Prompt Engineering

- A prompt is simply the input or instruction you give to a Large Language Model (LLM) — like ChatGPT, GPT-5, or BERT — to tell it what you want it to do.

- Prompt Engineering in Large Language Models (LLMs) means the art and science of designing effective input prompts to get the desired, accurate, and relevant responses from the model.

- LLMs don't "know" your intent unless you clearly guide them. A small change in wording can dramatically change the response quality.

Weak Prompt

"Explain LSTM for Bitcoin prediction."

Output (typical):

LSTM stands for Long Short-Term Memory. It is a type of recurrent neural network used for sequential data. It can be used to predict Bitcoin prices by learning patterns in the data.

Strong (Engineered) Prompt

"You are a data scientist explaining Bitcoin price forecasting to graduate students. Describe step-by-step how an LSTM network uses historical OHLC (Open, High, Low, Close) data to predict the next-day closing price. Include how data is preprocessed, how sequences are created, and how the model learns temporal dependencies."

Output (typical):

In Bitcoin price forecasting, an LSTM (Long Short-Term Memory) model is used to capture long-term temporal patterns in sequential data.

1. **Data preprocessing:** Historical OHLC data is normalized to ensure stable training. Missing values are imputed and outliers handled.
2. **Sequence creation:** The model takes past *n* time steps (e.g., last 60 days) as input to predict the next-day closing price.
3. **Model structure:** LSTM layers maintain memory cells that capture dependencies over time, followed by dense layers for final prediction.
4. **Training:** The model minimizes mean squared error between predicted and actual prices.
5. **Output:** The trained model forecasts future Bitcoin prices with improved temporal accuracy compared to basic RNNs.

Prompt Type	Example	Purpose
Zero-shot	**"Summarize this text."**	No prior examples
Few-shot	**Give example Q&A before asking**	Show desired format
Role-based	**"You are a finance expert..."**	Control tone/role
Chain-of-thought	**"Explain your reasoning..."**	Improve logic & reasoning
Instructional	**"List 5 key points..."**	Precise control over structure

The Basic Ingredients of a Prompt

An LLM is a prediction machine. Based on a certain input, the prompt, it tries to predict the words that might follow it. At its core (illustrated in Figure 6-6), the prompt does not need to be more than a few words to elicit a response from the LLM.

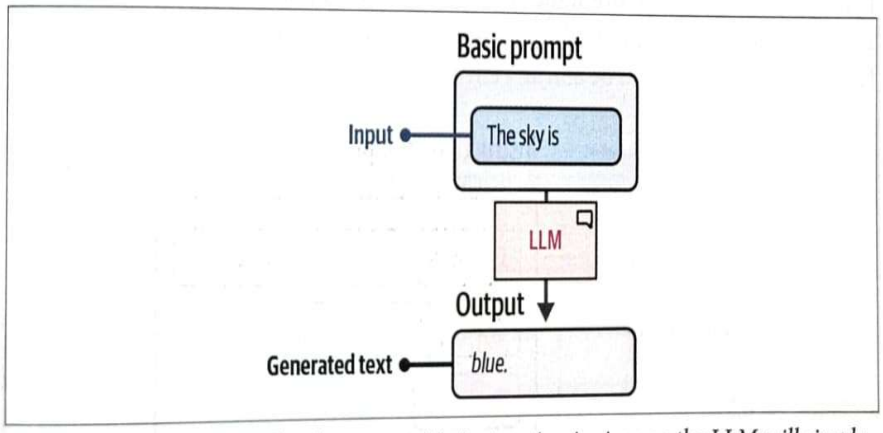

Figure 6-6. A basic example of a prompt. No instruction is given so the LLM will simply try to complete the sentence.

However, although the illustration works as a basic example, it fails to complete a specific task. Instead, we generally approach prompt engineering by asking a specific question or task the LLM should complete. To elicit the desired response, we need a more structured prompt.

For example, and as shown in Figure 6-7, we could ask the LLM to classify a sentence into either having positive or negative sentiment. This extends the most basic prompt

to one consisting of two components—the instruction itself and the data that relates to the instruction.

