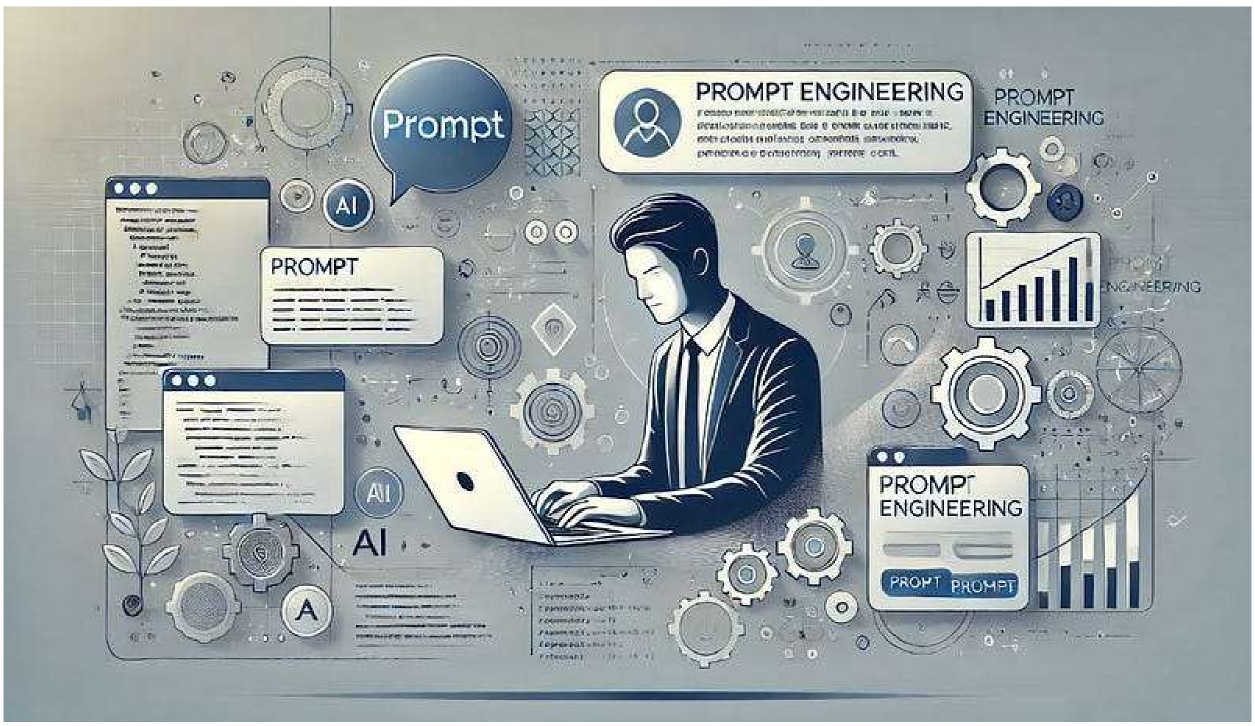
# Unlocking LLMs
# Advanced Techniques for AI Practitioners

## Frank P. Coyle, PhD

(preview chapter)

---




---

**Prompt Engineering 101**

If you've experimented with AI chatbots such as ChatGPT, you've probably experienced both the thrill of getting a spot-on response and the frustration of receiving utterly nonsensical or incorrect answers. You might have asked, "What's the capital of France?" and received a perfect answer, only to be baffled when "Tell me about the French Revolution" yielded a jumble of mixed-up dates and events. Welcome to the art and science of prompt engineering, where crafting the right input can make all the difference.

**What is a prompt?**

When you type a simple question into a chat window, you are creating a simple prompt. But simple prompts are only scratching the surface of what these models can do, and surface-level interaction often lead to those frustrating, incorrect results. It's important to realize that large language models (LLMs) like GPT-3 or GPT-4 are incredibly powerful and complex. They're not just glorified search engines or pre-programmed responders. Instead, think of them as vast neural networks that have digested and learned patterns from enormous amounts of text data.

Prompt engineering is about learning to communicate more effectively with AI models. It's like learning a new language, but instead of conjugating verbs, you're crafting inputs that guide the AI towards producing the kind of output you want. Learning how these models interpret your prompts and how to craft your inputs more effectively, can lead to dramatic improvement of the quality and accuracy of the responses you receive.

**How LLMs interpret your prompt**

LLMs interpret prompts through a process that's fundamentally different from how humans understand language. To grasp this, it's helpful to understand a bit about how these models work.

