Prompt Engineering

Need to learn: python skillset, ML, Deep Learning, LLM in deep learning, transfer learning

Different types of prompt

1. Open-ended -> detailed thoughtful
2. Closed-ended -> specific way or fixed options
3. Multi-part -> there are multiple parts to one question
4. Scenario-based -> real life scene or imagination
5. Opinion based -> ask for suggestions or opinions
6. Instructional
7. Priming, mixed, and many more

Deeplearning.ai course:
ChatGPT Prompt Engineering for Developers

(https://drive.google.com/drive/folders/1e6NnnULT24s043Kw--aoeXWRxYvn8291?usp=share_link)

How to create venv and save openai key in virtual environment:

Simple way: %env OPENAI_API_KEY=your_actual_api_key. (For current jupyter notebook session)
os.environ['OPENAI_API_KEY'] = your_actual_api_key. (For entire your system)

Create venv : python3. -m venv venv
First activate your venv : source venv/bin/activate
Edit bash profile : nano venv/bin/activate
Add environment variable : export OPENAI_API_KEY='your_actual_api_key'
Save and exit
Deactivate and re-activate venv : deactivate, then source venv/bin/activate

Principles for Effective Prompting:

1. **Write Clear and Specific Instructions:**
   - Use delimiters to indicate distinct parts of the input.
   - Ask for a structured output (e.g., HTML or JSON format).
   - Instruct the model to check whether conditions are satisfied.
   - Use few-shot prompting by providing examples before the task.
2. **Give the Model Time to Think:**
   - Specify steps required to complete a task.
   - Instruct the model to reason out its own solution before concluding.
   - Reframe queries to request a chain of relevant reasoning.
   - Be explicit about giving the model time for complex tasks.

Lecture 3: Iterative Prompt Development

- **Importance of Iteration:**
  - When building applications with large language models, the initial prompt used rarely ends up being the final one for the application.
  - The key is to have an iterative process for refining prompts to achieve the desired results.
- **Machine Learning Model Analogy:**
  - Analogous to the iterative process in machine learning development, where one refines the idea, implementation, and experiments to improve the model.
- **Example Task: Summarizing a Fact Sheet:**
  - Used the example of creating a product description for an online retail website based on a technical fact sheet for a chair.
- **Initial Prompt and Refinement:**
  - Started with an initial prompt to generate a product description.
  - Evaluated the output and refined the prompt based on the desired length, technical focus, and additional details like product IDs.
- **Guidelines for Refinement:**
  - Keep in mind best practices for clear and specific instructions.
  - If necessary, give the model time to think.
  - Refine the prompt based on the initial output to achieve better results.
- **HTML Formatting Example:**
  - Demonstrated a more complex prompt instructing the model to include an HTML-formatted table with

product dimensions.
- **Key Takeaways:**
  - Prompt development is an iterative process.
  - Effectiveness in prompt engineering is more about having a good process than knowing the perfect prompt.
  - For more mature applications, evaluating prompts against a larger set of examples may be necessary.
- **Next Steps:**
  - Encouraged users to play with Jupyter Code notebook examples and try different variations to observe results.
  - Teased the next video, focusing on the common use of large language models in summarizing text in software applications.

Lecture 4: Summarizing Text with ChatGPT

- **Overview:**
  - Large language models can be used to summarize text efficiently.
  - Application: Summarizing articles for quick content overview.
- **Summarizing Product Reviews:**
  - Demonstrated using a product review as an example.
  - Prompt instructs the model to generate a short summary in at most 30 words.
- **Customizing Summaries for Feedback:**
  - Modified prompt to focus on specific aspects for feedback (e.g., shipping or pricing).
  - Resulting summaries tailored to the designated areas of interest.
- **Extracting Relevant Information:**
  - Introduced the concept of extracting relevant information instead of summarizing.
  - Example: Extracting shipping-related information for feedback to the shipping department.
- **Workflow Example: Summarizing Multiple Reviews:**
  - Illustrated how to use the model in a workflow to summarize multiple reviews efficiently.
  - Prompted the model to summarize each review in a list.
- **Potential Applications:**
  - Emphasized the potential for applications with large amounts of text.
  - Enables quick understanding of content and efficient browsing.
- **Upcoming Topic: Making Inferences Using Text:**
  - Teased the next video's topic: making inferences using text.
  - Example scenario: Determining sentiment in product reviews.
- **Closing:**
  - Summarization as a powerful tool for managing and extracting insights from large volumes of text.
  - Encouraged envisioning applications where summarization can enhance user experience.

