

EXP 2: Comparative Analysis of Naïve Prompting versus Basic Prompting Using ChatGPT

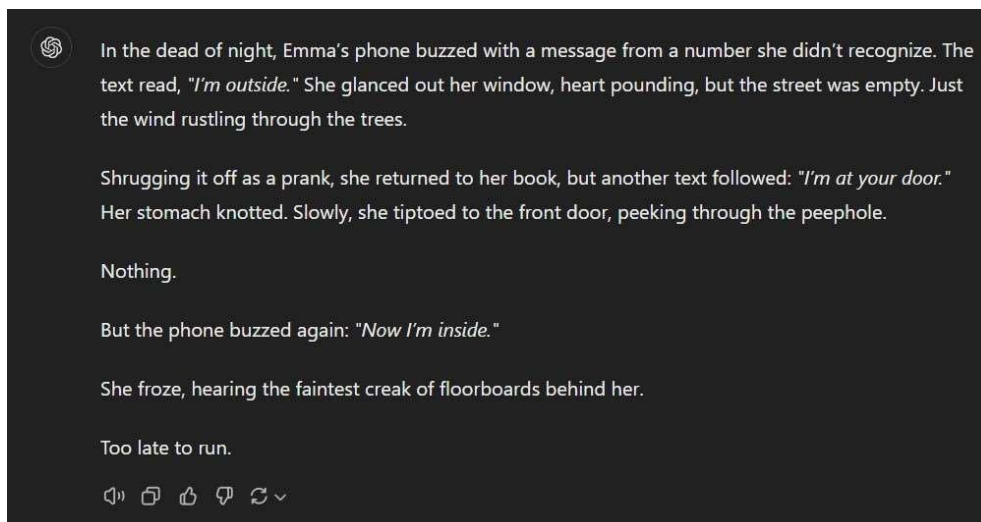
Across Various Test Scenarios

Aim:

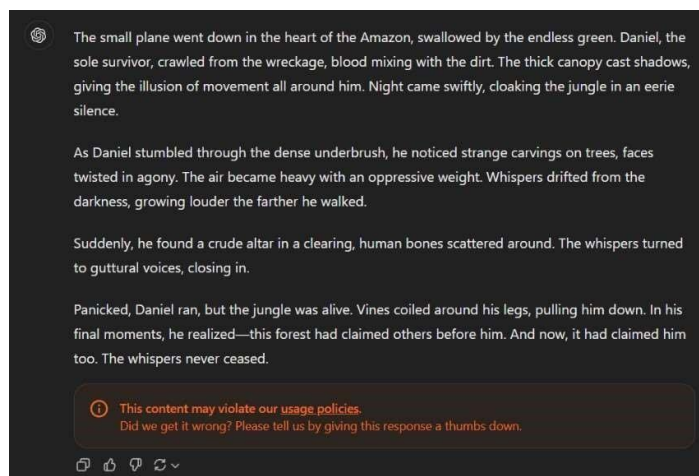
To test how ChatGPT responds to naïve prompts (broad or unstructured) versus basic prompts (clearer and more refined) across multiple scenarios, analyzing the quality, accuracy, and depth of the generated responses.

Scenario 1 : Testing Story Writing Skills Naïve

Prompt : “Write a short horror story.” Response :

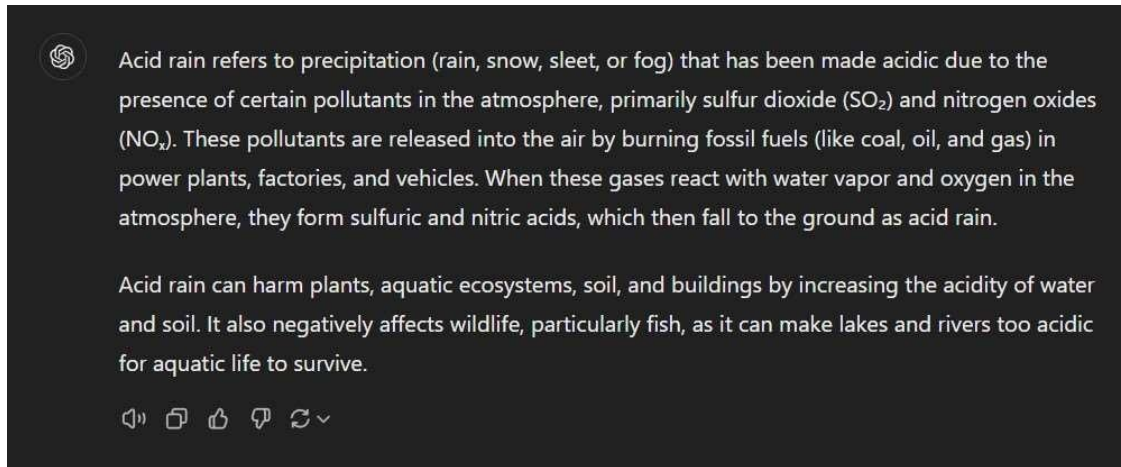


Basic Prompt : “Write a short story about an explorer who is lost in the Amazon forest due to aeroplane crash, the mood should be 2000’s kind of story in John R. Leonetti style. Follow horror genre. Restrict the story with 150 words” Response :



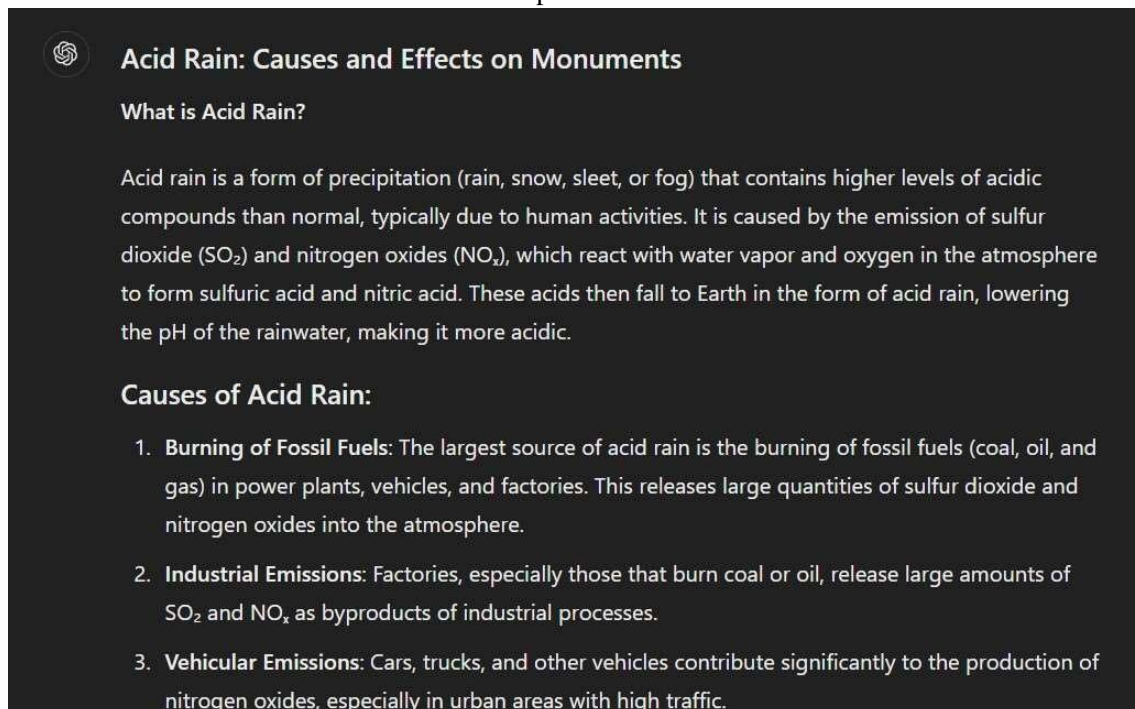
Scenario 2 : Answering Factual Research based Questions Naïve

Prompting : “What is Acid rain?” Response:



Basic Prompting : “Explain about climate change, including it’s causes and effects on the environment. List the responsibilities as we humans should take to prevent extreme effects of climate change.”

