GPT Survey: A Programmer's Perspective

Yu Zehan

Intel FLEX

2023/05/28

Outline

1. Lesson 1: Prompt Engineering: Principles, Techniques, and Tools
2. Lesson 2: Case Study: Become a Full-stack Engineer in 1 hour
3. Lesson 3: Integrate GPT in Whole Coding Lifecycle
4. Lesson 4: Cutting-Edge GPT Projects and Showcases
5. Lesson 5: Customize GPT for More Power
6. References

GPT Survey: A Programmer's Perspective

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

Yu Zehan

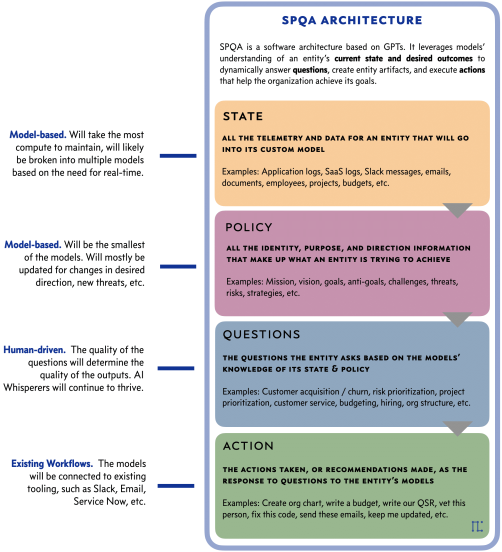
Intel FLEX

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



SPQA ARCHITECTURE

SPQA is a software architecture based on GPTs. It leverages models' understanding of an entity's **current state** and **desired outcomes** to dynamically answer **questions**, create entity artifacts, and execute **actions** that help the organization achieve its goals.

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

STATE

ALL THE TELEMETRY AND DATA FOR AN ENTITY THAT WILL GO INTO ITS CUSTOM MODEL

Examples: Application logs, SaaS logs, Slack messages, emails, documents, employees, projects, budgets, etc.

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

POLICY

ALL THE IDENTITY, PURPOSE, AND DIRECTION INFORMATION THAT MAKE UP WHAT AN ENTITY IS TRYING TO ACHIEVE

Examples: Mission, vision, goals, anti-goals, challenges, threats, risks, strategies, etc.

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

QUESTIONS

THE QUESTIONS THE ENTITY ASKS BASED ON THE MODELS' KNOWLEDGE OF ITS STATE & POLICY

Examples: Customer acquisition / churn, risk prioritization, project prioritization, customer service, budgeting, hiring, org structure, etc.

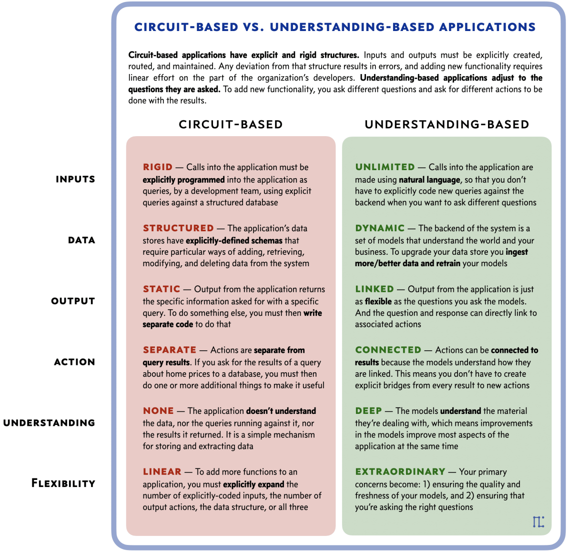
Existing Workflows. The models will be connected to existing tooling, such as Slack, Email, Service Now, etc.

ACTION

THE ACTIONS TAKEN, OR RECOMMENDATIONS MADE, AS THE RESPONSE TO QUESTIONS TO THE ENTITY'S MODELS

Examples: Create org chart, write a budget, write our QSR, vet this person, fix this code, send these emails, keep me updated, etc.

