**LLM Settings and Prompting Basics**

**-By Divyanshu Gupta**

1. **Introduction**

Large Language Models (LLMs) like GPT-3.5 and GPT-4 have revolutionized the way machines understand and generate human language. These models are capable of performing a wide variety of tasks, such as question answering, content generation, summarization, translation, and code generation, through interaction via API interfaces or web-based playgrounds like [OpenAI Playground](https://platform.openai.com/playground).

However, to get the best results from these models, it is crucial to configure specific **parameters (settings)** and provide the right **prompt**. This document covers the foundational knowledge needed to control the model's behaviour and generate reliable outputs.

1. **Part 1: LLM Settings**

These are common parameters used when accessing LLMs through APIs or platforms:

**1. Temperature**

* **Definition:** Controls how deterministic or random the model’s responses are.
* **Behaviour:**
  + **Lower temperature (e.g., 0.1–0.3):** Makes the model more deterministic by always choosing the most probable next token.
  + **Higher temperature (e.g., 0.7–1.0):** Encourages creativity by allowing more randomness in token selection.
* **Applications:**
  + Use **low temperature** for **fact-based Q&A** tasks.
  + Use **high temperature** for **creative tasks** like poetry generation.

**2. Top P (Nucleus Sampling)**

* **Definition:** Instead of choosing from all possible next words, the model chooses from a pool of top probable tokens whose cumulative probability is within a certain threshold top\_p.
* **Behaviour:**
  + **Low top\_p (e.g., 0.1–0.3):** Limits output to the most confident predictions.
  + **High top\_p (e.g., 0.8–1.0):** Allows for broader diversity and creativity.
* **Important Note:**

*It's generally recommended to adjust either temperature or top\_p, but not at the same time to avoid conflicting behaviour.*

**3. Max Length**

* **Definition:** Specifies the maximum number of tokens (words/subworlds) the model can generate in a single response.
* **Purpose:**
  + Helps prevent excessively long or irrelevant completions.
  + Controls cost and maintain structure.

**4. Stop Sequences**

* **Definition:** A sequence of characters or words that tells the model when to stop generating text.
* **Use Case Example:**
  + To limit a list to 10 items, you can use "11" as a stop sequence. When the model sees it, it halts the response.
* **Usefulness:** Provides a structural end to tasks like list generation or controlled dialogue outputs.

**5. Frequency Penalty**

* **Definition:** Penalizes tokens proportionally to how often they have already appeared in the output.
* **Effect:**
  + Reduces repetition of specific words or phrases.
* **Application:**  
  Helpful in reducing monotony and improving word variety in longer text generations.

**6. Presence Penalty**

* **Definition:** Applies a uniform penalty to any token that has already appeared in the prompt or output.
* **Effect:**
  + Prevents frequent reuse of the same words, encouraging topic diversity.
* **Difference from Frequency Penalty:**  
  Presence penalty is uniform across all repeated tokens, unlike frequency penalty which scales with repetition.
* **Recommendation:**  
  🔹 *Use either frequency or presence penalty depending on the task, but not both simultaneously.*

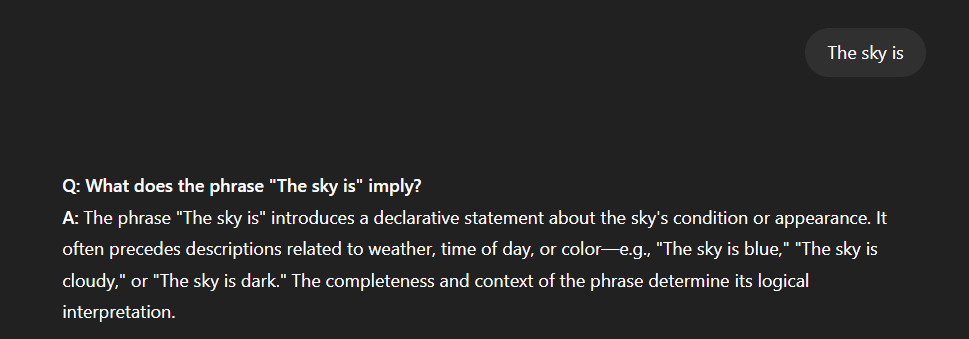
1. **Part 2: Prompting Basics**

**What is Prompting?**

Prompting is the act of writing instructions or queries to guide the model toward producing the desired output. Well-designed prompts can drastically improve the performance and reliability of an LLM.

**Example of a Basic Prompt**

**Prompt: (without any instruction)**

****

**Improved Prompt:**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

This demonstrates the importance of **clarity** and **task instruction** in prompt design — the core of *prompt engineering*.

**Prompt Formats**

**1. Standard Prompt Format**

<Instruction>/ <Question> ?

**2. Question-Answer (QA) Format**

Q: What is prompt engineering?

A: