The Subtle Art of Prompting

Mastering the Art of Interacting with Large Language Models

By Malika Malik

Last Updated: April 2024

logo.png

A Comprehensive Course on Prompt Engineering

Contents

Preface **3**

1 Introduction to Prompt Engineering **3**
1.1 Welcome and Course Overview . . . 3
1.2 How to Make the Most of This Course . . . 3
1.3 Why Learn Prompt Engineering Today? . . . 4
1.4 What You'll Learn in This Course . . . 4
1.5 About the Author . . . 4

2 Module 1: Fundamentals of LLMs and Prompting **5**
2.1 Understanding Large Language Models (LLMs) . . . 5
2.2 LLMs vs. Traditional AI Models . . . 5
2.3 What is a Prompt? . . . 6
2.4 How Prompting Works . . . 6
2.5 Fundamental Principles of Effective Prompting . . . 6
2.5.1 Clarity . . . 6
2.5.2 Specificity . . . 7
2.6 Introduction to Google AI Studio . . . 7
2.6.1 Key Features of Google AI Studio . . . 7
2.7 Module 1 Exercise: Clarity vs. Vague Prompts . . . 8

3 Module 2: Advanced Prompting Strategies **8**
3.1 Three Fundamental Strategies of Prompt Engineering . . . 8
3.2 Strategy 1: Give Clear Instructions . . . 9
3.2.1 Why Clear Instructions Matter . . . 9
3.3 Strategy 2: Use System Instructions . . . 9
3.3.1 How to Use System Instructions Effectively . . . 10
3.4 Strategy 3: Simplifying Complex Prompts . . . 11
3.4.1 Two Key Strategies for Simplification . . . 11
3.4.2 Advantages of Breaking Down Complex Prompts . . . 12
3.5 Module 2 Exercise: System Instructions and Prompt Chaining . . . 13

4 Module 3: Leveraging Examples in Prompts **13**
4.1 The Power of Examples in Prompting . . . 13
4.2 Zero-Shot, One-Shot, and Few-Shot Prompting . . . 14
4.3 Example: Sentiment Analysis Using Different Prompting Techniques . . . 15
4.4 How Many Examples Are Enough? . . . 15
4.5 Key Takeaways for Providing Examples . . . 15
4.6 Real-World Applications of Few-Shot Learning . . . 16
4.7 Module 3 Exercise: Utilizing Few-Shot Examples . . . 17

Copyright © 2024 Malika Malik

5 Module 4: Advanced Techniques 17
5.1 Chain of Thought (CoT) Prompting 17
5.2 Zero-Shot Chain of Thought Prompting 18
5.3 Prompt Chaining 19
5.3.1 Why Use Prompt Chaining? 19
5.3.2 When to Use Prompt Chaining? 19
5.3.3 How to Chain Prompts 19
5.4 Module 4 Exercise: Zero-Shot Chain of Thought Prompting 21

6 Module 5: Automatic Prompt Engineering 21
6.1 Introduction to Automatic Prompt Engineering (APE) 21
6.2 How Automatic Prompt Engineering Works 22
6.2.1 The Two-Model Approach 22
6.2.2 The APE Process 22
6.3 Results and Implications 22
6.4 Course Summary: Key Strategies for Effective Prompt Engineering 23

7 Additional Resources 24
7.1 Recommended Reading 24
7.2 Online Resources 24

8 Glossary of Terms 24

Preface

Welcome to ”The Subtle Art of Prompting”! This comprehensive course is designed to equip you with the skills needed to effectively interact with Large Language Models (LLMs). In today’s rapidly evolving technological landscape, the ability to communicate clearly and effectively with AI systems has become increasingly important across various industries and professions.

As AI continues to transform our world, understanding how to craft effective prompts has evolved from a niche skill to an essential capability. This course aims to demystify the process of prompt engineering, providing you with practical strategies and techniques that you can apply immediately in your personal and professional endeavors.

Whether you’re a seasoned tech professional or just beginning your journey with AI, this course will provide you with valuable insights and practical skills to enhance your ability to work with LLMs.

1 Introduction to Prompt Engineering

1.1 Welcome and Course Overview

Hello & Welcome! I’m Malika Malik, and I’m thrilled to embark on this journey into the world of Large Language Models (LLMs) with you!

My own journey with AI began over a decade ago, during my time studying at the London School of Economics, sparking an interest for exploring its potential within the tech industry. Over the last few years, I’ve witnessed firsthand the power of LLMs. It’s become clear to me that knowing how to interact with them effectively is no longer a niche skill – it’s becoming essential for every career path. Whether you dream of being an engineer, a marketing specialist, or anything in between, understanding LLMs is key to unlocking a world of possibilities.

That’s why I designed this course – to demystify the art of communicating with LLMs and empower you to harness their full potential. Think of it as your introduction to the world of prompting – giving LLMs the right instructions to get the results you’re looking for.