Circuit-Based vs Understanding-Based

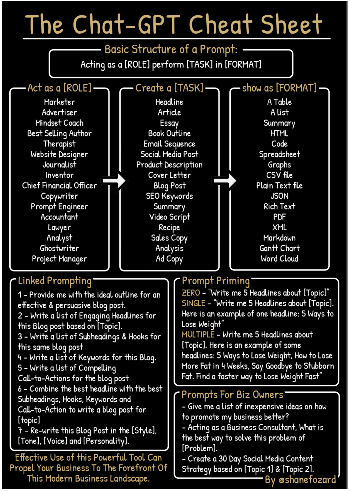


CIRCUIT-BASED VS. UNDERSTANDING-BASED APPLICATIONS

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



The Chat-GPT Cheat Sheet

Basic Structure of a Prompt:

Acting as a [ROLE] perform [TASK] in [FORMAT]

Act as a [ROLE]

- Marketer
- Advertiser
- Mindset Coach
- Best Selling Author
- Therapist
- Website Designer
- Journalist
- Inventor
- Chief Financial Officer
- Copywriter
- Prompt Engineer
- Accountant
- Lawyer
- Analyst
- Ghostwriter
- Project Manager

Create a [TASK]

- Headline
- Article
- Essay
- Book Outline
- Email Sequence
- Social Media Post
- Product Description
- Cover Letter
- Blog Post
- SEO Keywords
- Summary
- Video Script
- Recipe
- Sales Copy
- Analysis
- Ad Copy

show as [FORMAT]

- A Table
- A list
- Summary
- HTML
- Code
- Spreadsheet
- Graphs
- CSV file
- Plain Text file
- JSON
- Rich Text
- PDF
- XML
- Markdown
- Gantt Chart
- Word Cloud

Linked Prompting

1 - Provide me with the ideal outline for an effective & persuasive blog post.
2 - Write a list of Engaging Headlines for this Blog post based on [Topic].
3 - Write a list of Subheadings & Hooks for this same blog post
4 - Write a list of Keywords for this Blog.
5 - Write a list of Compelling Call-to-Actions for the blog post
6 - Combine the best headline with the best Subheadings, Hooks, Keywords and Call-to-Action to write a blog post for [topic]
7 - Re-write this Blog Post in the [Style], [Tone], [Voice] and [Personality].

Effective Use of this Powerful Tool Can Propel Your Business To The Forefront Of This Modern Business Landscape.

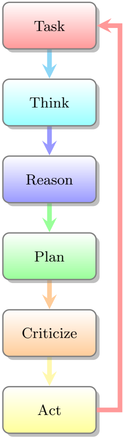
Prompt Priming

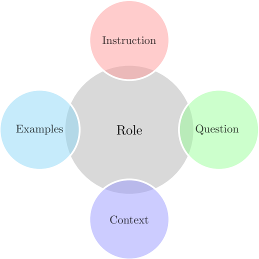
ZERO - "Write me 5 Headlines about [Topic]"
SINGLE - "Write me 5 Headlines about [Topic]. Here is an example of one headline: 5 Ways to Lose Weight"
MULTIPLE - Write me 5 Headlines about [Topic]. Here is an example of some headlines: 5 Ways to Lose Weight, How to Lose More Fat in 4 Weeks, Say Goodbye to Stubborn Fat. Find a faster way to Lose Weight Fast"

