取代提示工程師的 AI 代理

7 min read · Oct 24, 2024

Adam BEN KHALIFA Follow

Let’s build a multi-agent system that automates the prompt engineering process and transforms simple input prompts into advanced ones,

讓我們建立一個多代理系統，自動化提示工程流程，並將簡單的輸入提示轉換為進階提示，

aka. An Advanced Prompt Generator!

也就是說，一個進階提示產生器！

We will see the idea behind this solution, the first iteration, the second iteration, and the testing, troubleshooting, and optimizations involved.

我們將探討此解決方案背後的理念、第一次迭代、第二次迭代，以及其中涉及的測試、故障排除和最佳化。

Eventually, we’ll have a robust prompt enhancer tool, and I’ll share links to the GitHub repository for this project and the Hugging Face Space for a quick demo of the agent at the end of this article.

最終，我們將擁有一個強大的提示增強工具，我將在本文末尾分享該專案的 GitHub 儲存庫連結，以及 Hugging Face Space 的代理快速演示。

Table of Content 目錄

The idea 這個想法

After exploring various techniques for enhancing prompts, one idea clicked and stood out as particularly intuitive:

在探索了各種增強提示的技術之後，一個想法突然浮現，並且顯得特別直觀：

Utilizing a large language model (LLM) and guiding it with prompt engineering strategies and techniques to automate and generalize the process.

利用大型語言模型（LLM），並以提示工程策略和技術引導它，以自動化和通用化這個過程。

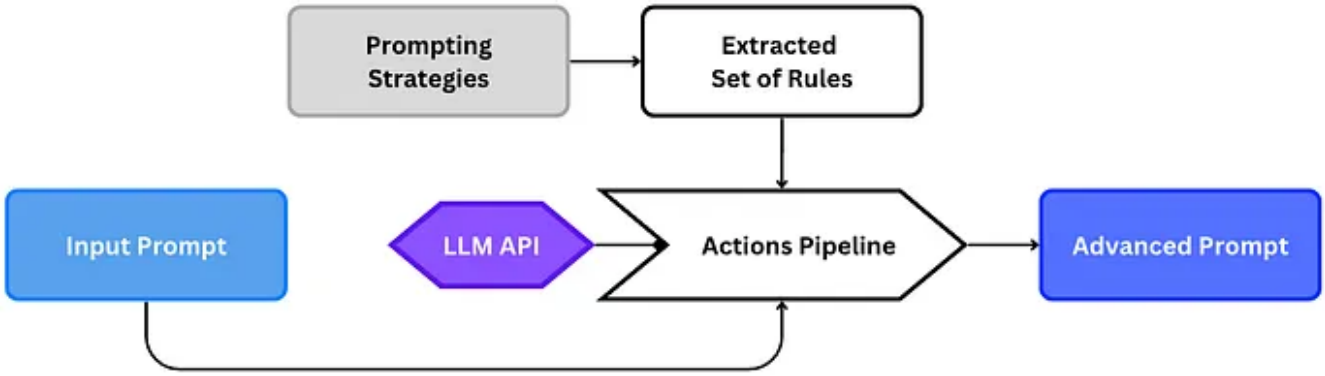
Here are the steps that were followed:

以下是所遵循的步驟：

- Use OpenAI’s prompting strategies to define a set of rules that ensure the creation of an advanced prompt.
  使用 OpenAI 的提示策略來定義一組規則，以確保建立進階提示。
- Define multiple agents to generate results based on the input prompt and the defined rules.
  定義多個代理程式，根據輸入提示和定義的規則產生結果。
- Create a pipeline to define the flow of interactions between the multiple agents
  建立一個管道來定義多個代理程式之間的互動流程。

Here’s a flowchart to better visualize these steps:

