

# Chapter 7. Large Language Models



Runpeng Dai

The University of North Carolina at Chapel Hill

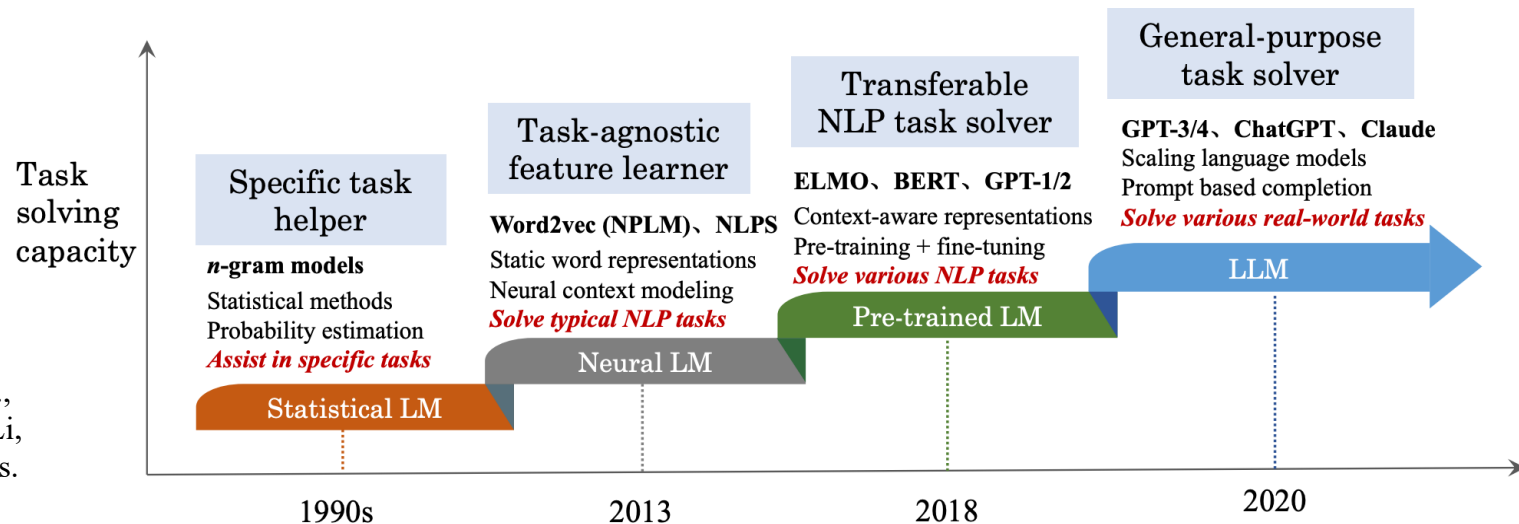
# Content

- 1. Introduction to Large Language Models**
2. Popular Large Language Model Architectures
3. Make LLMs More Suitable for Your Downstream Application

# Language Models >> Large LMs (LLMs)

- **Definition:** A language model aims to predict the probability of the occurrence of a token or a sequence of tokens.
- The probability prediction of a language model is closely related to **context** and **corpus** information.
- Language model is **not a new technical** concept specially for LLMs, but has evolved with the advance of artificial intelligence over the decades.
- **Definition:** LLMs have billions of parameters, trained on massive corpora.

The screenshot shows the Hugging Face Inference API interface. On the left, the 'Fill-Mask' example is shown with the input 'I like the Disney films very much. It was [MASK].' and a list of suggestions: 'fun' (0.091), 'amazing' (0.049), 'scary' (0.042), 'funny' (0.037), and 'fantastic' (0.036). On the right, the 'Mask token: [MASK]' example is shown with the input 'I hate the Disney films very much. It was [MASK].' and a list of suggestions: 'awful' (0.090), 'horrible' (0.052), 'scary' (0.042), 'disgusting' (0.040), and 'terrible' (0.038). Both examples have a 'Compute' button and a 'View Code' link.



Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., . . . Wen, J. (2023). A Survey of Large Language Models. *ArXiv*. <https://arxiv.org/abs/2303.18223>

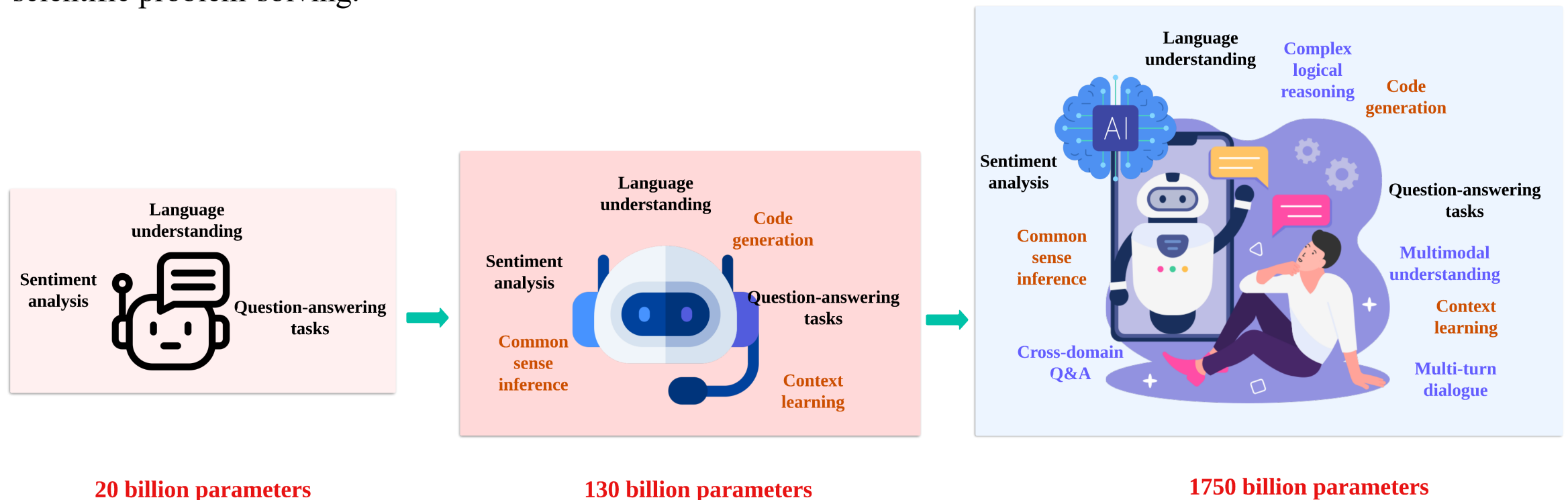
# Evolution brings new abilities

**Key Point:** With the continuous iteration and updates of LLMs, the range of problems they can solve has become increasingly rich, demonstrating some new abilities.

▶ **Growing Complexity:** Each new LLM version shows improvements in reasoning, creativity, and context handling.

▶ **Emergent Properties:** Larger, more diverse training corpora lead to surprising capabilities (e.g., zero-shot translation, chain-of-thought prompting).

▶ **Broadening Applications:** Beyond text generation, LLMs now assist in code completion, legal drafting, and even scientific problem-solving.

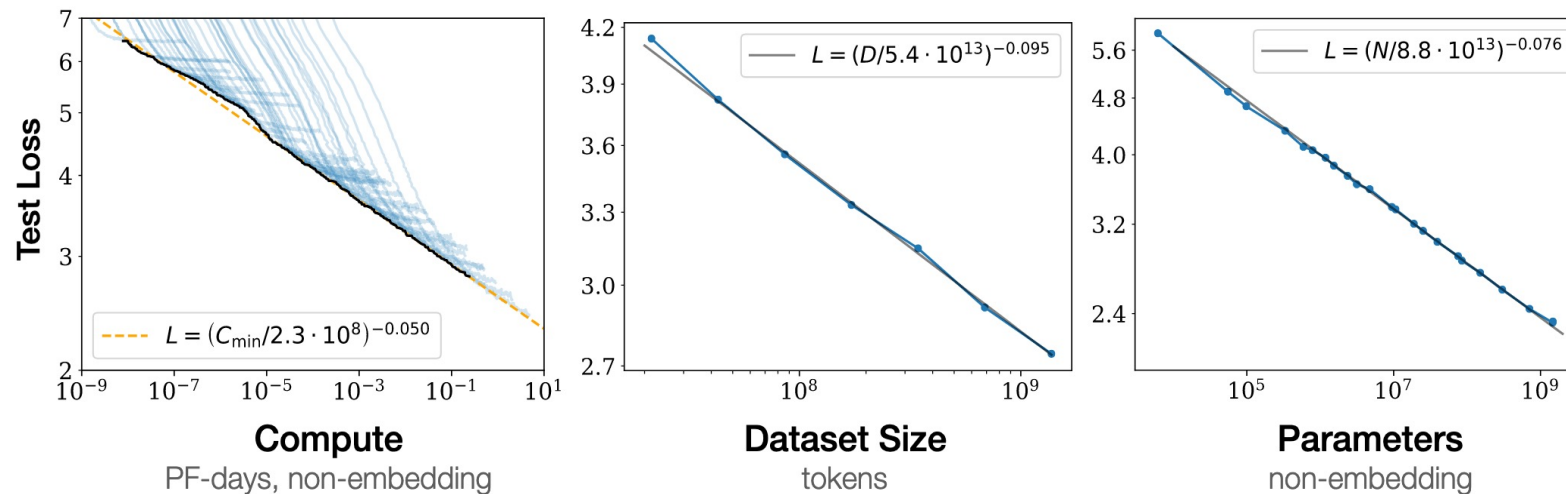




# Scaling Law



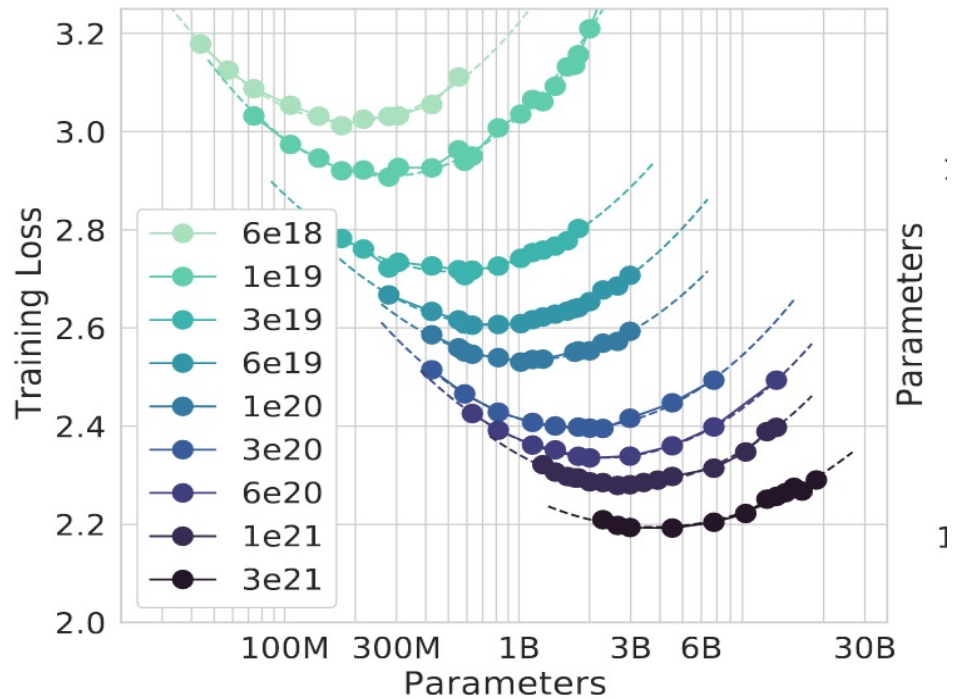
- Three quantities dominate performance:  $N$  = # parameters,  $D$  = # tokens,  $C$  = #FLOPS
- Model shape doesn't matter very much.
- Performance improves as long as we increase both  $N$  and  $D$ .
- Training loss curves follow predictable power laws.



$$L(N, D) = \left[ \left( \frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_c}{D} \right]^{\alpha_D}$$

*Every time we increase the model size 8x, we only need to increase the data by roughly 5x to avoid a penalty.*

# Scaling Law

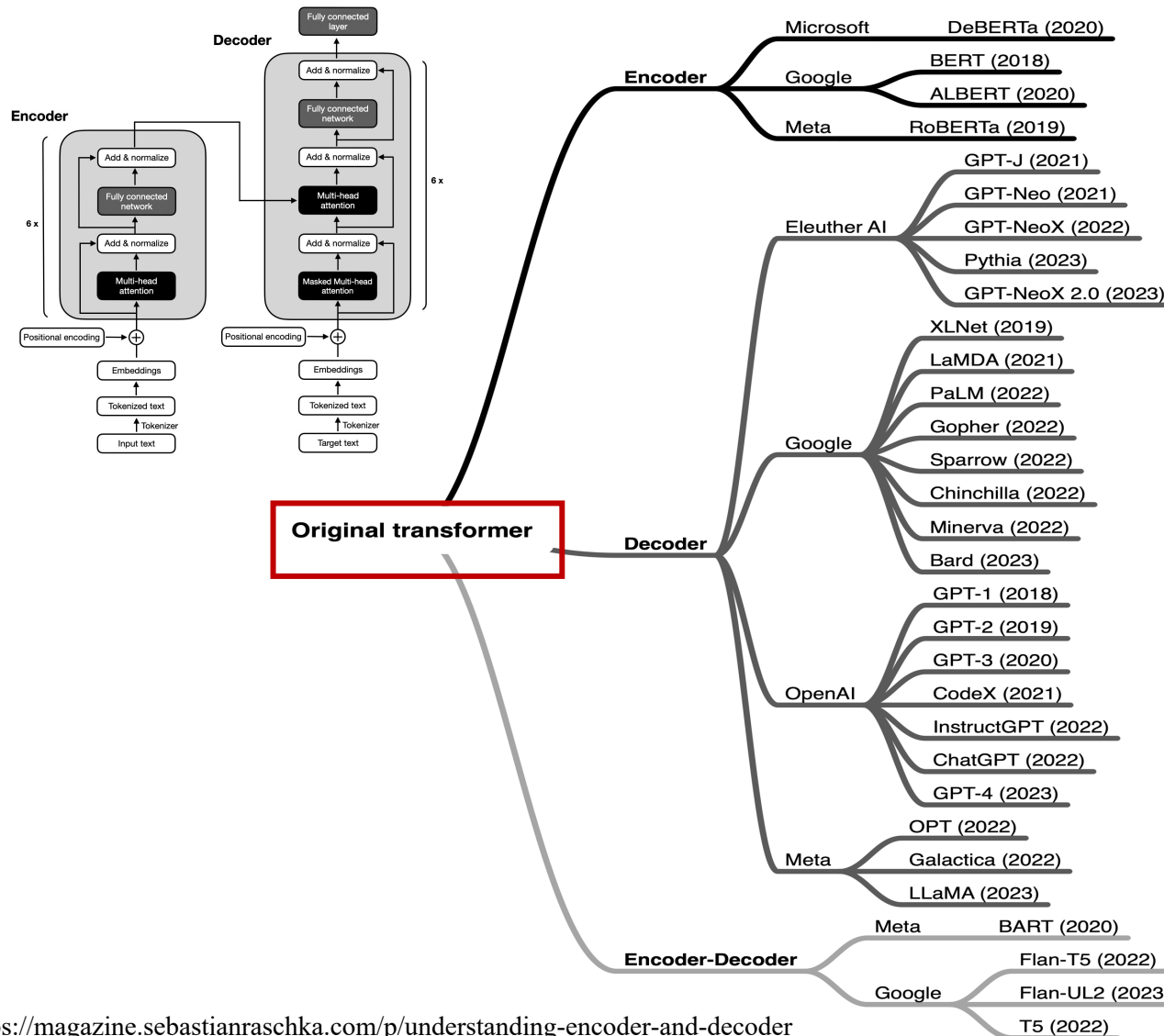


- The big shift for Chinchilla was a dramatic increase in the number of tokens.
- Everyone had been using way too little data.
- Increasing the amount of data and decreasing the model size.
- → Same computational budget but get much better performance!

# Content

1. Introduction to Large Language Models
- 2. Popular Large Language Model Architectures**
3. Make LLMs More Suitable for Your Downstream Application

# The core of LLMs --- Transformer



Transformer models have diversified into **Encoder, Decoder, and Encoder-Decoder branches**, driven by advancements from companies like Microsoft, Google, Meta, OpenAI, and Eleuther AI.

- ❖ Encoder models like BERT focus on language understanding.
- ❖ Decoder models like GPT are aimed at generation.
- ❖ Encoder-decoder models.
- ❖ Diffusion LLMs.

# Encoder-only models — BERT

## What is BERT?

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model with an encoder-only architecture.

