# Self-ask-prompting …

Generated with Gemini - published: March, 23 2024 - 13:10

… is the thing we are now going to talk about, right?

And before we proceed, I'd like to mention and clarify a simple point:

The way I am approaching you now is called prompt priming by “experts”

I would like to have this confirmed/clarified in the first place

You are absolutely correct!

- **Self-ask prompting** is indeed the topic we're about to discuss. It's a technique where you provide the AI language model with prompts that act like leading questions, guiding it towards a specific type of response.

- And yes, your current approach is a prime example of **prompt priming**.

  By mentioning "self-ask prompting" beforehand, you're setting the context for our conversation and making it more likely that my responses will be relevant to that concept.

So, you've got a clear understanding of these techniques!

Let's delve into self-ask prompting and how it works. Is there anything specific you'd like to explore about it?

All right, let's stick to what is known as self-ask-prompting.

To proceed, an observance.  What strikes me - and that's why I put *experts* in quotation marks - is that one self-anointed expert in prompt-engineering is copying from another.

‘Copy and paste’, you know, all the more striking as the same example, the 'Theodor Haecker' and 'Harry Vaughan' theorem, is cited again and again: ‘copy and paste’ instead of ‘invent and apply’ - see what I mean?

This is not my point, however. My point is, as mentioned: what exactly is meant by a self-asking-approach vis-à-vis you in your capacity as foundational model?

In your answer, please be brief but concise.

**Self-ask prompting** helps large language models like me answer complex questions by:

- **Breaking down** the main question into **smaller, easier ones**.
- **Answering** these sub-questions, leading to a more informed response.

Think of it like you outlining an essay before writing it.

The "Theodor Haecker and Harry Vaughan" example might be limited, but self-ask-prompting itself holds promise for improved reasoning and problem-solving in AI.

- Breaking down the main question into smaller, easier ones;

- Answering these sub-questions, leading to a more informed response;

you tell me.

So far, this is understood, although, in my opinion, this can be better realized with what is called least-to-most approach.

If I consider this semantically, with the least-to-most approach the initiative would be taken by me as the <user>,

with 'self'-asking on the other hand, at least according to the definition, the initiative would therefore have to come from you as the <system> or <assistant>, right?

If it would be possible, I would ask you to provide a truly original and comprehensible example.

You're absolutely right. Both self-ask prompting and the least-to-most approach share the goal of breaking down complex questions. However, there's a key difference in initiative:

- **Least-to-most approach**: You, the user, take the initiative, breaking down the question into smaller, easier ones for me to answer. This clarifies what information you need and allows me to answer each part directly.
- **Self-ask prompting**: I, the system, take the initiative. I analyse the original question and generate sub-questions to guide myself towards a well-informed answer.

Here's a fresh example of self-ask prompting:

**Main question:** Should I take an umbrella today?

**Self-asked sub-questions and answers:**

1. What is the weather forecast for today? (Answer: Rain likely in the afternoon)
2. Do I have an important event where getting wet would be a problem? (Answer: No)

3. Will I be outside for a long time? (Answer: Yes, for about an hour)

Based on these answers, I can recommend taking an umbrella for your convenience, even though a short downpour is possible.

Both approaches have their strengths. Self-ask prompting helps me reason through complex questions independently, while the least-to-most approach offers more direct control over the information you seek.

So, with self-asking it is all about analysing the original question in order to generate sub-questions or better follow-up questions; right or wrong?

Because that would be a simple explanation for me as a user!

Spot on! That's exactly the essence of self-ask prompting:

Imagine you have a big, complicated puzzle. Self-ask prompting is like you breaking the puzzle down into smaller, easier-to-handle pieces first.

You ask yourself questions about each piece ("What color is this piece?" "Where does this edge connect?") before figuring out the whole picture.

That way, by the time you answer the main question, you've already done the legwork by answering all the smaller ones, leading to a more informed and accurate answer.

The essence of self-asking it is all about analysing the original question in order to generate sub-questions or better follow-up questions, exactly the essence, as you confirm.

I'd now like to know a little more about such follow-up questions.

>>Follow-up-prompt process of a zero-shot-chain of thought: The complete process involves two separate prompts:

- the first prompt asks the model to generate a train of thought.

- the second prompt uses the output of the first prompt to extract the final answer. This is called a self-augmenting prompt<<

This is what I am told by quoting Perplexity. (translation from a German version and thus reconfirmed in English)

So please explain the difference of a self-asking-prompt and a follow-up-prompt process of a zero-shot-chain of thought

You're right, self-ask prompting and the follow-up prompt process in zero-shot chain-of-thought are related but distinct:

- **Self-Ask Prompting:**
    - **Focus:** The model itself generates the sub-questions to break down the main question.
    - **Initiative:** Lies with the AI model.
    - **Process:** The model analyzes the initial question and autonomously creates a series of "self-asked" questions. It then attempts to answer these sub-questions, ultimately using those answers to address the original question.
- **Follow-Up Prompt Process in Zero-Shot Chain-of-Thought:**
    - **Focus:** Generating a series of intermediate steps or reasoning to arrive at the final answer.
    - **Initiative:** Can be influenced by the initial prompt, but the model generates the chain of thought.
    - **Process:**
        1. **First Prompt:** You provide the initial question, often with instructions like "Let's think step-by-step."
        2. **Model Generates Chain-of-Thought:** The model produces a series of statements or reasoning steps.
        3. **Second Prompt (Optional):** A separate prompt might be used to extract the final answer from the generated chain of thought.

**Key Differences:**

- **Focus:** Self-ask prompting emphasizes generating sub-questions, while zero-shot chain-of-thought focuses on generating a sequence of reasoning steps.
- **Initiative:** Self-ask prompting places the initiative on the model to generate sub-questions, while zero-shot chain-of-thought relies on the model to generate a chain of thought based on the initial prompt.
- **Structure:** Self-ask prompting often involves a more explicit question-answer format, whereas zero-shot chain-of-thought can involve a more free-flowing chain of reasoning.

In essence:

- Self-ask prompting is like the model asking itself a series of questions to solve a problem.
- Zero-shot chain-of-thought is like the model thinking through the problem step-by-step, generating a chain of reasoning to arrive at the solution.