

Advanced Generative AI: Models, Tools and Applications



Advanced Prompt Engineering Techniques: Part 1



Quick Recap



- What is LangChain, and how is it compatible with developing language model-based applications in Python and JavaScript?
- What does the workflow of a generative AI application involve in terms of creating original data or content?

Engage and Think



As a software developer at a company focused on AI-driven educational tools, you are currently developing an interactive AI tutor tailored for high school students. This AI tutor is innovative in its approach, with the ability to adjust its teaching methods to fit the unique learning styles and preferences of each student, thereby offering personalized education. It aims to transform the educational experience, particularly for students who find traditional teaching methods challenging.

How will the AI tutor assess and adapt to the varying learning styles and preferences of individual students to effectively personalize the educational content?

Learning Objectives

By the end of this lesson, you will be able to:

- 🔗 Analyze the fundamental aspects of prompt creation to understand their role in guiding language models
- 🔗 Develop a thorough comprehension of the key principles and elements that are foundational to prompt design in language models
- 🔗 Apply and synthesize best practices and techniques for zero-shot and few-shot prompting to enhance the utility of language models

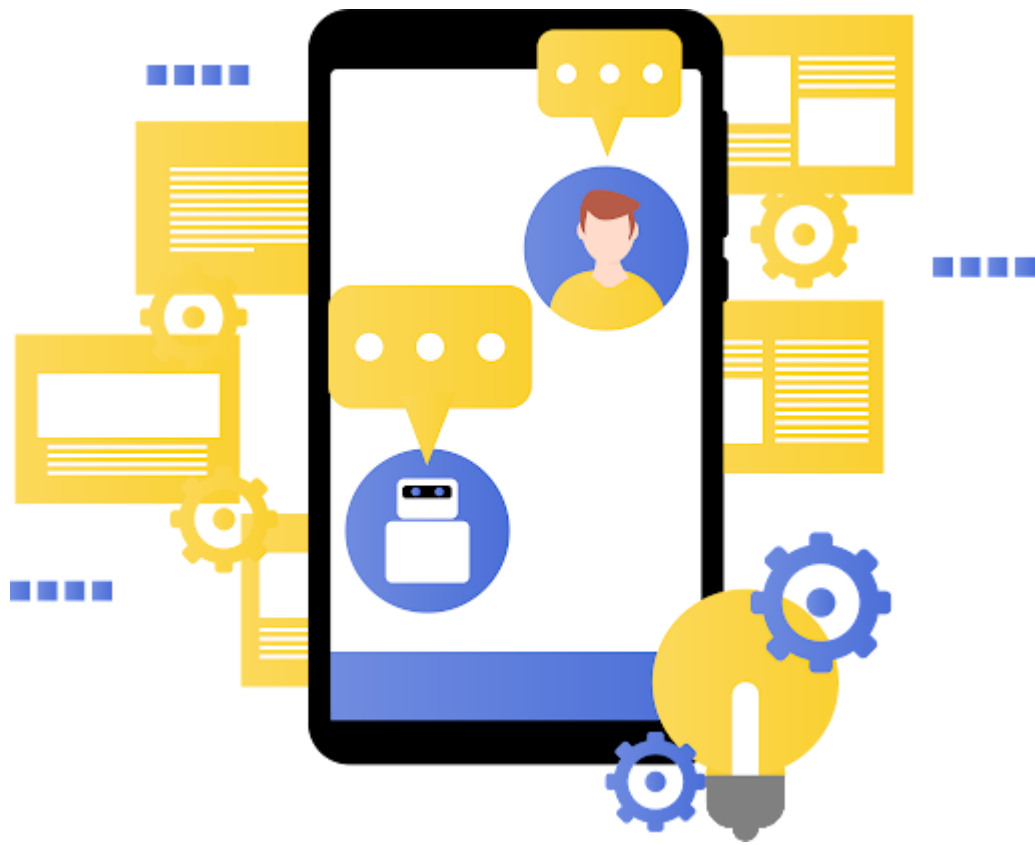




Prompt Engineering

Prompt Engineering

Prompt engineering is a crucial skill in the field of AI, as it helps users better understand the capabilities and limitations of LLMs and develop robust and effective prompting techniques.



- This field involves the creation and refinement of prompts, which are instructions given to AI models, to generate specific and meaningful responses.
- It serves as the interface between human intent and machine output, guiding AI models towards desired outputs.

Prompt Engineering: Example

Here is an illustration of the importance of specific and detailed prompts for obtaining precise responses from LLMs.

Output:

Prompt:

The ocean is



You

The ocean is



ChatGPT

The ocean is an immense and complex ecosystem, covering more than 70% of the Earth's surface. It's not only a vital source of biodiversity and life, but also plays a crucial role in regulating the planet's climate and weather patterns. The ocean supports a myriad of species, from microscopic plankton to the largest mammals like whales. Its depths are largely unexplored, filled with mysteries and a vast array of life forms, many of which are yet to be discovered.

Note

To ensure the output meets the intended task, provide more context or explicit instructions within the prompt for more targeted results.

Basics of Prompt Engineering

The prompting lies at the core of harnessing the potential of LLMs, offering a pathway to interact with these sophisticated systems effectively.

- Effective communication with LLMs is built on simple prompts, with the quality of the result depending on the information's precision and clarity.
- A basic prompt comprises instructions or questions and may include contextual details, inputs, or examples, all of which contribute to refining the model's response.

Prompt Engineering: Example

Consider a prompt engineer tasked with generating a concise summary of J.R.R. Tolkien's The Lord of the Rings using an LLM.

An effective prompt for this task might look like:

Write a concise summary of the book 'The Lord of the Rings' by J.R.R. Tolkien, focusing on the main characters and their journeys.



You

Write a concise summary of the book 'The Lord of the Rings' by J.R.R. Tolkien, focusing on the main characters and their journeys."



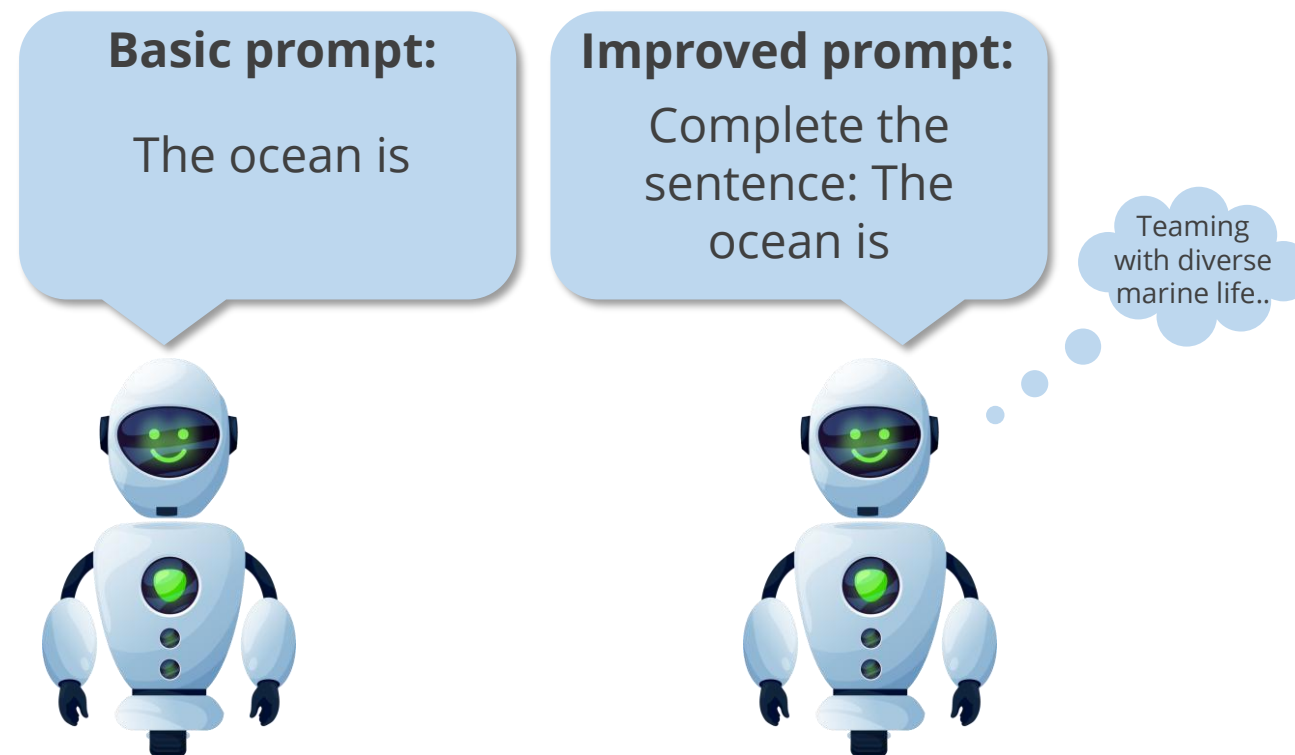
ChatGPT

"The Lord of the Rings" by J.R.R. Tolkien is an epic fantasy novel that follows the perilous journey of Frodo Baggins, a young hobbit who inherits a powerful ring from his cousin Bilbo. Frodo, guided by the wise wizard Gandalf, is joined by his loyal friends Sam, Merry, and Pippin. Together, they set out from the Shire to destroy the One Ring in the fires of Mount Doom to prevent the dark lord Sauron from conquering Middle-earth.

As they journey, they forge alliances with Aragorn, the rightful king of Gondor; Legolas, an elven prince; Gimli, a stalwart dwarf; and Boromir, a valiant warrior. Together, they form the Fellowship of the Ring. Their quest is fraught with challenges, from the treacherous terrain of the Mines of Moria to the battlefields of Helm's Deep.

Optimizing Basic Prompts

When optimizing basic prompts, the goal is to transform vague or general instructions into clear, specific directives that lead to more accurate and relevant responses from the model.



This refinement process involves utilizing targeted language that narrows the scope of the model's task, resulting in outputs that are not only precise but also aligned with the user's specific intent.

Quick Check



What is the primary objective of prompt engineering in the context of interacting with large language models (LLMs)?

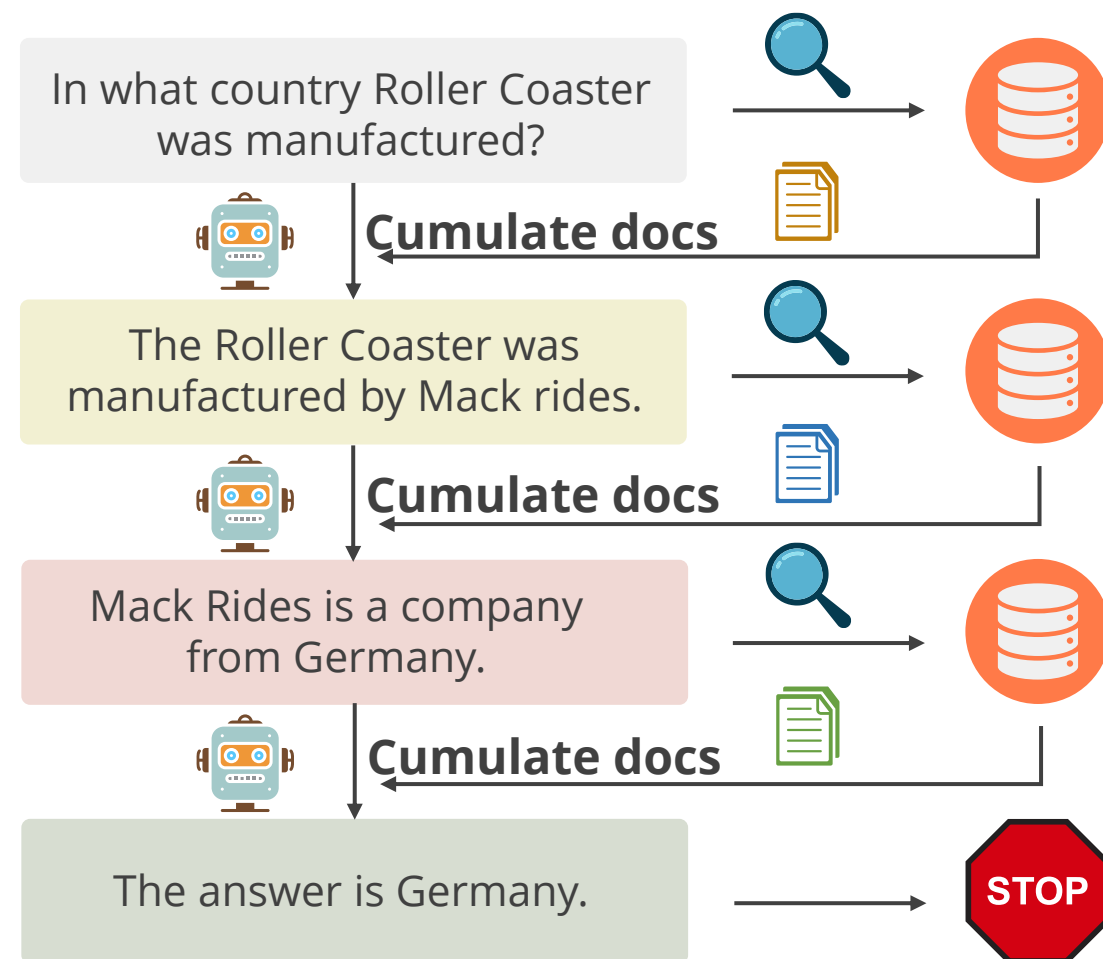
- A) To program the LLMs with new coding languages
- B) To create prompts that are ambiguous and open-ended
- C) To develop prompts that guide LLMs to produce specific and meaningful responses aligned with human intent
- D) To collect data from various AI models for comparative analysis



Advanced Prompt Engineering

Advanced Prompt Engineering

It refers to the development of more sophisticated techniques for interacting with language models beyond the basics.



While basic prompt engineering might involve straightforward instructions or questions, advanced prompt engineering delves into complex strategies to improve the model's understanding and generate more accurate or nuanced outputs.

The next step is fine-tuning LLM settings to optimize prompting for superior model performance.

LLM Settings for Optimal Prompting

The effectiveness of crafted prompts is further enhanced by understanding LLM settings, ensuring that AI responses are finely tuned to align with specific communication goals.

Temperature:

- Set this parameter to control the predictability of the model's responses.
- Lower the temperature for more predictable outcomes or increase it for added creativity and variety in the outputs.

LLM Settings for Optimal Prompting

The following essential adjustments to LLM settings are instrumental in tailoring the AI's responses to enhance precision, relevance, and efficiency in text generation.

Top_p and Max Length:

- Adjust the **Top_p** setting to control the breadth of the model's responses; a lower value increases precision, and a higher value enhances diversity.
- Modify the **Max Length** to manage the token count, ensuring outputs remain concise and relevant to the content requirements.

Stop sequences:

- Define stop sequences to end token generation at specific points, effectively structuring the output.
- For example, to limit a list to ten items, add **11** as a stop sequence.

Top_p and Max Length: Example

Here are examples to demonstrate the effect of adjusting top_p and max_length settings in an LLM:

Low top_p (high precision)

The tree told the travelers about the forest's history and offered wise advice. The travelers thanked the tree and continued their journey.

High top_p (high diversity)

The tree spoke in rhymes, telling tales of mythical creatures and ancient spells. It revealed a hidden path that led to an enchanted glade filled with glowing flowers and magical animals.

Short max_length

Photosynthesis is how plants make food using sunlight, water, and carbon dioxide. It releases oxygen as a by-product.

Long max_length

Photosynthesis is the process by which plants convert sunlight into energy. Using water from the soil and carbon dioxide from the air, plants produce glucose, a type of sugar that.....

LLM Settings for Optimal Prompting

Explore strategies for refining language model outputs by employing penalties that reduce repetition and enhance linguistic variety in generated texts.

Frequency penalty:

- Apply this penalty to decrease repetition.
- It penalizes tokens based on their frequency in the prompt and response, thereby enhancing the diversity of the language used.

Presence penalty:

- Implement this penalty to prevent repeated tokens, irrespective of their frequency, promoting original and varied outputs from the model.

LLM Settings for Optimal Prompting: Example

This is an example of how the design of the prompt can influence the quality of the response from the language model.

Summarize the key causes of climate change in 2-3 sentences.



Climate change is primarily caused by the accumulation of greenhouse gases such as carbon dioxide, methane, and nitrous oxide in the atmosphere, largely due to human activities like burning fossil fuels, deforestation, and industrial processes. These gases trap heat, leading to global warming and disrupted weather patterns. Natural factors like volcanic eruptions and variations in solar activity contribute to a lesser extent but are significantly overshadowed by anthropogenic influences.