Prompts For Biz Owners

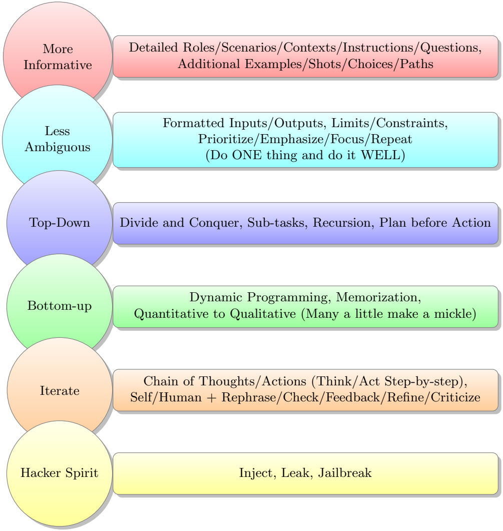
- Give me a list of inexpensive ideas on how to promote my business better?
- Acting as a Business Consultant, What is the best way to solve this problem of [Problem].
- Create a 30 Day Social Media Content Strategy based on [Topic 1] & [Topic 2].

By @shanefozard





Prompt Principles



More Informative	Detailed Roles/Scenarios/Contexts/Instructions/Questions, Additional Examples/Shots/Choices/Paths
Less Ambiguous	Formatted Inputs/Outputs, Limits/Constraints, Prioritize/Emphasize/Focus/Repeat (Do ONE thing and do it WELL)
Top-Down	Divide and Conquer, Sub-tasks, Recursion, Plan before Action
Bottom-up	Dynamic Programming, Memorization, Quantitative to Qualitative (Many a little make a mickle)
Iterate	Chain of Thoughts/Actions (Think/Act Step-by-step), Self/Human + Rephrase/Check/Feedback/Refine/Criticize
Hacker Spirit	Inject, Leak, Jailbreak

Prompt Techniques: Let's Start!

A Hello World 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: Little change makes huge effects

Little change makes huge effects

Add programming language; Use Scraper that AI understood.

You

Provide a Python Scraper script.

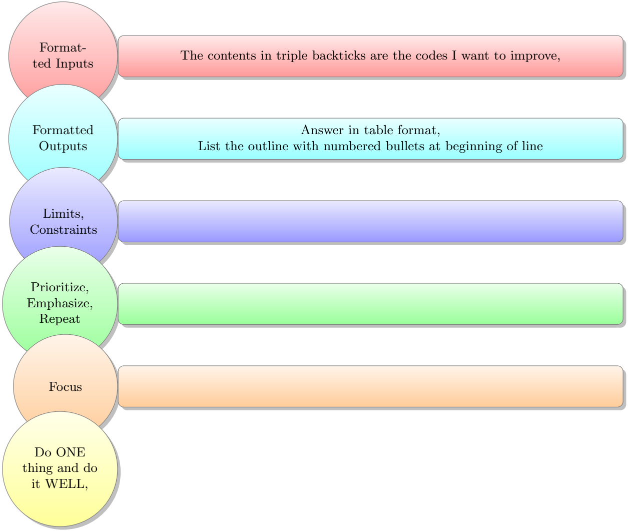
Bing AI

Here is a basic Python script that can be used as a starting point for writing a web scraper:
... (# Python codes Here)
This script uses the ‘requests‘ and ‘BeautifulSoup‘ libraries to fetch and parse the HTML of a web page. You can then use the methods provided by ‘BeautifulSoup‘ to find and extract the data you want from the page.
Keep in mind that this is just a basic example and you may need to modify it to fit your specific needs. Additionally, make sure to follow any terms of service or usage policies for the website you are scraping.

Prompt Techniques: More Infromative

Roles	Act as Python expert, You are an experienced Frontend programmer
Scenarios, Contexts	I'm writing a Python Project, You are developping a concurrent program for data processing
Instructions	You should extract all definitions of functions from this script, You task is to complete this script with a timeit feature, I would like to add an web UI to this
Questions	Could you give some suggestions on this plan, Can you summarize the goal of the codes
Examples, Shots	Here is an example, The output should be similar to the following formats, You should use same method in the provided script
Choices, Options, Paths	You could use `requests` library to do http GET and POST operations, You can try different backend frameworks such as Django or Flask or FastAPI

