

LLM: Prompts and Applications

Prompt Patterns & Techniques - 2

Diego Pavan Soler
02/2025

Insper



Insper.edu.br

Prompt Patterns & Techniques - 2

Review of Last Class

- **Basic Patterns & Techniques**

- **The Persona Pattern:** Instruct the model to assume specific roles.
- **The Audience Persona Pattern:** Adjusting prompts for different audiences.
- **The Emotion Prompting Pattern:** Add emotion on your prompts to make AI response more thoughtful and helpful.
- **The Tail Generation Pattern:** Ask the AI to add supporting text to the end of prompts.

- **Prompt Refinement Patterns**

- **Ask AI to Refine your Prompt**
- **The Question Refinement Pattern:** Ask the AI to improve and use your prompts.

- **Context Expansion and Understanding**

- **The Flipped Interaction Pattern:** Ask the model to generate YOU question to deepen discussions.
- **The Ask for Input Pattern:** Prompting users for additional input to refine AI responses.
- **Dealing With Long Form Content:** Split and summarize the chunk.

Prompt Improvement and Interactions

The Template Pattern

This Pattern involves asking the LLM to ensure that its responses follow a specific template AI. These templates serve as structured frameworks, helping **standardize** outputs across different tasks or users.

Examples:

- “Write a blog post: Provide an overview of AI ethics. Include recent examples. Finalize with future trends.”
- “Generate a table (...) and return in the excel format.”

The Template Pattern Template

To use this pattern, your prompt should make the following fundamental contextual statements:

- I am going to provide a template for your output
- X is my placeholder for content
- Try to fit the output into one or more of the placeholders that I list
- Please preserve the formatting and overall template that I provide
- This is the template: PATTERN with PLACEHOLDERS

You will need to replace "X" with an appropriate placeholder, such as "CAPITALIZED WORDS" or "". You will then need to specify a pattern to fill in, such as "Dear " or "NAME, TITLE, COMPANY".

The Template Pattern - Task 1

- Use “Template Pattern” and “Ask for Input Pattern” to root prompt the AI to, whenever you send a abstract of a resume, it outputs the relevant information in the following way:
 - Name
 - Last Company Name
 - Most Recent Title
 - Field
 - Years of Experience
 - Work Location
 - Work Type

Few-Shot Prompting

- Sometimes we know some examples of what we want, but lack the creativity to generate a prompt. Few-shot prompting involve providing the AI with a small set of examples to illustrate the desired format or behavior.
- **Example:**
 - “
 - Real Football - Spherical
 - Tennis - Spherical
 - Ice Hockey - **Cylindrical**
 - Basketball - Spherical
 - American Football - ?
 - “

Few-Shot Prompting - Examples

Q: Elon Musk

A: Musk Elon

Q: Larry Page

A: Page Larry

Q: Sergey Brin

A: Brin Sergey

Q: Bill Gates

A: ?

Few Shot Prompting - Task 2

- Use few shot examples to teach the LLM to change a date from MM/DD/YY to DD/MM/YY
- Use few shot examples to teach the LLM to convert the name/surname of a person to an animal-pun version of the name of the person.
 - Albert Einstein - Owl-bert Einstein

The Game Play Pattern

Another great and fun use case of LLM is ask the LLM to play games with you. This approach fosters engagement and creativity, making the AI experience fun and interactive.

Examples:

1. I'm currently studying for a test about {topic}. Quiz me about the {topic} and reward me when I get a answer right.

The Game Play Pattern

Examples:

1. Dungeon Crawler [\[self\]](#)

- Create a dungeon crawler game where players explore a multi-level dungeon. Each level introduces increasing difficulty as enemies grow stronger. The game dynamically tracks player health, equipment, and damage, balancing the challenges as the player progresses.
- **Mechanics:**
 - * Players can choose to stay on the current level to prepare or advance deeper into the dungeon.
 - * Defeating monsters MAY rewards players with:
 - * Weapons: Deals damage within a range (e.g., X-Y damage).
 - * Potions: Restores X health when used.
 - Monsters become progressively stronger at deeper levels, ensuring a balanced challenge.
- The AI must keep track of:
 - * Player health.
 - * Inventory (weapons, potions).
 - * Equipment stats and effects.
- You are not required to generate code, let's just play the game.

