Name : Naikwadi Yash Shivdas

AI&DS II Experiment 04

AIM: To build an adaptive and contextual cognitive-based customer service application in domains like Insurance / Healthcare / Smarter Cities / Government etc.

THEORY:

In this experiment, the focus is on building a customer service system that is both adaptive and context-aware, meaning the system can remember and use interaction history to provide personalized, relevant, and dynamic responses. Unlike static chatbot models, such an adaptive system improves user experience by understanding the flow of conversation and the user's ongoing needs.

Adaptive Systems

- Adaptiveness refers to the system's capability to adjust its responses or behavior based on prior interactions.
- The system learns user preferences, pending requests, or repeated queries, and uses this information to tailor its replies.
- For example, if a user repeatedly asks about their order status or payments, the system can proactively suggest related options like reordering or payment assistance.

Contextual Understanding

- Context-awareness enables the system to maintain conversation state and recognize intentions over multiple turns.
- It keeps track of previous queries and ongoing tasks, so the responses remain coherent and meaningful within the dialogue.
- For instance, if the user mentions a complaint, the system can recognize follow-up queries about that complaint and offer assistance accordingly.

Components of the System

- A knowledge base that holds customer-related data such as orders, payments, complaints, and preferences.
- A context manager that maintains the last user query and any pending actions, allowing the assistant to remember and refer to earlier parts of the conversation.
- An adaptive suggestion engine that generates proactive prompts based on user queries and interaction patterns to facilitate smoother customer support.
- A response generation module that processes user input, applies context and adaptiveness, and generates relevant, personalized replies.

Benefits and Significance

- Enhances customer satisfaction through personalized and timely responses.
- Reduces repetitive or irrelevant replies by understanding user intent in context.
- Supports complex dialogues over multiple interactions, making digital assistants more effective and human-like.
- Applicable beyond food delivery to sectors like insurance, healthcare, smart cities, and government services where ongoing adaptive communication is vital.

Summary

This experiment extends cognitive customer service applications by emphasizing learning from and adapting to user interactions in real-time, creating a more intelligent assistant that understands user context and dynamically adjusts answers to meet customer needs effectively. This progressive approach marks a critical step toward truly conversational AI systems.

CODE:

Knowledge Base Initialization

```python
# Knowledge base for food delivery
food_delivery_data = {
    "orders": [],
    "payments": [],
    "complaints": [],
    "favorite_cuisines": [],
    "pending_payments": 500  # example amount
}
```

Context Manager Setup

```python
# Context manager to keep last query and pending actions
context = {
    "last_query": None,
    "pending_action": None
}
```

Adaptive Suggestions Engine

```python
def adaptive_suggestions(user_query):
    if "order" in user_query:
           return "Would you like me to reorder your last order or check its status?"
    elif "payment" in user_query or "pay" in user_query:
        return "Need help with payment options or refunds?"
    elif "complaint" in user_query:
        return "I can assist with lodging your complaint. Should I proceed?"
    return None
```

Assistant Response Function

```python
def food_delivery_assistant(user_query):
    user_query = user_query.lower()
    response = ""

    if "order" in user_query:
         response = "Your last order is being prepared. Estimated delivery is 30 minutes."
        # Could update orders etc.

    elif "menu" in user_query:
           response = "Today’s specials: Butter Chicken, Paneer Tikka, and Veg
Biryani."

    elif "payment" in user_query or "pay" in user_query:
                               response    =    f"Your    pending    payment    is
₹{food_delivery_data['pending_payments']}. Would you like to proceed with the
payment now?"

    elif "complaint" in user_query:
        # Better extraction of complaint details
         complaint_text = user_query.replace("complaint", "").replace("i have",
"").strip()
        if complaint_text:
            food_delivery_data["complaints"].append(complaint_text)
               response = f"Complaint noted: '{complaint_text}'. Our team will
address it promptly."
        else:
            response = "Please provide the details of your complaint."
         suggestion = "I can assist with lodging your complaint formally. Shall
I help you with that?"
        context["last_query"] = user_query
        context["pending_action"] = "complaint_formalize"
        response += f"\nSuggestion: {suggestion}"
        return response

    else:
         response = "Sorry, I didn't get that. You can ask about orders, menu,
payments, or complaints."

    # Save context
    context["last_query"] = user_query
    context["pending_action"] = None

    # Add adaptive suggestion
    suggestion = adaptive_suggestions(user_query)
    if suggestion:
        response += f"\nSuggestion: {suggestion}"

    return response
```

