UTM
UNIVERSITI TEKNOLOGI MALAYSIA

CAIRO
UTM

# Guide to Prompting in Large Language Models (LLM)

Copyright for Center for Artificial Intelligence and Robotics, UTM

# What is a Prompt?

A prompt is any input or instruction you give a large language model (LLM) to specify what you want.

- questions,
- commands,
- role instructions, or
- text to continue.

**Example:**

**Prompt:** “Translate the sentence ‘Good morning’ into Spanish.”
**Output:** “Buenos días ”.

# Prompt vs. Engineered Prompt

**Prompt**

“Write a blog post about microservices.”

**Engineered Prompt**

“*You are a senior software architect with 15 years of experience in building distributed systems.* Write a **technical blog post** about **microservices architecture patterns**, including real-world examples and best practices for scalability and maintenance.”

**Role**

**Task**

**Focus**

**Specific additions**

# Types of Prompts

# Zero-Shot Prompting

- *Zero-shot prompting* means you ask the model to perform a task **without** showing examples.

- You simply provide an instruction or question and expect the model to respond based on its learned knowledge.

**Prompt:** "Summarize the following article in one paragraph”

**Output:** A one-paragraph summary of the article.

# One-Shot Prompting

*One-shot prompting* provides the model with **one** example of the task before asking it to perform the task on a new input

Example Q&A:
Q: "What's the capital of France?"
A: "Paris."

We gave one example question (*France* → *Paris*)

Now answer this question:
Q: "What's the capital of Japan?"
A: ***Tokyo***

It expected to follow the pattern, answering *"Tokyo."*

# Few-Shot Prompting

- *Few-shot prompting* provides a few examples (**more than one**) of the desired task in the prompt before the actual question or command.

- The examples act as demonstrations for the model, allowing it to **learn the pattern** or format from the prompt itself.

- Few-shot prompts are helpful for more complex tasks: Classification.

**Task:** Classify the sentiment of each sentence as Positive or Negative.

**Example 1:** "I love the new design of your website!" → **Positive**
**Example 2:** "The product stopped working after a week." → **Negative**

mean in context

Now classify this sentence:
"The service was very good" → ***Positive***

Copyright for Center for Artificial Intelligence and Robotics, UTM

# Instruction-Based Prompting

An *instruction-based prompt* is phrased as a direct instruction or command to the model, often using imperative verbs (like "write", "explain", "calculate") - ***what to do***

"**Write a short introduction** to quantum computing aimed at beginners."

# Dialogue-Style Prompting

- *Dialogue-style prompting* means interacting with the model in a conversational format.

- This is used in *chatbots and conversational agents*.

**User:** How do airplanes fly?
**Assistant:** Airplanes fly through a principle called lift. The wings are shaped to.......
**User:** Why are the wings shaped that way?
**Assistant:** …………….

# Structured Prompting

A *structured prompt* is a prompt that is formatted in a clear, organized way, often with multiple parts, sections, or formatting cues. They improve reliability and consistency.

**Context:** You are an AI tutor helping a student with math. The student is struggling with understanding prime numbers.

**Task:** Explain what prime numbers are and give 3 examples of prime numbers between 1 and 20.

**Format:** (structured output with specific sections)
1. Definition of prime numbers in simple terms (1-2 sentences).
2. A short explanation of why prime numbers matter.
3. Three examples of prime numbers between 1 and 20, listed in bullet points.

# Prompting Strategies

# Role Prompting

Assign a **role** (e.g., “You are a medical doctor…”) to shape tone, depth, and perspective.

Roles cue the model to draw on appropriate style and domain patterns seen during training.

# Chain-of-Thought (CoT) Prompting

Encourage step-by-step reasoning **(e.g., “Let’s think step by step”).**

Improves performance on multi-step problems.

# Self-Consistency (with CoT)

Generate **multiple** reasoning paths and **choose** the most consistent final answer.

Answer 1

Answer 2

# Retrieval-Augmented Generation (RAG) Prompting

**Retrieve** relevant context (from a database, PDF, or the web) and **insert it into the prompt** before generation.

**[Retrieved text (PDF): Memory] → Finds relevant chunks per query**
“Neptune is the eighth planet from the Sun. It was discovered in 1846 and is known for its striking blue color and strong winds.”
**Question:** “Which planet is the eighth from the Sun, and when was it discovered?”
**Explanation.** The model answers from the provided snippet (“Neptune”, “1846”), not just its memory. → **Generation**

# Building a Chatbot

| Stage | Component | Description |
|---|---|---|
| **Upload** | FastAPI + PyPDFLoader | Read and preprocess the PDF |
| **Split** | RecursiveCharacterTextSplitter | Breaks long text into chunks |
| **Embed** | embeddinggemma:300m (Ollama) | Converts text to vectors |
| **Store** | Chroma | Vector DB for semantic search |
| **Retrieve** | Chroma similarity search | Finds relevant chunks per query |
| **Generate** | DeepSeek-R1 (Ollama) | Writes the natural-language answer |
| **Clean** | Regex filters | Remove <think> / reasoning |
| **Display** | HTML + JS UI | Show answers and sources |

Copyright for Center for Artificial Intelligence and Robotics, UTM

## Ingest.py

CHUNK_SIZE = Maximum size of each sentence
CHUNK_OVERLAP = How much of the previous sentence to repeat in the next one

## Chat.py

TOP_K = 4, the model only looks at the 4 most likely next words and ignores the rest.
TEMPERATURE = Controls how random or confident the model’s word selection is.

- Low temperature (close to 0) → very focused, deterministic (almost always picks the top word).
- High temperature (like 1.0 or higher) → more random and creative output.

# THANK YOU

f Centre for Artificial Intelligence and Robotics

research.utm.my/cairo/

✉

Copyright for Center for Artificial Intelligence and Robotics, UTM