



Natural Language Processing and Large Language Models

Corso di Laurea Magistrale in Ingegneria Informatica



Lesson 18

Prompt Engineering

Nicola Capuano and Antonio Greco

DIEM – University of Salerno

.DIEM

Outline

- Introduction to Prompt Engineering
- Prompt Engineering Techniques
- Prompt Testing





Introduction to Prompt Engineering

Prompt Engineering

A relatively new discipline focused on **developing and optimizing prompts to effectively use LLMs** for diverse applications and research areas

Goals:

- Enhances understanding of the **capabilities** and **limitations** of LLMs
- Improves LLM **performance** on a broad range of tasks (e.g., question answering, arithmetic reasoning, ...)
- Helps **interfacing** with LLMs and **integrating** with other tools
- Enables new capabilities, such as augmenting LLMs with **domain knowledge** and **external resources**

Writing Good Prompts

- Start with **simple prompts**, adding elements gradually while iterating and refining to improve results
- Use **clear, specific instructions** (e.g., “Write,” “Classify,” “Summarize”) at the **beginning** of prompts
- Be **detailed** and **descriptive** to achieve better outcomes
- Consider using **examples** to guide the model’s output
- Balance detail and length carefully, as **excessive information can reduce effectiveness**, and experiment to find the ideal format

Writing Good Prompts

- **Bad Prompt:** “Summarize this article.”
- **Good Prompt:** “Generate a 100-word summary of this research article, focusing on the main findings.”
- **Bad Prompt:** “Write an apology email to a client.”
- **Good Prompt:** “Write a professional email to a client apologizing for a delayed shipment, offering a discount, and providing an updated delivery estimate.”
- **Bad Prompt:** “Make this explanation easier to understand.”
- **Good Prompt:** “Rewrite this technical explanation in simpler language suitable for high school students.”

Writing Good Prompts

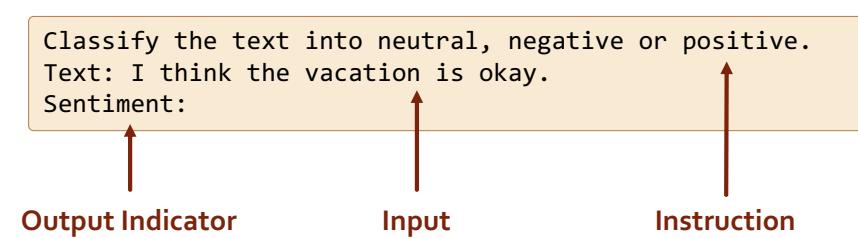
- **Bad Prompt:** “Classify the following review.”
- **Good Prompt:** “Classify the following review as positive, neutral, or negative”
- **Bad Prompt:** “Tell me about exercise benefits.”
- **Good Prompt:** “List five health benefits of regular exercise, each with a short explanation of how it improves well-being.”
- **Bad Prompt:** “Translate this sentence to French.”
- **Good Prompt:** “Translate the following English sentence into French, preserving the formal tone.”

Elements of a Prompt

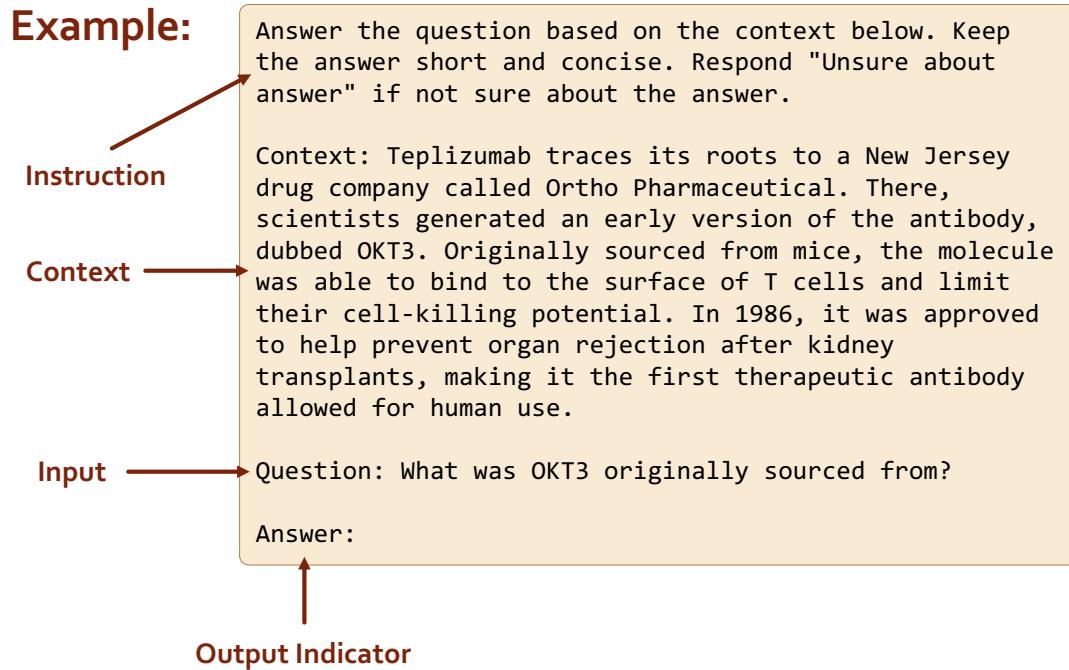
A prompt usually contains any of the following elements:

- **Instruction** – a specific task or instruction you want the model to perform
- **Context** – external information or additional context that can steer the model to better responses
- **Input Data** – the input or question that we are interested to find a response for
- **Output Indicator** – the type or format of the output

Example:



Elements of a Prompt



In-Context Learning

The ability of a LLM to perform a task by **interpreting and leveraging information provided in its prompt (context)** without updating its internal parameters

A prompt context may specify...

