Bangalore Institute of Technology

M.Tech, Department of Computer Science and Engineering
K R Road, V V Pura, Bengaluru-560004

Mini Project Synopsis on

AI Chat Bot

Submitted as Mini Project for the M.Tech Lab Component of IPCC subject

Artificial Intelligence and Machine Learning (22SCS22)

Submitted by
Spoorthy UK 1BI22SCS06

For academic year 2022-23

Under the guidance of
Dr. M S Bhargavi
Associate Professor
Dept. of CSE

Introduction to the Problem:

In a world increasingly driven by digital interaction, the development of conversational agents, or chatbots, has emerged as a pivotal challenge. These AI-powered systems are designed to simulate human-like conversations, offering a wide range of applications from customer support to information retrieval. This code delves into this challenge by presenting two distinct methodologies for crafting functional chatbots. The first approach employs NLTK's rule-based chat module, constructing a chatbot that responds based on predefined patterns and rules. The second approach utilizes the ChatterBot library to create a machine learning-based chatbot, capable of learning from data and generating contextually relevant responses. This code not only showcases the technical aspects of building chatbots but also allows users to interact with and compare these two contrasting methodologies, shedding light on the intricacies and trade-offs involved in creating effective conversational agents.

Problem Statement:

The problem addressed by this project is the development and implementation of functional chatbot systems. These systems, designed to engage in human-like conversations, have become increasingly important in various domains, including customer service, information delivery, and entertainment. The code explores this challenge by presenting two distinct strategies for creating chatbots. The first strategy involves creating a rule-based chatbot using NLTK's chat module, which responds based on predefined patterns and rules. The second strategy focuses on building a machine learning-based chatbot using the ChatterBot library, enabling the bot to learn from data and generate contextually appropriate responses. Through user interaction, the code provides insights into the differences between these approaches, highlighting the complexities and choices involved in developing effective and engaging chatbot systems.

Objective:

The primary objective of this AI chatbot project is to showcase and compare two distinct methodologies for creating functional chatbot systems. Through the implementation of a rule-based chatbot using NLTK's chat module and a machine learning-based chatbot using the ChatterBot library, the project aims to provide users with hands-on experience and insights into the differences between these two approaches. By interacting with both chatbot systems, users can gain an understanding of how rule-based patterns and machine learning techniques influence response generation and user engagement. Ultimately, the project seeks to offer an educational and experiential platform that highlights the challenges, capabilities, and trade-offs involved in developing effective and interactive chatbot systems.

Dataset description with Snapshot of dataset:

In this project we have not used any particular datasets, here we have used some sentences and some questions and responses to train the chatbot. Here the ChatterBot-

based chatbot is being trained using the built-in ChatterBot English corpus, and no external dataset is explicitly mentioned or used for training. The ChatterBot library comes with a pre-built corpus of conversational data that can be used to train the chatbot. This corpus includes a variety of text conversations covering different topics and contexts.

The code snippet that pertains to the training of the ChatterBot-based chatbot is as follows:

```
chatbot = ChatBot('MyChatBot')
trainer = ChatterBotCorpusTrainer(chatbot)
trainer.train("chatterbot.corpus.english")
```

Fig 1: Snapshot-1

In the given context , the list of pairs we've defined can be considered as a form of dataset. This list of pairs contains predefined user inputs (patterns) along with corresponding chatbot responses. Each pair in the list serves as a training example for the rule-based chatbot created using NLTK's chat module.

```
pairs = [
    [
        r"how are you ?",
        ["I'm doing good. How about you?"]
    ],
    [
        r"what is the time?|time|whats the time?",
        [f"The current time is {datetime.datetime.now().strftime('%H:%M')}"]
    ],
    [
        r"what is today's date?|today's date|date",
        [f"Today's date is {datetime.date.today().strftime('%B %d, %Y')}"]
    ],
    [
        r"sorry (.*)",
        ["No problem", "Don't worry"]
    ],
```

Fig 2: Snapshot-2

Techniques that could be used in the project:

1. Rule-Based Technique:
The rule-based technique involves constructing a chatbot using NLTK's chat module. This technique relies on predefined patterns and responses to handle user interactions. Regular expressions are used to match user inputs to specific patterns, and when a match is found, the chatbot generates a response from the associated list of responses. This approach is straightforward to implement and is suitable for scenarios where interactions follow specific patterns.

