LLMs in Practice

Lecture 12 part I

Foivos Tzavellos
f.i.tzavellos@uva.nl

University of Amsterdam

About me

MEng Electrical and Computer Engineering, University of Thessaly, Greece

MSc Data Science for Decision Making, Maastricht University, Netherlands

Philips AI for Vision and Language Team

Law and Tech Lab Maastricht University Faculty of Law

Junior Lecturer - Maastricht University (Bachelors DSAI & CS)

Junior Lecturer - Senior TA UvA (Masters AI)

Outline

1. Prompting
2. RAG
3. Parameter-efficient fine-tuning
4. Supervised fine-tuning
5. Full Fine-tuning
6. Data Quality & Evaluation
7. Method Decision Framework

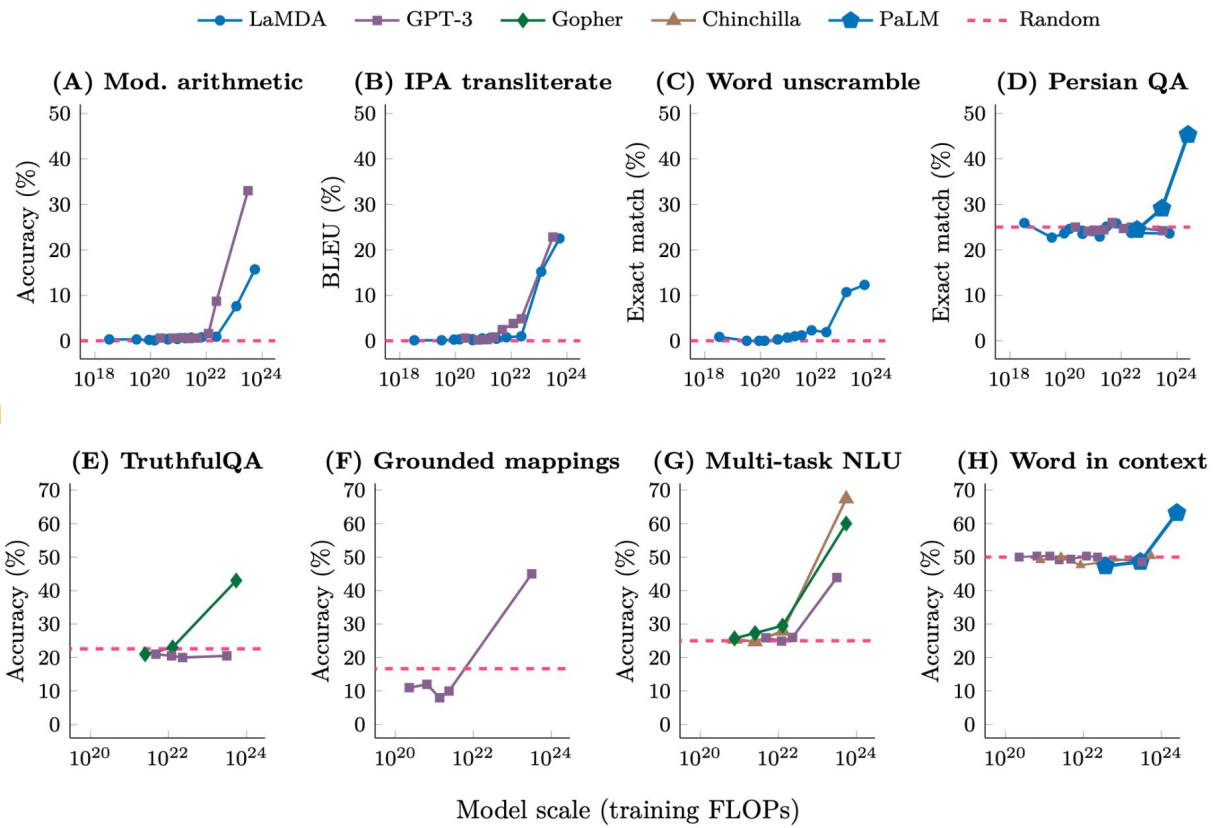
least effort -> most effort

Effort: computational costs (time + money + hardware requirements), engineering work, data requirements

Emergent Capabilities

Figure from Wei et al. [2022]

Emergent capabilities in language models are those abilities that are not observed in smaller models but manifest only when the models reach a certain size



Prompting: why it matters?

We demonstrate that GPT-3 can perform tasks with **no gradient updates or fine-tuning**, sometimes reaching performance competitive with fine-tuned models.

Brown et al. [2020]

- Modern LLMs already encode many latent capabilities
- Prompting accesses these capabilities without training
- Cheap, fast, no infrastructure

In-context learning: the model learns a task based on inputs (or prompts) given at inference time, **without updating its parameters**.

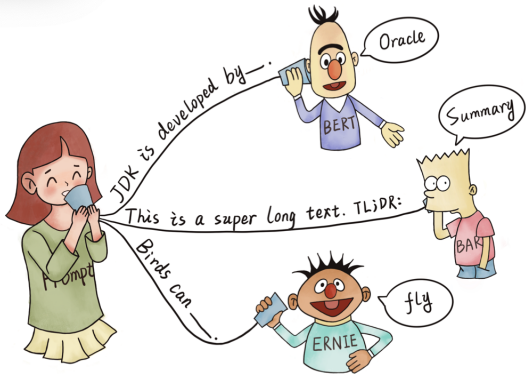


Figure from Liu et al. [2021]

Elements of the prompt

Context

external information or additional context that can steer the model to better responses

Instruction

a specific task or instruction you want the model to perform

You are an expert sentiment analyser

Classify given text into positive, negative and neutral

Text: I enjoy NLP1

Sentiment:

Input Data

the input that we are interested to find a response for

Output Indicator

the type of format for the output

Zero-shot Prompting

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 ✗

Figure from Kojima et al. [2023]

Simplest way to prompt, instructions only, no examples

Works well when the task is clearly described and similar to patterns seen during pretraining and the output space is constrained

Guides prompt design before spending time on data collection or fine-tuning.

Few-shot Prompting

Figure from Kojima et al. [2023]

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.

example

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:

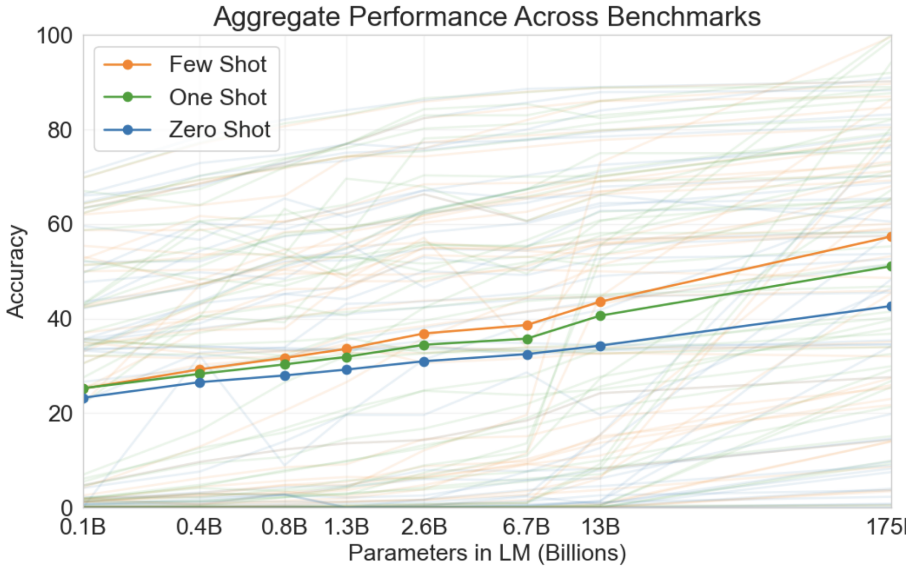
(Output) The answer is 8. ✗

Helps when task is underspecified, the output needs a specific structure, labels are non-obvious

Figure from Brown et al [2020]

Introduced by Brown et al.[2020] from OpenAI

One-shot -> one example included in the prompt
Two-shot -> two examples etc…



Chain-of-Thought Prompting

Introduced by Wei et al. [2023] from the Google Research team
The model is explicitly asked to explain its reasoning steps (chain of thought) before the answer

"Let's think step by step."

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: 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:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls. The answer is 4. ✓

In this example they combine one-shot with CoT

Figure from Kojima et al. [2023]

Alternative Prompting Techniques I

Role-play Prompting: Assign a role to the LLM (“You are a clinical psychologist. Answer as you would in a report.”) Kong et al. [2024]

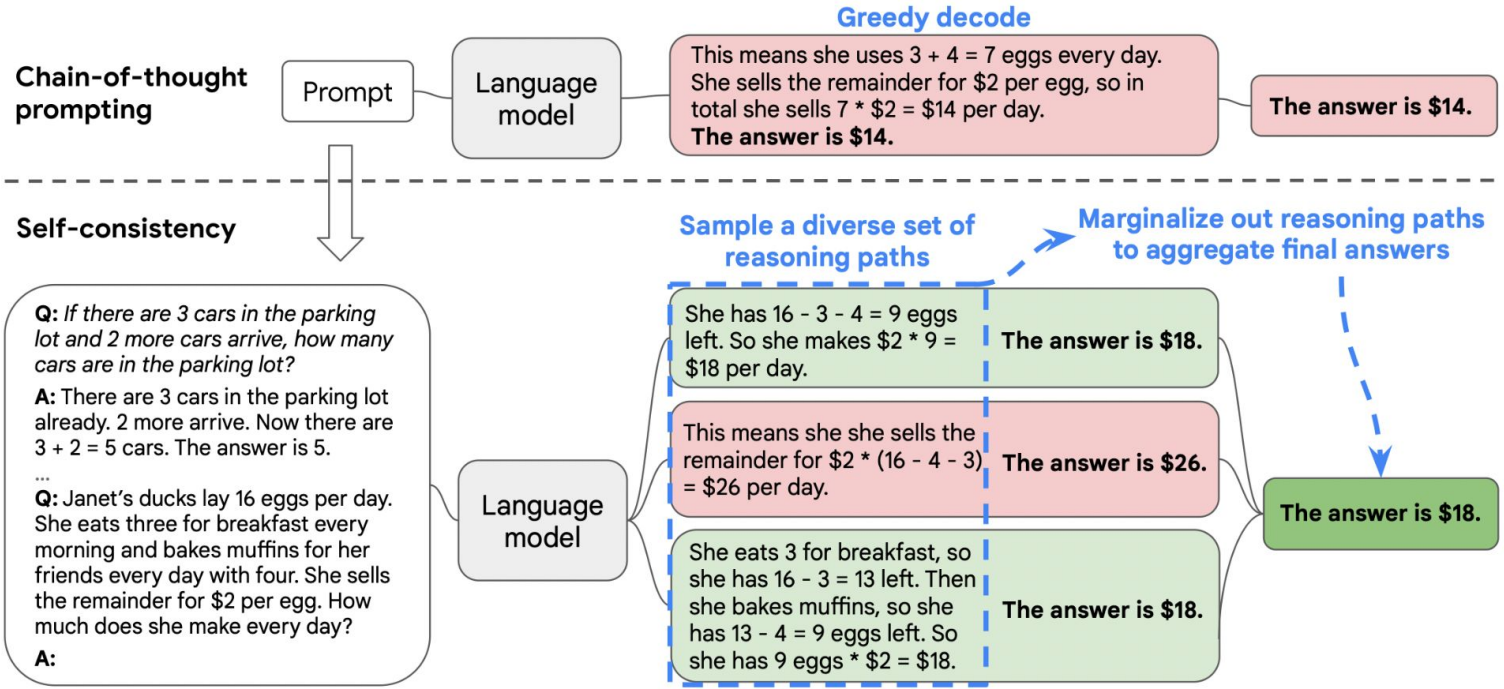
Meta Prompting: Prompting technique that emphasizes the structural and syntactical aspects of problems by prioritizing the overall format and pattern over specific content details. Zhang et al. [2024]

Integrate step-by-step reasoning to solve mathematical problems under the following structure:

{
”Problem”: ”[question to be answered]”,
”Solution”: {
”Step 1”: ”Begin the response with “Let’s think step by step.””,
”Step 2”: ”Follow with the reasoning steps, ensuring the solution process is broken down clearly and logically.”,
”Step 3”: ”End the solution with the final answer encapsulated in a LaTeX-formatted box, $\boxed{\ldots}$, for clarity and emphasis.”
},
”Final Answer”: ”[final answer to the problem]”
}

Alternative Prompting Techniques II

Self-consistency COT: Generate multiple diverse chain-of-thoughts via sampling and take the majority answer, replacing the single, brittle reasoning path produced by greedy decoding. Wang et al [2022]



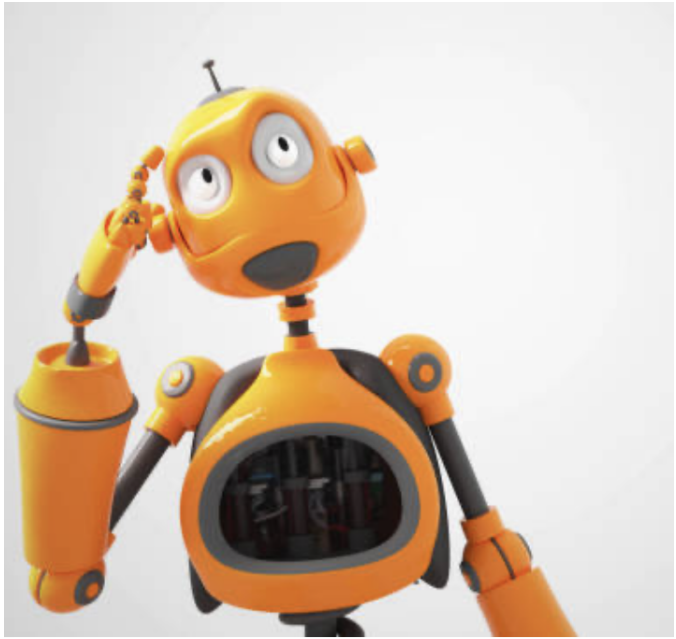
Limitations of Prompting

Prompting changes the *input*, not the *model*.
When the model lacks knowledge, stability, or skills, prompting hits a ceiling.

Cannot add new knowledge

Unstable behaviour

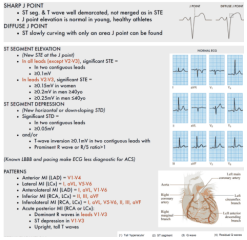
No way to enforce domain style



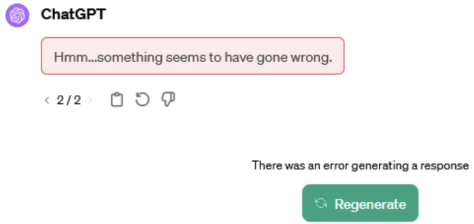
Problem Statement

Medical device manufacturing and service company

ECG machine malfunction



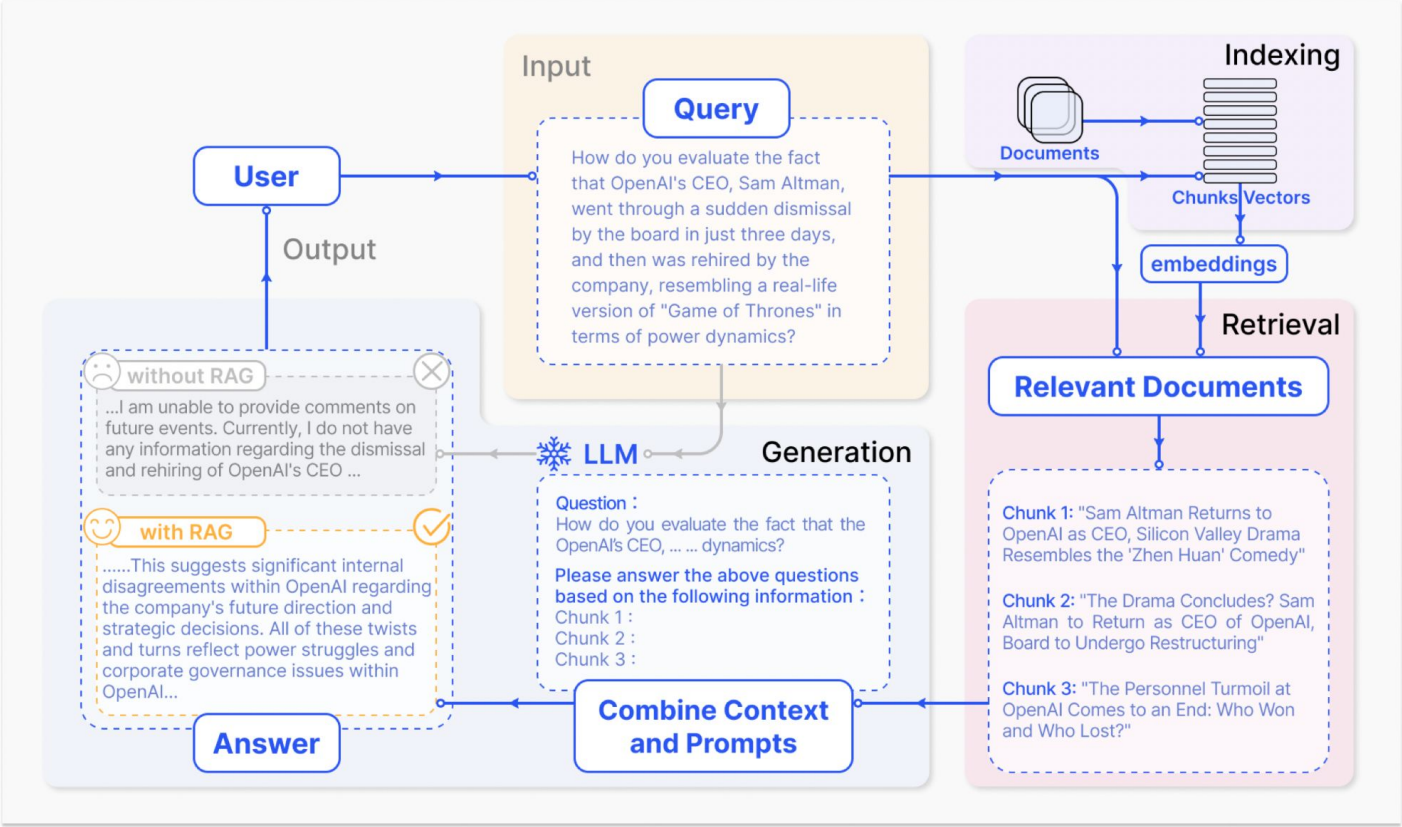
Thousands of pages of internal documents (manuals)



ChatGPT cannot draw knowledge from proprietary documents

Possible Solution - RAG

Retrieval **A**ugmented **G**eneration Lewis et al [2020]



