**A PROJECT REPORT**

**ON**

**CHATBOT USING DEEP LEARNING**

Submitted in partial fulfillment for the requirement of the award of

Internship

IN

Data Analytics, Machine Learning and AI using Python



*Submitted By*

**KISHORE RAJU** (SASTRA Deemed To Be University)

*Under the guidance of*

**Mr. Bipul Shahi**

**ABSTRACT**

Dialogue Generation or Intelligent Conversational Agent development using Artificial Intelligence or Machine Learning technique is an interesting problem in the field of Natural Language Processing. In many research and development projects, they are using Artificial Intelligence, Machine Learning algorithms and Natural Language Processing techniques for developing conversation/dialogue agent. Their research and development is still under progress and under experimentation. Dialogue/conversation agents are are predominately used by businesses, government organizations and non-profit organizations.

They are frequently deployed by financial organizations like bank, credit card companies, businesses like online retail stores and start-ups. These virtual agents are adopted by businesses ranging from very small start-ups to large corporations. There are many chatbot development frameworks available in market both code based and interface based. But they lack the flexibility and usefulness in developing real dialogues. Among popular intelligent personal assistants includes Amazon’s Alexa, Microsoft’s Cortana and Google’s Google Assistant. The functioning of these agents are limited, are retrieval based agent and also they are not aimed at holding conversations which emulate real human interaction. Among current chatbots, many are developed using rule based techniques, simple machine learning algorithms or retrieval based techniques which do not generate good results.

In this project, I have developed intelligent conversational agent using state of the art techniques proposed in recently published research papers. For developing intelligent chatbot, In this Python project with source code, we are going to build a chatbot using deep learning techniques. The chatbot will be trained on the dataset which contains categories (intents), pattern and responses. We use a special recurrent neural network (LSTM) to classify which category the user’s message belongs to and then we will give a random response from the list of responses. I create a retrieval based chatbot using NLTK, Keras, Python, etc.

**INTRODUCTION**

#### A chatbot is an intelligent piece of software that is capable of communicating and performing actions similar to a human. Chatbots are used a lot in customer interaction, marketing on social network sites and instantly messaging the client. There are two basic types of chatbot models based on how they are built; Retrieval based and Generative based models.

#### 1. Retrieval based Chatbots

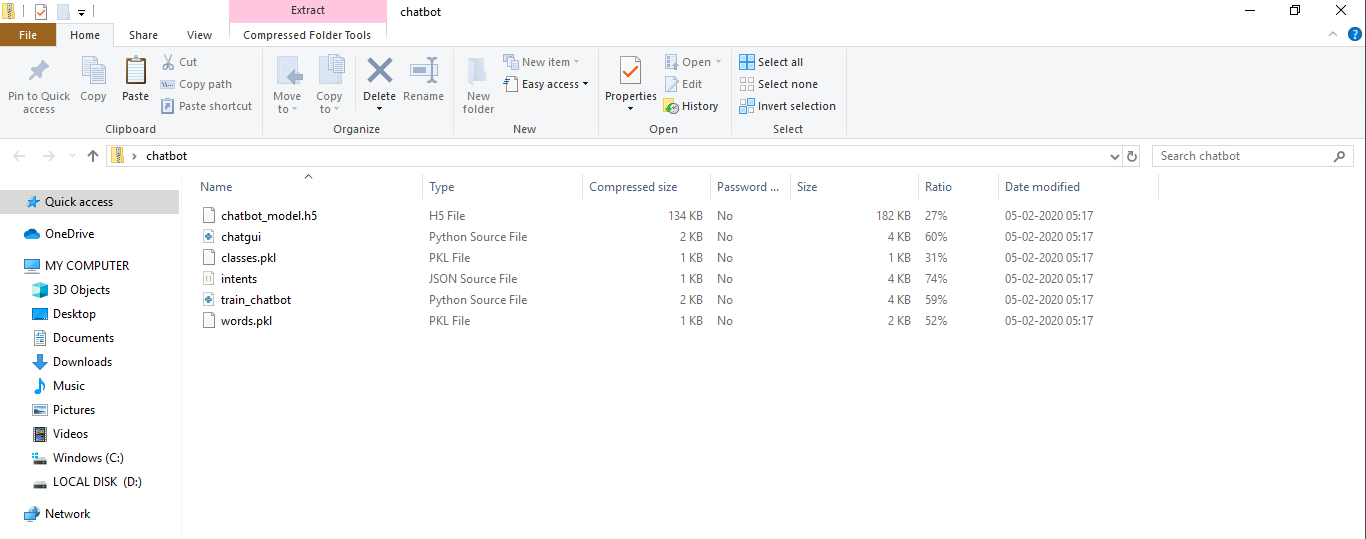
A retrieval-based chatbot uses predefined input patterns and responses. It then uses some type of heuristic approach to select the appropriate response. It is widely used in the industry to make goal-oriented chatbots where we can customize the tone and flow of the chatbot to drive our customers with the best experience.

#### 2. Generative based Chatbots

Generative models are not based on some predefined responses. They are based on seq 2 seq neural networks. It is the same idea as machine translation. In machine translation, we translate the source code from one language to another language but here, we are going to transform input into an output. It needs a large amount of data and it is based on Deep Neural networks.



**PROJECT FILES:**



* **Intents.json –** The data file which has predefined patterns and responses.
* **train\_chatbot.py –** In this Python file, we wrote a script to build the model and train our chatbot.
* **Words.pkl –** This is a pickle file in which we store the words Python object that contains a list of our vocabulary.
* **Classes.pkl –** The classes pickle file contains the list of categories.
* **Chatbot\_model.h5 –** This is the trained model that contains information about the model and has weights of the neurons.
* **Chatgui.py –** This is the Python script in which we implemented GUI for our chatbot. Users can easily interact with the bot.

**ENVIRONMENTAL SETUP:**

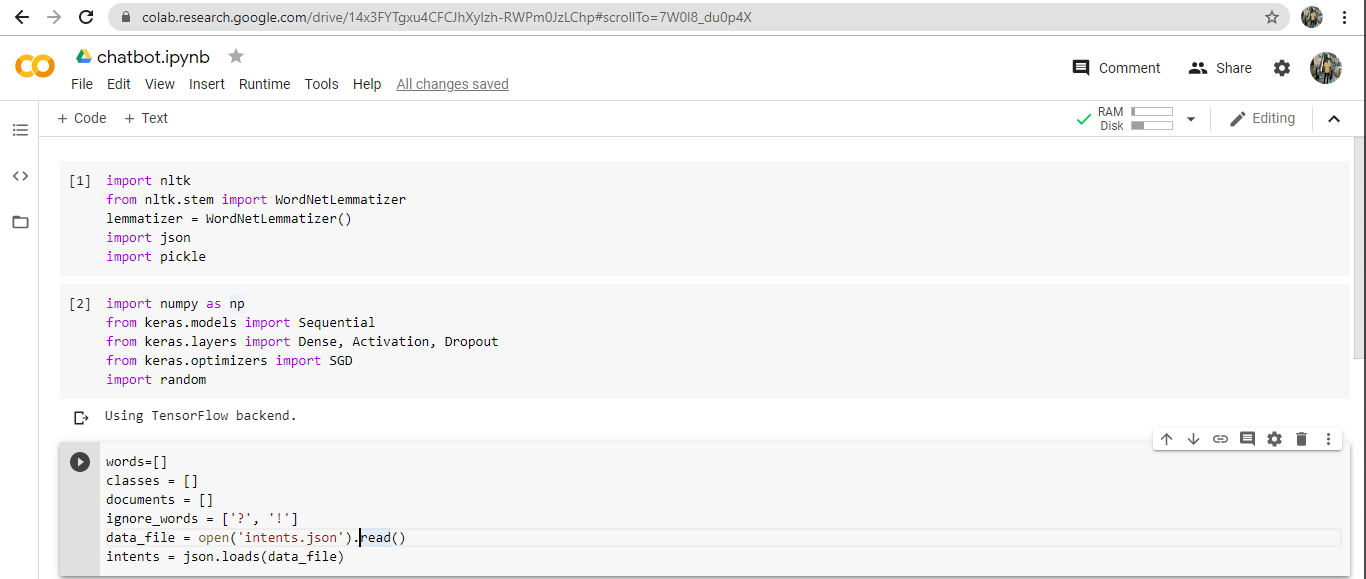
Here are the 5 steps to create a chatbot using Deep Learning:

1. Import and load the data file
2. Preprocess data
3. Create training and testing data
4. Build the model
5. Predict the response

**1. Import and load the data file**

First, make a file name as train\_chatbot.py. We import the necessary packages for our chatbot and initialize the variables we will use in our Python project.

