import spacy

import random

from datetime import datetime

# Load English language model for NLP

nlp = spacy.load("en\_core\_web\_sm")

# Sample order database (in a real app, this would connect to a real database)

order\_db = {

"100001": {

"customer\_name": "John Smith",

"product": "Wireless Headphones",

"status": "Processing",

"order\_date": "2023-05-01",

"amount": 99.99,

"shipping\_address": "123 Main St, Anytown, USA"

},

"100002": {

"customer\_name": "Sarah Johnson",

"product": "Smart Watch",

"status": "Shipped",

"order\_date": "2023-04-28",

"amount": 199.99,

"shipping\_address": "456 Oak Ave, Somewhere, USA"

},

"100003": {

"customer\_name": "Michael Brown",

"product": "Laptop Backpack",

"status": "Delivered",

"order\_date": "2023-04-25",

"amount": 49.99,

"shipping\_address": "789 Pine Rd, Nowhere, USA"

}

}

class OrderSupportChatbot:

def \_init\_(self):

self.current\_order = None

self.greetings = [

"Hello! How can I assist you with your order today?",

"Hi there! Do you need help with an order cancellation?",

"Welcome to our order support. How may I help you?"

]

self.fallback\_responses = [

"I'm not sure I understand. Could you rephrase your request about your order?",

"I'm here to help with order cancellations. Could you provide your order number?",

"Could you clarify what you need help with regarding your order?"

]

# Template responses from the dataset

self.cancellation\_responses = [

"I understand you want to cancel order {order\_number}. To proceed, please visit your account and go to the 'Order History' section.",

"I can help you cancel order {order\_number}. Please sign in to your account and look for the 'Cancel Order' option.",

"To cancel order {order\_number}, please access your order details and select the cancellation option."

]

self.financial\_issue\_responses = [

"I'm sorry to hear you're having financial difficulties with order {order\_number}. Let me help you cancel it.",

"I understand you can no longer afford order {order\_number}. Here's how to cancel it...",

"For order {order\_number} that you can't afford, we can assist with cancellation."

]

self.problem\_responses = [

"I see you're having trouble canceling order {order\_number}. Let me guide you through the process.",

"For issues canceling order {order\_number}, please try these steps...",

"Having problems with order {order\_number} cancellation? Here's what to do."

]

def greet(self):

return random.choice(self.greetings)

def extract\_order\_number(self, text):

# Simple pattern matching for order numbers (6 digits)

doc = nlp(text)

for token in doc:

if token.like\_num and len(token.text) >= 5: # Assuming order numbers are at least 5 digits

return token.text

return None

def process\_input(self, user\_input):

user\_input = user\_input.lower()

doc = nlp(user\_input)

# Extract order number if present

order\_number = self.extract\_order\_number(user\_input)

if order\_number:

self.current\_order = order\_number if order\_number in order\_db else None

# Intent recognition

intents = {

"cancel\_order": any(token.text in ["cancel", "cancellation", "terminate"] for token in doc),

"financial\_issue": any(token.text in ["afford", "can't pay", "financial"] for token in doc),

"problem": any(token.text in ["problem", "issue", "trouble", "difficulty"] for token in doc),

"status": any(token.text in ["status", "where", "track"] for token in doc),

"greeting": any(token.text in ["hello", "hi", "hey"] for token in doc),

"goodbye": any(token.text in ["bye", "goodbye", "exit"] for token in doc),

"help": any(token.text in ["help", "support", "assistance"] for token in doc)

}

# Handle identified intents

if intents["greeting"]:

return random.choice(self.greetings)

elif intents["goodbye"]:

self.current\_order = None

return "Thank you for contacting us. Have a great day!"

elif intents["help"]:

return self.show\_help\_menu()

elif intents["status"]:

return self.handle\_order\_status()

elif intents["cancel\_order"]:

return self.handle\_cancel\_order(financial\_issue=intents["financial\_issue"], problem=intents["problem"])

else:

return random.choice(self.fallback\_responses)

def handle\_order\_status(self):

if not self.current\_order:

return "To check your order status, please provide your order number."

if self.current\_order not in order\_db:

return f"I couldn't find order {self.current\_order}. Please verify the order number."

order = order\_db[self.current\_order]

return (f"Order {self.current\_order} status: {order['status']}\n"

f"Product: {order['product']}\n"

f"Order date: {order['order\_date']}\n"

f"Amount: ${order['amount']:.2f}\n"

f"Shipping to: {order['shipping\_address']}")

def handle\_cancel\_order(self, financial\_issue=False, problem=False):

if not self.current\_order:

return "To cancel an order, please provide your order number."

if self.current\_order not in order\_db:

return f"I couldn't find order {self.current\_order}. Please verify the order number."

order = order\_db[self.current\_order]

if order['status'] == "Delivered":

return (f"Order {self.current\_order} has already been delivered. "

"If you wish to return it, please contact our returns department.")

elif order['status'] == "Shipped":

return (f"Order {self.current\_order} has already been shipped. "

"You may refuse delivery or contact us about returns after receiving it.")

# Select appropriate response based on context

if financial\_issue:

response = random.choice(self.financial\_issue\_responses).format(order\_number=self.current\_order)

elif problem:

response = random.choice(self.problem\_responses).format(order\_number=self.current\_order)

else:

response = random.choice(self.cancellation\_responses).format(order\_number=self.current\_order)

# Add cancellation steps

steps = [

"\nHere's how to cancel:",

"1. Log in to your account",

"2. Go to 'Order History'",

f"3. Find order {self.current\_order}",

"4. Click 'Cancel Order'",

"5. Confirm cancellation"

]

return response + "\n" + "\n".join(steps)

def show\_help\_menu(self):

help\_options = [

"Here's what I can help with:",

"• Cancel an order (provide order number)",

"• Check order status",

"• Help with financial difficulties for an order",

"• Troubleshoot order cancellation problems",

"Just let me know what you need help with!"

]

return "\n".join(help\_options)

# Main interaction loop

def main():

chatbot = OrderSupportChatbot()

print(chatbot.greet())

print("(Type 'exit' or 'bye' to end the conversation)")

while True:

user\_input = input("You: ").strip()

if user\_input.lower() in ['exit', 'quit', 'bye']:

print("Bot:", chatbot.process\_input(user\_input))

break

response = chatbot.process\_input(user\_input)

print("Bot:", response)

if \_name\_ == "\_main\_":

main()