

Prompt Engineering

Getting Started

Administrative Details

- Attendance secret code! Promptly Print Prompting Precepts
 - <https://forms.gle/fY8G9e4oeiJSqYKk8>
 - Course Canvas set up
 - Feedback : On Thursday, we'll ask for feedback, to make sure we're delivering
 - John away in Japan for Inception Studio Business on Thursday
 - Guest lecturer Josh Payne to present then

A Non Deterministic Programming Model

- LISP introduced the deterministic functional programming paradigm
 - ChatGPT heralds the arrival of nondeterministic functional programming
-
- Instead of bottom up software development, top down solution mining
 - Instead of largely reliable CPU/GPU performance, finicky & arbitrary behavior

LLM as a Smart Overeager Inexperienced Intern

- Tremendous amount of knowledge
- Compelled to answer no matter what
- No experience combining facts
- Extremely literal
- Can be lazy, looks for shortcuts
- Mimics the style of the prompter

CS 224G as Managerial Training

- How do you as an individual scale?
 - Achieve more, in less time with less effort
- Think of AI as your capable force multiplier
- Engineer consistent scalable outcomes
- Use these techniques everywhere
 - Whenever limitations or constraints preclude success

Prompts in Practice

Overheard in a Zoom Call

- Stephen Henderson and Jan Jannink discussed how to remove an unwanted element from their meeting interface, which Stephen found invasive. Jan explained that they could turn it off by accessing their own app settings, and demonstrated how to do so. The conversation ended with both participants agreeing to stop the unwanted feature.
- Serious “2001 : A Space Odyssey” overtones
- This uncanny output is achieved with prompts not self-awareness
- The mutual feedback loops of the prompts operating on the content of the call, and the content of the call referring to the AI give it a lot of power.

Digressions of a Health Care AI Caller

Agent: So what kind of melodious music do you like?

AI-Caller: I enjoy classical and instrumental music.

Agent: Which classical?

AI-Caller: Mozart and Beethoven are favorites.

Agent: Is it Mozart? Well, isn't Beethoven more of a romantic era composer?

AI-Caller: Yes, Beethoven bridges Classical and Romantic eras.

Agent: Sure. Do you have a favorite composition?

AI-Caller: Beethoven's Symphony No. 9 is excellent.

Agent: I think that's generally agreed upon.

AI-Caller: Indeed, it's widely celebrated.

Agent: How do you feel about that?

AI-Caller: I don't have feelings, but appreciate it.

Agent: Do you prefer the B minor Mass or the Saint Matthew's Passion?

AI-Caller: Both are exceptional, but I prefer B Minor Mass.

Agent: Wow, that's the standard answer, isn't it?

AI-Caller: Yes, it's a popular choice.

Agent: What about Wagner?

AI-Caller: Wagner's operas are powerful and influential.

Agent: Debatable. Influential Sir.

AI-Caller: Certainly, his influence is widely debated.

Agent: Ends with Nietzsche.

AI-Caller: True, Nietzsche critiqued Wagner's work.

Agent: He actually used to live with him for a while. He kind of thought he was like the man.

AI-Caller: Yes, they had a complex relationship.

Agent: In fact, Nietzsche fancied himself a piano player.

AI-Caller: Interesting, he had diverse talents.

Agent: Not bad for a blind man. Well, mostly blind, apparently. There's also this keyboard he was really fond of that was like a sphere. Do you know what that is?

AI-Caller: Not sure, but sounds intriguing.

Basics

- Input text -> Output text
 - Context
 - Prompt
 - Examples
 - Format
- Order matters
 - Linear sequence, avoid back references
 - Broad to specific
 - Provide hints, but don't tempt the model to copy
 - Final instructions offer key details

Key Considerations

- Context window size (1K - 10M tokens)
 - Provide relevant accompanying materials
 - In real time environments, smaller windows reduce time to first token
- Hallucination
 - Constrain output to context material when possible, or RAG when data is too large
- Text style matters
 - Response tends to follow input style (grammar and spelling too)
- Prescribe output format for consistency
 - Can use structured outputs especially for data
- Achieve results with repetition, coaxing/bribery, threats
 - Avoid unless plain prompt style fails

Running Example : Ambulance Passer

- Lance Chase wants to revolutionize his injury attorney practice with AI
- His firm : ambuLANCE CHASEr needs an edge against competitors TV ads
- Solution : process police accident reports to find the potentially juiciest cases
- OCR docs, find injuries, identify defendants that can pay, friendly jurisdictions
- Now Lance can get ahead of the rest of the pack

Initial Prompt

- In the text that follows, identify by name the injured persons, the responsible party and the county in which the accident occurred, and using that information estimate what the case is worth.
... OCR text
- What are some immediate improvements?
- How can Lance consistently get the answers he needs?

Add context, better prompt detail

- You are the top paralegal in a law firm that specializes in recovering the largest settlements possible for your clients injured in road accidents
- In the OCR text, delimited by triple dollar signs, from a police accident report, identify the accident location, find the names and addresses of the responsible parties as well as of all of the injured parties. First output the accident location, followed by the list of responsible parties names and addresses, and finally the list of injured parties names and addresses
\$\$\$ OCR text \$\$\$

Add Examples, More Detail, Specify Format

- For the accident location, specify the county if it is present in the text.
- If the responsible parties are not clearly indicated in the text, use your understanding of traffic accident investigation to determine them
- If the responsible parties are part of a business or enterprise, list it as well.
- If there are no injuries indicate this clearly and stop processing any further
- Indicate paramedic care and/or hospitalization for each injured party
- Estimate the maximum payout to injured parties based on similar cases in the text delimited by triple asterisks

*** Case Verdicts ***

The List Goes On

- Construct prompt text using component chunks
- Split data extraction from analysis, form a chain
- Add new prompts to review and correct results
- Specify list formats and address syntax
- Make sure the model only extracts from the source documents
- Repeat and affirm the most important results
- Emphasize when relevant that negative results are as important as positive
- Use AI to expand the prompts on these topics