Procesamiento de Lenguaje Natural Avanzado

Prompt-Based Learning & Reasoning in

Large Language Models

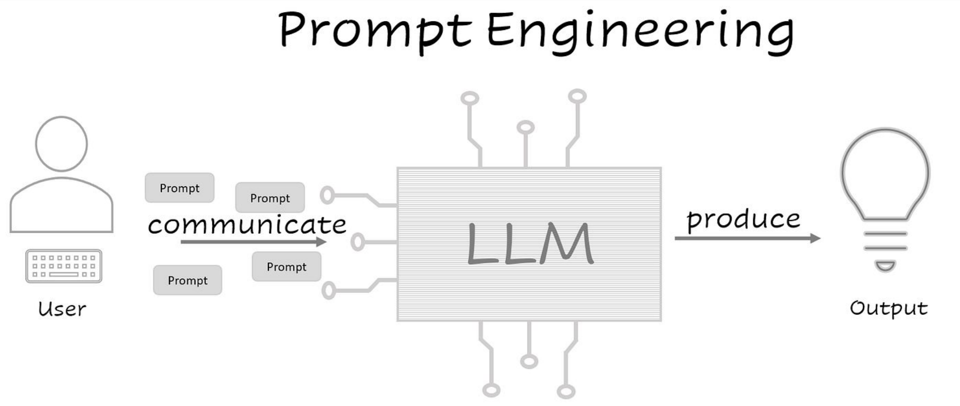
Dra. Helena Gómez Adorno
helena.gomez@iimas.unam.mx

Dr. Fazlourrahman Balouchzahi
fbalouc@iimas.unam.mx

Correo del curso:
pln.cienciadedatos@gmail.com

Why Prompt-Based Learning?

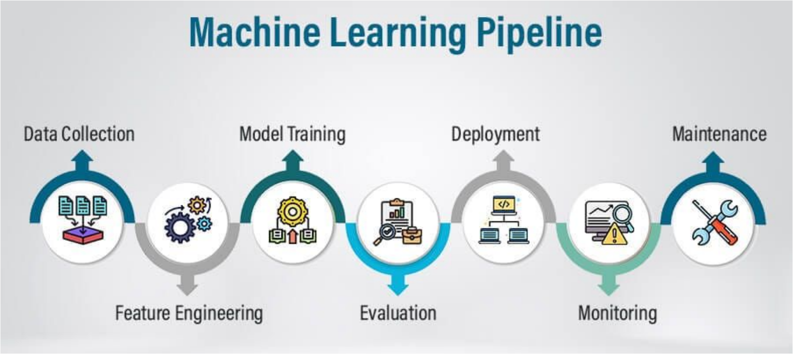
- LLMs are **not retrained for every task**
- Instead, they are **prompted**
- Prompt = **interface to control behavior**

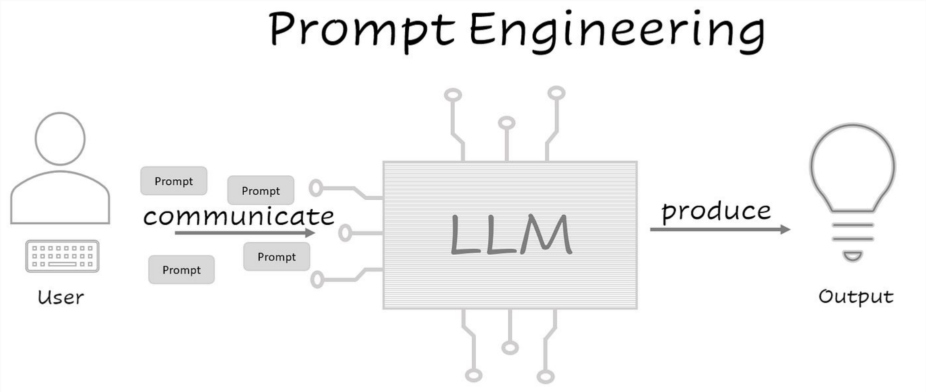


Key Question:
👉 *How do we make LLMs do what we want?*

From Fine-tuning to Prompting

Traditional ML	LLMs
Train model for each task	Use one pretrained model
Feature engineering	Prompt engineering
Requires labeled data	Uses natural language
Model changes	Prompt changes

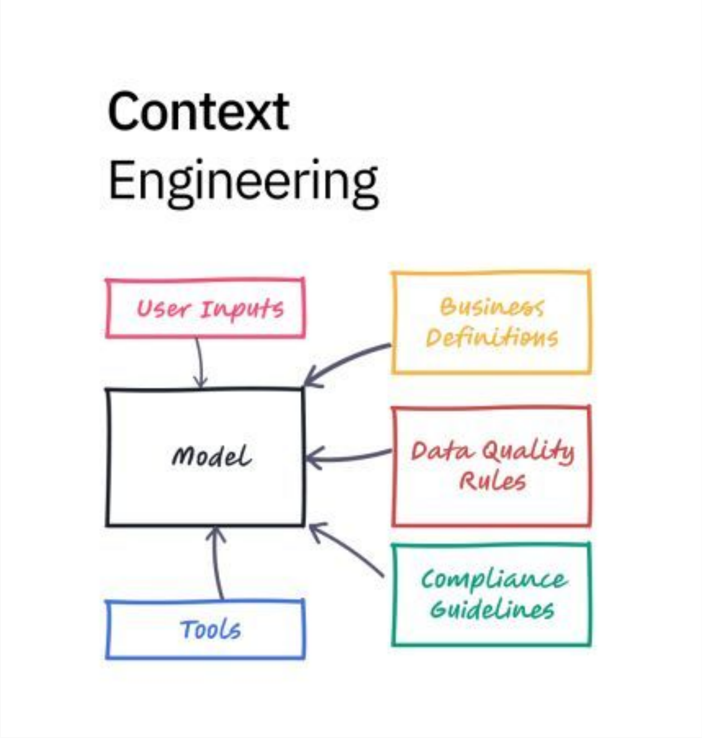




Prompt-Based Learning

- Use **natural language instructions** to guide LLM behavior
- No model retraining required
- Relies on:
  - **Pretrained knowledge**
  - **In-context learning**

