

Prompt engineering best practices

Large language models (LLMs) react to what we ask them — the better the input, the more useful their output. Use this guide to create effective prompts that help LLMs perform their best, so you can get the most valuable response.



Specify the task

LLMs are trained on massive amounts of data. You need to get specific about your desired result, so the model can deliver a focused output. Be clear about what you want LLMs to do by providing sufficient parameters. Use straightforward language, and structure queries in a logical way to enhance how the model interprets your request. No specific set up is best; write intuitively and focus on clarity.

Example: *Draft an informal email to my manager requesting time off.*

Provide necessary context

Context shapes how LLMs respond to a prompt by providing important information about your expectations. With relevant context, it is more likely LLMs will generate useful output.

Include key details about your request to give LLMs the information they need to generate useful output. Here are some questions to consider when writing an effective prompt:

Who's the target audience? Specify relevant qualities of the audience, such as their age, profession, or level of understanding on a topic.

What tone should the model use? Clarify the voice and style in which LLMs should use to most effectively convey their message. You might want an output that's casual and friendly if you're using it to communicate with a peer, or something more professional and persuasive for clients.

How should LLMs structure output? Specify the format LLMs should use to order the information it provides. You may include guidance about length or specify a type of layout, such as a bulleted list or table.

What's the output's goal? Identify what you want LLMs to accomplish with a given prompt. For example, if your prompt asks the model to explain a concept, the goal might be to ensure beginners in that specific field gain a working understanding of the topic. Giving an LLM a goal will help shape the outputs to your specific needs.

Example: *Write a friendly email to my HR coworker thanking them for their collaboration on a recent project so they know their contributions were invaluable.*

Provide references

Providing LLMs with reference materials that achieve your goals or resemble what you want to create can help generate more useful outputs. Whether you're including your own work, broader sources, or both, you also want to explain clearly how these reference materials relate to your prompt for the best possible results.

Example: *Draft a list of potential campaign slogans for a sunglass company in the writing style of 1960s billboard advertisements.*

Evaluate your output

Each model has a unique training set, relies on different programming techniques, and is developed at a specific point in time. As a result, some LLMs may know more about certain topics than others or may experience a knowledge cutoff. Models can occasionally hallucinate, too.

Before you use an AI-generated output, critically assess the output to ensure it's acceptable and beneficial to you. This may involve conducting some research after LLMs produce their output. When evaluating an output, ask yourself:

Is this response accurate? Confirm the information is up-to-date and factual.

Is this response unbiased? Evaluate whether the response is fair and impartial, accurately represents populations, and avoids preferential treatment for certain individuals or groups.

Does this response include sufficient information? Ensure the response delivers a comprehensive and satisfactory response to your query.

Is this response relevant to what I need? Check that the output relates to your prompt and aligns with the context, topic, and task you outlined.

Is this response consistent? Verify that your response isn't an outlier. If you aren't sure, try prompting LLMs multiple times in different ways to ensure the outputs give you similar information.

If you determine an output is unacceptable, try to add more context to the initial prompt to generate a more focused response:

Example: The output from a prompt like *What's a conditional?* might be broad, varied, or irrelevant to your needs, since that term has different meanings in various contexts.

Iteration: Instead, a prompt like *Explain 'conditionals' to a beginner coder like a textbook* would most likely produce a more targeted, useful output by specifying the audience, tone, and discipline.

Take an iterative approach

Whatever the reason, a LLM might not produce what you need the first time you ask for it. You can still arrive at your desired outcome with some iteration, by refining the initial prompt, issuing follow-up requests, or giving LLMs feedback.

To successfully revise a prompt, keep what worked and adjust the input from there. You might change some phrasing (like whether the prompt is a command or question), reorder the prompt's components (like whether you start or end with an example), or provide additional context to help narrow LLMs' responses.

Example: *Summarize the following meeting notes.*

Iteration: *Summarize the following meeting notes and identify key takeaways.*

Further iteration: *Summarize the following meeting notes, identify key takeaways, and list the most urgent action items with their deadlines.*

To efficiently issue follow-up requests, ask the model to make adjustments without repeating the initial prompt, like a back-and-forth dialogue. LLMs are able to build off of prior interactions within a conversation, which means you can focus on making targeted, individual adjustments until you have everything you need.

Example: *Summarize the following meeting notes.*

Follow-up: *What were key takeaways from this meeting?*

Second follow-up: *What are the most urgent action items and their deadlines?*