

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.

Instructions:

1. Define the Two Prompt Types:

- **Naïve Prompts:** Broad, vague, or open-ended prompts with little specificity.
- **Basic Prompts:** Clear, detailed, and structured prompts that give specific instructions or context to guide the model.

2. Prepare Multiple Test Scenarios:

- Select various scenarios such as:
 - Generating a creative story.
 - Answering a factual question.
 - Summarizing an article or concept.
 - Providing advice or recommendations.
 - Or Any other test scenario
- For each scenario, create both a naïve and a basic prompt. Ensure each pair of prompts targets the same task but with different levels of structure.

3. Run Experiments with ChatGPT:

- Input the naïve prompt for each scenario and record the generated response.
- Then input the corresponding basic prompt and capture that response.
- Repeat this process for all selected scenarios to gather a full set of results.

4. Evaluate Responses :

Compare how ChatGPT performs when given naïve versus basic prompts and analyze the output based on Quality, Accuracy and Depth. Also analyse does ChatGPT consistently provide better results with basic prompts? Are there scenarios where naïve prompts work equally well?

Deliverables:

- A table comparing ChatGPT's responses to naïve and basic prompts across all scenarios.
- Analysis of how prompt clarity impacts the quality, accuracy, and depth of ChatGPT's outputs.
- Summary of findings with insights on structuring prompts for optimal results when using ChatGPT.

The definitions for the two prompt types:

Naïve Prompts

Naïve prompts are characterized by their broad, vague, or open-ended nature, providing minimal specificity. They may lack clarity and structure, making it difficult for the model to understand the intended context or focus. Examples include questions like "Tell me about technology" or "What is your opinion on books?"

Basic Prompts

Basic prompts are clear, detailed, and structured, offering specific instructions or context to guide the model effectively. They are designed to elicit precise responses and often include relevant details or parameters. Examples include prompts like "Explain the benefits of renewable energy sources in detail" or "Summarize the main themes of 'Pride and Prejudice' in three paragraphs."

Few-Shot Pattern:

In Few-Shot Prompting, you put examples of what you want the GenAI to do in your prompt¹²³. These examples are also called shots⁴, which is how the technique gets its name.

Few-shot prompting builds upon the idea that large language models can learn new information from their prompt⁵. By providing a few demonstrations of the desired task or output format within the prompt itself, you guide the model to produce more accurate and consistently formatted responses.

Few-shot prompting sits on a spectrum of "shot" techniques:

1. Zero-shot prompting: No examples are provided
2. One-shot prompting: A single example is given
3. Few-shot prompting: Multiple examples (typically 2+) are included

Basic Prompts:

Make it in proper order Software Engineer - ML Engineer needed at Tesla. 2+ years of experience required. Salary range \$90,000 - \$120,000. Remote work available. Apply by June 05, 2024.

Chat GPT-Output:

Job Title: Software Engineer - ML Engineer

Company: Tesla

Experience Required: 2+ years

Salary Range: \$90,000 - \$120,000

Work Arrangement: Remote work available

Application Deadline: June 05, 2024

Naïve Prompts

AI Engineer needed at Tesla. 2+ years of experience required. Salary range \$90,000 - \$120,000. Remote work available. Apply by June 05, 2024.

Chat GPT-Output:

Job Title: AI Engineer

Company: Tesla

Experience Required: 2+ years

Salary Range: \$90,000 - \$120,000

Work Arrangement: Remote work available

Application Deadline: June 05, 2024

Application Deadline: August 15, 2024 INPUT: Data Scientist wanted at Unlimited Innovations Ltd. PhD in Computer Science or related field. Minimum 5 years industry experience. Competitive salary €80,000 - €1,00,000 based on experience. On-site work in Munich office. Apply by August 15, 2024.

Chat GPT-Output:

Job Title: Data Scientist

Company: Unlimited Innovations Ltd.