The data file is in JSON format so we used the json package to parse the JSON file into python.



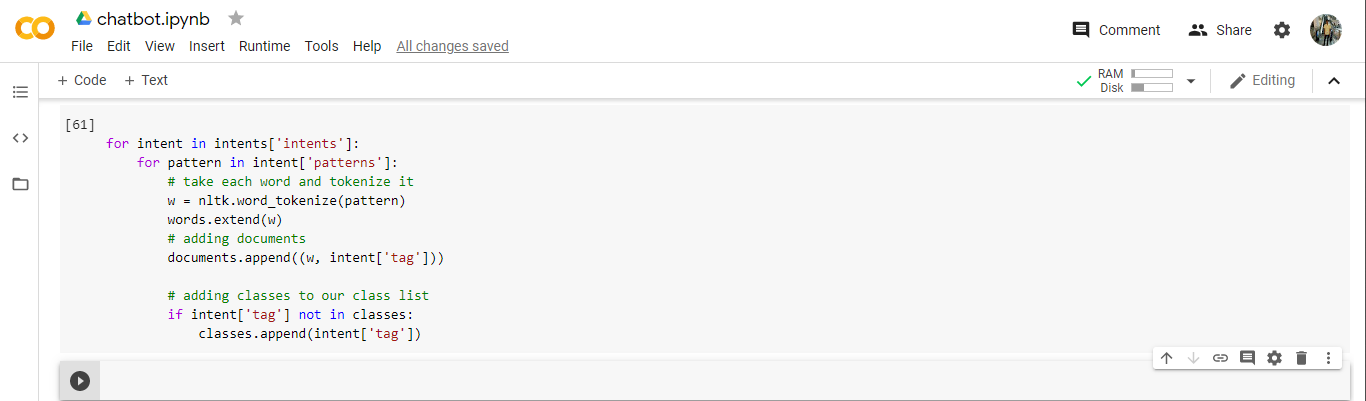
This is how our **intents.json** file looks like.

# 11 b

**2. Preprocess data**

When working with text data, we need to perform various preprocessing on the data before we make a machine learning or a deep learning model. Tokenizing is the most basic and first thing you can do on text data. Tokenizing is the process of breaking the whole text into small parts like words.

Here we iterate through the patterns and tokenize the sentence using nltk.word\_tokenize() function and append each word in the words list. We also create a list of classes for our tags.

