Lecture 03: Prompt Engineering & Language Reasoning

Learning Objectives

By the end of this lecture, you should be able to:

- Understand how prompts influence LLM behavior.
- Craft effective prompts for zero-shot, one-shot, and few-shot learning.
- Use techniques like Chain-of-Thought (CoT) to improve reasoning.
- Evaluate prompt-based reasoning in real tasks.

Key Concepts

Prompt Engineering

- The art of designing input text that guides the LLM to produce useful outputs.
- Common prompting styles:
  - **Zero-shot**: No examples provided.
  - **One-shot**: One example shown.
  - **Few-shot**: Multiple examples given to illustrate the desired behavior.

Chain-of-Thought Prompting

- Encourages step-by-step reasoning by including reasoning traces in examples.
- Improves performance on arithmetic, logic, and multi-step questions.

Role of Prompting in Agentic Systems

- Prompts encode goals, context, memory, and next actions.
- Serve as a lightweight alternative to reprogramming models.

Prompt Patterns

- Instruction-based: "Summarize this paragraph..."
- Role-based: "You are a legal assistant..."
- Delimiter-based: Use separators (e.g., `###`, `<context>`, etc.) to structure inputs.

Required Tools/Libraries

- OpenAI API (or any other LLM provider)
- Python
- (Optional) LangChain for prompt templates

Hands-on Exercise: Prompt Experiments

Goal: Explore how prompting changes LLM output and reasoning quality.

Step 1: Test Zero-shot Prompt

```
import openai

openai.api_key = "YOUR_API_KEY"

response = openai.ChatCompletion.create(
    model="gpt-4",
    messages=[
        {"role": "user", "content": "What is the capital of France?"}
    ]
)

print(response['choices'][0]['message']['content'])
```

Step 2: Few-shot Example Prompt

```
Q: What is the capital of Italy?
A: Rome

Q: What is the capital of Germany?
A: Berlin

Q: What is the capital of Japan?
A:
```

Step 3: Chain-of-Thought Prompt

```
Q: If I have 5 apples and give away 2, how many are left?
A: I started with 5 apples. I gave away 2. So, 5 - 2 = 3 apples left.
```

Bonus:

- Try different prompt styles (bullets, step numbers, roles).
- Observe hallucinations or failure cases and refine the prompt.