OpenAI in PHP

How AI works in theory & how to use it in PHP

Prompt Engineering

Weekly Topics

1. How Generative AI & LLMs Work
2. **Prompt Engineering**
3. Retrieval-Augmented Generation (RAG)
4. Fine-Tuning a Model
5. Agents
6. Using VUMC's OpenAI API in PHP

Key Concepts

- Role, Context, Tasks
- Shot-Learning with Examples
  - Code Examples to Fill in
  - Good Directions
- Giving AI Memory with Prompt Chaining
- Task Decomposition
- Task Chaining
- Five Tips

Goal

- Get the optimal, desired results from the AI bot
- How can you do this in a way that's better than just conversational English?

Role, Context & Tasks

- Processing breaks up all prompts into three categories:
  - Role – whose voice the AI is imitating
  - Context – the relevant background of the request, which can include examples & training
  - Tasks – what the actual request is

Role

- *E.g.*, a child therapist, a leadership coach
- Asking the same question in a different voice can result in different responses:
  - *E.g.*, a parent, a competitor
- Making this role explicit saves you from misinterpretation
- OpenAI will consult different source data based on your role

Context

- Obviously, context plays a huge role in results
- Give the necessary backstory so that OpenAI can understand what you want
- Needs to include all relevant data and all assumptions

- *E.g.*, "you're coaching a third-year PhD student in a public-health graduate program."

Tasks

- This is when you act as a programmer dictating what the AI bot needs to do
- Subdivide complex tasks when necessary [more to come…]
- *E.g.*, “tell the graduate student what to expect to do to graduate.”

All Together

EDGE FOR SCHOLARS

As a leadership coach, you're you're coaching a third-year PhD student in a public-health graduate program. Tell the graduate student what to expect to do to graduate.

0-, 1- & Few-Shot Learning

- Sometimes, example outputs are helpful to shape a response
- No examples = 0-shot learning
- One example = 1-shot learning
- Multiple examples = few-shot learning
- Few-shot learning increases the likelihood of getting the response that you're looking for
- "For example, ..."

Shot Learning

EDGE FOR SCHOLARS

For example, warn the student about specific obstacles related to writing the dissertation.

Or warn the student about how to anticipate challenges in their oral defense.

Filling in Coded Values

- *E.g.*, with JSONs

For each name, provide a row in a JSON that looks something like:
{“record_id”: [RECORD_ID HERE],
“identifier_first_name”: [FIRST NAME HERE],
“identifier_last_name”: [LAST_NAME HERE],
“identifiers_complete: “2”}

Start the RECORD_ID at 1 and increase it by 1 with each new name.

Good Directions

- ***Clear directions trump analogies in shot learning***
- That is, spend more time focusing on clear task directions instead of coming up with good examples

Prompt Chaining - Memory

- From a development perspective, AI prompts do not have memory
- Therefore, you must provide all the relevant data in one given prompt

- **Remember**: AI chat bots might remember context in the background
- **Remember**: You can program in contextual memory in the background, too, but this requires some overhead. (“Summarize the conversation: …”)

Task Decomposition

- OpenAI does not handle complex tasks well
- Sometimes, you have to provide an “algorithm” of steps to do in the task
- *E.g.*, making a timeline of things to do before graduation:
  1. Tell me how to conduct a job search.
  2. Tell me how to write a dissertation.
  3. Tell me how to defend a dissertation.

Task Chaining

- Like with basic computer coding, you can take the outputs of one task and feed them into another task.
- *E.g.*,
  - "Figure out what competencies I need to meet for a job as a regional public health coordinator…
  - "And then for each competency, tell me what I need to do to gain experience for an employer."

Tip #1

- “Show Your Reasoning”

- For complex tasks, asking the bot to show its reasoning will increase the accuracy of the results
- And it will allow you to verify its accuracy
- *E.g.*,
  - “When did the Newman Society at Vanderbilt start?” 2007
  - “When did the Newman Society at Vanderbilt start and show your reasoning?” 2002, as shown in this webpage…

Tip #2

- “Think Step by Step”

- Similar approach to “show your reasoning” except better for algorithmic thinking
- *E.g.*, this approach increases accuracy with mathematics

Tip #3

- “Without Commentary…”

- OpenAI will often add comments to “pretty up” its output
  - *E.g.*, “The JSON is…”
- This can drive programmers crazy if they just want to know what the JSON is.
- Add “Without commentary, provide a JSON that …” to cut down on needless chatter.

Tip #4

EDGE FOR SCHOLARS

- Iterate & Refine

- Defining a perfect AI prompt takes time and iteration
- Even though there is an expense for testing, it's much cheaper (pennies) than a bad prompt used over and over again!

Tip #5

- Get User Feedback – the “human in the loop”

- Always expect a verification step to be necessary
- AI bots are not experts; they’re tools
- In coding, always provide human review for adjudication of results
- Warn users of risks; allow users to point out shortcomings!
- Otherwise, blind AI is legally & ethically dangerous!

Recap

- Role, Context, Tasks
- Shot-Learning with Examples
  - Code Examples to Fill in
  - Good Directions
- Giving AI Memory with Prompt Chaining
- Task Decomposition
- Task Chaining
- Five Tips

Weekly Topics

1. ~~How Generative AI & LLMs Work~~
2. ~~Prompt Engineering~~
3. Retrieval-Augmented Generation (RAG)
4. Fine-Tuning a Model
5. Agents
6. Using VUMC's OpenAI API in PHP

Next Week: Retrieval-Augmented Generation (RAG)

OpenAI in PHP

How AI works in theory & how to use it in PHP

How LLMs Work