At their core, LLMs are pattern recognition machines. They've been trained on vast amounts of text data, learning to predict what words or tokens are likely to come next in a sequence. When you provide a prompt, the model doesn't "understand" it in the human sense. Instead, it analyzes the prompt as a pattern and tries to generate a continuation that matches similar patterns it has seen in its training data.

Let's walk through the process.

**Step1: Tokenization**

When you create your prompt, the model breaks it down into a series of tokens. Sometimes these are whole words, but most often they are parts of word—punctuation marks, or even individual characters. For example, "prompt engineering" might be broken into tokens like "pro", "mpt", " engineering".

All the tokens are collected in a block of memory called the *context window*. The amount of memory, i.e. the size of window, will be different depending on the LLM. For example, GPT-3.5 has a context window of 4,000 tokens, GPT-4 can handle 8,192 tokens and Claude 3 boasts a context window size of a whopping 200,000 tokens

Technically, these window sizes include your prompt *and* the model's ongoing response. What this means is that older parts of the conversation may be forgotten if they exceed this window. However, with context windows getting larger and larger, this should not be a problem.

**Step 2: Pattern Matching**

When the LLM receives the tokens in context window, the model examines the sequence of tokens and compares them to patterns it has seen in its training data. Based on these patterns, the model generates a probability distribution for what tokens are likely to come next. This distribution is influenced by many factors, including the frequency of token combinations in the training data and the patterns that appear in your prompt.

**Step3: Response Generation**

The model selects the next best word for its response based on the probabilities generated in Step 2. This process repeats, with each new generated token influencing the probabilities for the next word. This cycle continues until the model completes its response. Because text generation is based on a degree of randomness, it's the reason LLMs return slightly different responses even when given the exact same prompt.

**Prompt Construction**

There are several aspects of your prompt that influence the response. The first few words of your prompt are important in setting the tone and context for the entire response. For example, you might start with: "Respond in a professional tone:" or "Chat with me like a friend:"

*Emotional context* can also be helpful in shaping responses. Words like "exciting," "controversial," or "tragic" can influence emotional tone, as in: "Describe the thrilling discovery of:" vs "Explain the somber implications of:"

Generally, the most recent parts of a prompt tend to have more influence over the response than text in the middle of the prompt. By structuring your prompt carefully, you can significantly impact the output. Think about what is happening under the hood. You're essentially guiding the model's pattern-matching process towards the type of response you want. However, the effectiveness of these techniques can vary depending on the specific LLM and its training. Experimentation with different phrasings and keywords is one way to achieve the desired output.

**More context is better**

When you create a prompt, you have a fairly large context window at your disposal. Take advantage of it. If you ask a short question, you might get back a mishmash of facts. That's often because the AI lacks any context about what you may have in mind.

Instead of simply asking "Tell me about the French Revolution," zero in on what you want. For example: "Explain the key causes and major events of the French Revolution between 1789 and 1799, focusing on its impact on the French monarchy."

**Few-Shot Learning**

Few-shot learning is a technique used with LLMs to improve performance on specific tasks by providing a small number of examples within the prompt. This approach helps guide the model's output by demonstrating the desired format or type of response.

Here is an example of using few-shot learning for sentiment analysis:

Example 1:
Review: "The food was delicious and the service was excellent."
Sentiment: Positive
Example 2:
Review: "I waited for an hour and the meal was cold."
Sentiment: Negative
Example 3:
Review: "The restaurant was okay, nothing special."
Sentiment: Neutral
Now classify this review:
Review: "The ambiance was great, but the prices were too high."

Providing a few instances of the desired input-output pattern can help guide the LLM to produce more accurate and consistent responses for specific tasks. The model can learn from these examples and apply the pattern to new inputs.

**Role Playing**

Role-playing is another effective strategy to get the kind of output you are looking for. By assigning a specific role or expertise to the AI, you often get more focused and relevant response. To get a feel the effectiveness of role playing, let's look the prompt "Explain the Civil War" with different role-playing contexts. The responses in italics below are based on prompts to Claude 3.

Context 1: "You a third-grade teacher. Explain the Civil War in three sentences.

*The Civil War was a big fight between two parts of America, the North and the South, that happened a long time ago. The main reason for the fight was that the North wanted to stop slavery, but the South wanted to keep it. After a lot of battles and many people getting hurt, the North won, slavery ended, and America became one country again.*

Context 2: "You a professor of American History in graduate school. Explain the Civil War in three sentences.