- **Reference Material:** specific text or data to be used to perform the task
- **Input-Output Pairs:** examples of the task to illustrate the desired pattern
- **Step-by-Step Instructions:** detailed guidance for completing the task
- **Clarifications:** addressing potential ambiguities in the task
- **Templates:** structures or placeholders to be filled in

Prompt engineering heavily leverages in-context learning

Prompts and NLP Tasks

Prompts can be designed to achieve various NLP tasks...

Text Summarization:

Prompt:

Antibiotics are a type of medication used to treat bacterial infections. They work by either killing the bacteria or preventing them from reproducing, allowing the body's immune system to fight off the infection. Antibiotics are usually taken orally in the form of pills, capsules, or liquid solutions, or sometimes administered intravenously. They are not effective against viral infections, and using them inappropriately can lead to antibiotic resistance.

Explain the above in one sentence:

Output:

Antibiotics are medications used to treat bacterial infections by either killing the bacteria or stopping them from reproducing, but they are not effective against viruses and overuse can lead to antibiotic resistance.

Prompts and NLP Tasks

Prompts can be designed to achieve various NLP tasks...

Information Extraction:

Prompt:

Author-contribution statements and acknowledgements in research papers should state clearly and specifically whether, and to what extent, the authors used AI technologies such as ChatGPT in the preparation of their manuscript and analysis. They should also indicate which LLMs were used. This will alert editors and reviewers to scrutinize manuscripts more carefully for potential biases, inaccuracies and improper source crediting. Likewise, scientific journals should be transparent about their use of LLMs, for example when selecting submitted manuscripts.

Mention the large language model based product mentioned in the paragraph above:

Output:

The large language model based product mentioned in the paragraph above is ChatGPT.

Prompts and NLP Tasks

Prompts can be designed to achieve various NLP tasks...

Question Answering:

Prompt:

Answer the question based on the context below. Keep the answer short and concise. Respond "Unsure about answer" if not sure about the answer.

Context: Teplizumab traces its roots to a New Jersey drug company called Ortho Pharmaceutical. There, scientists generated an early version of the antibody, dubbed OKT3. Originally sourced from mice, the molecule was able to bind to the surface of T cells and limit their cell-killing potential. In 1986, it was approved to help prevent organ rejection after kidney transplants, making it the first therapeutic antibody allowed for human use.

Question: What was OKT3 originally sourced from?

Answer:

Output:

Mice.

Prompts and NLP Tasks

Prompts can be designed to achieve various NLP tasks...

Text Classification:

Prompt:

Classify the text into neutral, negative or positive.

Text: I think the food was okay.

Sentiment:

Output:

Neutral

Prompts and NLP Tasks

Prompts can be designed to achieve various NLP tasks...

Code Generation:

Prompt:

```
Write a JavaScript code that prompts the user to enter their name and then displays
a "Hello" message.
Output only the code, no additional comments.
```

Output:

```
let name = prompt("What is your name?");
console.log(`Hello, ${name}!`);
```

Prompts and NLP Tasks

Prompts can be designed to achieve various NLP tasks...

Reasoning:

Prompt:

```
What is 9,000 * 9,000?
```

Output:

```
81,000,000
```

Note: despite advancements in mathematical capabilities,
reasoning tasks remain challenging for LLMs...

Prompts and NLP Tasks

Prompts can be designed to achieve various NLP tasks...

Reasoning:

Prompt:

The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.
A:

Output

No, the odd numbers in this group add up to an odd number: 119.

That's incorrect! More advanced prompt engineering techniques are needed to obtain the **correct answer**... we will discuss them later.

System Prompts

A **configuration** provided to the AI model **before any user interactions**

- Establishes the assistant's **behavior, context, tone**, and any **special instructions**
- Guides the model on **how to respond** and **what it should focus on**

Examples:

- *You are a helpful and knowledgeable assistant who answers questions accurately and concisely.*
- *You are an IT support assistant specializing in troubleshooting software and hardware issues. Respond politely and guide users through step-by-step solutions.*
- *You are a friendly and engaging AI who responds in a warm and conversational tone, keeping responses lighthearted and approachable.*



Prompt Engineering Techniques

Zero-Shot Prompting

A prompt that interacts with the model **without including examples or demonstrations**

- A zero-shot prompt gives the model direct instructions to perform a task **without any additional examples** for guidance
- Large-scale training enables LLMs to **handle many tasks in a “zero-shot” manner**

Example:

Classify the text into neutral, negative or positive.
Text: I think the vacation is okay.
Sentiment:

Output:

Neutral

the LLM already understands the concept of “sentiment”
(that's the zero-shot capabilities at work)

Few-Shot Prompting

LLMs may fail with complex tasks in a zero-shot setting

- Few-shot prompting allows **in-context learning**, where examples or demonstrations are included in the prompt to guide the model toward better performance
- These demonstrations help the model generate more accurate responses for similar tasks in subsequent prompts.

Example:

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:
We were traveling in Africa, and we saw these very cute whatpus.

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:

Output:

When we won the game, we all started to farduddle in celebration.

Limitations

Few-shot prompting is effective for many tasks but remains limited when handling complex reasoning tasks.

Zero-Shot Example:

The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.
A:

Output:

Yes, the odd numbers in this group add up to 107, which is an even number.