2. Machine Learning-Based Technique:
The machine learning-based technique is implemented using the ChatterBot library. This approach involves training the chatbot using a dataset (in this case, the built-in ChatterBot English corpus) to learn from examples and generate responses. The ChatterBot library employs natural language processing and machine learning algorithms to understand patterns in user inputs and generate contextually relevant responses. This technique enables the chatbot to learn from a diverse range of conversations and generate responses that align with user queries.

Tools to be used:

1. NLTK (Natural Language Toolkit):

NLTK is a powerful Python library for natural language processing and text analysis. It provides tools and resources for working with human language data, including tokenization, stemming, part-of-speech tagging, and more. In this project, NLTK's `chat` module is used to create the rule-based chatbot. NLTK simplifies text processing tasks, making it easier to implement chatbot behavior based on predefined patterns.

2. ChatterBot Library:

ChatterBot is a Python library specifically designed for building chatbots using machine learning techniques. It provides a framework for creating conversational agents that can learn from text data and generate contextually relevant responses. In this project, ChatterBot is used to create the machine learning-based chatbot. It enables the chatbot to learn from the provided dataset and generate responses using algorithms trained on the data.

3.Datetime Module:

The built-in `datetime` module in Python is used to work with date and time-related functions. It's utilized to provide real-time information such as the current time and today's date in the responses generated by the chatbots.

IDE Used: Visual Studio Code

Methodology:

The project employs a two-fold methodology to create and compare two different chatbot systems: a rule-based chatbot using NLTK's chat module and a machine learning-based chatbot using the ChatterBot library. Here's a breakdown of the methodology:

1. **Rule-Based Chatbot (NLTK):**
   - **Data Collection:** The project begins by defining a list of conversation pairs consisting of user inputs (patterns) and chatbot responses. These pairs represent various scenarios and interactions that the chatbot is expected to handle.
   - **NLTK Implementation:**The NLTK library's `chat` module is utilized to create a rule-based chatbot. This involves configuring the predefined patterns and corresponding responses in the form of regular expressions and response lists.
   - **User Interaction:**Users are provided with the option to engage with the rule-based chatbot. The chatbot responds based on matching patterns and generating appropriate responses from the predefined list.

2. **Machine Learning-Based Chatbot (ChatterBot):**
   - **ChatterBot Library Integration:**The ChatterBot library is employed to create a machine learning-based chatbot. This chatbot employs natural language processing and machine learning techniques to generate contextually relevant responses.
   - **ChatterBot Training:**The ChatterBot instance is trained using the built-in ChatterBot English corpus. While the code doesn't explicitly detail the training process, the corpus provides a diverse range of conversation examples that the chatbot learns from.
   - **User Interaction:** Users are presented with the option to interact with the ChatterBot-based chatbot. The chatbot generates responses based on the training it has undergone on the ChatterBot English corpus.

3. **User Choice and Interaction:**
   - **User Selection:**The project allows users to choose between interacting with the rule-based chatbot or the machine learning-based chatbot. This choice determines the type of chatbot the user will engage with.
   - **Input and Output:**Users input their queries or messages, and the selected chatbot generates responses accordingly. The interactions aim to demonstrate the capabilities and behavior of both chatbot systems.

Final outcome expected

The final outcome of the project is a demonstrative showcase of two distinct chatbot systems—a rule-based chatbot and a machine learning-based chatbot—along with the opportunity for users to interact with and compare their functionalities. Users can engage with both chatbot systems, experiencing firsthand how each system responds to their queries and messages.