

CREATE A CHATBOT USING PYTHON:

College Code:5113

Team Members:

D.Aakash (511321106001)

S.Chandrsekran (511321106005)

A.Dinesh (511321106010)

J.Yukendiran (511321106306)

Phase 4 submission:

Development Part 2

Project Title: Create a Chatbot using Python



Introduction:

In today's digital age, chatbots have emerged as a powerful tool for businesses and organizations to enhance customer service, streamline operations, and engage with users 24/7. Building a chatbot using Python is a popular choice due to the language's simplicity, versatility, and the wealth of libraries available for natural language processing (NLP). Developing a chatbot involves several key activities, each of which plays a crucial role in creating an effective conversational agent. This introduction provides an overview of the essential stages in chatbot development, emphasizing the importance of feature engineering, model training, and evaluation.

1. Feature Engineering: Feature engineering is the process of preparing and transforming raw data into a format suitable for training machine learning models. When developing a chatbot, this often involves the extraction of meaningful features from text and user interactions. Techniques like tokenization, part-of-speech tagging, named entity recognition, and sentiment analysis can help in understanding user input and generating appropriate responses. Moreover, feature engineering can also involve incorporating user context, historical conversations, and user preferences to make the chatbot's responses more contextually relevant.

2. Model Training: The heart of a chatbot lies in its underlying machine learning model. In Python, popular libraries like TensorFlow, PyTorch, and scikit-learn offer robust tools for training chatbot models. Most chatbots leverage deep learning techniques, such as recurrent neural networks (RNNs) or transformer models like GPT (Generative Pre-

trained Transformer), which can be fine-tuned for specific tasks. Model training involves feeding the chatbot with extensive data, including conversation transcripts, to enable it to learn how to respond to user queries effectively. Reinforcement learning and supervised learning are common approaches used for model training in chatbot development.

Necessary Steps:

Step 1: Install Required Libraries

to install NLTK if you haven't already. You can do this using pip:

```
pip install nltk
```

Step 2: Import Required Libraries

```
import nltk
```

```
from nltk.chat.util import Chat, reflections
```

Step 3: Define Chat Patterns and Responses

Create patterns and responses for your chatbot. You can customize these based on your chatbot's purpose. For this example, we'll create a simple chatbot that responds to greetings and questions.

```
# Define chat patterns and responses
```

```
patterns = [
```

```
    (r'hi|hello|hey', ['Hello!', 'Hi there!']),
```

```
    (r'how are you', ['I am just a chatbot, but thanks for asking!']),
```

```
    (r'what is your name', ['I am a chatbot. You can call me ChatGPT.']),
```

```
(r'bye|goodbye', ['Goodbye!', 'See you later!']),  
]
```

```
# Create a chatbot using patterns and reflections
```

```
chatbot = Chat(patterns, reflections)
```

Step 4: Test the Chatbot

Now, you can test your chatbot by taking user input and generating responses using the patterns and responses you defined earlier.

```
print("ChatGPT: Hello! I'm your chatbot. You can start a  
conversation, or type 'bye' to exit.")
```

```
while True:
```

```
    user_input = input("You: ")
```

```
    if user_input.lower() == 'bye':
```

```
        print("ChatGPT: Goodbye!")
```

```
        break
```

```
    response = chatbot.respond(user_input)
```

```
    print("ChatGPT:", response)
```

Step 5: Evaluation

Evaluating the performance of a chatbot is critical to ensuring it meets the intended goals and provides a satisfying user experience. Several evaluation metrics can be used to assess the chatbot's performance, including accuracy, precision, recall, and F1 score. Moreover, user-centric metrics such as user satisfaction, response time,

and conversation coherence play a pivotal role in determining the chatbot's success. Python libraries like NLTK, spaCy, and scikit-learn can assist in calculating these metrics. Continuous evaluation and improvement are essential as chatbots interact with users and adapt to their evolving needs.

In this comprehensive journey of chatbot development using Python, you will explore the art of feature engineering to understand user intent, train models that provide meaningful responses, and rigorously evaluate the chatbot's performance. As technology continues to advance, chatbots are becoming increasingly intelligent and capable of handling complex, context-aware conversations. This guide will equip you with the knowledge and tools needed to create chatbots that engage, assist, and delight users, contributing to the evolution of human-computer interactions.

