

MODULE 3: Large Language Models (LLMs): Definition, Capabilities, and Prompt Engineering

Haitam EL-KHAMALI

22 March, 2025

1 Definition

A Large Language Model (LLM) is an AI system trained on vast amounts of text to identify patterns between words, phrases, and concepts. It generates responses by predicting the most probable next word in a sequence based on statistical analysis.

2 How LLMs Work

LLMs are trained on massive datasets from books, articles, and websites. The quality and diversity of this training data directly impact their performance. These models analyze the relationships between words and compute probabilities for thousands of possible next words in a sequence. For example, given the incomplete sentence :

- “After it rained, the street was...”
 - The word “wet” has a high probability.
 - The word “clean” has a lower probability.
 - The word “dry” has an extremely low probability.

An LLM would most likely select the word with the highest probability, such as “wet” or “damp”, to complete the sentence in a natural way.

3 Capabilities of LLMs

LLMs can generate responses for a wide range of tasks, including :

- Completing sentences and paragraphs.
- Writing detailed stories, reports, and marketing content.
- Summarizing long documents or answering questions.

This predictive ability makes them useful in various fields, from research and automation to creative writing and business communication.

4 Limitations of LLMs

Despite their capabilities, LLMs have several limitations :

1. Bias in Training Data

- Since LLMs learn from human-generated content, they can inherit societal biases present in that data.
- Example : An LLM might associate professional roles with specific genders due to biased training sources.

2. Limited Knowledge on Specific Topics

- If certain topics are underrepresented in the training data, LLMs might generate incomplete or inaccurate responses.
- Example : A niche scientific concept may not be well covered, leading to vague or incorrect explanations.

3. Hallucinations (False Information)

- LLMs sometimes generate factually incorrect content, even when responding confidently.
- Example : An LLM summarizing a company's history might provide a wrong founding date or incorrect number of employees.
- Causes of hallucinations :
 - ▷ Gaps in training data.
 - ▷ The probabilistic nature of predictions.
 - ▷ The phrasing of the prompt.

5 Best Practices for Using LLMs

To use LLMs effectively, it is crucial to critically evaluate their output by checking for :

- Accuracy – Ensuring factual correctness.
- Bias – Identifying any unfair or skewed responses.
- Relevance – Making sure the response aligns with the request.
- Completeness – Verifying if all necessary details are included.

Additionally, users should avoid making assumptions about an LLM's reliability. Just because it provides high-quality output once does not mean it will consistently do so with the same prompt in the future.

6 How to Write Effective Prompts for Large Language Models (LLMs)

1. Introduction

The quality of the input directly affects the quality of the output when using AI models. Just like cooking with fresh ingredients leads to better meals, providing a well-structured prompt results in more useful AI responses. This process is known as prompt engineering—the practice of designing clear and specific prompts to get the desired results from an LLM.

2. Key Principles of Prompt Engineering

1. Specify the Task Clearly

A well-defined task is the foundation of every good prompt. If the instructions are vague, the AI may produce irrelevant or incorrect responses. Elements of a clear task :

- Action : What do you want the AI to do ? (e.g., write an email, generate an image)
- Persona : What expertise should the AI adopt ? (e.g., IT professional, marketer)
- Format : How should the output be structured ? (e.g., bulleted list, comparison table)

Example Prompt :

“You’re a concert promoter specializing in alternative rock. Create a 125-character social media post to attract local and out-of-state festival-goers. Include 5 relevant hashtags.”

2. Provide Necessary Context

AI models lack background knowledge about the user’s intent unless explicitly stated. Adding relevant details helps narrow the AI’s focus and improves accuracy.

Example Prompt with Context :

“You’re a concert promoter targeting young professionals (ages 21-35) who follow indie rock. Create a 125-character social media post for a two-day music festival with 12 bands, camping options, and local food vendors. Include 5 hashtags.”

3. Use References to Guide Output

Providing examples can help steer the AI toward a particular style, tone, or format. References clarify expectations and improve consistency.