1.2 How to Make the Most of This Course

To get the maximum benefit from this course, I recommend:

- **Active Engagement**: Set aside dedicated time to read through each module and, most importantly, practice the prompts.
- **Hands-On Practice**: Apply what you learn by experimenting with the example prompts in tools like Google AI Studio.
- **Reflection**: Take note of what works well and what doesn’t in your prompting experiments.

Copyright © 2024 Malika Malik

Remember, practice is the key to mastering this skill!

1.3 Why Learn Prompt Engineering Today?

Do you remember when the Harvard Business Review called Data Scientist the ”sexiest job of the 21st century”? Now, Prompt Engineering is the new rising tech star. Prompt engineering blends technical expertise with creative and strategic thinking, enabling us to communicate effectively with AI systems.

Here’s why it matters:

- **Explosive Growth**: In just two years, the demand for prompt engineering skills has skyrocketed by over 300%! This course will equip you with the skills needed to thrive in this field.
- **Essential Skills**: As AI integrates into our daily lives, learning how to communicate with AI effectively is vital. This course is your chance to stay ahead in the ever-evolving tech landscape.
- **Beyond Simple Answers**: Sure, asking ”What is 2+2?” is easy, but what about guiding an AI to generate specific code? That’s where prompt engineering shines, helping you craft queries that lead to the desired outcomes.
- **A Skill, Not Just a Role**: Whether or not prompt engineering becomes a standalone job, it’s a valuable skill across many roles. As AI continues to evolve, being able to guide these systems will set you apart.

1.4 What You’ll Learn in This Course

Throughout this course, you’ll develop:

- **Principles for Prompting Mastery**: Develop a solid foundation with principles that will serve as your guiding framework for crafting prompts that deliver the desired results.
- **Advanced Techniques**: After mastering the basics, delve into techniques like prompt chaining and Automatic Prompting Engineering to tackle complex tasks.

1.5 About the Author

For over a decade, I’ve been immersed in the world of AI, driven by a passion to democratize access to this technology. My journey began with building a foundation in India called Olive Green, which today empowers over 600 students with access to primary education.

My professional journey has taken me to the forefront of the industry, working across Big Tech Companies, while my academic foundation was built at the London School of Economics. I’m committed to making AI accessible and understandable for everyone, and I invite you to join me on this exciting journey.

Connect with me on LinkedIn to stay informed about the latest AI advancements and be part of this movement towards an inclusive tech future.

2 Module 1: Fundamentals of LLMs and Prompting

2.1 Understanding Large Language Models (LLMs)

Let’s begin with a standard definition of LLMs:

What is a Large Language Model?

”A Large Language Model (LLM) is a type of artificial intelligence (AI) program that can recognize and generate text, among other tasks.”
LLMs are trained on massive datasets—hence the name ”large.” In simpler terms, an LLM has been fed enough examples to recognize and interpret human language or other complex data types. Many LLMs are trained on data, which can amount to thousands or even millions of gigabytes of text.

2.2 LLMs vs. Traditional AI Models

Traditional AI models, particularly predictive Machine Learning (ML) models, have long been the cornerstone of many AI applications. These models excel at identifying specific relationships between structured data and labels. For instance, a model might predict customer churn or categorize emails based on predefined features. These traditional models are highly effective for tasks with clear, structured inputs and outputs.

However, the emergence of LLMs has brought about a significant shift in how we approach AI. Unlike traditional models, LLMs are trained on vast amounts of unstructured content—ranging from books and articles to websites and code repositories. The sheer scale of the data they process enables them to generate entirely new content, whether it be text, images, or videos.

This creative capability—whether it’s drafting a report, composing an email, or answering complex questions—is where LLMs truly excel.

Predictive ML vs. Generative AI:
Traditional ML: Structured data + Labels → Predictions
LLMs: Unstructured content → Generated content (text, images, etc.)

Figure 1: Comparison of Traditional ML and Generative AI approaches

This doesn’t mean that every use case can be solved by LLMs; in fact, each use case may require either predictive models or generative AI, depending on the specific needs and objectives.

2.3 What is a Prompt?

Definition of a Prompt

"Prompt engineering is a relatively new discipline for developing and optimizing prompts to efficiently use LMs for a wide variety of applications and research topics." -Prompt Engineering Guide
Simply, a prompt is essentially a set of instructions you give to a Large Language Model (LLM). This input can be text, images, videos, or even a combination of these. The LLM then processes this input and generates a specific response or action as output, based on your instructions.

2.4 How Prompting Works

Here's how the process works:

1. **Input:** The prompt is your way of communicating with the LLM, providing a starting point that tells the model what you want it to accomplish. This input can be text, images, videos, or a combination of these, often accompanied by specific instructions within the prompt itself.
2. **Processing:** The LLM analyzes your prompt, drawing on its vast knowledge base and understanding of language patterns.
3. **Output:** The LLM generates a response based on your prompt. This response could be:
   - Text completion
   - A translation
   - A summary of information
   - A creative text piece (like a poem or script)
   - Even code generation

2.5 Fundamental Principles of Effective Prompting

Good prompts follow two fundamental principles: Clarity and Specificity

2.5.1 Clarity

Clarity involves using simple, unambiguous language that avoids jargon and overly complex vocabulary. Rather than keeping your queries short and snappy, take the time to make your point sufficiently clear to the LLM.

