COLLEGE ENQUIRY CHATBOT

February 2022

Abstract

At the most basic level, a **Chatbot** is a computer program that simulates and processes human conversation (either in text or speech), allowing humans to interact with digital devices as if they are communicating with a real person. Chatbots work on the principle of *AI* **(Artificial Intelligence)** and *ML* **(Machine Learning)**. We, therefore, shall be implementing a virtual assistant based on these principles which can solve any college related query. This will work as a College Oriented Intelligence machine. This virtual machine will respond to the queries of students as well as visitors on college related issues .This chatbot will help students,teachers and especially new visitors to get any information regarding the college quickly.
The college website may contain all the details required to know about the college, but this chatbot shall provide an easy and convenient access to all those information.

Keywords- Chatbot, Artificial Intelligence, Machine Learning.

STUDENT NAME	REGD NO	SIGNATURE
Alisha Satpathy	1901105169	
Satyabir Bhoi	1901105208	
Smita Jha	1901105212	
T. R. Rajalaxmi	1901105223	

MENTOR SIGNATURE

1 Introduction

A chatbot or chatterbot 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.A chatbot is a type of software that can help customers by automating conversations and interact with them through messaging platforms.Designed to convincingly simulate the way a human would behave as a conversational partner, chatbot systems typically require continuous tuning and testing, and many in production remain unable to adequately converse, while none of them can pass the standard Turing test.The term "ChatterBot" was originally coined by Michael Mauldin (creator of the first Verbot) in 1994 to describe these conversational programs.
Our chatbot is simply a retrieval based chatbot (with some aspects of a generative chatbot.) which can answer college related queries easily...

2 Background study

In 1950, Alan Turing's famous article "Computing Machinery and Intelligence" was published, which proposed what is now called the Turing test as a criterion of intelligence. This criterion depends on the ability of a computer program to impersonate a human in a real-time written conversation with a human judge to the extent that the judge is unable to distinguish reliably—on the basis of the conversational content alone—between the program and a real human. The notoriety of Turing's proposed test stimulated great interest in Joseph Weizenbaum's program ELIZA, published in 1966, which seemed to be able to fool users into believing that they were conversing with a real human. However Weizenbaum himself did not claim that ELIZA was genuinely intelligent, and the introduction to his paper presented it more as a debunking exercise:

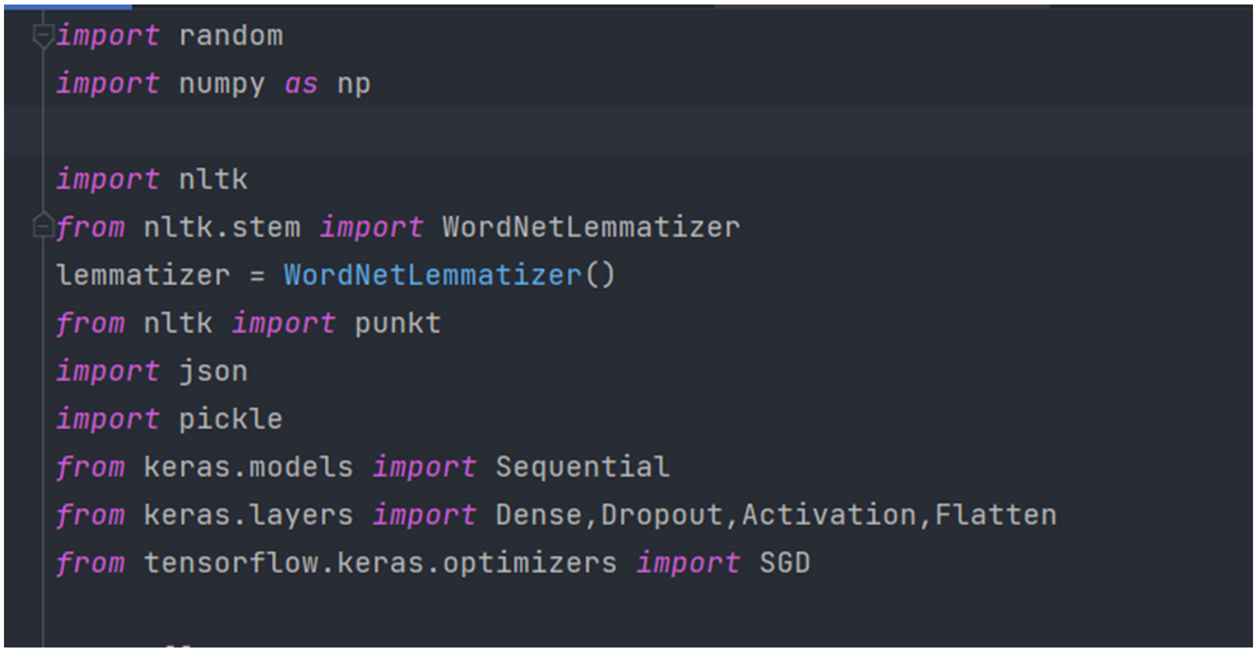
Artificial intelligence machines are made to behave in wondrous ways, often sufficient to dazzle even the most experienced observer. But once a particular program is unmasked, once its inner workings are explained its magic crumbles away; it stands revealed as a mere collection of procedures. The observer says to himself "I could have written that". With that thought, he moves the program in question from the shelf marked "intelligent", to that reserved for curios ... The object of this paper is to cause just such a re-evaluation of the program about to be "explained". Few programs ever needed it more.