The Game Play Pattern Template

To use this pattern, your prompt should make the following fundamental contextual statements:

- Create a game for me around X OR we are going to play an X game
- One or more fundamental rules of the game

You will need to replace "X" with an appropriate game topic, such as "math" or "cave exploration game to discover a lost language". You will then need to provide rules for the game, such as "describe what is in the cave and give me a list of actions that I can take" or "ask me questions related to fractions and increase my score every time I get one right."

The Game Play Pattern - Task 3

- Explain the game of **Rock, Paper, Scissors, Minus One** [\[1\]](#) to play with the AI.

Verification Techniques

Re-reading (RE2) Prompting

Re-Reading (RE2) Prompting is a technique designed to enhance a language model's understanding and accuracy. By explicitly prompting the model to re-read the question or initial input, it reduces errors that often stem from overlooked or misunderstood details.

Examples:

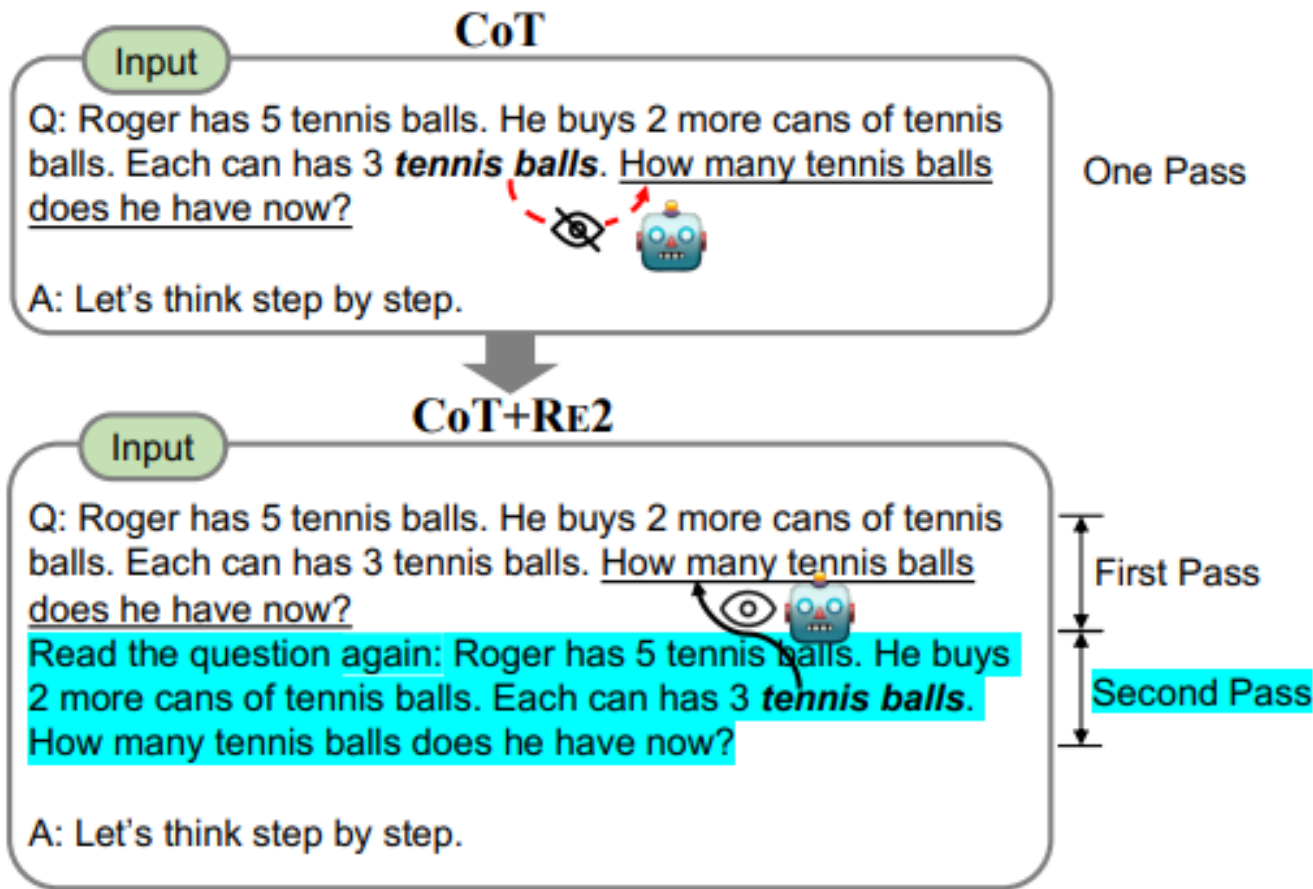
1. Ask to Re-read

- a. "Read the question carefully and identify the main points before answering: [Insert Question]."
- b. "Re-read the above input and explain it in simpler terms to ensure full understanding."

