

TITLE: CodTech IT Solutions Internship – Task Documentation **SIMPLE PYTHON CHATBOT** Using NLP Techniques.

INTERN INFORMATION:

Name: Bhagya Sri Hani Allu

ID: COD5944

INTRODUCTION

In today's digital age, chatbots have become an integral part of many online platforms, offering assistance, information, and even entertainment to users. Chatbots allow businesses to connect with customers in a personal way without the expense of human representatives. In python uses many libraries such as NLTK, spacy. A chatbot is a computer program that can communicate with humans in a natural language. Essentially, NLP is the specific type of artificial intelligence used in chatbots. NLP stands for Natural Language Processing. It's the technology that allows chatbots to communicate with people in their own language. In other words, it's what makes a chatbot feel human. The main use of chatbots can solve customer concerns and queries in multiple languages. Their 24/7 access enables customers to use them regardless of time or time Zone. Chatbot, short for chatterbot, is an artificial intelligence feature that can be embedded and used through any major messaging application.

Understanding Chatbots:

Chatbots are computers programs designed to simulate conversation with human users. They can understand natural language input from users, process it, and generate appropriate responses. Chatbots can be deployed in various domains, including customer service, e-commerce, healthcare, and education, to automate tasks, provide information, and engage users in interactive conversations.

Benefits of chatbots:

- **Cost Saving:** Chatbots automate repetitive tasks and reduce the workload on human agents, leading to cost savings for businesses in terms of labour and operational expenses.
- **Scalability:** Chatbots can handle multiple conversations simultaneously, scaling to meet increasing demand without the need for additional human resources.
- **24/7 Availability:** Chatbots can interact with users at any time, providing instant assistance and support, which improves customer satisfaction and response times.

- **Accessibility:** Chatbots are accessible through various channels, including websites, messaging apps, and voice assistants, making them convenient for users to access information and support wherever they are.

Role of Natural Language Processing (NLP):

NLP is a subfield of artificial intelligence concerned with the interaction between computers and human language. It encompasses tasks such as speech recognition, natural language generation, and language translation. NLP techniques enable chatbots to analyze and interpret textual data, extract meaning from unstructured text, and generate human-like responses. NLP powers many applications that use language, such as text translation, voice recognition, text summarization, and chatbots. You may be used some of these applications yourself, such as voice-operated GPS systems, digital assistants, speech-to-text software, and customer service bots. NLP also helps businesses improve their efficiency, productivity, and performance by simplifying complex tasks that involve language.

NLP Tasks:

- Speech recognition.
- Part of speech tagging.
- Word sense disambiguation.
- Names entity recognition.
- Natural language generation.

IMPLEMENTATION

Implementing a simple Python chatbot using NLP Techniques.

1. Installation: Ensure you have Python installed on your system. Additionally, install the Required libraries using the following command:

`pip install nltk`

Type this command in command prompt and install this after run the code in VS code.

2. Define the Objective: Decide on the purpose of your chatbot. Is it for customer service, providing information.

3. Choose a Framework or Library: Select an appropriate NLP framework or library for project like NLKT, spaCy.
4. Data Collection: Gather or generate conversational data relevant to your chatbot's purpose.
5. Preprocessing: Preprocess the collected data to clean and tokenize it. This may involve removing punctuation, stop words, and performing stemming.
6. Training Data Preparation: Split the pre-processed data into training and resting datasets.
7. Model Selection: Choose an NLP model architecture suitable for your task. This could be a rule-based model, sequence-to-sequence model.
8. Model Training: Train the NLP model using the prepared training data.
9. Integration: Integrate the trained model into the Python chatbot application. This typically involves setting up an interface for users to interact with the chatbot.
10. Use Input Processing: Process user input by tokenizing and preprocessing it, then feeding it into the trained NLP model for prediction.
11. Response Generation: Generate a response based on the output of the NLP model.

CODE EXPLANATION

- The **'chatbot_response'** function takes a user input as a parameter and returns an appropriate response based on that input.
- The **'nltk'** library is imported to utilize its natural language processing functionalities.
- The **'patterns'** list contains tuples where the first element is a regular expression pattern to match user input, and the second element is a regular expression pattern to match user input, and the second element is a list of possible responses.
- The **'reflections'** dictionary contains reflection patterns for transforming user input.
- An instance of the **'chat'** class is created using the defined patterns and reflections.
- The **'main()'** function handles user interactions by continuously prompting for input and providing responses until the user decides to quit.
- When the user enters **"quit"**, the conversation ends, and the program terminates.

Execution:

1. Run the Python script.
2. You will be greeted by the chatbot.
3. Type your messages in the console.
4. To end the conversation, type “quit”.

Marking Tasks as Completed:

Utilizes the source code by tasks marked as completed is when a task is implemented as simple chatbot for user interactions using natural language processing techniques for easy understanding as human users. Run the task and see the output in the VS code terminal.

Removing Tasks:

The close button(‘X’) added to each task allows users to remove tasks from the code .This is shows the removing tasks in the project terminal.

CONCLUSION:

In conclusion, developing a simple Python chatbot using Natural Language Processing(NLP) techniques offers a glimpse into the capabilities of leveraging computational linguistics to create conversational agents that can understand and respond to human language effectively. As NLP technology continues to advance, chatbots are becoming increasingly sophisticated, offering new opportunities for automation and improving user experiences across various domains. Additionally, you could expand the functionality by incorporating external APIs, databases, or machine learning models to enhance the chatbot’s capabilities. Overall, this code serves as a starting point for building more complex and interactive chatbots.

OUTPUT:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

sers\Bhagya Sri hani\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\Bhagya Sri hani\.vscode\extensions\ms-python.debugpy-2024.2.0-win32-x64\bundl
ed\libs\debugpy\adapter\..\..\debugpy\launcher' '49918' '--' 'c:\Users\Bhagya Sri hani\OneDrive\Desktop\chat.py'

Welcome to the chatbot!
You can start chatting. Type 'quit' to exit.
You: hello
Bot: Hey there!
You: what is your name?
Bot: I'm Chatbot, nice to meet you!
You: How are you?
Bot: I'm doing well, thank you!
You: good bye
Bot: None
You: Goodbye
Bot: See you later!
You: █
```

```
folder instead.
You can clone a repository locally.
Clone Repository

To learn more about how to use Git and source control in VS Code read our docs.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

You: hello
Bot: Hey there!
You: what is your name?
Bot: I'm Chatbot, nice to meet you!
You: How are you?
Bot: I'm doing well, thank you!
You: good bye
Bot: None
You: Goodbye
Bot: See you later!
You: exit
Bot: None
You: hello
Bot: Hi!
You: how are you?
Bot: I'm fine, thanks for asking.
You: what are doing?
Bot: None
You: what is your name?
Bot: You can call me Chatbot.
You: ok
Bot: None
You: goodbye
Bot: Bye!
You: █
```