

GPT Survey: A Programmer's Perspective

Yu Zehan

Intel FLEX

2023/05/09

Outline

- 1 Lesson 1: Prompt Engineering: Principles, Techniques, and Tools
- 2 Lesson 2: Integrate GPT in Whole Coding Lifecycle
- 3 Lesson 3: Cutting-Edge GPT Projects and Showcases
- 4 Lesson 4: Customize GPT for More Power
- 5 References

GPT Survey: A Programmer's Perspective

Lesson 1: Prompt Engineering: Principles, Techniques, and Tools

Yu Zehan

Intel FLEX

2023/05/09

Outline of Lesson 1

- 1 Lesson 1: Prompt Engineering: Principles, Techniques, and Tools
 - SPQA Architecture
 - Prompt Workflows
 - Prompt Principles
 - Prompt Techniques
 - Tools and Extensions

SPQA Architecture

Model-based. Will take the most compute to maintain, will likely be broken into multiple models based on the need for real-time.

Model-based. Will be the smallest of the models. Will mostly be updated for changes in desired direction, new threats, etc.

Human-driven. The quality of the questions will determine the quality of the outputs. AI Whisperers will continue to thrive.

Existing Workflows. The models will be connected to existing tooling, such as Slack, Email, Service Now, etc.



Circuit-Based vs Understanding-Based

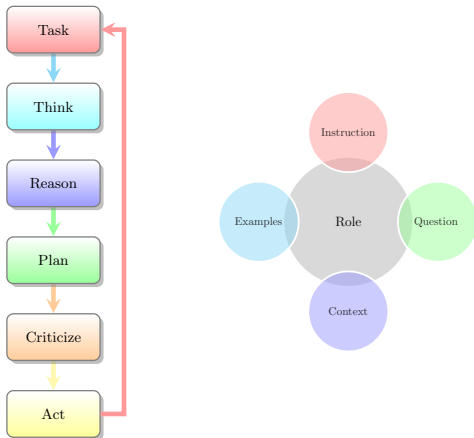
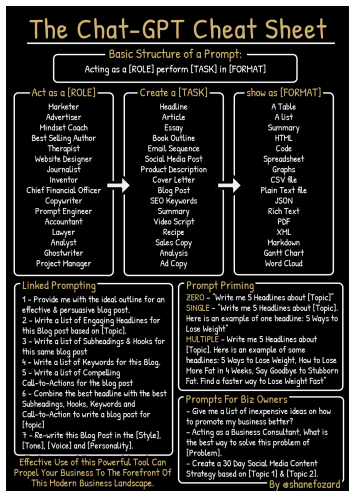
CIRCUIT-BASED VS. UNDERSTANDING-BASED APPLICATIONS

Circuit-based applications have **explicit** and **rigid** structures. Inputs and outputs must be explicitly created, routed, and maintained. Any deviation from that structure results in errors, and adding new functionality requires linear effort on the part of the organization's developers. **Understanding-based applications** **adjust to the questions they are asked**. To add new functionality, you ask different questions and ask for different actions to be done with the results.

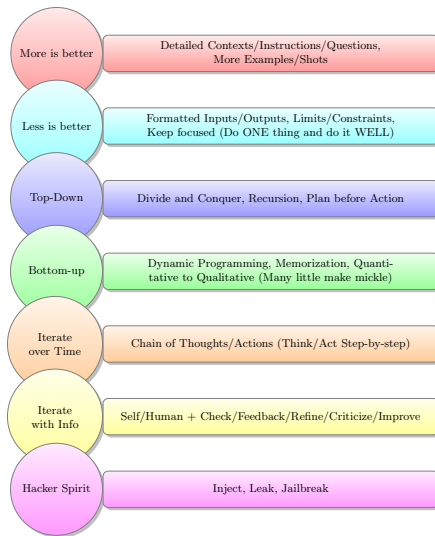
	CIRCUIT-BASED	UNDERSTANDING-BASED
INPUTS	RIGID — Calls into the application must be explicitly programmed into the application as queries, by a development team, using explicit queries against a structured database	UNLIMITED — Calls into the application are made using natural language , so that you don't have to explicitly code new queries against the backend when you want to ask different questions
DATA	STRUCTURED — The application's data stores have explicitly-defined schemas that require particular ways of adding, retrieving, modifying, and deleting data from the system	DYNAMIC — The backend of the system is a set of models that understand the world and your business. To upgrade your data store you ingest more/better data and retrain your models
OUTPUT	STATIC — Output from the application returns the specific information asked for with a specific query. To do something else, you must then write separate code to do that	LINKED — Output from the application is just as flexible as the questions you ask the models. And the question and response can directly link to associated actions
ACTION	SEPARATE — Actions are separate from query results . If you ask for the results of a query about home prices to a database, you must then do one or more additional things to make it useful	CONNECTED — Actions can be connected to results because the models understand how they are linked. This means you don't have to create explicit bridges from every result to new actions
UNDERSTANDING	NONE — The application doesn't understand the data, nor the queries running against it, nor the results it returned. It is a simple mechanism for storing and extracting data	DEEP — The models understand the material they're dealing with, which means improvements in the models improve most aspects of the application at the same time
FLEXIBILITY	LINEAR — To add more functions to an application, you must explicitly expand the number of explicitly-coded inputs, the number of output actions, the data structure, or all three	EXTRAORDINARY — Your primary concerns become: 1) ensuring the quality and freshness of your models, and 2) ensuring that you're asking the right questions



Prompt Workflows



Prompt Principles



Prompt Techniques 1: More is Better

A basic case

Here is where we start.