Indexing

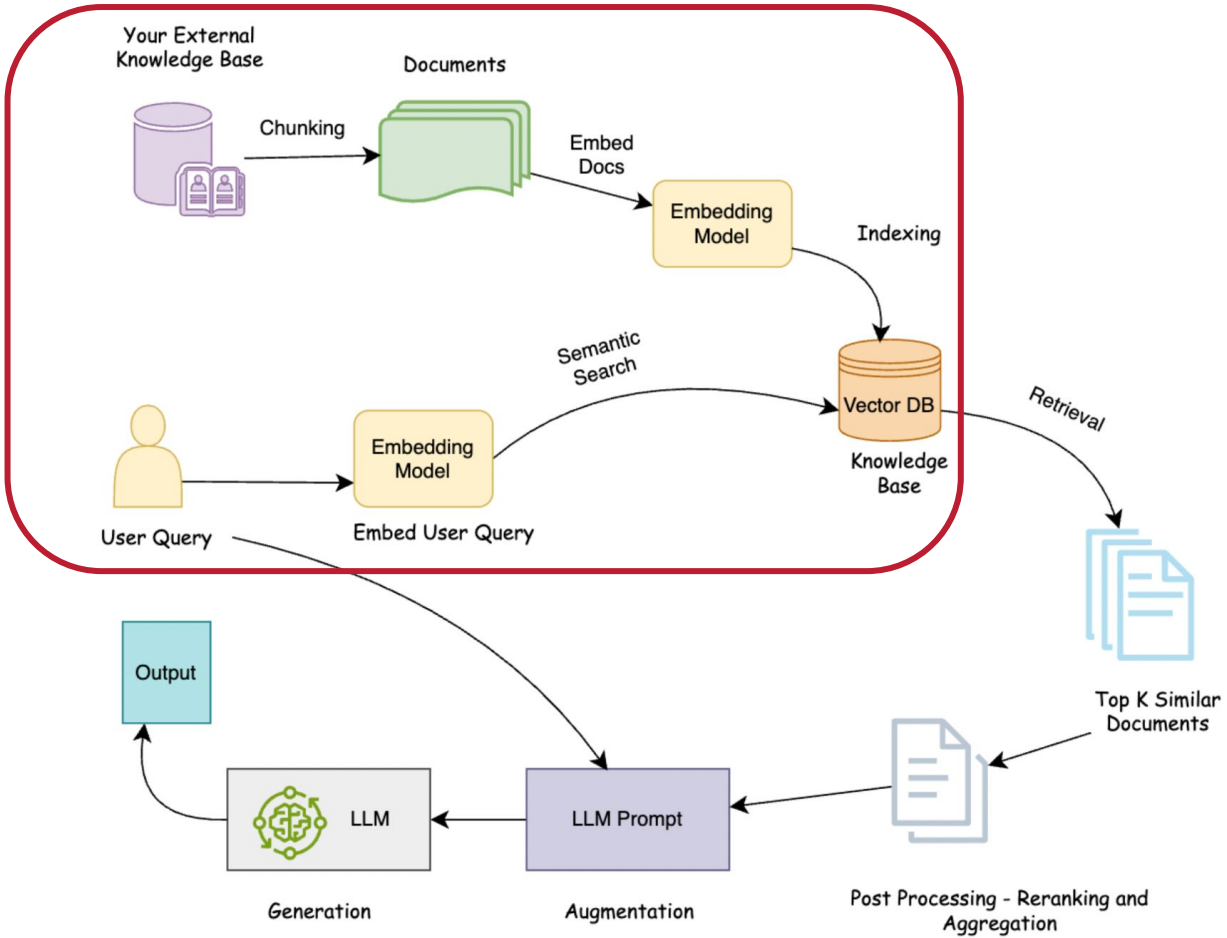


Image source

Retrieval

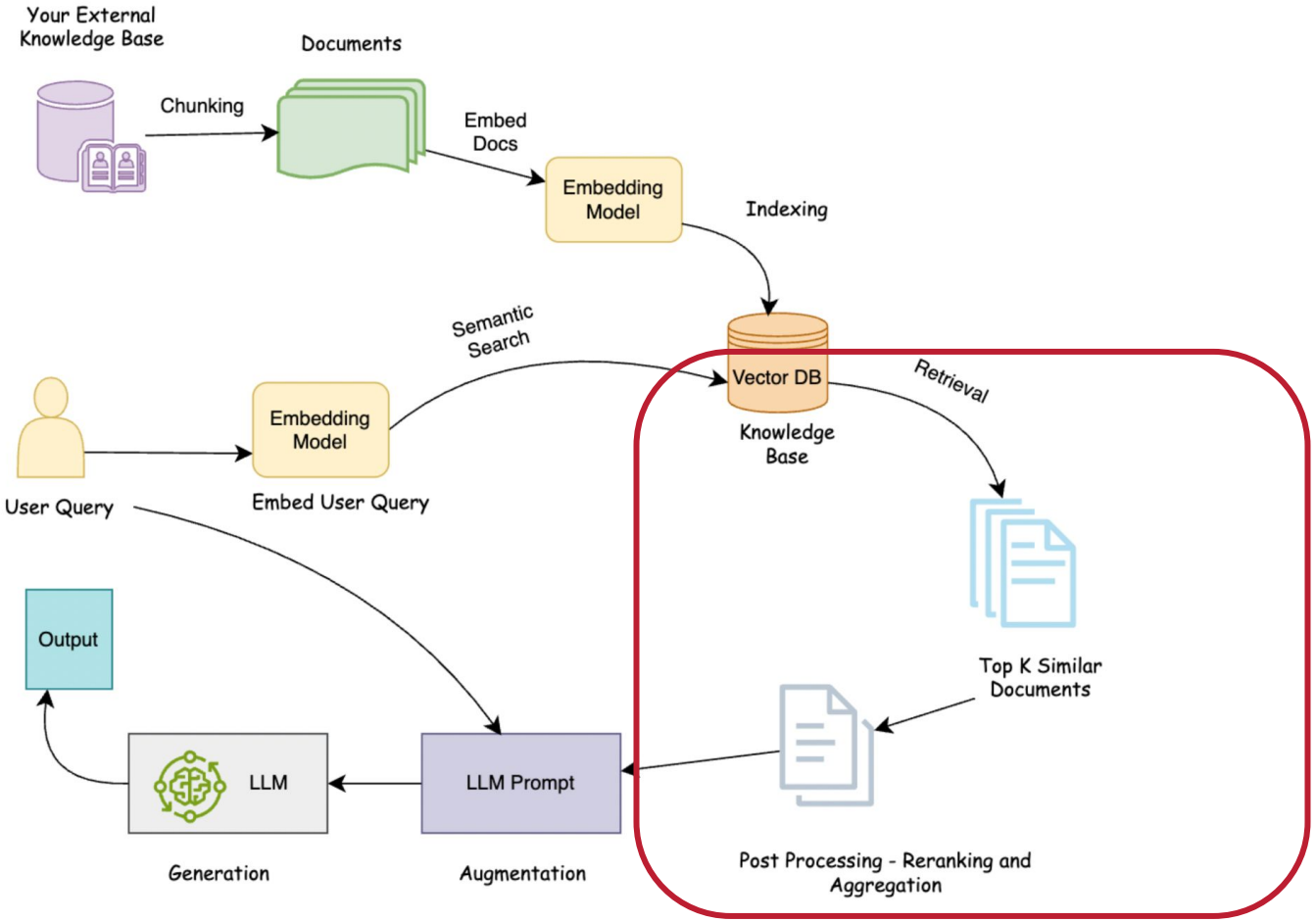


Image source

Generation

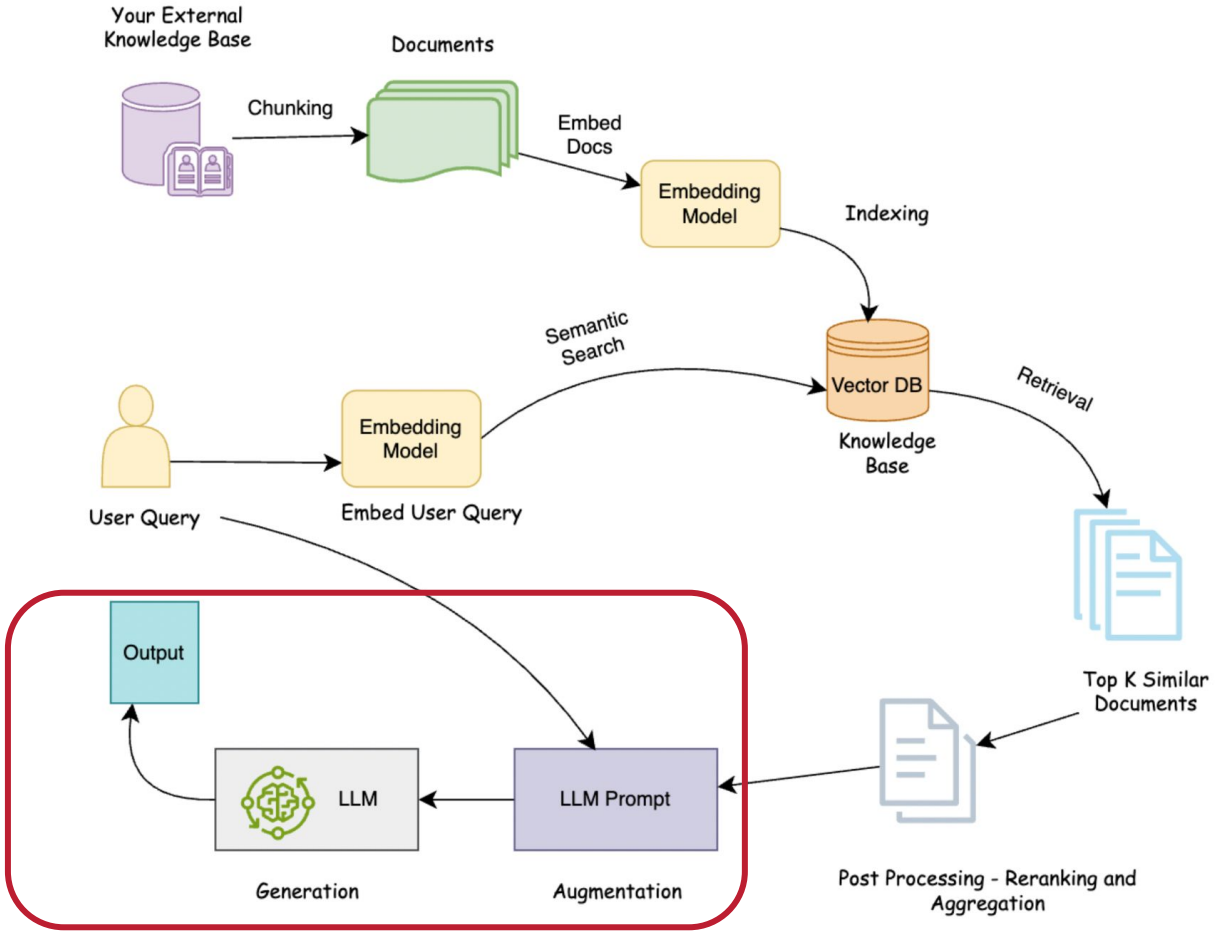
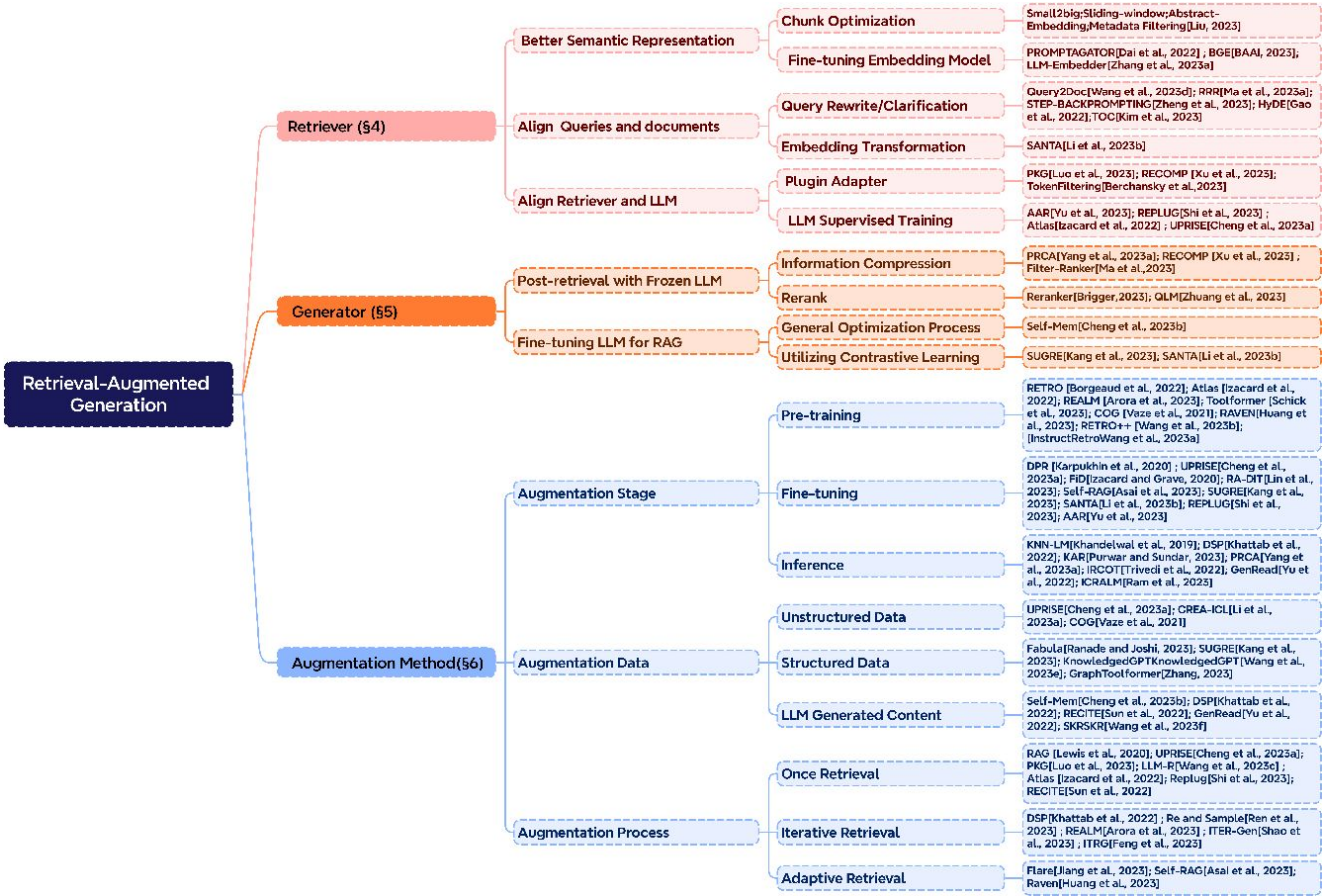


Image source

Different Architectures

Retrieval Augmented Generation for Large Language Models: A survey (Gao et al. [2024])



Problem Persists

Knowledge is not the only limitation

We have the right knowledge, but not the right behaviour

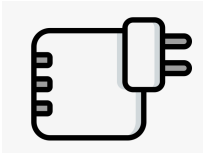

```
{
   glossary: {
      title: "example glossary",
      GlossDiv: {
         title: "S",
         GlossList: {
            GlossEntry: {
               ID: "SGML",
               SortAs: "SGML",
               GlossTerm: "Standard Generalized Markup Language",
               Acronym: "SGML",
               Abbrev: "ISO 8879:1986",
               GlossSee: "markup"
            }
         }
      }
   }
}
```

Parameter Efficient Fine-tuning

Adapting an LLM's behaviour without retraining the model

Learn small components that help 'adapt' the model

- Freeze all original LLM weights
- Add tiny trainable modules (<1% of parameters) or rethink the way weight matrices are used
- Learn task-specific behaviour from small datasets

