**Prompt Chaining – A Prompt Engineering Technique**

**-By Divyanshu Gupta**

1. **Overview**

Prompt Chaining is an **advanced technique in prompt engineering** used to improve the **performance, reliability, and clarity** of LLM outputs. It involves **breaking down a complex task into simpler, sequential subtasks**, each handled by a different prompt. The **output from one prompt becomes the input for the next**, forming a *chain of prompt operations*.

1. **Benefits of Prompt Chaining**

* **Improved performance** on complex or multi-step problems
* **Higher transparency** and easier debugging of intermediate stages
* **Greater control** over the model's outputs
* **Modular design** makes it easy to isolate, test, and improve parts of the task
* **Enhances user experience** in LLM-based conversational agents

Prompt chaining is especially beneficial in **document-based tasks**, **chatbot systems**, **educational tutors**, and **personalized assistants** where stepwise transformation of data is needed.

1. **Prompt Chaining Use Case: Document-Based Question Answering (QA)**

A classic use case is **document-based QA**, where the task is to answer a question by analyzing a large document. Handling this directly with a single prompt may overwhelm the model.

**Solution via Prompt Chaining:**

Split the task into **two stages**:

1. **Extract relevant quotes** from the document.
2. **Generate an answer** using those quotes and the original document.

**Prompt 1 – Extract Relevant Quotes**

**A computer screen with white text

AI-generated content may be incorrect.**

**Output:**

**A computer screen shot of a black screen

AI-generated content may be incorrect.**

**Prompt 2 – Generate Final Answer**

A screenshot of a computer program

AI-generated content may be incorrect.

**Output:**

A screenshot of a computer program

AI-generated content may be incorrect.

1. **Additional Applications of Prompt Chaining**

* **Multistep reasoning**: Break down reasoning tasks (e.g., math problems) into stepwise solutions.
* **Data extraction → summarization → QA**: Used in chatbots handling FAQs or support documentation.
* **Personalized feedback generation**: Collect input from users, analyze sentiment, and provide customized responses in stages.
* **Dynamic workflows in apps**: Use outputs to guide next steps in interactive assistants.

1. **Summary Points**

* Prompt Chaining divides a complex task into smaller, manageable sub-tasks.
* It enhances the transparency, control, and reliability of LLM-generated outputs.
* In document-based tasks, chaining improves QA accuracy and clarity.
* Each prompt can perform a transformation or filtering operation in the pipeline.
* Useful in developing scalable and personalized AI applications.