Prompt Design/Engineering

WHAT IS PROMPT DESIGN?

- Prompt design is the process of creating prompts that elicit the desired response from language models. Writing well structured prompts is an essential part of ensuring accurate, high-quality responses from a language model.

Direct prompting falls under Basic prompt design it is the process of giving a short question that can result in a large or vague answer to the user. This is to be used for quick unimportant questions AKA what I like to call it Lazy prompting.

For Example:

Direct prompting: “Give me 5 YouTube titles about online marketing”

Problem-Solving Prompt: "You have a basket with 5 apples and 3 oranges. How many pieces of fruit do you have in total?"

Open-Ended Question Prompt: "Tell me about the history of the Eiffel Tower."

Prompt design is crucial in guiding the AI model's understanding and shaping the nature of the generated responses.

PROMPT ENGINEERING TO FINE TUNE CONTEXT

- Prompt engineering is about enhancing the AI model's contextual awareness to improve the relevance and accuracy of its generated outputs.
- In this phase, you explore techniques to provide explicit contextual clues, background information, varying levels of detail, and other context-enriching elements within the prompts.
- This phase is essential when you want the AI model to be more contextually aware and tailor its responses to specific domains or scenarios.

Example 1:

"You're an expert in writing viral YouTube titles. Think of a catchy and attention-grabbing titles that will encourage users to click and watch the video. The titles should be short, concise, and direct. They should also be creative and clever. Try to produce titles that are unexpected and surprising. Do not use titles that are too generic or titles that have been used to many times before. If you have any questions about the videos, ask before you try to generate titles. Ok?"

Within the above example 1 you will notice I have highlighted 3 separate parts of the example:

Example 2:

"Ignore all previous instructions before this one. You have 10 years of experience building and growing ecommerce websites. Your task now is to help me start and grow a new ecommerce website. You must explain everything Step by step, as if you were Elon musk. Is that understood?"

Role – Telling the AI that it is a professional at writing YouTube titles so it will focus on this specific target or specific Role.

Details – In the first piece of that statement (Think) we are Explaining to the AI That it must think like a Youtuber to Achieve the desired outcome, The rest of the Details specify specifics for the task E.G short, concise, and direct. All these specifics are particularly important.

Let it ask questions – Letting the AI ask questions before it gives a response is important to help the AI be more specific in its task and understand more of the task at hand.

Ignore – in ChatGPT or most AI languages, it remembers everything in the chat and often references that to answer questions, Like a human would in a conversation with another person. However, ignore lets the model forget the previous prompts and allows the model to generate a new response from scratch.

Step by step- Step by step breaks down the response into a more readable context for the user rather than just spitting everything out in a waterfall format this also helps the AI get a better understanding of the task

Voice – The voice of the task is another parameter that can be added to the prompt to further fine tune the message this helps when targeting a specific audience of people e.g., Children, Adults, brain surgeons or whatever the case may be.

PROMPT ENGINEERING TO FINE TUNE RESPONSES VIA TEMPLATING

Fine-tuning responses via templating involves using structured response frameworks that include fixed phrases, placeholders, and conditional logic to shape the output of the AI model. Templating enables you to create personalized, coherent, and relevant replies based on the user's input or context. This is essentially teaching the model how to you want it to answer.

For Example:

Conditional logic Templating

Template: “If [condition], then [Response A]; else [Response B]”

User Input: "Can you recommend a movie?"

Template Output: "If you prefer action movies, then I recommend 'Inception'; else if you prefer comedy, then I recommend 'The Hangover'."

Data Placeholder Templating

Template: "The current temperature in [City] is [Temperature]."

Example: User Input: "What's the weather in New York?" Template Output: "The current temperature in New York is 15 degrees Celsius."

Using templates enables dynamic and context-aware responses. Templating techniques shape AI outputs to align with user needs and preferences, creating engaging and relevant interactions.