📚 Module 1: Foundations of Prompt Engineering

> ***Weeks 1-3*** | *The essential building blocks*

Week 1-2: Introduction to LLMs and Prompt Engineering

🤖 What Are Large Language Models?

Theory: Large Language Models (LLMs) are AI systems trained on massive amounts of text data to understand and generate human-like text. They predict the most likely next words based on the input they receive.

Key Concepts:

Concept	Description
Training	Models learn patterns from billions of text examples
Parameters	Billions of numerical values that encode knowledge
Inference	Using the trained model to generate responses
Generative	Creates new text rather than just classifying

Popular LLMs:

- **Claude** (Anthropic) - Known for safety and helpfulness
- **GPT-4** (OpenAI) - Versatile and widely used
- **Gemini** (Google) - Multimodal capabilities
- **LLaMA** (Meta) - Open-source foundation

🔤 Understanding Tokens

Theory: Tokens are the fundamental units LLMs process. A token can be a word, part of a word, or punctuation.

```
Example:
"Hello, world!" → ["Hello", ",", " world", "!"]  (4 tokens)
"Tokenization" → ["Token", "ization"]  (2 tokens)
```

Why Tokens Matter:

- **Cost** - Most APIs charge per token
- **Limits** - Context windows have token limits
- **Processing** - LLMs think in tokens, not words

Approximate Token Counts:

Text	Approx. Tokens
1 word	~1.3 tokens
1 page	~500 tokens

1000 words	~1300 tokens

📊 Context Windows

Theory: The context window is the maximum amount of text (input + output) an LLM can process at once.

Model	Context Window
GPT-3.5	4K - 16K tokens
GPT-4	8K - 128K tokens
Claude 3	200K tokens
Gemini 1.5	Up to 1M tokens

Best Practices:

```
✅ DO:
- Keep prompts concise when possible
- Prioritize important information at the start
- Summarize long documents before processing

❌ DON'T:
- Waste tokens on unnecessary repetition
- Ignore context limits
- Assume all context is used equally
```

🌡️ Temperature and Sampling

Theory: Temperature controls the randomness of LLM outputs.

Temperature	Behavior	Use Case
0.0 - 0.3	Deterministic, focused	Code, facts, analysis
0.4 - 0.7	Balanced	General tasks
0.8 - 1.0	Creative, varied	Stories, brainstorming
> 1.0	Highly random	Experimental only

Example:

```
Prompt: "Give me a name for a coffee shop"

Temperature 0.2: "The Daily Grind"
Temperature 0.7: "Mocha Dreams Café"
Temperature 1.0: "The Quantum Bean Paradox Emporium"
```

🎯 The Role of Prompt Engineering

Definition: Prompt engineering is the practice of designing inputs that guide LLMs to produce desired outputs.

Why It Matters:

```
Poor Prompt:
"Write about dogs"

Better Prompt:
"Write a 200-word informative paragraph about the
health benefits of owning a dog, citing at least
two scientific studies."
```

The Prompt Engineering Mindset:

1. **Be Specific** - Clear instructions get clear results
2. **Provide Context** - Help the model understand the situation
3. **Set Expectations** - Define the output format
4. **Iterate** - Test and refine your prompts

⚖️ Ethical Considerations

Key Principles:

Principle	Description
Transparency	Be clear when AI is being used
Accuracy	Don't use AI to spread misinformation
Privacy	Don't input sensitive personal data
Bias Awareness	LLMs can reflect training data biases
Responsibility	Human oversight for important decisions

Best Practices:

```
✅ Always verify factual claims
✅ Disclose AI-generated content when appropriate
✅ Don't use for deception or manipulation
✅ Consider potential misuse of your prompts
✅ Respect copyright and intellectual property
```

Week 3: Basic Prompting Techniques

📝 Clear and Specific Instructions

The Golden Rule: Tell the model exactly what you want.

❌ **Vague Prompt:**

```
Summarize this article.
```

✅ **Specific Prompt:**

```
Summarize this article in 3 bullet points, focusing on:
1. The main argument
2. Key supporting evidence
3. The conclusion

Keep each bullet under 25 words.
```

Specificity Techniques:

Technique	Example
Length	"Write exactly 100 words"
Format	"Use bullet points"
Tone	"Write in a professional tone"
Audience	"Explain for a 10-year-old"
Constraints	"Don't use jargon"

🖼️ The Importance of Context

Theory: Context helps the model understand the situation and produce relevant responses.

Without Context:

```
Prompt: "How do I fix it?"
Response: [Confused, asks for clarification]
```

With Context:

```
Prompt: "I'm a Python developer. My code throws a
TypeError when I try to concatenate a string and
integer. How do I fix it?"

Response: "Use str() to convert the integer:
result = 'Value: ' + str(number)"
```

Context Framework (CRISP):

```
C - Context: Background information
R - Role: Who should the AI act as
I - Intent: What you want to achieve
S - Specifics: Detailed requirements
P - Presentation: Output format
```

🔄 Single-Turn vs Multi-Turn Conversations

Single-Turn: One prompt, one response. Best for standalone tasks.

```
User: "Translate 'Hello' to French"
AI: "Bonjour"
```

Multi-Turn: Ongoing conversation with memory. Best for complex tasks.

```
User: "I want to plan a trip to Japan"
AI: "Great! When are you planning to visit?"

User: "Next spring, for 2 weeks"
AI: "Perfect timing for cherry blossoms! What's your budget?"

User: "Around $3000"
AI: [Provides detailed recommendations]
```

When to Use Each:

Single-Turn	Multi-Turn
Quick translations	Trip planning
Code generation	Debugging sessions
Fact lookup	Tutoring
Summarization	Creative writing

⚠️ Common Pitfalls and How to Avoid Them

Pitfall 1: Ambiguous Instructions

```
❌ "Make it better"
✅ "Improve the clarity by simplifying complex sentences
   and adding transition words"
```

Pitfall 2: Overloading the Prompt

```
❌ Asking 10 things in one prompt
✅ Break into smaller, focused requests
```

Pitfall 3: Assuming Prior Knowledge

```
❌ "Continue from where we left off" (new session)
✅ Provide necessary context in each session
```

Pitfall 4: Not Specifying Format

```
❌ "List some options"
✅ "List 5 options in a numbered list with brief
   descriptions (1-2 sentences each)"
```

Pitfall 5: Ignoring Edge Cases

```
❌ "Extract the email from this text"
✅ "Extract the email from this text. If no email
   is found, respond with 'No email found'"
```

💡 Key Takeaways

1. **LLMs predict text** - They work with tokens, not meaning
2. **Temperature controls creativity** - Low for facts, high for creativity
3. **Context is crucial** - Provide background information
4. **Be specific** - Vague prompts give vague results
5. **Iterate** - Good prompts are developed, not written once

🎯 Quick Reference Card

```
PROMPT STRUCTURE TEMPLATE:

[ROLE/PERSONA]
You are a [role] with expertise in [domain].

[CONTEXT]
Background: [relevant information]
Situation: [current scenario]

[TASK]
Your task is to [specific action].

[REQUIREMENTS]
- Requirement 1
- Requirement 2
- Format: [desired output format]

[CONSTRAINTS]
- Don't [limitation]
- Maximum [length/count]
```

Next: Module 2 - Core Prompting Strategies →