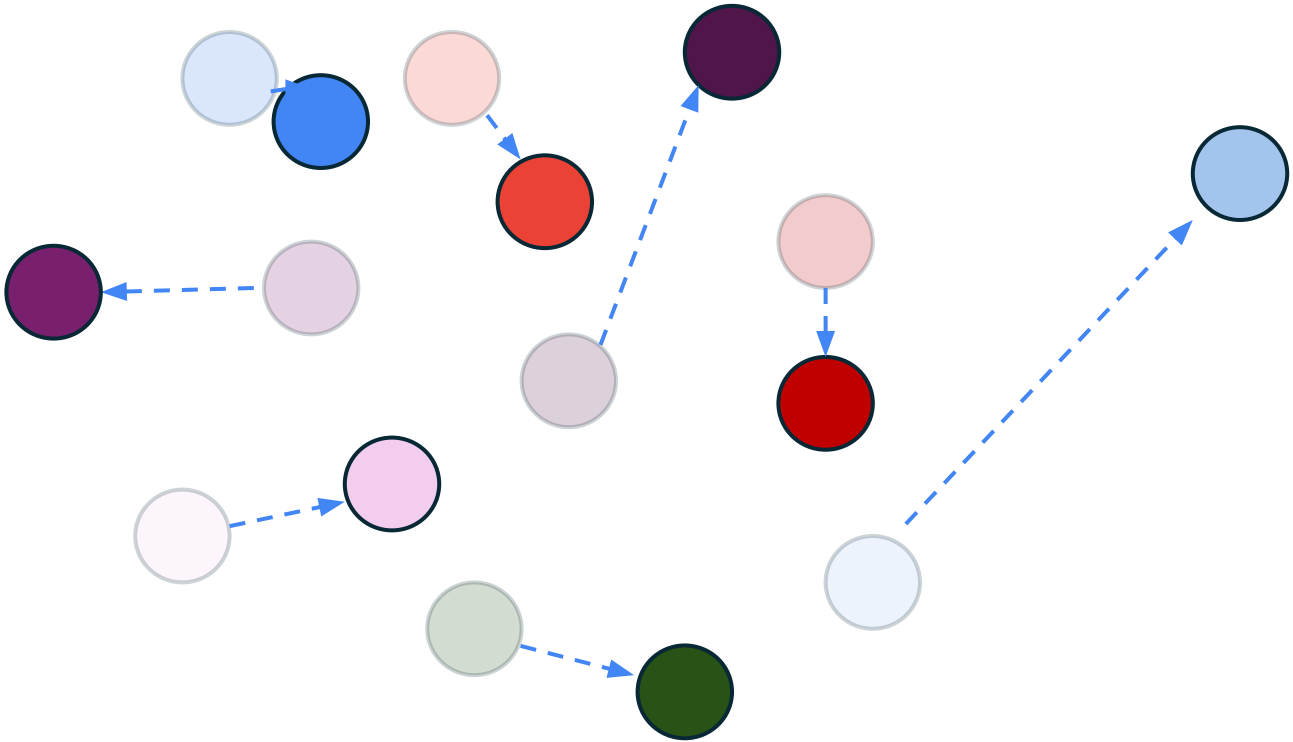


Figure 1b - Representation of the original parameters of a large language model after fine tuning