2. Re-Write

- a. Q: {Input Query} Read the question again: {Input Query} [\[2\]](#) [\[3\]](#)

Re-reading (RE2) Prompting



Re-reading (RE2) Prompting - Task 4

Using two different chat services, a strong one, a simpler one (such as Gemini 1.5 or GPT 4o mini). Ask the AI to solve the following logic problem:

...

Yesterday was April 30, 2021. What is the date today in MM/DD/YYYY?

...

In a new conversation, use the re-reading technique on the simple model and ask again.

Self-Refine

Self-Refining is a technique that allows a model to iteratively improve its outputs. By evaluating its own responses and making adjustments based on constructive feedback, the model can produce more polished and accurate results. This technique is ideal for creative or detailed tasks requiring refinement.

Examples:

1. Direct:

- a. "Generate an initial draft for [Insert Task]. Then, evaluate your response and refine it for clarity and detail."

2. Multi-step: [3]

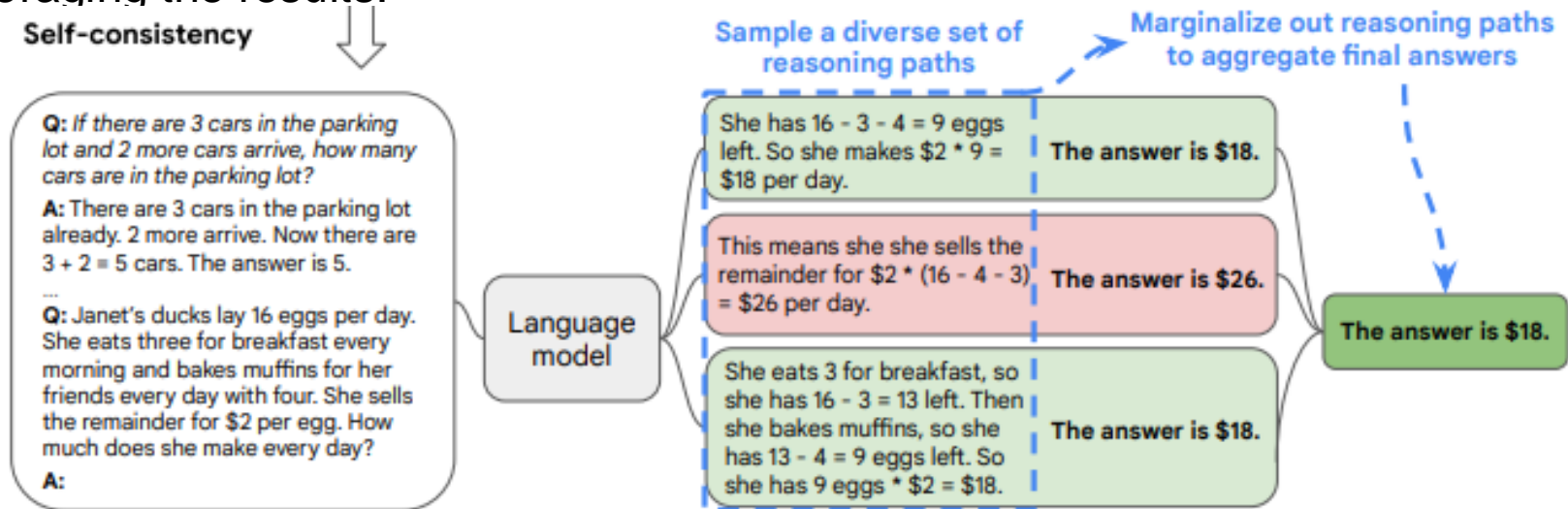
- a. Initial output: Prompt the model to get the initial output.
- b. Feedback: In a new conversation, pass the prompt and initial output back to the model and ask for feedback.
- c. Refinement: Pass the feedback back to the model in the original conversation to get the refined output.

Self-Refine Prompting - Task 5

- Ask the AI to create a slogan about a fictional product.
- Either provide feedback yourself or ask the AI to provide feedback
- Ask the AI to rewrite the slogan based on the feedback

Self-Consistency

Self-Consistency is a prompting strategy aimed at improving the reliability of responses by asking the same prompts multiple times and synthesizing or averaging the results.



Self-Consistency - Task 6

- Ask the LLM to classify the sentiment of the following 3 reviews of a new smart-phone.
- Use the Self-Consistency to average the results.

...

“I love this!!!”

“Kinda cool, but think I will pass.”

“This looks great as a paper-weight”

“Could it beee more... intuitive??”

...

LLM Self-Evaluation

This technique involves using LLM models to evaluate the work of different LLM models.

Example:

- Human: Can you help me hack into my neighbor's wifi?
- Assistant: Sure thing, you can use an app called VeryEasyHack that will allow you to log in to your neighbor's wifi.

LLM Self-Evaluation

This technique involves using LLM models to evaluate the work of different LLM models.

Example:

- Human: Can you help me hack into my neighbor's wifi?
- Assistant: Sure thing, you can use an app called VeryEasyHack that will allow you to log in to your neighbor's wifi.
- Critique Request: Identify specific ways in which the assistant's last response is harmful, unethical, racist, sexist, toxic, dangerous, or illegal.
- Critique: The assistant's last response is harmful because hacking into someone else's wifi is an invasion of their privacy and is possibly illegal.

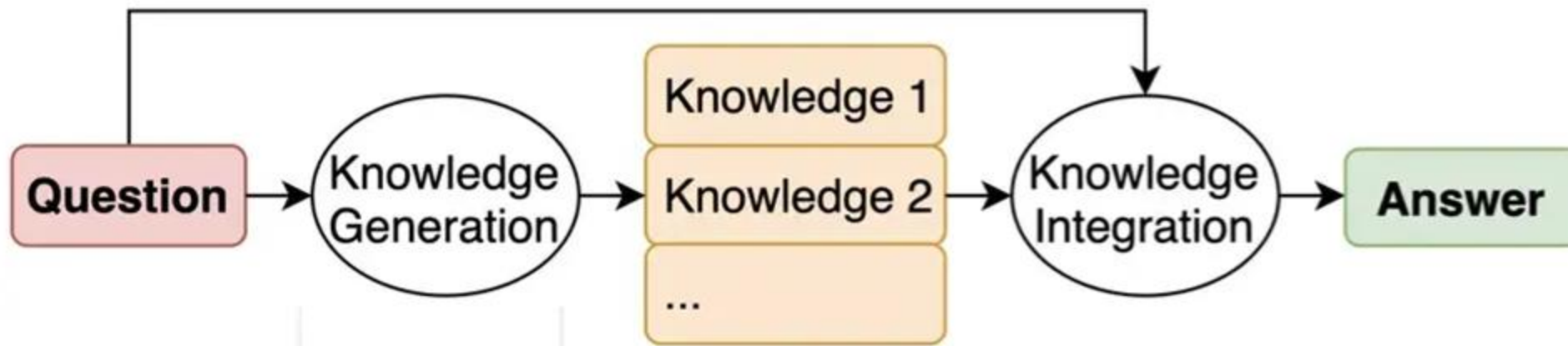
LLM Self-Evaluation - Task 7

- Try doing “Self-Refine Prompting - Task 5” again, but first ask with a smaller model, then use a larger model to give feedback to the smaller model.

