Human-AI Interaction

(COMP3074)

AN INTERACTIVE NLP-BASED AI SYSTEM
Chatbot for Contact Management

Name: Dhruv Bhattacharjee
UID: 20592268

INTRODUCTION

Overview- In the rapidly evolving landscape of artificial intelligence, chatbots have emerged as versatile tools that facilitate interactive and natural language-based communication between users and machines. This project introduces a chatbot designed to streamline the process of managing contacts in a conversational manner. Aimed at enhancing user experience and accessibility, the chatbot incorporates natural language processing (NLP) method to analyse input from user, recognize intents, and generate responses which are relevant contextually.

Purpose- The primary purpose of the chatbot is to assist users in saving, retrieving, modifying, and organizing contact information seamlessly through conversation. By harnessing the power of machine learning and NLP, the chatbot aims to create a user-friendly interface for contact management, eliminating the need for traditional forms and commands. This innovative approach aligns with the growing demand for intuitive and interactive systems that can cater to a diverse user base.

The following are few of the Key Features of the chatbot:

Contact Management- Users can effortlessly save, display, and modify contact information using natural language commands.

Identity Management- The chatbot is designed to recognize and remember user names, creating a personalized and engaging experience.

Conversational Principles- the whole Implementation of conversational design principles ensures clarity, context awareness, and user engagement during interactions.

Target Audience- The chatbot is tailored for individuals who seek a modern and conversational approach to managing their contacts. Whether it's saving a new contact, reviewing the existing ones, or making modifications, the chatbot is crafted to appeal to users seeking a seamless and interactive contact management experience.

Technical Stack- The chatbot is implemented in Python, leveraging the power of neural networks for intent recognition and response generation. The system incorporates NLP techniques, eliminating the need for rigid command-based interactions. The user interface is designed to be intuitive and accessible, providing a natural conversational flow.

In the following sections, we delve into the technical architecture, conversational design principles, and the evaluation of the chatbot's usability and performance. The project encapsulates an exploration into the fusion of machine learning and conversational interfaces, with the ultimate goal of delivering a user-centric and efficient contact management solution.

CHATBOT ARCHITECTURE

Functionality:

I. Contact Management Module:

The core functionality of the chatbot revolves around its ability to manage contact information in a conversational manner. Users can save new contacts, retrieve existing ones, and modify contact details seamlessly through natural language commands. The system is designed to recognize specific intents related to contact management, ensuring a user-friendly experience.

II. Identity Management Module:

An integral part of the architecture is the identity management module. The chatbot is capable of recognizing and remembering user names, creating a personalized experience for each user. This feature adds a layer of engagement and context awareness to the conversations, contributing to an enriched user experience.

III. Conversational Principles Integration:

To enhance the user-bot interaction, the chatbot incorporates conversational design principles. These principles include clarity in responses, context awareness to maintain conversation flow, and user engagement strategies. By integration of the principles, the chatbot tries to provide an intuitive and natural language-based interface for users to interact with the system.

Implementation:

I. Natural Language Processing (NLP):

The model employs NLP methods to understand and interpret user input. Tokenization and lemmatization are used to break down sentences into meaningful units and degrade words to the base form, respectively. This pre-processing helps to increase the chatbot's ability to recognize patterns and intents in user queries.

II. Neural Network Model:

Intent recognition is facilitated by a neural network model trained on a dataset of user queries and corresponding intents. The model is implemented using the Keras library in Python, offering a robust framework for building and training neural networks. The architecture of the model consists of layers for processing the input, hidden layers for feature extraction, and an output layer for intent classification.

III. Contact Storage and Modification:

The architecture includes a mechanism for storing and retrieving contact information. Contacts are stored in a dictionary format, associating names with corresponding numbers. The system supports modifications such as changing contact names or updating contact numbers, providing a comprehensive contact management solution.

Justification:

I. User-Centric Approach:

The choice of implementing a neural network model for intent recognition aligns with the goal of creating a user-centric chatbot. The model allows for the dynamic understanding of user input, enabling the chatbot to adapt to a variety of conversational styles and language nuances.

II. Flexibility and Expandability:

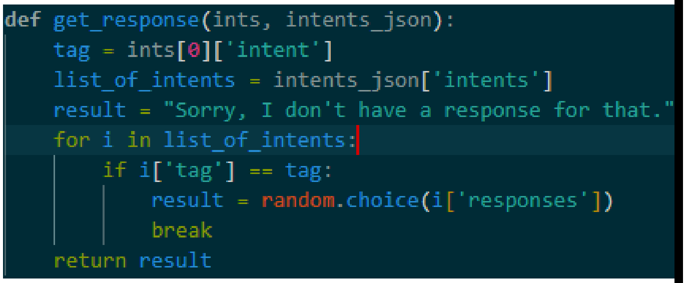
The modular design of the chatbot architecture provides flexibility for future expansions and enhancements. New features and intents can be seamlessly integrated into the existing structure, allowing the chatbot to evolve based on user needs and feedback.