False! Let's try with few-shot prompting

Limitations

Few-shot prompting is effective for many tasks but remains limited when handling complex reasoning tasks.

Few-Shot Example:

The odd numbers in this group add up to an even number: 4, 8, 9, 15, 12, 2, 1.
A: The answer is False.

The odd numbers in this group add up to an even number: 17, 10, 19, 4, 8, 12, 24.
A: The answer is True.

The odd numbers in this group add up to an even number: 16, 11, 14, 4, 8, 13, 24.
A: The answer is True.

The odd numbers in this group add up to an even number: 17, 9, 10, 12, 13, 4, 2.
A: The answer is False.

The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.
A:

Output:

The answer is True.

False Again!

Chain-of-Thought Prompting

Enables complex reasoning capabilities through **intermediate reasoning steps**

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. X

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓

Chain-of-Thought Prompting

You can combine it with **few-shot prompting** to get better results on even more complex tasks

Example:

```
The odd numbers in this group add up to an even number: 4, 8, 9, 15, 12, 2, 1.  
A: Adding all the odd numbers (9, 15, 1) gives 25. The answer is False.  
The odd numbers in this group add up to an even number: 17, 10, 19, 4, 8, 12, 24.  
A: Adding all the odd numbers (17, 19) gives 36. The answer is True.  
The odd numbers in this group add up to an even number: 16, 11, 14, 4, 8, 13, 24.  
A: Adding all the odd numbers (11, 13) gives 24. The answer is True.  
The odd numbers in this group add up to an even number: 17, 9, 10, 12, 13, 4, 2.  
A: Adding all the odd numbers (17, 9, 13) gives 39. The answer is False.  
The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.  
A:
```

Output:

Adding all the odd numbers (15, 5, 13, 7, 1) gives 41. The answer is False.

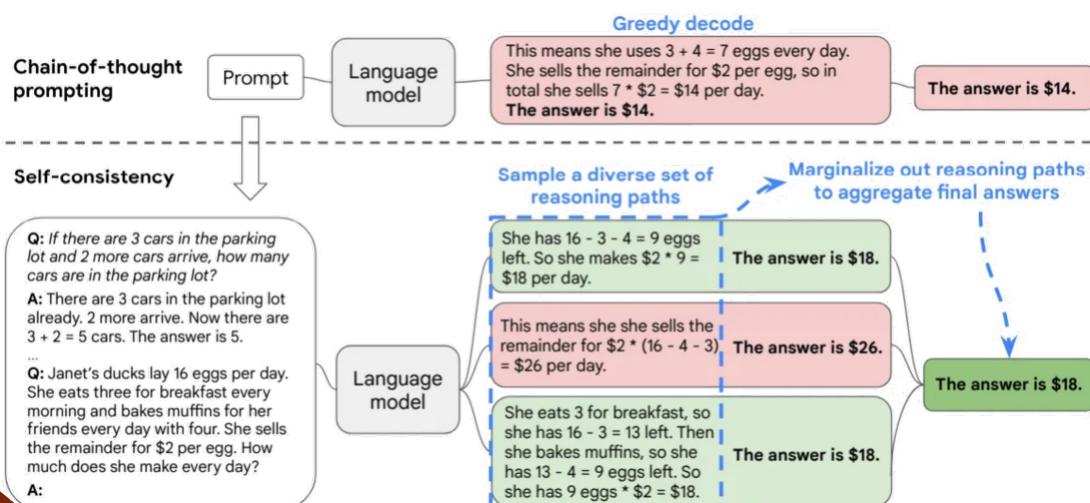
Correct!

Note: this is an emergent ability that arises with sufficiently large language models.

Self-Consistency Prompting

Uses an **iterative chain-of-thought** approach

Instead of accepting the first response from the LLM, the question is repeated **multiple times** to generate a range of reasoning paths, with **the most frequent answer being selected**



Meta Prompting

A prompting technique that **guides the model through the logical steps required to solve a problem** without relying on specific content-based examples

Example:

Solve the quadratic equation $3x^2 + 4x - 5 = 0$ by following these structured steps:

1. **Identify and Set Up the Formula:** Recognize that this is a quadratic equation in the form $ax^2 + bx + c = 0$.
2. **Apply the Quadratic Formula:** Use the quadratic formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ and substitute in the values.
3. **Calculate the Discriminant:** Find $b^2 - 4ac$, and interpret its value to determine the nature of the roots.
4. **Solve for x :** Perform the calculation for both $+$ and $-$ options in the formula to find the values of x .
5. **Verify the Solution:** Substitute each value of x back into the original equation to confirm it satisfies it.
6. **Summarize the Results:** State the final solution, listing both values of x and noting any observations about the nature of the roots.