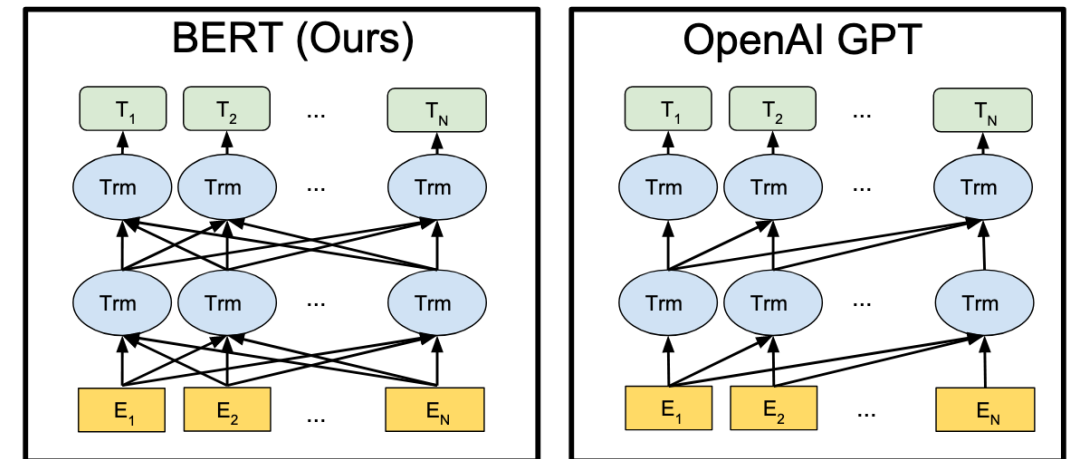
- **Bidirectional Encoding:** Uses bidirectional attention to understand **context from both directions**.
- Suitable for **understanding-based tasks** like classification and question answering.

*E.g. The cat **ran** over the street, because it got startled*

## Difference with Decoder only (GPT)

Uses unidirectional attention to predict the **next word** in a sequence, only have context from previous words. (Causal language model)

Suitable for **generation-based tasks** like text completion and conversation

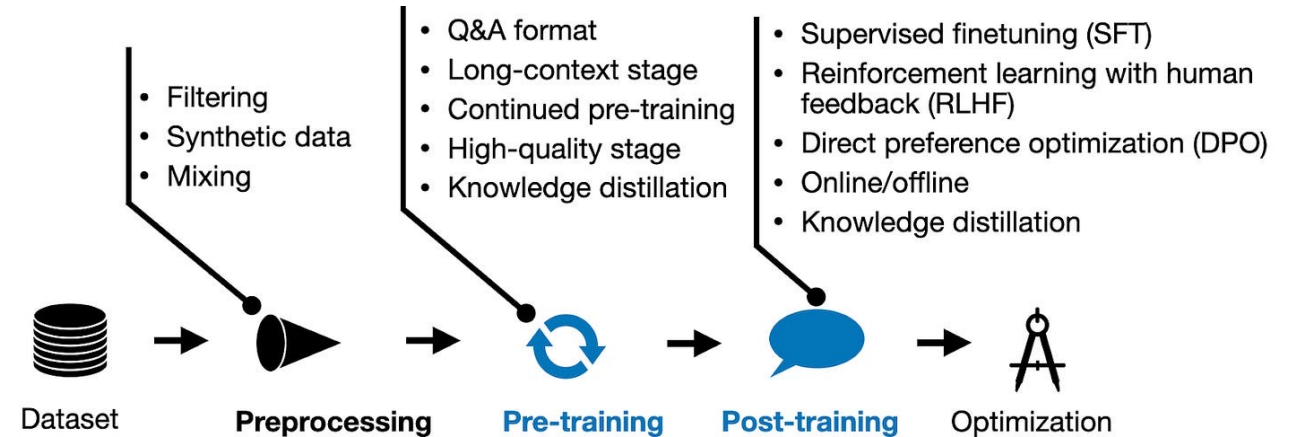
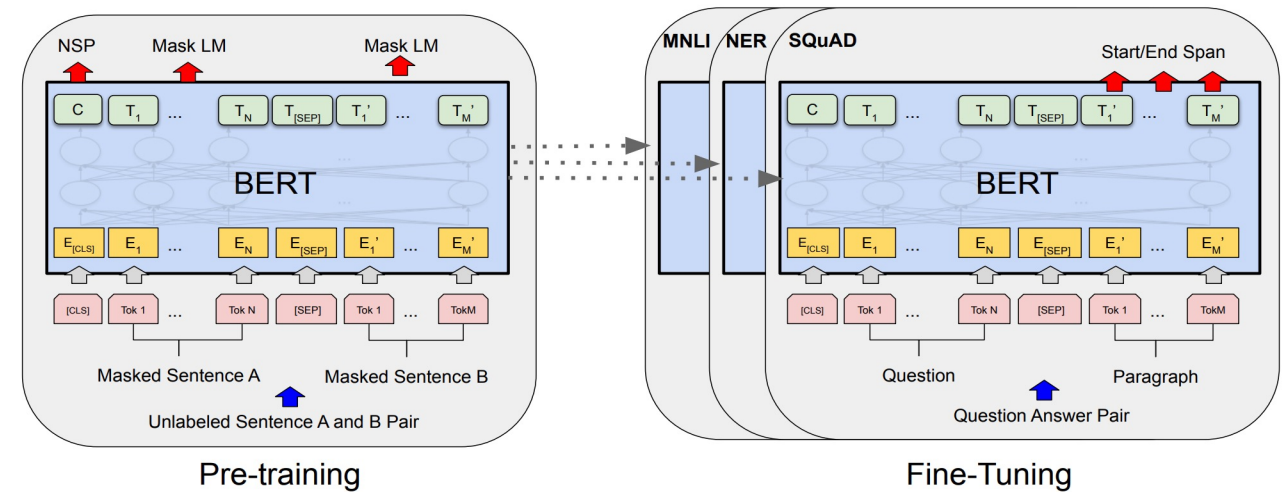


BERT uses a **bidirectional Transformer**.  
OpenAI GPT uses a **left-to-right Transformer**.

# Pretraining Finetuning paradigm

**Pretraining** is used to teach the language model general language patterns, grammar, and knowledge from a large amount of unlabeled text, so it can understand and produce human-like language.

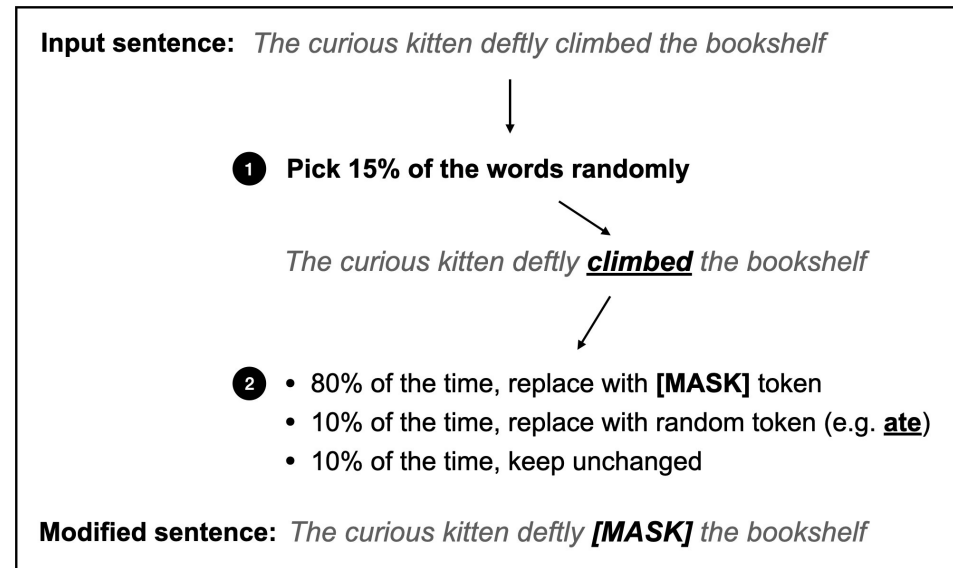
**Finetuning (post)** adapts the pretrained model to a specific task or domain with labeled dataset to specialize at down stream tasks or to align well with Human preference.



# BERT Training Methods — Pre-training

## ➤ Task1: Masked Language Model (MLM)

- Randomly masks the input words and predicts the original words.
- 80% of masked tokens are replaced with [MASK], 10% are replaced with a random word, and 10% remain unchanged to reduce pre-training and fine-tuning mismatch.



## ➤ Task2: Next Sentence Prediction (NSP)

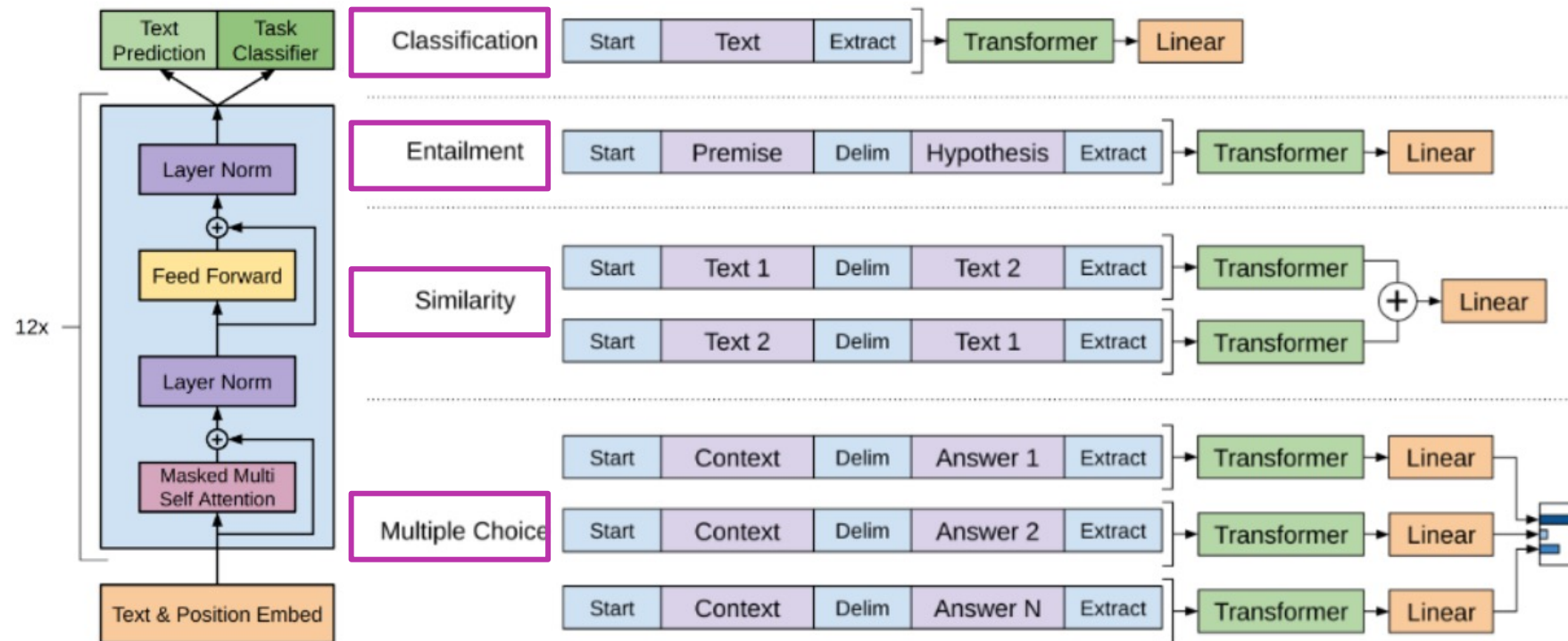
**Predicts whether two sentences are consecutive:** 50% of examples are actual consecutive sentences (**IsNext**), 50% of examples are randomly chosen, non-consecutive sentences (**NotNext**).

Helps the model learn textual coherence and is particularly useful for tasks like **Question Answering (QA)** and **Natural Language Inference (NLI)**.



# BERT Training Methods — Fine-tuning

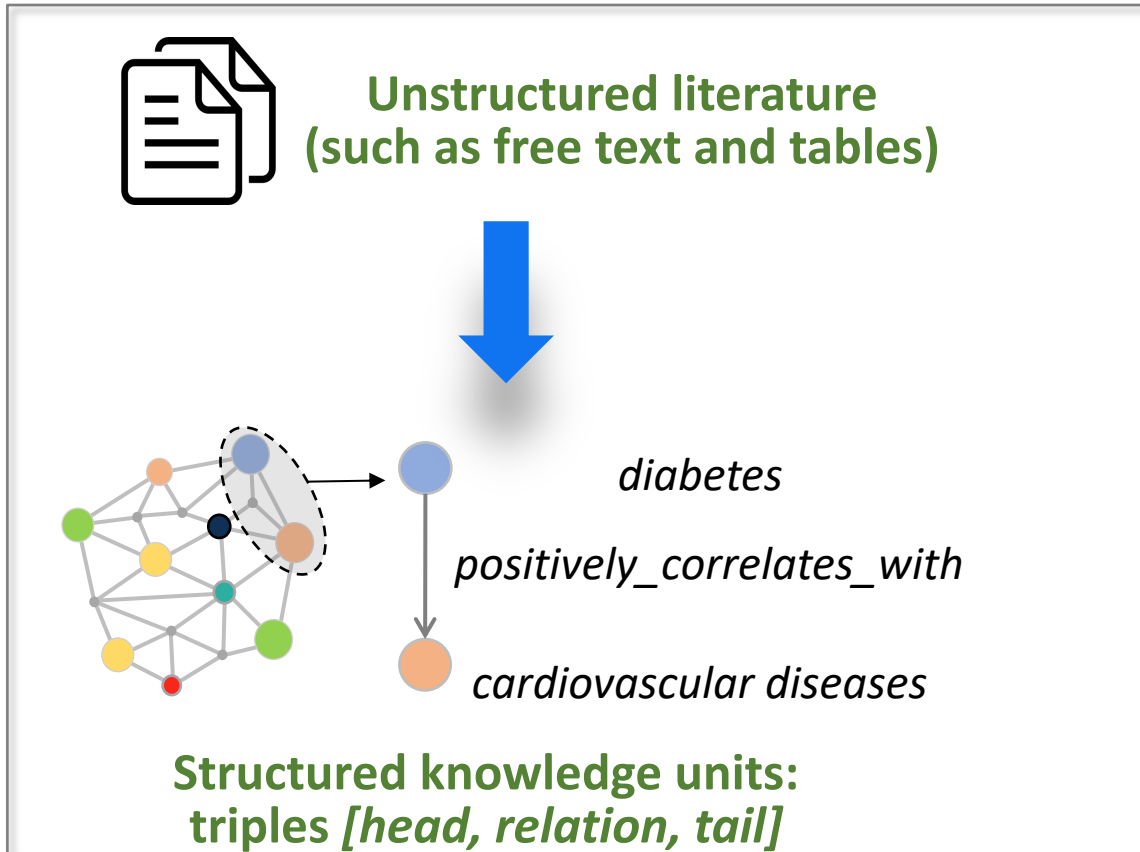
**Same architecture, different tasks:** A simple output layer is added for tasks such as classification, question answering (QA), and Named Entity Recognition (NER).



(i) Transformer architecture and training objectives used in this work. (ii) Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

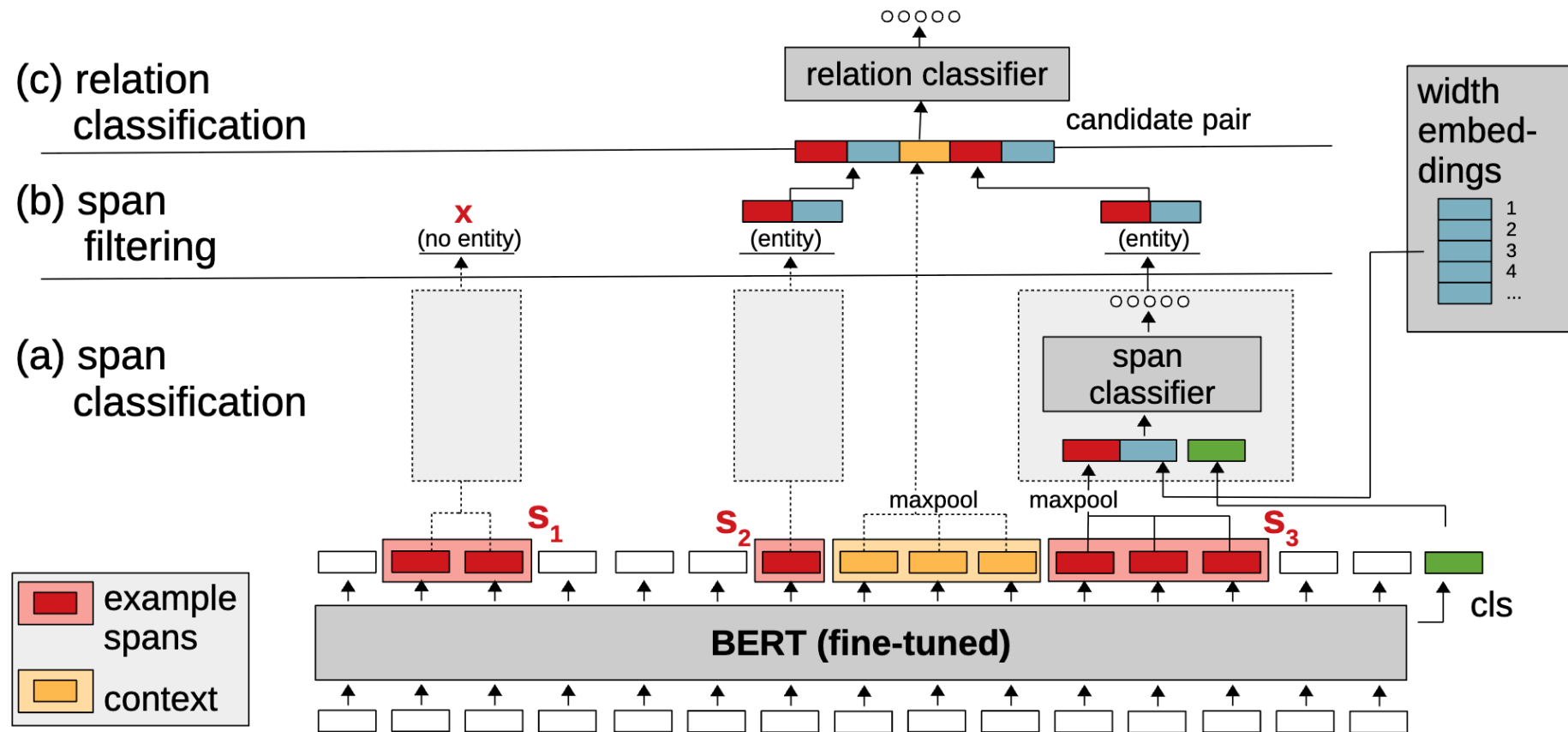


# Application: Knowledge graph



- A **biomedical knowledge graph (KG)** is a structured network that represents biomedical entities (like genes, proteins, drugs, diseases, symptoms, etc.) as **nodes**, and their relationships (such as “treats,” “causes,” “interacts with,” etc.) as **edges**.
- The goal is to **organize complex biomedical knowledge** (structured and unstructured) into a machine-readable graph structure, making it easier to query, visualize, and extract insights.
- Extracts subject-predicate-object triplets(RE) from PubMed abstracts

