

# PROJECT REPORT

ARTIFICIAL INTELLIGENCE (CS-323)

“CHATBOT USING CHATTERBOT”



## GROUP MEMBERS:

MISHA AKRAM BAIG (CS-18118)

IQRA IRFAN (CS-18123)

SONIYA SHAFI (CS-18133)

MUHAMMAD MUJTABA KHAN (CS-18136)

## **Abstract**

In this project we have implemented a chatbot name as “My Chat Bot” in python “Tkinter” by using Chatterbot library. A chatbot is a computer program that can converse with humans using artificial intelligence in messaging platforms. Our chatbot will provide answers on the basis of training in natural language.

## Table of Contents

<b>1.What is a Chatbot.....</b>	<b>3</b>
1.1 Kinds of Chatbot.....	3
1.1.1 Chatbot that uses limited set of rules .....	3
1.1.2 Chatbot and Machine learning .....	3
<b>2. About Chatterbot .....</b>	<b>4</b>
2.1 How Chatterbot Works .....	4
2.2 Language Independence.....	4
2.3 What kind of Machine learning Chatterbot use? .....	4
<b>3. Training .....</b>	<b>5</b>
<b>4. Adapters .....</b>	<b>5</b>
<b>5. Input Methods .....</b>	<b>6</b>
<b>6. Libraries used .....</b>	<b>6</b>
<b>7. Source code .....</b>	<b>7</b>
<b>8. Outputs.....</b>	<b>9</b>

## **1. WHAT IS A CHATBOT??**

A **chatbot** is a software application used to conduct an on-line chat conversation via text or text-to-speech, in lieu of providing direct contact with a live human agent. Designed to convincingly simulate the way a human would behave as a conversational partner.

### **1.1 KINDS OF CHATBOT:**

Chatbots come in two kinds:

- Limited set of rules
- Machine learning

#### **1.1.1 Chatbot that uses limited set of rules:**

This kind of bots are very limited to set of texts or commands. They have ability to respond only to those texts or commands. If user asks something different or other than the set of texts or commands which are defined to the bot, it would not respond as desired since it does not understand or it has not trained what user asked. These bots are not very smart when compared to other kind of bots.

#### **1.1.2 Chatbot and Machine learning:**

Machine learning chatbots works using artificial intelligence. User need not to be more specific while talking with a bot because it can understand the natural language, not only commands. This kind of bots get continuously better or smarter as it learns from past conversations it had with people.

## **2. About ChatterBot**

ChatterBot is a Python library that makes it easy to generate automated responses to a user's input. ChatterBot uses a selection of machine learning algorithms to produce different types of responses. This makes it easy for developers to create chat bots and automate conversations with users.

### **2.1 How ChatterBot Works:**

An untrained instance of ChatterBot starts off with no knowledge of how to communicate. Each time a user enters a statement, the library saves the text that they entered and the text that the statement was in response to. As ChatterBot receives more input the number of responses that it can reply and the accuracy of each response in relation to the input statement increase.

The program selects the closest matching response by searching for the closest matching known statement that matches the input, it then chooses a response from the selection of known responses to that statement.

### **2.2 Language Independence**

The language independent design of ChatterBot allows it to be trained to speak any language. Additionally, the machine-learning nature of ChatterBot allows an agent instance to improve its own knowledge of possible responses as it interacts with humans and other sources of informative data.

### **2.3 What kinds of machine learning does ChatterBot use?**

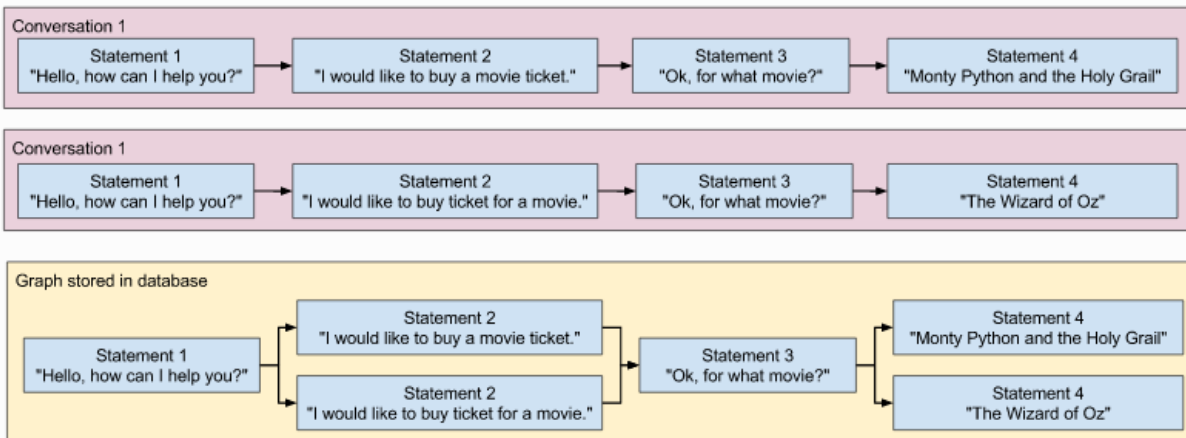
ChatterBot uses Search Algorithm. Search is a crucial part of how a chat bot quickly and efficiently retrieves the possible candidate statements that it can respond with. Some examples of attributes that help the chat bot select a response include

