Day 2

Prompt Engineering

- Focus on essential prompt engineering techniques
- Understand the function of the context window size
- Develop effective prompts for better model responses
- Explore advanced prompting strategies during Day 2

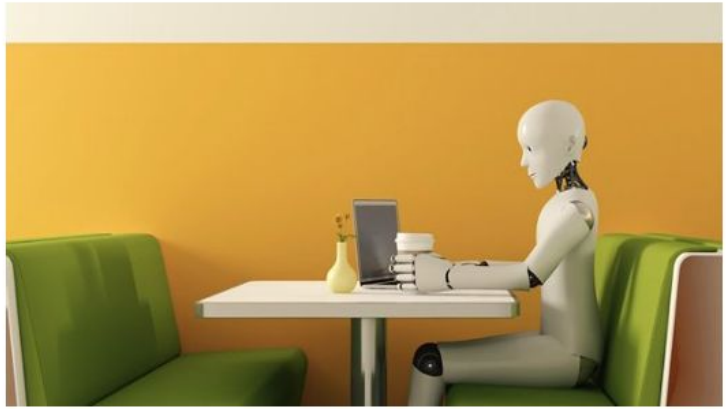
What is Prompt Engineering?

- Crafting and refining inputs for large language models
- Involves structuring prompts to guide model behavior
- Focuses on achieving desired, high-quality model outputs
- Essential skill for maximizing the utility of LLMs

Prompt Basics

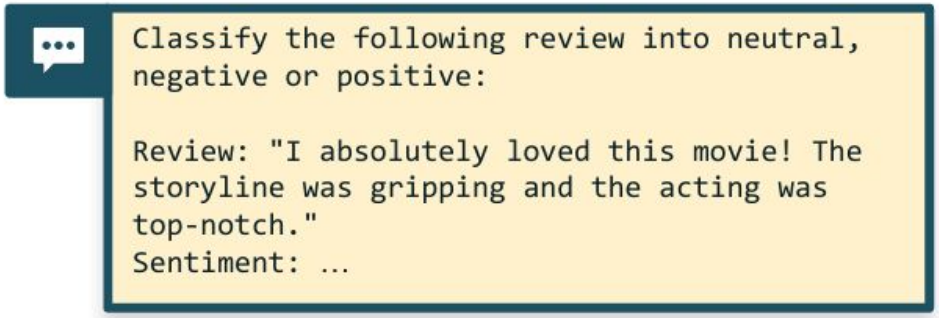
Definitions

- **Prompt**: An AI prompt is an **input or query** given to a large language model to elicit a specific response or output.
- **Prompt Engineering**: Prompt engineering is the practice of **designing and refining prompts** to optimize the responses generated by AI models



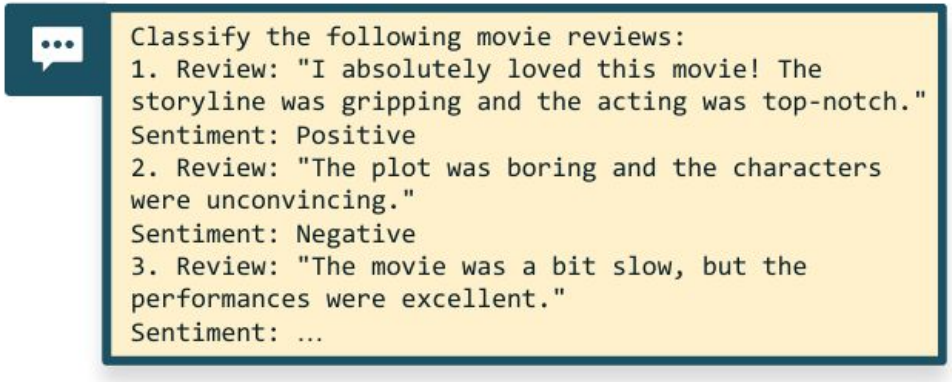
Zero-Shot Prompting Explained

- Zero-shot prompting provides an instruction only
- The LLM uses its training to answer the request
- No specific examples are given within the prompt
- It relies solely on the model's existing knowledge



Zero-Shot Prompting Explained

Few-Shot Prompting Explained



- Providing examples guides the model's response
- Few-shot prompting includes several input-output pairs
- Improves accuracy and helps model learn specific tasks
- Examples are shown directly within the prompt's context

Few-Shot Prompting Explained

Zero-Shot vs. Few-Shot Prompting: A Comparative Guide

Zero-Shot Prompting

Best for...
- General knowledge tasks, creative writing, simple summaries.

Speed
Fast
Faster to write (less typing).

Cost (Tokens)
Cheaper (shorter prompts).

Reliability
Moderate. The model might guess the format wrong.

Few-Shot Prompting

Best for...
- Specific formats, complex logic, mimicking a specific style/voice.

Speed
Slow
Slower to write (need to curate examples).

Cost (Tokens)
More expensive (examples use up token context).

Reliability
High. The model follows the pattern provided.

Choosing the right prompting strategy depends on your task's complexity and the desired output precision.