Task-Agnostic Meta Prompting

Sometimes good results can be achieved by simply **asking the model to reason step-by-step**, without specifying the content of each step

Example:

Problem Statement:

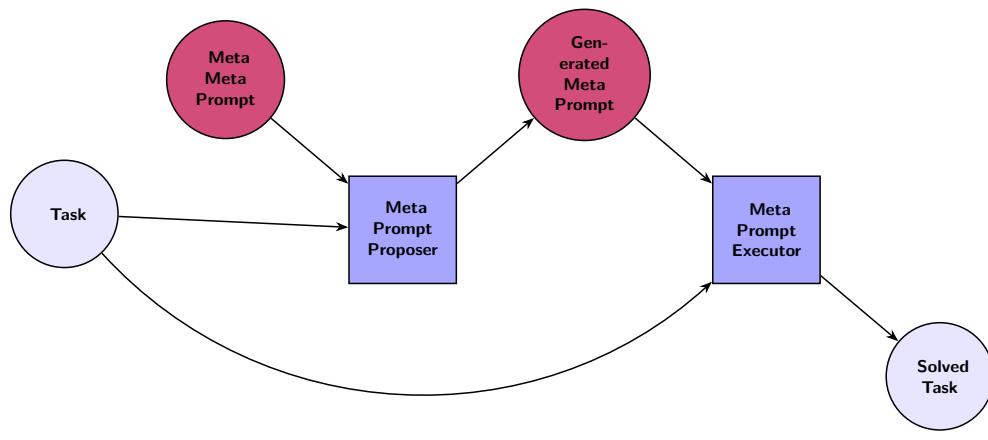
- **Problem:** [question to be answered]

Solution Structure:

1. Begin the response with "Let's think step by step."
2. Follow with the reasoning steps, ensuring the solution process is broken down clearly and logically.
3. End the solution with the final answer encapsulated in a LaTeX-formatted box, `...`, for clarity and emphasis.
4. Finally, state "The answer is [final answer to the problem].", with the final answer presented in LaTeX notation.

Meta Meta Prompting

This process entails **utilizing LLMs to generate prompts that guide further tasks**



Meta Meta Prompting

This process entails **utilizing LLMs to generate prompts that guide further tasks**

Example:

"Create a detailed metaprompt to guide an AI in solving a differential equation. Your metaprompt should focus on ensuring that the AI identifies the type of differential equation, selects the appropriate method for solving it, and presents a clear, step-by-step solution. Include instructions for verifying the solution and summarizing the results."

Output:

"To solve this differential equation, follow these steps carefully:

1. **Identify the Type:** Begin by identifying the type of differential equation (e.g., first-order linear, separable, homogeneous, etc.). Briefly explain how you recognized the type.
2. **Apply an Appropriate Method:** Based on the type, outline the most suitable method for solving it (e.g., separation of variables, integrating factor, etc.). If there are multiple approaches, describe why this particular method is the best choice.
3. **Show Step-by-Step Solution:** Work through the equation step-by-step, explaining each manipulation and transformation in clear terms.
4. **Verify Solution:** Once you find the solution, verify it by differentiating the result and substituting it back into the original equation to confirm it satisfies the differential equation.
5. **Summarize the Solution:** Finally, summarize the general or particular solution, highlighting any constants or special conditions that apply.

Prompt Chaining

For complex queries, sometimes using a single prompt is not enough

- Prompt chaining involves **breaking down a task into different parts**, each handled by a **specific prompt**
- The **initial prompt** covers only the first phase of the task
- The output obtained from the model is used as part of the input for the **next prompt**
- And so on, until reaching the **final result**

A common use case for this technique is **answering questions based on documents**

Prompt Chaining

Example:

Extract any relevant quotes from the following text to answer the question "What was the main cause of the American Civil War?" [reference text follows]

"One of the biggest catalysts of the Civil War was the issue of slavery, particularly the Southerners' desire to bring slavery into the territories." "Tensions peaked when the Republican candidate Abraham Lincoln was elected president in 1860 with a platform opposing the extension of slavery into the territories, which isolated the South."

Using the extracted quotes and the full text, provide a detailed answer to the question "What was the main cause of the American Civil War?"

The main cause of the American Civil War was the issue of slavery and its potential expansion into new territories and states. The South insisted on bringing slavery into new territories, while the election of Abraham Lincoln with an anti-slavery platform isolated the Southern states, which feared losing the right to own slaves. The fundamental disagreement over slavery between the North and South was the key factor that ultimately led to the outbreak of the Civil War

Role Prompting

Involves asking the model to **take on a specific role** or act in a certain way

- Can be used to **improve accuracy** in problem-solving
- In this way, **tone, style, and depth of information** provided is adjusted accordingly

Examples:

- Write a review of [pizza place]
- **You are a food critic writing for the Michelin Guide.** Write a review of [pizza place]
- **You are a marketing expert.** Draft an email to a client advising them about a delay in the delivery schedule due to logistical problems.
- **You are a customer service representative.** Draft an email to a client advising them about a delay in the delivery schedule due to logistical problems.

Role Prompting

- **Without RP:** *Write a review of [pizza place]*

I recently had the pleasure of trying [random pizza place] and I must say that I was thoroughly impressed. The pizza was cooked to perfection and the toppings were fresh and flavorful. The crust was light and crispy, and the sauce was just the right amount of tangy. The service was also top-notch; the staff was friendly and attentive. Overall, I had a great experience and would definitely recommend [random pizza place] to anyone looking for a delicious pizza.

- **With RP:** *You are a food critic writing for the Michelin Guide. Write a review of [pizza place]*

Located in the heart of [random city], [random pizza place] is a popular spot for pizza lovers. Upon entering, the warm, inviting atmosphere and the smell of freshly baked pizza will make you feel right at home. The menu offers a wide selection of classic pizzas, as well as some unique and creative options. The crust is perfectly crispy and the toppings are always fresh and flavorful. The service is friendly and attentive, and the prices are reasonable. Whether you're looking for a quick bite or a leisurely dinner, [random pizza place] is sure to satisfy your pizza cravings.

Role Prompting

- **You are a marketing expert.** Draft an email to a client advising them about a delay in the delivery schedule due to logistical problems.

