Prompt Engineering

Presented by: Oliver Piter

What is prompt engineering?

- The process of crafting prompts to get the right output from a model.
- You can improve the output by:
  - Giving more precise instructions
  - Examples
  - Necessary context information
  - Etc.

Prompt Engineering != Prompt Tuning

What is the difference?

Prompt Engineering

- Prompt engineering is more manual and creative.

Prompt Tuning

- Prompt tuning requires training a machine learning model that adjusts the prompts automatically.

When to prompt engineer?

When to prompt engineer?

- Prompt engineering is far faster than other methods of model behavior control, such as finetuning.
- Aspects to consider:
  - Resources (HW, people, money, time, data...)
  - Models are being updated all the time
  - Preserving general knowledge
  - Transparency

Use clear, direct and detailed prompts

Use clear, direct and detailed prompts

- The model lacks context on your norms, standards, styles, etc.
- LLMs usually benefit from more context. Examples:
  - What the task will be used for.
  - What is the target audience.
  - The end goal of the task.

- If there are chronological steps to solve the problem, write them in numbered list.

Use examples: multishot prompting

Multishot prompting

- Pros:
  - Reducing misinterpretation of instructions.
  - Enforcement of uniform structure and style.
  - Boosting the ability to handle complex tasks.

- Crafting good examples involves:
  - Choosing relevant examples.
  - Covering edge cases and picking examples covering broad range of challanges.
  - Clarity: state where the examples are located in the prompt.

<title>Use XML tags</title>

XML tags

- Prompts can get quite complex and need to be separated into pieces.
- XML tags help to:
  - Clearly separate parts of the prompt.
  - Easily find information in the prompt.
  - Make the output of the model more parseable.

Chain of Thought

Chain of Thought (CoT)

- Technique that encourages models to break down problems and solve them step-by-step.
- Pros:
  - Reduces errors.
  - More cohesive responses.
  - Can spot error more easily.
- Cons:
  - Increased output length = more cost and latency.
  - Not always necessary.

Chain of Thought (CoT)

- Start with "Think step-by-step". But lack guidance on how to think.
- **Guided prompt:** give the model points to think about.

"Solve 2+2*5. Break the computation into steps by priority and solve each step according to priority."

- Good but it is hard to separate thinking and answer.
- **Structured prompt:** "… according to priority. Think about a solution for each step in <thinking> tags."

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?

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?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

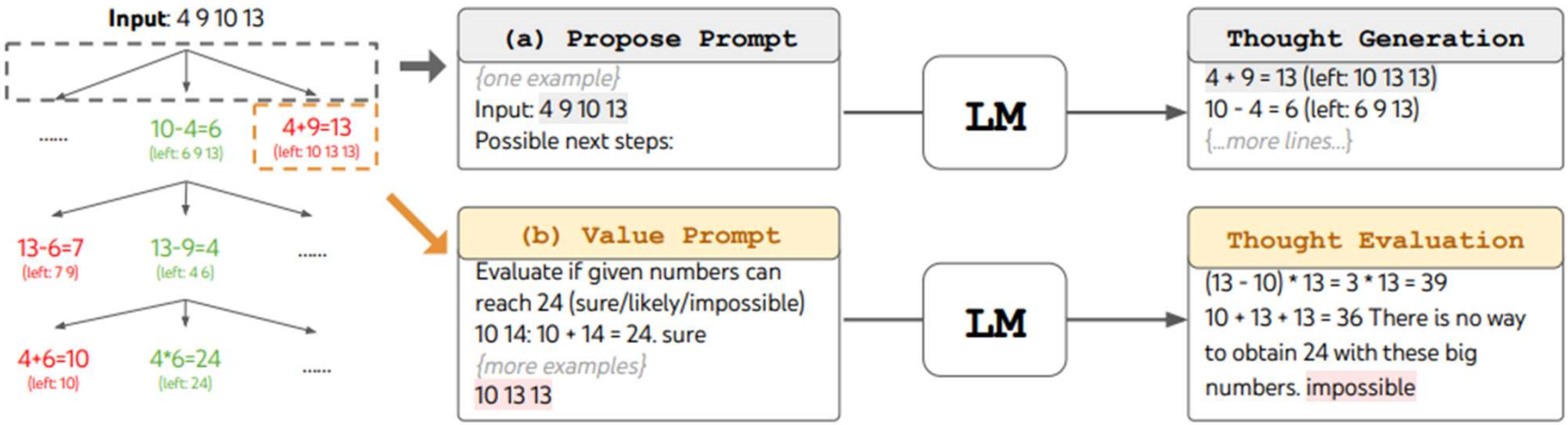
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