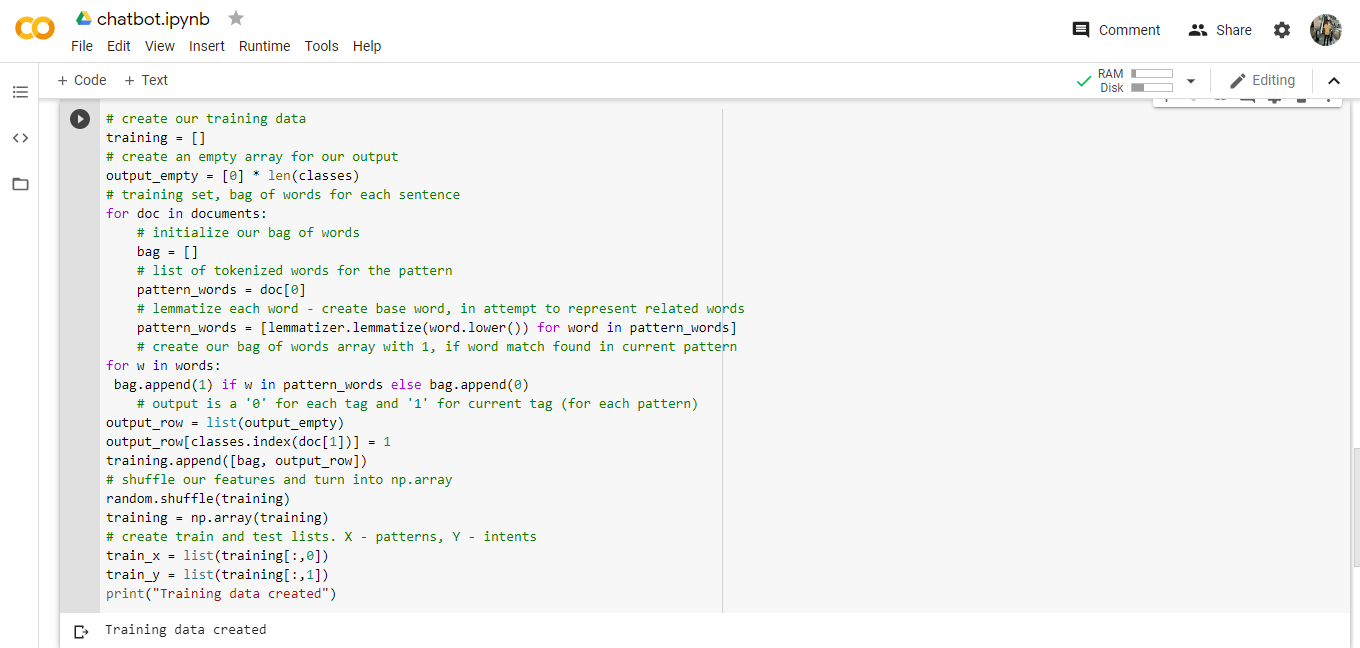


Now we will lemmatize each word and remove duplicate words from the list. Lemmatizing is the process of converting a word into its lemma form and then creating a pickle file to store the Python objects which we will use while predicting.