Example Prompt with References :

“You’re a concert promoter for an indie music festival. Create a 125-character social media post using a casual and engaging tone. Here are two successful examples from past festivals :”

- Example 1 : “Where mountains meet music : Indie Rocks Festival returns ! Your favorite local bands + national acts. Good eats & camping ! #IndieRocks #SupportLocalMusic”
- Example 2 : “Join Indie Fest under the desert skies ! 2 nights of raw sound to move you + camping vibes. #RoadTrip #CampLife #RockOn”

3. Evaluating and Iterating on AI Output

1. Evaluate AI Responses Carefully

Since AI tools vary in their training and capabilities, responses must be reviewed before use. Key factors to check include :

- Accuracy – Is the information correct ?
- Relevance – Does it match the request ?
- Bias – Are there any unintended biases in the response ?
- Consistency – Does the style align with expectations ?

2. Refine the Prompt Through Iteration

Effective prompting is often an iterative process. If the initial prompt does not yield the desired result :

- Analyze the response to identify missing details or misunderstandings.
- Revise the prompt to be clearer or add more context.
- Re-test and refine until the AI produces the best possible output.

Example of Prompt Refinement :

- First Attempt : “Generate five themes for a conference.” (Vague and unclear)
- Revised Attempt : “Generate five professional conference themes related to customer experience in the hospitality industry.” (More specific and relevant)

7 Prompts for Different Purposes

Recall that a large language model, or LLM, is an AI model that is trained on large amounts of text to identify patterns between words, concepts, and phrases so that it can generate responses to prompts. As you’ve been learning, good prompting can help guide an LLM to generate useful output for workplace tasks. In this section, you’ll further explore how to write clear and specific prompts for a variety of workplace use cases.

7.1 Use Cases

You might use an LLM at work to help boost your productivity and creativity and complete any of these useful tasks :

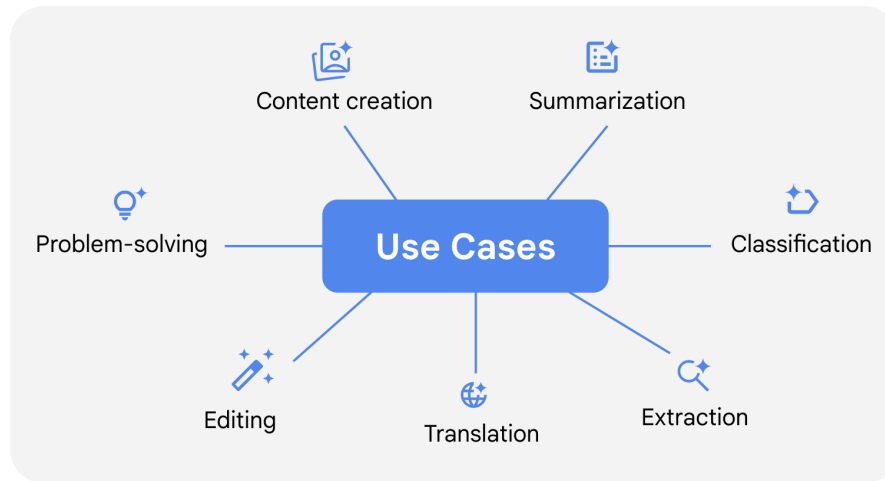
- Content creation
- Summarization
- Classification
- Extraction
- Translation
- Editing
- Problem-solving

Note : The following examples illustrate prompting best practices ; they aren’t exact templates to copy for every situation. Your results will vary based on a number of factors, including the specific LLM you’re using. Remember to critically evaluate all LLM output and to iterate on your initial prompt to get the most useful output.

7.2 General Guidelines for Effective Prompts

- Consider what you want the LLM to produce. The LLM will generate more useful output when you include a specific instruction in your prompt, like create, summarize, classify, extract, translate, edit, or solve.
- Provide necessary context. The LLM will generate more useful output when you include detailed instructions, with specific guidance about the style or format of the output you want.