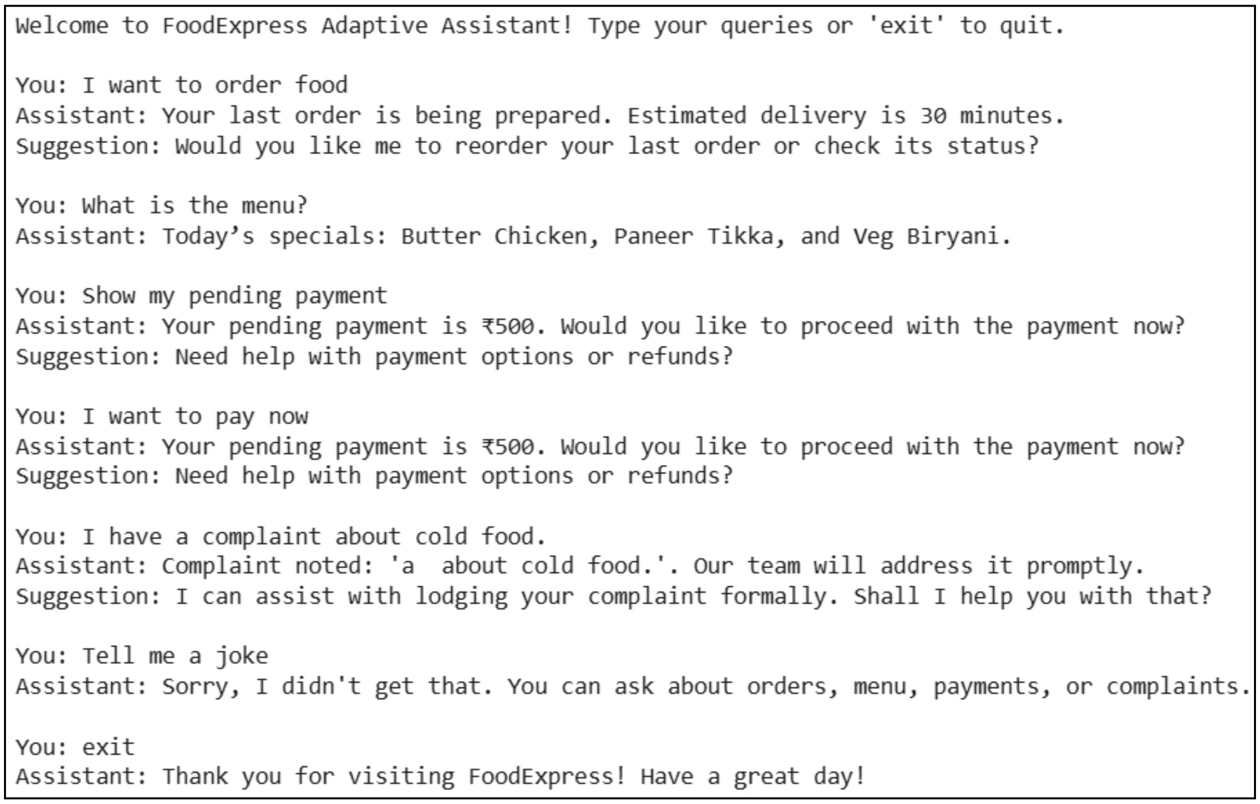
Interactive Session Execution

```
# Example interactive session
def run_food_delivery_assistant():
      print("Welcome to FoodExpress Adaptive Assistant! Type your queries or
'exit' to quit.")
    while True:
        user_text = input("\nYou: ").strip()
        if user_text.lower() == "exit":
            print("Assistant: Thank you for visiting FoodExpress! Have a great
day!")
            break
        reply = food_delivery_assistant(user_text)
        print("Assistant:", reply)
```

Run

```
run_food_delivery_assistant()
```

<u>OUTPUT:</u>



```
Welcome to FoodExpress Adaptive Assistant! Type your queries or 'exit' to quit.

You: I want to order food
Assistant: Your last order is being prepared. Estimated delivery is 30 minutes.
Suggestion: Would you like me to reorder your last order or check its status?

You: What is the menu?
Assistant: Today’s specials: Butter Chicken, Paneer Tikka, and Veg Biryani.

You: Show my pending payment
Assistant: Your pending payment is ₹500. Would you like to proceed with the payment now?
Suggestion: Need help with payment options or refunds?

You: I want to pay now
Assistant: Your pending payment is ₹500. Would you like to proceed with the payment now?
Suggestion: Need help with payment options or refunds?

You: I have a complaint about cold food.
Assistant: Complaint noted: 'a  about cold food.'. Our team will address it promptly.
Suggestion: I can assist with lodging your complaint formally. Shall I help you with that?

You: Tell me a joke
Assistant: Sorry, I didn't get that. You can ask about orders, menu, payments, or complaints.

You: exit
Assistant: Thank you for visiting FoodExpress! Have a great day!
```

<u>CONCLUSION:</u>

This experiment successfully demonstrated how an adaptive and contextual cognitive system can enhance customer service by remembering user interactions and providing personalized, relevant responses. By maintaining conversational context and proactively suggesting helpful actions, the system offers a more intuitive and effective user experience. Such adaptive intelligence is essential in creating smarter and more human-like digital assistants for service applications.

<u>39_colab_04</u>