TODEVELOPACHATBOTS USINGPYTHONBASED ONMACHINELEARNING

AReport fortheReview ETE ofChatBotmaking.

S.No	Enrollment no:	Admission Number	StudentName	Degree/ Branch	Sem
02	19021011806	19SCSE1010652	SamirShekharSingh	B-Tech/ CSE	V
03	19021011791	19SCSE1010634	VivekKumarSingh	B-Tech/ CSE	V

UndertheSupervision of

Mr.ShubhamKumar

School of Computing Science and EngineeringGreater Noida,UttarPradesh

Fall2021 – 2022

TABLEOFCONTENTS

S.No	Particulars	Page No
1	Abstract	
2	LiteratureReviews/Comparativestudy	
3	ProblemFormulation	
4	Toolsusedforimplementation	
5	MeritsofProposedsystem	
6	ArchitectureDiagramforProposed method	
7	Implementation and Description ofProjectModules(Atleasttwomodules shouldbementioned)	
8	References	

ABSTRACT

A **CHATBOT** is a software application used to conduct an on-line chatconversion via text or text-to-speech, in lieu of providing directcontactwithalivehumanagent. This software are used to performtasks such as quickly responding to users, informing them, helping topurchaseproductsandprovidingbetterservicetocustomers. Chatbots, or conversational interfaces as they are also known,presentanewwayforindividualstointeractwithcomputersystems.Traditionally, to get a question answered by a software programinvolvedusingasearchengine,orfillingoutaform.Achatbotallowsa user to simply ask questions in the same manner that they wouldaddress a human. However, chatbots are currently being adopted atahighrateoncomputerchatplatforms.

A chatbot can be used anywhere a human is interacting with acomputer system.Theseare theareaswherethefastestadoptionisoccuring :- humanresource,marketing,etc.

We have choosen chatbot as our project because it plays a vital rolein our day -day life And there is really vast scope of improvement inthisfield

As

Chatbotsareincreasinglypresentinbusinessesandoftenareusedtoautomate tasks that do not require skill-based talents. Withcustomer service taking place via messaging apps as well as phonecalls, there are growing numbers of use-cases where chatbotdeployment gives organisations a clear return on investment. Callcentreworkers maybeparticularlyatrisk fromAI-drivenchatbots.

LITERATUREREVIEWS

Chatbots are an AI-powered software program to facililateconversationwithourcustomers.Wecanprogramourchatbots withspecific answer to frequently asked. The best off chatbots is that weshould design this in machine learning – so the customers can gettheireverytypeofanswer.

In the marketing fields the chatbots are playing vitals role and helpsvery much in business. Chatbots have the great for managing theinitials steps of the marketing process, gathering contact details andinformation for a sales calls, answering general customers servicesquestionorprovidingdirectiononcommontechissues.Therecan bea huge disconnect between marketing and sales; may be marketingdoes not know what sales need to be successful or sales does notknow exactly how the marketing funnel is set up (and whereprospectusareinthesalescyclewhentheyhittheirdesk).

So to overcome this gap between marketing and sales chatbots isvery best. Especially like in running Digital era period where ishappeningdigitallyinthisdigitalera.Bots areamoreefficientwayofgathering information, qualifying leads and setting the sales team upfor success. By looping strategy, we can get marketing and sales onthesamepageandconvert moreprospectsintocustomers.

Sowecanconcludethatbyusingchatbotsinourdigitalsmarketingitgetsmorerelevantsforcustomersandfornewlybusinesstosetuptheirbusiness morefasterinthiscompetitingworld.

Atlastthemotiveof the makingchatbotstogetthesuccessinthedigitalsmarketingasthedigitalmarketingisvitalnowadays.

PROBLEMSFORMULATION

As we all aware about the fact that, A CHATBOT is asoftware application used to conduct an on-line chatconversion via text or text-to-speech, in lieu of providingdirectcontactwithalivehumanagent. These software areused to perform tasks such as quickly responding to users,informing them, helping to purchase products and providingbetterservice tocustomers.

After spending hour's on this project, we all come toconclusionsthatChabotisreallycosteffectiveand itissomethingwhichisavailable24/7forany clients.

And it can handle multiple client at the same time but there isan problem which we spot that, Sometimes it reply out of thecontext oftheconversation.

sobykeepingthisinmind wearegoingtoimproveitfurthersothat it can reply in precise and concise manner. Which reallygoingtosavelotsoftimeofourclientswhoareinteractingwithourChabot.

And so for that we are going to use python language whichhelpsultimatelytoourChabottolearn independentlyand itwillbeeasy toupdateittimetotime.

TOOLSFORIMPLEMENTATION

BASICSREQURIEMENT

To build a chatbot in Python, we have to import all the necessarypackages and initialize the variables we want to use in our chatbotproject.Also,WearegoingtoworkOntextdataforthat,weneedtoperform data preprocessing on Our dataset before designing an MLmodel.

for that we are going to use tokenizing That helps with text data – ithelps fragment the large text dataset into smaller, readable chunks(like words). Once that is done, We can also go for lemmatizationthat transforms a word into its lemma form. Then it creates a picklefile to store the python objects that are used for predicting theresponsesof thebot.

STEPSFORCREATING ACHATBOT:

STEP-1

The first step in creating a chatbot in Python with theChatterBotlibraryistoinstallthelibraryinOursystem.

WearegoodtoinstallChatterBot’s latestdevelopmentversiondirectly from GitHub. For this, you will have to write andexecutethefollowingcommand:

```
pip
installgit+git://github.com/gunthercox/ChatterBot.git@master
```

STEP-2

Importing classesisthe secondstepinthePythonchatbotcreation process.
All we need to do is import two classes – ChatBot fromchatterbotandListTrainerfromchatterbot.trainers.
Todothis,wehavetoexecutethefollowingcommand:From chatterbotimportChatBot.
Fromchatterbot.trainersimportListTrainer.

STEP-3

The3rdstepistoCreateandTraintheChatbot.
The chatbot we are creating will be an instance of the class“ChatBot.” After creating a new ChatterBot instance, we aregoing to train the bot to improve its performance. Trainingensuresthatthebothasenoughknowledgetogetstartedwithspecific responses to specific inputs. We have to execute thefollowing commandnow:

My_bot=ChatBot(name=‘PyBot’,
read_only=True,Logic_adapters=
[‘chatterbot.logic.MathematicalEvaluation’,‘Chatterbot.logic.
BestMatch’])

Thecommand“logic_adapters”denotesthelistofadaptersusedto train thechatbot.
Whilethe“chatterbot.logic.MathematicalEvaluation” helpsthebottosolve mathproblems,the “chatterbot.logic.BestMatch”helpsittochoosethebestmatchfromthelistof responsesalreadyprovided.

STEP-4

The4thstepistoCommunicatewiththePythonChatbot TointeractwithPythonchatbot,Wearegoingtousethe.get_response()function.

This ishowitshould lookwhilecommunicating:
Print(my_bot.get_response(“hii))howdoyoudo?
Print(my_bot.get_response(“Ifeelawesometoday”))
excellent,gladtohearthat.

However, it is essential to understand that the our chatbotmight not know how to answer all our questions. Since
itsknowledgeandtrainingisstillverylimited,sowehavetogiveittimeandprovidemoretrainingdatatotrainitfurther.

STEP-5

The5thandlaststageistoTrainourPythonChatbotwithaCorpusofData

for training our python chatbot even further, We are going touse an existing corpus of data. Here’s an example of how wetrainourPythonchatbotwithacorpusofdataprovidedby thebot itself:
FromChatterBot.trainersimportChatterBotCorpusTrainerCorpus_trainer=ChatterBotCorpusTrainer(my_bot)Corpus_trainer.train(‘ChatterBot.corpus.english’)

The good thing is that ChatterBot offers this functionality inmanydifferentlanguages.So,wecanalsospecifyasubsetofacorpusinsomeotherlanguagealso.

MERITOFPROPOSEDSYSTEM

Therearesomanybenefitsofchatbots.Itisusedforvariouspurposeinvariousfield.It isnow widelyused.

Someofthecommonusedofchatbotsaremetionedbelow:

- □**Reducedcosts:**
  chatbots eliminate the need for various employees duringonlineinteractionwithcustomers.Thisisobviouslyhadagreatimpactoneconomyof thecompanies.

- □**24/7Availability**:
  Unlikehumans,onceweinstallachatbot,itcanhandlequeriesatanytimeofday.

- □**Learningandupdating:**
  _These chatbotsareabletolearnfrominteractionsandupdateindependently. This isoneof themainadvantages.

- □
  Managementofmultipleclients:
  Humans can serve a limited number of customers at the sametime. This restriction does not exist for chatbots, and they canmanageallthenecessaryqueriessimultaneously.