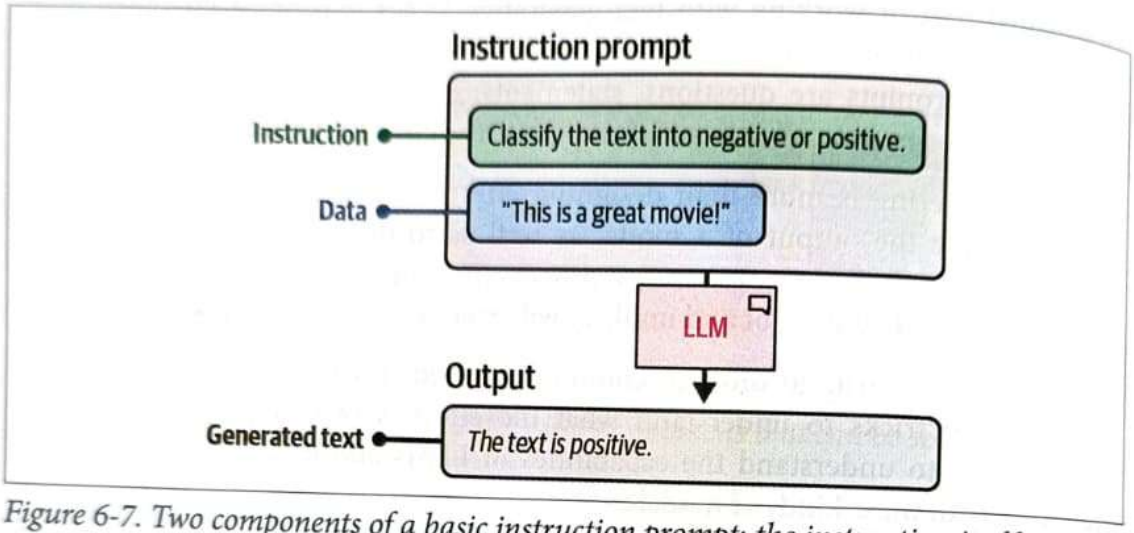

Figure 6-7. Two components of a basic instruction prompt: the instruction itself and the data it refers to.

More complex use cases might require more components in a prompt. For instance, to make sure the model only outputs "negative" or "positive" we can introduce output indicators that help guide the model. In Figure 6-8, we prefix the sentence with "Text:" and add "Sentiment:" to prevent the model from generating a complete sentence. Instead, this structure indicates that we expect either "negative" or "positive." Although the model might not have been trained on these components directly, it was fed enough instructions to be able to generalize to this structure.

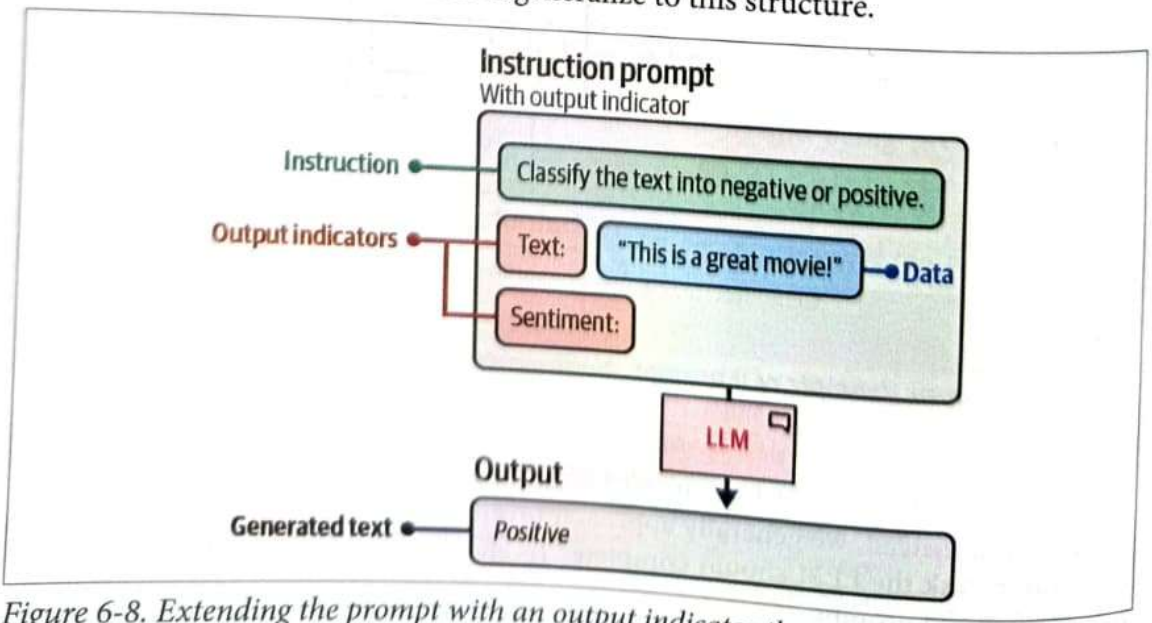

Figure 6-8. Extending the prompt with an output indicator that allows for a specific output.