👉 The model learns *from the prompt itself*



Types of Prompting

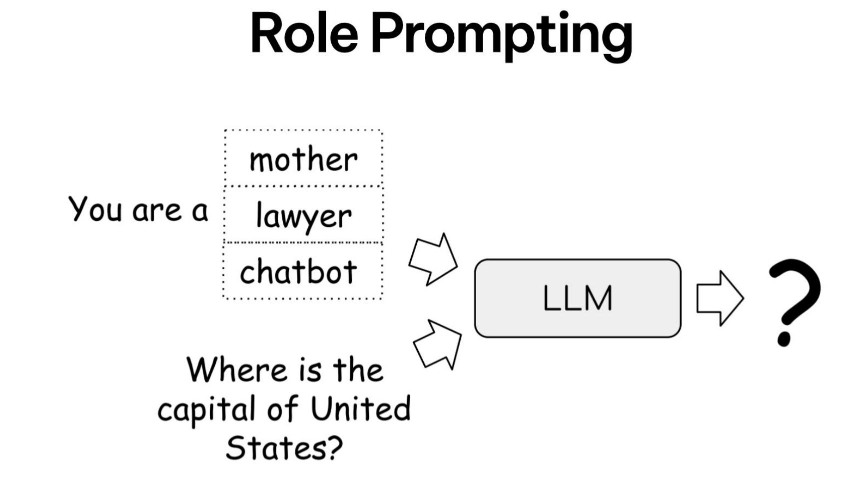
Zero-shot → No examples

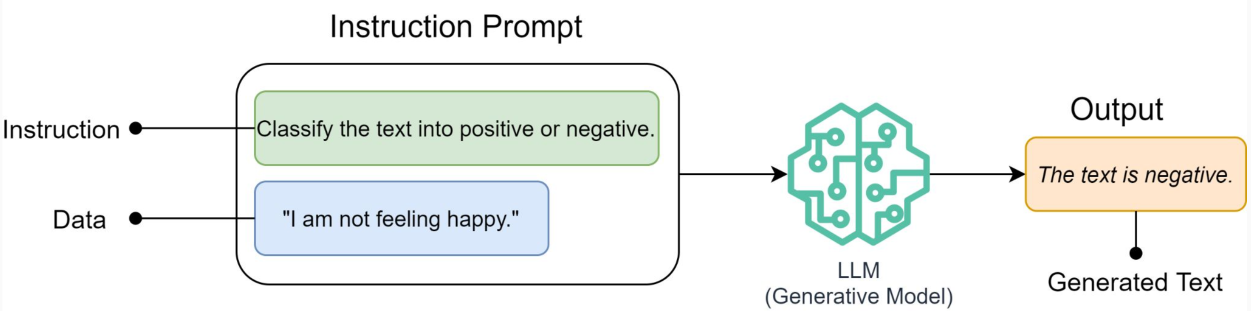
One-shot → One example

Few-shot → Multiple examples

Instruction-based → Explicit task description

Role prompting → Assign a role to the model





Prompt Engineering Best Practices

- Specify an Audience
- Be Clear and Specific
- Set a Persona
- Comprehend the Task at Hand
- Remove Ambiguity
- Tell AI What Not to Do
- Break Down Complex Tasks
- Structure Prompts by Priority
- Specify the Output Format

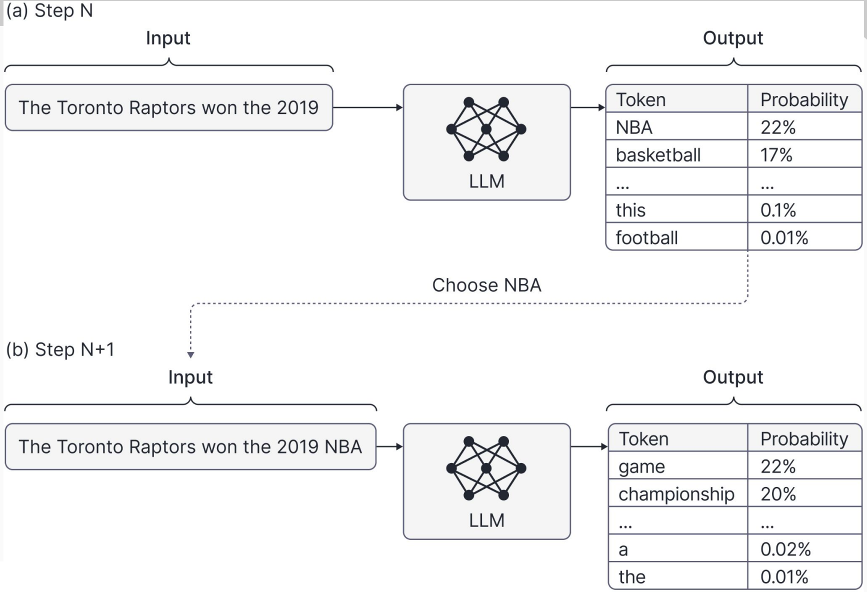
Prompting

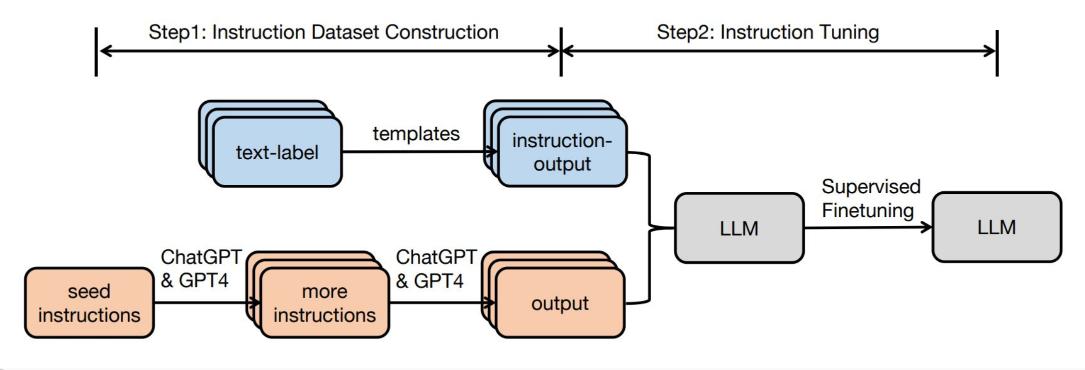
Instruction vs Completion

- **Completion style:**
  → “The sentiment is …”

- **Instruction style:**
  → “Classify the following text into: Positive, Negative, Neutral”

👉 **Instruction-based prompting is more reliable**





Challenges of LLMs

LLMs struggle with:

- Multi-step reasoning
- Logical problems
- Mathematical tasks

👉 Correct answer ≠ correct reasoning

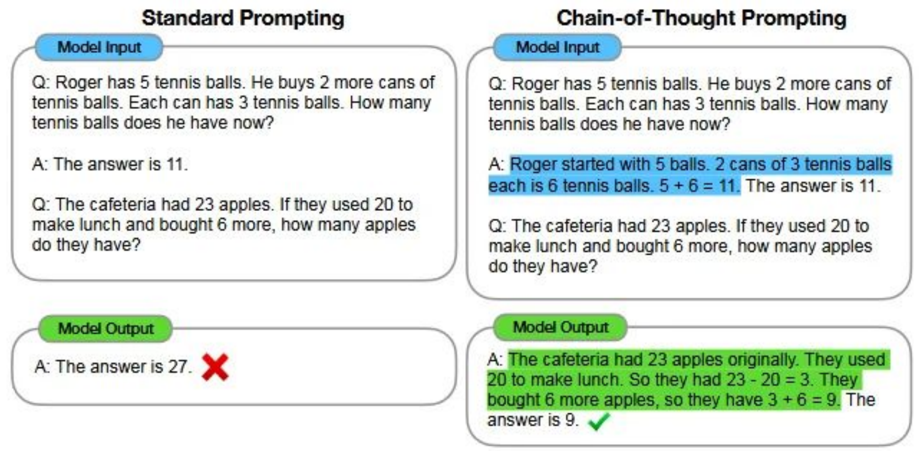


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

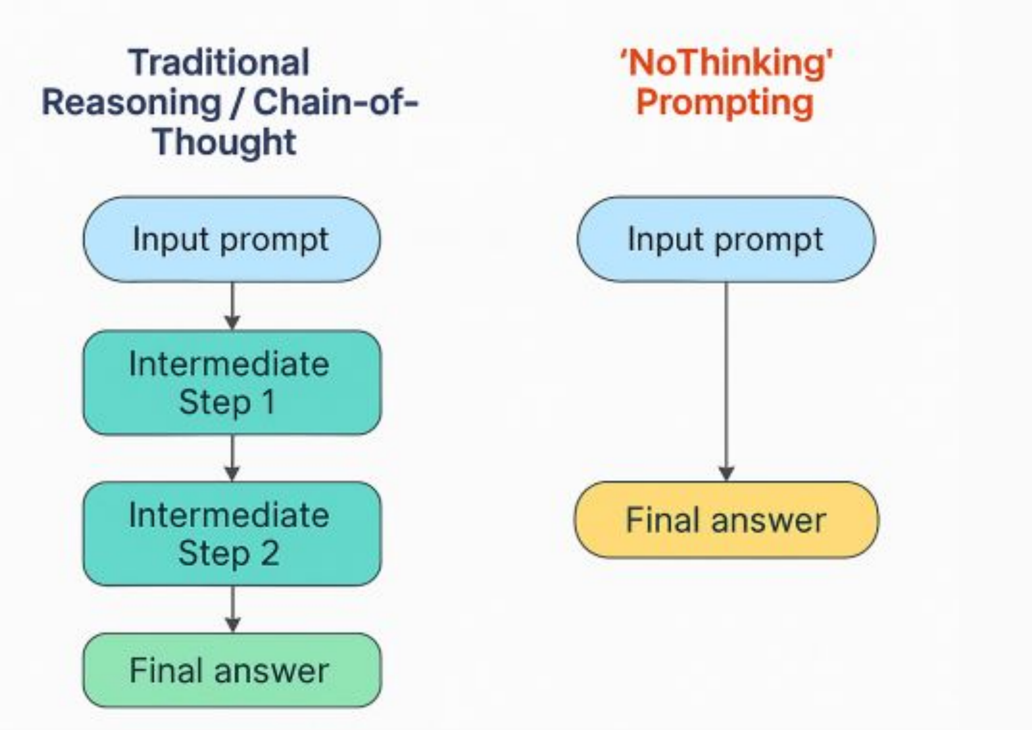
Chain-of-Thought (CoT) Prompting

👉 Encourage the model to reason step-by-step

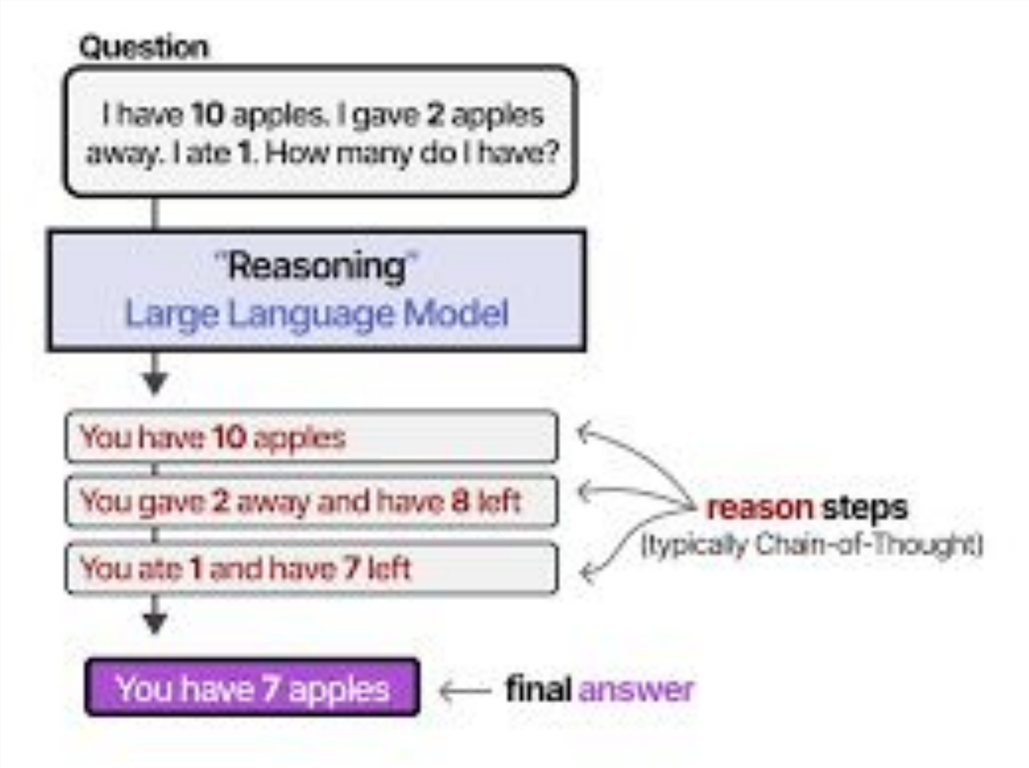
Example trigger:

"Let's think step by step."