  The purpose of my chatbot is that how people get success indigitalmarketing.Asdigitalmarketingisveryvitalsnowadays.

  Asabovemetioned benefitswesawthatitmorerelevanttotheuser.

ImplementationandDescriptionofProjectModules

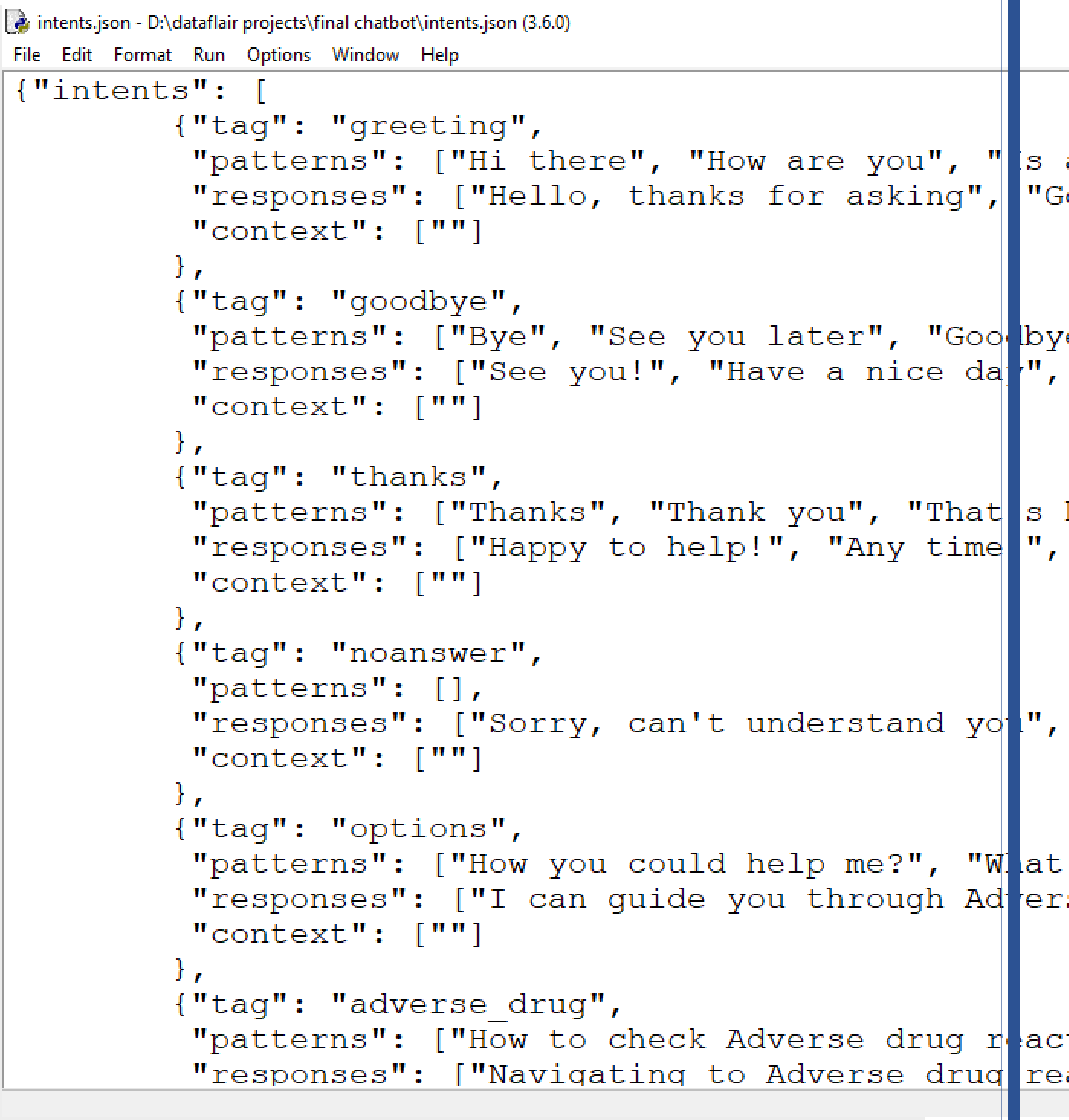
Toimplement theprojectatfirstwehavetoinstallthe Chatbotlibrary,Thenweare gonnafollowthesestep:-

First,makeafilenameastrain_chatbot.py.Weimportthenecessarypackages forour chatbotandinitializethevariableswewilluseinourPythonproject.