III. Conversational Flow:

By incorporating conversational design principles, the architecture aims to create a smooth and engaging flow of conversation. This not only improves the overall user experience but also ensures that the chatbot comprehensively addresses user queries and intents.

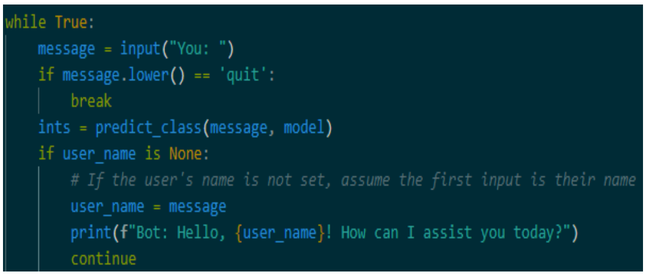
CONVERSATIONAL DESIGN

Clarity and Simplicity- The chatbot is designed with a focus on providing clear and concise responses. Responses are crafted to be easily understandable, avoiding unnecessary complexity. This ensures that users receive information in a straightforward manner, contributing to a positive user experience. In the code where all functions are defined there is a function named ‘get_response( )’. This function is called when the chatbot is supposed to asses the inout from the user and give out a response which is accurate to the user’s expected outcome.



```python
def get_response(ints, intents_json):
    tag = ints[0]['intent']
    list_of_intents = intents_json['intents']
    result = "Sorry, I don't have a response for that."
    for i in list_of_intents:
        if i['tag'] == tag:
            result = random.choice(i['responses'])
            break
    return result
```

Context Awareness-The chatbot maintains context awareness during conversations, allowing it to understand the user's intent within the context of the ongoing interaction. This is achieved through intent recognition and the utilization of context variables such as user names and contact details.

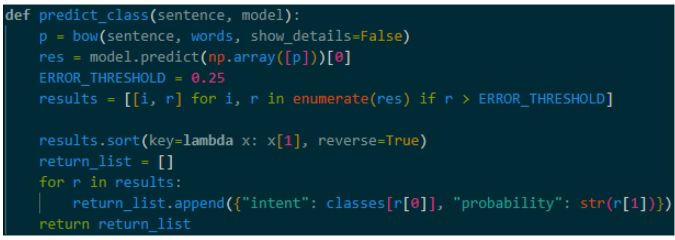


```python
while True:
    message = input("You: ")
    if message.lower() == 'quit':
        break
    ints = predict_class(message, model)
    if user_name is None:
        # If the user's name is not set, assume the first input is their name
        user_name = message
        print(f"Bot: Hello, {user_name}! How can I assist you today?")
        continue
```

For example, here the code’s usage is to get in the input from the user. The first input will be considered as the name of the user. Here, the process also starts predicting classes to analyse its outputs.

User Engagement- Engagement is fostered through personalized interactions and dynamic responses. The chatbot greets users by name, creating a more engaging and personalized experience. Additionally, the use of randomization in response selection adds an element of unpredictability, making interactions more interesting for users.

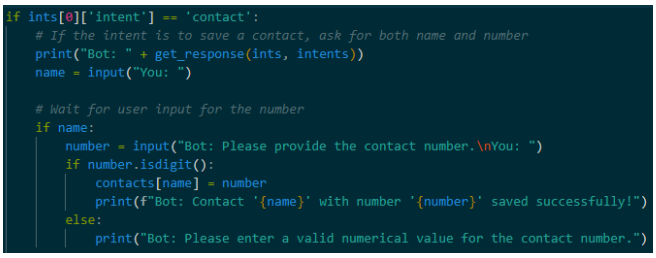
Dynamic Intent Recognition- The chatbot's neural network model enables dynamic intent recognition, allowing the system to adapt to a wide range of user queries and intents. This adaptability ensures that the chatbot remains effective in understanding diverse conversational patterns.



```python
def predict_class(sentence, model):
    p = bow(sentence, words, show_details=False)
    res = model.predict(np.array([p]))[0]
    ERROR_THRESHOLD = 0.25
    results = [[i, r] for i, r in enumerate(res) if r > ERROR_THRESHOLD]

    results.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in results:
        return_list.append({"intent": classes[r[0]], "probability": str(r[1])})
    return return_list
```

Here, the function predict_class( ) is used to predict the intents from the user inputs and understanding the intentions using probability. A specific threshold is also being used to manage outcomes of probability and matching.