# 44

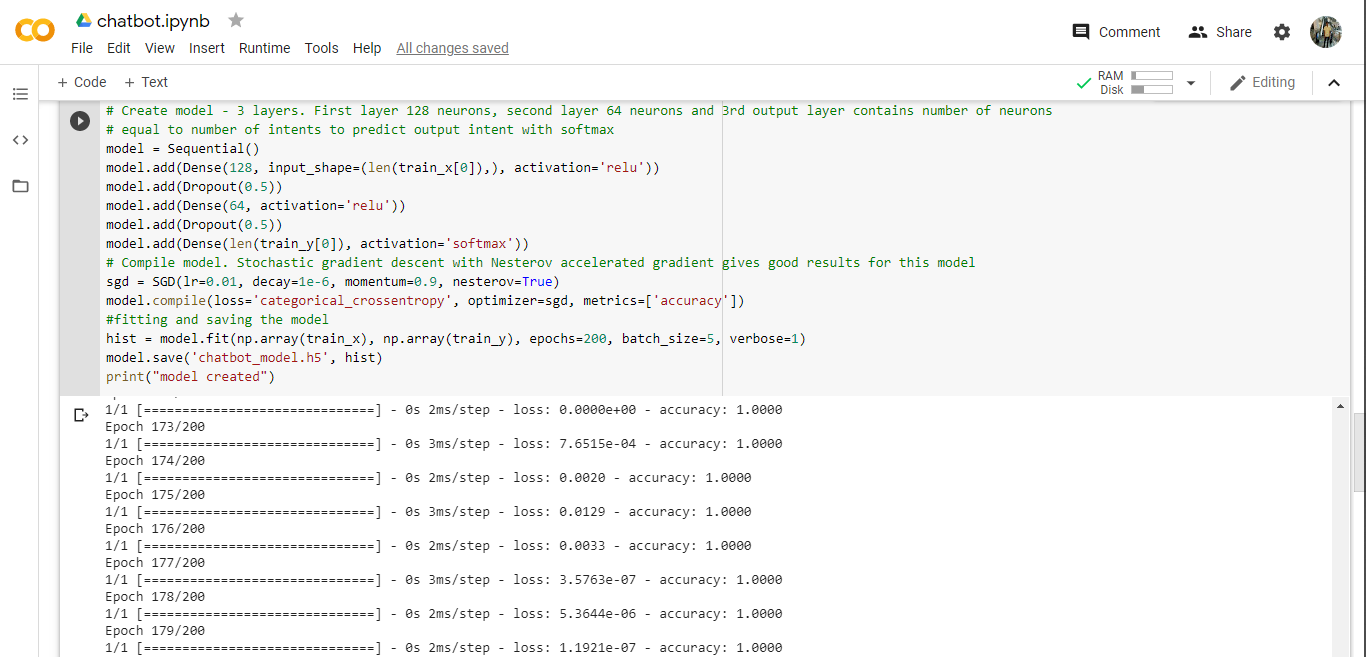
**3. Create training and testing data**

Now, we will create the training data in which we will provide the input and the output. Our input will be the pattern and output will be the class our input pattern belongs to. But the computer doesn’t understand text so we will convert text into numbers.



**4. Build the model**

We have our training data ready, now we will build a deep neural network that has 3 layers. We use the Keras sequential API for this. After training the model for 200 epochs, we achieved 100% accuracy on our model. Let us save the model as ‘chatbot\_model.h5’.



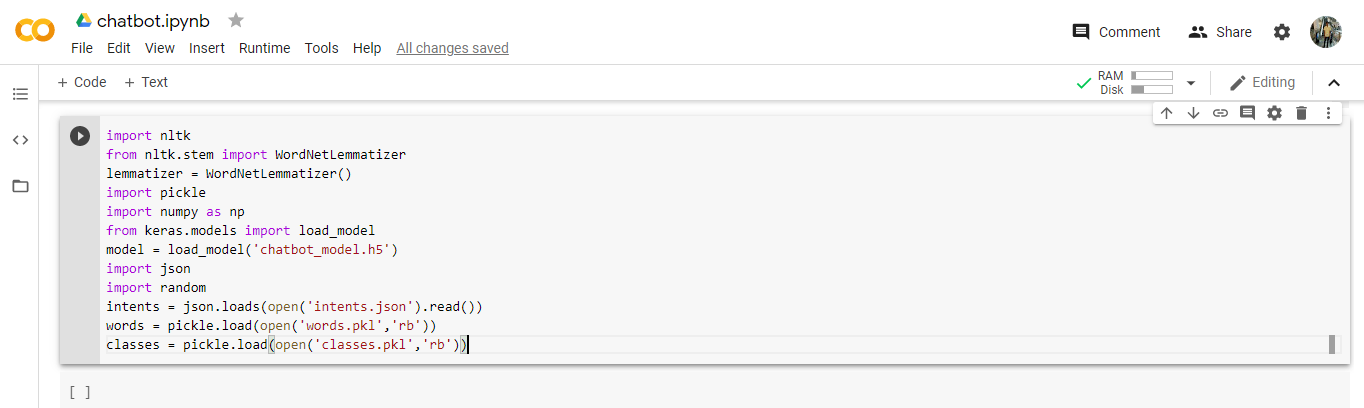
# 77

**5. Predict the response (Graphical User Interface)**

Now to predict the sentences and get a response from the user to let us create a new file ‘chatapp.py’.

We will load the trained model and then use a graphical user interface that will predict the response from the bot. The model will only tell us the class it belongs to, so we will implement some functions which will identify the class and then retrieve us a random response from the list of responses.

