AICamp

Advance Prompt Engineering
Day 6

How to master the Art of Conversation with AI?

- Subash Natarajan
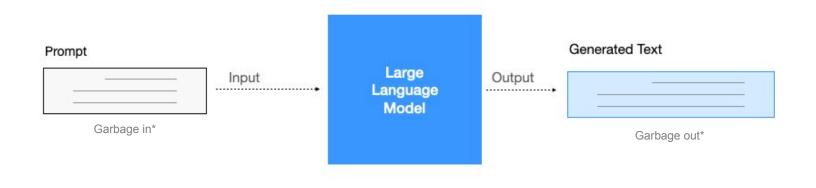
Agenda

1. What is Prompt Engineering?

2. Prompting Formula (not a perfect formula)

3. Prompting Techniques - Demo

4. Decoding Prompting Myths

5. The Future of Prompting

What is Prompt Engineering?

The Art of How We Communicate with LLM/AI

Prompt

Input →

Large Language Model

Output →

Generated Text

Garbage in*

Garbage out*

Prompting Best Practices

Not a Perfect Formula:

Role + Context + Task + Example + Format + Tone + Goals

Basic Prompt (One-Shot):

"You are a (Role/Persona) senior AI engineer. (Context) A junior developer on your team is trying to decide between TensorFlow and PyTorch for a new deep learning project, focused on Image classification or object detection. (Task) Explain the key differences between TensorFlow and PyTorch, focusing on their strengths and weakness. (Example) You can compare their approaches to building and executing models. (Format) Keep your explanation concise, under 75 words. (Tone) Use a professional yet clear tone. (Goals) The goal is for the junior developer to have a clear understanding of the main differences and help them choose the right framework.

Advance Prompting Techniques (Demo)

1. ## Chain-of-Thoughts (CoT)
   Giving the AI a roadmap (to-do list). Instead of asking for the answer directly, we break down the problem into smaller steps.

2. ## Tree of Thoughts (ToT)
   We're not just asking for steps, LLM to consider multiple possibilities (brainstorm), like branches on a tree.

3. ## Self-Consistency
   Generating multiple response to same prompt and identifying most consistent answer as final answer, .Reduce bias, hallucination.

Debunking Prompting Myths

Myths:
1. *Role Prompting is always effective*
2. *Always follow the formula*
3. *Longer prompt is better*

Approach:
1. *Clear & effective thinking is important than just relaying formula*
2. *Have an end in mind, and goal clearly*
3. *Be clear with your tasks (Research, Summary…)*

Tips:
1. *Provide clear and relevant Context*
2. *Start simple and iterate*
3. *Ask questions to find blind-spots (do you understand?)*
4. *Use natural language like talking to human*

Future of prompting?

1. AI-assisted Prompt Generation (eg, Prompt Perfect)

2. A de-facto standard for Low-Code and No-Code Platforms

3. App ➡ Query Interface ⬅ Prompting (native language)

Thank you!

Subash Natarajan
https://www.linkedin.com/in/subashn/