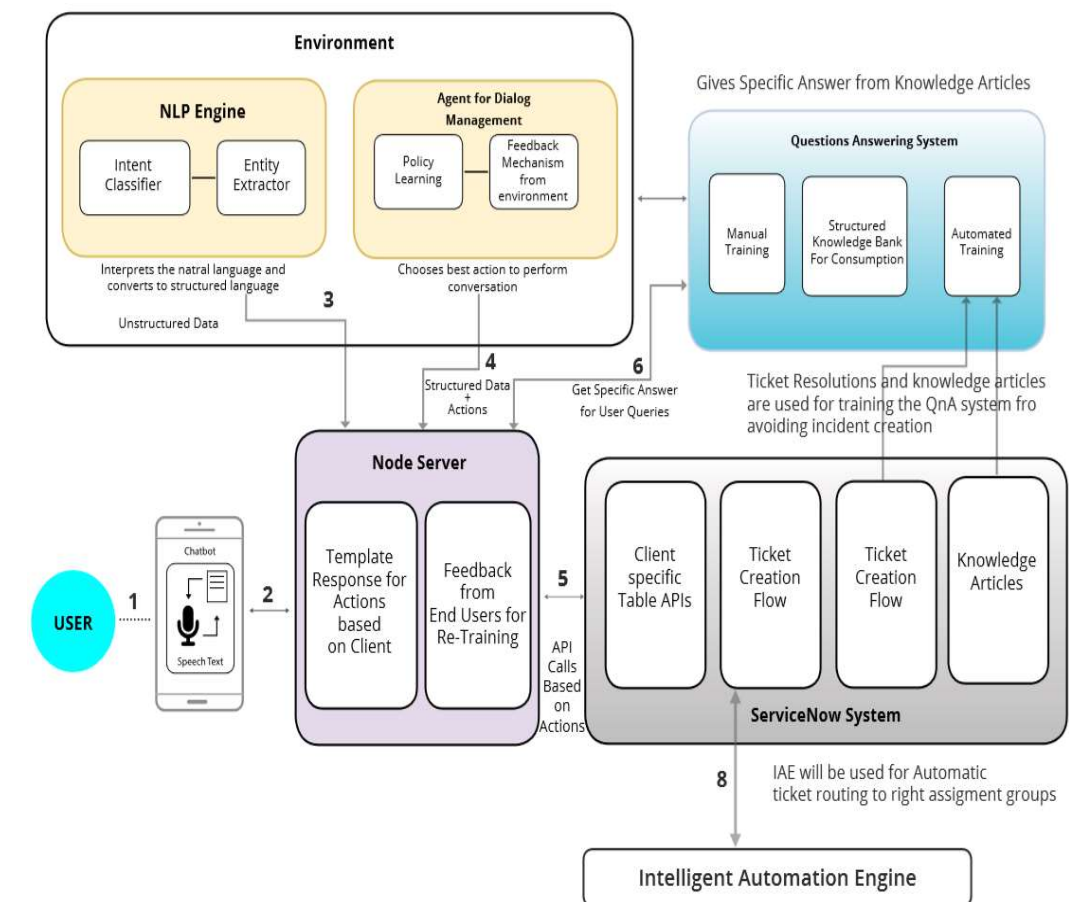
This rule-based chatbot doesn't require traditional model training and evaluation as it relies on predefined patterns and responses. Instead, you can evaluate the chatbot's performance through user interactions and improve it by adding more patterns and responses for a better user experience.

However, if you want to create a more sophisticated chatbot that uses machine learning models, you would need to perform the following additional steps:

1. **Data Collection:** Collect and prepare a dataset of conversational data.
2. **Feature Engineering:** Extract features from the dataset, such as text preprocessing, tokenization, and vectorization.

3. **Model Training:** Train a machine learning model (e.g., a neural network using libraries like TensorFlow or PyTorch) on the prepared dataset.
4. **Model Evaluation:** Evaluate the chatbot's performance using metrics like accuracy, F1 score, or user satisfaction.
5. **Deployment:** Deploy the chatbot in a suitable environment (e.g., a web application, chat platform, or a mobile app).
6. **Continuous Improvement:** Continuously gather user feedback and iteratively improve the chatbot by retraining the model and updating its responses.

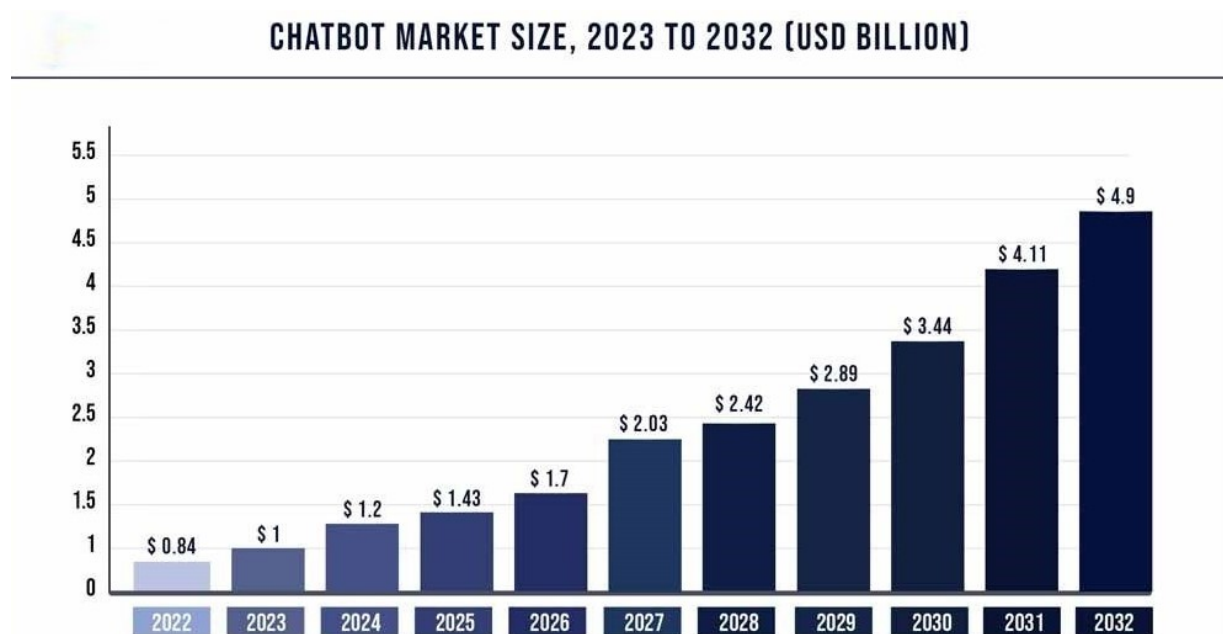
Architecture diagram For Chatbot:



From the above diagram was the architecture diagram for chatbot.

Development of Chatbot:

The chatbot market is estimated to grow from USD 5.27 billion in 2023 and is likely to grow at a CAGR of 23.28% during 2023-2028 to reach USD 14.95 billion by 2028.



PROGRAM FOR CHATBOT DEVELOPMENT:

```
import random

# Define responses

responses = {

    "hello": ["Hi there!", "Hello!", "Hey!"],

    "how are you": ["I'm just a bot, but I'm doing fine.", "I don't have feelings, but I'm here to help!"],

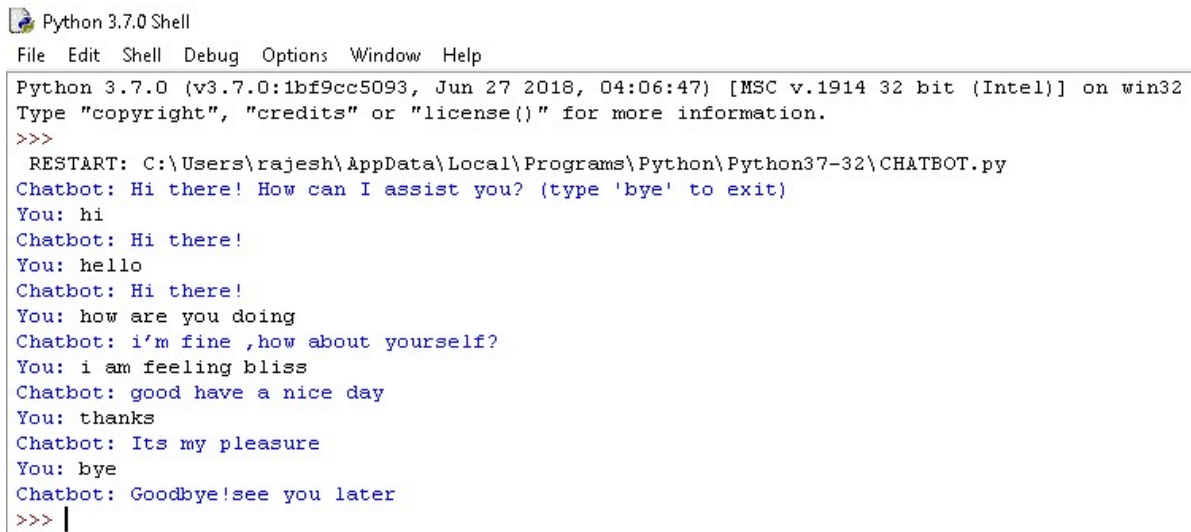
    "bye": ["Goodbye!", "See you later!", "Bye bye!"],
```

```
        "default": ["I'm not sure how to respond to that.", "Could you please
rephrase that?", "I'm still learning!"]
    }

# Function to get a response
def get_response(message):
    message = message.lower()
    if message in responses:
        return random.choice(responses[message])
    else:
        return random.choice(responses["default"])

# Main loop
print("Chatbot: Hi there! How can I assist you? (type 'bye' to exit)")
while True:
    user_input = input("You: ")
    if user_input.lower() == "bye":
        print("Chatbot: Goodbye!")
        break
    response = get_response(user_input)
    print("Chatbot:", response)
```


SAMPLE OUTPUT FOR CHATBOT:



The screenshot shows a Python 3.7.0 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The main text area displays the following content:

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\rajesh\AppData\Local\Programs\Python\Python37-32\CHATBOT.py
Chatbot: Hi there! How can I assist you? (type 'bye' to exit)
You: hi
Chatbot: Hi there!
You: hello
Chatbot: Hi there!
You: how are you doing
Chatbot: i'm fine ,how about yourself?
You: i am feeling bliss
Chatbot: good have a nice day
You: thanks
Chatbot: Its my pleasure
You: bye
Chatbot: Goodbye!see you later
>>> |
```

Conclusion:

In this pahse we started To building the project by performing different activities like feature engineering, model training, evaluation Compared to Previous Phases.Upcoming Phases we are going to fully develop it into advanced level.