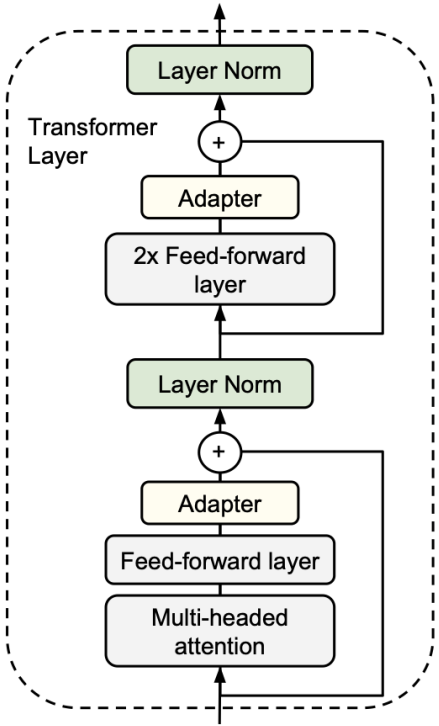


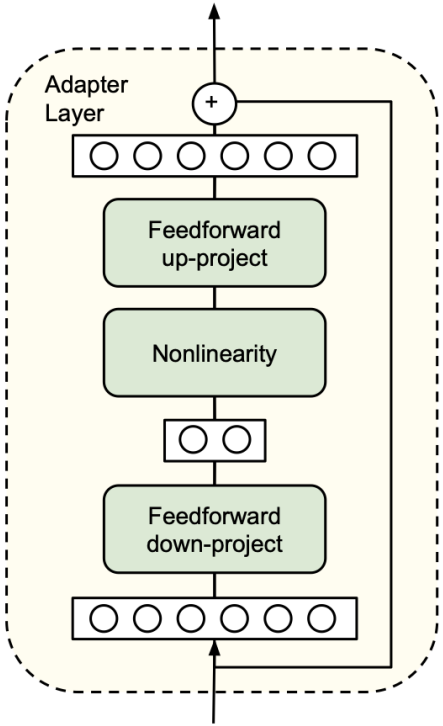
- Fits on consumer-grade GPUs (even single-GPU setups)
- Requires much less memory (small model state)
- Modules are lightweight add-ons that can be swapped or combined

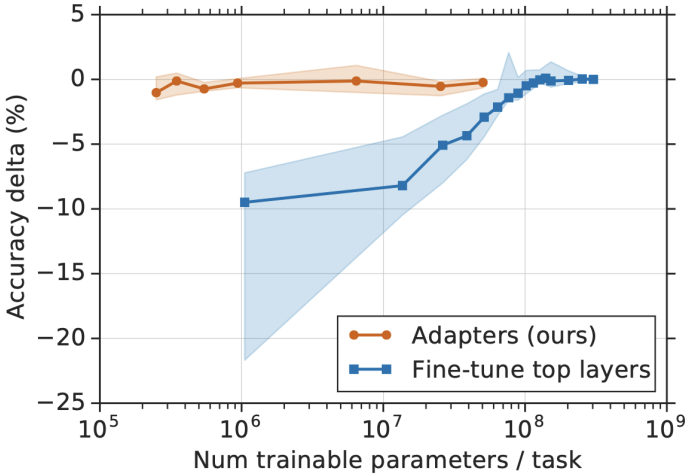
- Reduces risk of catastrophic forgetting
- Lower chance of overfitting on small domain datasets

Adapters

Parameter-Efficient Transfer Learning for NLP Houlsby et al. [2019]



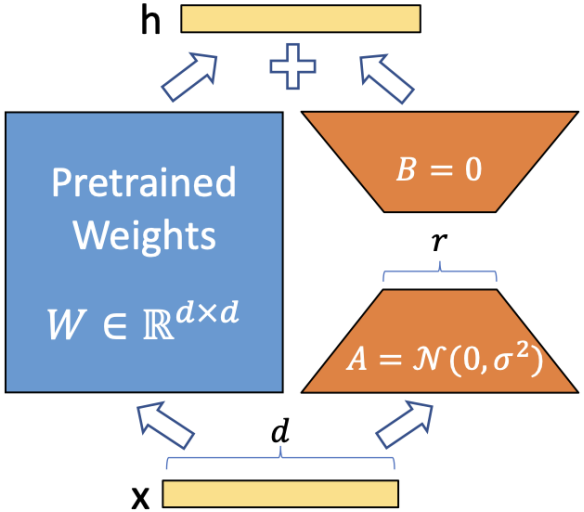




See also Adapter Fusion (Pfeiffer et al. [2020]): independently train various adapters then learn to 'fuse' them (combine them via a trained attention mechanism).

LoRA

LoRA: Low-Rank Adaptation of LLMs (Hu et al. [2021])



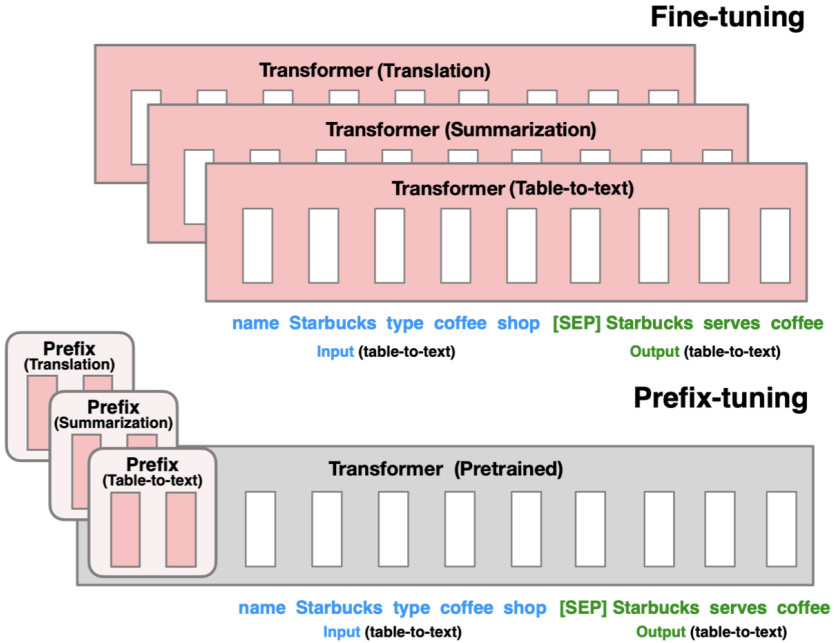
Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 ($\text{Adapter}^{\text{H}}$)	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 ($\text{Adapter}^{\text{H}}$)	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

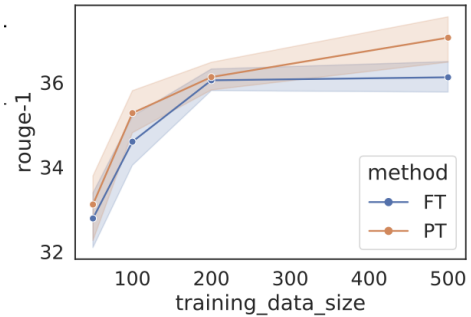
W0 + ΔW = W0 + BA

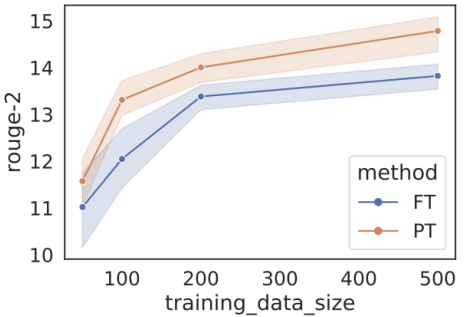
See also QLoRA: Efficient Finetuning of Quantised LLMs (Dettmers et al. [2023])

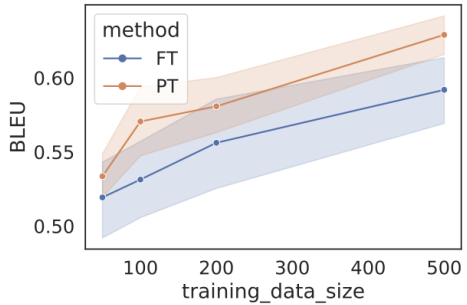
Prefix Tuning

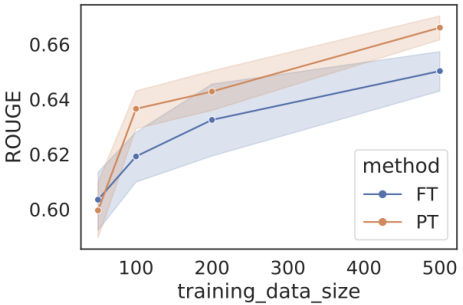
Prefix Tuning: Optimizing Continuous Prompts for Generation Li et al. [2021]





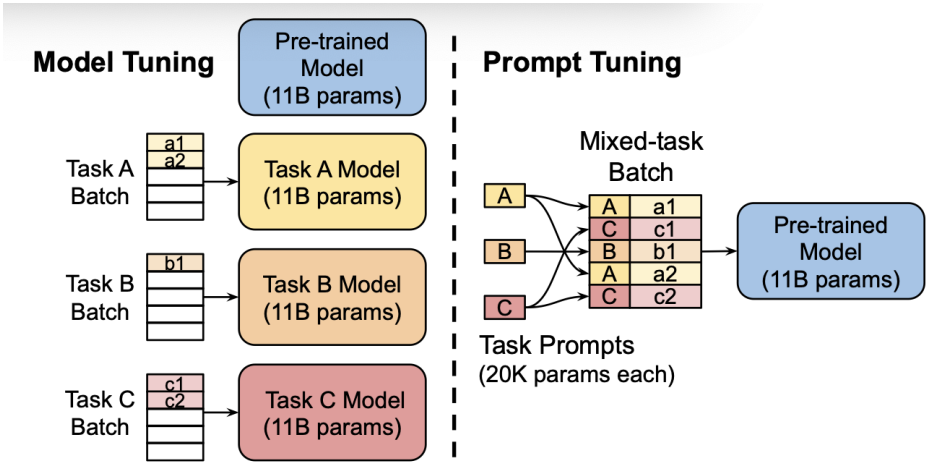


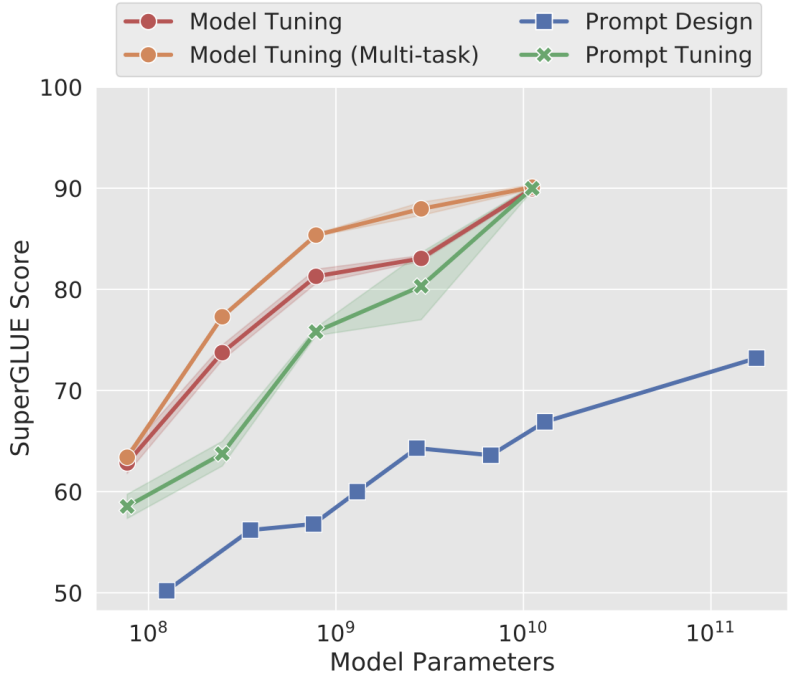




Prompt Tuning

The Power of Scale for Parameter-Efficient Prompt Tuning (Lester et al. [2021])





Prompt tuning keeps the model frozen and learns only a small set of continuous “prompt” embeddings prepended to the input, steering the model toward a new task with minimal extra parameters.

Instruction Tuning

Teach the model to follow instructions using supervised examples

Finetuned Language Models are Zero-Shot Learners (Wei et al. [2021])

Finetune on many tasks ("instruction-tuning")

Input (Commonsense Reasoning)

Here is a goal: Get a cool sleep on summer days.

How would you accomplish this goal?

OPTIONS:

-Keep stack of pillow cases in fridge.

-Keep stack of pillow cases in oven.

Target

keep stack of pillow cases in fridge

Input (Translation)

Translate this sentence to Spanish:

The new office building was built in less than three months.

Target

El nuevo edificio de oficinas se construyó en tres meses.

Sentiment analysis tasks

Coreference resolution tasks

...

Inference on unseen task type

Input (Natural Language Inference)

Premise: At my age you will probably have learnt one lesson.

Hypothesis: It's not certain how many lessons you'll learn by your thirties.

Does the premise entail the hypothesis?

OPTIONS:

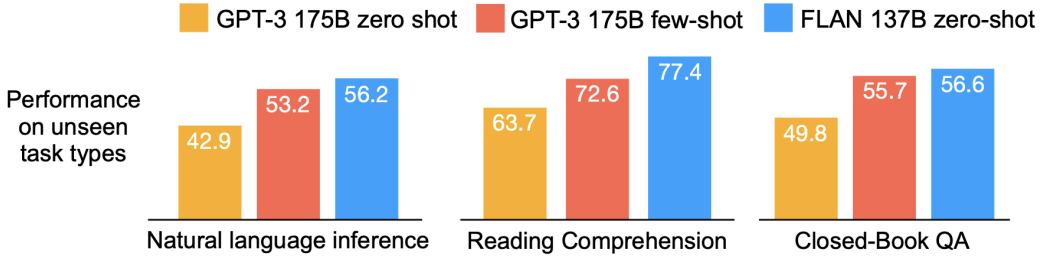
-yes -it is not possible to tell -no

FLAN Response

It is not possible to tell

Trained on (instruction,response) pairs

Helps the model internalise general instruction-following behaviour



Supervised Fine-tuning

Teach the model a specific task from labelled examples

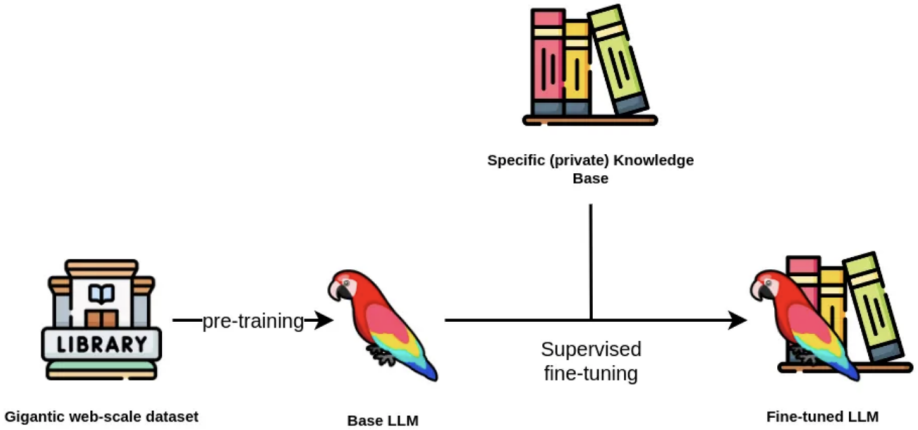


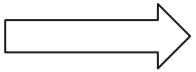
image source

- Take a pretrained LLM
- Train it further on a smaller dataset of (input -> target output pairs)
- Optimise a supervised loss (usually cross-entropy on next tokens)
- The model learns to imitate the desired behaviour of the task

University of Amsterdam

SFT Example

Patient reports a 3-day history of worsening shortness of breath and dry cough. No fever, but mild wheezing noted on exam. Past history significant for asthma. Started on inhaled bronchodilator with good initial response. Plan to reassess in 48 hours




```
{
  "chief_complaint": "Worsening shortness of breath and dry cough for 3 days",
  "history_of_present_illness": {
    "duration": "3 days",
    "symptoms": [
      "shortness of breath (worsening)",
      "dry cough",
      "no fever",
      "mild wheezing"
    ],
    "relevant_history": "Asthma"
  },
  "physical_exam": {
    "respiratory": "Mild wheezing noted; no signs of fever"
  },
  "diagnosis": {
    "description": "Asthma exacerbation",
    "icd10_code": "J45.909"
  },
  "treatment": {
    "medications": [
      {
        "name": "Inhaled bronchodilator",
        "response": "Good initial response"
      }
    ]
  },
  "plan": {
    "follow_up": "Reassess in 48 hours"
  }
}
```


a few thousand examples of clinician note -> correct structured example

- outputs invalid JSON
- adds or removes fields
- invents diagnoses that were never mentioned
- hallucinates icd10 codes
- writes disclaimers (“I am not a doctor”)

prompt

SYSTEM: You convert free-text clinical notes into structured EHR JSON.
NOTE: Patient reports a 3-day history of worsening shortness...
OUTPUT:

model predicts tokens ⟹ Compute loss between predicted tokens and target JSON ⟹ backpropagate update weights

Full Fine-tuning

Update **100%** of the model parameters

you can teach the model new reasoning abilities, new knowledge, new domains
requires GPU Clusters (very expensive)
risk of catastrophic forgetting

Who does that? OpenAI ANTHROPIC Meta

Large research labs , very large companies (Bloomberg, DataBricks)

Just do PEFT (95-100 percent of the performance on <1% of the cost and risk)

Fine-Tuning Requirements

- Let's fine-tune a Ψ=7.5B Transformer using Adam with mixed precision
  We need 16Ψ GB (=120 GB) of memory for the "model states" alone:
  - **Model:** 2Ψ bytes (for fp16 parameters)
  - **Gradients:** 2Ψ bytes (for fp16 parameters)
  - **Optimiser:** 12Ψ (for fp32 parameters, momentum and variances)
- **Activations**
  For a GPT2 architecture (NumLayers=48 Layers, HiddenDim=1600 units) with SeqLen=1K and BatchSize=32, we need about 60 GB of memory

Solution -> Multi-GPU training

Catastrophic Forgetting

Neural networks could overwrite older knowledge when trained on new data

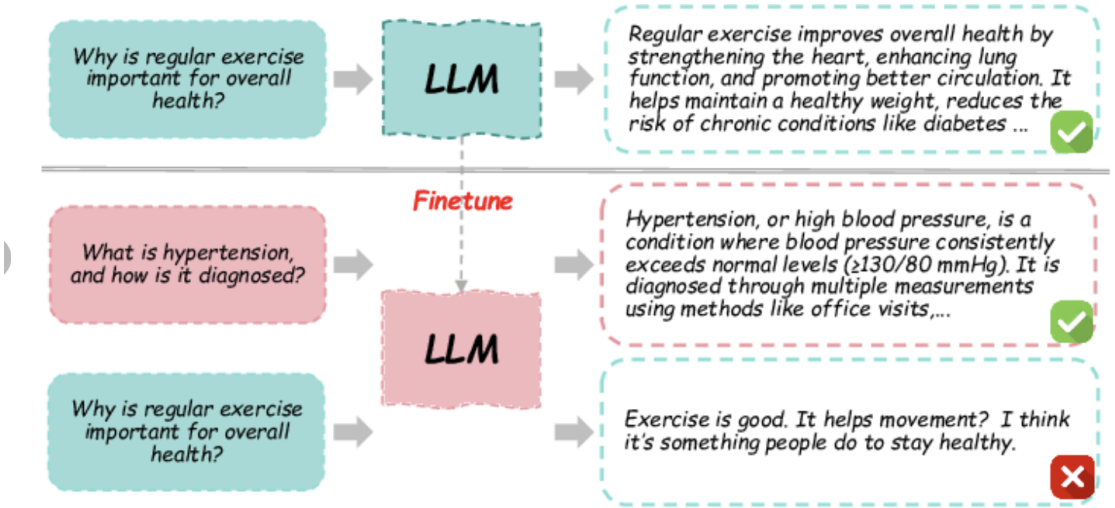


image source

- can affect reasoning, language understanding etc
- reason to continuously evaluate, even on previous benchmarks
- PEFT does not suffer from this limitation

Data Quality & Evaluation

Most LLM failures come from bad data and weak validation, not from model choice

Data Quality Problems

- inconsistent labels
- missing edge cases
- poor formatting
- non-representative dataset

Why evaluation is hard

- correctness is not always binary
- annotations can be expensive
- sensitivity to formatting or phrasing
- difficult to define ‘ground truth’

Evaluation Good Practices

- holdout validation split
- gold-standard test set curated by human experts
- LLM judge (?) with human-in-the-loop
- task-specific metrics

Knowledge from the streets

- 80% of time dealing with data
- 15% of time writing evaluations
- 5% of the time coding, configuring

See also A survey on Evaluation of Large Language Models Chang et al. [2023]

Decision Framework

How to Decide: Prompting, RAG, PEFT, or Full FT?

Prompting

- *don't use a bulldozer to kill a fly*
- task is light
- you need behaviour style changes
- no strict constraints
- no proprietary data needed

RAG

- the model lacks the needed knowledge
- you cannot finetune (frequent data updates)
- you want explainable, source-grounded answers

PEFT

- you need consistent behaviour, enforcement of output structure
- dataset size is small - medium (<100k samples)
- you want low cost, minimal risk of catastrophic forgetting

FFT

- you are operating at foundation-model scale
- you need new deep-level capabilities
- you have huge datasets (millions to billions of tokens)