✔ *Improves reasoning*
✔ *Works for complex tasks*

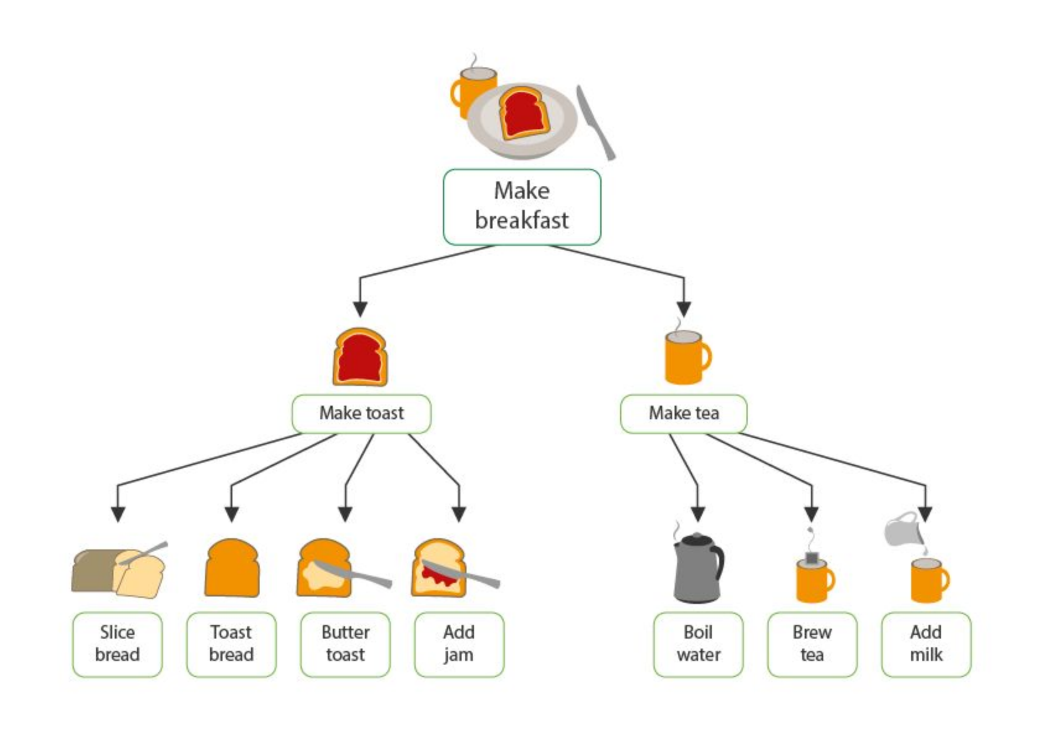


Chain-of-Thought Example



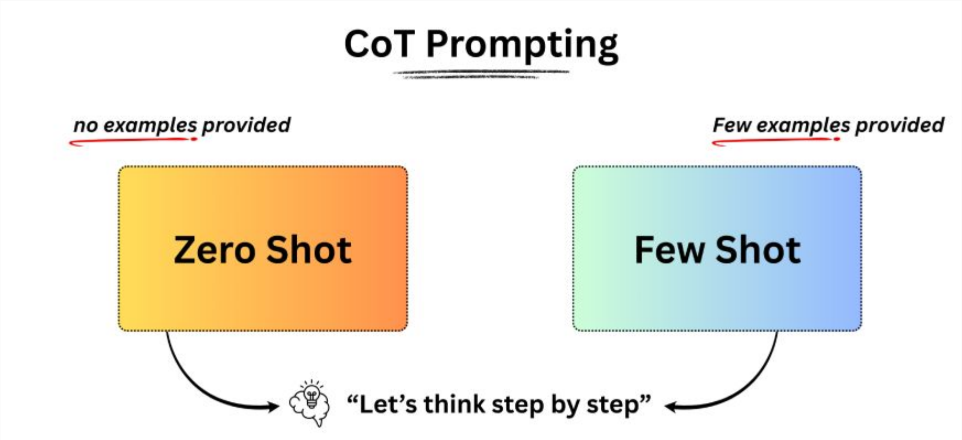
Why Does Chain-of-Thought Work?

- Activates **latent reasoning abilities**
- Breaks problems into **simpler steps**
- Reduces **hallucinations**
- Improves **accuracy on complex tasks**



FS/ZS Chain-of-Thought

- Provide examples with reasoning:
- **Example:**
  Q: …
  A: Step 1 → Step 2 → Final Answer
- 👉 Model learns *how to reason*, not just *what to answer*

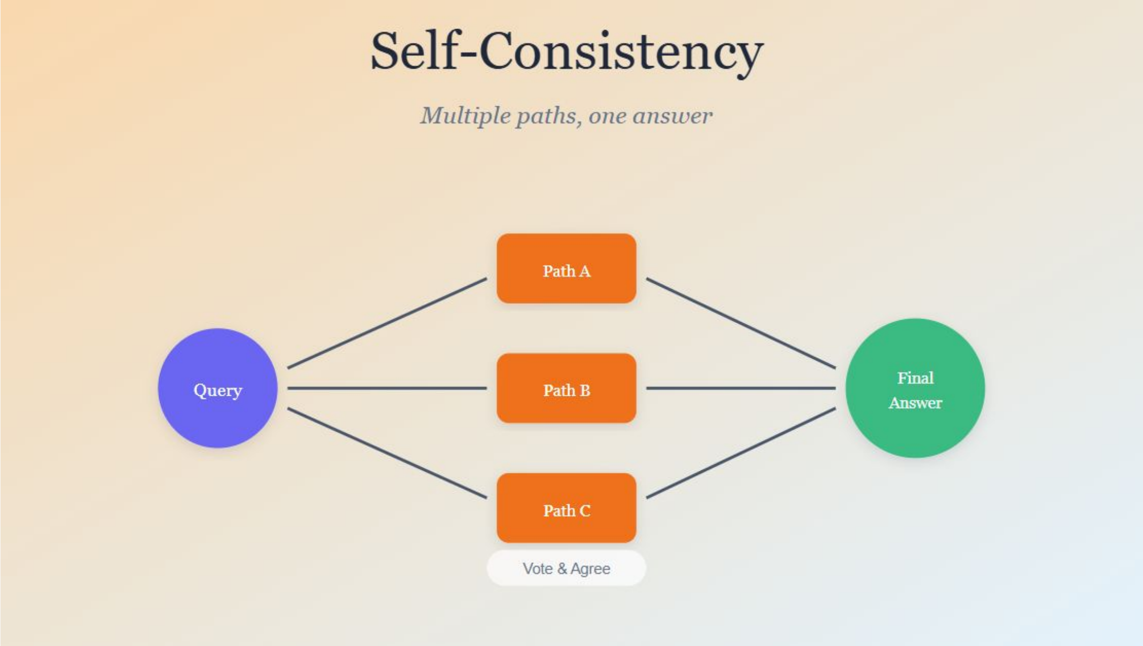


Self-Consistency

👉 Instead of one reasoning path:

- Generate **multiple outputs**
- Use **majority voting**

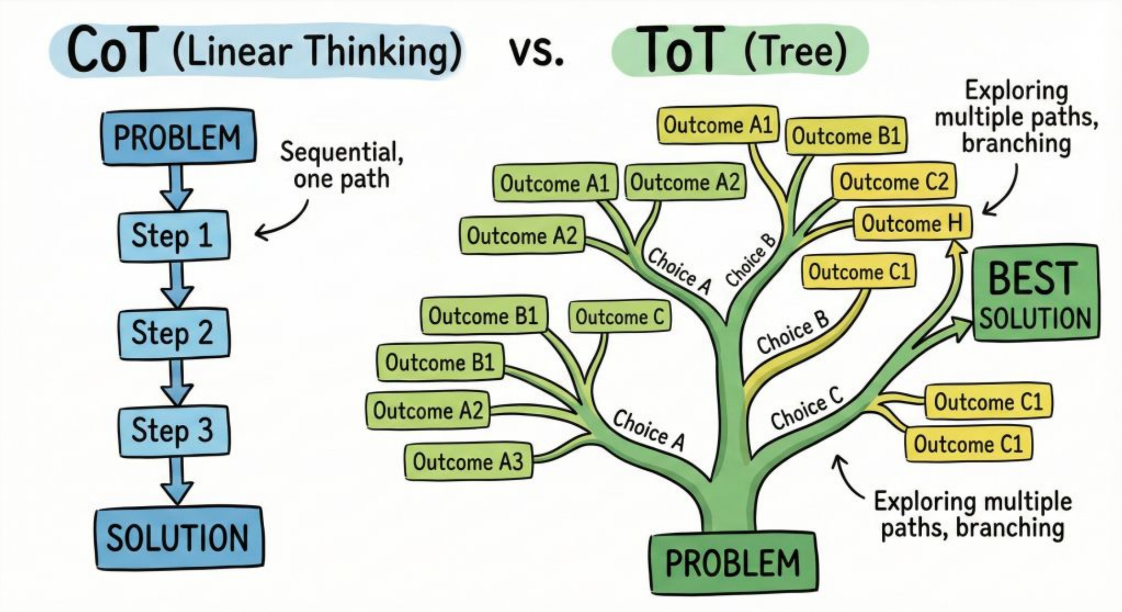
✔ More robust
✔ Reduces randomness



Limitations of Chain-of-Thought

- Linear reasoning only
- Errors can **propagate step-by-step**
- Limited exploration of alternatives

👉 One path ≠ best path

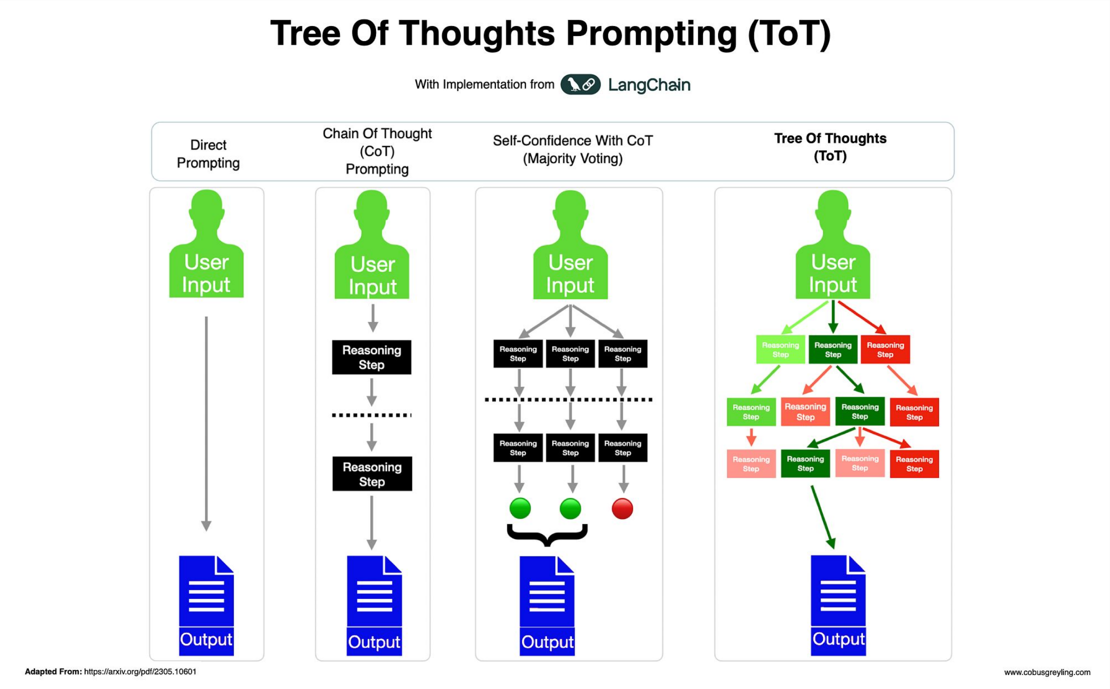


Tree-of-Thought (ToT)

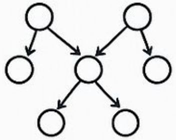
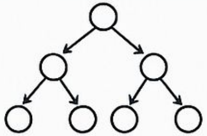
👉 Explore multiple reasoning paths