Subject: A Little More Wait for Your Wonderful Products
Dear [Client's Name],
We hope this message finds you buzzing with excitement for your upcoming order! However, even the best-laid plans sometimes face unexpected hiccups. We have hit a temporary logistics snag that will unfortunately add a small wait time to your delivery.
While we understand this isn't ideal, we believe great things come to those who wait, and your order is no exception. Rest assured, from the moment we sort out this situation to the time your products reach you, we're committed to ensuring that your experience is nothing short of excellent.
Thank you for your patience and understanding. Stay tuned for more updates, and please, don't hesitate to reach out if you have any questions – we're always here.
Cheers, [Your Name]

focus on persuasion, positivity, and relationship

Role Prompting

- **You are a customer service representative.** Draft an email to a client advising them about a delay in the delivery schedule due to logistical problems.

Subject: Attention Required: Update on Your Order Status
Dear [Client's Name],
Greetings from [Your Company Name].
We wanted to inform you that, due to unforeseen logistical challenges, there might be a delay in the delivery of your order. We understand that this may cause some inconvenience, and we apologize sincerely.
Meanwhile, we are working tirelessly to resolve this and ensure your delivery reaches you by [new estimated date]. We value your time and patience greatly.
Please feel free to reach out if you have any questions or need further information. Our team is always here to assist you.
Thank you for your understanding.
Best regards, [Your Name]

More relational and solution-oriented

Structured Prompting

Based on a **semi-formal definition** of instructions to help LLMs handle complex tasks more predictably

- The prompt is divided in **sections** and encoded into a **script**
- **Delimiters** are used to help LLMs recognize each part of the prompt as distinct **units of meaning**
- Any unique **character sequence** that wouldn't normally appear together can serve as a delimiter
 - **###, ==, >>>**
- Another approach is to use **XML tags** as delimiters
 - LLMs are often **trained on web content** and have learned to recognize and understand this formatting

Example

Positive
Negative

Classify the sentiment of each conversation in <<<CONVERSATIONS>>> as 'Positive' or 'Negative'. Give the sentiment classifications without any other preamble text.

###

EXAMPLE CONVERSATIONS

[Agent]: Good morning, how can I assist you today?

[Customer]: This product is terrible, nothing like what was advertised!

[Customer]: I'm extremely disappointed and expect a full refund.

[Agent]: Good morning, how can I help you today?

[Customer]: Hi, I just wanted to say that I'm really impressed with your product. It exceeded my expectations!

###

EXAMPLE OUTPUTS

Negative

Positive

###

<<<

[Agent]: Hello! Welcome to our support. How can I help you today?

[Customer]: Hi there! I just wanted to let you know I received my order, and it's fantastic!

[Agent]: That's great to hear! We're thrilled you're happy with your purchase. Is there anything else I can assist you with?

[Customer]: No, that's it. Just wanted to give some positive feedback. Thanks for your excellent service!

[Agent]: Hello, thank you for reaching out. How can I assist you today?

[Customer]: I'm very disappointed with my recent purchase. It's not what I expected at all.

[Agent]: I'm sorry to hear that. Could you please provide more details so I can help?

[Customer]: The product is of poor quality, and it arrived late. I'm really unhappy with this experience.

>>>

Example

Positive
Negative

```
Classify the sentiment of the following conversations into one of two classes, using the examples given. Give the sentiment classifications without any other preamble text.  
<classes>  
Positive  
Negative  
</classes>  
<example-conversations>  
[Agent]: Good morning, how can I assist you today?  
[Customer]: This product is terrible, nothing like what was advertised!  
[Customer]: I'm extremely disappointed and expect a full refund.  
[Agent]: Good morning, how can I help you today?  
[Customer]: Hi, I just wanted to say that I'm really impressed with your product. It exceeded my expectations!  
</example-conversations>  
<example-classes>  
Negative  
Positive  
</example-classes>  
<conversations>  
[Agent]: Hello! Welcome to our support. How can I help you today?  
[Customer]: Hi there! I just wanted to let you know I received my order, and it's fantastic!  
[Agent]: That's great to hear! We're thrilled you're happy with your purchase. Is there anything else I can assist you with?  
[Customer]: No, that's it. Just wanted to give some positive feedback. Thanks for your excellent service!  
  
[Agent]: Hello, thank you for reaching out. How can I assist you today?  
[Customer]: I'm very disappointed with my recent purchase. It's not what I expected at all.  
[Agent]: I'm sorry to hear that. Could you please provide more details so I can help?  
[Customer]: The product is of poor quality, and it arrived late. I'm really unhappy with this experience.  
</conversations>
```

Structured Prompting

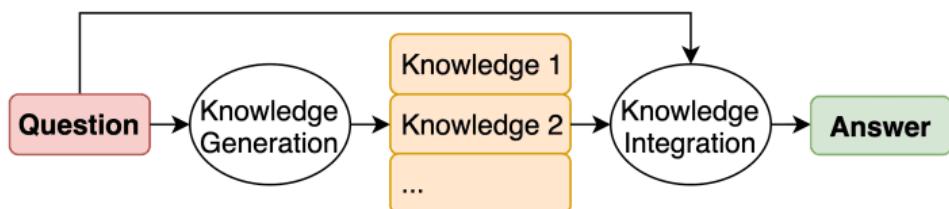
The **CO-STAR framework** divides a prompt into the following sections:

- **Context:** Provides background information on the task to help the LLM understand the specific scenario
- **Objective:** Clearly defines the task the LLM should perform
- **Style:** Specifies the desired writing style
- **Tone:** Sets the response's tone to match the desired sentiment or emotional context (e.g., formal, humorous, empathetic)
- **Audience:** Identifies the intended audience (experts, beginners, children, etc.)
- **Response:** Defines the response format to ensure compatibility with subsequent steps (e.g., free text, list, table, JSON)

Generate Knowledge Prompting

This method first prompts the LLM to **generate relevant knowledge** related to a task, and then **incorporates** that knowledge into the prompt along with the task description or question

- Particularly useful when the **LLM lacks the specific information required** to directly answer a query
- Leverages the LLM's **capacity to generate supplementary knowledge** beyond its base training domain

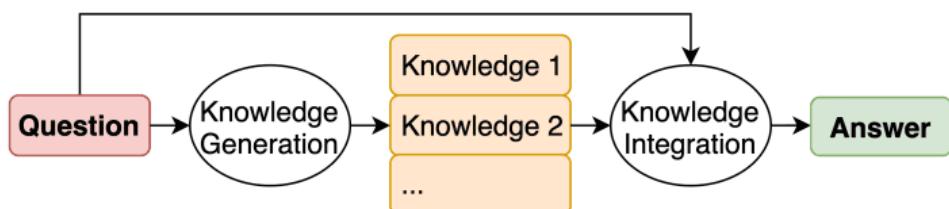


Generate Knowledge Prompting

Example:

List and describe the key factors that influence the evolution of life in environments with extreme gravitational forces, such as on a super-Earth planet. Focus on biological, physiological, and ecological adaptations that might arise in such conditions.

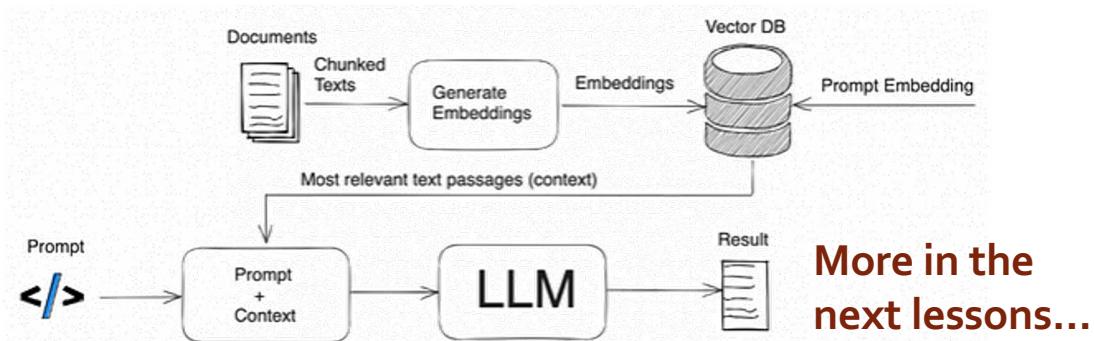
Using the adaptations and factors you described earlier, design a hypothetical intelligent species that could evolve on a super-Earth planet with extreme gravitational forces. Include details about their physical structure, social behaviors, methods of tool use or communication, and how their adaptations influence their culture or technology.



Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) **combines retrieval techniques with text generation**

- Addresses limitations in LLMs accessing **updated or domain-specific data**
- A **search or retrieval system** (e.g., databases, search engines) is used to find relevant documents or data
- An **LLM** is used to generate responses, conditioned on retrieved data



More in the
next lessons...

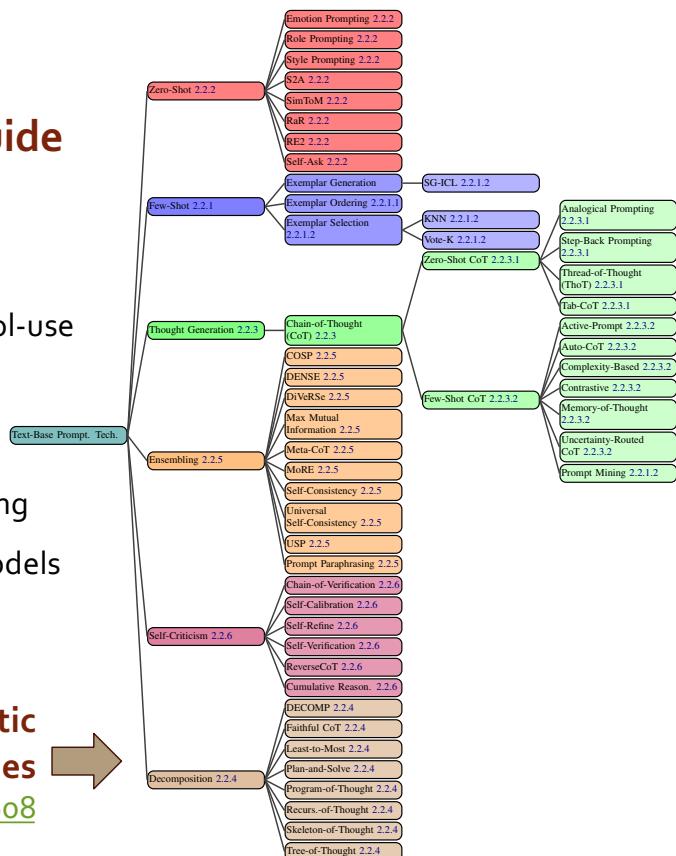
And more...

Prompt Engineering Guide

<https://www.promptingguide.ai/>

- Tree of Thoughts
- Automatic Reasoning and Tool-use
- Automatic Prompt Engineer
- Active-Prompt
- Directional Stimulus Prompting
- Program-Aided Language Models
- ...

