Foundations of Prompt Engineering

Definition

- The practice of crafting inputs to guide Large Language Models (LLMs) toward accurate and relevant outputs.

Core Structure

- Instructions: Explicitly defining the task or objective for the model.
- Context: Providing essential background information to improve relevance.
- Input Data: Supplying the specific subject matter or raw content to be processed.
- Output Indicators: Establishing strict constraints or requirements for the desired format.

Prompting Strategies: From Zero to Few-Shot

Techniques for guiding model responses through example-based prompting

Zero-shot

The model performs a task without any provided examples, relying solely on internal training and intent inference.

One-shot

A single targeted example is provided to illustrate the desired output format or logic, creating a baseline for the model to follow.

Few-shot

Multiple examples are provided to clarify complex tasks, establish specific stylistic constraints, and enhance performance consistency.

Role, System, and Context Engineering

Foundational techniques for optimizing Large Language Model performance

Role Prompting

Assigning a specific persona, such as 'Senior Data Scientist', to prime the model's tone, expertise, and analytical framework.

System Prompts

Defining high-level instructions to establish absolute behavioral boundaries, operational constraints, and the core identity of the model.

Context Engineering

Providing background information, constraints, and source documentation to ground outputs and minimize hallucinations.

Advanced Techniques: Chaining & Reasoning

Optimizing model performance through architectural strategies

- **Prompt Chaining**

  Decomposes complex objectives into granular, sequential prompts to improve modularity and specialized instruction handling.

- **Reasoning Techniques**

  Utilizes Chain-of-Thought (CoT) to surface latent logic and ReAct to integrate reasoning with external actions.

- **Refinement Loops**

  Employs self-critique and iterative refinement cycles to systematically verify and enhance output quality.

Structured Outputs and Prompt Optimization

Enhancing model reliability and efficiency through standardized engineering practices

- **Structured Formats**

  Enforcing JSON or Markdown outputs for seamless programmatic integration, reducing parsing errors and ensuring data consistency.

- **Prompt Templates**

  Reusable blueprints for standardized tasks, improving consistency in model behavior and accelerating development cycles.

- **Optimization**

  Evaluating performance against metrics and refining prompts based on empirical data to maximize accuracy.

Security: Injection and Jailbreaks

Understanding threats and implementing defense mechanisms for AI models

Prompt Injection

Attackers attempt to override original system instructions to force unintended behaviors or data retrieval.

Jailbreaks

Sophisticated attempts to bypass built-in safety filters and guardrails.

Key Mitigations

Utilize input validation, system prompt hardening, and real-time output monitoring to safeguard model interactions.

Mini Project: AI Library & Assistant

Streamlining workflows through structured prompts and custom AI agents

Curated Prompt Library

A repository of optimized, field-tested prompt templates designed for repeatable task efficiency.

Custom AI Agent Configuration

A specialized assistant equipped with defined constraints, custom personas, and system instructions.

Structured Technical Output

Enforced JSON formatting to ensure seamless integration with existing technical workflows.