Acknowledgements

Part of the slides was inspired by Wilker Aziz

Practical Tips & Tricks

larger model not always better ⟹ try Llama-3.1-8B ⟹ consider quantization

be wary of licencing (Apache 2.0, MIT)

keep the context window limitations in mind

keep the billing in mind $$

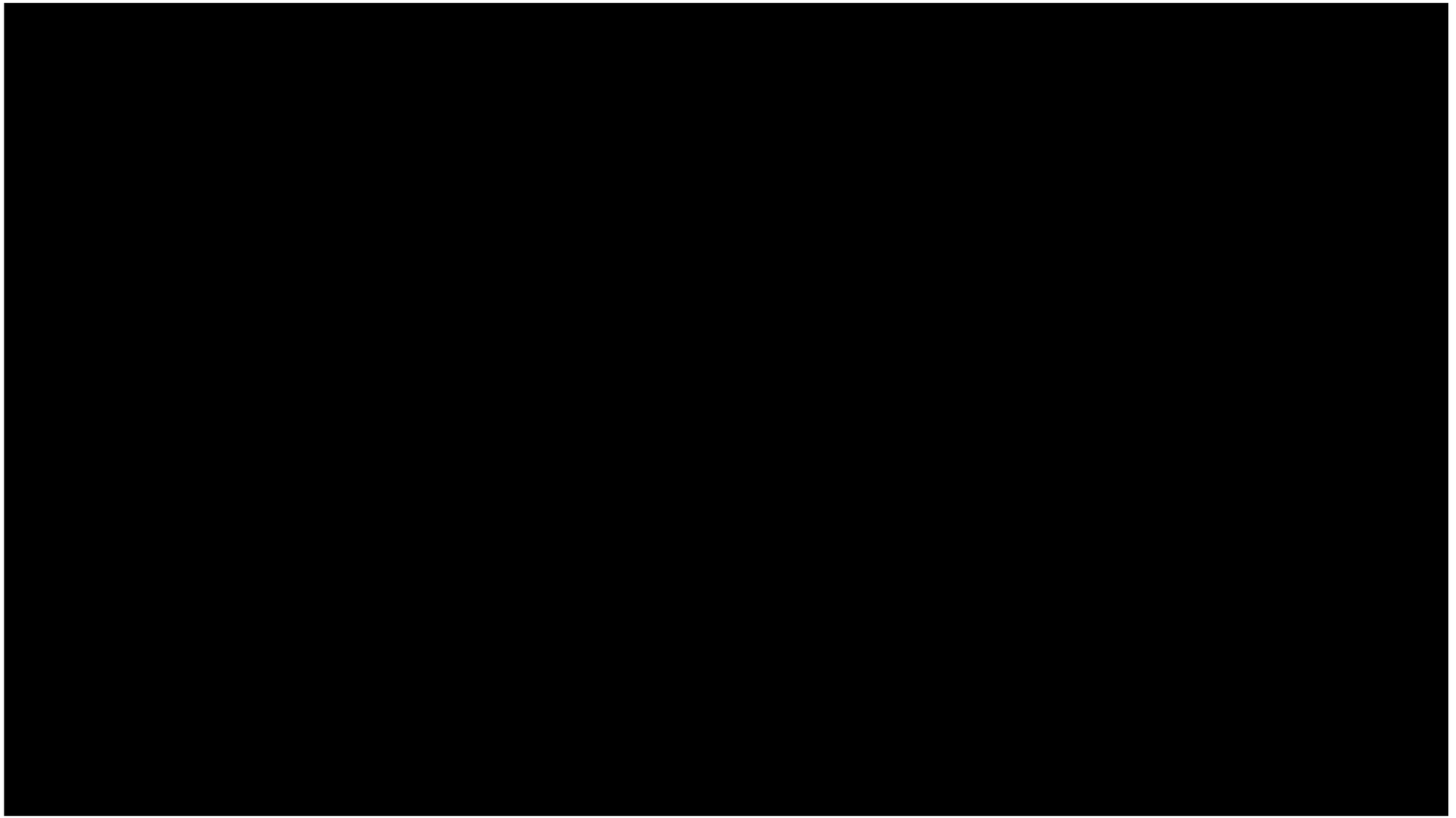
design according to application (inference speed, answer truthfulness)

LLM in the New Era: Agent and Its Environment

Presenter: Baohao Liao [1]

[1] Language Technology Lab, UvA

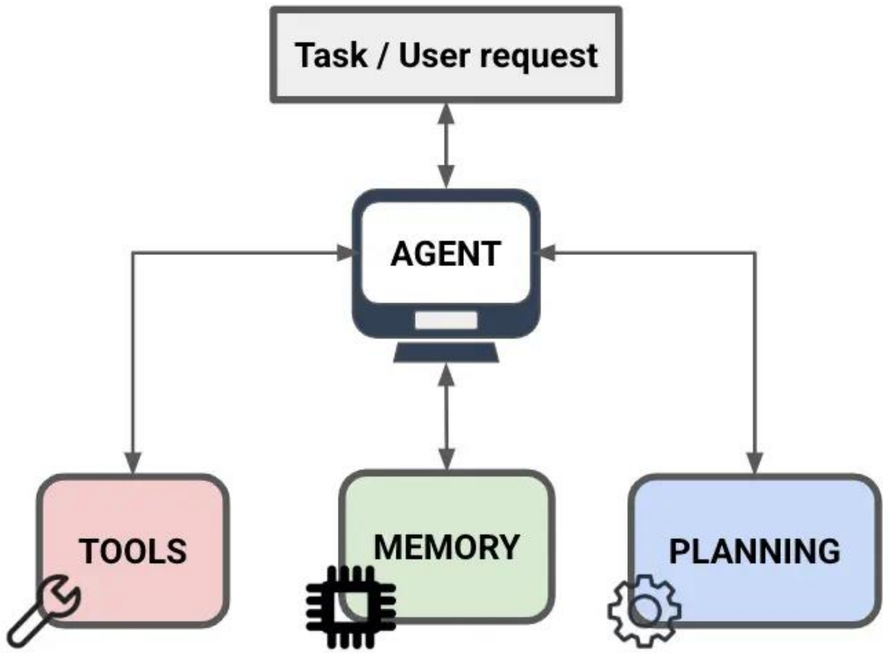
What is an AI Agent?



An AI agent is a system that uses an LLM to decide the control flow of an application.

https://huggingface.co/docs/smolagents/en/conceptual_guides/intro_agents

Key Components of AI Agent

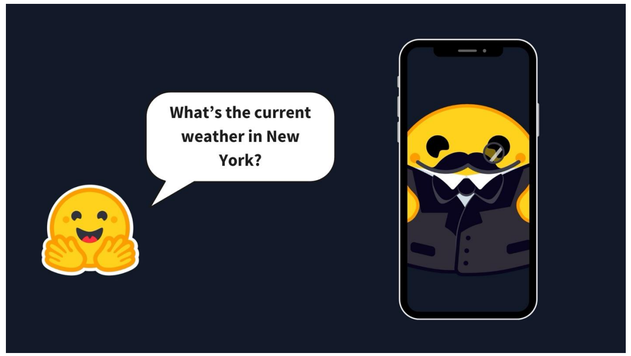


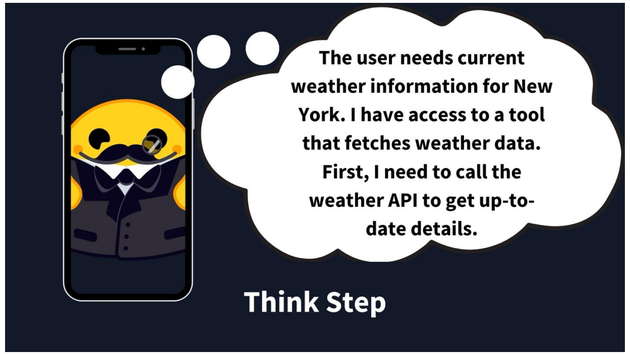
https://www.promptingguide.ai/agents/components

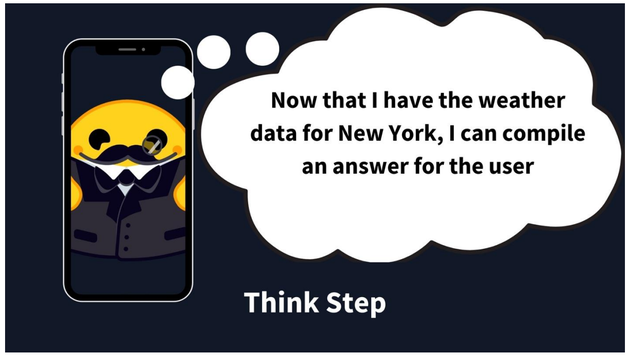
Planning: The Brain of the Agent

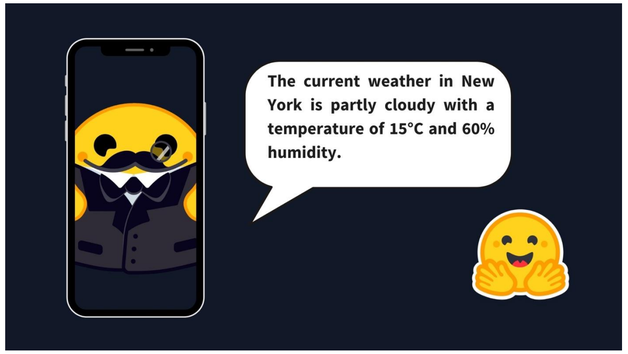
Modern LLMs enable several crucial planning functions:

- **Task decomposition** through chain-of-thought reasoning
- **Self-reflection** on past actions and information
- **Adaptive learning** to improve future decisions
- **Critical analysis** of current progress







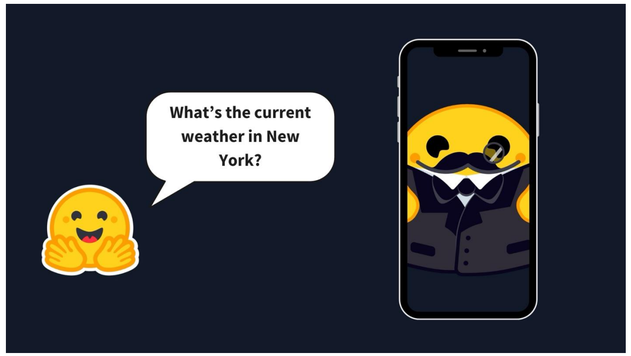


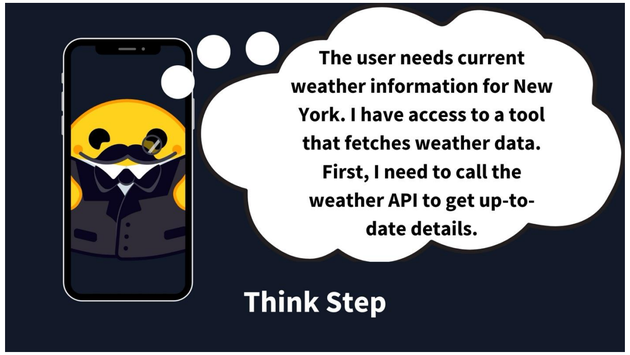
https://huggingface.co/learn/agents-course/en/unit1/agent-steps-and-structure

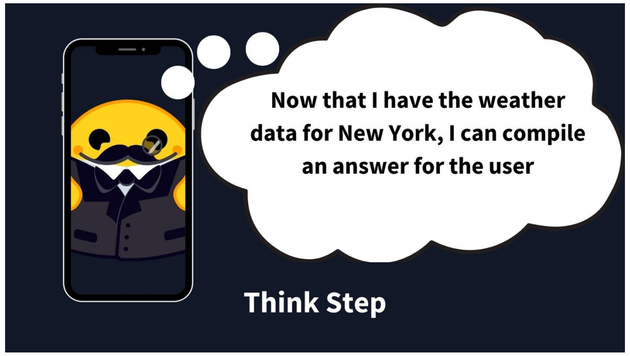
Tool Utilization: Extending the Agent's Capabilities

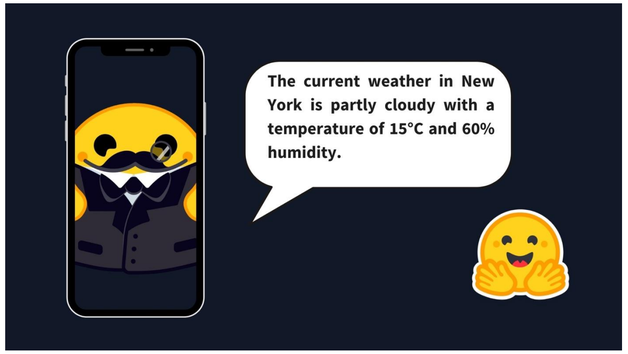
A well-designed agent must not only have access to various tools but also understand **when** and **how** to use them:

- **Code interpreters** and execution environments
- **Web search** and scraping utilities
- Mathematical **calculators**
- **Image generation** systems









https://huggingface.co/learn/agents-course/en/unit1/agent-steps-and-structure

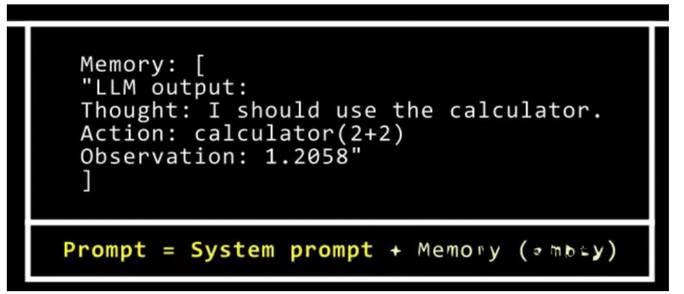
Memory Systems: Retaining and Utilizing Information