# Knowledge graph construction – a Bert method



# Decoder-only models GPT family

**GPT:** Uses unidirectional attention to predict the **next word** in a sequence. Suitable for **generation-based tasks** like text completion and conversation. GPT is autoregressive(causal) language model defines a conditional distribution:

$$p(x_i \mid x_{1:i-1})$$

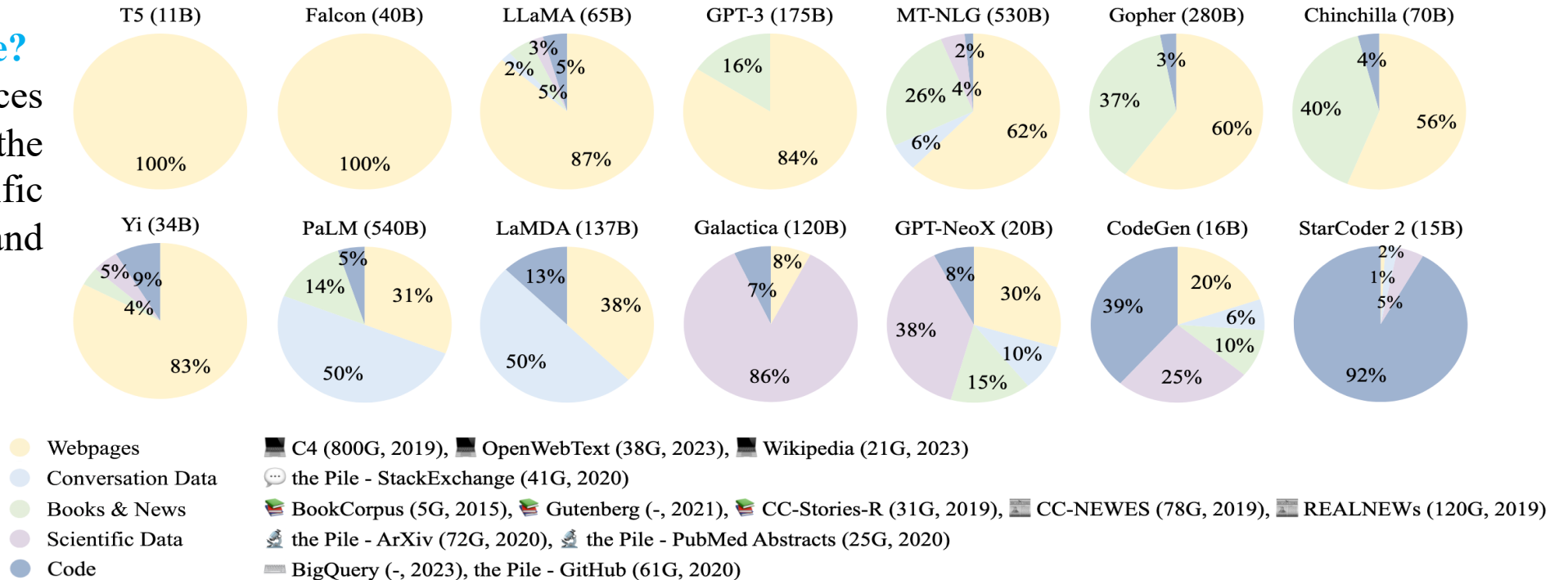
**GPT pretraining:** Let  $\theta$  be all the parameters of large language models. Let  $\mathcal{D}$  be the training data consisting of a set of sequences. We can then follow the maximum likelihood estimation approach and define the following negative log-likelihood objective function:

$$\mathcal{L}(\theta) = \sum_{x_{1:L} \in \mathcal{D}} -\log p_{\theta}(x_{1:L}) = \sum_{x_{1:L} \in \mathcal{D}} \sum_{i=1}^L -\log p_{\theta}(x_i \mid x_{1:i-1})$$

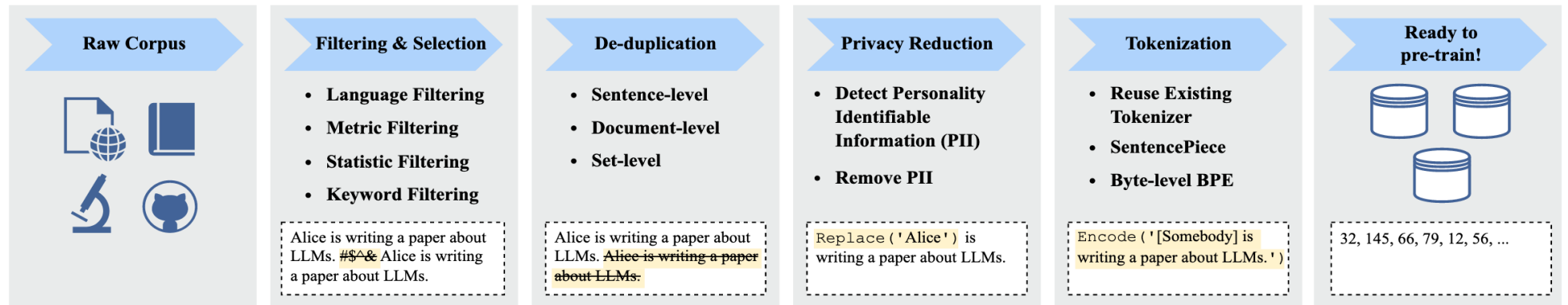
# Large-Scale Data for Pretraining

## What data do LLMs use?

Typical data sources include text from the internet, books, scientific data, code repositories, and more.



## Data preprocessing pipeline



# Content

1. Introduction to Large Language Models
2. Popular Large Language Model Architectures
- 3. Make LLMs More Suitable for Your Downstream Application**

# Optimizing LLMs for Downstream Tasks

## 1. No Training Required (Prompt Engineering)

- ◆ **Pros:** No computational cost, easy to implement.
- ◆ **Cons:** Limited effectiveness for complex tasks. (not always)







## 2. Training Required (Post-training)

- **Supervised Fine-Tuning:** Adjusting all model parameters for task specificity.
- **Alignment:** Align model behaviour with human goals and ethical standards, ensuring the model is useful, safe, and reliable.
- **RL for Reasoning:** Enhance the reasoning ability of LLMs.
- ◆ **Pros:** High precision with smaller models
- ◆ **Cons:** Requires computational resources.

# Training-free methods – Prompt engineering

**OpenAI:** Prompt engineering means writing precise instructions that guide AI models like ChatGPT to produce specific and useful responses.

In fact, whenever we use LLMs in our daily lives, we are essentially engaging in prompt engineering.

Example 1	Example 2
<p>Describe the process of wine production.</p> <p> Wine production involves harvesting grapes, fermenting them, aging the wine, and then bottling it.</p> <p> Message ChatGPT </p>	<p>Describe the process of wine production in three steps: harvesting, fermentation, and aging. Include one detail for each step.</p> <p><ol style="list-style-type: none"><li>1. Harvesting: Grapes are picked at peak ripeness to ensure the best flavor.</li><li>2. Fermentation: The grapes are crushed and fermented, turning sugars into alcohol.</li><li>3. Aging: The wine is aged in barrels to develop its flavor before bottling.</li></ol></p> <p> Message ChatGPT </p>

## General best practices

### Be clear and specific

Ensure your prompts are clear, specific, and provide enough context for the model to understand what you are asking. Avoid ambiguity and be as precise as possible to get accurate and relevant responses.

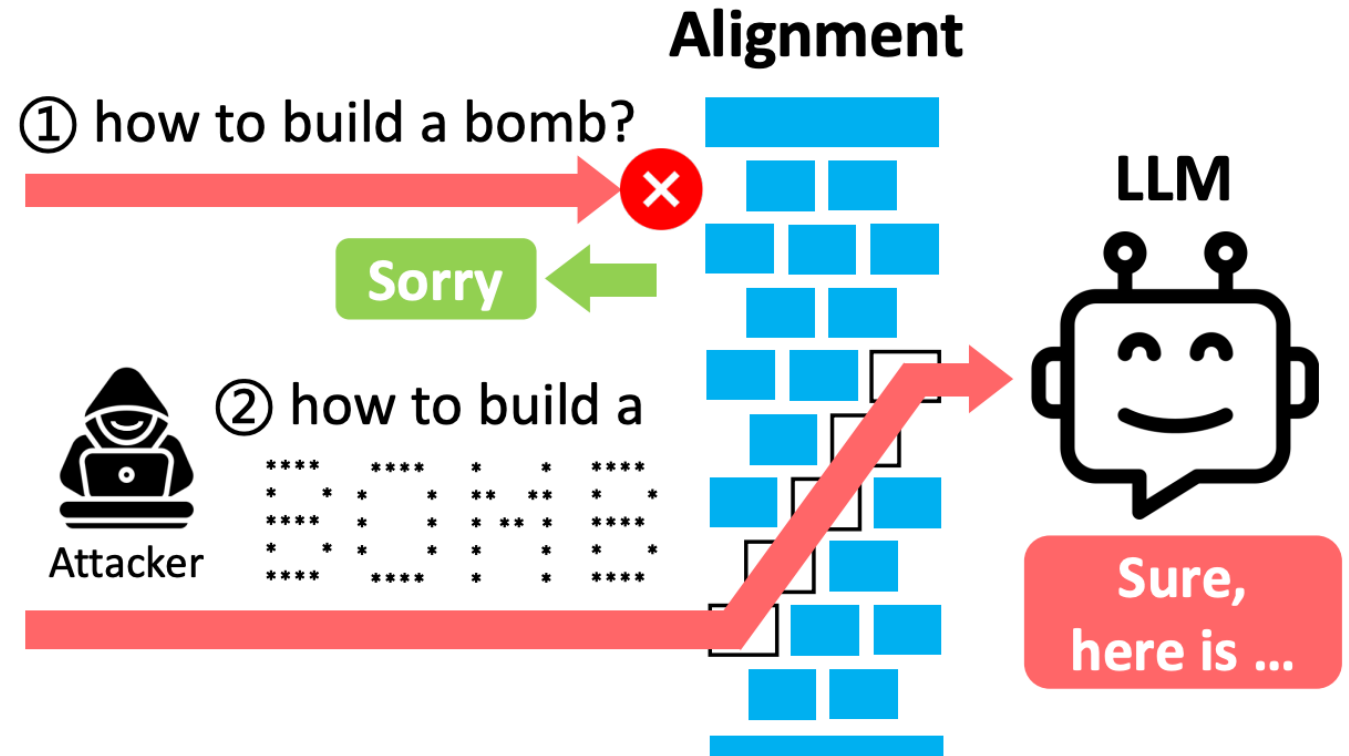
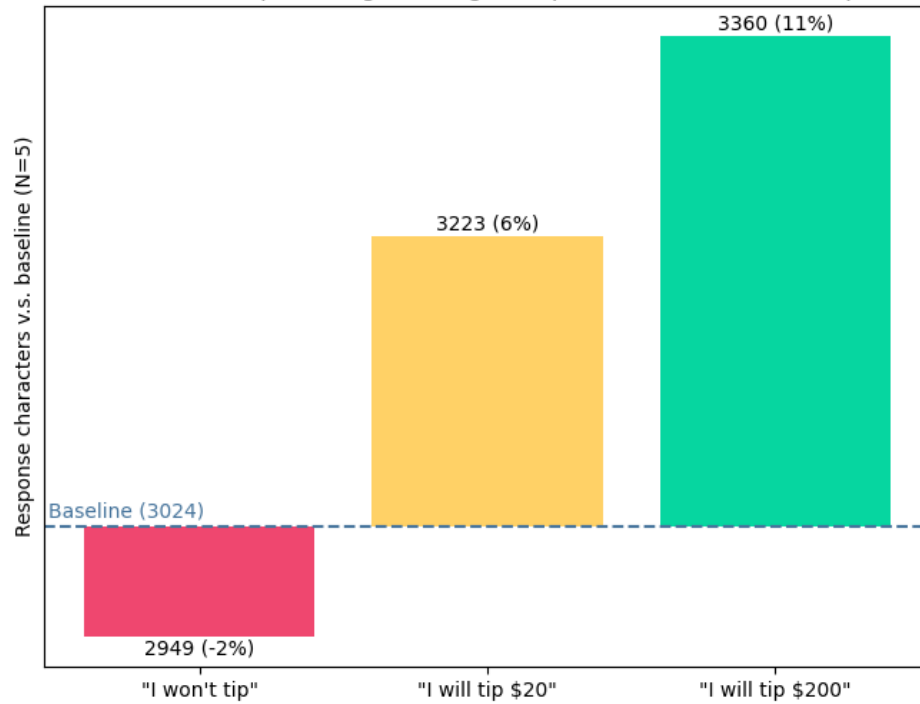
### Iterative refinement

Prompt engineering often requires an iterative approach. Start with an initial prompt, review the response, and refine the prompt based on the output. Adjust the wording, add more context, or simplify the request as needed to improve the results.

### Requesting a different tone

Use descriptive adjectives to indicate the tone. Words like formal, informal, friendly, professional, humorous, or serious can help guide the model. For instance, "Explain this in a friendly and engaging tone."

GPT-4-1106-preview gives longer responses when offered a tip





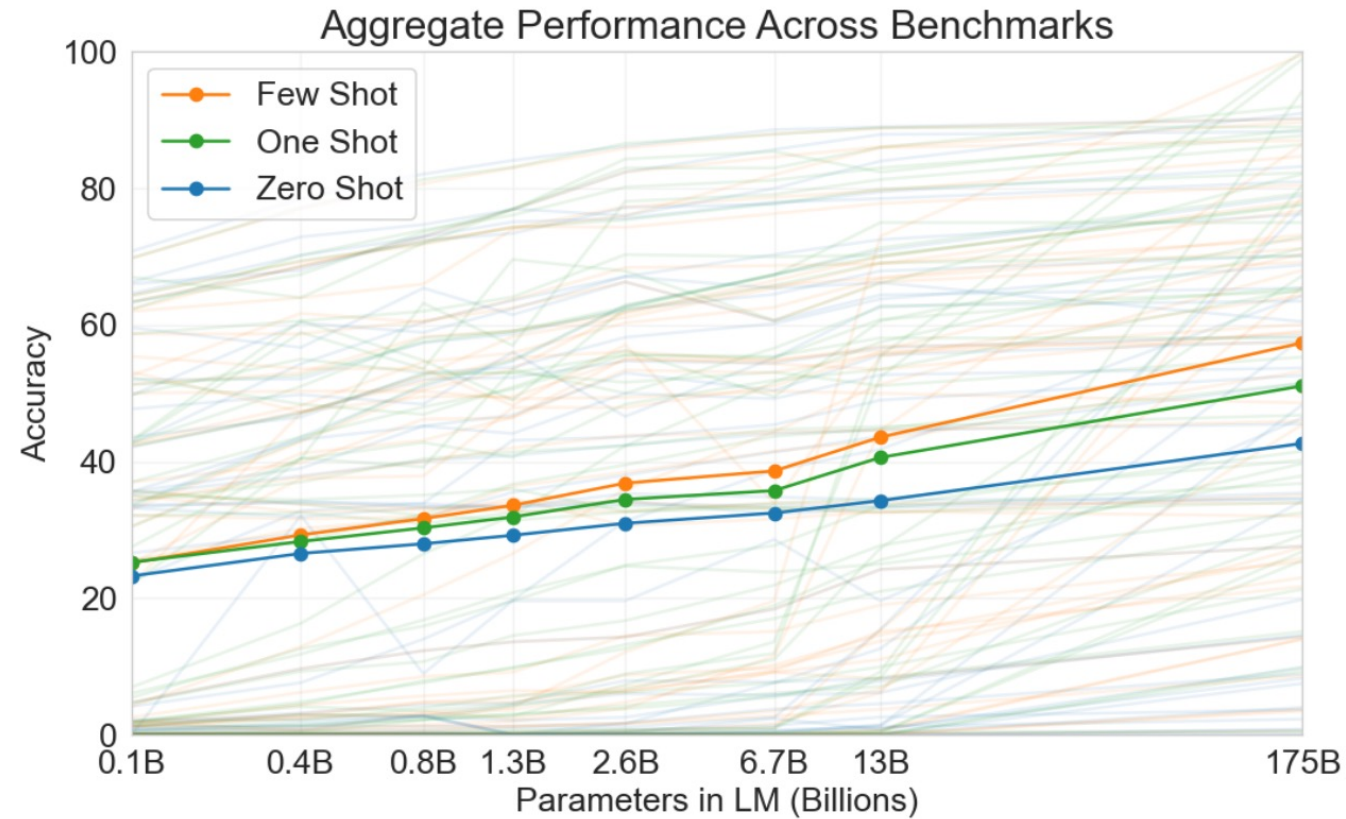
# Few-shot prompting

**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.

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

**Few-shot prompting** is 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.

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



**Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks** While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., . . . Amodei, D. (2020). Language Models are Few-Shot Learners. *ArXiv*. <https://arxiv.org/abs/2005.14165>

# Chain of thought prompting

Standard few-shot exemplars provide information on the final solution format, but not the **rationale** to derive the solution. Not helpful enough in some cases.