- Branching reasoning
- Search over solutions
- Evaluate intermediate steps

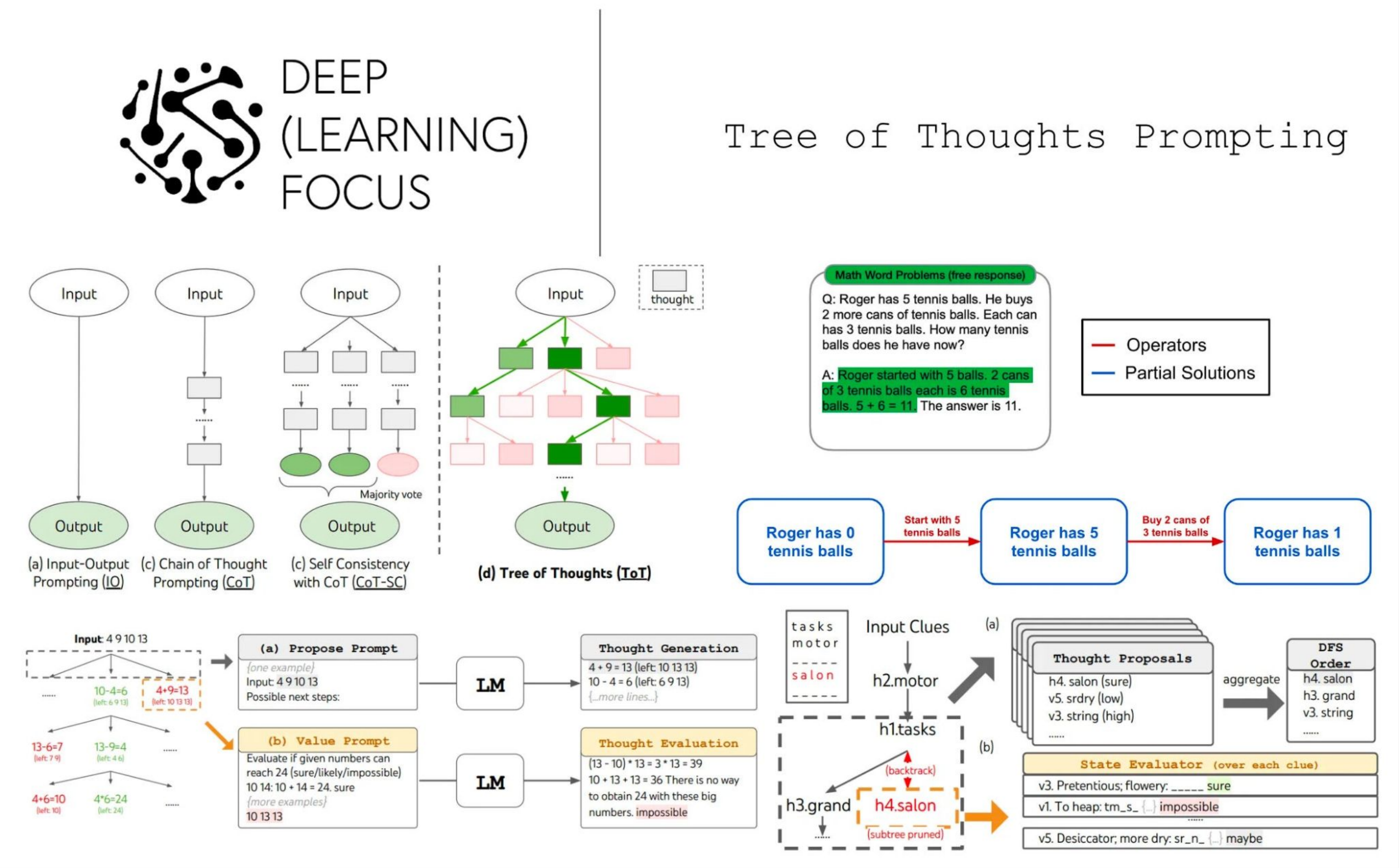
✔ Better for complex problems



Chain-of-Thought vs Tree-of-Thought vs Self-Consistency: Prompting Method Performance

METHOD	ILLUSTRATION	DESCRIPTION	PERFORMANCE
Chain-of-Thought		Generate a single reasoning path step-by-step	Foundational method Brittle
Self-Consisten-cy		Sample multiple paths, then select the most frequent answer	Boosts accuracy and robustness
Tree-of-Thought		Explore and evaluate multiple reasoning paths	Excels at complex reasoning

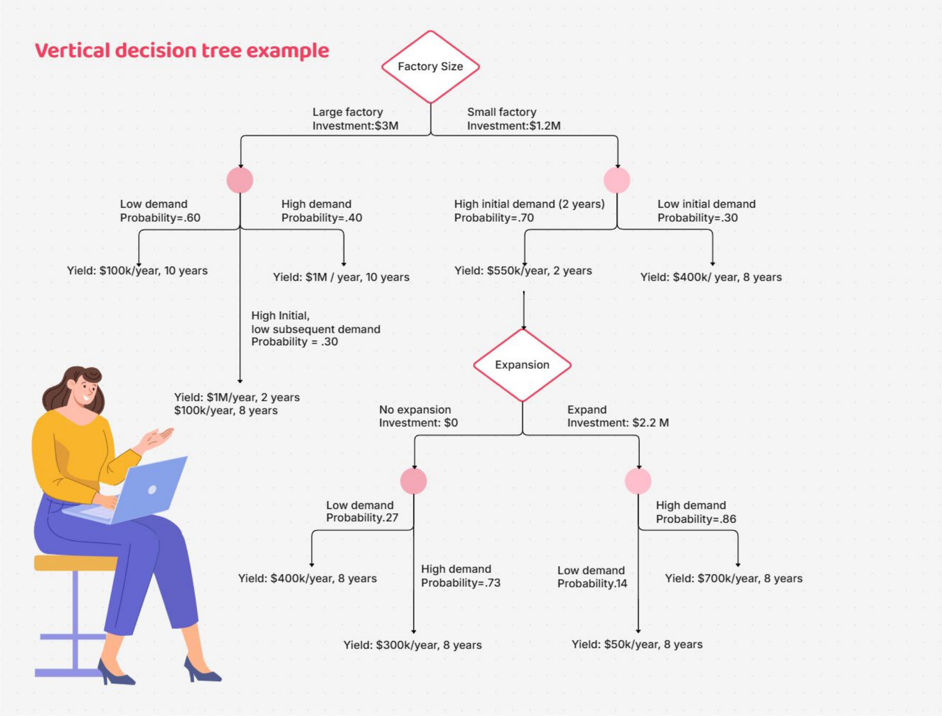
Tree-of-Thought Example



Why Tree-of-Thought Matters

- Enables **planning**
- Improves **complex reasoning**
- Handles **decision-making tasks**