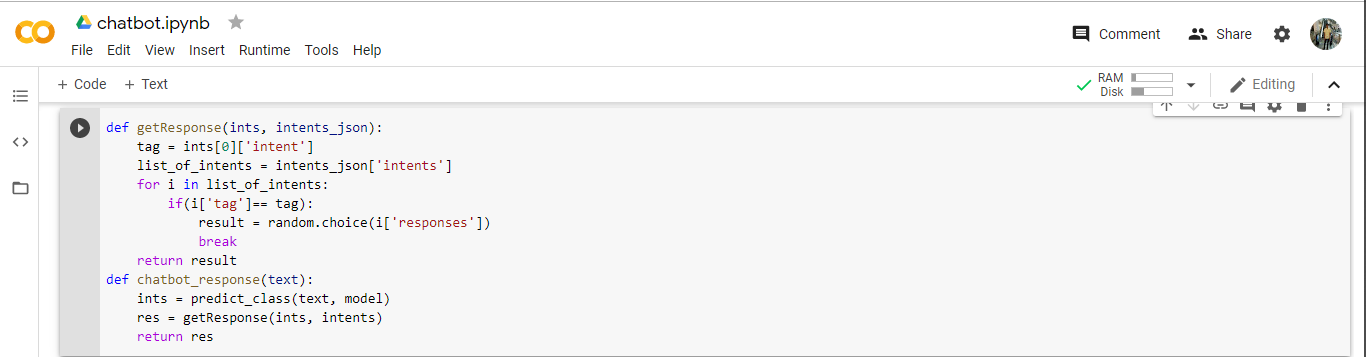
Again we import the necessary packages and load the ‘words.pkl’ and ‘classes.pkl’ pickle files which we have created when we trained our model.



To predict the class, we will need to provide input in the same way as we did while training. So we will create some functions that will perform text preprocessing and then predict the class.



After predicting the class, we will get a random response from the list of intents.



Now we will code a graphical user interface. For this, we use the Tkinter library which already comes in python. We will take the input message from the user and then use the helper functions we have created to get the response from the bot and display it on the GUI. Here is the full source code for the GUI.



**6. Run the chatbot**

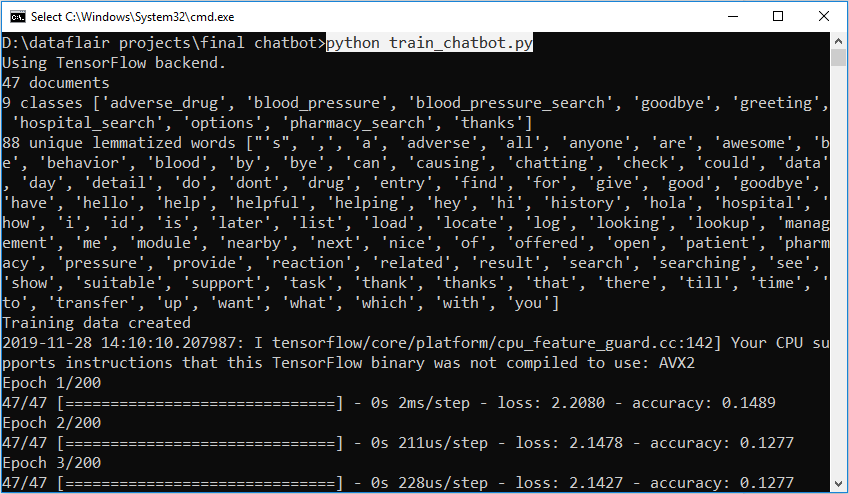
To run the chatbot, we have two main files; **train\_chatbot.py** and **chatapp.py**. First, we train the model using the command in the terminal:

**python train\_chatbot.py**

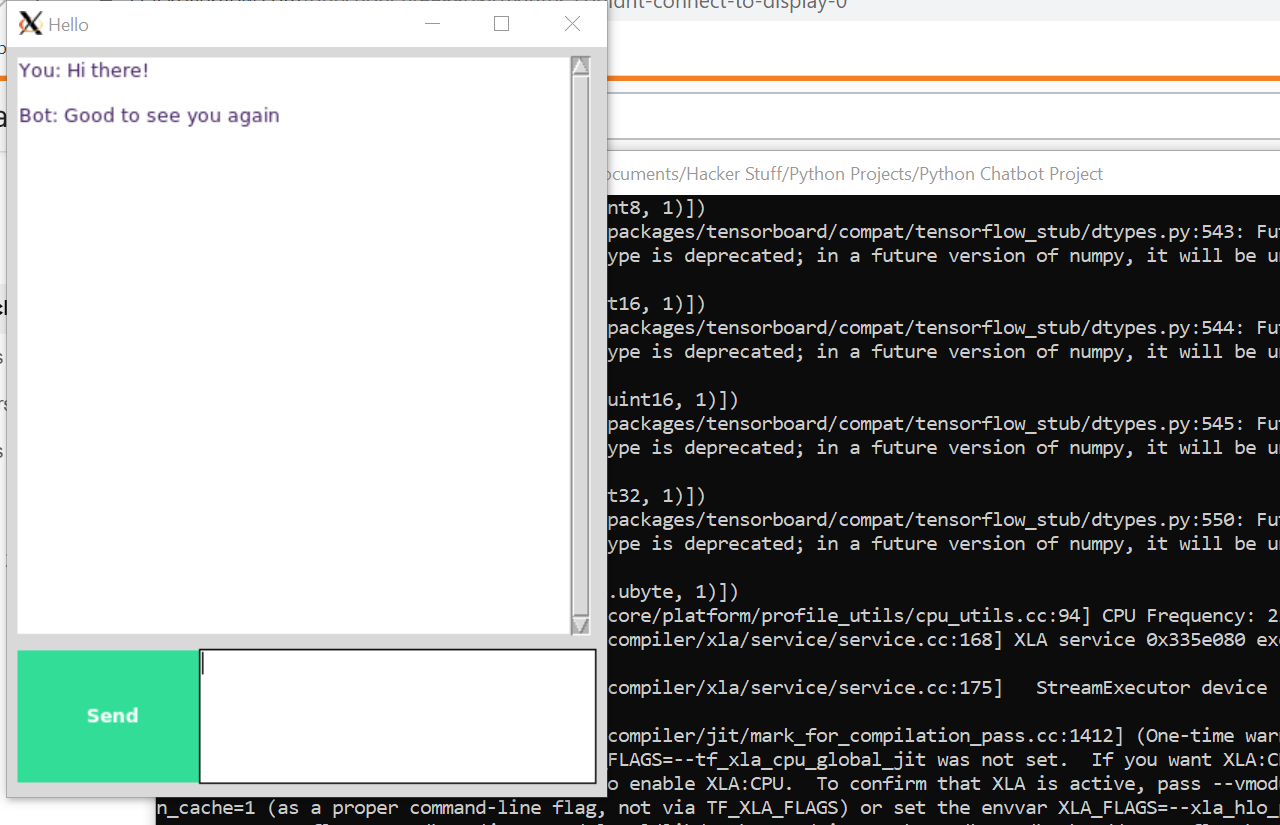
If we don’t see any error during training, we have successfully created the model. Then to run the app, we run the second file.

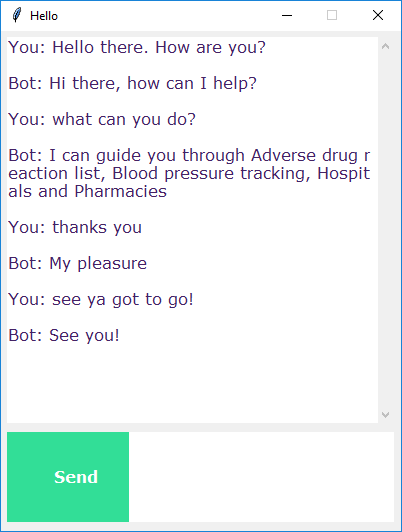
**python chatgui.py**

The program will open up a GUI window within a few seconds. With the GUI you can easily chat with the bot.



**OUTPUT**

****



## CONCLUSION

## In this Python data science project, we understood about chatbots and implemented a deep learning version of a chatbot in Python which is accurate. You can customize the data according to business requirements and train the chatbot with great accuracy. Chatbots are used everywhere and all businesses is looking forward to implementing bot in their workflow.

# RESOURCES

# [Deep Learning](https://machinelearningmastery.com/what-is-deep-learning/) <https://machinelearningmastery.com/what-is-deep-learning/>

# [Natural Language Processing](https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32?gi=2610d4885f73) <https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32>

# Researchpath <https://www.researchgate.net/publication/328582617_Intelligent_Chatbot_using_Deep_Learning>