以下是流程圖，可讓您更清楚地了解這些步驟：



The described solution's flowchart

所描述解決方案的流程圖

Modeling and building the solution

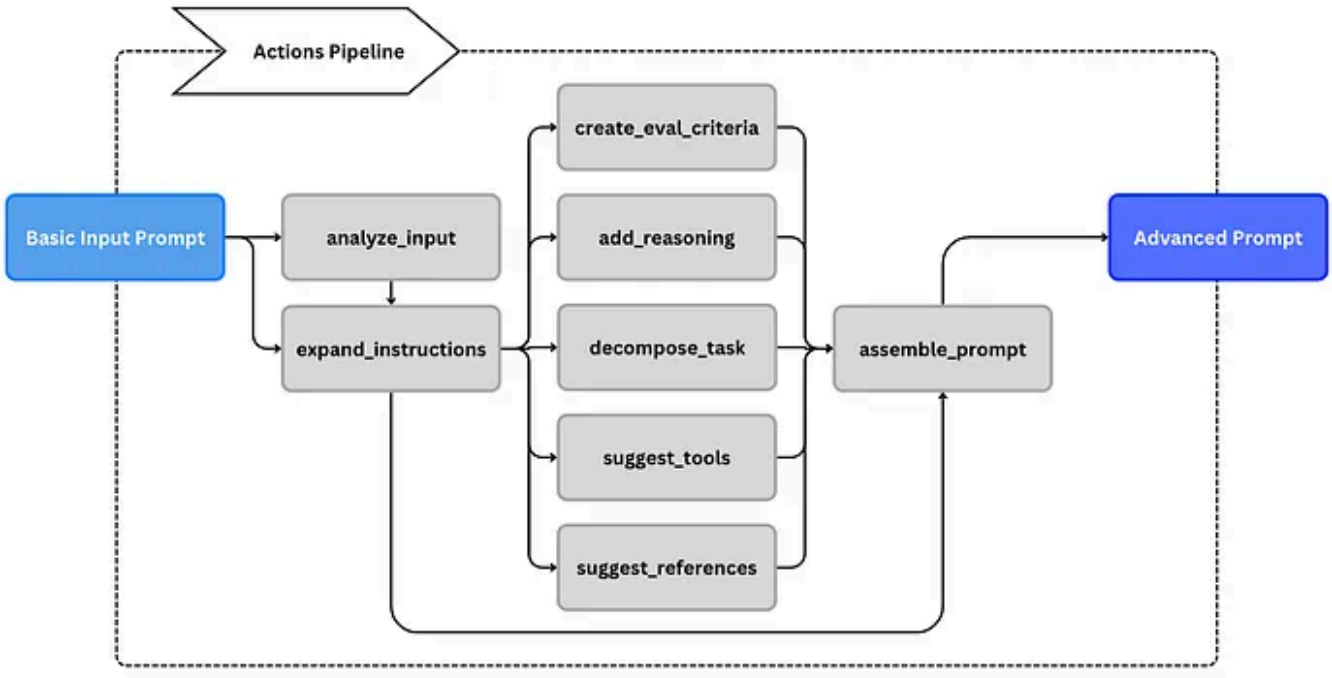
建模與建構解決方案

After we defined the rules, we modeled the pipeline. We will group these rules into specific components and assign an agent to each component. That means every agent will be instructed using a predefined prompt to generate information based on the given rules and input.

在我們定義了規則之後，我們對流程進行了建模。我們將這些規則分組為特定的組件，並為每個組件分配一個代理。這表示每個代理都將使用預定義的提示來接收指令，以根據給定的規則和輸入生成資訊。

Here's a flowchart to better visualize this step:

以下是流程圖，以便更好地視覺化此步驟：



The pipeline's flowchart 此管線的流程圖

Now, all that is left is to code everything and add access to the LLM.

現在，剩下的就是編寫所有程式碼並新增對 LLM 的存取。

We will use OpenAI's GPT-4o-mini as the LLM because it is really powerful, stable, and cost-efficient. However, you can adjust the code later to work with any LLM API.

我們將使用 OpenAI 的 GPT-4o-mini 作為 LLM，因為它功能強大、穩定且具成本效益。然而，您可以稍後調整程式碼以與任何 LLM API 搭配使用。

Testing and troubleshooting
測試與故障排除

The results were promising, so we're on to something here. We often achieved high-quality advanced prompts. However, there were multiple problems.

結果令人鼓舞，所以我們在這裡有所發現。我們經常能產生高品質的進階提示。然而，也存在多個問題。

The performance was not stable, and we frequently encountered irrelevant results, such as generation output from the prompt, information redundancy, and hallucinations.

效能不穩定，我們經常遇到不相關的結果，例如來自提示的生成輸出、資訊冗餘和幻覺。

Therefore, we needed to optimize our solution.

因此，我們需要最佳化我們的解決方案。

A stable and optimized solution
一個穩定且最佳化的解決方案

The identified causes of the problems mentioned above were the overlap of certain components in the pipeline, the underdevelopment of the agent's prompts, and a lack of restrictions within.