7.3 Examples of Use Cases



7.3.1 Content Creation

No matter your industry, an LLM can help you create content for a variety of purposes, such as blog posts, reports, product descriptions, and taglines. For example :

Prompt :

“Act like you are a creative advertising professional who can apply innovative thinking to develop original taglines that project the positive qualities of a product. Create a concise tagline for a washing machine that gets clothes extra clean, has 25 settings, and fits in a small space.”

Pro tip : Assign the LLM a role, job, or function to reinforce the purpose of the prompt and help guide the LLM to produce useful output.

7.3.2 Summarization

An LLM can help you summarize many types of texts : reports, customer surveys, meeting notes, emails, and more. For example :

Prompt :

“The following text is an email from a software vendor. Summarize its main points in a bulleted list :

[Email text here]”

Note : Be aware that LLMs can sometimes hallucinate, or produce AI outputs that aren’t true. Always evaluate LLM output for accuracy before using it.

7.3.3 Classification

Text classification is another common workplace application for LLMs. For example :

Prompt :

“Read these customer reviews and tell me whether the sentiment of the reviews is positive, negative, or neutral.

- Customer Review : [Review 1]
- Customer Review : [Review 2]”

7.3.4 Extraction

You can also use an LLM to pull data from text and transform it into a structured format. For example :

Prompt :

“Read the blog post below and extract all of the references to items of clothing I can buy and how much each item costs. Create a bulleted list of just these items.

- Blog post : [Blog text here]”

7.3.5 Translation

You can leverage an LLM to translate text between different languages very quickly. For example :

Prompt :

“Translate our product descriptions from English to Spanish. Maintain the same structure and casual tone that is used in the English version in the Spanish translation.

- Bicycle : [Description]

- Rollerblades : [Description]”

Note : As a best practice, confirm that an LLM’s translations are accurate by cross-checking with another translation tool.

7.3.6 Editing

You can also use an LLM to edit and rewrite text. For example :

Prompt :

“Edit the language of the following paragraph so that it’s easy for a general audience to understand it. Use simpler vocabulary and grammatical structures but maintain the same ideas.

7.3.7 Problem-solving

One more use case is problem-solving. For example :

Prompt :

“We are running a community program to teach children gardening skills. The program runs from June 1 to August 15. We want the children to be able to grow plants that will be ready for harvest by the time the program ends. First, identify a list of 10 plants that can be planted and grown in that time period. Include sources that support the time to harvest for each plant.

We want the children to grow three plants. These plants should be as different from each other as possible. So next, choose three plants from the list that will provide the children with this variety.”

8 Few-Shot Prompting and the Use of Examples

Including examples in prompts helps LLMs generate more accurate and relevant responses. The term “shot” refers to the number of examples provided :

- Zero-shot prompting : No examples are given ; the model relies only on its training data and the task description.
- One-shot prompting : A single example is provided to guide the model.

- Few-shot prompting : Two or more examples are given to establish a pattern, improving response quality.

Few-shot prompting is especially useful for tasks requiring specific styles or formats. Providing well-structured examples allows the AI to generate outputs that align with desired characteristics. However, too many examples may reduce flexibility and creativity. Experimentation is key to finding the right balance for optimal results.

9 Chain-of-Thought Prompting & Prompt Chaining

Chain-of-thought prompting is a technique that improves AI problem-solving by requesting step-by-step reasoning. Using prompts like “Explain your reasoning” or “Go step by step”, AI can generate more structured, informative, and accurate responses. This approach helps validate outputs and is useful for complex tasks, such as developing onboarding materials.

Prompt chaining involves linking multiple prompts together, using the output of one as input for the next. This iterative process helps break down complex tasks logically and ensures more refined results.

Combining both techniques enhances structured problem-solving. However, challenges arise when chains become too long, leading to inconsistent responses or loss of context. Strategies to mitigate this include using checkpoints, working on sub-tasks, and periodically summarizing progress.