Week–2 Assignment

Lokareddy Richitha.

**Task 1: Advanced Prompt Engineering Techniques**

**Prompting Techniques**

Prompt Engineering helps to effectively design and improve prompts to get better results on different tasks with LLMs.

Here are few advanced prompt engineering techniques:

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

Zero-shot Prompting:

Zero-shot prompting is a technique that leverages the generalization ability of large language models (LLMs) to perform tasks without any prior examples or demonstrations. Instead, the model is given a clear and concise instruction, relying solely on its pre-trained knowledge to respond appropriately.

This approach highlights how well LLMs can apply their understanding to a wide range of tasks using only the information in the prompt. Whether the goal is to answer a question, generate content, or classify data, zero-shot prompts allow the model to operate flexibly across domains.

Let’s look at an example:

Prompt: *Identify the fruit based on the description. This fruit is small, red, has tiny seeds on its outer skin, and is sweet in taste.*

Output: *Strawberry.*

In this case, the model receives a clear task—*identify the fruit*—and a brief description. Even though no examples or categories were provided beforehand, the model accurately responds using its internal knowledge about fruit characteristics. The result shows how effectively zero-shot prompting can guide LLMs with nothing more than a well-phrased instruction.

## Applications of Zero-Shot Prompting:

1. Text generation:

Zero-shot prompting works great in various text generation tasks, allowing models to produce diverse types of content without specific training. Some include summarization, creative writing, and translation.

1. Classification and sentiment analysis

Zero-shot prompting can be used for various classification tasks, including topic classification, sentiment analysis, and intent classification, without needing labeled training data.

1. Question answering

Zero-shot prompting gives models the ability to answer many different questions, such as factual, explanatory, and comparative questions, based on their general knowledge and understanding of language.

Few-shot prompting:

Few-shot prompting is a method where a language model is given a few examples within the prompt to demonstrate the desired task. These examples act as guidance, helping the model understand both the task and the format of the expected output before it handles a new input.

This technique is useful when the task may be ambiguous or when a specific output structure is needed. By seeing multiple examples, the model can more accurately imitate the behavior shown and generate consistent results.

Let’s go over an example:

Prompt:

Classify the sentiment of the third movie review. Use the information from the first two examples:

Review: "This movie was a waste of time."

Sentiment: Negative

Review: "I couldn't stop laughing throughout the film!"

Sentiment: Positive

Review: "The special effects were amazing, but the plot was confusing."

Sentiment:

Output:

Sentiment: Mixed

In this case, the model is given two examples that clearly connect movie reviews to their sentiment (positive or negative). With that pattern established, the model is then asked to classify a new review. The third review contains both praise and criticism, and the model correctly identifies the overall sentiment as **Mixed**. This demonstrates how few-shot prompting helps guide the model's understanding by showing it how the task should be handled before asking it to respond.

### Applications of Few-Shot Prompting:

Language translation and text formatting:  
 Few-shot prompting is useful for tasks where the model needs to follow a specific structure or style. By showing a few examples, it can translate languages, rephrase sentences, or format text consistently.

Text classification with domain-specific cues:  
 When dealing with niche or domain-specific tasks (like medical sentiment classification or legal text tagging), a few examples help guide the model to understand domain language and label content correctly.

Task adaptation and customization:  
 Few-shot prompting enables quick adaptation to new tasks—such as custom chat replies, summary styles, or recommendation formats—by simply providing a few demonstrations without requiring model retraining.

Chain-of-thought Prompting:

Chain-of-thought prompting is a technique where the model is encouraged to show its reasoning steps before giving the final answer. Instead of jumping straight to a result, the model walks through the logic or thought process required to solve the task. This helps in solving complex problems that require multi-step reasoning, such as math word problems, logic puzzles, or decision-based tasks.

By prompting the model to "think aloud," it becomes more likely to produce accurate and interpretable answers, especially when multiple steps are involved.

Let’s look at an example:

Prompt:

Question: If there are 12 apples and you give 5 to your friend, then buy 3 more, how many apples do you have now? *Let’s think step by step*

Output:

First, you start with 12 apples.

You give away 5 apples, so you have 12 - 5 = 7 apples left.

Then you buy 3 more apples, so now you have 7 + 3 = 10 apples.

Answer: 10

In this example, the phrase *“Let’s think step by step”* prompts the model to break the problem into parts and explain the reasoning clearly. Instead of giving just the answer ("10"), the model shows **each step** it used to reach the conclusion. This technique improves reliability and is especially valuable when solving tasks that could otherwise lead to errors if rushed.