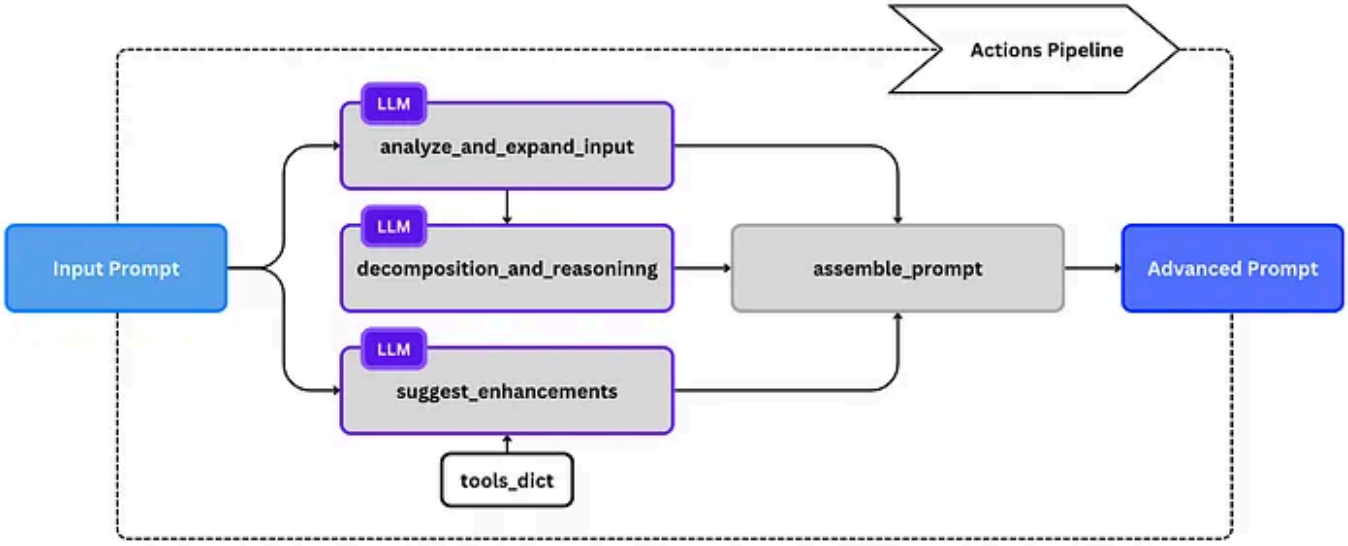
上述問題的成因在於管線中某些元件的重疊、代理程式提示的開發不足，以及內部缺乏限制。

So, we revisited and updated our pipeline by reducing the number of components and providing robust, well-detailed system prompts for the agents to tackle each component.

因此，我們重新檢視並更新了管線，減少了元件數量，並為代理程式提供了強大且詳細的系統提示，以處理每個元件。

Here's a flowchart of the updated pipeline:

以下是更新後的管線流程圖：



The updated pipeline's flowchart

更新後的管線流程圖

With this update, we ensured a stable and robust system that resolved all the previous issues while reducing complexity and generation costs.

透過這次更新，我們確保了系統的穩定性和穩固性，解決了之前所有的問題，同時降低了複雜度和生成成本。

Here are some examples to evaluate the solution:

以下是一些評估解決方案的範例：

(You can also check the Github repo or the HF Space and try it yourself)

（您也可以查看 Github 儲存庫或 HF Space 親自試用）

Example 1: 範例 1：

```
>>> INPUT PROMPT:
how to write a book?
```

```
>>> ADVANCED PROMPT:
As a knowledgeable writing coach, please provide a comprehensive guide on how to write a book.

Requirements:
1. Outline the key steps involved in the book writing process, including brainstorming, outlining, drafting, and revisi
2. Offer tips for maintaining motivation and overcoming writer's block.
3. Include advice on setting a writing schedule and establishing a writing environment.
4. Suggest resources for further learning about writing techniques and publishing options.

Structure your response as follows:
- Introduction to the book writing journey
- Step-by-step guide with actionable tips
- Strategies for motivation and productivity
- Recommended resources for aspiring authors

Keep the response medium in length (approximately 200-300 words) to ensure thorough coverage of the topic.

##REFERENCE SUGGESTIONS##
- "On Writing: A Memoir of the Craft" by Stephen King
  Purpose: Offers insights into the writing process and practical advice for aspiring authors
  Integration: Use as a guide for understanding the nuances of writing a book

- "The Elements of Style" by William Strunk Jr. and E.B. White
  Purpose: Provides essential rules of English style and composition
  Integration: Reference for improving writing clarity and effectiveness
```

```
- "Bird by Bird: Some Instructions on Writing and Life" by Anne Lamott
  Purpose: Shares personal anecdotes and practical tips for overcoming writing challenges
  Integration: Use for motivation and strategies to tackle the writing process

##THOUGHT PROCESS##
*Subtask 1*:
- **Description**: Introduce the book writing journey and its significance.
- **Reasoning**: Providing an introduction sets the context for aspiring authors, helping them understand the importanc
- **Success criteria**: The introduction should clearly articulate the purpose of writing a book and inspire readers ab

*Subtask 2*:
- **Description**: Outline the key steps involved in the book writing process: brainstorming, outlining, drafting, and
- **Reasoning**: Breaking down the writing process into clear steps helps authors navigate their journey systematically
- **Success criteria**: Each step should be defined with actionable tips, such as techniques for brainstorming ideas, s

*Subtask 3*:
- **Description**: Provide tips for maintaining motivation and overcoming writer's block.
- **Reasoning**: Addressing common challenges like writer's block is crucial for sustaining progress and ensuring autho
- **Success criteria**: Include practical strategies, such as setting small goals, taking breaks, and using prompts to

*Subtask 4*:
- **Description**: Advise on setting a writing schedule and establishing a conducive writing environment.
- **Reasoning**: A structured writing schedule and a supportive environment are essential for productivity and focus du
- **Success criteria**: Offer specific recommendations for daily writing routines and tips for creating a distraction-f

*Subtask 5*:
- **Description**: Suggest resources for further learning about writing techniques and publishing options.
- **Reasoning**: Providing additional resources empowers authors to deepen their knowledge and explore various publishi
- **Success criteria**: List reputable books, websites, and courses that cover writing skills and the publishing proces
```