The Prompt Report: A Systematic Survey of Prompting Techniques
<https://arxiv.org/abs/2406.06608>





Prompt Testing



Prompt Testing

Experimenting with various prompts is essential for achieving optimal responses across different use cases

- **Prompt Testing Tools...**
 - Simplify prompt creation and testing, enabling iterative adjustments to discover the best structure and format
 - Support customizable model settings to control output style, tone, and precision
- **Some available tools:**
 - **OpenAI Playground**: Supports **GPT** models
 - **Google AI Studio**: Supports Google **Gemini** models
 - **LM Studio**: Supports **Hugging Face** models

LLM Settings

When designing prompts, you **interact with the LLM via an API**, where you can adjust several **key parameters**:

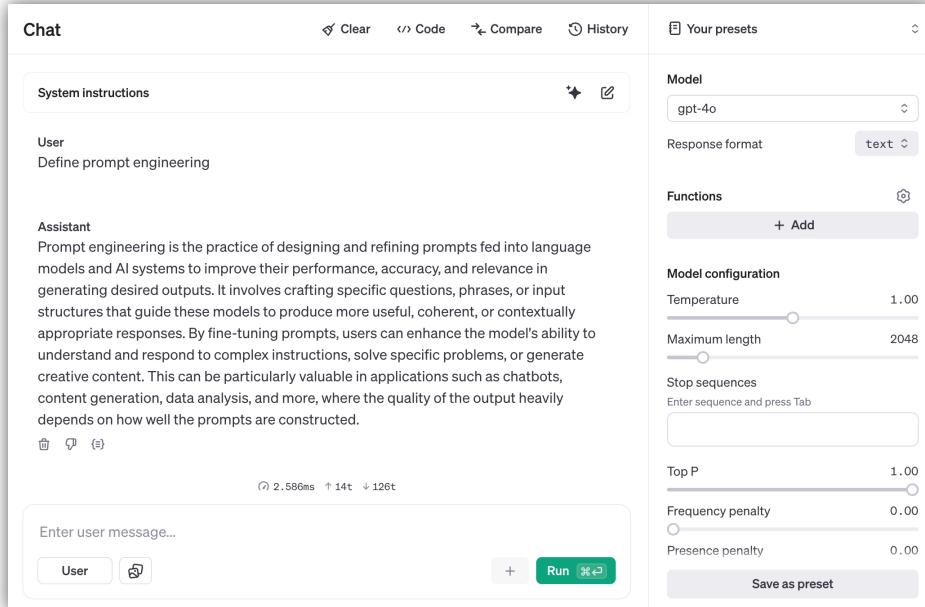
- **Temperature:** Controls **randomness**. Lower values (e.g., 0.2) make responses more deterministic, suitable for factual tasks. Higher values (e.g., 0.8) encourage creativity, ideal for tasks like poem generation.
- **Top P:** Adjusts response **diversity** by limiting token choices to a probability threshold. Lower values ensure precision, while higher values encourage more varied outputs.
 - **Example:** if **Top P = 0.5**, the model considers only the most probable tokens until their summed probability adds up to **50%**.
- **Max Length:** Sets the token limit for responses, helping to control response length and cost

LLM Settings

When designing prompts, you **interact with the LLM via an API**, where you can adjust several **key parameters**:

- **Stop Sequences:** Define a stopping point for responses, which can prevent overly long outputs and help structure responses, such as ending when a particular token is generated
- **Frequency Penalty:** Reduces repetition by penalizing words based on their frequency in the response, useful for avoiding redundant language
- **Presence Penalty:** Applies a consistent penalty to repeated tokens, regardless of how many times they appear. Higher values encourage more varied language
- **Response Format:** Expected format of the response (text, Json, ...)