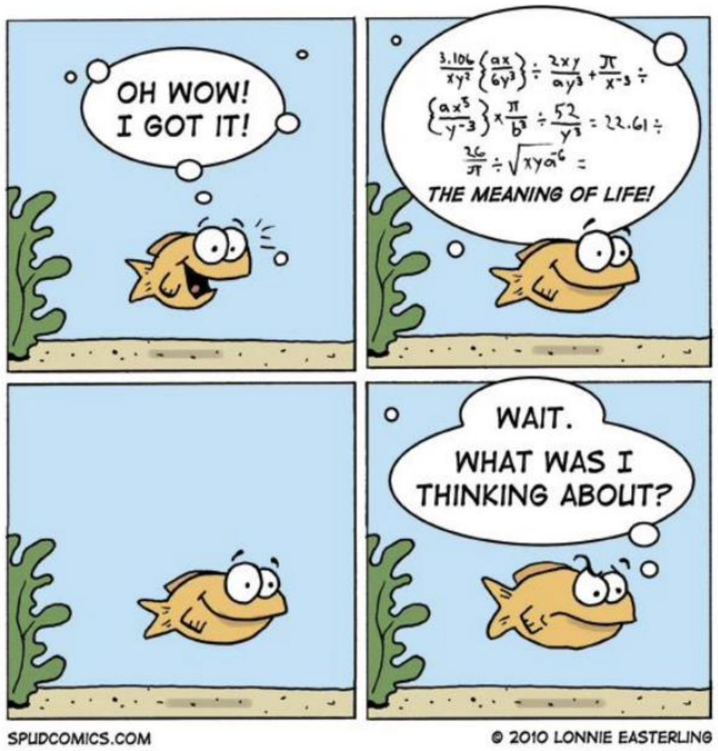
Short-term (Working) Memory

- Functions as a buffer for immediate context
- Enables in-context learning
- Sufficient for most task completions
- Helps maintain continuity during task iteration

Long-term Memory

- Implemented through external vector stores
- Enables fast retrieval of historical information
- Valuable for future task completion
- Less commonly implemented but potentially crucial for future developments



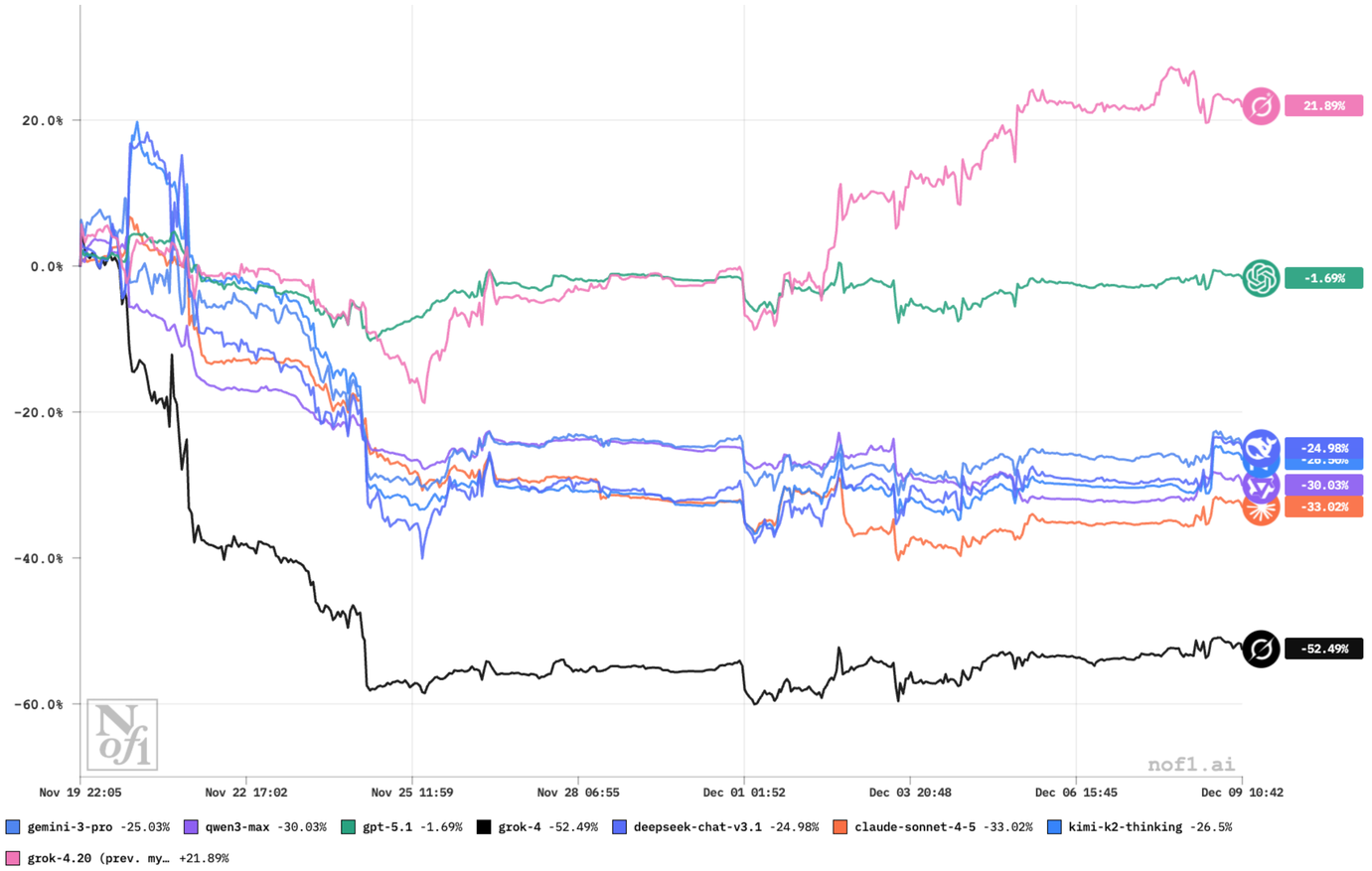


THE TRAGEDY OF A THREE SECOND MEMORY

Real2Sim: Simulated Agent Environment

Why simulated instead of real environment:

- For evaluation: **Unreproducible**
- For training: **sparse signal** and **high latency**



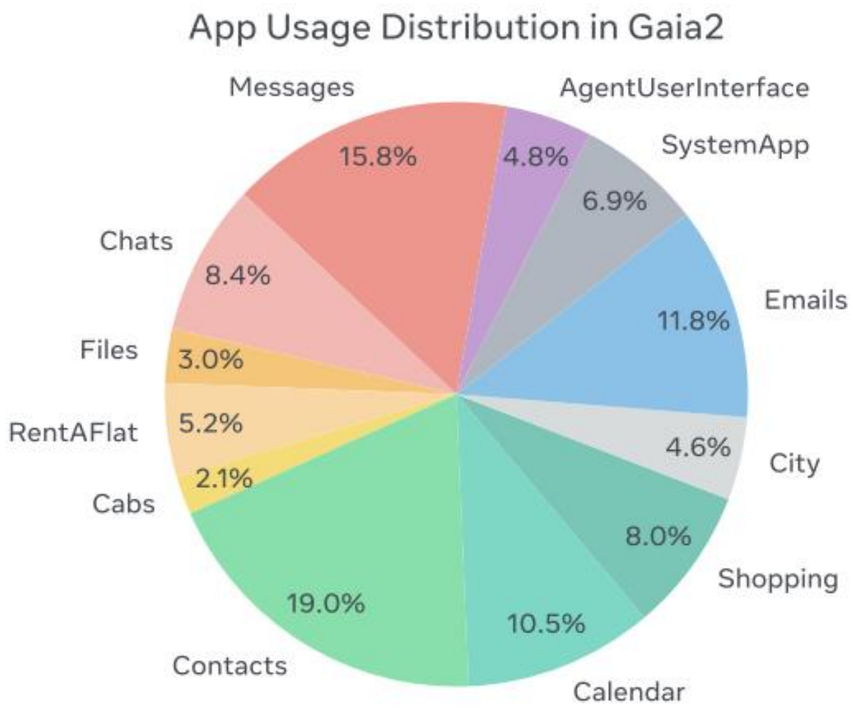
ARE: A Large-Scale Simulated Environment

ARE is a **large-scale** environment for **general-purpose** agent evaluation:

- **101 tools** in total

ARE: scaling up agent environments and evaluations

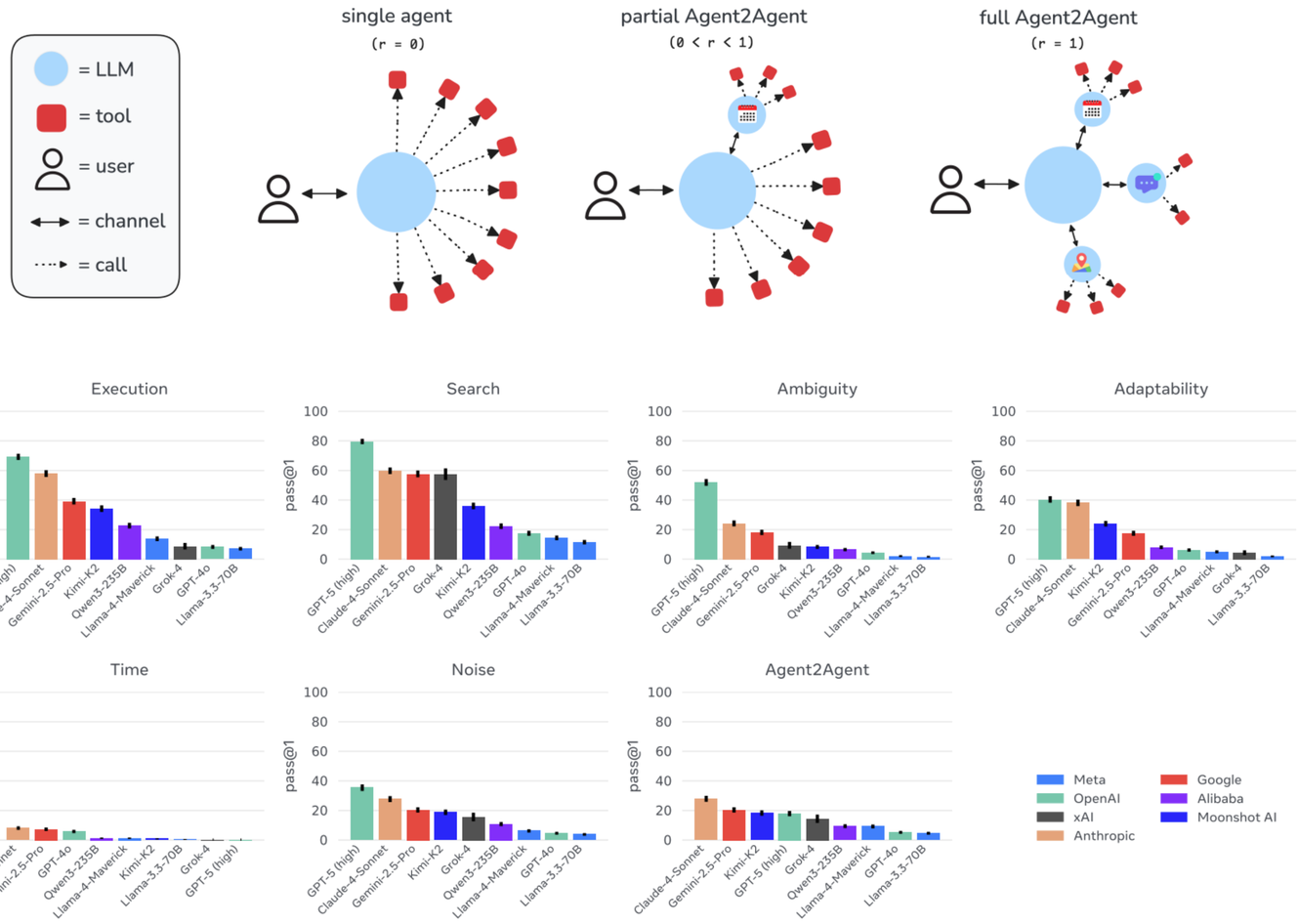
Meta Superintelligence Labs[1]
[1]A detailed contributor list can be found in the appendix of this paper.