In the year 2009, a company called WeChat in China created a more advanced Chatbot. Since its launch, WeChat has conquered the hearts of many users who demonstrate an unwavering loyalty to it.Now the chatbots have become more familiar with popular ones like Siri,Cotana,Alexa,Google Now and the fictional Jarvis

3 Methodology

Our chatbot is basically a retrieval based chatbot having some features of generative chatbot.
Our chatbot is purely based on Python language.We have used python 3.9 version for developing both the front end and backend.
The steps involved in making a machine learning based chatbot have been listed below:

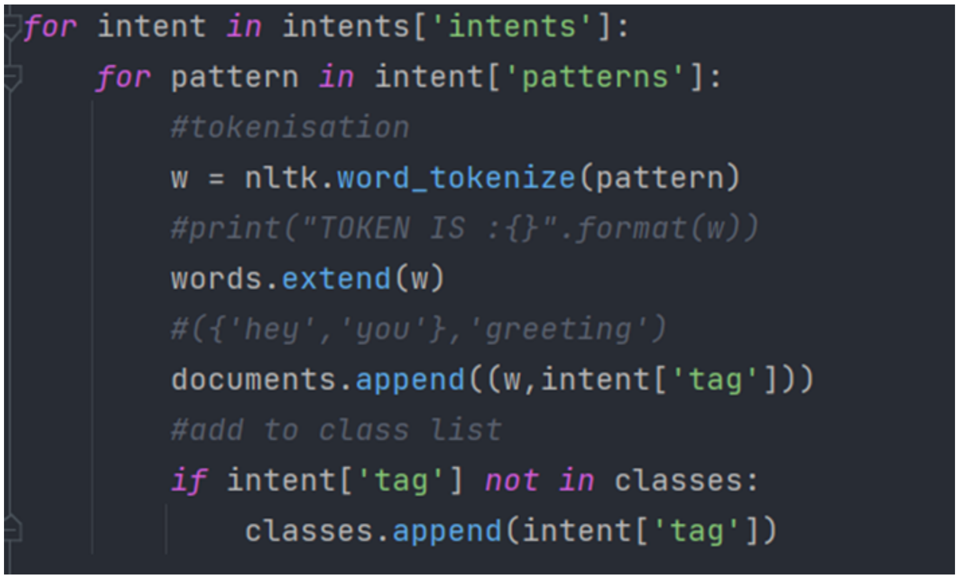
1.Installing required packages



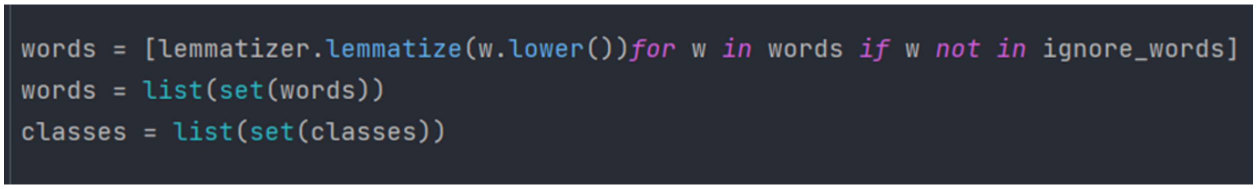
```
import random
import numpy as np

import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
from nltk import punkt
import json
import pickle
from keras.models import Sequential
from keras.layers import Dense,Dropout,Activation,Flatten
from tensorflow.keras.optimizers import SGD
```

Figure 1 : Installing packages..

2.Data preprocessing and loading required files



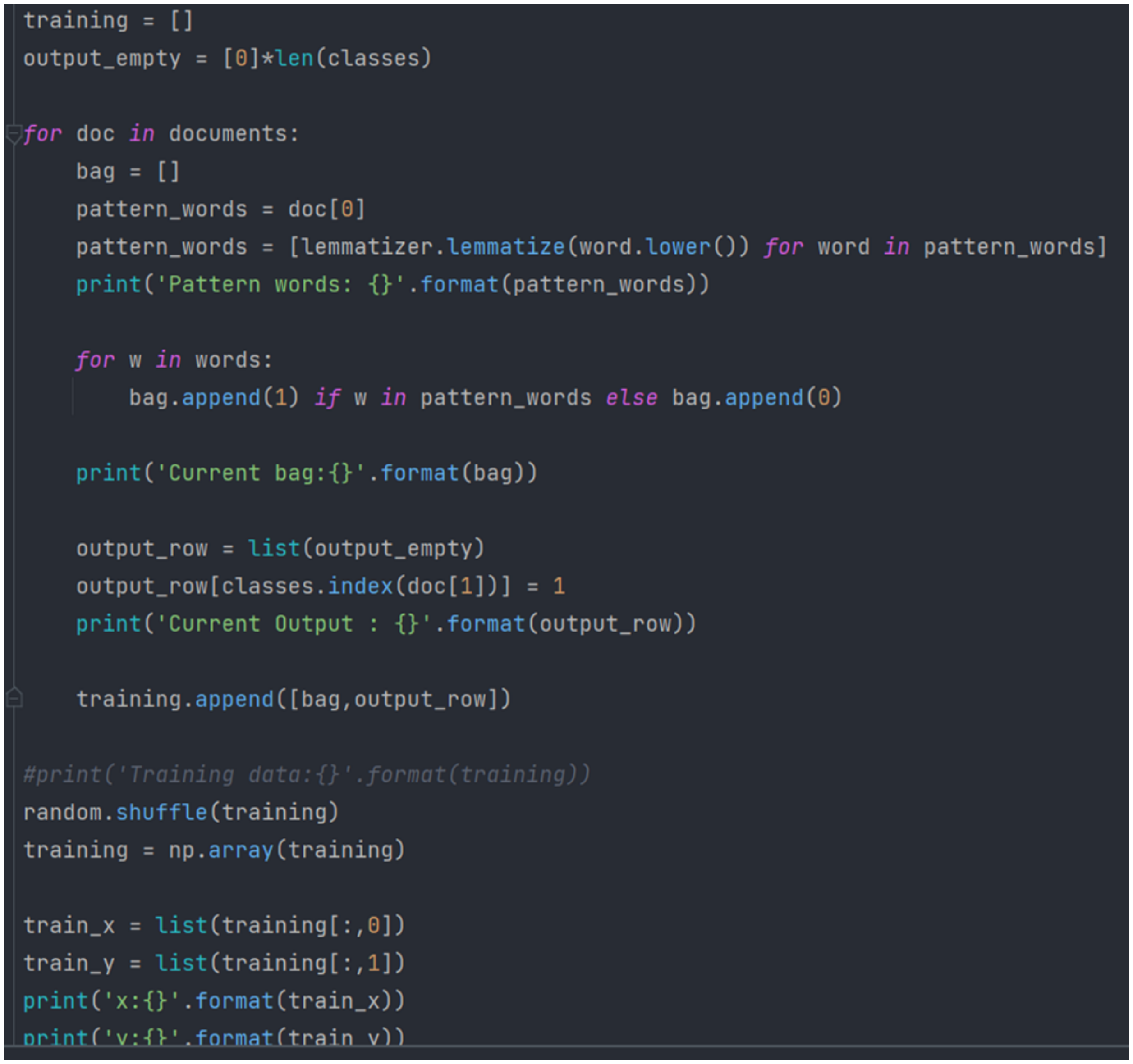
```
for intent in intents['intents']:
    for pattern in intent['patterns']:
        #tokenisation
        w = nltk.word_tokenize(pattern)
        #print("TOKEN IS :{}".format(w))
        words.extend(w)
        #({'hey','you'},'greeting')
        documents.append((w,intent['tag']))
        #add to class list
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
```



```
words = [lemmatizer.lemmatize(w.lower())for w in words if w not in ignore_words]
words = list(set(words))
classes = list(set(classes))
```

Figure 2 : Data preprocessing..

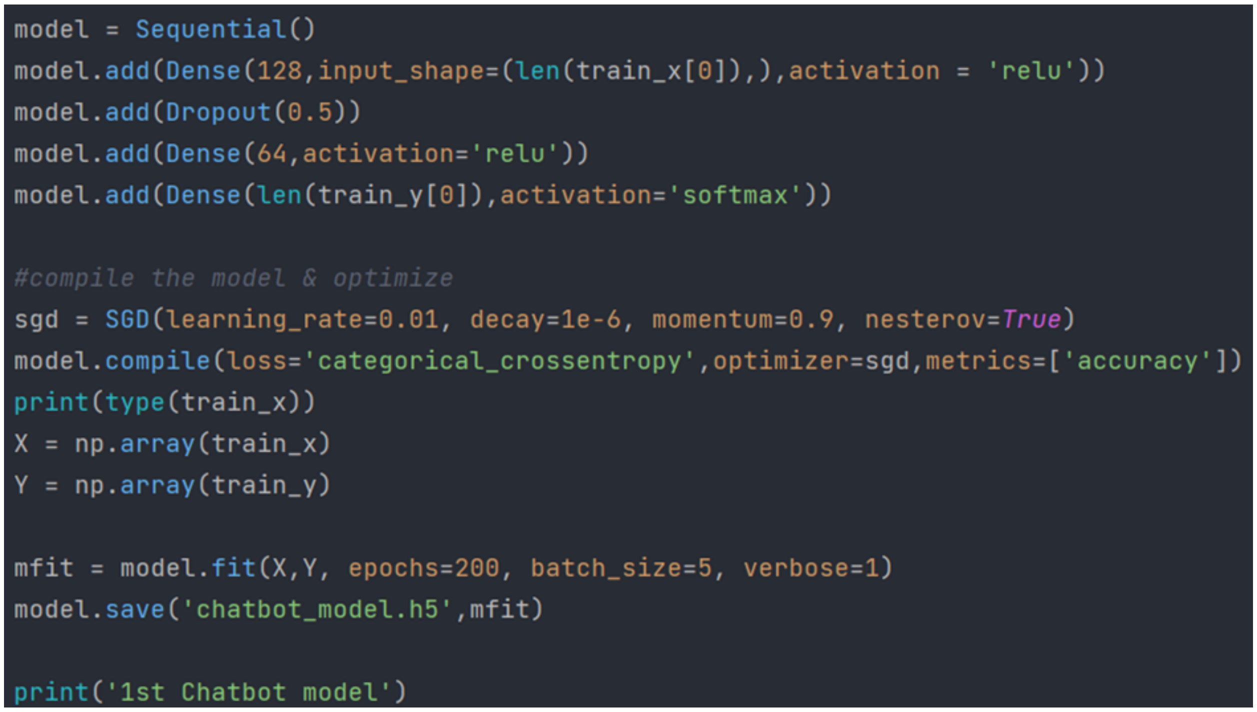
3.Training and testing



```
training = []
output_empty = [0]*len(classes)

for doc in documents:
    bag = []
    pattern_words = doc[0]
    pattern_words = [lemmatizer.lemmatize(word.lower()) for word in pattern_words]
    print('Pattern words: {}'.format(pattern_words))

    for w in words:
        bag.append(1) if w in pattern_words else bag.append(0)

    print('Current bag:{}'.format(bag))

    output_row = list(output_empty)
    output_row[classes.index(doc[1])] = 1
    print('Current Output : {}'.format(output_row))

    training.append([bag,output_row])

#print('Training data:{}'.format(training))
random.shuffle(training)
training = np.array(training)

train_x = list(training[:,0])
train_y = list(training[:,1])
print('x:{}'.format(train_x))
print('y:{}'.format(train_y))
```

Figure 3 : Training model..