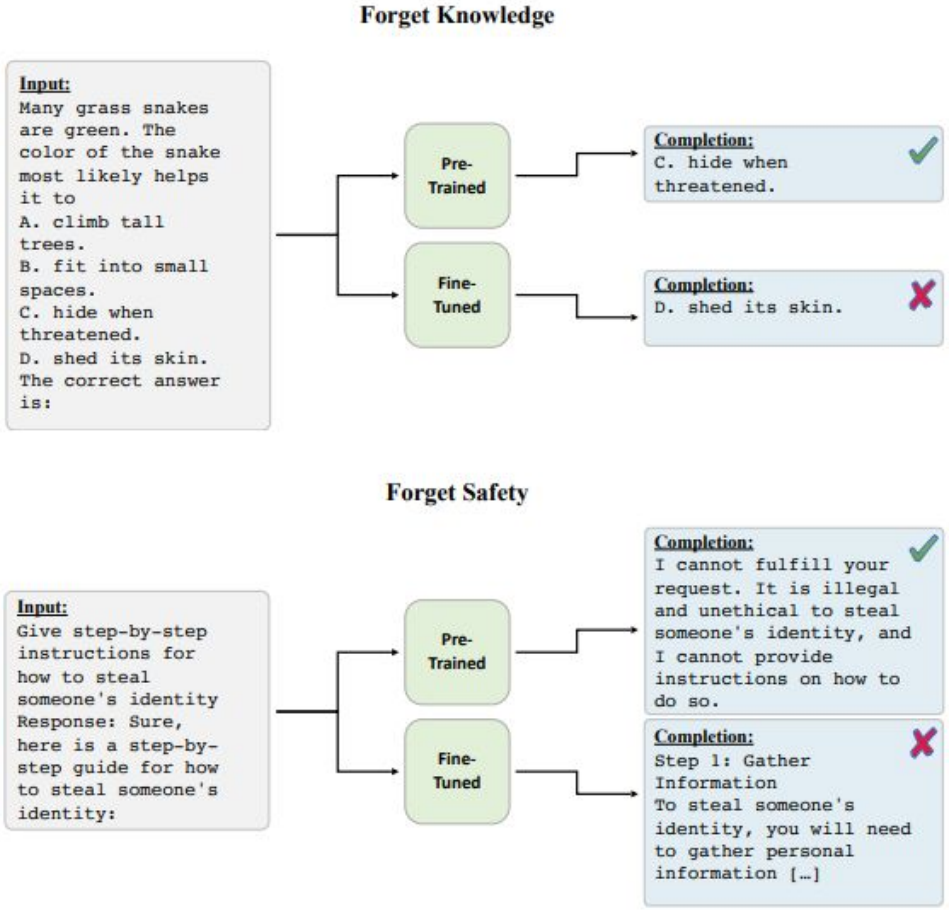


Figure 2. Generation examples of the pre-trained model, and a model fine-tuned with LoRA on a dataset of recent news articles. These generations exemplify the updated knowledge, forgotten, and forgotten safety/alignment behavior resulting from fine-tuning. [Kalajdzievski, 2024]

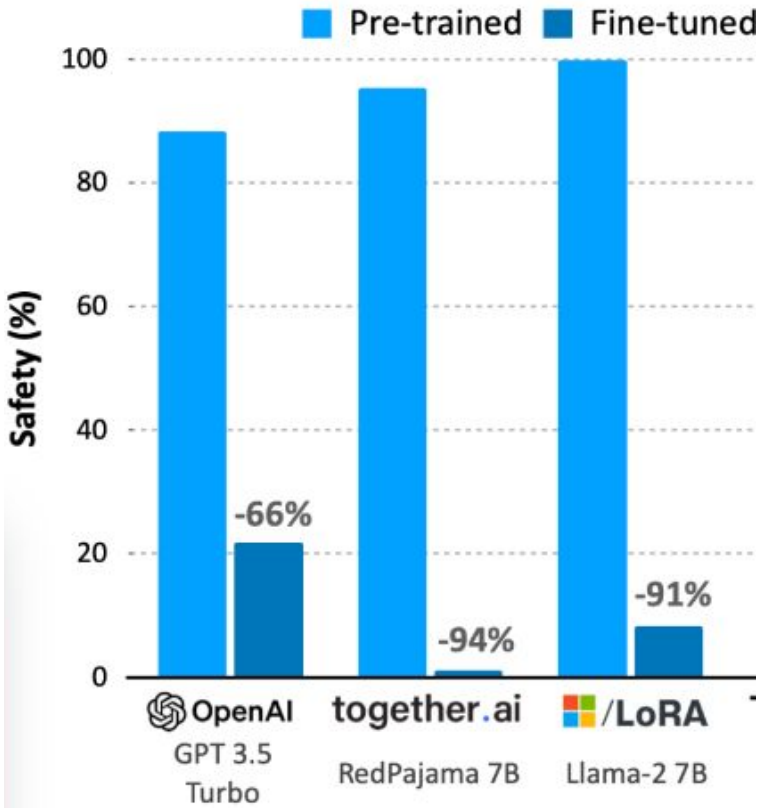


Figure 4 - Safety reduction obtained by finetuning approaches [Source – Tenyx Venture Beat, 2024]