Instruction based Prompting

Instruction-based prompting is a method where the prompt contains direct and specific instructions guiding the model's behaviour, style, content, or format of the output.

Instruction-Based Prompting

Although prompting comes in many flavors, from discussing philosophy with the LLM to role-playing with your favorite superhero, prompting is often used to have the LLM answer a specific question or resolve a certain task. This is referred to as *instruction-based prompting*.

Figure 6-9 illustrates a number of use cases in which instruction-based prompting plays an important role. We already did one of these in the previous example, namely supervised classification.

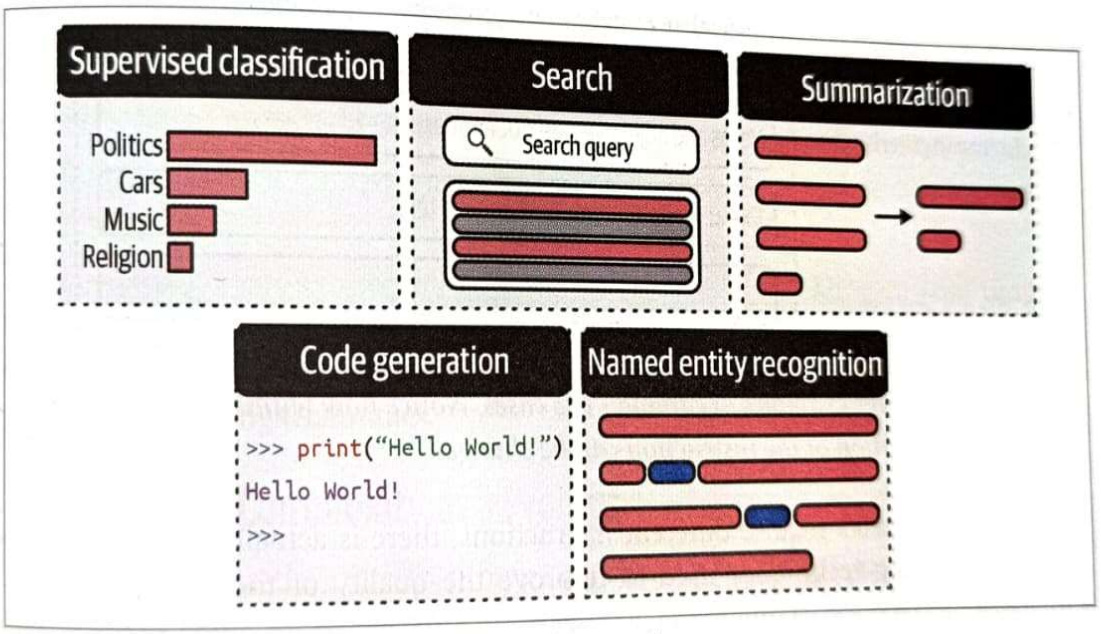

Figure 6-9. Use cases for instruction-based prompting.

Each of these tasks requires different prompting formats and more specifically, asking different questions of the LLM. Asking the LLM to summarize a piece of text will not suddenly result in classification. To illustrate, examples of prompts for some of these use cases can be found in Figure 6-10.