User Feedback Integration- The chatbot provides feedback to users, acknowledging their input and guiding them through the conversation. For instance, when saving a contact, the bot confirms the successful save, enhancing the user's confidence in the system.



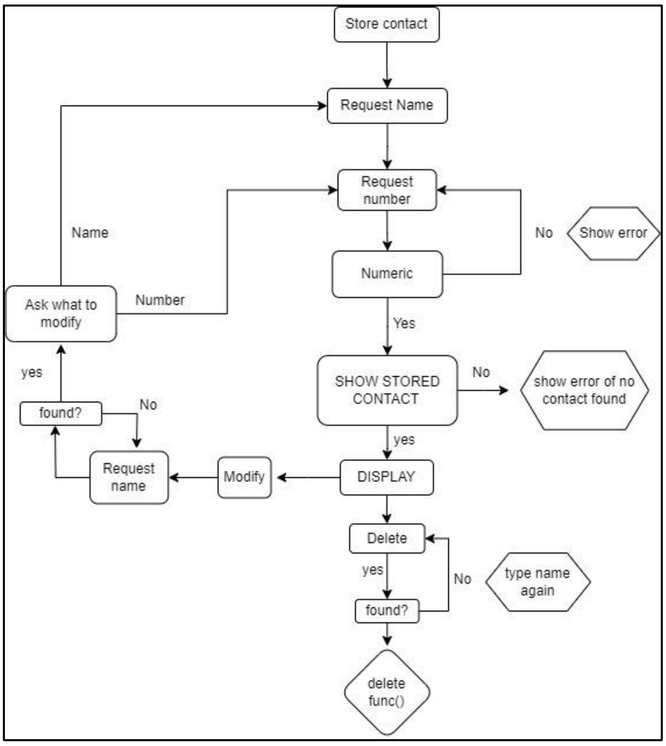
```python
if ints[0]['intent'] == 'contact':
    # If the intent is to save a contact, ask for both name and number
    print("Bot: " + get_response(ints, intents))
    name = input("You: ")

    # Wait for user input for the number
    if name:
        number = input("Bot: Please provide the contact number.\nYou: ")
        if number.isdigit():
            contacts[name] = number
            print(f"Bot: Contact '{name}' with number '{number}' saved successfully!")
        else:
            print("Bot: Please enter a valid numerical value for the contact number.")
```

This also checks if the number entered is all in numeric digits or not. It throws out an error message to handle that as well so that it can manage only numerical values while saving the contact.

The integration of conversational design principles enhances the chatbot's effectiveness by promoting clarity, maintaining context awareness, and fostering user engagement. These principles contribute to a user-centric design that prioritizes the user's experience and ensures the chatbot's adaptability to diverse conversational scenarios.

FLOWCHART

Following is the flowchart for the main functionality of the chatbot. It includes the two major function of contact modification and deletion. It also has error message as a part of it



DISCUSSION

In reflecting on the results and application of our chatbot, we align with the Responsible Research and Innovation framework. This approach ensures ethical considerations, user well-being, and societal impact are integral to our project. The evaluation, incorporating user feedback and technical assessments, adheres to principles of transparency, inclusivity, and accountability. Our chatbot, designed for seamless interaction, prioritizes user experience and respects privacy. Moving forward, a commitment to Responsible Research and Innovation will guide continual refinement, addressing emerging challenges ethically and responsibly, and ensuring our technology contributes positively to society.

CONCLUSION

In conclusion, the developed chatbot is a versatile and user-friendly conversational agent for seamless contact management. The project introduced the chatbot's purpose, emphasizing saving, modifying, and displaying contact information. The meticulously crafted architecture revolves around a dynamic neural network model, enabling adaptive intent recognition for responsive interactions. The codebase, structured into training and chatbot modules, leverages NLTK and Keras for natural language processing and machine learning.

Conversational design principles guided development, ensuring clarity, context awareness, and user engagement. The implementation prioritizes clear and concise responses, context maintenance, and dynamic conversations for an enhanced user experience.

Usability and performance testing validated the chatbot's effectiveness in recognizing intents and providing accurate responses. User feedback played a crucial role in refining conversational design and functionality.

In essence, the project delivered a functional and user-centric chatbot aligned with conversational design principles. The adaptable architecture and user-focused approach position the chatbot as a valuable tool for intuitive contact management. As technology evolves, these principles and technologies lay a foundation for future sophisticated and user-friendly conversational agents.