You're making changes in a project you don't have write access to. Submitting a change will write it to a new branch in your fork **Sanjeevkarthick/Exno.4-Prompt-Engg**, so you can send a pull request.

Exno.4-Prompt-Engg / README.md in main

Cancel changes Commit changes...

Edit Preview

Show Diff

Exno.4-Scenario-Based Report Development Utilizing Diverse Prompting Techniques

DATE:

REGISTER NUMBER :

Aim: To design an AI-powered chatbot that assists customers in resolving issues related to product troubleshooting, order tracking, and general inquiries. The chatbot should handle various customer queries efficiently while maintaining a conversational and user-friendly tone. In this experiment, we will employ different prompt patterns to guide the development process of the chatbot, ranging from basic task-oriented prompts to more complex, persona-driven prompts.

Algorithm: 1. Direct Instruction Prompts

Objective: Guide the chatbot to respond concisely to customer inquiries. Prompt Pattern: Prompt: "When a customer asks for the status of their order, reply with: 'Your order is currently being processed and will be delivered by [date].'" 2. Contextual Prompting Objective: Incorporate specific context to provide detailed answers based on the user's previous interaction. Prompt Pattern: Prompt: "If the customer previously mentioned that they haven't received their order, say, 'I see that you mentioned your order hasn't arrived yet. Let me check the details for you and get back shortly.'" 3. Persona-Based Prompting Objective: Design the chatbot to adopt a specific persona, making the interaction more engaging. Prompt Pattern: Prompt: "Pretend you are a friendly, helpful customer service representative. Use a conversational tone, such as 'Hey there! I'm here to help with any questions you might have. Let's get your issue sorted!'" 4. Few-Shot Prompting Objective: Teach the AI how to respond using a few examples, enabling it to generalize for similar situations. Prompt Pattern: Prompt: "Here are some examples of how to handle technical questions: 'My phone isn't charging.' → 'Have you tried using a different cable? If that doesn't work, it may be an issue with the port.' 'The screen is flickering.' → 'It sounds like a display issue. Have you tried restarting the device?' Now, respond to: 'My app keeps crashing.'" 5. Chain of Thought Prompting Objective: Use a step-by-step reasoning approach for resolving more complex or technical issues. Prompt Pattern: Prompt: "When a customer reports their laptop overheating, guide them through the following steps: Ask if they are using the laptop on a soft surface. Suggest moving the laptop to a flat, hard surface for better airflow. Ask if they've cleaned the vents recently. Recommend restarting the device to see if the issue persists. Now, solve: 'My laptop fan is making a loud noise.'" 6. Instruction with Constraints Objective: Instruct the chatbot to provide assistance while adhering to specific constraints (e.g., response length or tone). Prompt Pattern: Prompt: "Respond to order inquiries in no more than 50 words and avoid using technical jargon. For example, 'Your order is on the way and should arrive by [date]. Feel free to reach out if you need anything else.'" 7. Reflective Prompting Objective: Ensure that the chatbot reflects the user's query back to them before providing a response, reducing misunderstandings. Prompt Pattern: Prompt: "When a customer asks for help, first reflect their question back to them. For example, if they ask 'How can I reset my password?' respond with 'You're asking how to reset your password, correct? Here's how you can do it.'" Result: The various types of Prompts are executed successfully.
1.Designing an AI-Powered Customer Support Chatbot Objective: To design an AI-powered chatbot that assists customers in resolving issues related to product troubleshooting, order tracking, and general inquiries. The chatbot should handle various customer queries efficiently while maintaining a conversational and user-friendly tone.

Approach: This experiment will use different prompt patterns to guide the development of the chatbot. The patterns will range from simple, task-oriented instructions to complex, persona-driven prompts to evaluate how each affects the chatbot's performance and customer experience.

Algorithm 1: Direct Instruction Prompts Objective: Guide the chatbot to respond concisely to customer inquiries.

Prompt Pattern: Prompt: When a customer asks for the status of their order, reply with: "Your order is currently being processed and will be delivered by [date]."

Use Case: This prompt is used to ensure quick, informative, and standardized responses for order tracking inquiries. **2. Contextual Prompting** Objective: Incorporate specific context to provide detailed answers based on the user's previous interaction.

Prompt Pattern: Prompt: If the customer previously mentioned that they haven't received their order, say: "I see that you mentioned your order hasn't arrived yet. Let me check the details for you and get back shortly."

Use Case: This prompt pattern allows the chatbot to reference prior messages, enhancing continuity and demonstrating awareness of the customer's ongoing issue. It creates a more human-like and empathetic interaction by acknowledging user concerns directly. **3. Persona-Based Prompting** Objective: Design the chatbot to adopt a specific persona, making the interaction more engaging and relatable for the user.