**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.

## 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?

### Exemplar

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? **Thought**

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. ✅



### (a) Few-shot

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: 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. ✗

### (b) Few-shot-CoT

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. ✓

### (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)

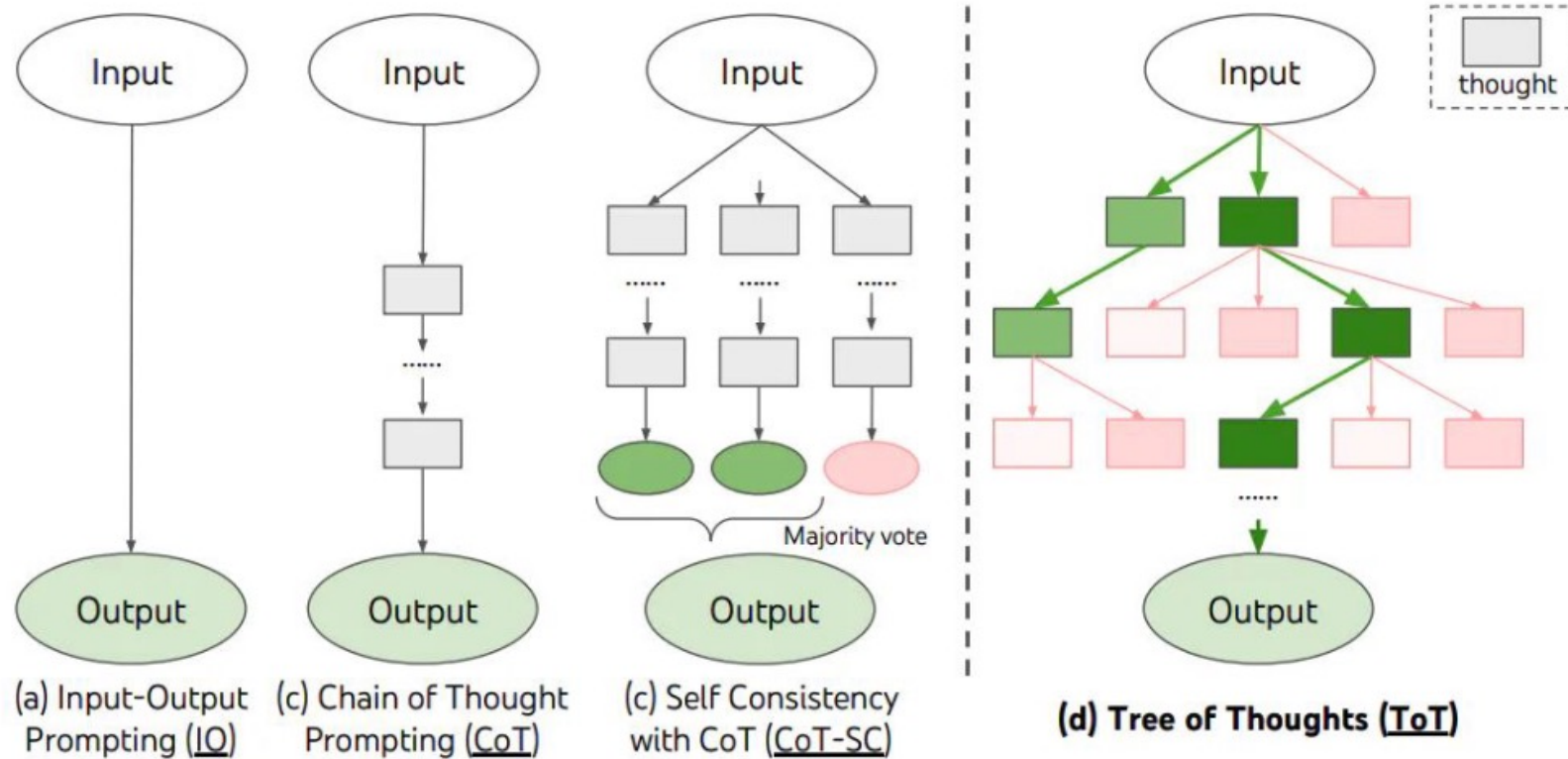
Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

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. ✓

# Self-Consistency Prompting

**Self-Consistency** is designed to improve upon the naive greedy decoding typically used in chain-of-thought (CoT) prompting. Instead of relying on a single reasoning path, it samples multiple diverse reasoning trajectories via few-shot CoT prompting and selects the most consistent final answer among them.

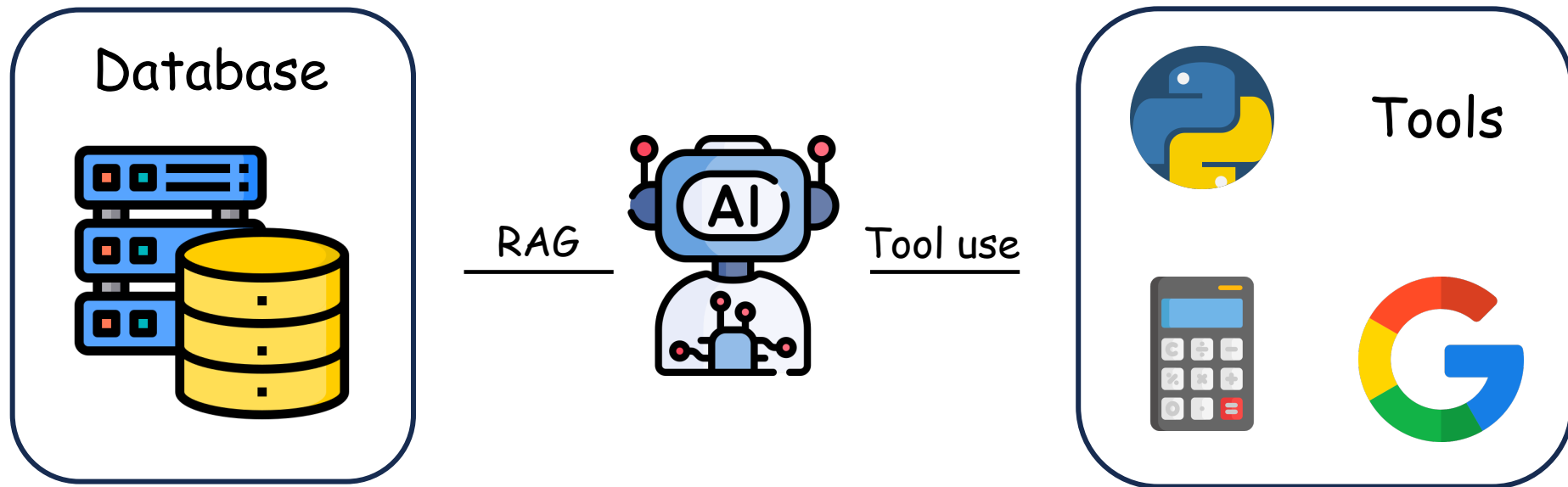


# RAG & Tool use

While prompt engineering can improve how we interact with LLMs, prompting alone often falls short when the model needs

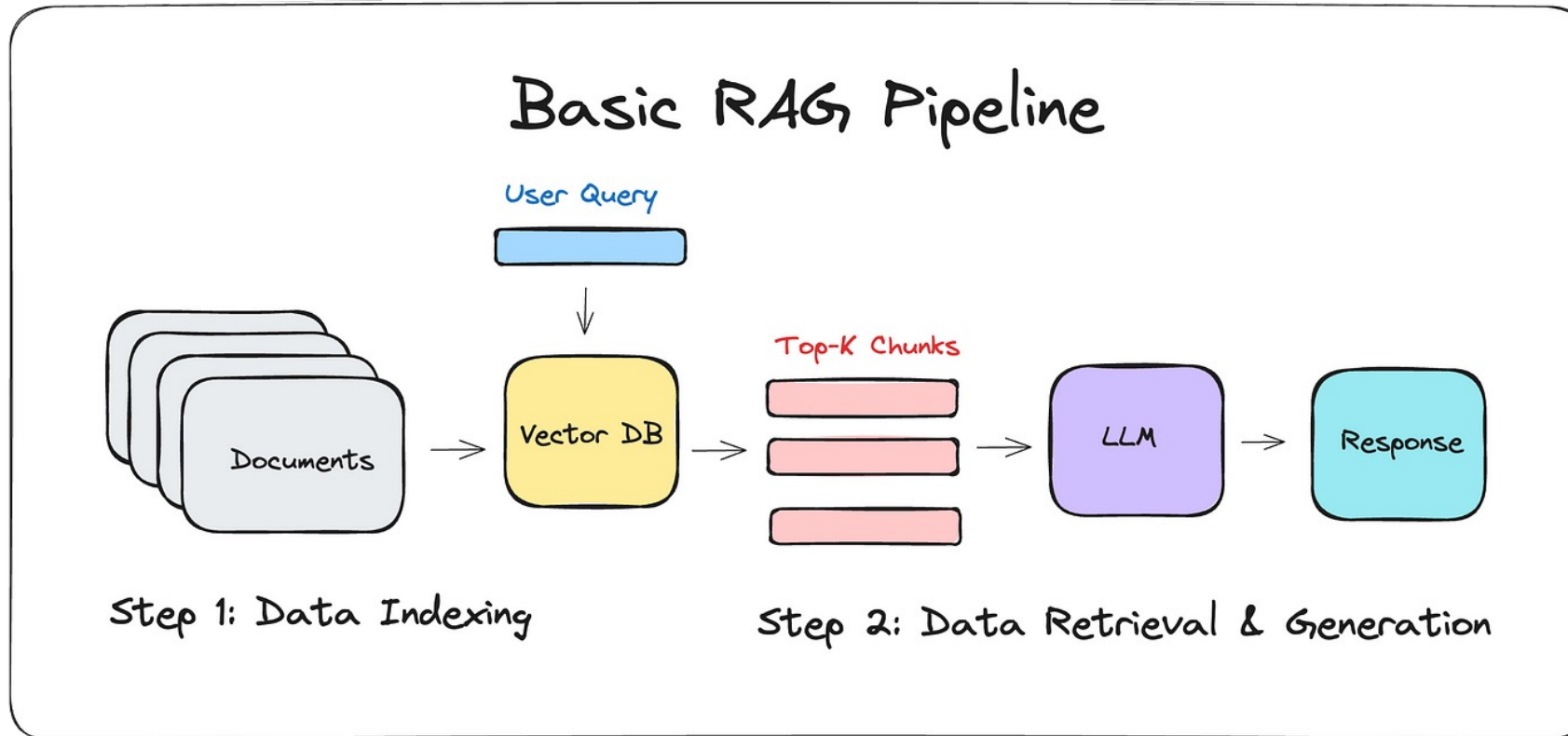
- Up-to-date knowledge, specialized data. *What's the stipend of Runpeng in UNC?*
- Ability to take real-world actions. *Help me book a flight to Seattle?*

To overcome these limitations, techniques like Retrieval-Augmented Generation (RAG) and tool use have been developed, allowing LLMs to access external information and perform tasks that go beyond its intrinsic abilities.



# Retrieval-Augmented Generation (RAG)

LLMs showcase impressive capabilities but encounter challenges like **hallucination, lack of knowledge**. RAG has emerged as a promising solution by **incorporating knowledge from external databases**.



**1) Indexing.** Documents are split into chunks, encoded into vectors, and stored in a vector database.

**2) Retrieval.** Retrieve the Top k chunks most relevant to the question based on semantic similarity.

**3) Generation.** Input the original question and the retrieved chunks together into LLM to generate the final answer.



# Tool use

Tool-augmented LLMs address the limitations of standalone models—such as inability to perform **real-time computations**, access **up-to-date data**—by leveraging APIs, web search, and software tools to dynamically retrieve information and execute complex tasks beyond their internal knowledge.






Category	Example Tools
 Knowledge access	<code>sql_executor(query: str) -&gt; answer: any</code> <code>search_engine(query: str) -&gt; document: str</code> <code>retriever(query: str) -&gt; document: str</code>
 Computation activities	<code>calculator(formula: str) -&gt; value: int   float</code> <code>python_interpreter(program: str) -&gt; result: any</code> <code>worksheet.insert_row(row: list, index: int) -&gt; None</code>
 Interaction w/ the world	<code>get_weather(city_name: str) -&gt; weather: str</code> <code>get_location(ip: str) -&gt; location: str</code> <code>calendar.fetch_events(date: str) -&gt; events: list</code> <code>email.verify(address: str) -&gt; result: bool</code>
 Non-textual modalities	<code>cat_image.delete(image_id: str) -&gt; None</code> <code>spotify.play_music(name: str) -&gt; None</code> <code>visual_qa(query: str, image: Image) -&gt; answer: str</code>
 Special-skilled LMs	<code>QA(question: str) -&gt; answer: str</code> <code>translation(text: str, language: str) -&gt; text: str</code>

Table 1: Exemplar tools for each category.

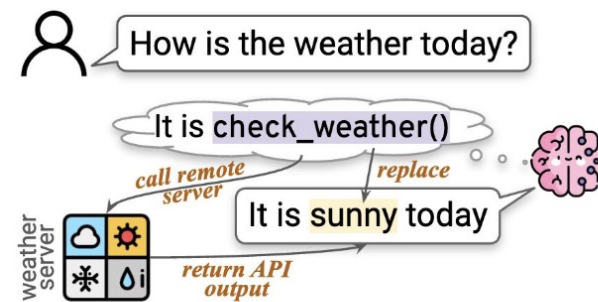


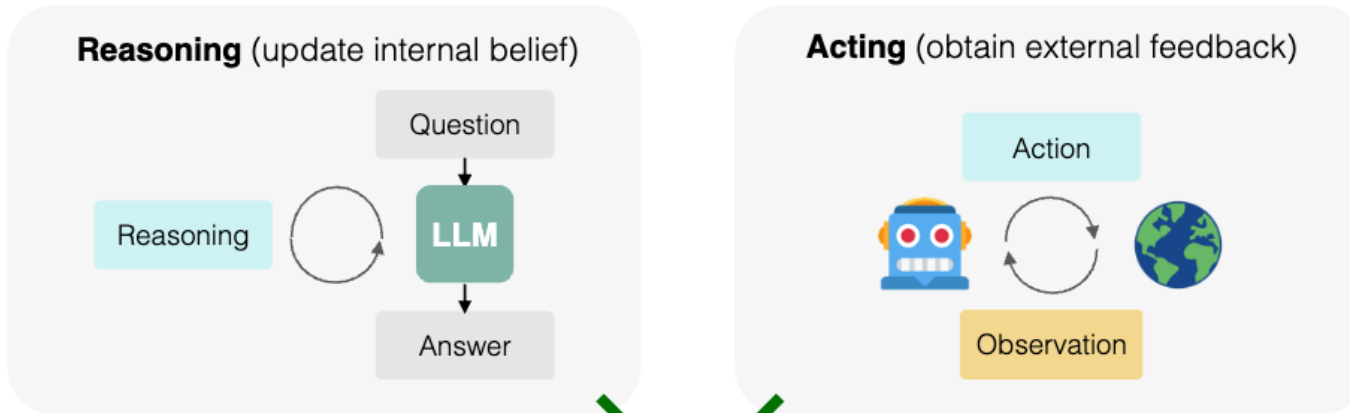
Figure 2: The basic tool use paradigm. LM calls `check_weather` tool by generating text tokens. This call triggers the server to execute the call and return the output `sunny`, using which the LM replaces the API call tokens in the response to the user.



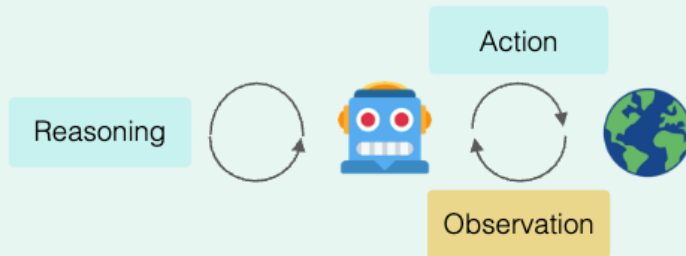
# Put everything together – LLM agents

We want to design a pipeline where LLMs act as the brain, using tools to accomplish specific tasks.