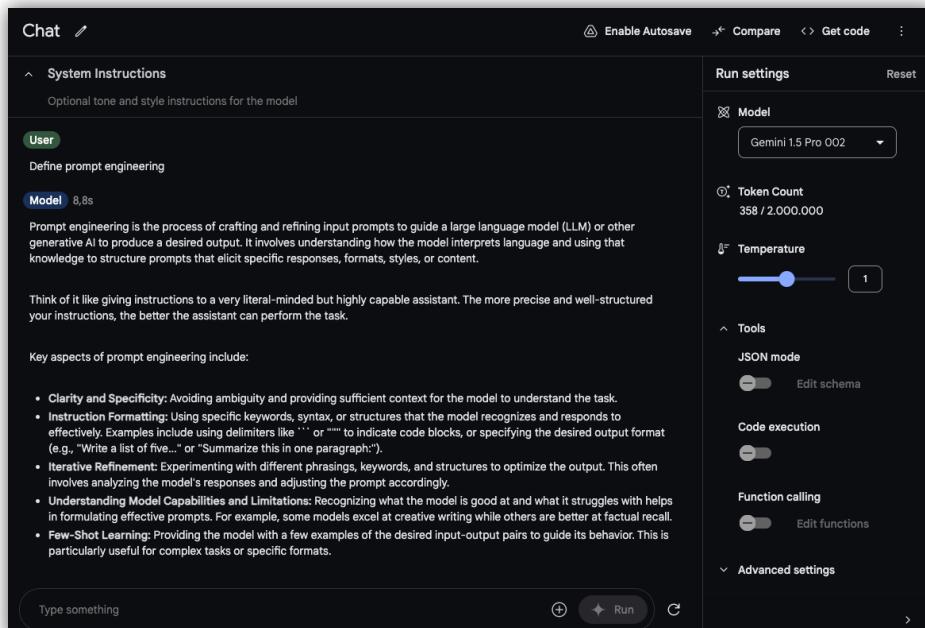
OpenAI Playground



The screenshot shows the OpenAI Playground interface. On the left, a chat window displays a user message: "Define prompt engineering". The assistant's response is: "Prompt engineering is the practice of designing and refining prompts fed into language models and AI systems to improve their performance, accuracy, and relevance in generating desired outputs. It involves crafting specific questions, phrases, or input structures that guide these models to produce more useful, coherent, or contextually appropriate responses. By fine-tuning prompts, users can enhance the model's ability to understand and respond to complex instructions, solve specific problems, or generate creative content. This can be particularly valuable in applications such as chatbots, content generation, data analysis, and more, where the quality of the output heavily depends on how well the prompts are constructed." On the right, the "Model" section is set to "gpt-4o" and "text". The "Functions" section has a "+ Add" button. The "Model configuration" section includes sliders for "Temperature" (1.00), "Maximum length" (2048), and "Stop sequences" (empty input field). Below these are "Top P" (1.00), "Frequency penalty" (0.00), and "Presence penalty" (0.00). A "Save as preset" button is at the bottom.

<https://platform.openai.com/playground/>

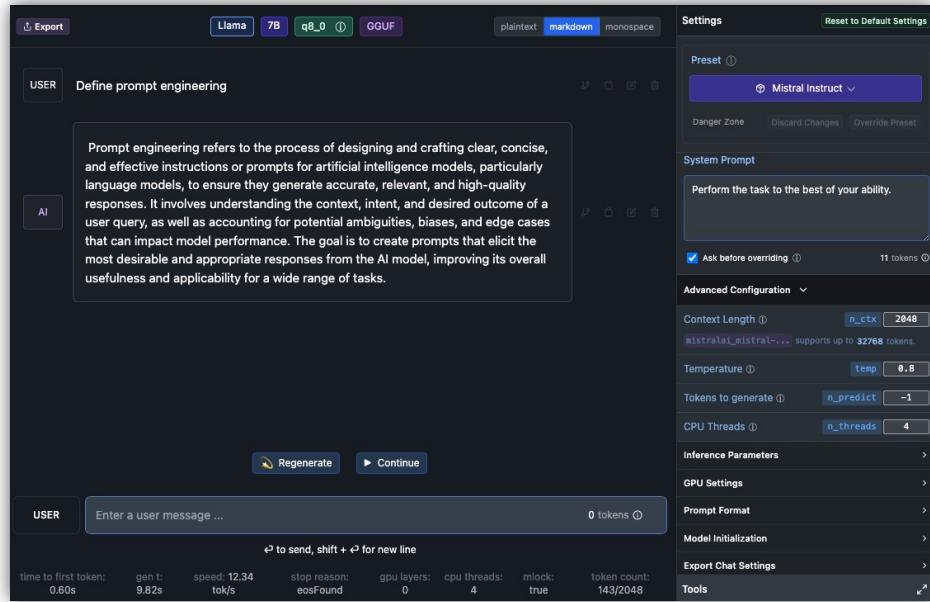
Google AI Studio



The screenshot shows the Google AI Studio interface. The chat window displays a user message: "Define prompt engineering". The assistant's response is: "Prompt engineering is the process of crafting and refining input prompts to guide a large language model (LLM) or other generative AI to produce a desired output. It involves understanding how the model interprets language and using that knowledge to structure prompts that elicit specific responses, formats, styles, or content." Below the response, a note says: "Think of it like giving instructions to a very literal-minded but highly capable assistant. The more precise and well-structured your instructions, the better the assistant can perform the task." A list of "Key aspects of prompt engineering include:" follows, with bullet points: "Clarity and Specificity: Avoiding ambiguity and providing sufficient context for the model to understand the task.", "Instruction Formatting: Using specific keywords, syntax, or structures that the model recognizes and responds to effectively. Examples include using delimiters like '```' or '```' to indicate code blocks, or specifying the desired output format (e.g., 'Write a list of five...' or 'Summarize this in one paragraph.').", "Iterative Refinement: Experimenting with different phrasings, keywords, and structures to optimize the output. This often involves analyzing the model's responses and adjusting the prompt accordingly.", "Understanding Model Capabilities and Limitations: Recognizing what the model is good at and what it struggles with helps in formulating effective prompts. For example, some models excel at creative writing while others are better at factual recall.", "Few-Shot Learning: Providing the model with a few examples of the desired input-output pairs to guide its behavior. This is particularly useful for complex tasks or specific formats." On the right, the "Run settings" sidebar shows "Model" set to "Gemini 1.5 Pro 002", "Token Count" (358 / 2,000,000), and "Temperature" (1). Other sections include "Tools" (JSON mode, Code execution, Function calling), "Advanced settings", and a "Compare" tab.

<https://aistudio.google.com>

LM Studio



<https://lmstudio.ai/>

LM Studio

A Desktop Application for Local LLM Development and Experimentation

Key functionality:

- **Search & Download Models** directly from Hugging Face
- **Run LLMs** on your computer
- **Interactive Chat Interface** to test and interact with LLMs
- **Local API Server** enabling LLM integration with external applications
- **Model Management Tools** to organize and configure local models

Additional info on <https://lmstudio.ai/>



Natural Language Processing and Large Language Models

Corso di Laurea Magistrale in Ingegneria Informatica

Lesson 18



Prompt Engineering

Nicola Capuano and Antonio Greco

DIEM – University of Salerno

.DIEM