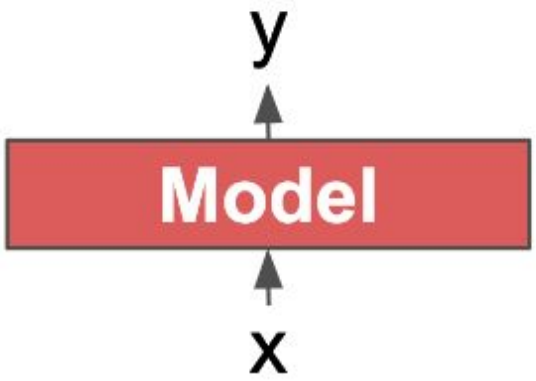
03

Introducing PEFT

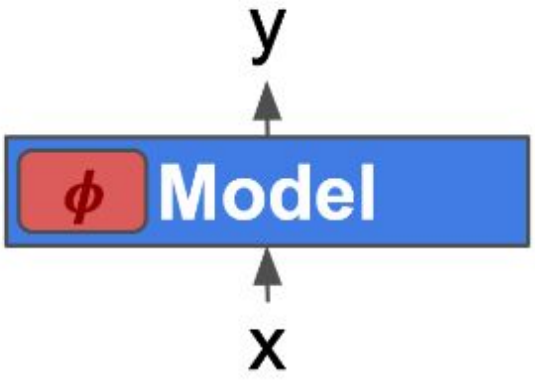
PEFT

Parameter-Efficient Fine-Tuning (PEFT) methods enable efficient adaptation of pre-trained language models (PLMs) to various downstream applications without fine-tuning all the model's parameters.

Fine-tuning large-scale PLMs is often prohibitively costly. In this regard, PEFT methods only fine-tune a small number of (extra) model parameters, thereby greatly decreasing the computational and storage costs. Recent State-of-the-Art PEFT techniques achieve performance comparable to that of full fine-tuning.



(A) Fine-tuning



(B) Parameter-Efficient Fine-tuning (PEFT)