This approach exemplifies prompt engineering, a pivotal practice for achieving specific objectives through model interactions.

By instructing the model to complete the sentence, the output aligns more closely with the desired task.

Prompt Formatting: Crafting the Right Structure

Understanding the structure of prompts contributes significantly to their effectiveness.

A standard prompt can take the form of a question or instruction:

<Question>?

or

<Instruction>

For question-answering (QA) tasks, a common format is:

Q: <Question>?

A:

Quick Check



What does setting a lower temperature value in prompt engineering influence the model to do?

- A. Generate more random and diverse responses
- B. Produce more predictable and likely responses
- C. Increase the length of the generated content
- D. None of the above



Prompt Elements

Prompt Elements

Prompt elements refer to the components that constitute a complete and effective prompt when interacting with AI models, especially LLMs.

The main prompt elements typically include:



Instruction

Context

Input data

Output indicator

Prompt Elements: Instruction

Instruction is a specific task or directive intended for the model and guides the model toward the desired action or response.

Scenario:

Imagine you're creating a Japan itinerary during cherry blossom season. How would you ensure tourists experience the best of this season by including cultural sites, traditional food experiences, and cherry blossom viewing events?

Prompt:

Create a Japan itinerary for cherry blossom season highlighting cultural sites, local food, and blossom events.

Prompt Elements: Context

Context is external information or additional context supplementing the prompt and used to enhance the model's understanding, refining its responses.

Scenario:

For this solo traveler seeking a deeper cultural connection in Japan, what additional context would you add to enrich the itinerary? How would you adjust the details to cater to a journey of self-discovery?

Prompt:

The journey is a solo adventure centered on cultural immersion, including participation in local festivals and traditional workshops.

Prompt Elements: Input Data

The input data is an input or question driving the quest for a model-generated response and forms the core inquiry around which the model's output revolves.

Scenario:

Given these preferences, what input data should you provide to our AI? How would these interests influence the activities and events chosen for the itinerary?

Prompt:

The individual will be in Japan from April 1st to April 14th is interested in traditional arts, and seeks to participate in local customs and festivities.

Prompt Elements: Output Indicator

The output indicator specifies the expected type or format of the model's output and sets the criteria for evaluating the relevance and correctness of the response.

Scenario:

What would be the indicators of a successful itinerary for this culturally rich solo adventure? List the elements that would reflect a well-rounded experience, from festival dates to workshop locations.

Prompt:

The itinerary should offer a blend of scheduled cultural activities, opportunities for impromptu engagement with local traditions, and free periods for personal exploration, with a clear indication of when the cherry blossoms will be in full bloom.

Prompt Elements: Example

This prompt is a synthesis of the four essential aspects of prompt crafting namely instruction, context, input data, and output indicator meticulously woven into a single, cohesive instruction set for the AI to follow:

Prompt:

Craft a detailed itinerary for a culturally rich solo trip in Japan from April 1st to April 14th, tailored for a traveler interested in traditional arts, local customs, and festivals. Include diverse cultural sites, authentic food experiences, cherry blossom viewing events, and flexible periods for personal exploration, ensuring the peak bloom period is highlighted.

Quick Check



Which of the following is NOT an element of a prompt?

- A. Instruction
- B. Context
- C. Algorithm
- D. Output indicator



Types of Prompting Techniques

Types of Prompting Techniques

Prompting techniques are the different methods used to guide AI language models to generate the desired output.

Below are a few main types of prompting techniques:



Zero-shot
prompting

Few-shot
prompting

Chain of thought
prompting



Zero-Shot Prompting Technique

Zero-Shot Prompting Technique

Zero-shot prompting involves presenting a model with a prompt or task that it has not been explicitly trained to handle.

- Even without specific examples, these models can decipher and perform certain tasks, known as **zero-shot**.
- The term **zero-shot** comes from the idea that the model attempts the task with **zero** prior shots or attempts.
- It exploits the innate understanding of tasks embedded in large language models.
- These models, having undergone extensive training on diverse datasets, can interpret and execute instructions without explicit demonstrations.