*The American Civil War (1861–1865) was a pivotal conflict that arose from long-standing sectional tensions over slavery, states' rights, and economic disparities between the industrializing North and the agrarian South. The war's prosecution saw unprecedented military innovations and strategies, resulting in massive casualties and ultimately leading to the preservation of the Union and the abolition of slavery. However, the war's aftermath, particularly the complex and often fraught Reconstruction era, left many underlying social and racial issues unresolved, shaping American politics, culture, and race relations for generations to come*.

As you can see, specifying a role for the LLM to assume, can make a huge difference in the response..

**Tag-Based Prompting**

Structuring prompts using bracketed tags is an effective way to organize and clarify different components of your prompt. Tags are terms enclosed in start and end angle brackets (<....>). Note that the terminating angle bracket begins with slash as in "</tagname>.

Here are some suggestions for tags, making it much easier for the LLM to figure out exactly what you want it to do and how it should craft its reply.

1. **Role Tags:**
<role>Expert physicist</role>
<role>Creative writer</role>
<role>Business analyst</role>
2. **Style or Tone Tags:**
<style>Professional and formal</style>
<style>Casual and friendly</style>
<style>Academic</style>
3. **Format Tags:**
<format>Bullet points</format>
<format>Essay</format>
<format>Q&A</format>
4. **Constraints or Requirements:**
<constraints>Use simple language</constraints>
<constraints>Include at least three examples</constraints>
5. **Output Length:**
<length>300 words</length>
<length>2 paragraphs</length>
6. **Specific Instructions:**
<instructions>Start with a brief overview, then dive into details</instructions>
<instructions>Use metaphors to explain complex concepts</instructions>
7. **Audience:**
<audience>5th grade students</audience>
<audience>Technical professionals</audience>
8. **Desired Outcome:**
<outcome>Inspire curiosity</outcome>
<outcome>Persuade the reader</outcome>

These are just some of the tags that can be used with LLM's. Feel free to create your own tags that apply to your problem. Remember that the LLM understands natural language and will work to satisfy your request.

The following is an example of a prompt that combines different tags to create a comprehensive and clear prompt structure

```
<role>Science educator</role>
<audience>Middle school students</audience>
<task>Explain photosynthesis</task>
<style>Engaging and simple</style>
<format>Short paragraphs with bullet points</format>
<constraints>Use at least two analogies</constraints>
<length>300 words</length>
```

**Chain of Thought Prompting**

Chain of thought prompting is a technique used to improve the performance of large language models on complex reasoning tasks. It involves prompting the model to show its reasoning process step-by-step, rather than just providing a final answer.

For example, if you ask an LLM "what's the capital of Australia", it might correctly answer "Canberra" or might hallucinate and give an incorrect answer, such as "Melbourne". One way to obtain a correct answer with a higher probability is to ask: "What is the capital of Australia and explain how you arrived at the answer". When you ask the questions this way, expect a response along the following lines:

Explained answer: The capital of Australia is Canberra. Here's how I arrived at this answer:

I recall that many people mistakenly believe Sydney or Melbourne is the capital due to their size and prominence.

However, I remember that Australia's capital was purposely chosen as a compromise between these two competing cities.

Canberra was specifically built to serve as the capital, located between Sydney and Melbourne.

It officially became the capital in 1927.

I’m confident in this answer because Canberra’s status as the capital is a well-established fact that hasn’t changed.

By asking for a comprehensive analysis in one go, you might lead the AI through a step-by-step process, asking for intermediate outputs along the way and reducing the chance of a mistaken answer like Sydney or Melbourne. It also provides context that shows why the answer is reliable.

***Warning****: Using Chain of Thought prompting may incur a higher cost because LLM pricing models are based on the number of tokens in your prompt and the number of tokens generated by the LLM.*

**Summary**

Prompt engineering has emerged as a crucial skill in maximizing the potential of large language models (LLMs). By carefully structuring prompts, leveraging role-playing scenarios, and utilizing methods like one-shot and tag-based prompting, users can guide LLMs to produce more accurate, context-appropriate, and nuanced outputs.

The art of crafting effective prompts is not just about asking the right questions, but about creating a framework that allows AI models to leverage their vast knowledge in more focused and productive ways. Mastering these techniques can dramatically improve the efficiency and effectiveness of AI-assisted tasks across various domains, from content creation to problem-solving. Go forth and take control of your favorite LLM.