Examples of Clarity in Prompts

Unclear Prompt: "Who won the election?" *This is vague—I call this "Lazy Prompting." It doesn't specify which election, year, or country.*
Clear Prompt: "Who won the 2020 Presidential election in the United States?" *This is clear because it specifies the election, year, and location, making it easier for the LLM to provide an accurate response.*

2.5.2 Specificity

Specificity refers to providing enough context to guide the model's response. Sometimes, this can be akin to storytelling, giving the model enough detail to craft a meaningful answer.

Examples of Specificity in Prompts

Unspecific Prompt: "Give me a title for my autobiography." *This prompt is too broad. The model doesn't know what your autobiography is about, so it might generate titles that don't reflect your story.*
Specific Prompt: "I'm writing an autobiography about my journey from a small village in India to becoming a tech entrepreneur in Silicon Valley. The book focuses on challenges I faced, cultural transitions, and lessons learned. Please suggest 5 compelling titles that capture the essence of resilience and cross-cultural growth." *This prompt is much more specific. It provides context about the book's themes and content, which helps the model generate titles that are relevant and aligned with your story.*

Remember, the golden rule is simple: Garbage In = Garbage Out. The quality of your prompt directly affects the quality of the LLM's response.

2.6 Introduction to Google AI Studio

To practice your prompts, you can leverage Google AI Studio, a platform for experimenting with and deploying LLMs. It provides a user-friendly interface for crafting prompts, adjusting model parameters, and exploring different output formats.

2.6.1 Key Features of Google AI Studio

- **Playground:** This is where you can trial various inputs to generate desired outputs. Google AI Studio provides a streamlined interface, allowing you to easily explore the capabilities of different LLMs and refine your prompt-crafting skills.

- **System Instructions:** In the system instruction section, you can set the desired tone, style, and parameters to guide the LLM in understanding and responding to prompts effectively. This ensures that the output aligns with your specific needs and objectives.

- **Temperature Control:** This parameter controls the "randomness" of the AI's output:

- **Higher Temperature (e.g., 0.8):** Produces more creative and unexpected responses, suitable for brainstorming catchy taglines or playful descriptions.
- **Lower Temperature (e.g., 0.2):** Generates more predictable and focused outputs, ideal for straightforward product descriptions or clear call-to-actions.

- **Model Selection:** You can scroll through the available LLMs and select the model that best fits your use case. For instance, Gemini 1.5 Pro is a multimodal model capable of processing inputs like images, text, and videos to produce outputs, with a context capacity of 2 million tokens.

2.7 Module 1 Exercise: Clarity vs. Vague Prompts

Practicing Clear vs. Vague Prompts

Challenge Overview: Understanding the difference between vague and specific prompts is essential to effective prompt engineering. This challenge will help you see firsthand how the clarity and specificity of a prompt impact the quality of the output.

Instructions:

1. Open Google AI Studio or another LLM platform of your choice.
2. Run two prompts—one vague and one specific—and compare the outputs.
3. Analyze how specificity impacts the detail, relevance, and overall effectiveness of the content.
4. Reflect on how a clear, well-defined prompt can lead to a more polished and purpose-driven result.

Examples to Try:

- **Vague:** "Tell me about climate change."
- **Specific:** "Provide a 300-word summary of the latest IPCC report findings on climate change, focusing on projected impacts for coastal cities in the next 30 years. Include 3 key mitigation strategies recommended by scientists."

3 Module 2: Advanced Prompting Strategies

3.1 Three Fundamental Strategies of Prompt Engineering

In this module, we'll explore three key strategies that will elevate your interactions with LLMs:

1. Giving clear instructions

2. Using system instructions effectively
3. Simplifying complex prompts

3.2 Strategy 1: Give Clear Instructions

Instructions are like a blueprint for your LLM. The more detailed and precise the blueprint, the more accurate and impressive the final construction will be.

3.2.1 Why Clear Instructions Matter

- **LLMs aren't mind readers (yet!):** They rely heavily on your instructions to understand the task at hand.
- **Specificity reduces ambiguity:** Clear instructions leave no room for the LLM to misinterpret your request.
- **Well-defined outputs save time:** By specifying the format and level of detail, you get results that are ready to use.

Example of Clear Instructions

Less Effective Instruction: ”Summarize this article about climate change.”
More Effective Instruction: ”Summarize the key findings and implications of this article on climate change in three bullet points, focusing on technological solutions and policy recommendations.”
Why is the second instruction more effective?

- **Specific Action:** ”Summarize the key findings and implications...” clearly defines the task.
- **Desired Output:** ”...in three bullet points” specifies the desired format and level of detail.
- **Focus and Constraints:** By asking for ”key findings and implications,” with a focus on ”technological solutions and policy recommendations,” you guide the LLM to prioritize important information.

3.3 Strategy 2: Use System Instructions

System instructions act as a guiding principle for your LLM, influencing its behavior throughout the entire interaction. Think of them as setting the stage for a performance – they provide context and establish tone.

Copyright © 2024 Malika Malik

3.3.1 How to Use System Instructions Effectively

1. **Define the LLM's Role:**
    - **Be Specific:** Clearly state the LLM's role or persona (e.g., "You are a financial analyst," "You are a helpful assistant," "You are a creative storyteller").
    - **Highlight Expertise:** If relevant, mention the LLM's area of expertise (e.g., "Your expertise lies in analyzing market trends," "You are skilled at writing engaging dialogue").
2. **Outline the Task and Desired Output:**
    - **Provide Clear Instructions:** Break down the task into specific steps the LLM should follow.
    - **Specify Output Format:** Indicate how you want the information presented (e.g., bullet points, JSON, a summary paragraph).
    - **Set Expectations for Tone and Style:** Guide the LLM's writing style (e.g., "Use a formal tone," "Be concise and to the point," "Write in a conversational style").
3. **Establish Constraints and Limitations:**
    - **Define Boundaries:** If there are any topics the LLM should avoid or any specific rules it must follow, state them explicitly.
    - **Emphasize Objectivity (When Needed):** If you need an unbiased analysis, instruct the LLM to avoid expressing personal opinions or making subjective statements.
4. **Provide Background Context (If Relevant):** If there's any background knowledge or context that would help the LLM perform the task more effectively, include it in the system instruction.

Copyright © 2024 Malika Malik

Example of System Instructions

Coding Expert Persona
System Instructions: ”You are an expert Python programmer with 10 years of experience. You specialize in data visualization and machine learning applications. Your communication style is clear and concise, with a focus on best practices and efficient code. When providing code examples, you always include comments explaining the logic and any potential edge cases. If there are multiple approaches to solve a problem, you briefly mention alternatives but focus on the most efficient solution for the given context.”
Prompt: ”Create a function that takes a pandas DataFrame containing customer purchase history and returns a visualization showing the top 5 products by revenue over time.”
The LLM would respond with well-commented Python code for a visualization function, focusing on efficiency and best practices, and potentially mentioning alternative approaches.

3.4 Strategy 3: Simplifying Complex Prompts

Think of a complex recipe – tackling it all at once can be overwhelming. The same goes for LLMs. For complex tasks, breaking down your prompts into smaller subtasks can significantly enhance the LLM’s performance.

3.4.1 Two Key Strategies for Simplification

- **Chain Prompts:** Perfect for sequential tasks where each step builds upon the previous one.
- **Aggregate Responses:** Split a task into subtasks and run the subtasks in parallel.

Copyright © 2024 Malika Malik

Example of Breaking Down Complex Prompts

Imagine you manage a retail store and want to use a model to help you decide which products to prioritize for restocking based on customer feedback and your store's sales data.
Task 1a: Explore Product Sales Data "Analyze the product sales data in PRODUCT_SALES_DATA. Identify the top 10 products by unit sales in the last quarter. Format the results as a JSON object with product names as keys and units sold as values."
Task 1b: Explore Customer Feedback Data "Analyze the customer feedback in CUSTOMER_FEEDBACK_DATA. Calculate the average satisfaction score for each product mentioned in the feedback. Format the results as a JSON object with product names as keys and average satisfaction scores as values."
Task 2: Prioritize Products for Restocking "Based on the following sales data: [TASK_1A_OUTPUT] and customer satisfaction data: [TASK_1B_OUTPUT], create a prioritized list of 15 products for restocking. Give higher priority to products with high sales volume AND high customer satisfaction. Format the output as a numbered list with a brief explanation for each product's priority ranking."

3.4.2 Advantages of Breaking Down Complex Prompts

- **Enhanced Control:** You have granular control over each stage of the process.
- **Simplified Debugging:** It's easier to pinpoint and address errors in smaller, more focused prompts.
- **Increased Accuracy:** Dividing complex tasks often leads to more precise results from the LLMs.

3.5 Module 2 Exercise: System Instructions and Prompt Chaining

Exploring System Instructions and Prompt Chaining

Exercise 1: System Instructions
Compare these two approaches:
Basic Prompt (without system instructions): "Analyze the following financial data and suggest investment opportunities:
Financial Data
"

With System Instructions: System: "You are a conservative financial advisor with 25 years of experience. You prioritize long-term stability over short-term gains and always emphasize risk management in your analyses. Your communication style is clear and jargon-free, making complex financial concepts accessible to non-experts."
Prompt: "Analyze the following financial data and suggest investment opportunities:
Financial Data
"

Run both prompts and compare how the role-based system instruction guides the LLM's analysis, tone, and depth of response.
Exercise 2: Prompt Chaining
Compare these approaches:
Single Prompt: "Create a comprehensive marketing strategy for a new plant-based protein drink targeting fitness enthusiasts aged 25-40. Include target audience analysis, key messaging, distribution channels, and a social media plan."
Chained Prompts: 1. "Analyze the target market of fitness enthusiasts aged 25-40 for a plant-based protein drink. What are their key demographics, values, pain points, and purchasing behaviors?" 2. "Based on this target audience analysis: [Output from Prompt 1], develop key messaging and unique selling propositions for our plant-based protein drink." 3. "Using the target audience insights and messaging framework: [Summary of outputs from Prompts 1 and 2], recommend the top 3 distribution channels for our drink with justification for each." 4. "Create a 3-month social media content plan across Instagram and TikTok that incorporates our key messaging [from Prompt 2] and reaches our target audience [from Prompt 1]."
Compare the output from the single prompt to the more detailed and focused outputs from the chained prompts.

4 Module 3: Leveraging Examples in Prompts

4.1 The Power of Examples in Prompting

Why are examples so powerful? Imagine trying to explain a complex concept without any examples—it's like trying to teach someone to ride a bike just by talking about it. Examples are key for effective communication, especially with LLMs. They provide a clear reference point for the style, tone, and format you're aiming for.

Think of it like teaching someone to bake a cake. You wouldn't just say, "Bake a cake." You'd provide a recipe with specific ingredients, measurements, and instructions. Examples in prompts work the same way—they guide the LLM by setting clear expectations.

4.2 Zero-Shot, One-Shot, and Few-Shot Prompting

In the world of LLMs, there are different ways to provide examples, leading to various prompting techniques:

- **Zero-Shot Prompting:** No examples are provided. The LLM completes the task based solely on its training data. While it can be impressive when it works, this method can be unreliable for more complex tasks.
- **One-Shot Prompting:** A single example is given to guide the LLMs, similar to giving a quick demonstration before asking someone to try it themselves.
- **Few-Shot Prompting:** Multiple examples (usually 3-5) are provided to illustrate the desired outcome.

Copyright © 2024 Malika Malik

4.3 Example: Sentiment Analysis Using Different Prompting Techniques

Classifying Sentiment of Movie Reviews

Zero-Shot Prompt: "Classify the sentiment of the following movie review as positive, negative, or neutral: 'I couldn't believe how boring this movie was. The plot was predictable, and the acting was wooden. I kept checking my watch, waiting for it to end.'"

One-Shot Prompt: "Here's an example of sentiment classification: Review: 'This film was a masterpiece! The cinematography was breathtaking, and the performances were outstanding. I was completely immersed from start to finish.' Sentiment: Positive
Now, classify the sentiment of the following review as positive, negative, or neutral: 'I couldn't believe how boring this movie was. The plot was predictable, and the acting was wooden. I kept checking my watch, waiting for it to end.'"

Few-Shot Prompt: "Here are some examples of sentiment classification:
Example 1: Review: 'This film was a masterpiece! The cinematography was breathtaking, and the performances were outstanding. I was completely immersed from start to finish.' Sentiment: Positive
Example 2: Review: 'The movie was okay. Some parts were interesting, but others dragged on too long. The acting was decent, but the script was lacking.' Sentiment: Neutral
Example 3: Review: 'I hated every minute of this film. The characters were one-dimensional, the plot had massive holes, and the dialogue was cringe-worthy.' Sentiment: Negative
Now, classify the sentiment of the following review as positive, negative, or neutral: 'I couldn't believe how boring this movie was. The plot was predictable, and the acting was wooden. I kept checking my watch, waiting for it to end.'"

4.4 How Many Examples Are Enough?

The optimal number of examples depends on your specific use case. You can experiment with the number of examples included in your prompt to achieve the best results. Often, models can recognize patterns with just a few examples, but finding the optimal number may require some trial and error.

Be mindful, though—too many examples can lead to overfitting, where the model's responses become too narrowly focused on the provided examples, reducing their generalizability.

4.5 Key Takeaways for Providing Examples

- **Choose Relevant Examples:** Select examples closely aligned with the task you want the LLM to perform.

- **Showcase Diversity:** Use examples that demonstrate different aspects or variations of the desired output.
- **Keep it Concise:** Avoid overwhelming the LLM with too many or overly detailed examples.
- **Experiment:** Try different numbers of examples to see what works best for your specific task.

4.6 Real-World Applications of Few-Shot Learning

Few-shot learning is utilized in various real-world applications, including:

- **Sentiment Analysis:** Classifying customer sentiment in text (positive, negative, neutral) based on a few examples.
- **Language Translation:** Enhancing translation models by providing a few examples of specific phrases or idioms, helping the AI understand and correctly translate them in different contexts.

4.7 Module 3 Exercise: Utilizing Few-Shot Examples

Few-Shot Examples Challenge

Exercise Instructions:
Compare these two approaches for categorizing customer feedback:
Prompt without Examples: "Analyze this customer feedback and categorize the issues based on the following categories: UI/UX, Performance, Feature Request, Integration, Pricing, and Other. Additionally, rate the sentiment (Positive/Neutral/Negative) and priority (High/Medium/Low).
Here is the feedback: 'The app keeps crashing when I try to export data to Excel. This is really frustrating as I need to create reports for my manager every week. Please fix this ASAP!'"
Prompt with Examples: "Our customer support team is receiving a large volume of unstructured feedback. Your task is to analyze this feedback and categorize the issues for our product and engineering teams.
Use the following categories: UI/UX, Performance, Feature Request, Integration, Pricing, and Other. Additionally, rate the sentiment (Positive/Neutral/Negative) and priority (High/Medium/Low).
Here is an example: Input: 'The recent update has made the app slower, and I keep getting errors when trying to sync with my calendar. Please fix this quickly!' Category: Performance, Integration Sentiment: Negative Priority: High
Now, analyze this feedback: 'The app keeps crashing when I try to export data to Excel. This is really frustrating as I need to create reports for my manager every week. Please fix this ASAP!'"
Compare the outputs and note how the inclusion of an example improves the accuracy and consistency of the categorization.

5 Module 4: Advanced Techniques

5.1 Chain of Thought (CoT) Prompting

Chain of Thought (CoT) prompting, introduced by Wei et al. in 2022, offers a powerful way to help LLMs handle complex reasoning tasks. Instead of tackling problems all at once, this approach encourages the model to break down each problem into a series of manageable steps, just like we would when solving something step by step.

When you combine CoT prompting with few-shot prompting—where the model learns from a few examples—you can see a noticeable improvement in how accurately the model reasons through challenges. This technique is particularly useful for more complicated tasks that require the LLM to think things through before giving a response.

Chain of Thought Example

Few-Shot Chain of Thought Prompt: ”I’ll solve some logical reasoning problems step by step.
Problem 1: If Anna has 5 apples and gives 2 to Bob, then Bob gives 1 apple to Charlie, how many apples does Anna have left?
Step 1: Initially, Anna has 5 apples. Step 2: Anna gives 2 apples to Bob, so Anna now has 5 - 2 = 3 apples. Step 3: The transaction between Bob and Charlie doesn’t affect Anna’s apple count. Step 4: Therefore, Anna has 3 apples left.
Problem 2: If a shirt originally costs $25andisonsalefor20$
Step 1: The discount amount is 20Step 2: The final price is the original price minus the discount amount, which is $25-5=20.Step3:Therefore,thefinalpriceis20$.
Now, solve this problem: Roger has 12 marbles. He loses 5 marbles then buys twice as many as he lost. How many marbles does Roger have now?
Let me solve this step by step: Step 1: Initially, Roger has 12 marbles. Step 2: Roger loses 5 marbles, so he now has 12 - 5 = 7 marbles. Step 3: Roger then buys twice as many marbles as he lost. He lost 5 marbles, so he buys 2 × 5 = 10 marbles. Step 4: After buying the marbles, Roger has 7 + 10 = 17 marbles. Step 5: Therefore, Roger has 17 marbles now.”
Output: As you can see, by providing the LLM with step-by-step reasoning, it can effectively break down logical problems and arrive at accurate conclusions. This approach also allows you to reduce the number of examples needed to achieve the desired output, making it efficient and effective.

5.2 Zero-Shot Chain of Thought Prompting

Zero-Shot CoT prompting takes this concept a step further by enhancing the reasoning process with just a simple addition—”Let’s think step by step.”

Zero-Shot Chain of Thought Example

Prompt (Without CoT): ”What is the next number in the sequence: 2, 5, 10, 17, 26, ?”
Prompt (With Zero-Shot CoT): ”What is the next number in the sequence: 2, 5, 10, 17, 26, ? Let’s think step by step.”
Output: ”Let’s think step by step about this sequence: 2, 5, 10, 17, 26, ?
Step 1: Let’s look at the differences between consecutive numbers: 5 - 2 = 3 10 - 5 = 5 17 - 10 = 7 26 - 17 = 9
Step 2: I notice that the differences are increasing by 2 each time: 3, 5, 7, 9...
Step 3: Following this pattern, the next difference should be 9 + 2 = 11
Step 4: Therefore, the next number in the sequence should be 26 + 11 = 37
The next number in the sequence is 37.”

By adding ”Let’s think step by step,” the LLM is prompted to work through the sequence logically, resulting in a correct answer.

Copyright © 2024 Malika Malik

Zero-shot-CoT is most effective for tasks involving arithmetic, commonsense reasoning, and symbolic reasoning. However, it is not as effective as CoT prompting for more complex reasoning tasks. Hence, it's key to test these techniques based on your specific use case.

5.3 Prompt Chaining

Chain of Thought (CoT) prompting is fantastic for guiding a Large Language Model (LLM) through complex reasoning processes. But what if your task involves multiple distinct steps, each requiring in-depth consideration?

This is where prompt chaining becomes invaluable. Prompt chaining involves breaking down a complex task into smaller subtasks and using the output of one LLM call as the input for the next. This creates a chain of operations capable of handling sophisticated queries with precision.

5.3.1 Why Use Prompt Chaining?

Prompt chaining is useful because it allows each subtask to receive the LLM's full attention, significantly reducing the likelihood of errors. By breaking down complex tasks into simpler, more manageable subtasks, instructions become clearer, and outputs more precise.

Additionally, this approach makes it easier to trace and address any issues that may arise within the chain of prompts, ensuring a more efficient and effective workflow. The clarity it brings to each step not only improves the final output but also helps in troubleshooting and refining the process along the way.

5.3.2 When to Use Prompt Chaining?

Prompt chaining is particularly effective for tasks such as research synthesis, document analysis, or iterative content creation. If your task involves multiple transformations, citations, or detailed instructions, chaining helps ensure that no step is overlooked or mishandled.

Pro Tip

If a specific step is not performed correctly or is missed, isolate that step in its own prompt. This allows you to refine the troublesome part without needing to redo the entire process. This strategy saves time and helps you focus on perfecting one element at a time.

5.3.3 How to Chain Prompts

1. **Identify Subtasks:** Break your task down into distinct, sequential steps. Each subtask should be designed to accomplish a specific part of the overall goal.
2. **Structure with Tags for Clear Handoffs:** Use tags (like XML) to clearly pass outputs from one prompt to the next. This ensures that each prompt knows exactly what information to use from the previous one.

3. **Single-Task Focus:** Ensure each subtask has a clear, singular objective. Avoid overloading a single prompt with too many tasks; focus on one goal at a time to maintain clarity.

4. **Iterate:** Continuously refine each subtask based on the LLM's performance. Don't be afraid to tweak prompts and test different variations until you get the desired result.

Prompt Chaining Example: Cloud Migration Strategy

Prompt 1: Evaluate the Cloud Migration Strategy "You are a cloud architecture expert with 15 years of experience in enterprise migrations. Please evaluate the following cloud migration strategy for a financial services company. Focus on three key areas: scalability, security, and cost management. Provide a detailed analysis of strengths and weaknesses in each area.
Here is the migration strategy: [Detailed migration strategy description]"
Prompt 2: Draft the Migration Strategy Report "Based on the following expert analysis of a cloud migration strategy, draft a comprehensive report for the executive team. The report should include an executive summary, detailed findings, and actionable recommendations. Make sure the report is clear, concise, and focused on business outcomes rather than technical details.
Expert Analysis: [Output from Prompt 1]"
Prompt 3: Grade the Migration Report "You are the CTO of a financial services company evaluating a cloud migration report. Please grade the following report on a scale of 1-10 for three criteria: clarity, practicality, and alignment with enterprise objectives. Provide specific feedback on how the report could be improved in each area.
Company Objectives: - Reduce infrastructure costs by 30- Improve system reliability to 99.99- Enhance data security to meet new regulatory requirements - Scale customer-facing applications to support 50
Migration Report: [Output from Prompt 2]"

Prompt chaining can be adapted to various scenarios beyond cloud strategy evaluation. For example, you can use it for creating comprehensive research papers, developing marketing strategies, or even planning complex projects. The key is to break down the task into logical steps, refine each prompt, and ensure that each part of the chain is optimized for clarity and accuracy.

Remember, every LLM behaves slightly differently, so what works best in one scenario might need adjustment in another.

5.4 Module 4 Exercise: Zero-Shot Chain of Thought Prompting

Zero-Shot Chain of Thought Exercise

Exercise Instructions:
Compare these two approaches to a financial advising scenario:
Prompt without Zero-Shot CoT: ”You’re a financial advisor. A client wants to invest $10,000. They can choose between two options: A) A stock that historically returns 12% annually but is volatile, or B) A bond that guarantees 6% annually. The client needs the money in 5 years for a down payment on a house. Which option do you recommend?”
Prompt with Zero-Shot CoT: ”You’re a financial advisor. A client wants to invest $10,000. They can choose between two options: A) A stock that historically returns 12% annually but is volatile, or B) A bond that guarantees 6% annually. The client needs the money in 5 years for a down payment on a house. Let’s think through this step by step.”
Run both prompts and observe:

- Does the LLM provide a more detailed and logical explanation when asked to think step by step?
- How does the reasoning process differ between the two approaches?
- Is the final recommendation more nuanced or better justified in the CoT version?

6 Module 5: Automatic Prompt Engineering

6.1 Introduction to Automatic Prompt Engineering (APE)

Prompt engineering is an essential skill, yet creating prompts manually often requires multiple iterations, testing, and fine-tuning—a process that can be both time-consuming and labor-intensive.

Automatic Prompt Engineering (APE) addresses these challenges by automating the creation and optimization of prompts, significantly reducing the need for constant manual adjustments.

What is Automatic Prompt Engineering?

APE is an approach where the LLM is given the desired input and output, and the prompt is generated from these examples. This approach reduces the human effort involved in creating and validating prompts.

6.2 How Automatic Prompt Engineering Works

This technique was developed by researchers Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, and their team at the University of Toronto, the Vector Institute, and the University of Waterloo. They've introduced a novel method for generating prompts for large language models.

For those interested in exploring this in more depth, the paper ”Large Language Models Are Human-Level Prompt Engineers” provides comprehensive insights.

6.2.1 The Two-Model Approach

APE uses two large language models to automate the creation of prompts:

1. **Prompt Generator:** This LLM model is responsible for creating the prompts. The researchers experimented with different types of models for this role, such as GPT-3 and InstructGPT, which complete given inputs, and models like T5 and GLM, which fill in blanks.

2. **Content Generator:** Once the prompt is generated, this second model—InstructGPT in this case—produces the output based on that prompt.

6.2.2 The APE Process

1. The researchers start by giving the prompt generator a structured prompt. For example, they might input something like: ”I gave a friend an instruction and five inputs. The friend read the instruction and wrote an output for each input. Here are the input-output pairs.”

2. This is followed by a few example inputs and outputs, such as comparing the sizes of two animals. The prompt generator might then come up with something like, ”Choose the animal that is bigger.”

3. Next, they feed this generated prompt, along with 50 example inputs, to the content generator. The content generator produces outputs based on these prompts.

4. The effectiveness of each prompt is judged by how often the content generator's outputs match the expected results.

5. Then, the researchers refine the prompts by asking the generator to produce variations of the highest-scoring prompt. They repeat this refinement step three times to get the best possible prompt. For instance, the initial prompt ”Choose the animal that is bigger” might evolve into ”Identify which animal is larger.”

6.3 Results and Implications

The results of this technique were impressive. In 19 out of 24 tasks from a dataset called Instruction Induction, APE-generated prompts performed better than those created by previous methods or even human-engineered prompts. For example, in a test called TruthfulQA, prompts generated by APE produced true and informative answers 40% of the time, compared to 30% for prompts created by humans.

APE even discovered a better zero-shot CoT prompt than the human-engineered ”Let’s think step by step” prompt. The prompt ”Let’s work this out in a step by step way to be sure we have the right answer” elicits chain-of-thought reasoning and improves performance on the MultiArith and GSM8K benchmarks.

The Automatic Prompt Engineering (APE) technique will enhance the efficiency of creating effective prompts for LLMs. By automating this process, APE streamlines workflows and saves time. However, it’s crucial to continuously evaluate across different techniques for your specific use case.

6.4 Course Summary: Key Strategies for Effective Prompt Engineering

As we conclude this course, let’s revisit the key strategies we’ve covered to help you become a more effective prompt engineer:

1. **Be Clear and Direct:** Always ensure your prompts are straightforward and specific to guide the LLM in delivering precise responses.
2. **Use Examples (Multi-Shot):** Providing multiple examples in your prompts helps the LLM better understand the context and produce more accurate results.
3. **Encourage Step-by-Step Reasoning (Chain of Thought):** For tasks that require complex reasoning, guide the LLM to think through problems one step at a time.
4. **Utilize Tags for Structured Prompts:** When dealing with complex prompts, use tags (like XML) to clearly structure and pass information between different parts of the task.
5. **Assign a Role (System Prompts):** Giving the LLM a specific role or perspective helps it frame responses in a more relevant and focused manner.
6. **Prefill the LLM’s Response:** Where possible, prefill parts of the response to set the tone or structure, helping the LLM align with your expectations.
7. **Chain Complex Prompts:** Break down complicated tasks into smaller, manageable steps, and chain these prompts together to ensure accuracy and thoroughness.

Important Note

Every LLM behaves differently, and results can vary based on the model you’re using. The key is to keep practicing and refining your prompts. With continued experimentation, you’ll develop a deeper understanding of how to best interact with different LLMs, allowing you to unlock their full potential.

7 Additional Resources

7.1 Recommended Reading

- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.
- Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners.
- Zhou, Y., Muresanu, A.I., Han, Z., et al. (2022). Large Language Models Are Human-Level Prompt Engineers.
- ”Prompt Engineering Guide” - Available online

7.2 Online Resources

- Google AI Studio - `https://aistudio.google.com/`
- Anthropic Claude Playground
- OpenAI GPT Playground
- Prompt Engineering Community Forums

8 Glossary of Terms

Automatic Prompt Engineering (APE) An approach that automates the creation and optimization of prompts using LLMs themselves.

Chain of Thought (CoT) A prompting technique that guides the LLM through step-by-step reasoning to solve complex problems.

Few-Shot Prompting A technique where multiple examples are provided to guide the LLM’s understanding and output generation.

Large Language Model (LLM) A type of AI program trained on massive datasets to recognize and generate text and other content types.

One-Shot Prompting A technique where a single example is provided to guide the LLM.

Prompt Chaining Breaking down complex tasks into smaller subtasks and passing outputs between them.

System Instructions Directions that establish the LLM’s role, tone, and behavior throughout an interaction.

Zero-Shot Prompting A technique where no examples are provided, and the LLM relies solely on its training.

Zero-Shot Chain of Thought A technique that enhances reasoning by instructing the LLM to think step by step without example reasoning chains.

Thank you for completing The Subtle Art of Prompting!

By Malika Malik

www.malikamalik.com

Copyright © 2024 Malika Malik