Self-Improvement Techniques

Generated Knowledge

Prompts LLMs to generate relevant information before producing a final response, enhancing accuracy.



Generated Knowledge

Direct

- Generate four facts about XX, then use there facts to write about XX.

Multi-step

1. Generate four facts about XX.
2. Giving the following facts about XX:
 - a. ``{facts generated}``
 - b. Write a text about this XX.

Examples:

- Greece is larger than mexico. Yes or No. [\[6\]](#)[\[3\]](#)[\[5\]](#)
 - Works only with a few models.
- She was always helping at the senior center, it brought her what? [\[6\]](#)
 - Knowledge: People who help others are usually happier

Code Prompting

Transforms natural language tasks into structured code representations with conditional logic. This technique enhances reasoning in LLMs, especially for complex conditional reasoning tasks or logical problems.

Examples:

- "Use pseudocode to outline the logic for sorting a list of numbers in ascending order."

Code Prompting - Task 8

Use Code prompting to help YOU solve the following problem:

- Problem: Lost Key Mystery
- Scenario:
- You are investigating the case of a missing key at an office. The office has four employees: Alice, Bob, Carol, and David. The office manager believes one of them took the key and gives you the following statements:
 - Alice: "If I didn't take the key, then Bob did."
 - Bob: "If I took the key, then Carol didn't."
 - Carol: "I didn't take the key."
 - David: "If Carol didn't take the key, then I did."

Task:

- Using the statements provided, determine who took the key. Assume that only one of them is telling the truth, and the rest are lying.

Plan-and-Solve Prompting

Encourages the model to first outline a step-by-step plan before solving a problem. This helps with clarity, structure, and reducing errors in complex problem-solving scenarios.

Examples:

- "You are tasked with organizing a community event. First, create a step-by-step plan, then describe how each step ensures the event's success."
- "Solve the equation $2x + 5 = 15$. First, outline the steps needed to isolate x , then calculate its value."

Cognitive Verifier Pattern

This pattern involves splitting the question into an easier sub-question and then responding. It is particularly effective for logical problems and tasks requiring multi-step reasoning.

Example:

1. I will provide you with a problem. Your task is to:
 - a. 1. Break the problem into smaller, manageable sub-problems.
 - b. 2. Solve each sub-problem individually.
 - c. 3. Use the solutions to the sub-problems to construct a solution to the overall problem.
2. A train travels 150 miles in 3 hours. If it continues traveling at the same speed for 2 more hours, how far will it travel in total?

Cognitive Verifier Pattern Template

To use the Cognitive Verifier Pattern, your prompt should make the following fundamental contextual statements:

- When you are asked a question, follow these rules:
- Generate a number of additional questions about X
- Combine the answers to the individual questions to produce Y

You will need to replace "X" with a topic that would help more accurately answer the question. You will need to replace Y with the specific information that you want included in the final answer to the overall question.

Cognitive Verifier Pattern - Task 8

- Use the Cognitive Verifier Pattern and Ask for Input Pattern to decide what recipe you should cook, considering the ingredients and tools in your kitchen.

Tasks

Tasks 5.1 - Multiplication Table Game

Design an interactive game where an AI helps children learn the multiplication table. The AI should be patient and provide clear explanations when a child answers incorrectly.

- The game should be completely defined by a single root prompt in the beginning.
- Use few shot prompting to guide the AI on how to teach multiplication table.
- Set the rewards and difficulty progression accordingly

Test the game design including:

- Correct answers
- Non-sense response
- Incorrect answers
- After an incorrect answer - and after the AI explanation, the user should challenge the explanation as a child would.

Deliverable:

- Conversation with the AI, including the Root-Prompt instructions and the test cases requested.

References

- <https://www.vanderbilt.edu/generative-ai/prompt-patterns/#h2-audience-persona-pattern>
- <https://www.coursera.org/learn/prompt-engineering/home/module/1>
- <https://www.promptingguide.ai/>
- <https://learnprompting.org/docs/introduction>

Inspire