You

Write a Scrawler for me.

Bing AI

I'm sorry, but I'm not able to write a web scraper for you. However, I can provide you with information on how to write one yourself or where to find resources to learn how to do so. Would you like me to help you with that?

Prompt Techniques 1: More is Better

Improve the basic case

Let's add some info.

You

Provide a Python Scraper script.

Bing AI

Search Engine is Still Your Friend

Human Skills is Still Important

GPT applications comparisons

Tools and Extensions

GPT Survey: A Programmer's Perspective

Lesson 2: Integrate GPT in Whole Coding Lifecycle

Yu Zehan

Intel FLEX

2023/05/15

Outline of Lesson 2

2 Lesson 2: Integrate GPT in Whole Coding Lifecycle

GPT Survey: A Programmer's Perspective

Lesson 2: Integrate GPT in Whole Coding Lifecycle

Yu Zehan

Intel FLEX

2023/05/15

GPT Survey: A Programmer's Perspective

Lesson 3: Cutting-Edge GPT Projects and Showcases

Yu Zehan

Intel FLEX

2023/05/15

Outline of Lesson 3

3 Lesson 3: Cutting-Edge GPT Projects and Showcases

GPT Survey: A Programmer's Perspective

Lesson 3: Cutting-Edge GPT Projects and Showcases

Yu Zehan

Intel FLEX

2023/05/15

GPT Survey: A Programmer's Perspective

Lesson 4: Customize GPT for More Power

Yu Zehan

Intel FLEX

2023/05/15

Outline of Lesson 4

4 Lesson 4: Customize GPT for More Power

GPT Survey: A Programmer's Perspective

Lesson 4: Customize GPT for More Power

Yu Zehan

Intel FLEX

2023/05/15

References I

1. SPQA: The AI-based Architecture That'll Replace Most Existing Software.
- <https://danielmiessler.com/blog/spqa-ai-architecture-replace-existing-software>
2. Introduction to prompt engineering - Azure OpenAI.
- [Introduction: https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/prompt-engineering](https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/prompt-engineering)
- [Techniques: https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/advanced-prompt-engineering](https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/advanced-prompt-engineering)
3. Azure OpenAI Service models.
- <https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/models>
4. Chat GPT For Programming: 100+ Coding Prompts For Chat GPT.
- <https://www.greataiprompts.com/chat-gpt/best-coding-prompts-for-chat-gpt>
5. Chat-GPT Cheat Sheet V2 : coolguides.
- [Reddit: https://www.reddit.com/r/coolguides/comments/12ouw8o/chatgpt_cheat_sheet_v2](https://www.reddit.com/r/coolguides/comments/12ouw8o/chatgpt_cheat_sheet_v2)
- [Cheatsheet: https://i.redd.it/eunvmybl9cua1.png](https://i.redd.it/eunvmybl9cua1.png)
6. ChatGPT Prompt Engineering Principles for Developers — Clarity and Specificity in Instructions.
- <https://levelup.gitconnected.com/chatgpt-prompt-engineering-principles-for-developers-clarity-and-specificity-in-instructions-61f60ffd1eed>
7. How to Communicate with ChatGPT – A Guide to Prompt Engineering.
- <https://www.freecodecamp.org/news/how-to-communicate-with-ai-tools-prompt-engineering>

References II

8. Learn Prompting: Your Guide to Communicating with AI.
 - Basics: <https://learnprompting.org/docs/category/-basics>
 - Intermediate: <https://learnprompting.org/docs/category/%EF%B8%8F-intermediate>
 - Advanced Applications: <https://learnprompting.org/docs/category/-advanced-applications>
 - Reliability: <https://learnprompting.org/docs/category/%EF%B8%8F-reliability>
 - Prompt Hacking: <https://learnprompting.org/docs/category/-prompt-hacking>
 - Additional Resources: <https://learnprompting.org/docs/additional>
9. StarCoder: A State-of-the-Art LLM for Code.
 - HuggingFace: <https://huggingface.co/bigcode/starcoder>
 - GitHub: <https://github.com/bigcode-project/starcoder>
 - Blog: <https://huggingface.co/blog/starcoder>
 - Paper: <https://drive.google.com/file/d/1cN-b9GnWtHzQRoE7M7gAEyivY0kl4BYs/view>
10. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond.
 - arXiv: <https://arxiv.org/abs/2304.13712>
 - GitHub: <https://github.com/mooler0410/llmspracticalguide>
11. ChatGPT Prompt Engineering for Developers.
 - Official Course: <https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/>
 - Notebooks: <https://github.com/datawhalechina/prompt-engineering-for-developers/tree/main>
 - Bilibili: <https://www.bilibili.com/video/BV1No4y1t7Zn>
12. FlowGPT — the best ChatGPT prompts&AI prompts community.
 - <https://flowgpt.com/>
13. awesome-chatgpt-prompts: This repo includes ChatGPT prompt curation to use ChatGPT better.
 - GitHub: <https://github.com/f/awesome-chatgpt-prompts>
 - Webpage: <https://prompts.chat/>
 - HuggingFace: <https://huggingface.co/datasets/fka/awesome-chatgpt-prompts>

References III

14. The Art of ChatGPT Prompting: A Guide to Crafting Clear and Effective Prompts.
- <https://app.gumroad.com/d/e2e51298d50b8c1bd6c6950a6d11a772>
15. PGF/TikZ Manual.
- [PDF](http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf): <http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf>
- [Online Doc](https://tikz.dev/): <https://tikz.dev/>
- [GitHub](https://github.com/pgf-tikz/pgf): <https://github.com/pgf-tikz/pgf>