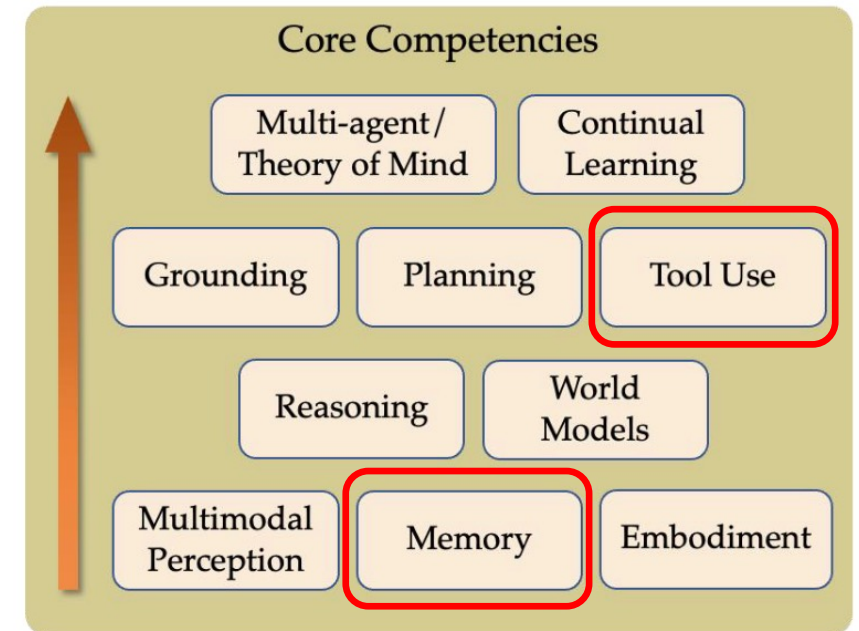
They can **think** ahead, remember past conversations, and use different **tools** to adjust their responses based on the situation and style needed.



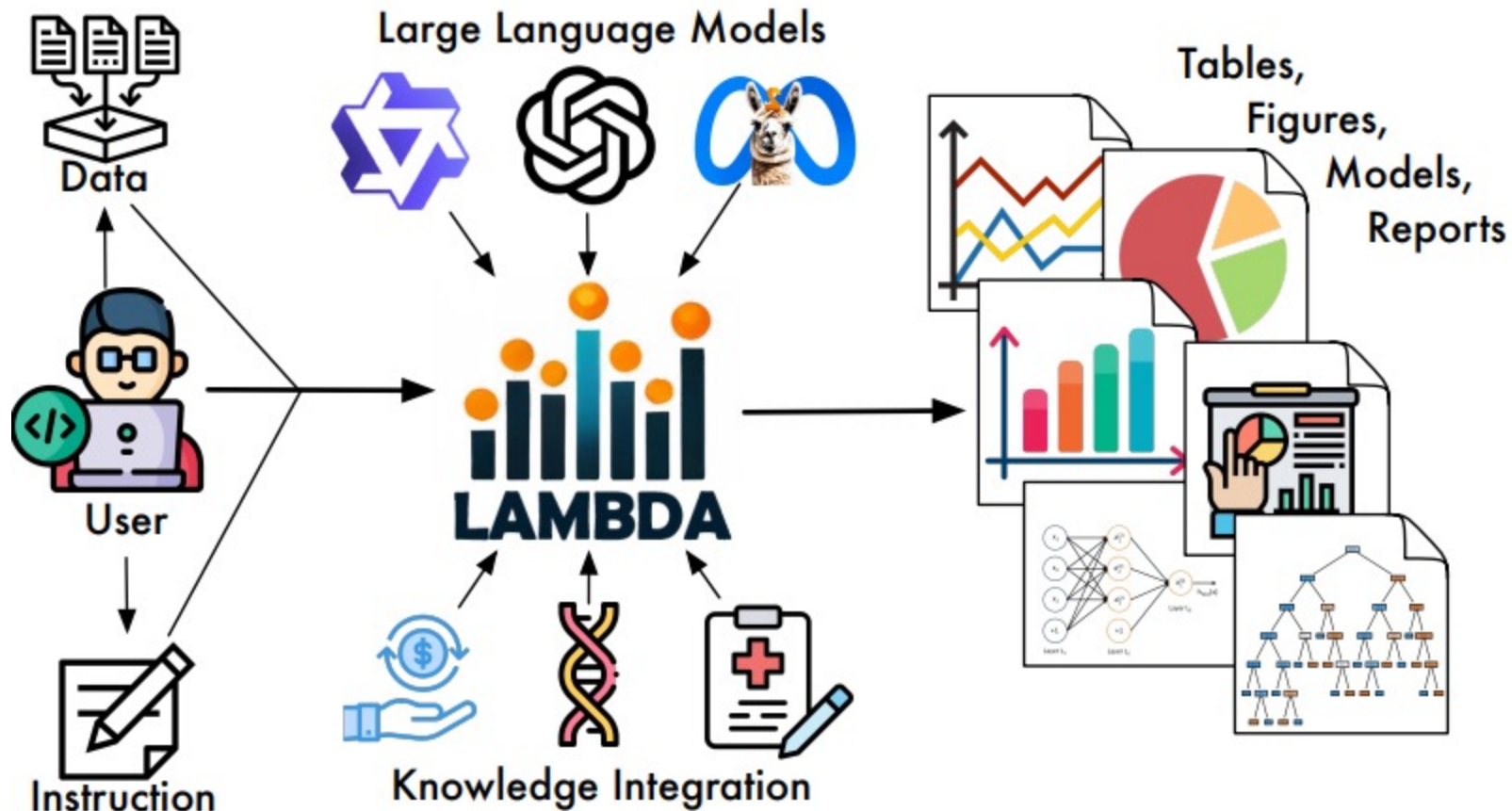
**ReAct**: a new paradigm of agents that **reason and act**



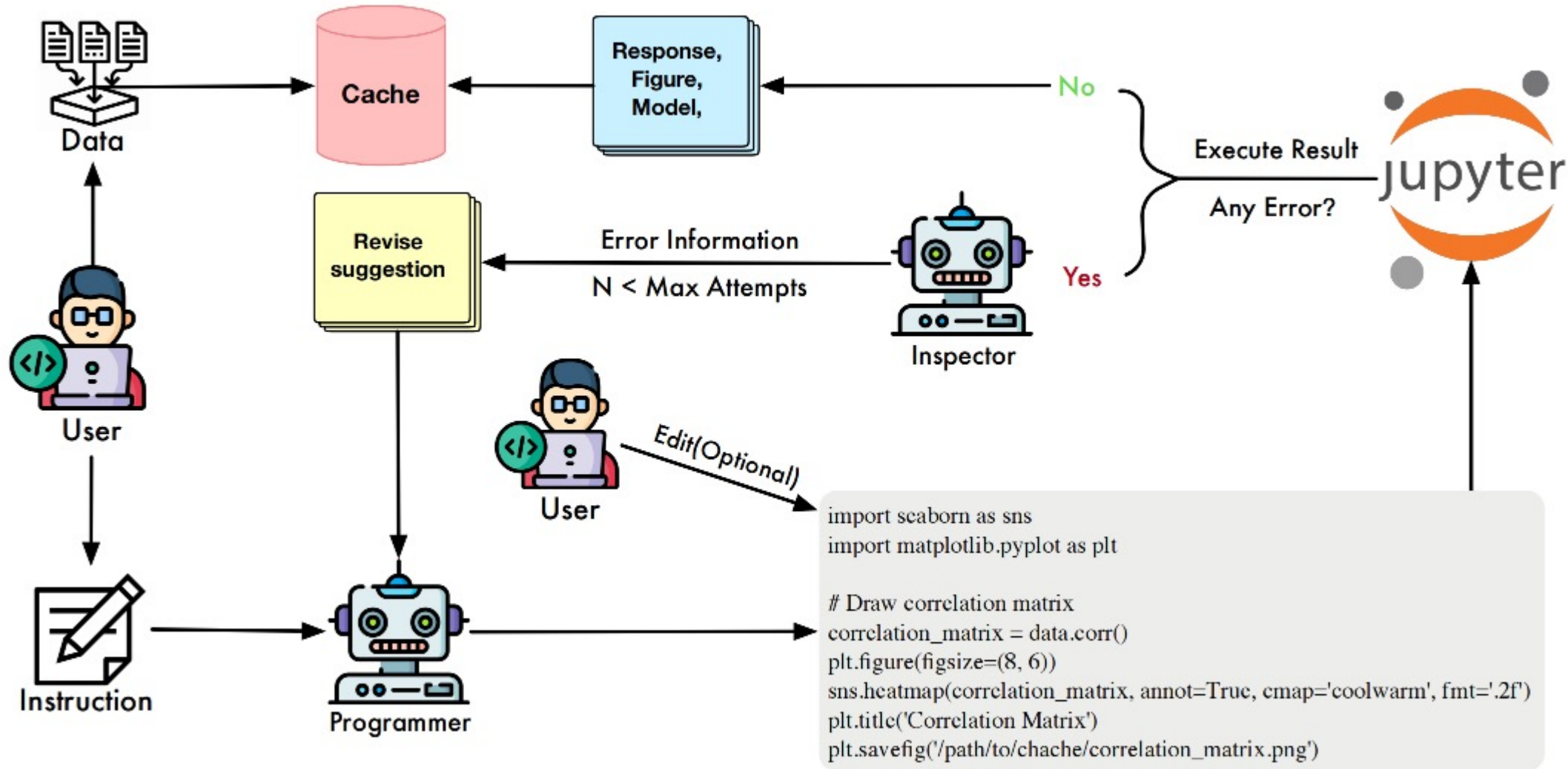
- **Synergy** of reasoning and acting
- **Simple** and intuitive to use
- **General** across domains



# LLM agents Examples - LAMBDA



# LLM agents Examples





Train and select parameters for XGBoost, Print the best parameters and score based on cross-validation.

```
from xgboost import XGBClassifier
from sklearn.model_selection import GridSearchCV
param_grid = {
    'max_depth': [3, 5, 7],
    'learning_rate': [0.1, 0.5, 1.0],
    ....
}
grid_search = GridSearchCV(XGBClassifier(), param_grid, cv=5, scoring='accuracy')
```

Execution result:

Best Parameters: {'learning\_rate': 0.1, 'max\_depth': 3, 'n\_estimators': 100}

Best Score: 0.9553968253968254

[best\\_xgb\\_model.pkl](#)



## Report

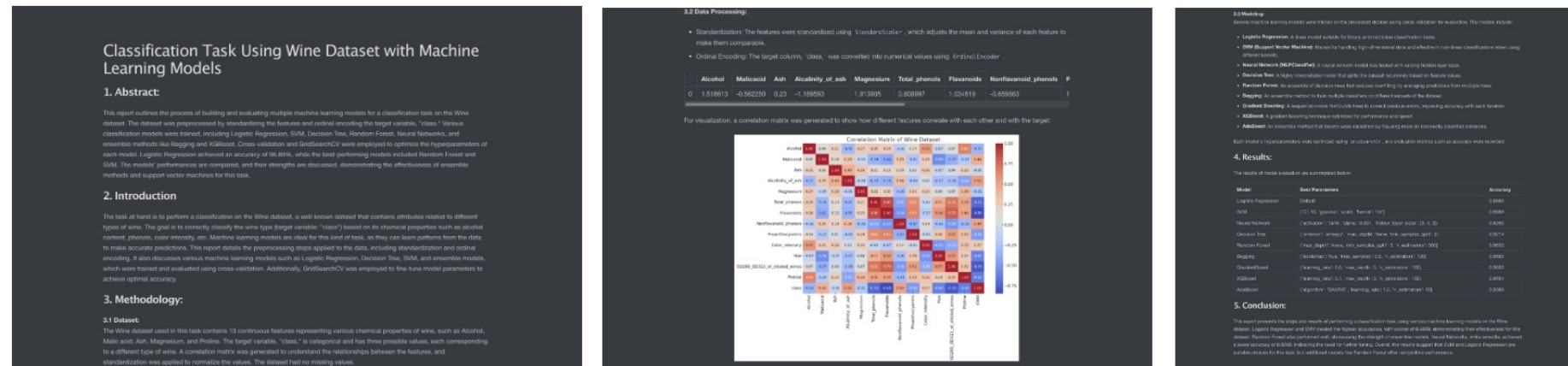


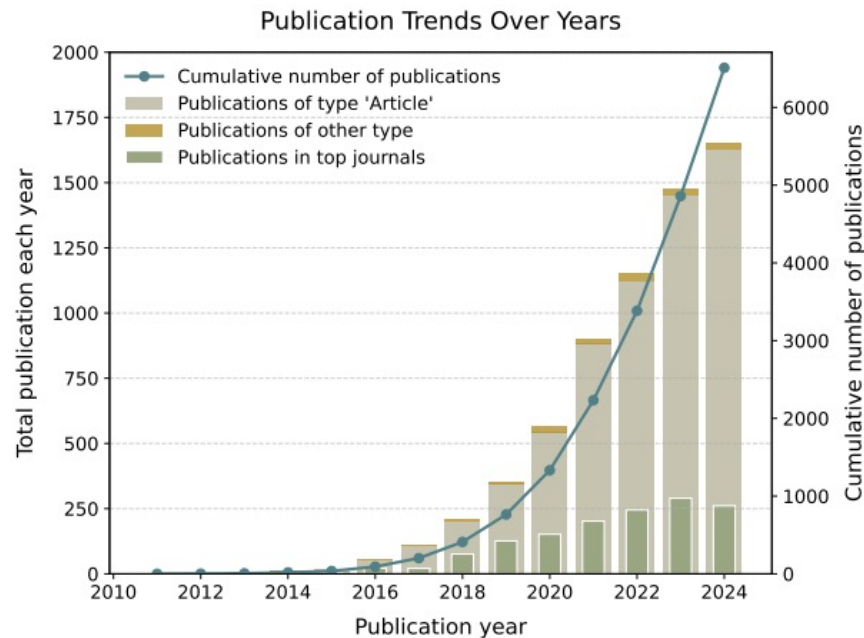
Figure 9: An example of using LAMBDA for classification analysis with the Wine dataset.



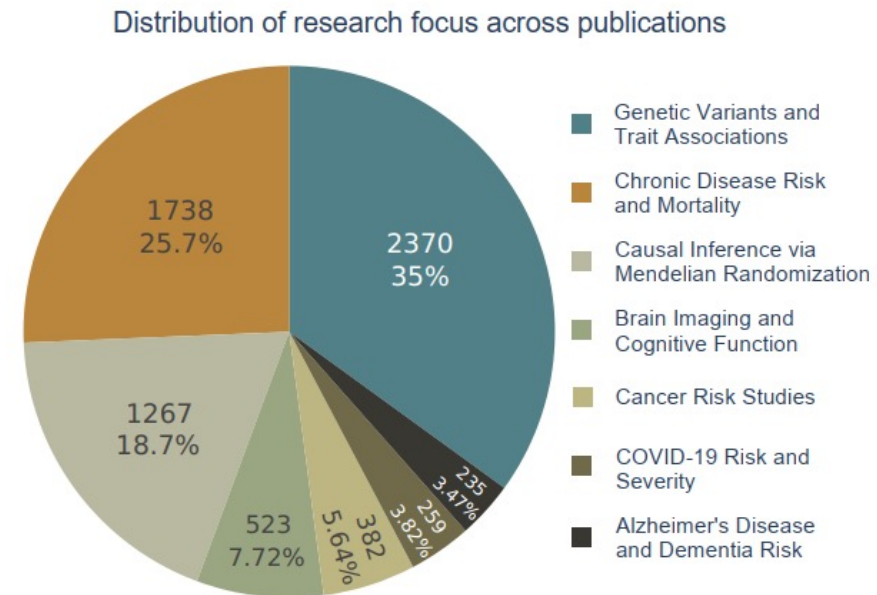
# LLM agents Examples – UKB-KG

**UK Biobank (UKB)** - The world's most important health research database

**However**, the dispersion of UKB-related research across numerous publications poses challenges for efficiently synthesizing and integrating findings.