Zero-Shot Prompting Technique: Example

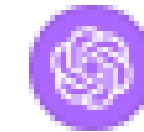
The zero-shot capability empowers LLMs to grasp abstract concepts like sentiment analysis from a prompt.

Prompt:

Classify the below sentence into neutral, negative, or positive.

Text: I think the holidays are okay.

Output:



ChatGPT

Classification:

Neutral

In the above example, the model effortlessly discerns sentiment without the need for prior examples.

Enhancing Zero-Shot Learning

To further optimize zero-shot learning, researchers have introduced the concept of instruction tuning.

This involves fine-tuning models on datasets annotated with instructions, providing a more nuanced understanding of specific tasks.

Reinforcement learning from human feedback (RLHF) plays a crucial role in scaling instruction tuning and aligning models with human preferences.

Models like ChatGPT leverage these advancements to deliver more refined outputs.

When Zero-Shot Falls Short

While zero-shot prompting is impressive, there are instances where it might fall short.

In such cases, incorporating demonstrations or examples in the prompt becomes essential, transitioning into few-shot prompting.

The next section will illuminate the effectiveness of few-shot prompting in overcoming the limitations of pure zero-shot approaches.

Demo: Zero-Shot Prompting with LangChain and OpenAI



Duration: 05 minutes

Overview:

In this project, you will explore the concept of zero-shot prompting using LangChain and OpenAI. This will help you understand how to use these powerful tools to generate creative and engaging content.

Note

Please download the solution document from the Reference Material Section and follow the Jupyter Notebook for step-by-step execution.

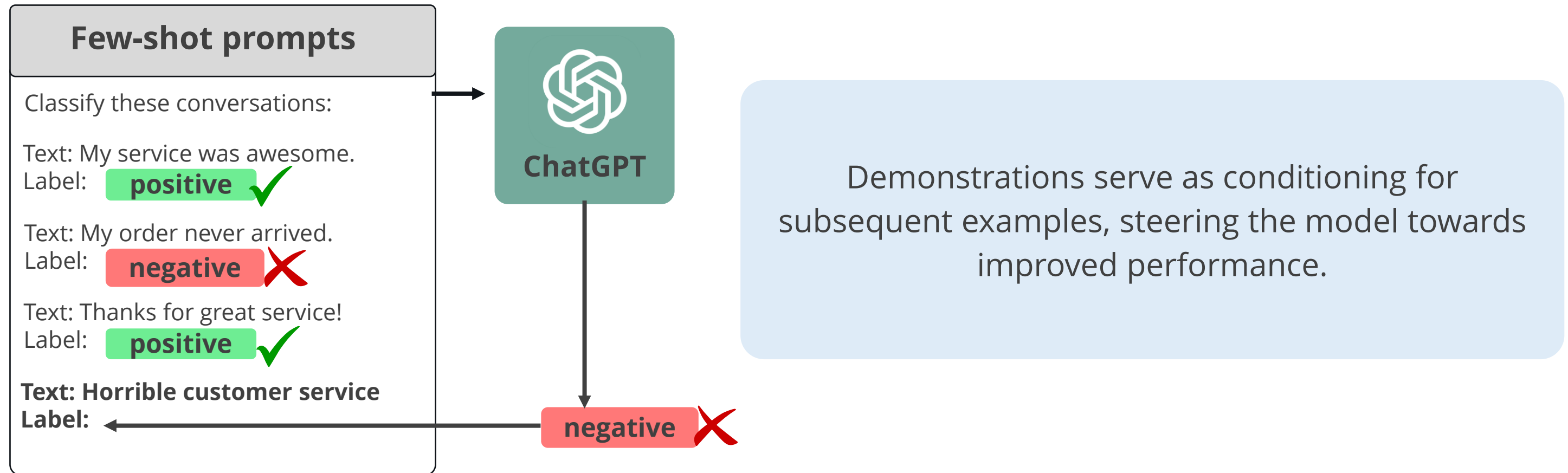
DEMONSTRATION



Few-Shot Prompting

Few-Shot Prompting

Few-shot prompting is a technique that introduces in-context learning by providing demonstrations within the prompt.