Example 2: 範例二：

```
>>> INPUT PROMPT:
write a python script to compute and plot the fibonacci spiral
```

```
>>> ADVANCED PROMPT:
As a programming expert, please create a Python script that computes and plots the Fibonacci spiral.

Requirements:
- Use the Fibonacci sequence to generate the necessary points for the spiral.
- Utilize libraries such as Matplotlib for plotting and NumPy for numerical calculations.
- Include comments explaining each step of the code.
- Ensure the script is modular, allowing for easy adjustments to the number of Fibonacci terms used in the spiral.

Provide the implementation with:
- Function to compute Fibonacci numbers up to a specified term.
- Function to plot the Fibonacci spiral using the computed points.
- Example usage demonstrating the script in action, including a plot display.

The Fibonacci spiral should start with the first few Fibonacci numbers and visually represent the growth of the spiral

##REFERENCE SUGGESTIONS##
- Matplotlib Documentation
  Purpose: Provides details on plotting functions and customization options
  Integration: Reference for creating and customizing plots in the script
- Fibonacci Sequence Article
  Purpose: Offers insights into the mathematical properties and applications of the Fibonacci sequence
  Integration: Use as a reference for understanding the sequence's generation and its relation to the spiral
- Python Programming Guide
  Purpose: Serves as a comprehensive resource for Python syntax and libraries
  Integration: Reference for general Python programming practices and functions used in the script

##THOUGHT PROCESS##
*Subtask 1*:
- **Description**: Create a function to compute Fibonacci numbers up to a specified term.
- **Reasoning**: This function is essential for generating the sequence of Fibonacci numbers, which will be used to det
- **Success criteria**: The function should return a list of Fibonacci numbers up to the specified term, with correct v

*Subtask 2*:
```

```
- **Description**: Implement a function to plot the Fibonacci spiral using the computed points.
- **Reasoning**: Plotting the spiral visually represents the growth of the Fibonacci sequence, making it easier to unde
- **Success criteria**: The function should create a plot that accurately represents the Fibonacci spiral, with appropr

*Subtask 3*:
- **Description**: Include comments explaining each step of the code.
- **Reasoning**: Comments enhance code readability and maintainability, allowing others (or the future self) to underst
- **Success criteria**: The code should have clear, concise comments that explain the purpose of each function and key

*Subtask 4*:
- **Description**: Ensure the script is modular, allowing for easy adjustments to the number of Fibonacci terms used in
- **Reasoning**: Modularity allows users to easily modify the number of terms without altering the core logic of the sc
- **Success criteria**: The script should allow the user to specify the number of Fibonacci terms as an input parameter

*Subtask 5*:
- **Description**: Provide example usage demonstrating the script in action, including a plot display.
- **Reasoning**: Example usage helps users understand how to implement and run the script, showcasing its functionality
- **Success criteria**: The example should include a clear demonstration of how to call the functions, specify the numb
```

Conclusion 結論

In the end, we achieved very promising results. Of course, there is still a lot of room for improvement and further optimization, but most importantly, it was a fun project to work on. I'd love to hear your thoughts and ideas, so feel free to share them in the comments below!

最終，我們取得了非常有前景的成果。當然，仍有很大的改進和進一步優化的空間，但最重要的是，這是一個有趣的工作專案。我很樂意聽取您的想法和意見，請隨時在下面的評論中分享！