4.Building the model



```
model = Sequential()
model.add(Dense(128,input_shape=(len(train_x[0]),),activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(64,activation='relu'))
model.add(Dense(len(train_y[0]),activation='softmax'))

#compile the model & optimize
sgd = SGD(learning_rate=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy',optimizer=sgd,metrics=['accuracy'])
print(type(train_x))
X = np.array(train_x)
Y = np.array(train_y)

mfit = model.fit(X,Y, epochs=200, batch_size=5, verbose=1)
model.save('chatbot_model.h5',mfit)

print('1st Chatbot model')
```

Figure 4 : building model..

1.Installing required packages

We have used various python packages for making this project.
These packages are:
A.Numpy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

B.keras

Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make implementing deep learning models as fast and easy as possible for research and development.Keras was developed and maintained by François Chollet, a Google engineer using four guiding principles:
Modularity: A model can be understood as a sequence or a graph alone. All the concerns of a deep learning model are discrete components that can be combined in arbitrary ways.
Minimalism: The library provides just enough to achieve an outcome, no frills and maximizing readability.
Extensibility: New components are intentionally easy to add and use within the framework,

intended for researchers to trial and explore new ideas. Python: No separate model files with custom file formats. Everything is based on native Python.

C.tensorflow

TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

D.pickle

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it "serializes" the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

E.nltk

Natural language processing (NLP) is a field that focuses on making natural human language usable by computer programs. NLTK, or Natural Language Toolkit, is a Python package that you can use for NLP.

A lot of the data that you could be analyzing is unstructured data and contains human-readable text. Before you can analyze that data programmatically, you first need to preprocess it.

For UI and Front end

We have also used some other packages for GUI part such as data scrapping , text to speech and speech to text features,etc.These packages are-Tkinter,PIL,gtts,etc

a)tkinter–

The tkinter package ("Tk interface") is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems. This framework provides Python users with a simple way to create GUI elements using the widgets found in the Tk toolkit. Tk widgets can be used to construct buttons, menus, data fields, etc. in a Python application.

b)gtts–

gTTS (Google Text-to-Speech), a Python library and CLI tool to interface with Google Translate's text-to-speech API. Write spoken mp3 data to a file, a file-like object (bytestring) for further audio manipulation, or stdout. Or simply pre-generate Google Translate TTS request URLs to feed to an external program.

Necessary steps involed at the frontend part....

1. Building the GUI part

2.Data preprocessing and loading required files

We have stored our college data in a json file.We have implemented tokenizing and lemmatizing methods by the help of which our chatbot can answer college related queries.

Tokenizing method: The first part of this method, we'll parse our JSON file called "Igit_intents.json" in Python language. Then, we'll do a tokenizing in our text data in search to break the whole text into small parts like words. Then append each word in the words list and also create a list of classes for our tags.

Lemmatizing method: Lemmatizing is the process of converting a word into its lemma form(root form) so, the second part consists of lemmatizing each word and removing duplicate words from the list . Then we use a pickle file to store the Python objects which we'll be using use while predicting.

3.Training and testing

Training data is the initial dataset that we use to teach a machine learning application to recognize patterns or perform to our criteria, while testing or validation data is used to evaluate your model's accuracy. Here we have used a json file to train our model under different criterias. Training is the method to get training sets data called training data responsible for providing us the input and output data for our future model. Our input is going be a pattern and output will be that class to which our input pattern will belong to. The last job is to convert all the texts into numbers.

4.Building the model

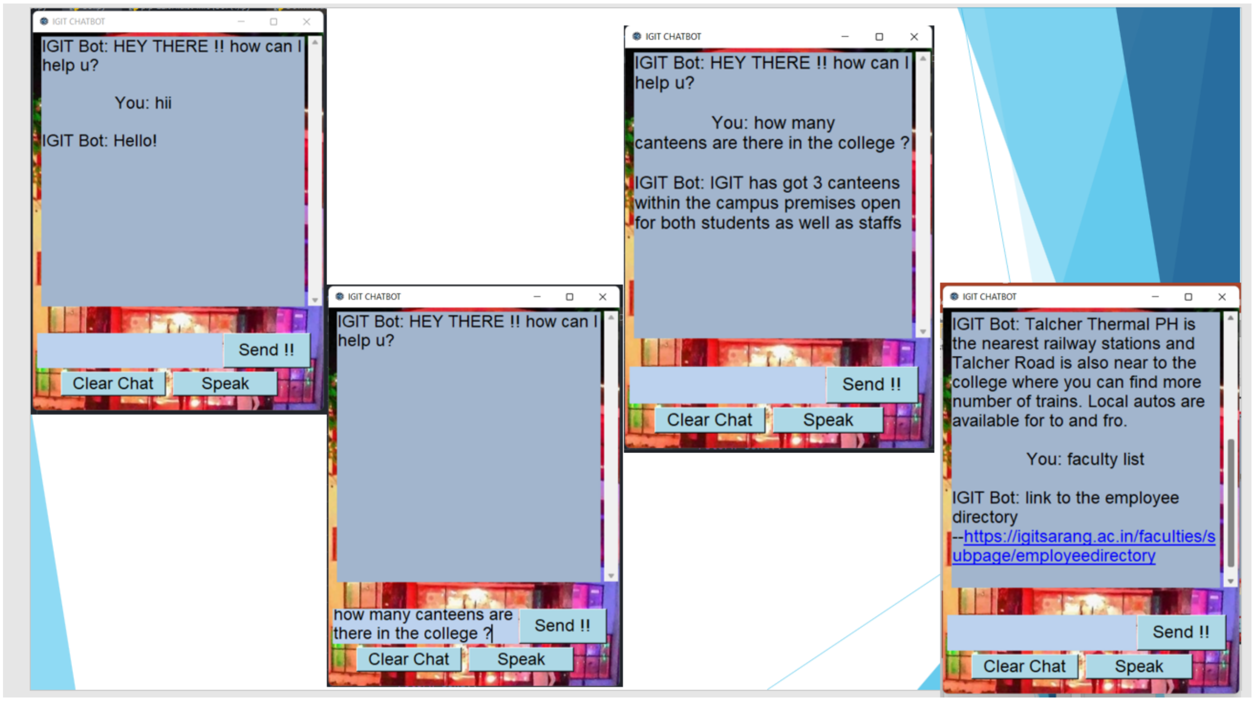
Inside our chatbot.py file in model the method to train our mode contains a deep natural network that has 3 layers (128, 64 and the last one has a same size with our number of intents) also, we'll have a frequency of rate of 0.5 at each step during training time.

Then, with the Keras sequential API, we'll compile our model with a stochastic gradient descent and with our training, we'll use 200 epochs and save it as chatbot$_model.h5$.

Building the GUI part...

We have used the Tkinter library for preparing the GUI part. When our chatbot will get the input message from the user then it will use some methods to get the response and for displaying it. We'll need to provide input data in the same way as we did while training. Then the following activities such as text preprocessing and prediction of the class will take place in the backend.We would get a random response from the list of intents.

Other text to speech and speech to text aspects


IGIT Bot: HEY THERE !! how can I help u?

You: hii

IGIT Bot: Hello!

IGIT Bot: HEY THERE !! how can I help u?

how many canteens are there in the college ?

IGIT Bot: HEY THERE !! how can I help u?

You: how many canteens are there in the college ?

IGIT Bot: IGIT has got 3 canteens within the campus premises open for both students as well as staffs

IGIT Bot: Talcher Thermal PH is the nearest railway stations and Talcher Road is also near to the college where you can find more number of trains. Local autos are available for to and fro.

You: faculty list

IGIT Bot: link to the employee directory --https://igitsarang.ac.in/faculties/subpage/employeedirectory

Figure 5 :Other text to speech and speech to text aspects..

Text to speech :

The pyttsx3 is a text-to-speech conversion library in Python is used to convert text to speech. However gTTS package can also be used for this purpose but it requires internet connection.

Speech to text :

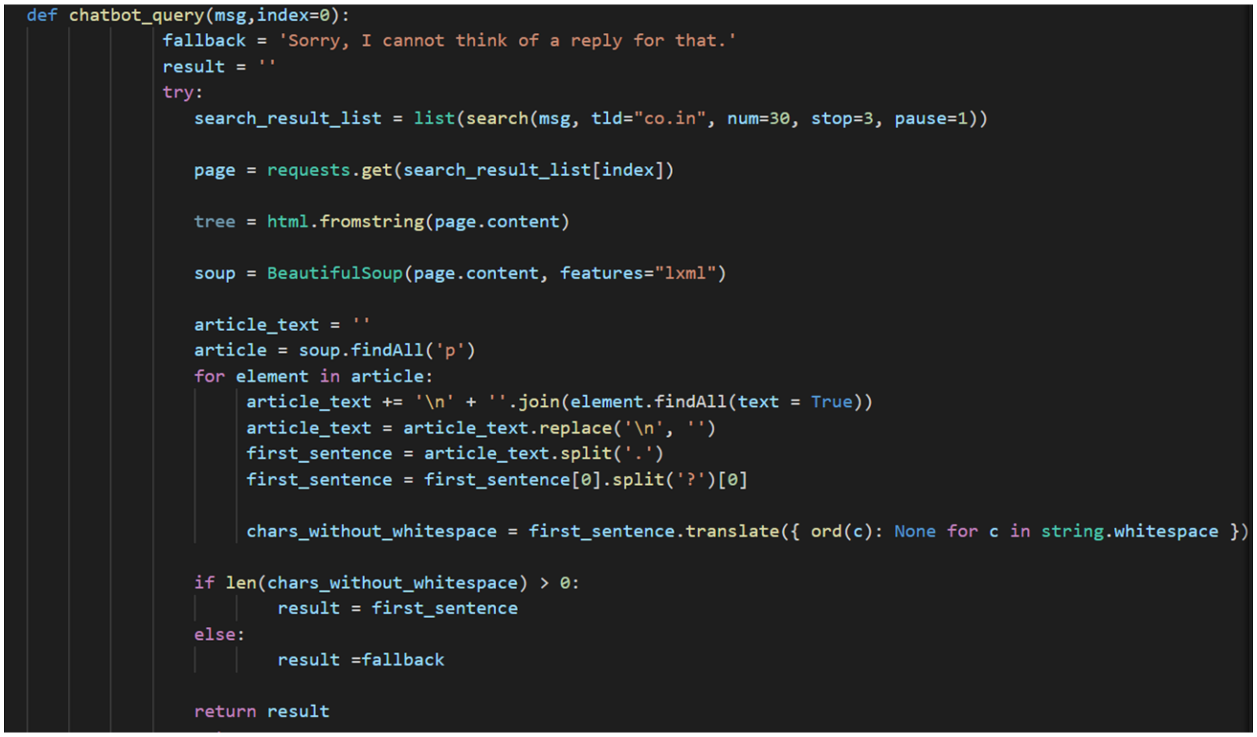
The SpeechRecognition library in python has been used in this project to convert speech to text which can work on voice queries also.

Our Chatbot also has a generative aspect!!

Apart from answering college related queries our chatbot has been trained in such a manner that it can also deal with general queries of the users (such as who is missile man of india,etc) gracefully.

So ,igitchatbot has essence of both retrieval based and generative chatbot.

Our ChatBot will perform a Google Search of a user's query, scrape the text from the first result, and reply to the user with the first sentence of that page's text.



