Prompting Guide 101

LLM Settings:

1. Temperature:

Temperature controls the overall randomness of the model's output: high temperature increases diversity but decreases predictability, while a low temperature makes outputs more predictable and less varied.

2. Top-p (Nucleus Sampling):

Top-p (nucleus sampling) focuses on selecting words from a dynamic subset of the most probable options, where the cumulative probability exceeds a certain threshold (p). It allows for controlled diversity by excluding less likely words, maintaining a balance between creativity and relevance.

3. Max Length:

Max Length - You can manage the number of tokens the model generates.

4. Stop Sequence:

A stop sequence is a string that stops the model from generating tokens. Specifying stop sequences is another way to control the length and structure of the model's response. For example, you can tell the model to generate lists that have no more than 10 items by adding "11" as a stop sequence.

5. Frequency Penalty:

The frequency penalty applies a penalty on the next token proportional to how many times that token already appeared in the response and prompt. The higher the frequency penalty, the less likely a word will appear again. This setting reduces the repetition of words in the model's response by giving tokens that appear more a higher penalty. Presence Penalty discourages the model from mentioning the same topics or entities already covered.

6. Presence Penalty:

The main difference between Frequency Penalty and Presence Penalty is that Frequency Penalty targets word repetition, while Presence Penalty aims to diversify the content's topics and concepts.

Basics of Prompting:

Roles:

1. **System:**
   The system message is not required but helps to set the overall behavior of the assistant.

2. **Assistant:**
   The assistant message in the example above corresponds to the model response.

3. **User:**
   Used for prompting.

Prompt Formatting:

1. **Zero-Shot Prompting:**
   You are prompting the model for a response without examples of the task you want to achieve. It depends upon the complexity and knowledge of task at hand.

   Example prompt:

   <Question>?
   What is prompt engineering?

2. **Few-Shot Prompting:**
   One popular and effective technique to prompting is referred to as few-shot prompting where you provide examples.

   Example prompt:

- <Question>?
  <Answer>

- <Question>?
  <Answer>

- <Question>?
  <Answer>

- This is awesome! // Positive
- This is bad! // Negative
- Wow that movie was rad! // Positive
- What a horrible show! // Negative

Few-shot prompts enable in-context learning, which is the ability of language models to learn tasks given a few demonstrations.

Elements of Prompt:

A prompt contains any of the following elements:

1. **Instruction:**
   A specific task or instruction you want the model to perform

2. **Context:**
   External information or additional context that can steer the model to better responses

3. **Input Data:**
   The input or question that we are interested to find a response for

4. **Output Indicator:**
   The type or format of the output.

Examples:

- Classify the text into neutral, negative, or positive //instruction
- Text: I think the food was okay. //input
- Sentiment: //output

General Tips for Designing Prompts:

1. You can design effective prompts for various simple tasks by using commands to instruct the model what you want to achieve, such as "Write", "Classify", "Summarize", "Translate", "Order", etc.
2. Another recommendation is to use some clear separator like "###" to separate the instruction and context.
3. The more descriptive and detailed the prompt is, the better the results.
4. Another common tip when designing prompts is to avoid saying what not to do but say what to do instead.

Techniques:

1. Zero-shot Prompting
2. Few-shot Prompting
3. Chain of Thought Prompting