Prompt Pattern: Prompt: Pretend you are a friendly, helpful customer service representative. Use a conversational tone, such as: "Hey there! I'm here to help with any questions you might have. Let's get your issue sorted!"

Use Case: This pattern enables the chatbot to mirror a warm, human-like tone that builds rapport with the customer. By adopting a persona, the chatbot becomes more approachable, helping to ease user frustration and foster a positive service experience. **4. Few-Shot Prompting** Objective: Teach the AI how to respond using a few examples, enabling it to generalize and respond effectively to similar situations.

Prompt Pattern: Prompt: Here are some examples of how to handle technical questions:

"My phone isn't charging." → "Have you tried using a different cable? If that doesn't work, it may be an issue with the port."

"The screen is flickering." → "It sounds like a display issue. Have you tried restarting the device?"

Now, respond to: "My app keeps crashing."

Expected Output: "It might be a software glitch. Try force closing the app and reopening it. If the issue continues, reinstalling the app may help."

Use Case: Few-shot prompting allows the chatbot to learn through examples and adapt to a variety of similar queries. This method is especially effective in scenarios involving troubleshooting, where patterns in user problems are common and solutions are often reusable. **5. Chain of Thought Prompting** Objective: Use a step-by-step reasoning approach to help the chatbot resolve more complex or technical issues effectively.

Prompt Pattern: Prompt: When a customer reports their laptop overheating, guide them through the following steps:

Ask if they are using the laptop on a soft surface.

Suggest moving the laptop to a flat, hard surface for better airflow.

Ask if they've cleaned the vents recently.

Recommend restarting the device to see if the issue persists.

Now, solve: "My laptop fan is making a loud noise."

Expected Output: "A loud fan can be caused by overheating or dust buildup. Are you using your laptop on a soft surface like a bed or couch? If so, try placing it on a hard surface to improve ventilation. Also, have you cleaned the vents recently? Dust can build up and cause the fan to work harder. Finally, restarting your laptop can sometimes help reset the system if it's stuck in a high-performance mode."

Use Case: This prompting style helps the chatbot reason through problems in logical steps, especially for technical issues that benefit from guided troubleshooting. It mirrors the process a human support agent might follow, improving problem resolution accuracy. **6.Instruction with Constraints:** The chatbot will follow the constraint of responding to order inquiries in no more than 50 words, avoiding the use of technical jargon. The prompt example will ensure that the chatbot is brief, clear, and conversational.

Example Prompt: "Respond to order inquiries in no more than 50 words and avoid using technical jargon. For example, 'Your order is on the way and should arrive by [date]. Feel free to reach out if you need anything else.'"

Expected Outcomes: Efficiency: The chatbot will be able to provide concise responses without overwhelming the user with unnecessary details.

Clarity: The chatbot's language will be simple, avoiding jargon to make the conversation easy to understand for users of varying technical backgrounds.

User-Friendly Tone: The chatbot's tone will be friendly, approachable, and supportive, ensuring customers feel comfortable engaging with it.

Future Experimentation: Testing variations of this prompt for different types of inquiries (e.g., product troubleshooting, returns, and refunds).

Evaluating the chatbot's response based on user feedback to ensure the prompt is leading to effective interactions. **7.Reflective Prompting** Objective: The objective is to design the chatbot to reflect the user's query back to them before providing a solution. This ensures that the chatbot has understood the query correctly, reducing the chance of misunderstandings and improving the accuracy of responses.

Prompt Pattern: "When a customer asks for help, first reflect their question back to them. For example, if they ask 'How can I reset my password?' respond with 'You're asking how to reset your password, correct? Here's how you can do it.'"

Example Prompt Implementation:

Customer's Query: "How do I track my order?"

Chatbot Response: "You're asking how to track your order, correct? Here's how you can do it: [Tracking Instructions]."

Customer's Query: "Can you help me with a refund?"

Chatbot Response: "You're asking about a refund, is that right? Here's the process for requesting a refund: [Refund Instructions]."

Result: The prompt ensures that the chatbot acknowledges the customer's inquiry by reflecting it back. This helps the user feel heard and clarifies their request before proceeding with the assistance. The chatbot's responses are thus more likely to meet the user's needs effectively.

Outcome: Improved Clarity: Reflecting the query helps prevent misinterpretation of the user's question.

Increased User Satisfaction: By confirming the question before providing an answer, the chatbot demonstrates attentiveness to the user's needs, leading to a more positive interaction.

Error Reduction: By addressing the exact query first, the chatbot reduces the risk of offering irrelevant or incorrect answers.

Future Experimentation: Testing the effectiveness of reflective prompting in complex or multi-step inquiries.

Analyzing user feedback to measure if this reflective approach leads to higher engagement and fewer follow-up questions.

Result: Thus the Prompts were exected succcessfully .