Experience Required: Minimum 5 years in the industry

Education Required: PhD in Computer Science or related field

Salary Range: €80,000 - €100,000 based on experience

Work Arrangement: On-site work in Munich office

Application Deadline: August 15, 2024

Persona Pattern:

Basic Prompt:

Input - Develop an app

Output given:

To develop an app, you can follow these key steps:

1. Define the Purpose and Audience

- Identify the problem your app will solve.
- Define your target audience and their needs.

2. Research and Plan

- Analyze similar apps in the market.
- Determine your app's unique selling points (USPs).
- Create a detailed app plan, including features, user interface, and user experience.

3. Choose a Development Approach

- **Native App Development:** For iOS (Swift) or Android (Kotlin).
- **Cross-Platform Development:** Use frameworks like React Native or Flutter to build for both platforms simultaneously.

4. Design the App

- Create wireframes or prototypes using tools like Figma or Sketch.
- Design a user-friendly interface focusing on aesthetics and functionality.

5. Develop the App

- Set up your development environment.
- Code the front end (user interface) and back end (server, database).
- Ensure you follow best practices for coding and maintainability.

6. Test the App

- Conduct various testing methods, including unit tests, integration tests, and user acceptance testing (UAT).
- Fix any bugs or issues that arise during testing.

7. Launch the App

- Prepare for deployment by creating marketing materials.
- Publish the app on platforms like the App Store and Google Play.

8. Gather Feedback and Iterate

- Collect user feedback to identify areas for improvement.
- Release updates and new features based on user suggestions and changing needs.

9. Market the App

- Utilize social media, SEO, and content marketing to promote your app.
- Consider paid advertising for wider reach.

10. Monitor and Maintain

- Continuously monitor app performance and user engagement.
- Regularly update the app to fix bugs, add features, and improve performance.

Additional Considerations

- **Budget and Timeline:** Establish a budget and a realistic timeline for development.
- **Legal Considerations:** Ensure compliance with legal requirements, including privacy policies and app store guidelines.
- **Team Composition:** Depending on your expertise, consider collaborating with developers, designers, and marketers.

Naïve Prompts:

Input :

Consider you as an App developer and I am your client where you need to develop an app for me from scratch and you need to ask questions one by one about how the app should be designed with its difficulty level, obstacles, and other stuff.

Sure! Let's start the process of designing your app. I'll ask you questions step by step to gather all the necessary information.

1. What is the primary purpose of the app?

- What problem do you want the app to solve or what functionality do you want it to provide?

Input Given:

Project Management

2. Who is your target audience?

- Are you aiming for individual users, teams, small businesses, or large organizations?

Input Given:

For college Students

3. What specific features do you want in the app?

- Here are some common features to consider:
 - Task and project tracking
 - Collaboration tools (chat, file sharing)
 - Calendar integration
 - Deadline reminders
 - Note-taking
 - Templates for project plans
 - Progress tracking and analytics
- Are there any specific features you want to include or prioritize?

Input Given :

Add all features

4. What difficulty level do you want the app to have?

- Should it be user-friendly for beginners, or do you want to include advanced features that may require some familiarity with project management tools?

Input Given:

No, make it for beginners, don't add advanced features

5. What obstacles or challenges do you anticipate for users?

- Are there any specific pain points you want to address, such as managing multiple projects, staying organized, or collaborating with peers?

Input Given :

managing multiple projects, staying organized, or collaborating with peers

6. What platforms do you want the app to be available on?

- Should it be a mobile app (iOS, Android), a web app, or both?

Input Given :

Android

7. Do you have a specific design style or theme in mind for the app?

- For example, should it be minimalist, colorful, professional, or playful?

Input Given :

Minimalist with less colorful

8. What is your preferred timeline for developing this app?

- Do you have a specific deadline in mind for when you want the app to be launched?

Input Given :

2 months

9. What is your budget for this app development?

- Do you have a specific budget range in mind for the project?

Input Given :

2lakh Indian rupees

10. Do you need any specific integrations or technologies?

Input Given :

No

11. How do you plan to handle user feedback and updates after the app is launched?

- Do you have a process in mind for collecting user feedback and making updates?

Input Given :

Yes

12. Do you need a marketing plan for the app launch?

- Would you like assistance in creating a marketing strategy to promote the app once it's ready?

Input Given :

No

13. Finally, do you have a name in mind for the app, or would you like suggestions?

Input Given :

Trackify

Trackify is a great name for a project management app!

Summary of Your App Requirements:

- **Purpose:** Project management for college students
- **Target Audience:** College students
- **Features:**
 - Task and project tracking
 - Collaboration tools (chat, file sharing)
 - Calendar integration
 - Deadline reminders
 - Note-taking
 - Templates for project plans
 - Progress tracking and analytics

- **Difficulty Level: Beginner-friendly**
- **Obstacles Addressed: Managing multiple projects, staying organized, and collaborating with peers**
- **Platform: Android**
- **Design Style: Minimalist with subtle colors**
- **Timeline: 2 months for development**
- **Budget: 2 lakh Indian Rupees**
- **Integrations: None**
- **User Feedback: You have a process in mind**
- **App Name: Trackify**

Next Steps

If you're ready, we can move forward with the planning phase, including wireframing and development strategies. Let me know how you'd like to proceed!

Context Pattern:

Basic Prompt:

Input - Hardware used in mobiles

Output From Chat-GPT:

Mobile phones consist of several key hardware components that enable their functionality. Here are some of the most important ones:

1. **System on Chip (SoC):** Combines the CPU, GPU, RAM, and other essential processing units into one chip. Common examples are Qualcomm's Snapdragon, Apple's A-series, and Samsung's Exynos.
2. **Central Processing Unit (CPU):** The brain of the phone, responsible for executing instructions and managing tasks. Modern mobile CPUs are multi-core for better performance and efficiency.
3. **Graphics Processing Unit (GPU):** Handles rendering images, animations, and videos, especially in games and graphics-intensive applications.

4. **Display:** The screen, typically an OLED or LCD panel. It shows the user interface, videos, and apps. Most modern phones use high-resolution displays like AMOLED.
5. **Memory (RAM):** Used by the system to run applications and manage active processes. Typical modern smartphones have between 4GB and 12GB of RAM.
6. **Storage:** Used for storing apps, media, and the operating system. Flash storage (UFS) is commonly used, with sizes ranging from 32GB to 512GB or more.
7. **Battery:** Powers the phone, usually a lithium-ion battery. Modern smartphones come with fast charging, wireless charging, and various battery optimization technologies.
8. **Camera System:** Includes multiple sensors for capturing images and videos. Common sensors are wide-angle, ultra-wide, and telephoto lenses, with megapixel counts ranging from 12MP to 108MP.
9. **Modem:** Facilitates wireless communication, such as 4G LTE, 5G, Wi-Fi, and Bluetooth.
10. **Sensors:** Various sensors like accelerometer, gyroscope, proximity sensor, ambient light sensor, and fingerprint scanner add functionality like orientation detection, screen brightness adjustment, and biometric security.
11. **Speakers and Microphones:** Used for audio output and input, enabling phone calls, music playback, and voice commands.
12. **Ports and Connectors:** Includes USB-C or Lightning ports for charging and data transfer, as well as SIM card slots, and possibly headphone jacks.

Naïve Prompts:

Input - Hardware is used in mobiles with its architecture, design, and biggest brand.

Output From Chat-GPT:

Mobile hardware consists of several critical components, each with its own architecture, design, and leading brands. Let's break them down:

1. System on Chip (SoC):

- **Architecture:** SoCs integrate the CPU, GPU, memory controller, modems, and other components on a single chip. They use **ARM architecture** because it's energy-efficient, crucial for mobile devices.
- **Design:** SoCs are designed to be compact and efficient, enabling multitasking, gaming, and communications to function together without overheating or draining battery.

- **Biggest Brands:**
 - **Qualcomm** (Snapdragon series)
 - **Apple** (A-series, M-series chips)
 - **Samsung** (Exynos series)
 - **MediaTek** (Dimensity series)

2. Central Processing Unit (CPU):

- **Architecture:** Most mobile CPUs are based on ARM architecture (e.g., ARM Cortex-A series), which prioritizes low power consumption. CPU cores are often arranged in **big.LITTLE architecture**, where high-performance cores handle intensive tasks, and power-efficient cores handle lighter tasks.
- **Design:** CPUs in mobile SoCs are designed with multi-core configurations (usually between 6 and 8 cores). They focus on optimizing both performance (for tasks like gaming) and efficiency (for everyday use).
- **Biggest Brands:**
 - **ARM** (Cortex-A series)
 - **Apple** (A14, A15 Bionic)
 - **Qualcomm** (Kryo cores in Snapdragon chips)

3. Graphics Processing Unit (GPU):

- **Architecture:** Mobile GPUs are designed to handle graphic rendering and AI tasks. They typically use **ARM's Mali**, **Qualcomm's Adreno**, or **Apple's proprietary GPU**. Their architecture ensures smooth video playback, gaming, and graphics-intensive applications.
- **Design:** GPUs in mobile SoCs are designed for real-time rendering with minimal power consumption, supporting high-resolution displays and augmented reality applications.
- **Biggest Brands:**
 - **ARM** (Mali series)
 - **Qualcomm** (Adreno series)
 - **Apple** (custom GPUs in A-series chips)
 - **Imagination Technologies** (PowerVR)

4. Memory (RAM):

- **Architecture:** Mobile RAM is usually based on **LPDDR (Low-Power DDR)** architecture. LPDDR is optimized for energy efficiency compared to traditional desktop RAM, while still delivering fast performance.
- **Design:** LPDDR RAM is designed in **dual-channel** or higher configurations for better data throughput, ensuring that the phone can handle multitasking without lag.
- **Biggest Brands:**
 - **Samsung**
 - **SK Hynix**
 - **Micron Technology**

5. Storage (Flash Memory):

- **Architecture:** Mobile devices use **NAND flash storage**, typically in the form of **Universal Flash Storage (UFS)**, which offers higher read/write speeds compared to older standards like eMMC.
- **Design:** UFS is designed to handle fast sequential read/write speeds and better power efficiency, enabling quicker app launches and smoother multitasking.
- **Biggest Brands:**
 - **Samsung**
 - **Western Digital**
 - **Kioxia** (formerly Toshiba Memory)

6. Display:

- **Architecture:** Mobile displays are typically **AMOLED** (Active Matrix Organic Light-Emitting Diode) or **LCD** (Liquid Crystal Display) technology. AMOLED is preferred for its deep blacks, vibrant colors, and energy efficiency.
- **Design:** AMOLED displays are flexible and can be designed to be curved or folded. Displays also incorporate high refresh rates (90Hz, 120Hz, etc.) for smoother scrolling and better gaming experiences.
- **Biggest Brands:**
 - **Samsung Display**
 - **LG Display**
 - **BOE Technology Group**

7. Camera System:

- **Architecture:** Mobile camera systems use **CMOS (Complementary Metal-Oxide-Semiconductor)** sensors. Modern cameras use multiple lenses and computational photography techniques to improve image quality.