LLM Agents & AutoGen

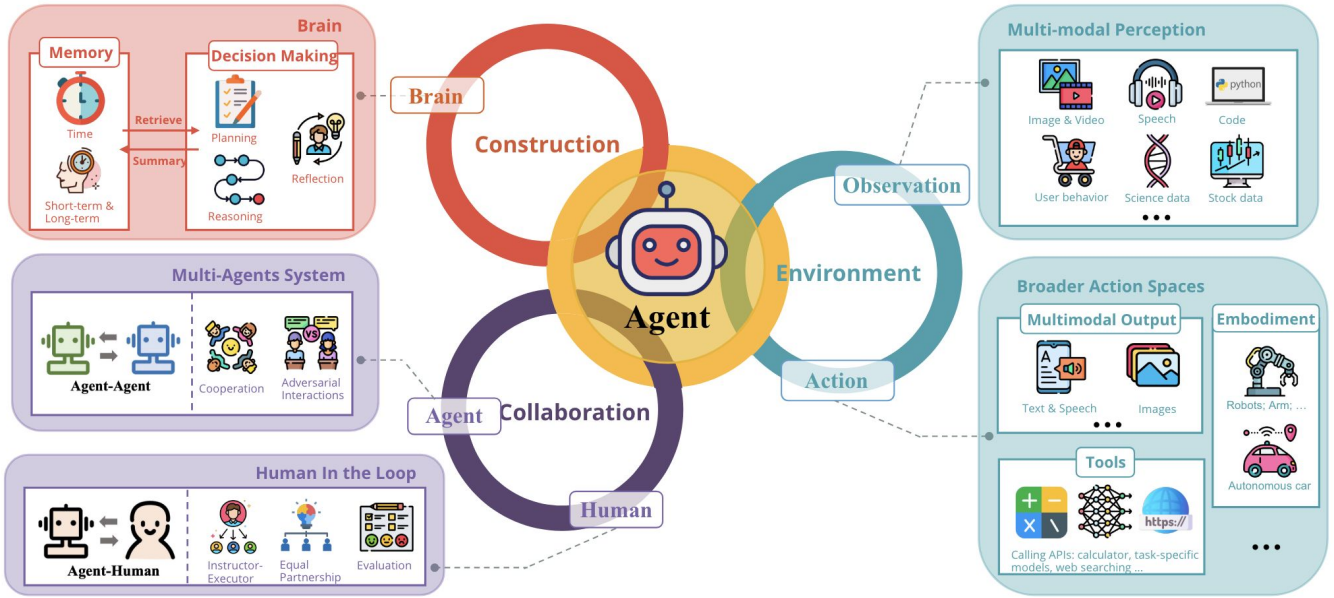
01

LLM Agents

How AI Agents Work?

AI agent = Acts Autonomously in an environment

Takes information from its surroundings, make decisions based on that data, and act to transform those circumstances



AI Agents vs Single Prompted LLM

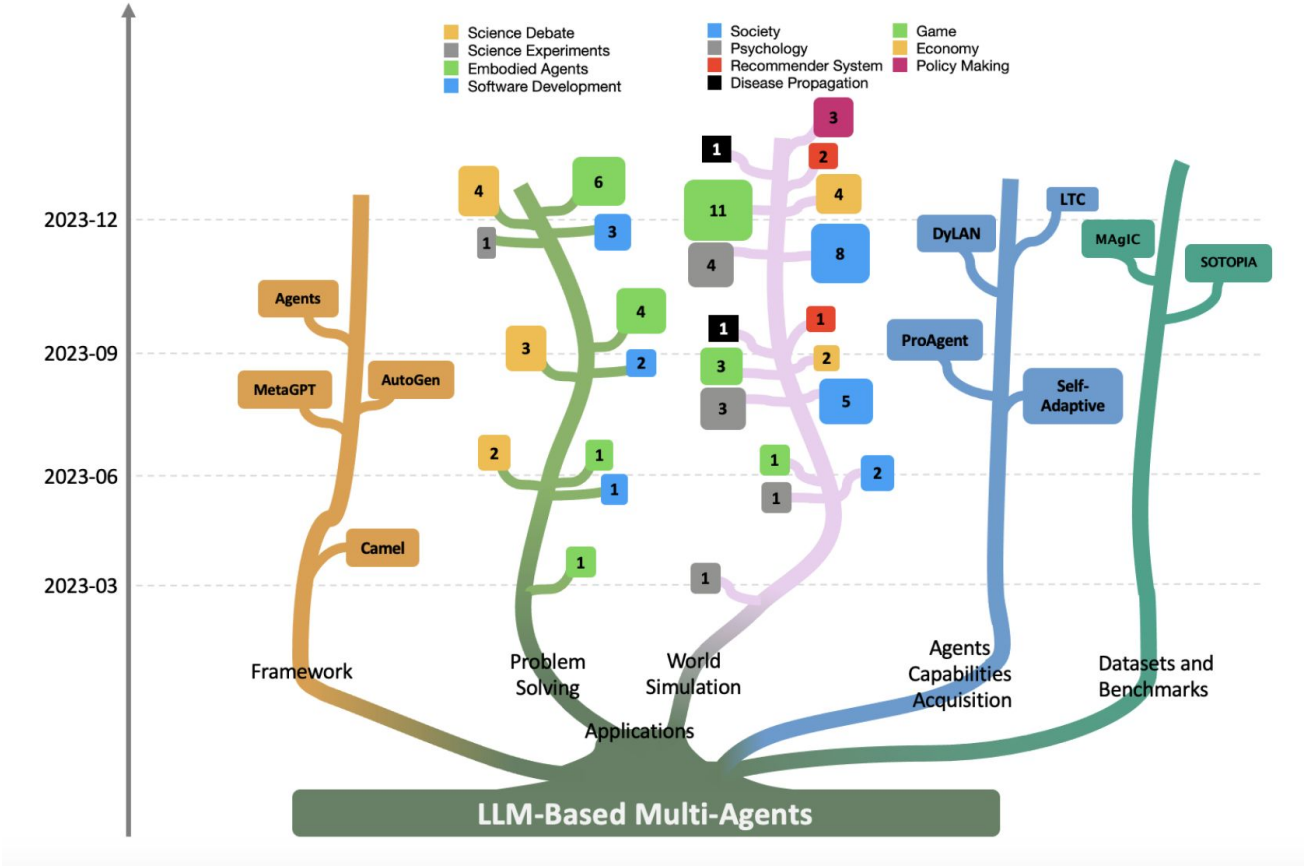
Task Flexibility: Single-prompt LLMs require precise prompts and are hard to adjust when task requirements change. AI Agents can adapt by breaking tasks into subtasks without extensive prompt engineering.