### Applications of Chain-of-Thought Prompting:

Mathematical and logical reasoning:  
 Chain-of-thought prompting is highly effective for multi-step math problems, logical puzzles, and scenarios requiring calculation or deduction. It helps break down complex problems into manageable steps.

Commonsense and situational reasoning:  
 In tasks where understanding everyday situations or human behavior is key (e.g., "What should someone do if they spill water near electronics?"), showing reasoning improves decision-making accuracy.

Multi-hop question answering and explanation tasks:  
 For questions that require combining multiple facts (multi-hop QA) or giving a reasoned explanation (e.g., "Why is X better than Y in this case?"), chain-of-thought prompting helps produce transparent, step-by-step responses.

**Task 2: Design and Experiment**

Description:

Prompt Design Experiment: -Select an AI model available to you for experimentation (e.g., GPT-3.5 or newer versions, if accessible). -Design a set of prompts that illustrate zero-shot, few-shot, and chain-of-thought techniques. Choose a complex topic or task for these prompts (e.g., solving a math problem, explaining a scientific concept, or providing advice on a niche topic). -Document the AI's responses to each prompt type and analyze the effectiveness and limitations observed in each approach.

The three prompt styles:

1. zero-shot:

Why does the sky appear blue?

1. few-shot:

Q: Why do leaves change color in the fall?

A: Leaves change color in the fall because chlorophyll breaks down, revealing pigments like carotenoids (yellow and orange) and anthocyanins (red and purple).

Q: Why do we get sunburns?

A: Sunburns occur because ultraviolet (UV) rays from the sun damage the skin cells, causing inflammation and redness.

Q: Why does the sky appear blue?

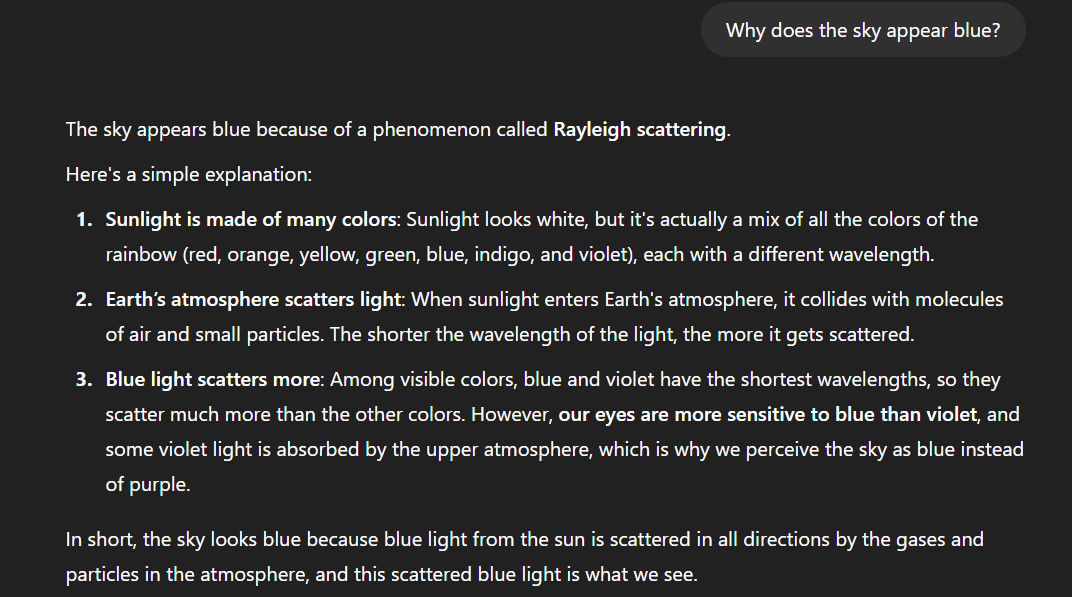
A:

1. chain-of-thought:

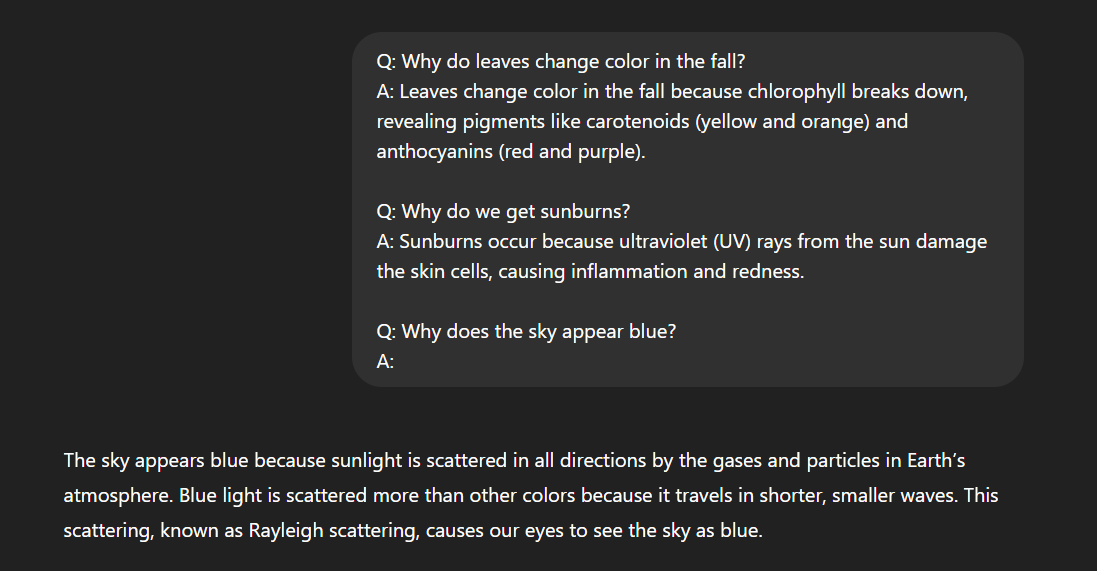
Why does the sky appear blue? Let's think step by step.

AI Response:

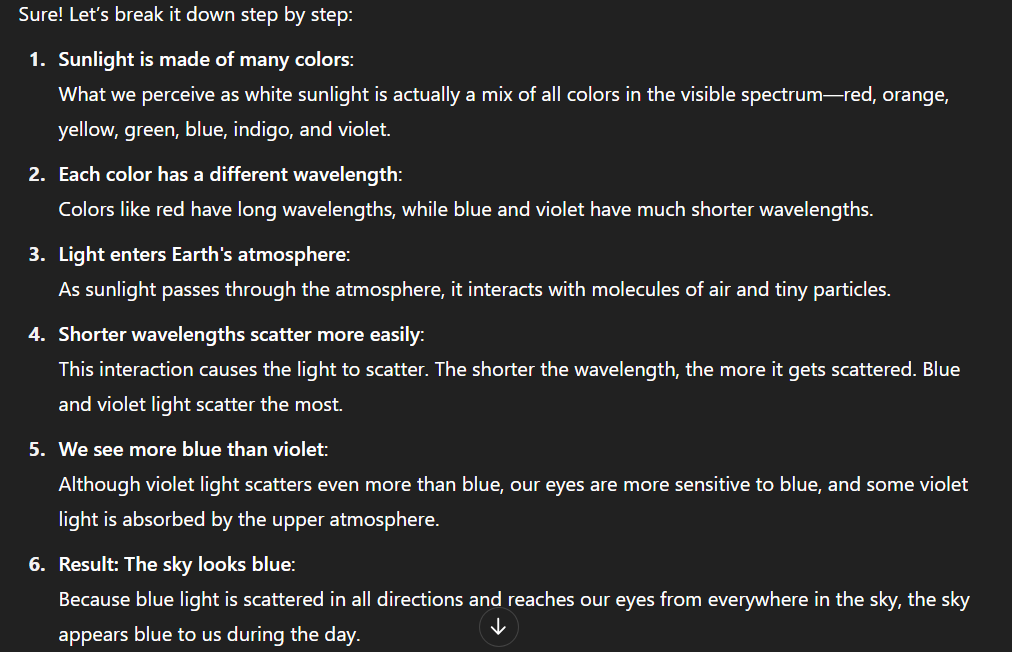
1. zero-shot:



1. Few-shot:



1. Chain-of-thought:



Analysis:

|  |  |  |  |
| --- | --- | --- | --- |
| **Technique** | **Accuracy** | **Explanation Quality** | **Use Case** |
| Zero-shot | Good | Just final answer | Simple tasks with known format |
| Few-shot | Good | Matches pattern | When you want consistency |
| Chain-of-Thought | Good | Best explanation | Complex or educational tasks (math, logic) |

Conclusion:

Chain-of-Thought prompting was most useful for learning-based tasks.