- the similarity of an input statement to known statements
- the frequency in which similar known responses occur

- the likeliness of an input statement to fit into a category that known statements are a part of

### **3. Training:**

ChatterBot includes tools that help simplify the process of training a chat bot instance. ChatterBot's training process involves loading example dialog into the chat bot's database. This either creates or builds upon the graph data structure that represents the sets of known statements and responses. When a chat bot trainer is provided with a data set, it creates the necessary entries in the chat bot's knowledge graph so that the statement inputs and responses are correctly represented.



ChatterBot comes with a corpus data and utility module that makes it easy to train bot to communicate.

### **4. Logic Adapters**

Logic adapters determine the logic for how ChatterBot selects a response to a given input statement. It is possible to enter any number of logic adapters for a bot to use. If multiple adapters are used, then the bot will return the response with the highest calculated confidence value. If multiple adapters return the same confidence, then the adapter that is entered into the list first will take priority.

### Best Match Adapter

The best match adapter uses a function to compare the input statement to known statements. Once it finds the closest match to the input statement, it uses another function to select one of the known responses to that statement.

### Time Logic Adapter

This adapter identifies statements in which a question about the current time is asked. If a matching question is detected, then a response containing the current time is returned.

### Mathematical Evaluation Adapter

The MathematicalEvaluation logic adapter parses input to determine whether the user is asking a question that requires math to be done. If so, the equation is extracted from the input and returned with the evaluated result.

## **5. INPUT METHODS:**

User can give input by both typing or by speech. Our chatbot is able to treat with both types of input methods.



## **6.LIBRARIES USED:**

### **6.1 Tkinter:**

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

### **6.2 SPEECH-RECOGNITION:**

Speech recognition means that when humans are speaking, a machine understands it. Here we are using Google Speech API in Python to make it happen.

### 6.3 PYAUDIO:

PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms.

### 6.4 THREADING:

Threading in python is used to run multiple threads (tasks, function calls) at the same time. Python threads are used in cases where the execution of a task involves some waiting.

## 7. SOURCE CODE:

```
from chatterbot import ChatBot
from tkinter import *
import speech_recognition as s
import threading

bot = ChatBot('My Bot',
              logic_adapters=[
                  "chatterbot.logic.BestMatch",
                  "chatterbot.logic.MathematicalEvaluation",
                  "chatterbot.logic.TimeLogicAdapter",
                  {"import_path": "chatterbot.logic.BestMatch",
                   "default_response": "sorry! I can't understand",
                   "maximum_similarity_threshold": 0.90}
              ])

from chatterbot.trainers import ChatterBotCorpusTrainer

trainer = ChatterBotCorpusTrainer(bot)

trainer.train([
    "chatterbot.corpus.english.greetings",
    "chatterbot.corpus.english.ai",
    "chatterbot.corpus.english.botprofile",
])

main=Tk()
main.geometry("500x650")
main.configure(background='grey')
main.option_add('*Font', 'Times 19')

main.title("My Chat bot")
img = PhotoImage(file="botg.png")
photoL = Label(main, image=img, width=2000, height=170)
```



```

photoL.pack(pady=5)

def takequery():
    sr=s.Recognizer()
    sr.pause_threshold=1
    with s.Microphone() as m:
        try:
            audio=sr.listen(m)
            query=sr.recognize_google(audio,language='eng-in')
            print(query)
            textF.delete(0,END)
            textF.insert(0,query)
            ask_from_bot()
        except Exception as e:
            print(e)

def ask_from_bot():
    query = textF.get()
    answer_from_bot = bot.get_response(query)
    msgs.insert(END, "you : " + query)

    msgs.insert(END, "bot : " + str(answer_from_bot))
    textF.delete(0, END)
    msgs.yview(END)

frame=Frame(main)
sc=Scrollbar(frame)
sc1= Scrollbar(frame, orient=HORIZONTAL)
msgs=Listbox(frame,width=200,height=10,background="lightblue",
foreground="blue",xscrollcommand=sc1.set)
sc.pack(side=RIGHT, fill=Y)
sc1.pack(side=BOTTOM, fill=X)
msgs.pack(side=LEFT, fill=BOTH, pady=10)

frame.pack()

textF=Entry(main ,background="lightblue", foreground="blue")
textF.pack(fill=X,pady=10)

btn=Button(main,text="Ask from Bot",command=ask_from_bot,background="lightblue",
foreground="blue" )
btn.pack()
def enter_function(event):
    btn.invoke()

main.bind("<Return>",enter_function)

def repeat():
    while TRUE:
        takequery()

t=threading.Thread(target=repeat)
t.start()
main.mainloop()

```

## 8. OUTPUTS:

