Meta-prompting

(Summary from openai tutorials by JW 2.1.2025, sample code available)

Optimizing to production level accuracy is one of the biggest pain points developers experience when working with LLMs.

So much guidance for prompt engineering, RAG and fine tuning out there.
Figuring out which optimization you need to hillclimb on your evals can be a difficult problem to frame and solve.

Luckily, it appears to be one of the use cases that o1 is very capable of.
–
-> To use o1-mini to work with a set of evals to optimize prompt for a task and improve on eval scores. Using an intelligent model like o1 to iteratively improve the instructions of a less intelligent model.
–

Example use case: Routine generation

Common use case where existing knowledge base articles (KB) are written for human consumption, and are difficult for LLMs to follow reliably.

- O1 read the KB articles and convert them to routine, which is optimized for LLM usage.
- Then feed that routine to a 4o model, which will try and pass an eval with that routine. Doing this a number of times, and after every iteration going to hand the eval results and the routine used back to o1, which is going to try and improve the routine to solve the issues that it saw in the eval. This iterative process is known as Meta Prompting, where more intelligent model is used to improve the instructions or guidance for a less intelligent model and then use the eval results to iteratively improve them.

- **Step 1**: Generate 4o routine (prompt)
  - o1: Read “Flight Cansellations and Changes Policy” human policy
  - o1: Generate a 4o routine to implement that policy (promp + set of tools)
- **Step 2:** Evaluation
  - Develop methods to evaluate 4o implementing policy
- **Step 3:** Improve 4o routine
  - Iteratively
    - evaluate -> score
    - Provide o1
      - Initial human policy
      - Current 4o routine
      - Evaluation results
    - o1: improve the 4o routine to address issues in eveluation