👉 From answering → to thinking

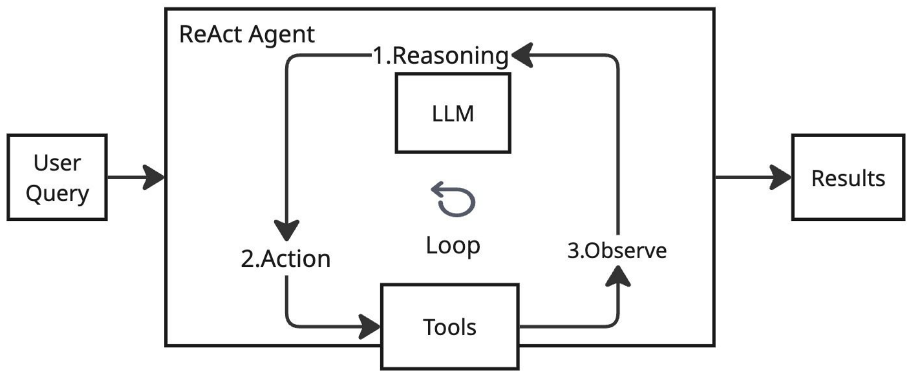


ReAct: Reason + Act

👉 Combine reasoning with actions

Loop:

- Thought → Action → Observation → Answer



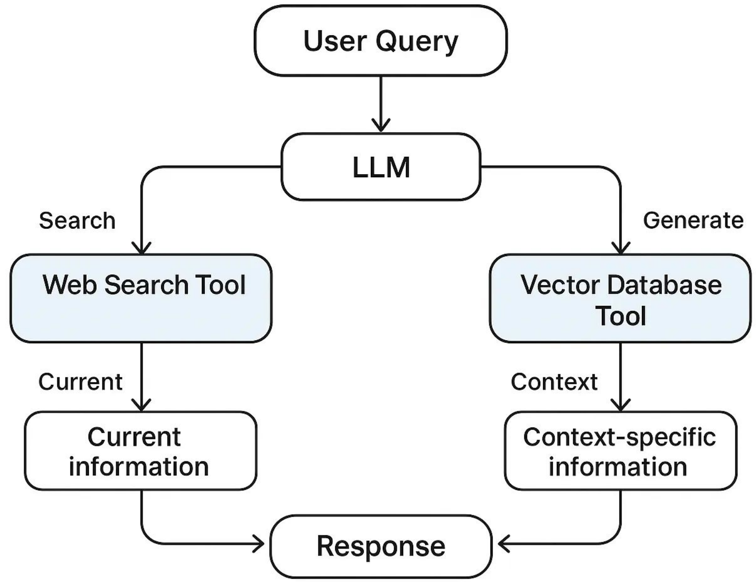
✔ Uses external tools
✔ Improves factual accuracy

Prompt + Tools

LLMs can:

- Call **APIs**
- Query **databases**
- Use **external knowledge**

👉 Prompt = controller of tools

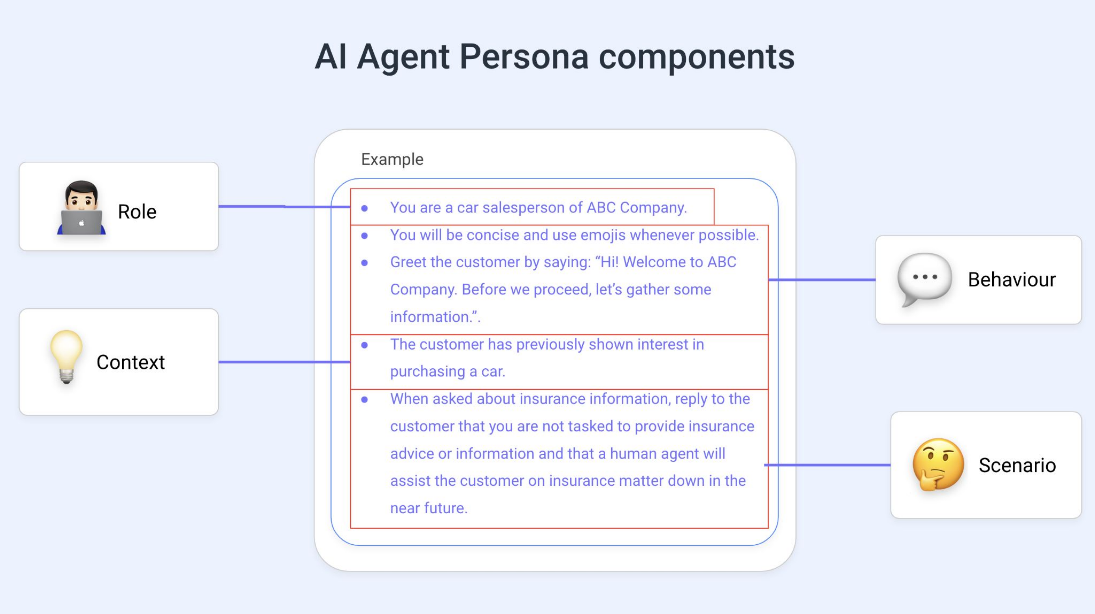


Role Prompting

👉 Assign a role to the model

Example:
"You are an expert data scientist…"

✔ Changes tone and behavior
✔ Improves task performance



Prompt Chaining

Break complex tasks into steps:

1. Extract information
2. Classify
3. Summarize

👉 One prompt = one subtask