ARTIFICIAL INTILLIGENCE PROJECT REPORT

CHATBOT FOR EXAMINATION QUERY

Submitted To: Mr. Ishan Kumar Sir

Submitted By:

Name	Reg. I.D.	Rollno.
1.Nekkanti.Chakradhar	11811685	15
2.P. HarshaVardhan	11811680	14
3.Sai Sandesh	11811728	13

==

Abstract:

This project report is on our assigned topic- “Design a chatbot for examination query”.

As we know people like to interact with computers/systems more through voice assistants and chatbots, the project aims to build a chatbot that helps the user to get answers of his examination related queries.

In this project we have used python programming language to design it. We have used inbuilt python libraries to get the job done with ease. In our project we have under-taken LPUNEST exam to be the one about which the user will raise his queries. We trained our model with a set of questions that are most frequently asked by the students/parents and the chatbot answers them appropriately.

Introduction:

A chatbot is a software application used to make an on-line chat conversation via text or text-to-speech, in lieu of providing direct contact with a live human agent. It is a program that simulates a conversation between user and computer.

Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, hereby passing the Turing test.

Chatbots are typically used in dialog systems for various practical purposes including customer service or information acquisition.

Some chatterbots use sophisticated natural language processing systems, but many simpler systems scan for keywords within the input, then pull a reply with the most matching keywords, or the most similar wording pattern, from a database.

It should take care of the following things:

1. Understand who is the target audience

2. Understand the Natural Language of the communication.

3. Understand the intent or desire of the user

4. Provide responses that can answer the user

We can categorise chatbots in two types:

A) Self-learning chatbots: use machine learning and A.I. to save the input they get from user and use them later.

B) Rule-based chatbots: they follow certain rules on which the responses are based.

Chatter Bot is a Python library that makes it easy to generate automated responses to a user’s input and to create software that engage in conversation. Chatter Bot 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. The design of the chatbot is such that it allows the bot to be trained in variety of languages such as English, Hindi, Chinese, etc.

Additionally, the machine-learning nature of Chatter Bot allows an agent instance to improve its own knowledge of possible responses as it interacts with humans and other sources of informative data.

An untrained instance of Chatter Bot 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 Chatter Bot 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.

Steps involved in making a chatbot:

1. Creating a chatbot instance:

Before anything can be performed the chatter bot library needs to be imported.

from chatterbot import Chat Bot

bot= Chat Bot (‘Assignment’)

2. Setting the storage adapter:

Chatter Bot comes with built in adapter classes that allow it to connect to different types of databases.

In our project we have used SQL Storage Adapter which allows the chat bot to connect to SQL databases. It is a default adapter and it automatically gets executed if nothing is mentioned at first place.

By default, this adapter will create a SQLite database. The database parameter is used to specify the path to the database that the chat bot will use.

3. Specifying logical adapters:

The logic adapters parameter is a list of logic adapters. In Chatter Bot, a logic adapter is a class that takes an input statement and returns a response to that statement

For example:

bot = ChatBot(

```
'Assignment',storage_adapter='chatterbot.storage.SQLStorageAdapter',logic_adapters=[
        'chatterbot.logic.MathematicalEvaluation',
'chatterbot.logic.TimeLogicAdapter'],
database_uri='sqlite:///database.sqlite3' )
```

4. Getting response from the chatbot:

creating a while loop for bot to run in. We can exit the loop and stop the program when a user enters ctrl+c. while True:

```
 try:

 bot_input=bot.get_reponse(input())

 print(bot_input)

except(keyboardInterrupt, EOFError, SystemExit):

break
```

5. Training chatbot: we do it by importing a library

```
from chatterbot.trainers import ListTrainer

            trainer=ListTrainer(bot)

            trainer.train([ ‘hi’, ‘hello, how can I help you’,  ‘what can you tell me’, ‘I am here to help you about LPUNEST exam’, ‘thanks’, ‘happy to help!’]
```

Literature Review

For this project, we made a great use of Internet. We went through tutorial videos present on you tube to get an idea how will the whole process would be like.

We also made a great use of articles present on quora as they are written by engineering students themselves so from there we got an idea about what techniques could we use to get it done easily.

Sources used:

1. https://chatterbot.readthedocs.io/en/stable/tutorial.html : this is a online free tutorial for introduction to chatterbot library of python that is

widely used to make chat bots. From here we got a full fledge explanation of how to give input, train the bot and get a response back from it.

2. https://flask.palletsprojects.com/en/1.1.x/ : this is free documentation of FLASK, a web framework that provides tools, libraries, and technologies that allow you to build a web application, which contains the bot.

Methodology/Technique:

Working of chatterbot:

USER INPUT

to

PROCESS THE USER INPUT AND HIGHEST GENERATED CONFIDENCE VALUE IS RETURNED

to

RETURN THE RESPONSE FROM THE INPUT

<u>Training process:</u>

Conversation 1:

Statement 1: Hi! How can I help you?

Statement 2: What it the duration of Exam?

Statement 3: 3 hrs.

Conversation 2:

Statement 1: Hi! How can I help you?

Statement 2: How long will the exam take?

Statement 3: It is of 3 hrs, but can be submitted if done before time.

<u>Conclusion:</u>

Hi! How can I help you?

to

What is the duration of exam? How long will the exam take

To to

3 hrs. It is of 3 hrs, but can be submitted

if done before.

FLASK CHATBOT:

Make a web app using FLASK

to

APP HAVE THE CHATBOT

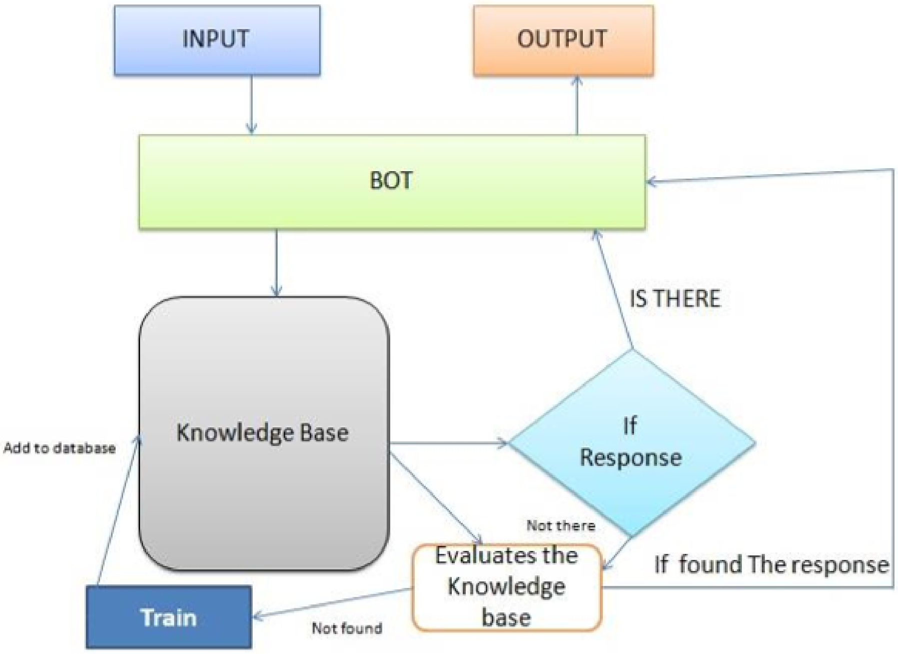
to

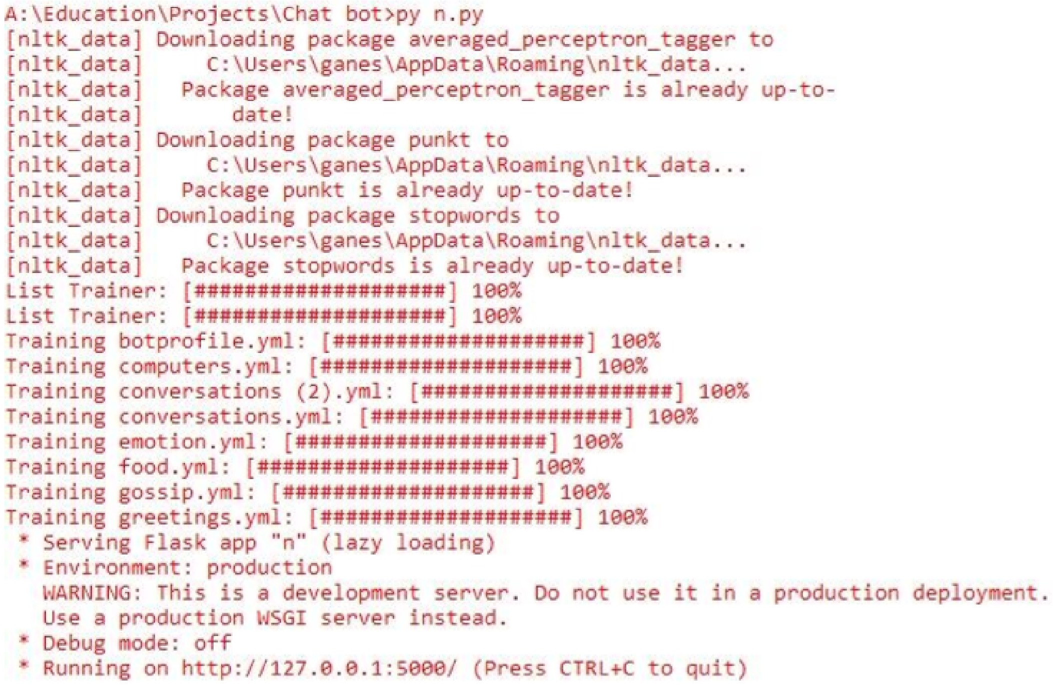
TRAIN THE BOT

to

PERFORM CONVERSATIONS

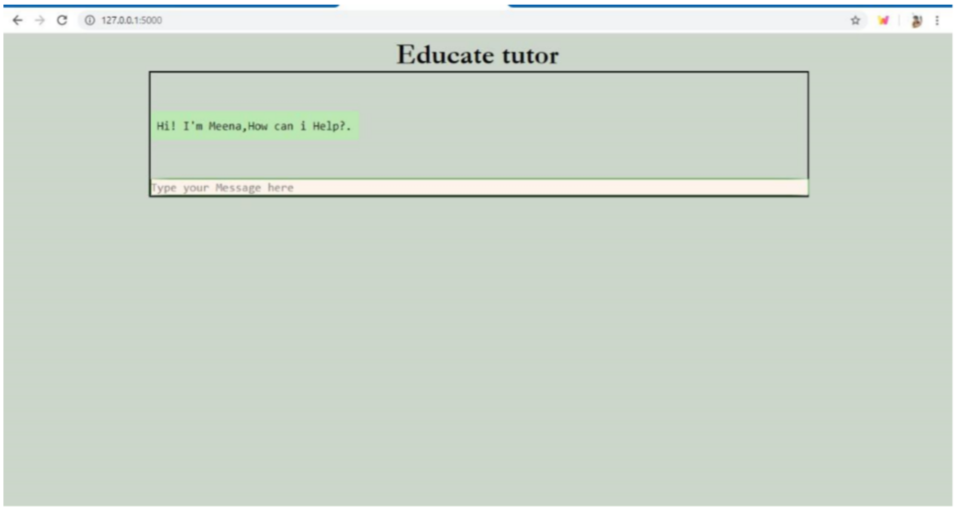
Flowchart of the program:





```
A:\Education\Projects\Chat bot>py n.py
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\ganes\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\ganes\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\ganes\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
List Trainer: [####################] 100%
List Trainer: [####################] 100%
Training botprofile.yml: [####################] 100%
Training computers.yml: [####################] 100%
Training conversations (2).yml: [####################] 100%
Training conversations.yml: [####################] 100%
Training emotion.yml: [####################] 100%
Training food.yml: [####################] 100%
Training gossip.yml: [####################] 100%
Training greetings.yml: [####################] 100%
 * Serving Flask app "n" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

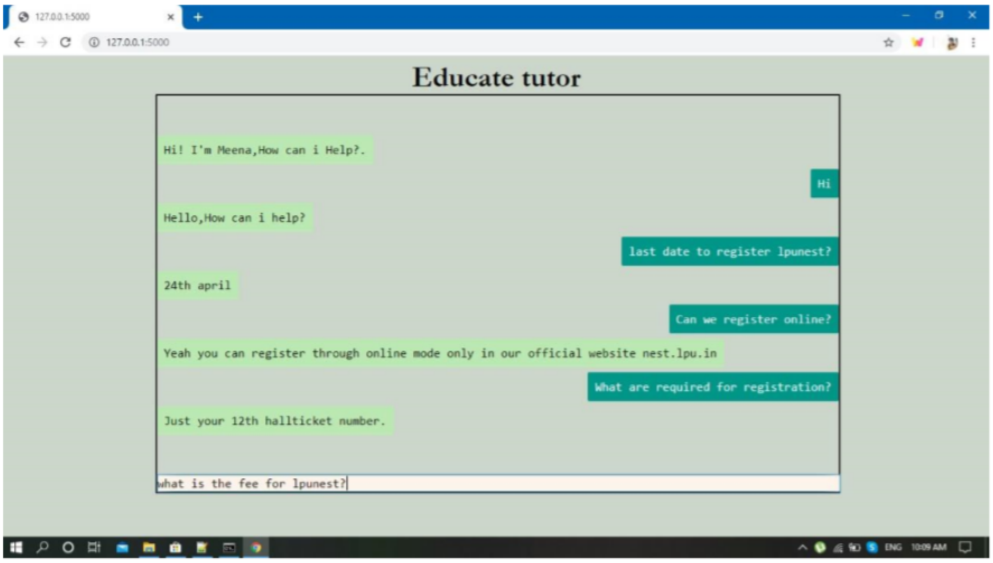
This is how the chatbot looks like:



Educate tutor

Hi! I'm Meena,How can i Help?.

Conversation with the user:



Educate tutor

Hi! I'm Meena,How can i Help?.

Hi

Hello,How can i help?

last date to register lpunest?

24th april

Can we register online?

Yeah you can register through online mode only in our official website nest.lpu.in

What are required for registration?

Just your 12th hallticket number.

what is the fee for lpunest?

Conclusion

At the end of this project we got to know a lot about python libraries and its versatility. We got a deep insight of how the world depends on simple chatbots and the science behind their working.

We understood that in coming years there will be a drastic increase in the number of chatbots being implemented within the service industry.

The vast amount of research that has been carried out, and is currently being done within the A.I.field has led to the rise of more advanced and intellectual chatbots. This will prove to be hugely advantageous in providing convenient and accessible customer service at a rapid scale.

References

https://chatterbot.readthedocs.io/en/stable/tutorial.html

https://flask.palletsprojects.com/en/1.1.x/

https://www.youtube.com/watch?v=tSjR7bk1Y9U

https://data-flair.training/blogs/python-chatbot-project/

https://towardsdatascience.com/build-your-first-chatbot-usingpython-nltk-5d07b027e727