- **Design:** Mobile cameras integrate several lenses (wide-angle, telephoto, macro) and advanced AI-driven algorithms to enhance photo quality, focus speed, and night mode performance.
- **Biggest Brands:**
 - **Sony** (IMX series)
 - **Samsung** (ISOCELL series)
 - **OmniVision**

8. Battery:

- **Architecture:** Mobile devices use **Lithium-ion (Li-ion)** or **Lithium Polymer (Li-Po)** batteries. These batteries have high energy density and a relatively low self-discharge rate, making them ideal for mobile use.
- **Design:** Batteries are designed to be thin and light while offering higher capacities (3000-5000 mAh is typical). Modern batteries support **fast charging** (via USB-PD or proprietary technologies) and **wireless charging**.
- **Biggest Brands:**
 - **Samsung SDI**
 - **LG Chem**
 - **Panasonic**

9. Modem (Cellular & Connectivity Chipset):

- **Architecture:** Mobile modems are based on technologies like **4G LTE**, **5G NR**, **Wi-Fi**, and **Bluetooth**. These modems follow standards defined by organizations like **3GPP** (for cellular).
- **Design:** Modern modems support dual-SIM, carrier aggregation, and advanced MIMO (multiple input, multiple output) techniques for faster data transfer.
- **Biggest Brands:**
 - **Qualcomm** (X series modems)
 - **Samsung**
 - **MediaTek**

10. Sensors:

- **Architecture:** Sensors like accelerometers, gyroscopes, proximity sensors, and ambient light sensors are based on MEMS (Micro-Electro-Mechanical Systems) architecture.
- **Design:** These sensors are designed to be ultra-compact and power-efficient, allowing them to continuously track motion, orientation, and environmental changes with minimal battery impact.
- **Biggest Brands:**
 - **Bosch Sensortec**
 - **STMicroelectronics**
 - **Invensense** (TDK)

11. Audio (Speakers & Microphones):

- **Architecture:** Mobile speakers use **dynamic drivers** or **balanced armature drivers**, while microphones typically use MEMS-based designs for better sound pickup and noise cancellation.
- **Design:** Mobile audio systems are designed to provide high-quality sound in small form factors. Dolby Atmos or similar technologies are often integrated for enhanced audio experience.
- **Biggest Brands:**
 - **AAC Technologies**
 - **Knowles Corporation**
 - **Goertek**

Conclusion:

Analysis of How Prompt Clarity Impacts Quality, Accuracy, and Depth

1. **Depth of Responses:**
 - Naïve prompts lead to broader, more generic responses. These responses tend to cover basic concepts without diving into specific details or niche applications.
 - Basic prompts provide clarity, guiding ChatGPT to focus on more relevant details and nuances. The responses tend to be more structured and comprehensive, offering in-depth insights on the subject.
2. **Accuracy and Relevance:**
 - Naïve prompts may produce accurate information but can lack specificity. For example, asking a vague question like "Tell me about SoC" may not yield

responses that align with the user's underlying intent (e.g., understanding SoC in the context of mobile hardware).

- Basic prompts improve accuracy and relevance by narrowing the focus. By specifying architecture, design, and major brands, for example, the responses contain precise information that aligns closely with the user's needs.

3. Contextual Understanding:

- Naïve prompts sometimes fail to provide enough context, leading to a more generalized response. For instance, "What's the weather like in Paris?" could return current or average weather without addressing specific timeframes.
- Basic prompts clarify the user's intentions and provide ChatGPT with context to generate responses that are more appropriate. Asking for "current weather and forecast" ensures the response is timely and focused.

4. Tailored Recommendations:

- Naïve prompts often return surface-level recommendations. For instance, a request to "Plan a trip to France" might result in suggestions of popular tourist destinations without accounting for timeframes or preferences.
- Basic prompts that specify preferences or time constraints (e.g., "10-day trip focusing on historical sites") yield tailored and highly actionable responses.

5. Technical Precision:

- Naïve prompts can lead to broad technical responses that may lack specific examples, explanations, or contrasting concepts (e.g., "Tell me about functions in Python").
- Basic prompts often yield technically precise explanations, comparing specific types (e.g., regular vs. lambda functions) with examples that directly address the user's query.