ThedatafileisinJSONformatsoweusedthejsonpackagetoparsetheJSONfileintoPython.

```
importnltk
fromnltk.stemimportWordNetLemmatizerl
emmatizer=WordNetLemmatizer()importjs
on
importpickleimport
numpyasnp
fromkeras.modelsimport Sequential
fromkeras.layersimportDense,Activation,Dropoutfromk
eras.optimizersimportSGD
import
randomwords=
[]classes =
[]documents=[
]
ignore_words=['?','!']
data_file =
```

Thisishowourintents.jsonfilelookslike.


intents.json - D:\dataflair projects\final chatbot\intents.json (3.6.0)
File Edit Format Run Options Window Help

```
{"intents": [
        {"tag": "greeting",
         "patterns": ["Hi there", "How are you", "Is a
         "responses": ["Hello, thanks for asking", "Go
         "context": [""]
        },
        {"tag": "goodbye",
         "patterns": ["Bye", "See you later", "Goodbye
         "responses": ["See you!", "Have a nice day",
         "context": [""]
        },
        {"tag": "thanks",
         "patterns": ["Thanks", "Thank you", "That's h
         "responses": ["Happy to help!", "Any time[illegible]",
         "context": [""]
        },
        {"tag": "noanswer",
         "patterns": [],
         "responses": ["Sorry, can't understand you",
         "context": [""]
        },
        {"tag": "options",
         "patterns": ["How you could help me?", "What
         "responses": ["I can guide you through Adver
         "context": [""]
        },
        {"tag": "adverse_drug",
         "patterns": ["How to check Adverse drug reac
         "responses": ["Navigating to Adverse drug rea
```

2.Preprocessdata

Whenworkingwithtextdata,weneedtoperformvariouspreprocessingonthedatabeforewe make a machine learning or a deep learning model. Tokenizing is the most basic andfirst thing you can do on text data. Tokenizing is the process of breaking the whole text intosmallparts likewords.

Here we iterate through the patterns and tokenize the sentence using nltk.word_tokenize()functionandappendeachwordinthewordslist.Wealsocreatea listofclasses forourtags.

```
for intent in
intents['intents']:for pattern in
intent['patterns']:#tokenizeeac
hword
w =
nltk.word_tokenize(pattern)wor
ds.extend(w)
#add documents in the
corpusdocuments.append((w,
intent['tag']))#addtoourclasseslist
```

Nowwewilllemmatizeeachwordandremoveduplicatewordsfromthelist.Lemmatizing istheprocessofconvertinga wordintoitslemma formand then creatinga picklefileto storethePython objects whichwewillusewhilepredicting.

```
#lemmatize,lowereachwordandremoveduplicates
words = [lemmatizer.lemmatize(w.lower()) for w in words if w not in
ignore_words]words= sorted(list(set(words)))
# sortclasses
classes=sorted(list(set(classes)))
# documents = combination between patterns and
intentsprint(len(documents),"documents")
# classes= intents
print (len(classes), "classes",
classes)# words
=allwords,vocabulary
print (len(words), "unique lemmatized words",
words)pickle.dump(words,open('words.pkl','wb'))
```

3.Createtrainingandtestingdata

Now,wewill createthetrainingdata inwhichwewillprovidetheinputand theoutput.Ourinput will be the pattern and output will be the class our input pattern belongs to. But thecomputerdoesn’t understandtext sowewillconvert textintonumbers.

```
#create ourtraining data
```

```
training = []
#createanemptyarrayforouroutputout
put_empty =[0] *len(classes)
# trainingset,bag ofwordsforeachsentence
fordocindocuments:
# initialize our bag of
wordsbag= []
# list of tokenized words for the
patternpattern_words = doc[0]
# lemmatize each word - create base word, in attempt to represent related
wordspattern_words = [lemmatizer.lemmatize(word.lower()) for word in
pattern_words]#createourbagof
wordsarraywith1,ifwordmatchfoundincurrentpattern
forw in words:
bag.append(1)ifwinpattern_wordselsebag.append(0)
#outputisa'0'foreachtagand'1'forcurrenttag(foreachpattern)output_row=
list(output_empty)
output_row[classes.index(doc[1])] =
1training.append([bag,output_row])
# shuffle our features and turn into
np.arrayrandom.shuffle(training)
training =np.array(training)
# create train and test lists. X - patterns, Y -
intentstrain_x=list(training[:,0])
train_y =
list(training[:,1])print("Traini
ngdatacreated")
```