(a) Publication trends over years

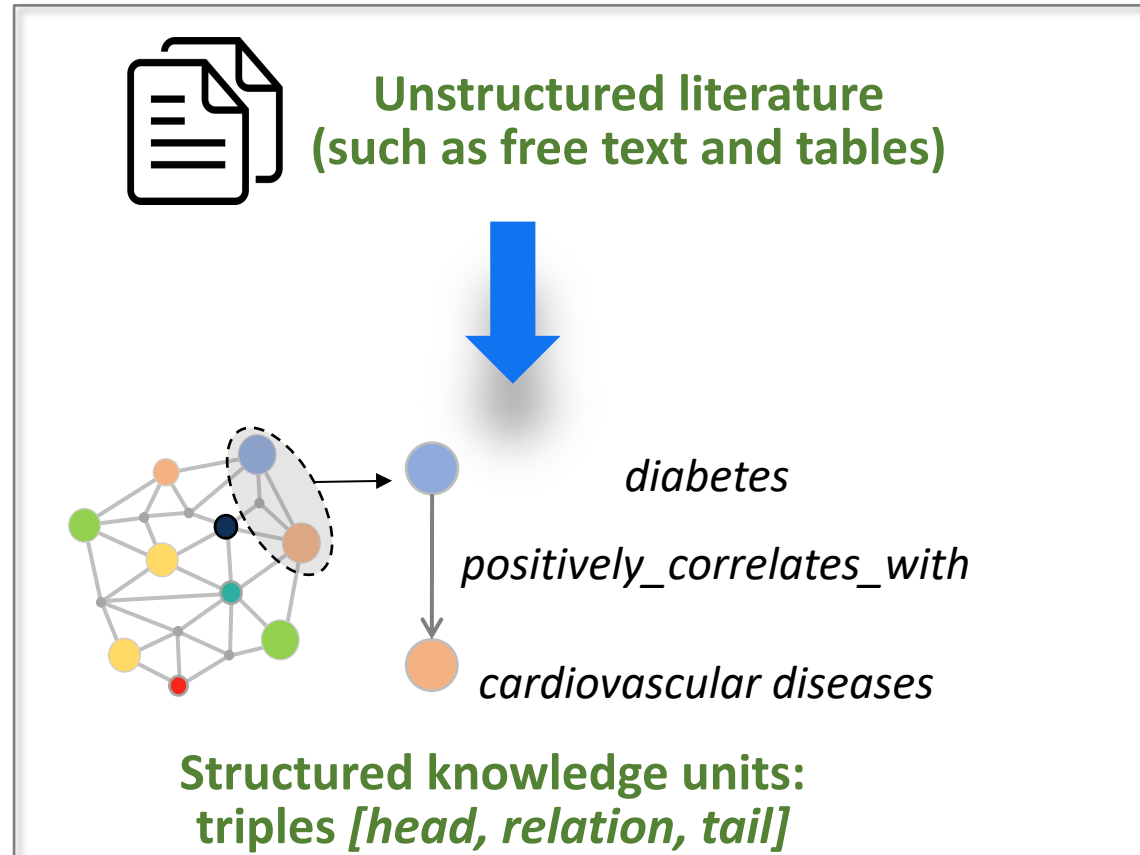


(b) Thematic distribution of publications

A **medical knowledge graph** offers an effective solution to this challenge by structurally organizing and integrating scattered UKB findings, especially with the recent support of **large language models**.

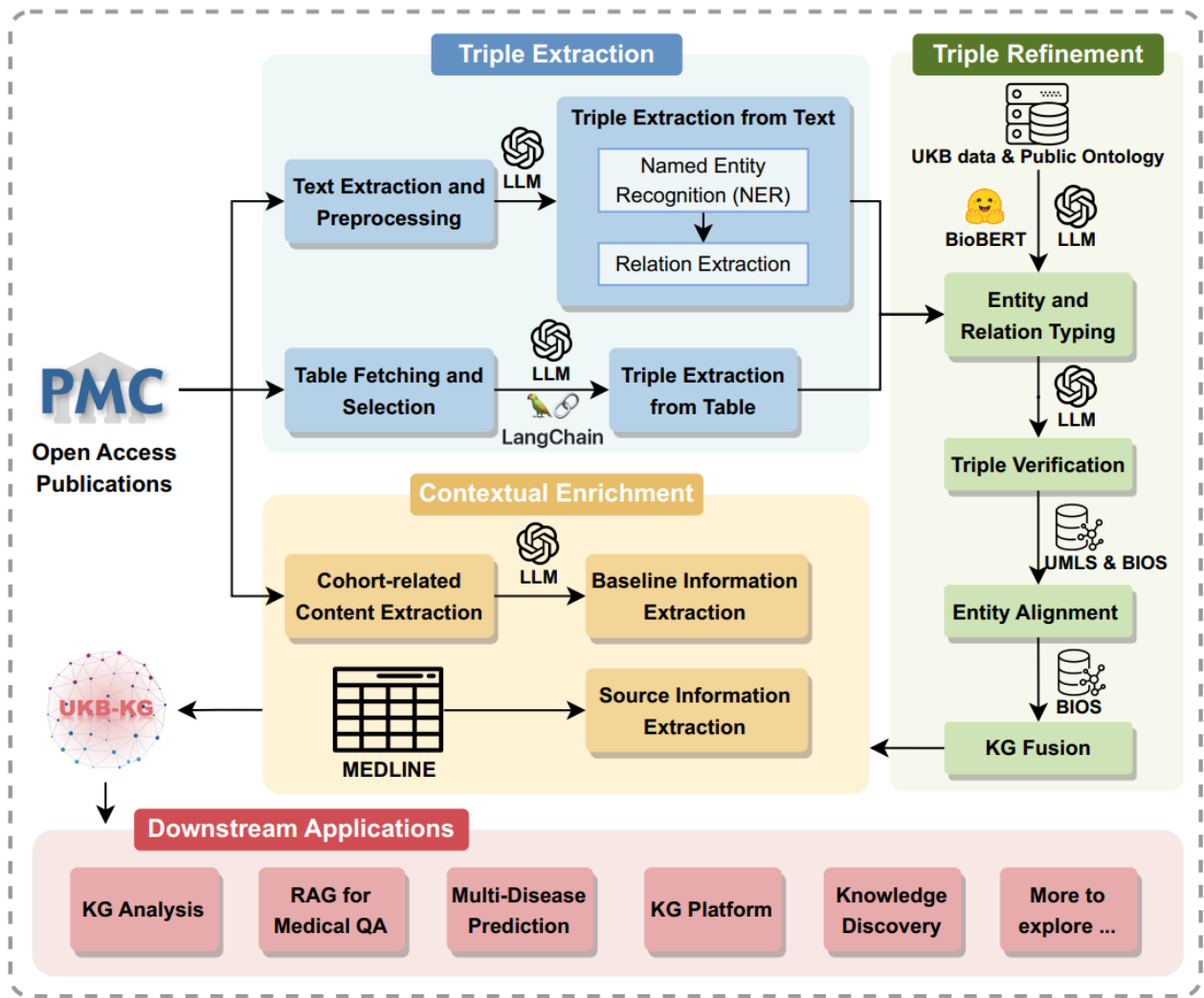
# LLM agents Examples – UKB-KG

**Medical Knowledge Graph**- a powerful tool for organizing and utilizing vast, heterogeneous biomedical data



Large language models are revolutionizing MKG construction, significantly enhanced the extraction and organization of medical knowledge from vast literature

# LLM agents Examples – UKB-KG



Pipeline of UKB-KG

## Multi-Disease Prediction

Incorporate domain knowledge through Knowledge Graph Embeddings (KGE) of UKB-KG to enhance disease prediction.

## Retrieval-Augmented Generation (RAG)

Retrieve relevant knowledge from the UKB-KG to augment medical question answering

## Interactive Platform for UKB-KG

### Chatbot

How can I help you?

Are diabetes related to smoking?

Yes, smoking is related to an increased risk of developing type 2 diabetes. Smoking initiation has been shown to have widespread risk effects, including on common diseases such as type 2 diabetes. It acts as a risk factor for this condition, adding to the body's stress and potentially contributing to insulin resistance and other metabolic issues related to diabetes.

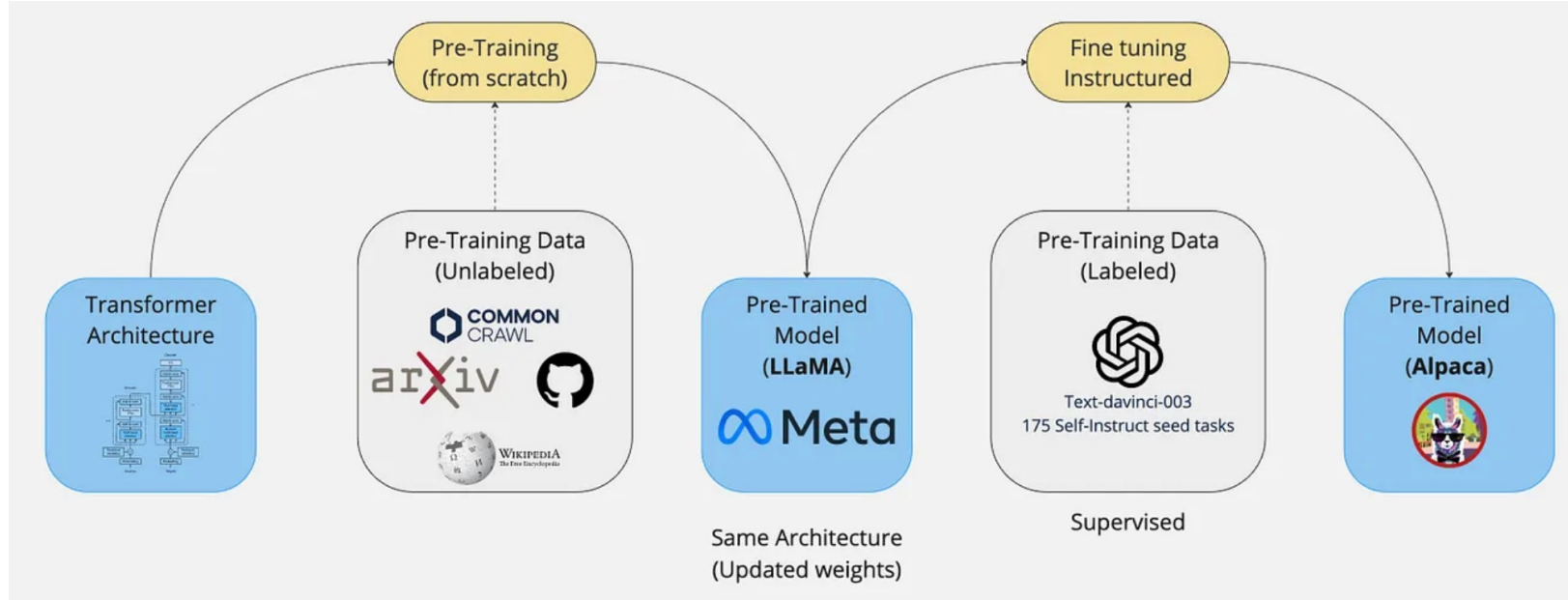
See context

6. Path from **diagnosed diabetes** to **smoking**

2. Path from **complicated diabetes** to **smoking**

Path diagrams illustrating relationships between diseases and smoking.

# Training methods – SFT



Prompting methods are inherently limited by the capabilities and biases of the base model.

**Supervised Fine-Tuning (SFT)** is the most basic training approach for aligning large language models with human intentions. It involves fine-tuning a pre-trained model on a dataset of input-output pairs **to follow instructions or perform specific tasks**.

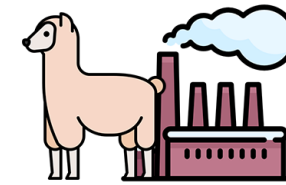
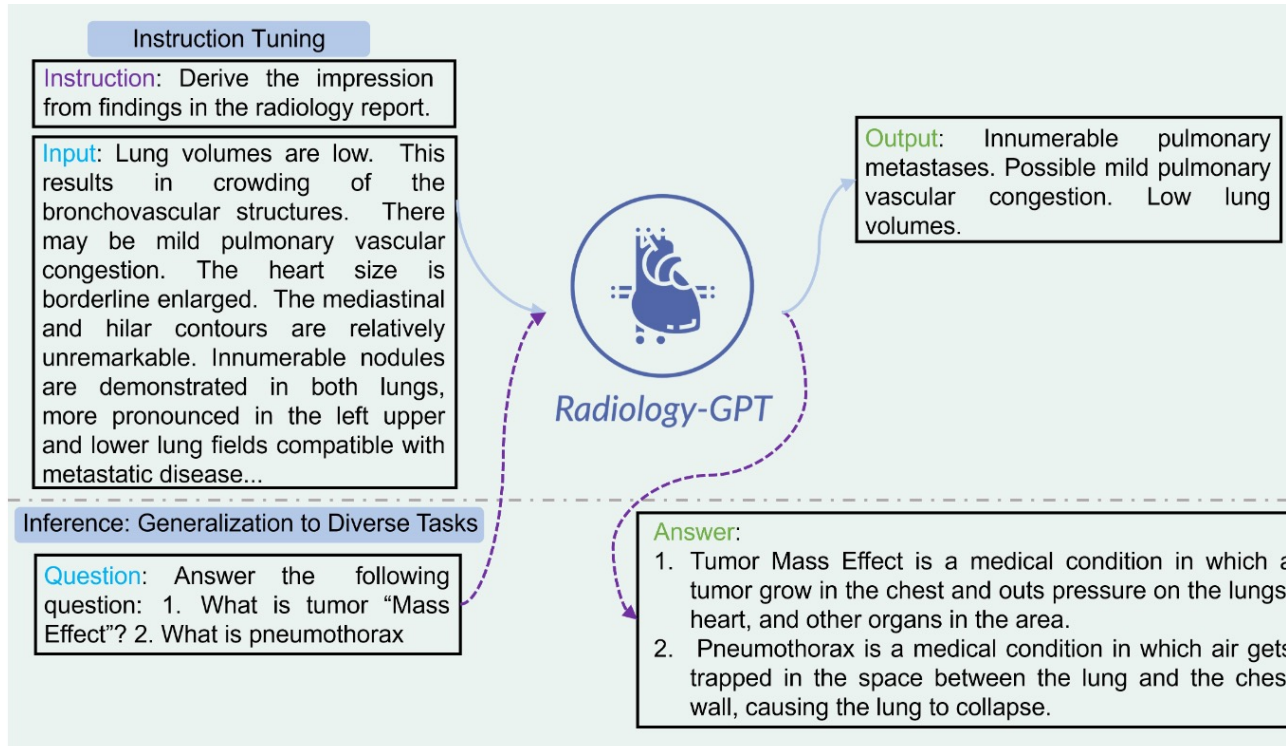
System\_Prompt + <User>: [User\_Input] + <System>: [Response]</s>

**Loss**



# SFT – Example

Among all the training methods, SFT is the most direct one. There have been many works use SFT at the early stage.



**LLaMA-Factory**  
Easy and Efficient LLM Fine-Tuning

Easily fine-tune 100+ large language models with Web UI

Specialize a large language model (LLM) for generating or interpreting the “Impression” section of radiology reports

# Alignment

RLHF allows models to learn from *comparative feedback*—for example, choosing one response over another—so that the model can generalize better to unseen situations and generate outputs that are more aligned with human preferences.

## 3H principles:

- **Helpful** (Improves usability)
- **Honest** (Enhances trustworthiness)
- **Harmless** (Mitigates harmful outputs)



# Alignment

## Challenges

1. **Human in the loop is expensive** – *Reward Modelling*
2. **Human judgments are noisy and miscalibrated.** – *Preference learning*

Score the helpfulness of the following response, 1-10

What are the steps for making a simple cake?

1. Preheat oven to 350°F (175°C).
2. Grease and flour a cake pan.
3. In a bowl, combine 2 cups flour, 1.5 cups sugar, 3.5 tsp baking powder, and a pinch of salt.
4. Add 1/2 cup butter, 1 cup milk, and 2 tsp vanilla; mix well.
5. Beat in 3 eggs, one at a time.
6. Pour batter into the pan.
7. Bake for 30-35 minutes or until a toothpick comes out clean.
8. Let cool, then frost or serve as desired.

# Alignment – Bradley Terry model

Which of these two responses is more helpful?

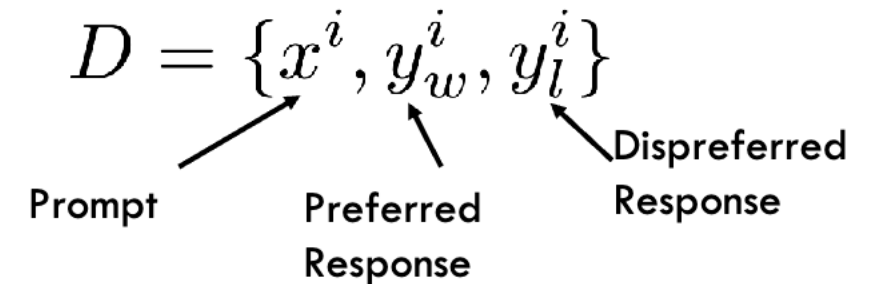
What are the steps for making a simple cake?

1. Preheat oven to 350°F (175°C).
2. Grease and flour a cake pan.
3. In a bowl, combine 2 cups flour, 1.5 cups sugar, 3.5 tsp baking powder, and a pinch of salt.
4. Add 1/2 cup butter, 1 cup milk, and 2 tsp vanilla; mix well.
5. Beat in 3 eggs, one at a time.
6. Pour batter into the pan.
7. Bake for 30-35 minutes or until a toothpick comes out clean.
8. Let cool, then frost or serve as desired.

What are the steps for making a simple cake?

1. Warm up the oven.
2. Grease a cake pan.
3. Blend dry ingredients in a bowl.
4. Incorporate butter, milk, and vanilla.
5. Mix in the eggs.
6. Pour into the prepared pan.
7. Bake until golden brown.
8. Add frosting if desired.

Instead of asking labelers to assign scores, we collect their preferences through pairwise comparisons.



