Curated course notes

Building Systems with the ChatGPT API

Moderation

OpenAI has a moderation API to validate input/output

- Input: to reduce prompt injections.
    - Strip the user's prompt from delimiter you've specified in the System role.
    - Use a system prompt to check if the user is trying a prompt injection.
- Output: Ask a more advanced model to check the quality of a less advanced model.

Chaining prompt vs. Chain of thoughts

- For workflow with complex tasks, to keep track of the system state.
- Can use external tools after a chain.
- Reduces number of tokens > cheaper.
- Skip some chains of the workflow when not needed.

Evaluation

- To help with prompt injection, you can ask the model to summaries the user's input, or to extract intent and entities, rather than adding it to the final prompt.
- To reducing hallucinations, find relevant information and answer the question based on these (i.e. RAG).
- To prevent regression when tuning prompts (i.e. the new prompt works on new examples but not as much on old examples), run tuned prompt on all examples.

Structured output

- Tune prompts on a handful of examples (~10).
- Collect encountered "tricky" examples as a dev-set to tune prompts further.
- Develop metrics to measure performance on dev-set.
- To increase accuracy further, add random example to dev-set (e.g.,100) to tune prompts further.
- To get an unbiased estimate of the system, use a hold-out test-set.

Unstructured output

- Use openai Evals Framework
- Use a rubric (a list of criteria) and use it with a second API call to LLM to evaluate the output of the first API call (for more rigorous evaluation use a more powerful LLM to evaluate a less powerful one, for example GPT4 to evaluate 3.5 turbo).