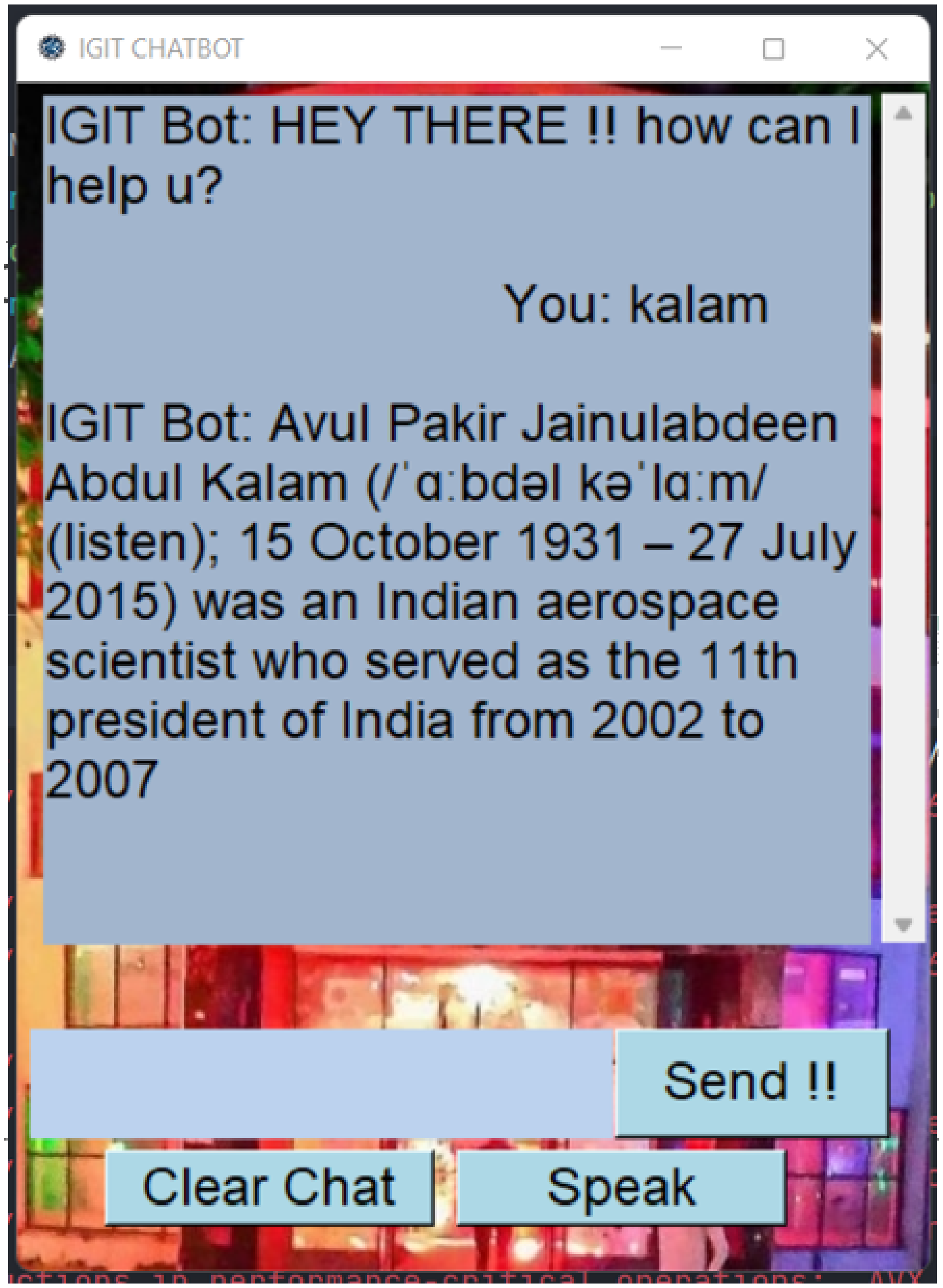
```python
def chatbot_query(msg,index=0):
            fallback = 'Sorry, I cannot think of a reply for that.'
            result = ''
            try:
               search_result_list = list(search(msg, tld="co.in", num=30, stop=3, pause=1))

               page = requests.get(search_result_list[index])

               tree = html.fromstring(page.content)

               soup = BeautifulSoup(page.content, features="lxml")

               article_text = ''
               article = soup.findAll('p')
               for element in article:
                    article_text += '\n' + ''.join(element.findAll(text = True))
                    article_text = article_text.replace('\n', '')
                    first_sentence = article_text.split('.')
                    first_sentence = first_sentence[0].split('?')[0]

                    chars_without_whitespace = first_sentence.translate({ ord(c): None for c in string.whitespace })

               if len(chars_without_whitespace) > 0:
                       result = first_sentence
               else:
                       result =fallback

               return result
```

Figure 6 :data scrapping from google..

4 RESULT ANALYSIS

Our chatbot not only deals with college related queries but can also answer random questions in a graceful manner.But still there is a scope of development .] Chatbots are based on artificial intelligence.They are been developed just to assist human beings and provide them information quickly .

Our chatbot not only deals with college related queries but can also answer random questions in a graceful manner.But still there is a scope of development .


IGIT Bot: HEY THERE !! how can I help u?

You: kalam

IGIT Bot: Avul Pakir Jainulabdeen Abdul Kalam (/ˈɑːbdəl kəˈlɑːm/ (listen); 15 October 1931 – 27 July 2015) was an Indian aerospace scientist who served as the 11th president of India from 2002 to 2007

Figure 7 : View of igit chatbot