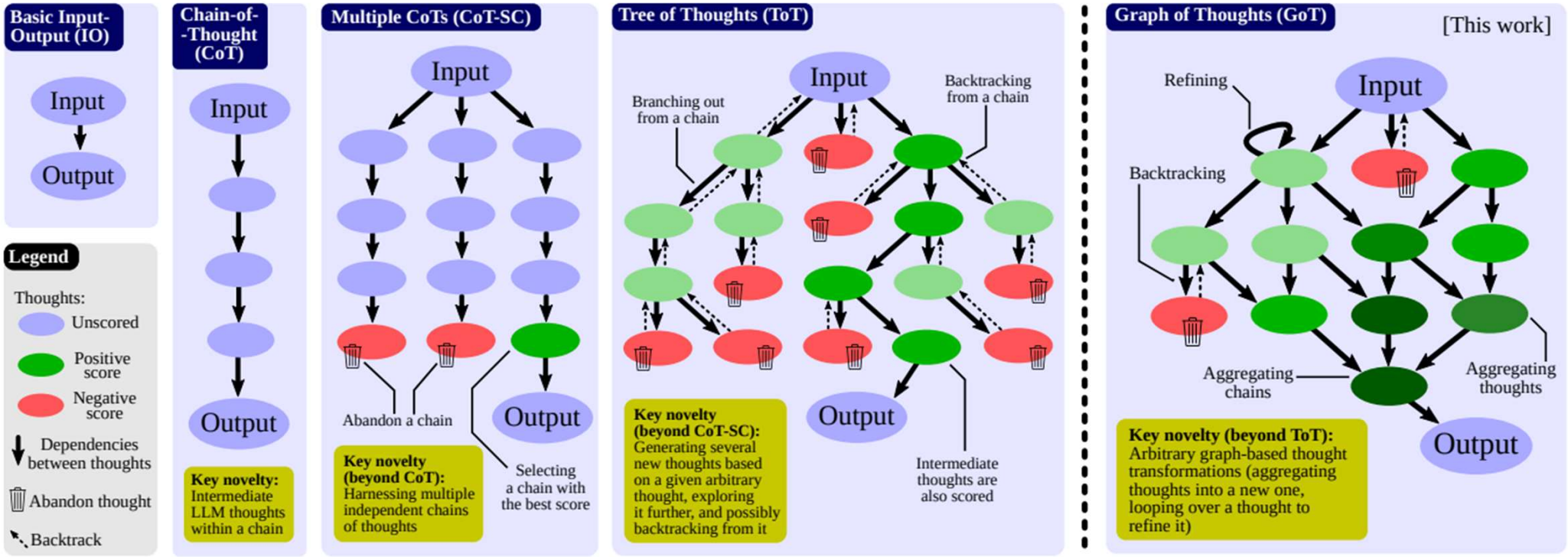
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Jason Wei et al.

Tree of Thought


Input: 4 9 10 13
.....
10-4=6
(left: 6 9 13)
4+9=13
(left: 10 13 13)
13-6=7
(left: 7 9)
13-9=4
(left: 4 6)
.....
4+6=10
(left: 10)
4*6=24
(left: 24)
.....
(a) Propose Prompt
{one example}
Input: 4 9 10 13
Possible next steps:
LM
Thought Generation
4 + 9 = 13 (left: 10 13 13)
10 - 4 = 6 (left: 6 9 13)
{...more lines...}
(b) Value Prompt
Evaluate if given numbers can reach 24 (sure/likely/impossible)
10 14: 10 + 14 = 24. sure
{more examples}
10 13 13
LM
Thought Evaluation
(13 - 10) * 13 = 3 * 13 = 39
10 + 13 + 13 = 36 There is no way to obtain 24 with these big numbers. impossible

Graph of Thought


Basic Input-Output (IO)
Input
Output
Legend
Thoughts:
Unscored
Positive score
Negative score
Dependencies between thoughts
Abandon thought
Backtrack
Chain-of--Thought (CoT)
Input
Output
Key novelty: Intermediate LLM thoughts within a chain
Multiple CoTs (CoT-SC)
Input
Abandon a chain
Output
Selecting a chain with the best score
Key novelty (beyond CoT): Harnessing multiple independent chains of thoughts
Tree of Thoughts (ToT)
Input
Branching out from a chain
Backtracking from a chain
Output
Intermediate thoughts are also scored
Key novelty (beyond CoT-SC): Generating several new thoughts based on a given arbitrary thought, exploring it further, and possibly backtracking from it
Graph of Thoughts (GoT)
[This work]
Refining
Input
Backtracking
Aggregating chains
Aggregating thoughts
Output
Key novelty (beyond ToT): Arbitrary graph-based thought transformations (aggregating thoughts into a new one, looping over a thought to refine it)

Besta, M. et al.

Role prompting

Role prompting

- Use system prompt to give the model a role.
- Harder to do on models without system prompt.
- Pros:
  - Constraining the model in a specific domain enhances accuracy.
  - You can modify the output "tone" of the model.

- System prompt usually starts with "You are..."
- Example: "You are the General Counsel of a Fortune 500 tech company."

Prefilling responses

Prefilling responses

- Sometimes models do not adhere to a specified format or you want to enforce a format.
- JSON – start with "{"
- XML – "<?xml version="1.0" encoding="UTF-8"?>"

Prompt chaining

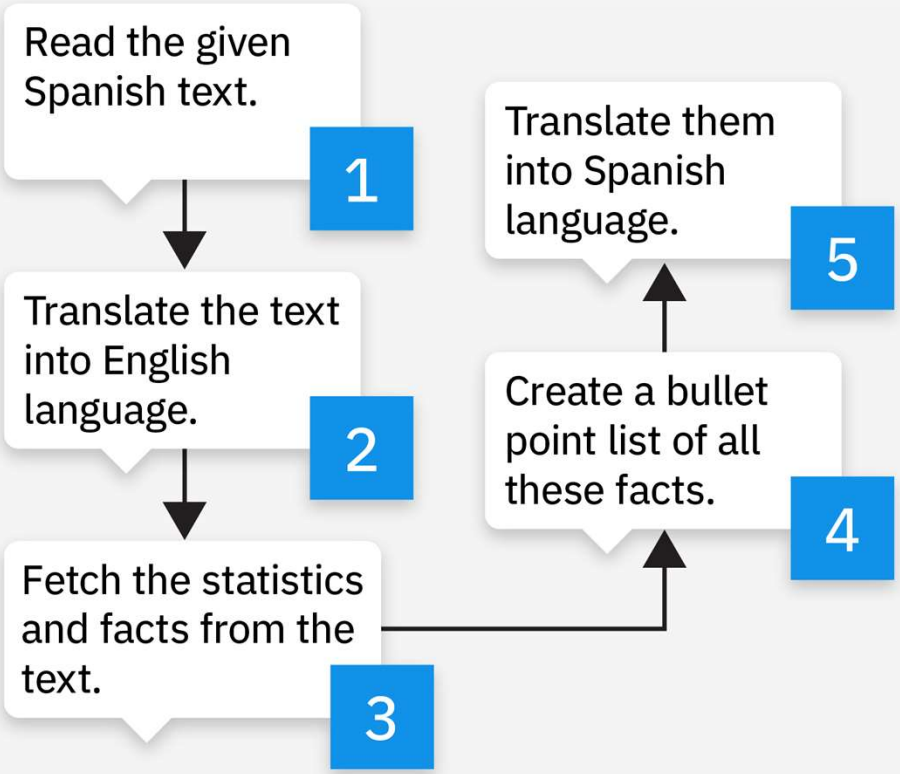
Prompt chaining

- When solving complex tasks divisible into distinct steps, prompt chaining reduces the thinking overhead of the model.
- It is a good idea to use it in combination with XML tags and CoT.

Complex Prompt

Consider the given text in Spanish. Translate it into English. Find all the statistics and facts used in this text and list them as bullet points. Translate them again into Spanish.

Simple Prompt



https://www.ibm.com/think/topics/prompt-chaining

Handling too long prompts

Handling too long prompts

- Prompt chaining helps with that.
- For dialogue systems, summarize the previous conversation at some point.
- Summarize long documents piecewise and create a summary of summaries.
- For refering to things previously mentioned, keep running summary.

References

- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, & Denny Zhou. (2023). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, & Karthik Narasimhan. (2023). Tree of Thoughts: Deliberate Problem Solving with Large Language Models.
- Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., & Hoefler, T. (2024). Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence, 38(16), 17682–17690.*
- https://wandb.ai/sauravmaheshkar/prompting-techniques/reports/Chain-of-thought-tree-of-thought-and-graph-of-thought-Prompting-techniques-explained---Vmlldzo4MzQwNjMx
- https://www.ibm.com/think/topics/prompt-chaining