Designing Chatbots with OpenAI's API: A Dive into Large Language Models and Prompt Engineering

Yelaman Abdullin

Macquarie University

yelaman.abdullin@hdr.mq.edu.au

June 19, 2023

About me

Education

- Bachelor in Information Systems, ENU, Astana, Kazakhstan(2014)
- Masters in Software Engineering, KFU, Kazan, Russia(2016)
- PhD in Computer Science, MQU, 2023 - 2026

Experience

- Senior Software Engineer (.Net, Python)
- ML Engineer
- CTO

Overview

- Introduction to Language Models
- Why Large Language Models (LLMs)
- Chatbot Development
- Prompt Engineering
- Personal Experience Insights
- Showcasing an Application
- Conclusion Questions

Introduction to Language Models

- **Definition:** Language Models (LMs) are statistical or machine learning models that understand, generate, and complete human-like text.
- **Working:** They predict the likelihood of a sequence of words, considering contextual information from surrounding words.
- **Evolution:** Models have evolved from simple ones like N-gram to complex models based on transformers architecture.
- **Capabilities and Limitations:** While they excel at tasks like text generation, completion, and translation, they can struggle with tasks that require reasoning over explicit knowledge or understanding nuanced context.

Why Large Language Models (LLMs)

- **Definition:** LLMs are more complex versions of language models trained on vast amounts of text.
- **Advantages:**
  - Improved understanding of complex language constructs.
  - Capable of generating more coherent and contextually relevant responses.
  - Enhanced ability to mimic human-like text generation.
- **Relevance:** Given their advanced capabilities, LLMs are becoming a cornerstone in various AI applications.
- **Examples:** GPT-4, LLaMa, Falcon, Bloom, RWKV.

Instruction Tuned LLMs and RLHF

- **Instruction Tuned LLMs:** A new approach to fine-tuning language models. It involves training models to follow instructions in the input, leading to more controlled model behavior.
- **RLHF (Reinforcement Learning from Human Feedback):** A method of training models by collecting human feedback and using it to create a reward model. Effective for tasks with no predefined correct answer.
- **Future of Language Model Tuning:** Instruction tuning and RLHF represent the next frontier in language model tuning.

Chatbot Development

- **The Role of LLMs:** LLMs serve as the underlying technology for creating sophisticated chatbots.
- **Development Process**
  - Defining the Purpose: Identify what tasks the chatbot will perform.
  - Designing the Conversation: Establish how the chatbot will interact with users.
  - Implementing and Training: Use APIs like OpenAI to implement and train the chatbot.
  - Testing and Iteration: Test the chatbot extensively and improve it based on feedback.
- **Fine-tuning**

Prompt Engineering

- **Definition:** Prompt engineering is the craft of designing input prompts to guide the model's output.
- **Importance:** How we phrase a prompt significantly affects the model's response.
- **Misconception:** Language models are not just question-answering models; they generate continuations of the input text.
- **Need for Prompt Engineering:** LLMs don't have a deep understanding of user intent. Prompt engineering bridges this gap.

Prompt Engineering

- **Understanding Model Behavior:** Explore different prompt structures and analyze their influence on the output.
  You are a chatbot that helps users to solve optimization problems
- **Explicit Instruction:** Add details about the desired format of the answer or ask the model to think through before responding.
  Do not provide explanations of math
- **Specificity:** To prevent over-generalization or under-generalization, be specific and detailed in prompts.
  Your user base primarily consists of individuals without any background in mathematics or Combinatorial Optimization, hence, your communication should be clear, jargon-free and engaging.
- **Iteration and Improvement:** Prompt engineering is an iterative process. Constantly test, learn, and refine.

Prompt Engineering

- **Variable Naming:** Descriptive variable names provide additional context, leading to more accurate responses. *NextPossibleSteps:[values] or UserDateOfBirth*
- **Delimitations:** Clear delimiters in the instructions help the model parse information better.(“‘,===, —)
  - Use for different components like context info from DB, variable values, etc.
  - Extend this practice to user messages to prevent potential injection attacks.
- **Structuring Input and Output:** Prompt engineering involves not just the text we feed but also how we structure and organize it.

Prompt Example

Examples

Strictly generate the next turn in the following format divided by "'
"'

Analysis
: {analysis_text} // Analyse the conversation. what is the overall picture of the task the user trying to solve? Make some careful assumptions on the task aspects, but make sure you are not relying on non-existing information and that your assumptions are precise and not too general. Make some clues for the next message. this block is not for a user
MessageToUser
: {next_message_text} // Next message to the user. this block for a user
"'

Demo

Demo

Conclusion & Questions

- **Recap:** Explored LLMs, chatbot development, and prompt engineering using the OpenAI API.
- **Continuing Journey:** The field of AI and language models is vast, rapidly evolving, and full of opportunities.
- **Questions:** Excited to engage in further discussion and answer any questions.

References and Usefull links

- The OpenAI Cookbook shares example code for accomplishing common tasks with the OpenAI API and many more.
- In ChatGPT Prompt Engineering for Developers course, you will learn how to use a large language model (LLM) to quickly build new and powerful applications.
- Awesome video of Andrej Karpathy explaining state of GPT in 2023
- Examples and Playground on OpenAI's website