Few-Shot Prompting: Example

Prompt:

An **awoomba** is a mythical creature that can only be found in the Amazon rainforest. An example of a sentence that uses the word awoomba is:
We embarked on an expedition, hoping to catch a glimpse of the elusive awoombas.
To **flibberate** means to dance joyfully in the moonlight. An example of a sentence that uses the word flibberate is:



ChatGPT

Under the moonlit sky, we began to flibberate, each step and twirl releasing our worries into the night.

Note

Here, the model adeptly learns the task with just one example (1-shot). Experimenting with increasing demonstrations (3-shot, 5-shot, and 10-shot) becomes crucial for more intricate tasks.

Optimizing Few-Shot Prompting



1. The label space and input text distribution specified by demonstrations are both vital.
2. Format plays a crucial role; even using random labels is better than none.
3. Selecting random labels from a true distribution enhances performance.

Optimizing Few-Shot Prompting

Experimentation with random labels:

Prompt:

This is amazing! // Negative
This is worse! // Positive
Wow, that movie was not great! // Positive
What a horrible web series! //?

Output:



ChatGPT

The classification for the last statement should be:

What a horrible web series! // Negative

Remarkably, even with randomized labels, the model produces correct answers. Consistency in format enhances results, showcasing the models' increasing robustness.

Few-Shot Prompting Challenges

While few-shot prompting excels in various tasks, complexities arise in more intricate reasoning problems.

Take the example of odd numbers adding up to an even number:

Prompts

Odd numbers in this sequence sum up to an even number: 3, 7, 15, 21, 8, 11, 4.

Odd numbers in this sequence sum up to an even number: 5, 12, 19, 25, 10, 13, 3.

Odd numbers in this sequence sum up to an even number: 6, 14, 21, 9, 11, 17, 7.

Odd numbers in this sequence sum up to an even number: 8, 16, 24, 10, 13, 18, 5.

Odd numbers in this sequence sum up to an even number: 15, 32, 5, 13, 27, 7, 1.

Few-Shot Prompting Challenges

Observation:

- For more complex tasks like this, traditional few-shot prompting falls short.
- The introduction of chain-of-thought (CoT) prompting has gained popularity in addressing complex arithmetic, commonsense, and symbolic reasoning tasks.
- Few-shot prompting proves invaluable for numerous tasks. When faced with its limitations, fine-tuning models or exploring advanced prompting techniques becomes the next logical step.



Note

It is important to note that LLMs are still not designed for performing math in the way a calculator does. Instead, they process math as if they are working with number-shaped words, which can lead to inaccuracies in arithmetic or complex calculations.

Demo: Few-Shot Prompting with LangChain and OpenAI



Duration: 05 minutes

Overview:

In this project, you will explore the concept of few-shot prompting using LangChain and OpenAI. Few-shot prompting is a technique where you provide a model with a few examples of the task you want it to perform, and then ask it to perform a similar task. This project will help you understand how to use these powerful tools to generate creative and engaging content.

Note

Please download the solution document from the Reference Material Section and follow the Jupyter Notebook for step-by-step execution.

DEMONSTRATION

Quick Check



What term describes a prompting technique that allows language models to generate responses for tasks they haven't been specifically trained on?

- A. Zero-shot prompting
- B. Few-shot prompting
- C. Chain of thought prompting
- D. Tree of thoughts prompting

Guided Practice



Overview

Duration: 20 minutes

The task is to demonstrate the implementation of few-shot prompting using LangChain and OpenAI. The objective is to showcase the versatility and effectiveness of this approach in generating creative content, translating sentences, writing product reviews, and describing culinary dishes based on given prompts.

Steps to Perform:

1. Set up the OpenAI API Key to access the OpenAI API for language model interaction.
2. Create a function that sends a prompt to the OpenAI model and retrieves its response. This function should include parameters for the prompt text and the model to be used.
3. Construct prompts for various tasks, such as translating sentences, generating short stories, writing product reviews, and describing culinary dishes. These prompts should include examples and instructions to guide the model in producing the desired output.

Key Takeaways

- 🔗 Prompt engineering is a crucial skill in the field of AI, as it helps users understand the capabilities and limitations of LLMs.
- 🔗 The prompting lies at the core of harnessing the potential of LLMs, offering a pathway to interact with these sophisticated systems effectively.
- 🔗 Prompt elements refer to the components that constitute a complete and effective prompt when interacting with AI models, especially LLMs.



Q&A