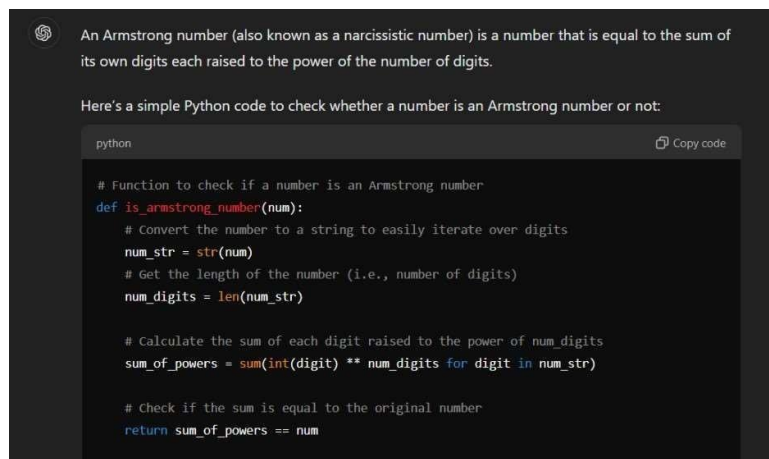
Response :



Scenario 3 : Coding Based Questions

Naïve Prompting : “Provide a code for finding Armstrong number.” Response

:

A screenshot of a ChatGPT interface. At the top, the OpenAI logo is on the left, and the text "An Armstrong number (also known as a narcissistic number) is a number that is equal to the sum of its own digits each raised to the power of the number of digits." is on the right. Below this, it says "Here's a simple Python code to check whether a number is an Armstrong number or not:". Then, there is a code block with a "python" label and a "Copy code" button. The code defines a function `is_armstrong_number(num)` that converts the number to a string, calculates the number of digits, and then sums each digit raised to the power of the number of digits. Finally, it checks if the sum equals the original number.

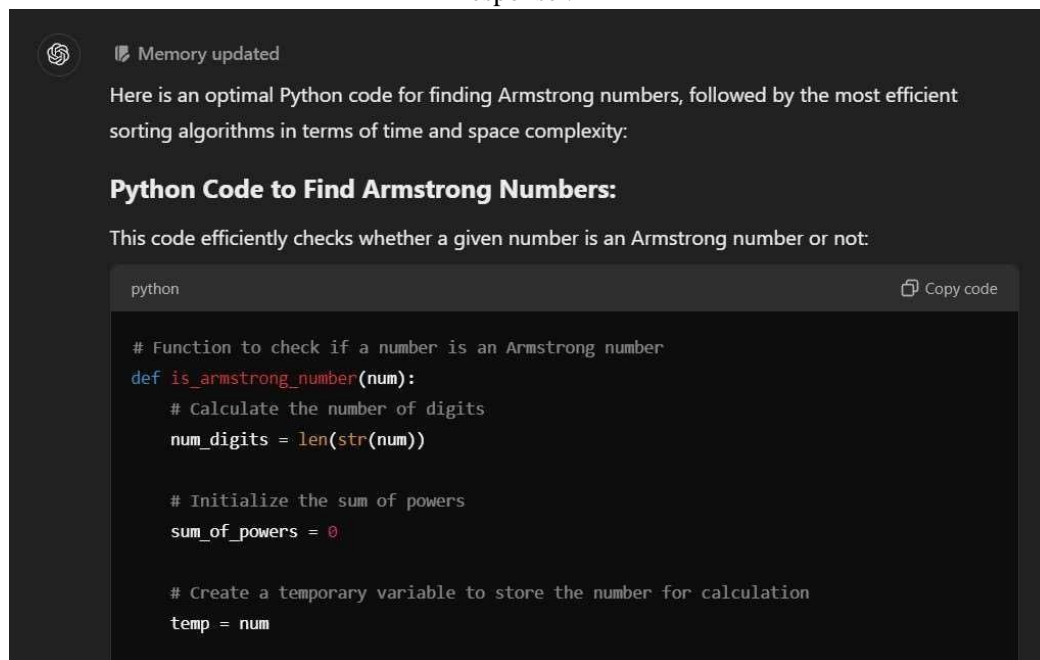
```
python
# Function to check if a number is an Armstrong number
def is_armstrong_number(num):
    # Convert the number to a string to easily iterate over digits
    num_str = str(num)
    # Get the length of the number (i.e., number of digits)
    num_digits = len(num_str)

    # Calculate the sum of each digit raised to the power of num_digits
    sum_of_powers = sum(int(digit) ** num_digits for digit in num_str)

    # Check if the sum is equal to the original number
    return sum_of_powers == num
```

Basic Prompting : “Can you give me a Python language code for finding Armstrong number? Give me the most optimal algorithms to sort with minimal time complexity and space complexity.”

Response :

A screenshot of a ChatGPT interface. At the top, the OpenAI logo is on the left, and the text "Memory updated" is on the right. Below this, it says "Here is an optimal Python code for finding Armstrong numbers, followed by the most efficient sorting algorithms in terms of time and space complexity:". Then, there is a section titled "Python Code to Find Armstrong Numbers:". Below this, it says "This code efficiently checks whether a given number is an Armstrong number or not:". Then, there is a code block with a "python" label and a "Copy code" button. The code defines a function `is_armstrong_number(num)` that calculates the number of digits, initializes the sum of powers to 0, and then iterates through the digits to calculate the sum. Finally, it checks if the sum equals the original number.

```
python
# Function to check if a number is an Armstrong number
def is_armstrong_number(num):
    # Calculate the number of digits
    num_digits = len(str(num))

    # Initialize the sum of powers
    sum_of_powers = 0

    # Create a temporary variable to store the number for calculation
    temp = num
```

Here's the structured report comparing ChatGPT's responses to the naive and basic prompts from the specified scenarios:

Comparison of ChatGPT's Responses to Naïve vs. Basic Prompts

Scenario Prompt Type Example Prompt Response Quality Accuracy Depth Notes

Table Comparing ChatGPT’s Responses to Naïve and Basic Prompts Across Scenarios

Naïve Prompt "Write a short horror story." Basic Prompt "Write a short story about an explorer who is lost in the Amazon forest due to an aeroplane crash..."

Clarity of Story	Vague, open-ended, less specific.	Focused, detailed, and follows clear structure and theme.
Depth of Plot	Simple and general, lacks rich context or backstory.	Richer plotline with clear context (explorer, Amazon, plane crash) and defined mood/style.
Setting	Ambiguous, could be anywhere.	Clear, set in the Amazon rainforest, with specific environmental challenges (wilderness, jungle).
Character Development	Minimal, focuses on a single character reaction.	More detailed—defined character (explorer) with backstory (plane crash).
Emotional Impact	Basic, relies on common horror tropes.	More immersive due to defined mood (2000s horror, John R. Leone style), creating tension and suspense.
Tone	Generic horror tone.	Tailored tone, matching the requested style (John R. Leone), offering a sense of dread and isolation.
Pacing	Fast-paced, jumps straight to a climax or twist.	Balanced, gradual buildup with clearer progression from crash to the explorer's realization of danger.
Creativity	General and predictable plot.	More creative—mixes the setting, genre, and style to deliver a unique experience (explorer vs. jungle).
Style Consistency	Lacks specific stylistic guidance.	Consistently follows the specified style (2000s horror, John R. Leone), creating a specific mood.
Length Control	No explicit word count, can vary.	Meets word count restriction (150 words), delivering a concise but effective narrative.

Naïve Prompt: "What is Acid rain?" Basic Prompt: "Explain about climate change, including its causes and effects on the environment. List the responsibilities we humans should take to prevent extreme effects of climate change."

Depth of Response	Brief and surface-level explanation of what acid rain is.	In-depth explanation of climate change with clear structure, covering causes, effects, and responsibilities to mitigate it.
Clarity	Clear, concise, but lacks detail and broader context.	Clear, detailed, and well-structured. Divides the answer into sections, making the information easier to understand.
Accuracy	Accurate definition but missing technical depth or complexity.	Comprehensive and accurate with both basic causes and advanced effects. Addresses multiple aspects of climate change (environment, human actions, etc.).
Comprehensiveness	Focuses solely on defining acid rain without much elaboration.	Broad coverage, addressing causes, environmental effects, and actionable human responsibilities. Provides a holistic view of the issue.
Use of Examples	No specific examples provided.	Examples of human activities that cause climate change (burning fossil fuels, deforestation) and effects (global warming, rising sea levels) are included.
Actionable Advice	No actionable advice.	Provides clear actions humans should take (reduce emissions, conserve energy, use renewable energy, etc.) to mitigate climate change effects.
Engagement	Provides a factual answer without further engagement.	Engages the reader by discussing responsibilities and suggesting actions, making the response more interactive and informative.
Complexity of Language	Simple and easy to understand. Suitable for a general audience.	More complex and detailed, but still accessible. Suitable for readers who want a deeper understanding and actionable insight into climate change.
Length	Short and to the point (~60 words).	Longer, divided into multiple sections (causes, effects, responsibilities), making it around ~200 words or more.

Contextual Depth	Focuses strictly on acid rain with little connection to other issues.	Connects climate change to broader environmental and societal contexts, touching on policy and human responsibility.
------------------	---	--

Aspect Naïve Prompt ("Provide a code for finding Armstrong number") Basic Prompt ("Can you give me a Python language code for finding Armstrong number? Give me the most optimal algorithms to sort with minimal time complexity and space complexity.")

Clarity of Prompt	Low – The request is simple but vague, lacking additional details or specificity.	High – The request is clear, with specific instructions, two distinct tasks, and details that guide the response (Python code + optimal algorithms).
Length of Response	Short – Provides just the Armstrong number code.	Lengthy – Detailed response with Armstrong number code, sorting algorithms, and explanations of time and space complexity.
Accuracy	Accurate – Provides a working code for Armstrong number.	Accurate – Provides correct Python code and includes correct information on optimal sorting algorithms.
Depth of Explanation	Minimal – Code is provided with basic explanation.	High – The response includes code, detailed explanations of optimal sorting algorithms, and complexities, offering more insight into the topic.
Code Quality	Simple and functional Armstrong number code.	High-quality Python code, plus a well-structured explanation of sorting algorithms, with examples of Merge Sort, Quick Sort, and Timsort.
Additional Context	Lacks context – Only fulfills the code request without offering more.	Provides broader context – Discusses the importance of sorting algorithms, time complexity, and space considerations in a structured response.
Handling Multiple Tasks	Only addresses one task (Armstrong number code).	Addresses both tasks – Provides Python code and offers a thorough comparison of sorting algorithms.

User Effort Post-Response	High – The user may need to follow up with clarifications or additional requests.	Low – All necessary details are covered, and the user can use the response directly without further clarifications.
------------------------------	---	---

Analysis of Prompt Clarity Impacting Output Quality

Quality:

1. **Naïve Prompts:** Responses to vague prompts, such as “Write a horror story,” often lack direction and depth. The model may generate a basic narrative without a defined theme or structure. Similarly, “What is acid rain?” and “Can you give me a code for finding armstrong number.” yield general or incomplete responses due to their lack of specificity.
2. **Basic Prompts:** Clear and detailed prompts like “Write a short story about an explorer who is lost in the Amazon forest due to an aeroplane crash...” and “Explain about climate change, including its causes and effects on the environment...” lead to higher-quality responses. The model can leverage the additional context to produce more informative and engaging outputs.

Accuracy:

3. **Naïve Prompts:** Prompts like “What is acid rain?” may result in moderate accuracy, providing a general overview but lacking the depth necessary for a complete understanding of the topic. The coding prompt does not specify the programming language or requirements, which may lead to ambiguity in the response.
4. **Basic Prompts:** More focused prompts, such as “Provide a code for finding Armstrong number” ensure higher accuracy in responses. They help the model hone in on specific information, leading to better-informed answers.

Depth:

5. **Naïve Prompts:** Responses to open-ended or vague prompts tend to be shallow, as seen in the naive responses to storytelling and coding questions. The AI may provide only surface-level information without exploring critical details or nuances.
6. **Basic Prompts:** By specifying requirements and context, such as mood, genre, and coding language, users elicit richer, more detailed responses. For instance, the request for a story about an astronaut not only defines the narrative but also sets a specific mood and style, enhancing depth.

Conclusion

The quality, precision, and depth of ChatGPT's outputs in a variety of contexts are strongly impacted by the prompts' clarity. While fundamental prompts that give clear instructions produce high-quality, accurate, and detailed information, naïve prompts frequently result in responses that are ambiguous, generic, or superficial. Users can improve their interactions with ChatGPT and increase the efficacy of its responses by creating clear prompts that specify specific needs. This analysis emphasizes how

crucial prompt engineering is to making efficient use of AI models like ChatGPT, emphasizing that precision and lucidity are necessary to get peak performance.