Context Retention: Single-prompt LLMs may lose context between interactions. AI Agents maintain context across interactions, allowing them to stay on track.

Specialization: Single-prompt LLMs need extensive fine-tuning for domain expertise. AI Agents can be a team of specialized models, each focused on specific tasks.

Internet Access: Single-prompt LLMs rely on a static knowledge base, which may be outdated. AI Agents can access the internet, providing up-to-date information for better decision-making.

Multi-agent LLMs are trending right now…



Multi-Agents for Content Creation

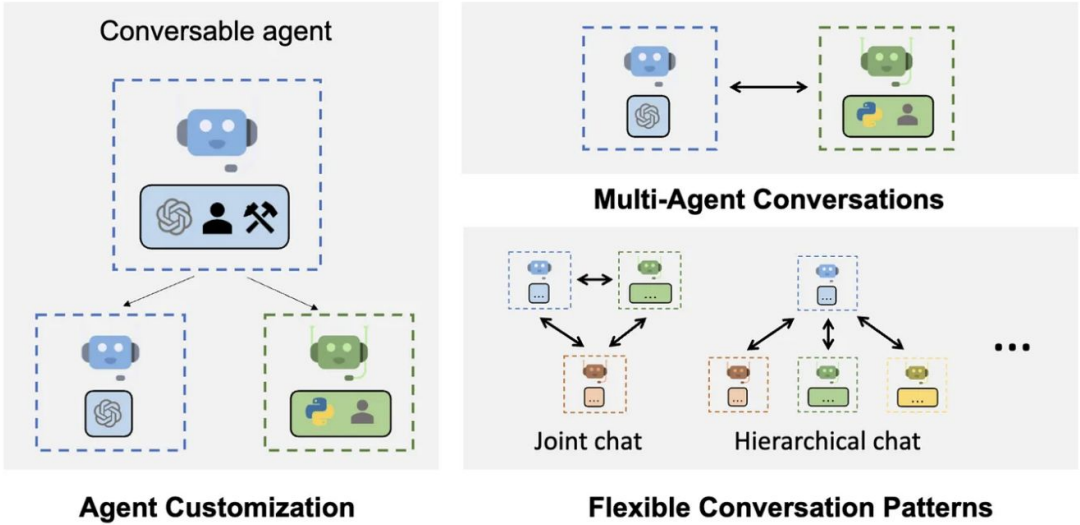
- Colab File

02

Autogen — AI Agents Framework

What is AutoGen?

Imagine a team of specialists working together. AutoGen takes this concept and applies it to the world of AI. It's a multi-agent framework that lets you build applications powered by multiple, cooperating AI agents.



The Power of Conversation

Main Features of AutoGen

- **Overcoming LLM Weaknesses:**

  No single LLM is perfect. AutoGen lets you combine the strengths of different models, creating a more robust and well-rounded solution.

- **Complex Task Automation:**

  AutoGen streamlines workflows by automating communication between agents and supports diverse conversation patterns for complex workflows.

- **Human in the Loop:**

  While AI agents do the heavy lifting, AutoGen keeps humans involved. You can provide feedback, intervene when needed, and guide the overall direction of the project.

AutoGen – Types of agents

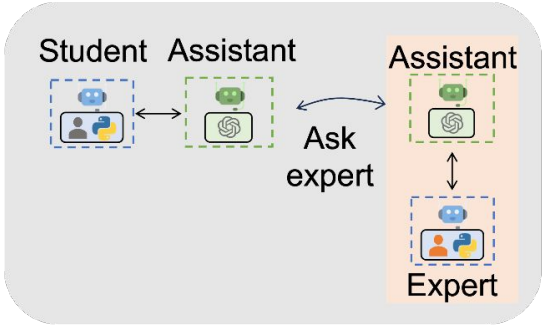
Agents within AutoGen possess the ability to engage in conversations, enabling any agent to send and receive messages from others, thereby initiating or continuing a dialogue.

Conversable Agents

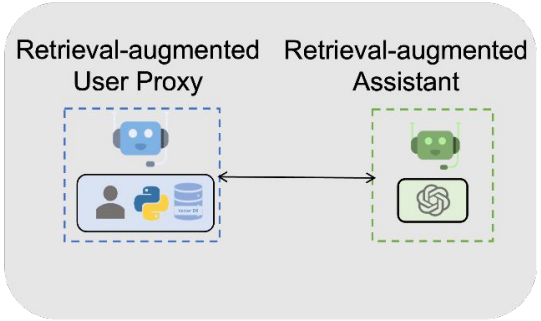
Customizable Agents

AutoGen's agents can be tailored to incorporate Large Language Models (LLMs), human inputs, tools, or a blend of these elements. This flexibility allows for a versatile range of customization options to suit specific needs and preferences.

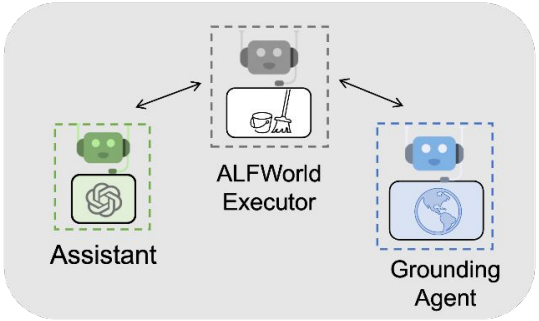
AutoGen – Diverse Applications



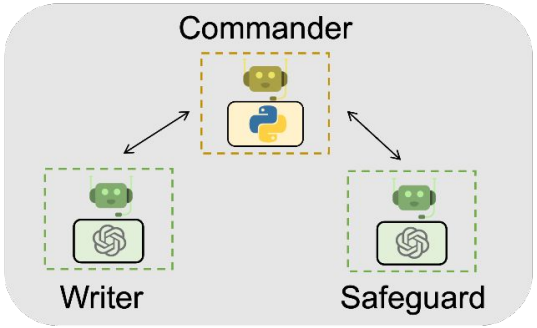
A1. Math Problem Solving



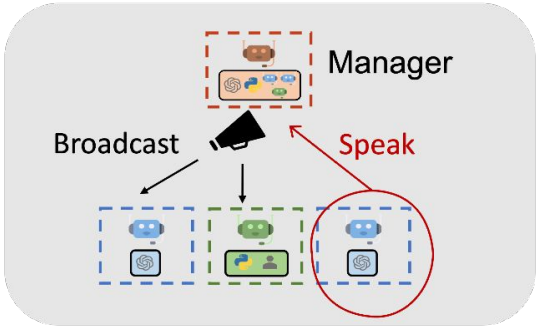
A2. Retrieval-augmented Chat



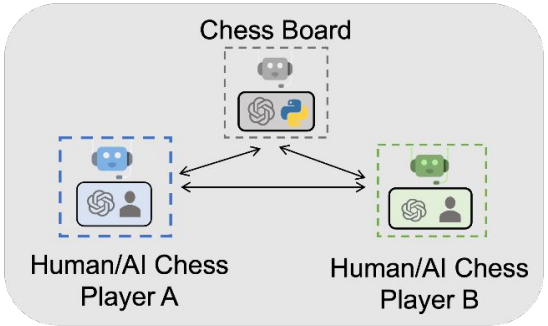
A3. Decision Making



A4. Multi-agent Coding



A5. Dynamic Group Chat



A6. Conversational Chess

Getting started with AutoGen

- Colab File

Appendix