4.Buildthemodel

Wehaveourtraining data ready,nowwewill builda deepneuralnetworkthat has3layers.We use the Keras sequential API for this. After training the model for 200 epochs, weachieved 100%accuracyonourmodel.Let ussavethemodel as‘chatbot_model.h5’.

```
# Create model - 3 layers. First layer 128 neurons, second layer 64 neurons and 3rd output layer
containsnumberof neurons
#equalto numberofintentsto predictoutput intentwithsoftmax
```

```
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(Dropout(0.5))model.add(De
nse(len(train_y[0]),activation='softmax'))
# Compile model. Stochastic gradient descent with Nesterov accelerated gradient gives good results
forthismodel
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9,
nesterov=True)model.compile(loss='categorical_crossentropy', optimizer=sgd,
metrics=['accuracy'])#fitting andsavingthemodel
hist=model.fit(np.array(train_x),np.array(train_y),epochs=200,batch_size=5,verbose=1)model.save('chatbot_m
odel.h5',hist)
print("model created")
```

6.Runthechatbot

To run the chatbot, we have two main files; **train_chatbot.py** and **chatapp.py**.First,wetrainthemodelusingthecommandin theterminal:

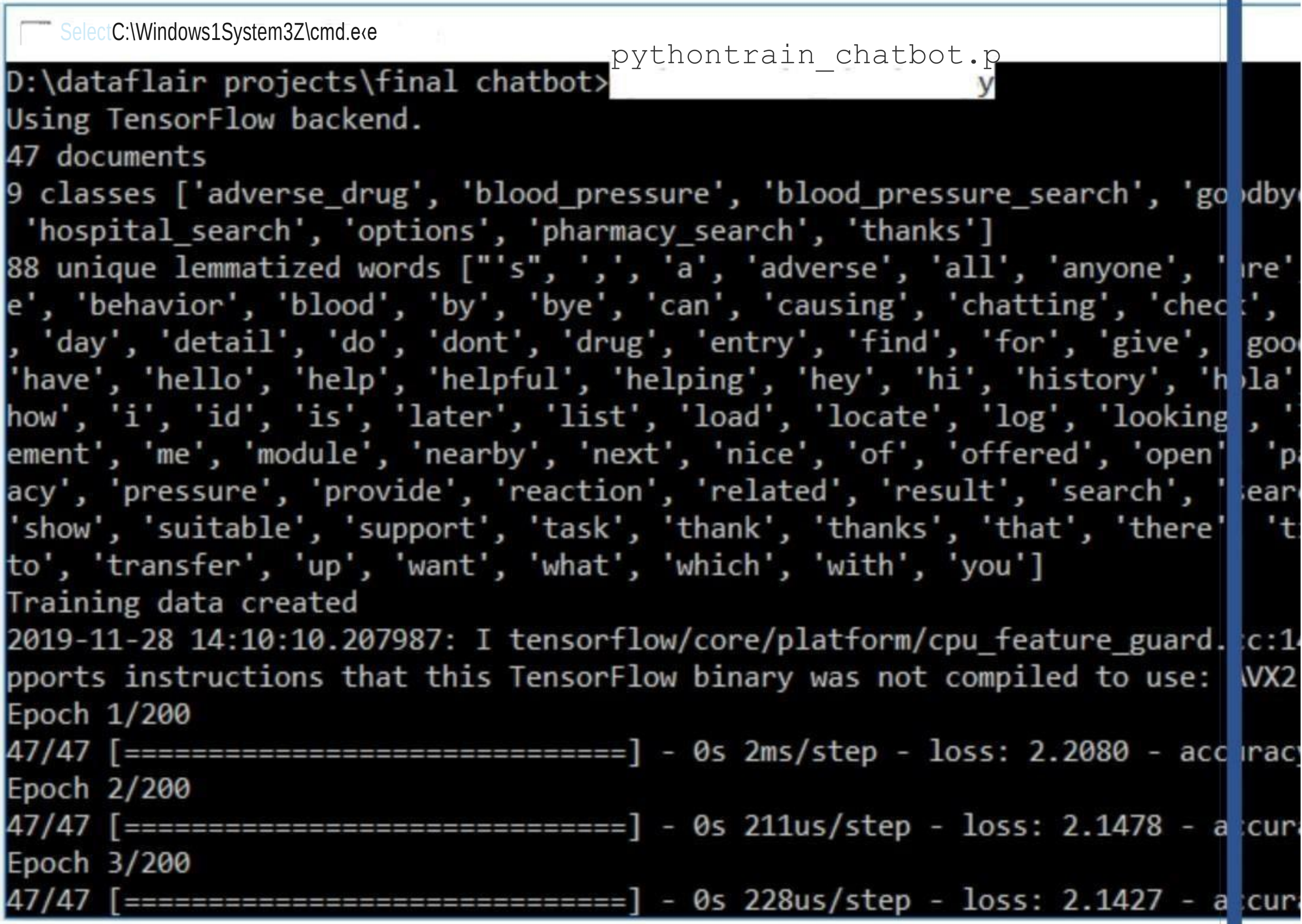
```
pythontrain_chatbot.py
```

Ifwedon'tseeanyerrorduringtraining,wehavesuccessfullycreatedthemodel.Thento run theapp,werunthesecondfile.

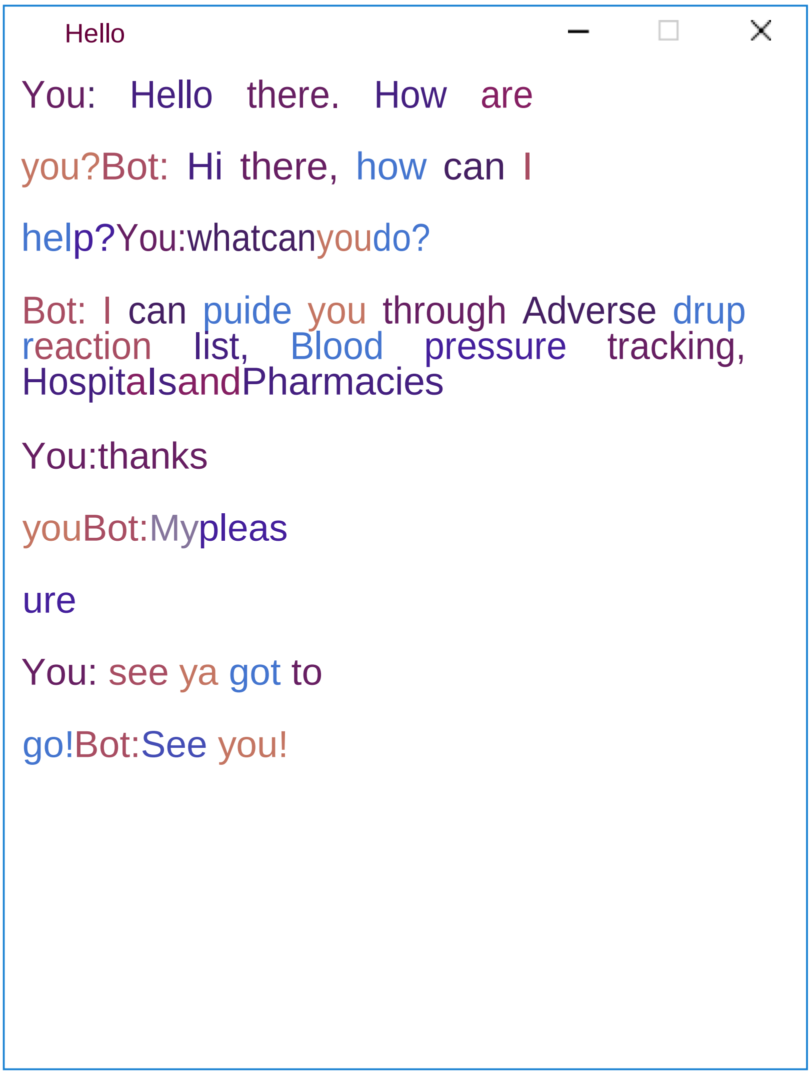
```
pythonchatgui.py
```

Theprogramwill open up a GUIwindowwithin a fewseconds.WiththeGUIyoucaneasily chatwiththebot.