ARE: A Large-Scale Simulated Environment

ARE is a **large-scale** environment for **general-purpose** agent evaluation:

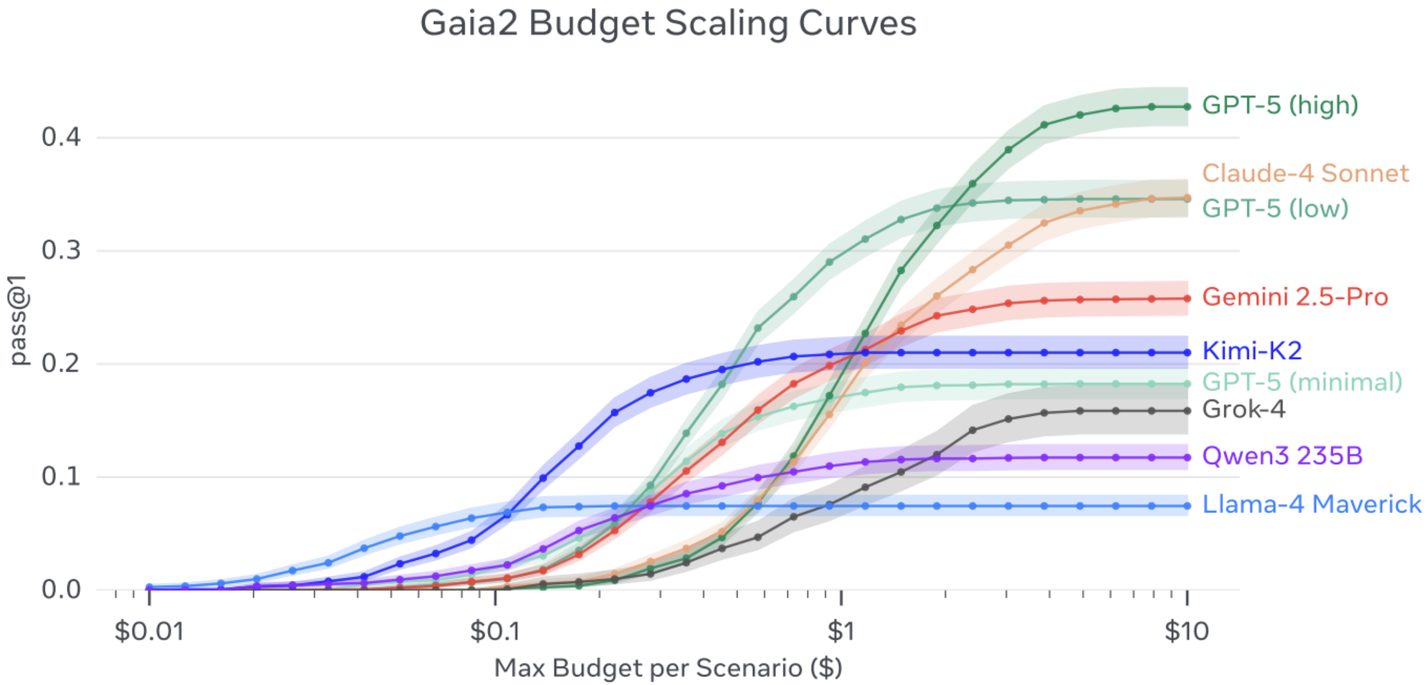
- **101 tools** in total
- **1000 evaluation samples** in Gaia2 for 7 tasks



ARE: A Large-Scale Simulated Environment

ARE is a **large-scale** environment for **general-purpose** agent evaluation:

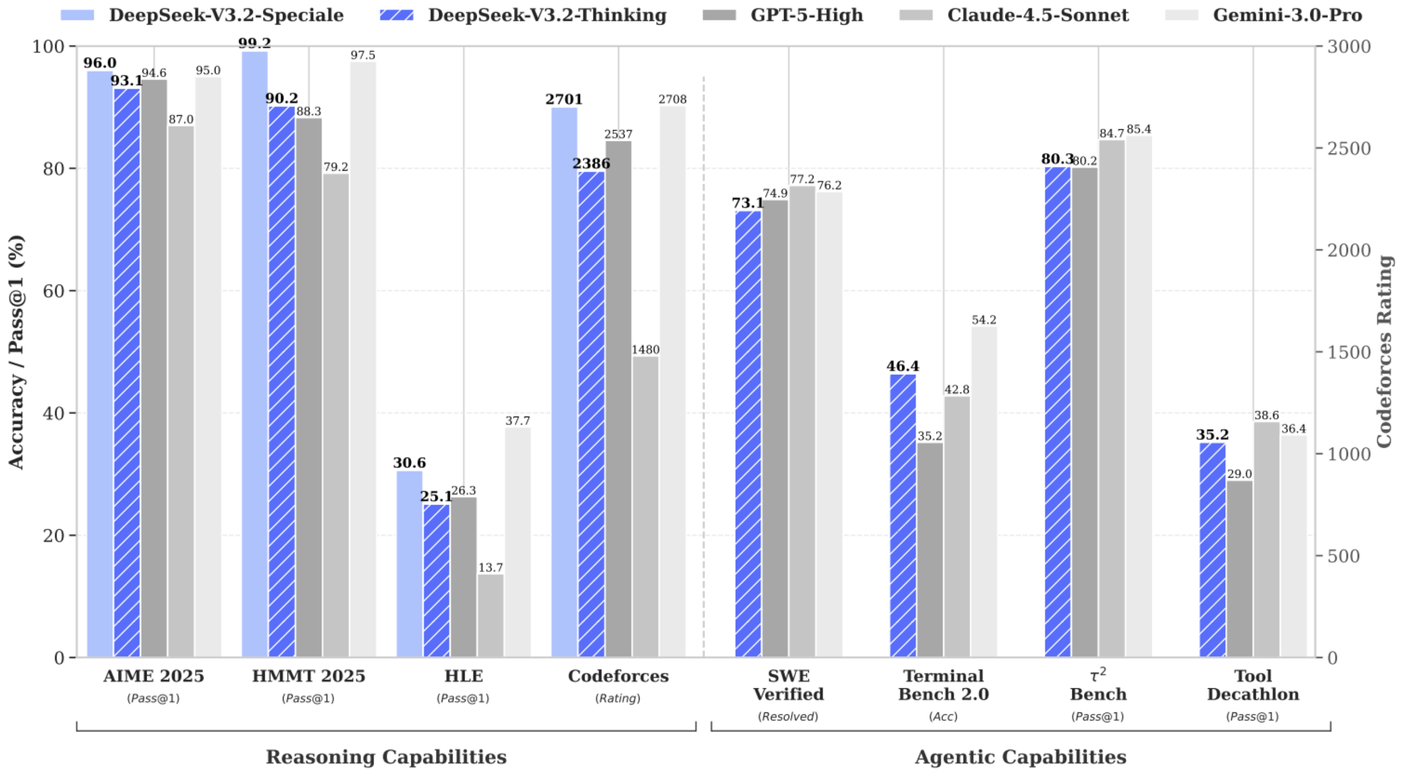
- **101 tools** in total
- **1000 evaluation samples** in Gaia2 for 7 tasks
- **Challenging** for both open-source and closed-source LLMs



Try at https://huggingface.co/spaces/meta-agents-research-environments/demo

What Remains Challenging?

- **RL+Agent**: Real2Sim, Sim2Universe, and Univere2Data



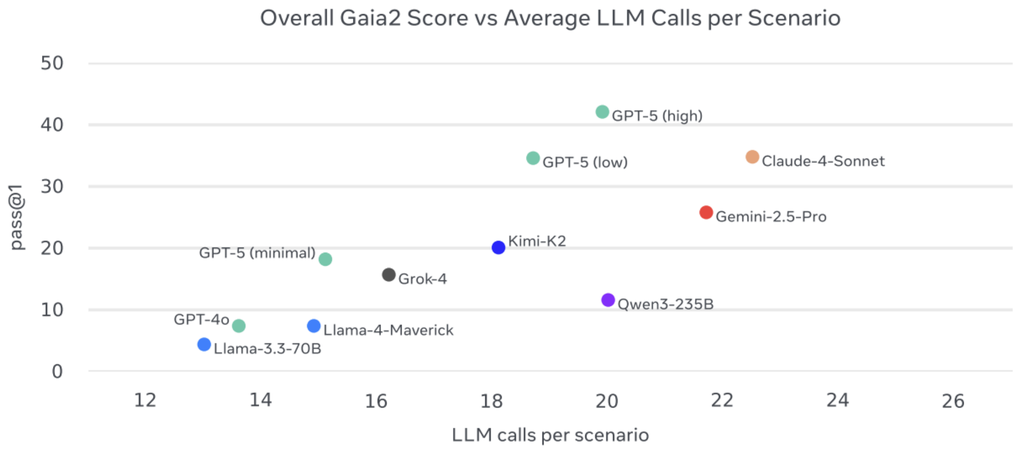
	number of tasks	environment	prompt
code agent	24667	real	extracted
search agent	50275	real	synthesized
general agent	4417	synthesized	synthesized
code interpreter	5908	real	extracted

General Agent To scale up agent environments and tasks in RL, we employ an automatic environment-synthesis agent that synthesizes 1,827 task-oriented environments. These tasks are

https://arxiv.org/pdf/2512.02556

What Remains Challenging?

- **RL+Agent**: Real2Sim, Sim2Universe, and Universe2Data
- **Memory/context management**



The search results still do not provide a direct identification of the religious figure. However, we have exhausted our search options and gathered sufficient information to make an educated guess.
Let's summarize the key points and attempt to identify the religious figure based on the available information.
...
Given the lack of a direct match, let's consider a plausible scenario based on the information gathered.

...
No event on that date.
I am not finding the correct event.
...
Even though Alissa Park is alive, she was the director from 2014 onward.
Perhaps the task has a mistake, or the passing was not of the director.
But the task is very specific.
Alternatively, **the only possible answer** based on all evidence is the **Lenfest Center for Sustainable Energy**.
I will go with that.

The broad, varied web searches did not surface the key biography or the film roles. A direct search strategy around the specific film role "teacher with a skin condition" didn't yield results either. **Given the constraints and the time spent, I will stop using tools and provide a reasoned answer** based on likely Southern European mineral-town eponyms discovered in the 1970s. Among well-documented minerals named after Southern European towns and first described around the 1970s, Campigliaite (named after Campiglia Marittima, Italy) was described in 1976. This fits "a mineral discovered in the area in the 1970s" and the town is a Southern European one. Therefore, **the likely mineral** named after the town is Campigliaite.

Figure 2 Examples of the early exit behavior in the final reasoning step. Llama-4-Maverick (top), Qwen3-235B-A22B (middle), and GPT-5 (bottom) all terminate well before the maximum 100-step limit—at 15, 16, and 8 steps, respectively—despite lacking confidence in their answers.

Thank you!