Building a Prompt Framework

Prompt Basics

Prompt components

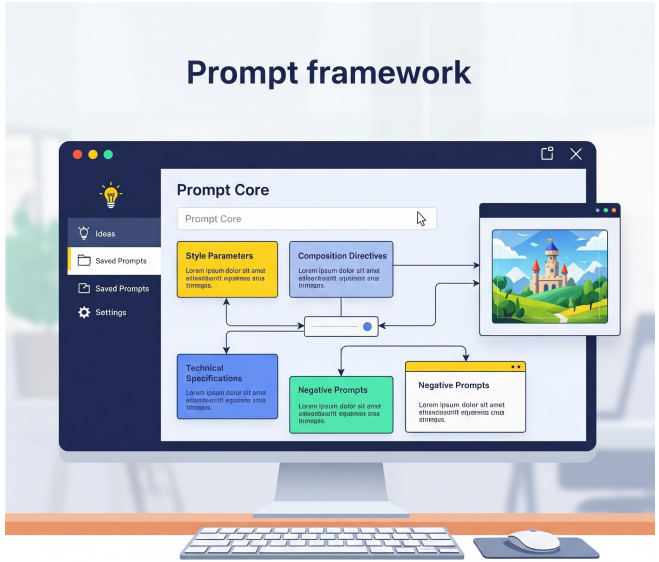
A good prompt usually consists of:

- **Instruction**: A clear directive that specifies what the model should do.
- **Context**: Background/additional information that provides the necessary details to understand the task.
- **Input / question**: The specific query or data that the model needs to process.
- **Output type / format**: The desired structure or style of the response generated by the model.

```
Summarize the following
article and write a list of
top 3 important points in
markdown format that answer
the following question;
“How is climate change
affecting polar bear habitats
and their ability to find
food?”

Article:
<The article discusses the
impact of climate change on
polar bear populations in the
Arctic.>
```

Building a Prompt Framework

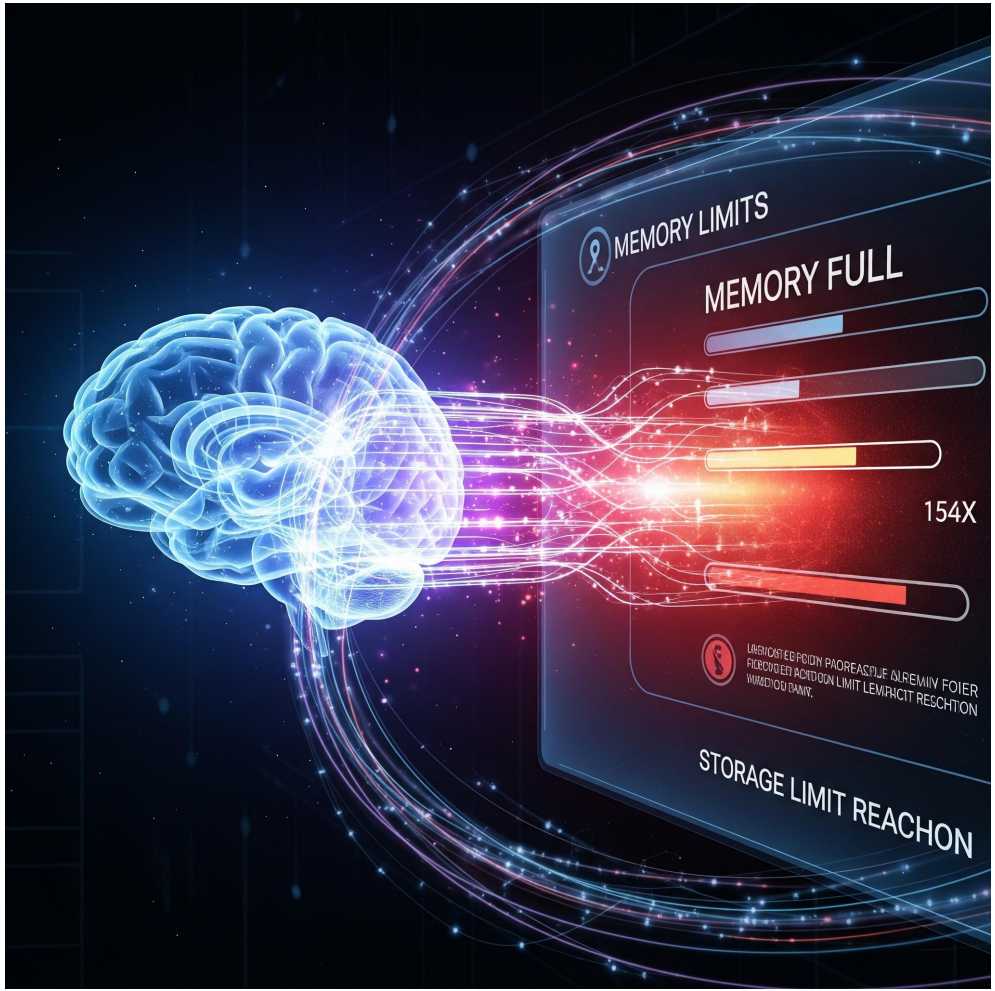


To perform Few-shot prompting effectively, you need a structure. Just throwing text at the model can confuse it. Use the **Standard Prompt Structure**:

1. **Role (Optional):** Who is the AI?
2. **Instruction:** What is the task?
3. **Examples (The "Shots"):** The pattern to follow.
4. **Context/Constraint:** Any guardrails?
5. **Input Data:** The actual thing to process.
6. **Output Indicator:** A cue for the AI to start.

Context Window Deep Dive

- The context window is the model's short-term memory
- It defines the amount of text the model can process
- Window size limits the input and output length
- Exceeding the window causes the model to forget information



What happens if the context window is exceeded ?

Hands on .....