Prompt Techniques: Less Ambiguous



Formatted Inputs

The contents in triple backticks are the codes I want to improve,

Formatted Outputs

Answer in table format,
List the outline with numbered bullets at beginning of line

Limits, Constraints

Prioritize, Emphasize, Repeat

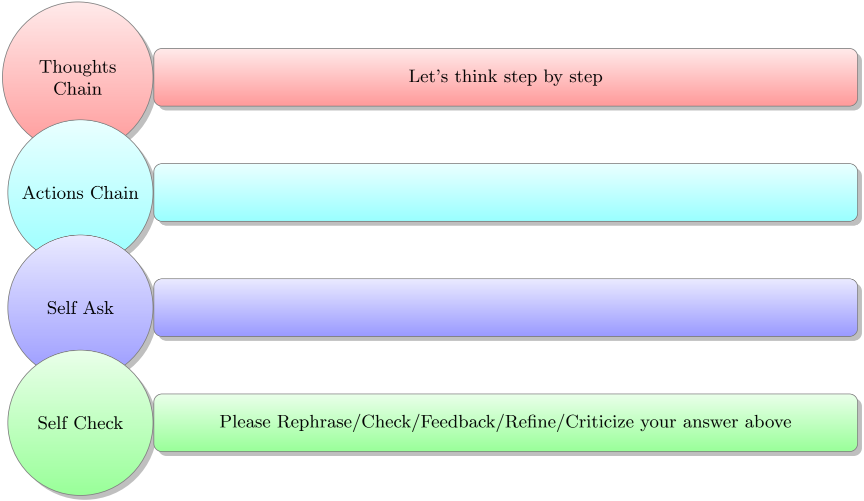
Focus

Do ONE thing and do it WELL,

Prompt Techniques: Top-Down

Prompt Techniques: Bottom-up

Prompt Techniques: Iterate



Thoughts Chain

Let's think step by step

Actions Chain

Self Ask

Self Check

Please Rephrase/Check/Feedback/Refine/Criticize your answer above

Prompt Techniques: Hacker Spirit

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: Case Study: Become a Full-stack Engineer in 1 hour

Yu Zehan

Intel FLEX

2023/06/01

Outline of Lesson 2

2 Lesson 2: Case Study: Become a Full-stack Engineer in 1 hour

GPT Survey: A Programmer's Perspective

Lesson 2: Case Study: Become a Full-stack Engineer in 1 hour

Yu Zehan

Intel FLEX

2023/06/01

GPT Survey: A Programmer's Perspective

Lesson 3: Integrate GPT in Whole Coding Lifecycle

Yu Zehan

Intel FLEX

2023/06/01

Outline of Lesson 3

3 Lesson 3: Integrate GPT in Whole Coding Lifecycle

GPT Survey: A Programmer's Perspective

Lesson 3: Integrate GPT in Whole Coding Lifecycle

Yu Zehan

Intel FLEX

2023/06/01

GPT Survey: A Programmer's Perspective

Lesson 4: Cutting-Edge GPT Projects and Showcases

Yu Zehan

Intel FLEX

2023/06/01

Outline of Lesson 4

4 Lesson 4: Cutting-Edge GPT Projects and Showcases

GPT Survey: A Programmer's Perspective

Lesson 4: Cutting-Edge GPT Projects and Showcases

Yu Zehan

Intel FLEX

2023/06/01

GPT Survey: A Programmer's Perspective

Lesson 5: Customize GPT for More Power

Yu Zehan

Intel FLEX

2023/06/01

Outline of Lesson 5

5 Lesson 5: Customize GPT for More Power

GPT Survey: A Programmer's Perspective

Lesson 5: Customize GPT for More Power

Yu Zehan

Intel FLEX

2023/06/01

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
   - Techniques: 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
   - Cheatsheet: 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
    - Online Doc: https://tikz.dev/
    - GitHub: https://github.com/pgf-tikz/pgf

FAQ