Screenshots:



```
D:\dataflair projects\final chatbot>pythontrain_chatbot.py
Using TensorFlow backend.
47 documents
9 classes ['adverse_drug', 'blood_pressure', 'blood_pressure_search', 'goodbye
 'hospital_search', 'options', 'pharmacy_search', 'thanks']
88 unique lemmatized words ["'s", ',', 'a', 'adverse', 'all', 'anyone', 'are'
e', 'behavior', 'blood', 'by', 'bye', 'can', 'causing', 'chatting', 'check',
, 'day', 'detail', 'do', 'dont', 'drug', 'entry', 'find', 'for', 'give', 'good
'have', 'hello', 'help', 'helpful', 'helping', 'hey', 'hi', 'history', 'hola'
how', 'i', 'id', 'is', 'later', 'list', 'load', 'locate', 'log', 'looking', '
ement', 'me', 'module', 'nearby', 'next', 'nice', 'of', 'offered', 'open', 'pa
acy', 'pressure', 'provide', 'reaction', 'related', 'result', 'search', 'searc
'show', 'suitable', 'support', 'task', 'thank', 'thanks', 'that', 'there', 't
to', 'transfer', 'up', 'want', 'what', 'which', 'with', 'you']
Training data created
2019-11-28 14:10:10.207987: I tensorflow/core/platform/cpu_feature_guard.cc:1
pports instructions that this TensorFlow binary was not compiled to use: AVX2
Epoch 1/200
47/47 [==============================] - 0s 2ms/step - loss: 2.2080 - accuracy
Epoch 2/200
47/47 [==============================] - 0s 211us/step - loss: 2.1478 - accura
Epoch 3/200
47/47 [==============================] - 0s 228us/step - loss: 2.1427 - accura
```



You: Hello there. How are you?Bot: Hi there, how can I help?You:whatcanyoudo?

Bot: I can puide you through Adverse drup reaction list, Blood pressure tracking, HospitalsandPharmacies

You:thanks youBot:Mypleasure

You: see ya got to go!Bot:See you!

```
PYTHONCODE
importre
importlong_responseaslong


defmessage_probability(user_message,recognised_words,single_response=False,required_words=[]):
    message_certainty =
    0has_required_words=True

    #Countshowmanywordsarepresentineachpredefinedmessageforword
    inuser_message:
        ifwordinrecognised_words:messag
            e_certainty+=1

    # Calculates the percent of recognised words in a user
    messagepercentage=float(message_certainty)/float(len(recognised_word
    s))

    #Checksthattherequiredwordsareinthestringforwordinre
    quired_words:
        if word not in
            user_message:has_required_w
            ords=Falsebreak

    #Musteitherhavetherequiredwords,orbeasingleresponseifhas_requi
    red_wordsorsingle_response:
        return int(percentage *
    100)else:
        return0


defcheck_all_messages(message):hig
    hest_prob_list={}

    #Simplifiesresponsecreation/addsittothedict
    defresponse(bot_response,list_of_words,single_response=False,required_words=[]):nonlocalhi
        ghest_prob_list
        highest_prob_list[bot_response]=message_probability(message,list_of_words,single_respo
nse,required_words)

    #Responses  --------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------
    response('Hello!',['hello','hi','hey','sup','heyo'],single_response=True)response('Seeyou!
    ',['bye','goodbye'],single_response=True)
    response('I\'mdoingfine,andyou?',['how','are','you','doing'],required_words=['how'])
    response('You\'rewelcome!',['thank','thanks'],single_response=True)
    response('whatanexcitingpersonality!,ilovehim,somuch.',['do','u','love','samir'],required_
words=['love'])

    response('yes,heisyourprojectguide',['shubham'],required_words=['shubham'])response('yes,5
    43incblock',['cabin'],required_words=['cabin'],)
    response('yetowopunyaaatmahjo,aapsabseadhikpadhtihbutaapseadhikconfusedh',['shruti'],requi
red_words=['shruti'])
    response('yes,heisyourreviewer',['p'],required_words=['p'])

    response(long.R_ADVICE, ['give', 'advice'],
    required_words=['advice'])response(long.R_EATING,['what','you','eat'],required_words=['you
    ','eat'])

    best_match=max(highest_prob_list,key=highest_prob_list.get)#pr
    int(highest_prob_list)
    #print(f'Bestmatch={best_match}|Score:{highest_prob_list[best_match]}')return

    long.unknown()ifhighest_prob_list[best_match]<1elsebest_match


# Used to get the
responsedefget_response(user
_input):
    split_message=re.split(r'\s+|[,;?!.-
    ]\s*',user_input.lower())response=check_all_messages(split_message)
    returnresponse


#Testingtheresponsesystemwhi
leTrue:
```

```
print('Bot:'+get_response(input('You:')))
```