Lecture 5: Making Inferences with ChatGPT

- **Introduction:**
  - Inference involves tasks where the model analyzes text, extracting labels, names, sentiment, etc.
  - Traditional machine learning workflow for sentiment analysis involves collecting labeled data, training models, and deployment.
- **Advantages of Large Language Models:**
  - Large language models offer a more efficient approach.
  - Using a single model and API, various tasks like sentiment analysis can be performed with just a prompt.
- **Sentiment Analysis Example:**
  - Used a product review for a lamp as an example.
  - Prompted the model to classify the sentiment, receiving a positive sentiment classification.
- **Customizing Responses:**
  - Modified the prompt to receive a concise response (single word: positive/negative).
  - Streamlined output for easier post-processing.
- **Identifying Emotions:**
  - Demonstrated the model's ability to identify emotions in a review.
  - Emphasized potential applications for customer support.
- **Information Extraction:**
  - Explored information extraction, asking the model to identify the item purchased and the company that made it.
  - Formatted the response as a JSON object for easier processing.
- **Combining Multiple Tasks:**
  - Showed how to extract sentiment, identify anger, and extract item and brand information in a single prompt.
  - Efficiently extracted multiple fields from a piece of text.
- **Topic Inference:**
  - Introduced inferring topics from a long piece of text.

    - Example: Determining five topics discussed in a fictitious newspaper article.
- **Zero-Shot Learning Algorithm:**
    - Applied zero-shot learning to determine if specific topics are covered in a given text.
    - Highlighted the potential for quickly generating news alerts based on topic detection.
- **Recommendation for Robustness:**
    - Advised using JSON format for more robust output in a production system.
    - Encouraged viewers to experiment with modifying the prompt for improved functionality.
- **Closing:**
    - Highlighted the excitement of quickly building systems for complex natural language processing tasks using prompting.
    - Teased the next video on "Transforming," focusing on how text can be transformed, such as translation to different languages.

Lecture 6: Transformative Capabilities of ChatGPT

- **Introduction:**
    - Large language models, like ChatGPT, excel in transforming input text into various formats.
    - Applications include language translation, tone transformation, and format conversion.
- **Language Translation:**
    - Models trained on diverse sources enable language translation proficiency.
    - Examples: English to Spanish translation, identifying languages in a multilingual context.
- **Tone Transformation:**
    - ChatGPT can alter the tone of text, such as translating slang to a formal business letter.
    - Demonstrates adaptability in generating content for different audiences.
- **Format Conversion:**
    - Proficient in translating between formats like JSON to HTML, XML, or Markdown.
    - Example: Converting a Python dictionary from JSON to an HTML table.
- **Spell Check and Grammar Correction:**
    - Valuable for proofreading and correcting text, aiding non-native language use.
    - Iterative prompt development enhances reliability.
- **Review Enhancement:**
    - ChatGPT can proofread reviews, correct errors, and enhance content for specific styles.
    - Example: Transforming a review, ensuring APA style, and presenting the output in Markdown.
- **Conclusion:**
    - ChatGPT's transformative capabilities extend to diverse linguistic tasks.
    - Adaptable for translation, tone modification, format conversion, and content enhancement.

Lecture 7: Generating Text with Temperature Control

- **Introduction:**
    - Expanding is the task of using a language model to generate longer text based on a shorter input.
    - Acknowledgment of responsible use, emphasizing the potential for both positive and problematic applications.
- **Temperature Parameter:**
    - Introduced the "temperature" parameter, influencing the variety and randomness of the model's responses.
    - At temperature zero, the model chooses the most likely next word, ensuring predictability.
    - Higher temperatures introduce randomness, providing a wider variety of outputs.
- **Summary:**
    - Higher temperatures result in more random and creative outputs.
    - Temperature control allows developers to tailor the model's behavior based on the application's needs.
    - Recommended temperature zero for predictable responses and higher temperatures for creative applications.