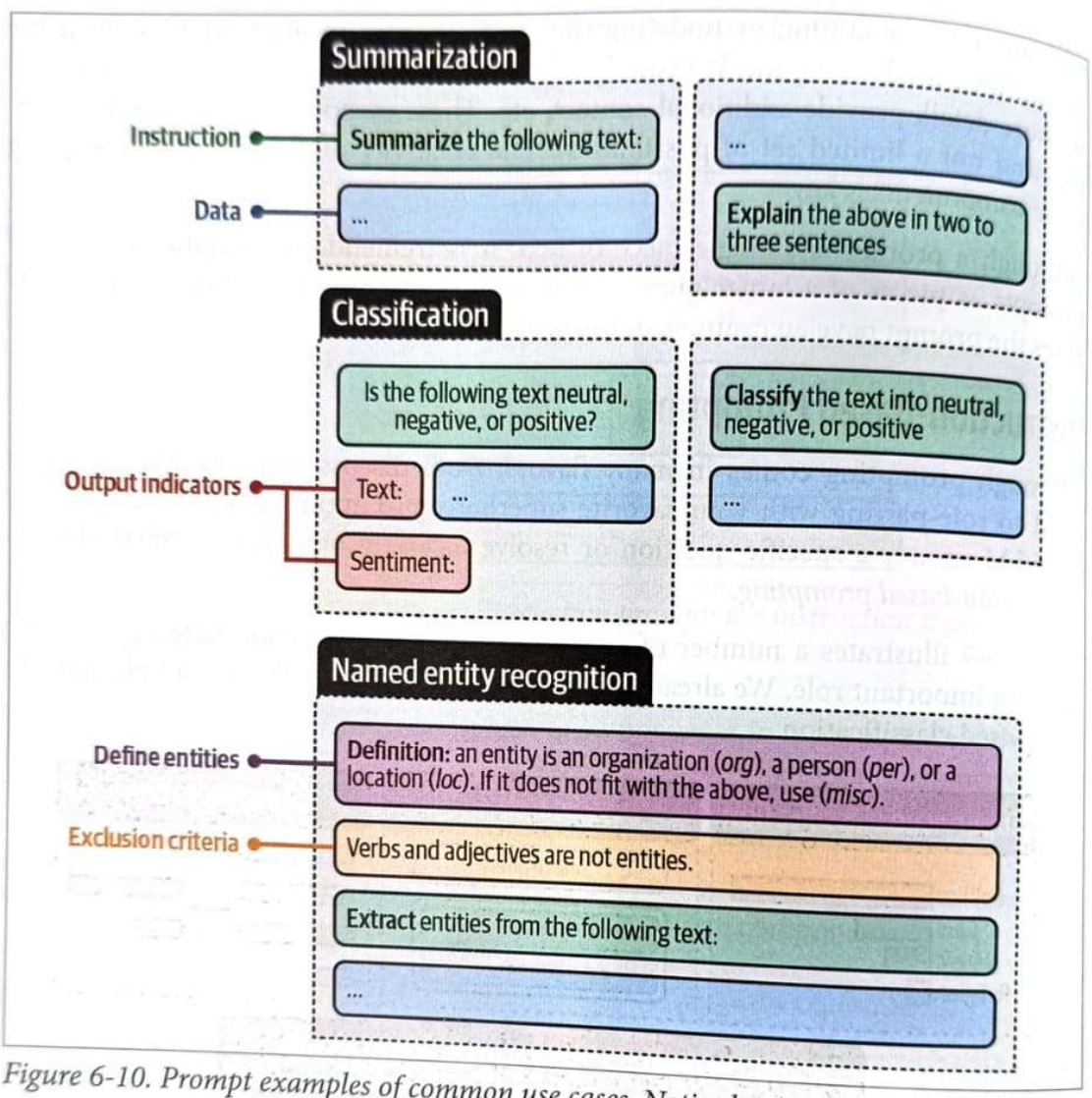

Figure 6-10. Prompt examples of common use cases. Notice how within a use case, the structure and location of the instruction can be changed.

Although these tasks require different instructions, there is actually a lot of overlap in the prompting techniques used to improve the quality of the output. A non-exhaustive list of these techniques includes:

Specificity

Accurately describe what you want to achieve. Instead of asking the LLM to "Write a description for a product" ask it to "Write a description for a product in less than two sentences and use a formal tone."

Hallucination

LLMs may generate incorrect information confidently, which is referred to as hallucination. To reduce its impact, we can ask the LLM to only generate an answer if it knows the answer. If it does not know the answer, it can respond with "I don't know."

Order

Either begin or end your prompt with the instruction. Especially with long prompts, information in the middle is often forgotten.[1] LLMs tend to focus on information either at the beginning of a prompt (primacy effect) or the end of a prompt (recency effect).

Advance Prompt Engineering

Advanced Prompt Engineering is the practice of using expert-level prompting methods to guide an LLM to think, reason, analyse, and generate better results for complex tasks.

What Advanced Prompt Engineering Achieves

✓ Higher accuracy in reasoning

✓ More predictable and consistent output

✓ Better performance on coding, math, NLP tasks

✓ Reduction of hallucinations

✓ Ability to handle multi-step complex workflows

✓ Better control over tone, format, and reasoning style

Where Advanced Prompt Engineering Is Used

- LLM-based apps
- Chatbots
- Research automation
- Code generation
- Data extraction
- Finance & forecasting models
- Academic writing
- Teaching and learning