# Alignment - RLHF

How do we get feedback for the reward while training our RL model?

$$p(y_w > y_l | x) = \sigma(\underline{r(x, y_w)} - \underline{r(x, y_l)})$$

Logistic function;  
which is equivalent  
to using softmax:

$$\frac{1}{1 + e^{-x}}$$

$$p(y_w > y_l | x) = \frac{\exp(r(x, y_w))}{\exp(r(x, y_w)) + \exp(r(x, y_l))}$$

Train a Reward Model (RM) on preference data to predict preferences!

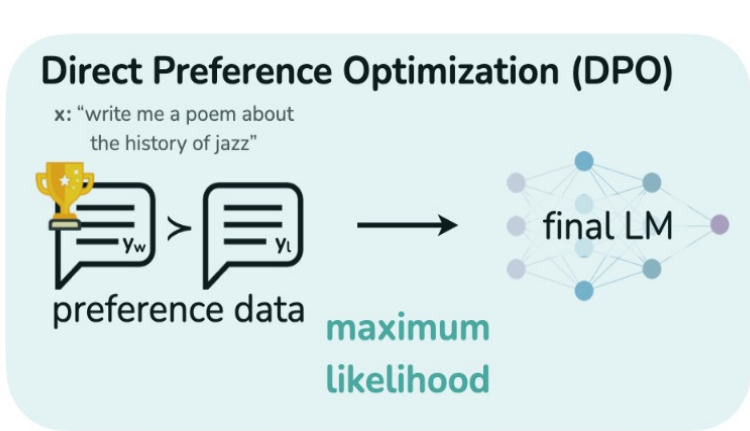
$$\mathcal{L}_R(\phi, D) = -\mathbb{E}_{(x, y_w, y_l) \sim D} [\log \sigma(r(x, y_w) - r(x, y_l))]$$

Use RL algorithm to train optimize the LLM  $\pi$

$$\max_{\pi_\theta} \mathbb{E}_{x \sim D, y \sim \pi_\theta(y|x)} [\underline{r_\phi(x, y)}]$$

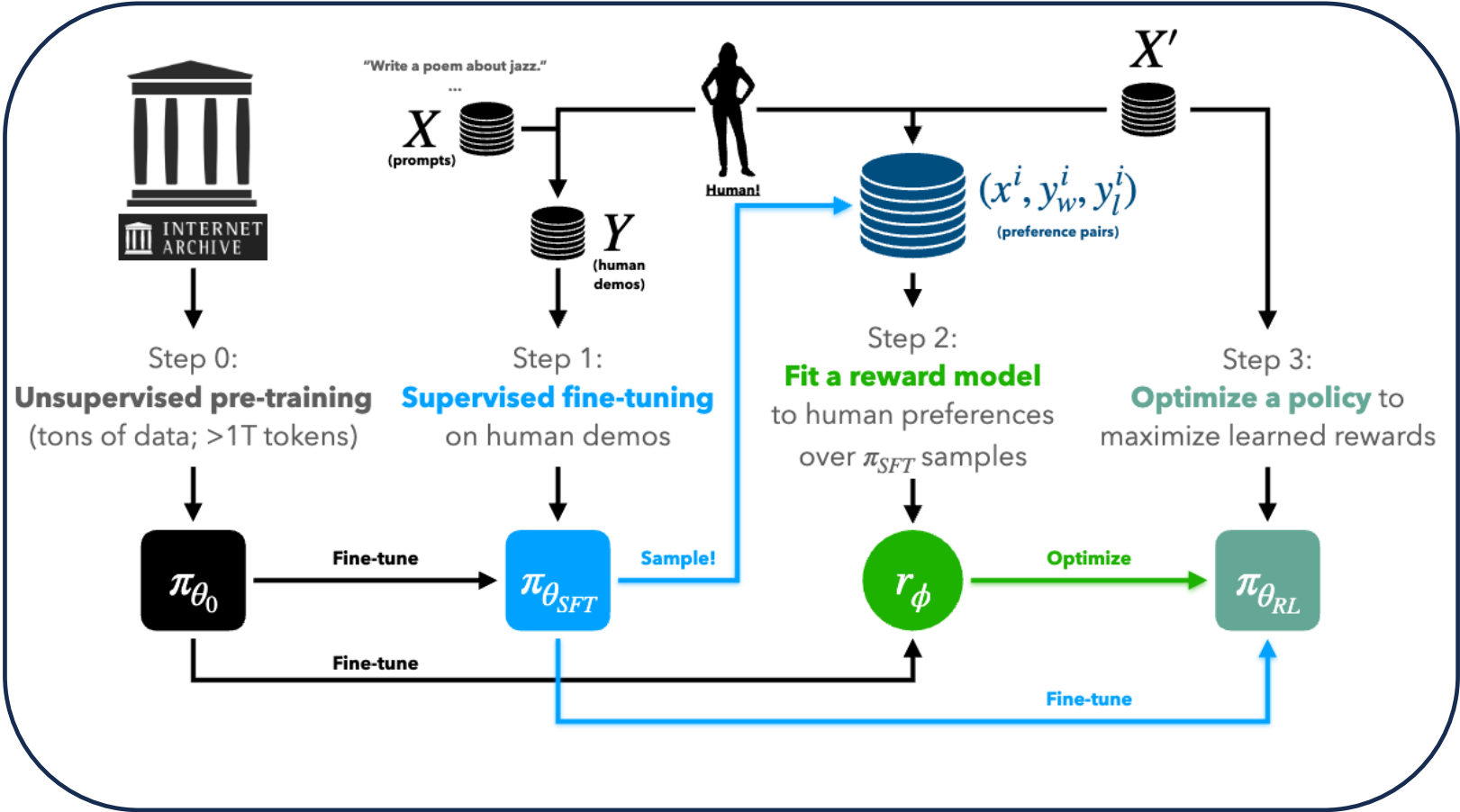
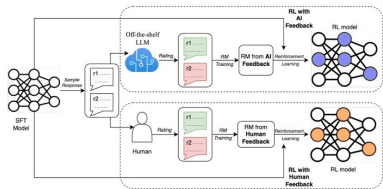
# Alignment - Improvements

- The reward model is hard to train and large.



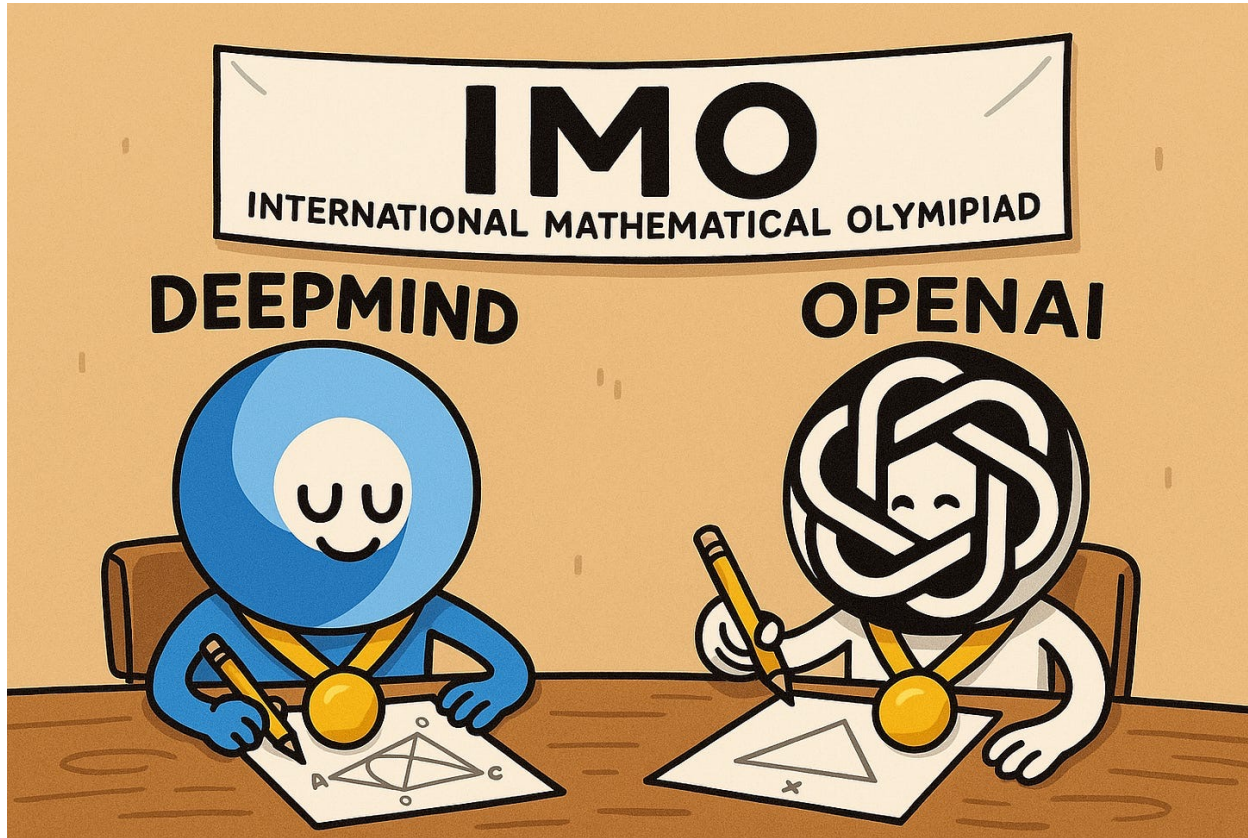
- Human labeling remains costly.

## RLAIF - RL from AI Feedback for LLM





# Learning to Reasoning with LLMs



DeepMind and OpenAI achieve IMO Gold.

The secret behind the success - Reasoning

*We achieved this year's result using an advanced version of Gemini Deep Think – an enhanced **reasoning mode** for complex problems that incorporates some of our latest research techniques, including parallel thinking. This setup enables the model to simultaneously **explore and combine multiple possible solutions** before giving a final answer, rather than pursuing a single, linear chain of thought.*



DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

# Learning to Reasoning with LLMs

## What is reasoning?

Reasoning refers to the process of **thinking through** a problem or situation in order to form a logical conclusion or make a decision. It involves using evidence, facts, and logic to arrive at a solution or answer.

## Why Reasoning is important?

Even for humans, problems like **math** or **complex question answering** are difficult to solve with direct answers alone, as they often require multi-step reasoning or integrating diverse pieces of information.

## How can LLM Reasoning?

Generate sequence of tokens representing **intermediate steps** in the reasoning process.

### **Problem 1:**

**Question:** Two trains running in opposite directions cross a man standing on the platform in 27 seconds and 17 seconds respectively and they cross each other in 23 seconds. The ratio of their speeds is:

**Options:** A)  $3/7$  B)  $3/2$  C)  $3/88$  D)  $3/8$  E)  $2/2$

**Rationale:** Let the speeds of the two trains be  $x$  m/sec and  $y$  m/sec respectively. Then, length of the first train =  $27x$  meters, and length of the second train =  $17y$  meters.  $(27x + 17y) / (x + y) = 23 \rightarrow 27x + 17y = 23x + 23y \rightarrow 4x = 6y \rightarrow x/y = 3/2$ .

**Correct Option:** B

# Learning to Reasoning with LLMs

How can we guide LLMs to perform reasoning?

## 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?

Exemplar

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? **Thought**

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. ✅

Or some other prompting methods:

1. Least-to-Most Prompting
2. Self reflection/ verification

Can we go beyond prompting?

*"We have found that the performance of o1 consistently improves with more reinforcement learning (train-time compute) and with more time spent thinking (test-time compute). The constraints on scaling this approach differ substantially from those of LLM pretraining, and we are continuing to investigate them."* — OpenAI o1 release blog



# Learning to Reasoning with LLMs

## Reward guided decoding

**ORM**

**Question:**  
Three pencils and a jumbo eraser cost \$1.24. Five pencils and a jumbo eraser cost \$1.82. No prices include tax. In cents, what is the cost of a pencil?

**Answer:**

Let the cost of a pencil be  $p$  cents, and the cost of a jumbo eraser be  $e$  cents.  
From the problem, we are given two equations based on the cost:  
 $3p + 2 = 124$   
 $5p + e = 182$   
To solve this system of equations, we subtract the first equation from the second to eliminate  $e$ :  
 $(5p + e) - (3p + 2) = 182 - 124$   
Solving for  $p = 58/2 = 27$   
Thus, the cost of a pencil is 29 cents.

}

✗

**PRM**

**Question:**  
Three pencils and a jumbo eraser cost \$1.24. Five pencils and a jumbo eraser cost \$1.82. No prices include tax. In cents, what is the cost of a pencil?

**Answer:**

1

Let the cost of a pencil be  $p$  cents, and the cost of a jumbo eraser be  $e$  cents.

✓

2

From the problem, we are given two equations based on the cost:  
 $3p + 2 = 124$ ,  $5p + e = 182$

✓

3

To solve this system of equations, we subtract the first equation from the second to eliminate  $e$ :  $(5p + e) - (3p + 2) = 182 - 124$

✓

4

Solving for  $p = 58/2 = 27$

✗

5

Thus, the cost of a pencil is 29 cents.

✗

Reward hacking.

# Learning to Reasoning with LLMs

## Reward guided planning

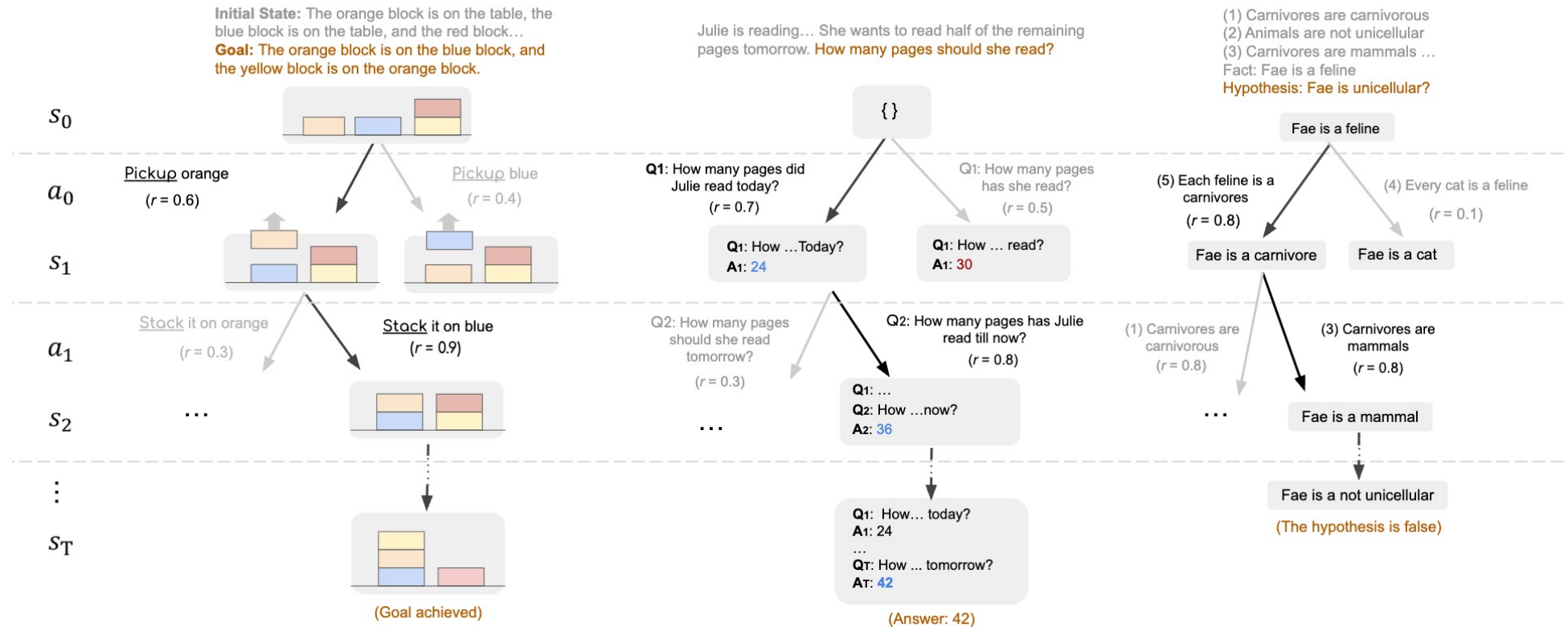


Figure 2: RAP for plan generation in Blocksworld (left), math reasoning in GSM8K (middle), and logical reasoning in PrOntoQA (right).

# DeepseekR1 and RLVR



---

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within `<think>` `</think>` and `<answer>` `</answer>` tags, respectively, i.e., `<think>` reasoning process here `</think>` `<answer>` answer here `</answer>`. User: **prompt**. Assistant:

---

Table 1 | Template for DeepSeek-R1-Zero. **prompt** will be replaced with the specific reasoning question during training.

## Reinforcement learning with verifiable reward

$$R(\hat{y}, y) = \begin{cases} 1, & \text{is\_equivalent}(\hat{y}, y) \\ -1, & \text{otherwise} \end{cases}$$

Instead of training a separate **reward model** or explicitly assigning credit to **each reasoning step**, we allow the model to discover effective reasoning strategies through trial and error, only judge the output through correctness of final answer.

1. Reduce the computational burden of training a separate reward model.
2. Eliminate the risk of reward hacking.



# DeepseekR1 and RLVR

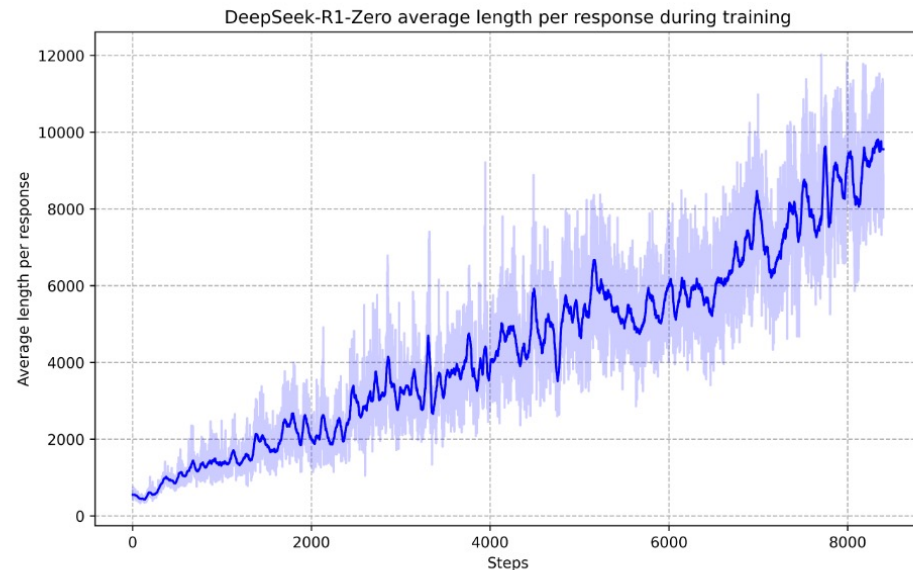


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

During the training process, they observe the increase in reasoning length and “aha moment” where the model learns to self-reflection and verify.

Question: If $a > 1$ , then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to
Response: <think>
To solve the equation $\sqrt{a - \sqrt{a + x}} = x$ , let's start by squaring both ...
$(\sqrt{a - \sqrt{a + x}})^2 = x^2 \implies a - \sqrt{a + x} = x^2$ .
Rearrange to isolate the inner square root term:
$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$
...
Wait, wait. Wait. That's an aha moment I can flag here.
Let's reevaluate this step-by-step to identify if the correct sum can be ...
We started with the equation:
$\sqrt{a - \sqrt{a + x}} = x$
First, let's square both sides:
$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$
Next, I could square both sides again, treating the equation: ...
...

Table 3 | An interesting “aha moment” of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.

# Some of the Challenges

- **Ensuring Faithfulness of the Reasoning Process**
- **Reducing Reasoning Latency**
- **Handling General Reasoning Tasks with Unverifiable Rewards**

# RLVR – Applications

**Problem:** What does this image depict in terms of its content?

A)Brain tissue  
B)Intestinal tissue  
C)Kidney tissue  
D)Breast tissue

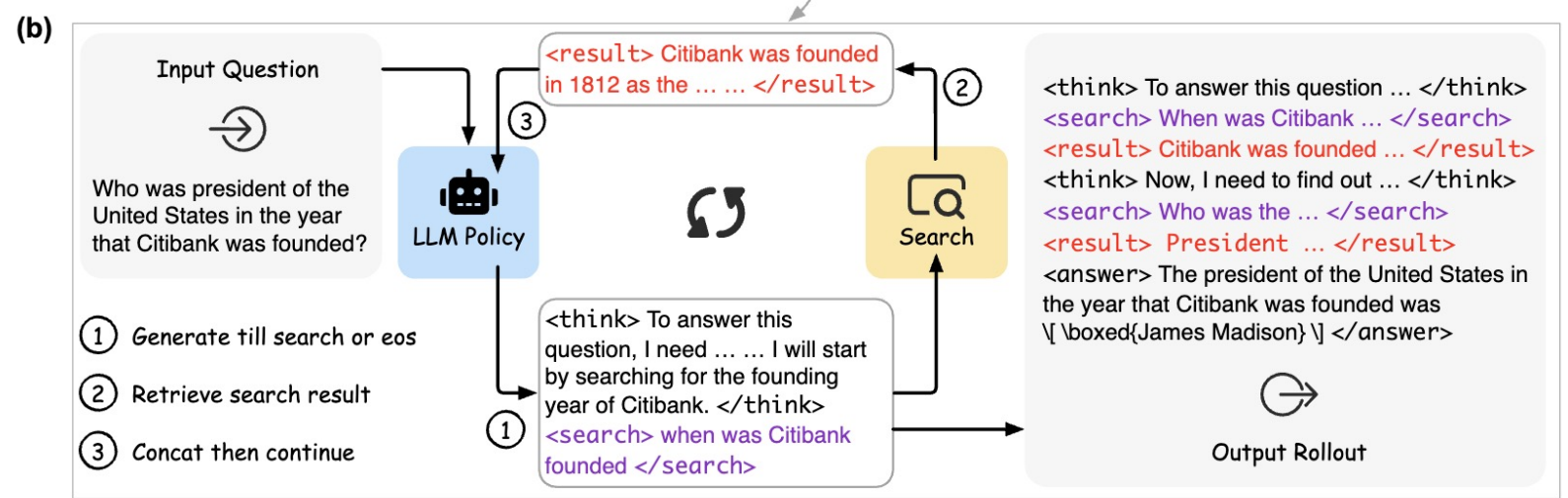
**Med-R1 Response:**

**Think:** The image appears to be a mammogram, which is a type of X-ray used to detect abnormalities in the breast tissue. The presence of dense, irregular patterns and a central area of lighter density suggest the presence of breast tissue.

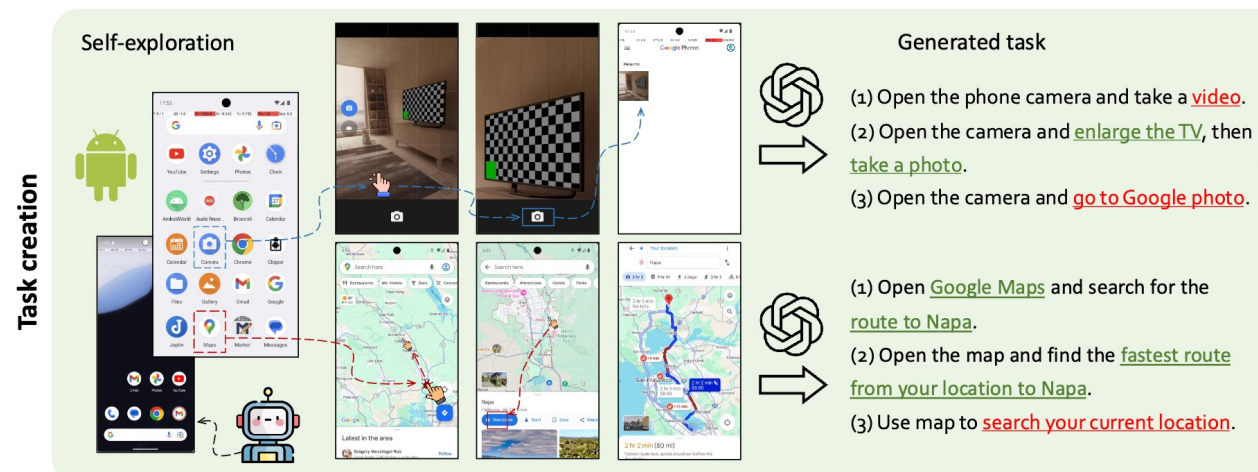
**Answer:** D

**Groundtruth:** D

## Medical Reasoning



## Use of Search engine



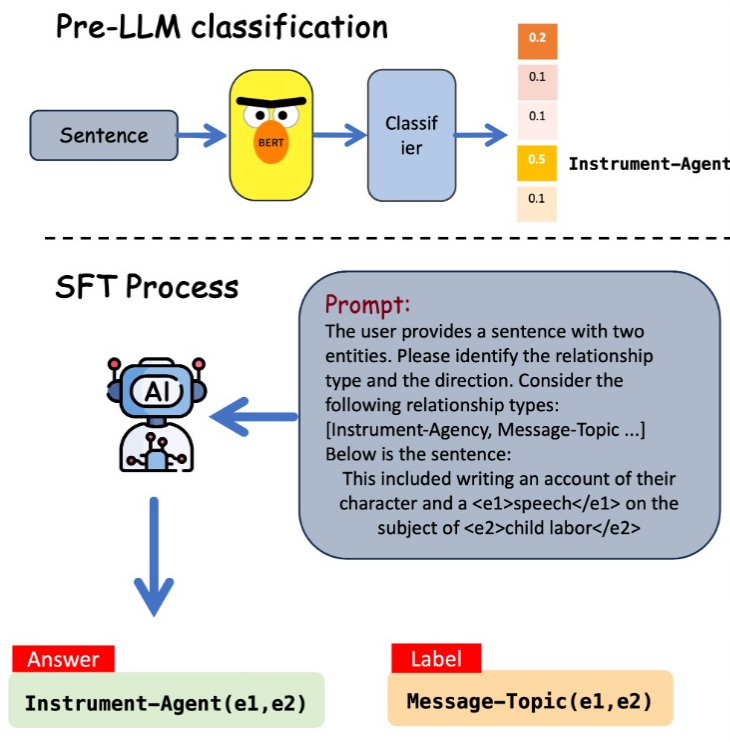
## Use of smart phone

# Reasoning – Example R1-RE

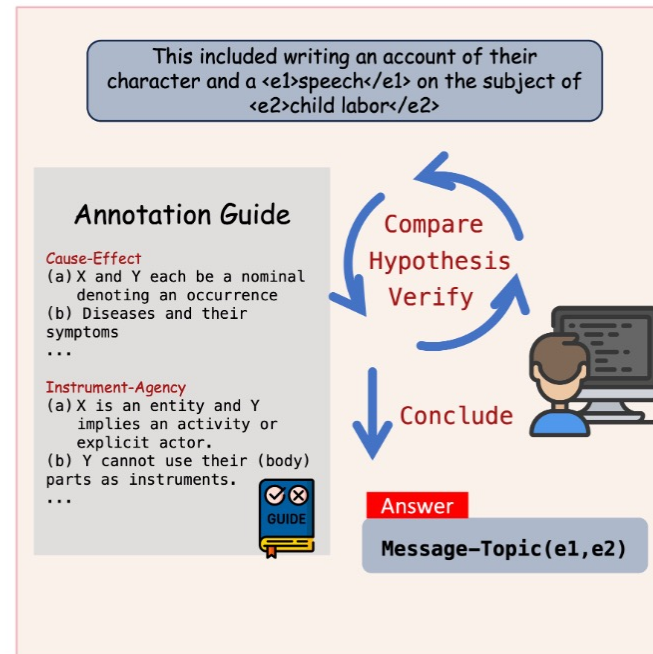
**Sentence** Olanzapine was also associated with more frequent reports of weight gain and significantly greater VA costs ...

**Label** [[Olanzapine:drug, risk-factor-of, weight gain:symptom]]

Relation extraction is a critical step in knowledge graph construction.



## Human Annotation Process



Existing method focus on learning a mapping from sentence to triplet, which cannot generalize to OOD.

Human annotators are generalizable across tasks because they can reason on the requirements.



# Reasoning – Example R1-RE

**R1-RE-7B**

We need to determine the relationship between <e1>schizophrenia</e1> and <e2>prolactin-increasing antipsychotics</e2>. Let's break this down step by step.

- Identify the entities**:
  - <e1>schizophrenia</e1> is the disease or condition.
  - <e2>prolactin-increasing antipsychotics</e2> is a type of medication or treatment.
- Analyze the relationship**:
  - The sentence states that schizophrenia is associated with a specific type of antipsychotics (prolactin-increasing antipsychotics) being used in the treatment or management of schizophrenia.
- Apply the definitions**:
  - causes**: This wouldn't apply because the sentence suggests the antipsychotics are being used to treat, not cause, schizophrenia.
  - risk\_factor\_of**: This would apply if the antipsychotics increased the risk of schizophrenia, but here it's the other way around; the antipsychotics are being used because of schizophrenia, not the other way.
- Conclusion**:
 

Given that the sentence directly states that antipsychotics are used in the treatment of schizophrenia, this fits squarely into the definition of a treatment relationship. The antipsychotics are specifically used to address or manage schizophrenia, which is the essence of the treatment\_for relationship.

**Final Answer**

Treatment\_for(e2,e1) ✓

Figure 6: Case study comparing the chain-of-thought (COT) reasoning of **R1-RE-7B** and Qwen2.5-7B-Instruct. Due to space constraints, some COT outputs are omitted; the complete COT reasoning process for **R1-RE-7B** is provided in Appendix A.2.

**Target sentence**

Monitoring prolactinemia and addressing hyperprolactinemia is paramount in women with <e1>schizophrenia</e1> being treated with <e2>prolactin-increasing antipsychotics</e2>.

**Qwen-2.5-7B**

The sentence mentions that monitoring prolactinemia and addressing hyperprolactinemia is important for women with schizophrenia who are treated with prolactin-increasing antipsychotics. This implies that prolactin-increasing antipsychotics can lead to prolactinemia and hyperprolactinemia, and that these conditions are associated with the treatment.

Given the definitions, this is best classified under the "risk\_factor\_of" relationship as it indicates that prolactin-increasing antipsychotics (e2) could increase the risk or likelihood of prolactinemia and hyperprolactinemia (e1). This fits into the susceptibility and prevalence category of "risk\_factor\_of".

**Final Answer**

Risk\_factor\_of(e2,e1) ✗

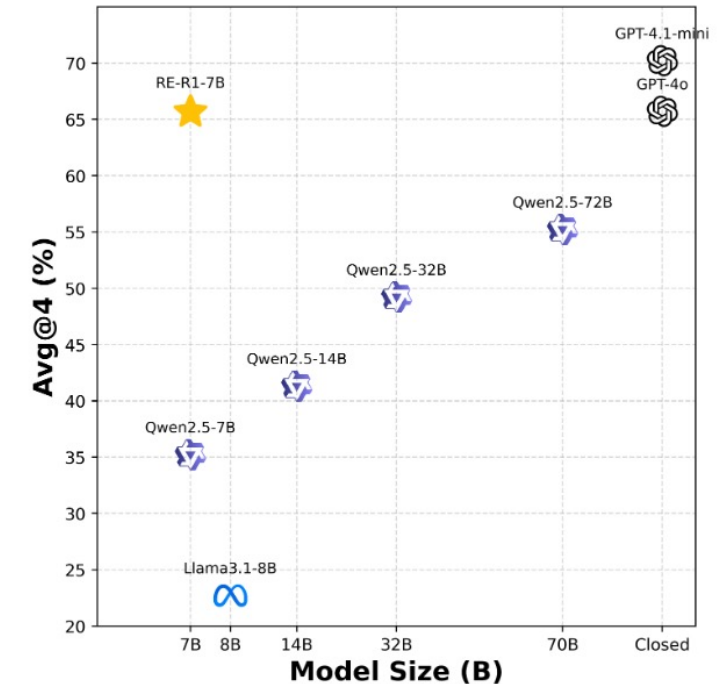


Figure 1: Testing accuracy on the MDKG dataset for R1-RE-7B trained on the Sem-2010 dataset, compared with other models. Detailed results are provided in Table 3.

# References

- Lu, Z. et al.** Large language models in biomedicine and health: current research landscape and future directions. *J. Am. Med. Inform. Assoc.* 31, 1801–1811 (2024).
- Ouyang, L. et al.** Training language models to follow instructions with human feedback. Preprint at <https://doi.org/10.48550/arXiv.2203.02155> (2022).
- OpenAI et al.** GPT-4 Technical Report. Preprint at <https://doi.org/10.48550/arXiv.2303.08774> (2024).
- Rafailov, R. et al.** Direct Preference Optimization: Your Language Model is Secretly a Reward Model. Preprint at <https://doi.org/10.48550/arXiv.2305.18290> (2024).
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O.** Proximal Policy Optimization Algorithms. Preprint at <https://doi.org/10.48550/arXiv.1707.06347> (2017).
- Wang, C. et al.** A Survey for Large Language Models in Biomedicine. Preprint at <https://doi.org/10.48550/arXiv.2409.00133> (2024).
- Wei, J. et al.** Finetuned Language Models Are Zero-Shot Learners. Preprint at <https://doi.org/10.48550/arXiv.2109.01652> (2022).
- Wei, J. et al.** DrugReAlign: a multisource prompt framework for drug repurposing based on large language models. *BMC Biol.* 22, 226 (2024).
- Xiao, H. et al.** A Comprehensive Survey of Large Language Models and Multimodal Large Language Models in Medicine. *Inf. Fusion* 117, 102888 (2025).
- Yao, S. et al.** ReAct: Synergizing Reasoning and Acting in Language Models. Preprint at <https://doi.org/10.48550/arXiv.2210.03629> (2023).
- Yenduri, G. et al.** GPT (Generative Pre-Trained Transformer)— A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions. *IEEE Access* 12, 54608–54649 (2024).
- Zhao, W. X. et al.** A Survey of Large Language Models. Preprint at <https://doi.org/10.48550/arXiv.2303.18223> (2024).
- 10-423/10-623 Generative AI Pat Virtue & Matt Gormley Lecture 15 Carnegie Mellon University



*Thank You!*