

**EAI 6010: Applications of Artificial Intelligence**

**Week-6: Project**

**Building a Chatbot**

**Shivam Chauhan**

**&**

**Shubham Pratap Singh**

**March 26th, 2020**

**Introduction**

A bot is a software application that performs automated task and chatbots come under the category of bots that live in various chat platforms. A chatbots special ability is its ability to converse with humans, hence the knowledge of chatting with someone is key for a chatbot.

Chatbots can be classified into two types:

Command based: Chatbots that function on predefined rules and can answer to only limited queries or questions. Usually, users need to select an option to determine their next step.

Intelligent/AI Chatbots: Chatbots that leverage Machine Learning and Natural Language Understanding to understand the user’s language and are intelligent enough to learn from conversations with their users.

Today, users can converse via text, speech or even interact with a chatbot using graphical interfaces.

Chatbots can be found on platforms such as Facebook Messenger, Slack, Telegram, Skype, SMS and even on websites. Each platform has its own noticeable features which determine the possible ways in which the chatbot can interact with the user.

AI is revolutionizing businesses, as chatbots powered by AI are becoming a reasonable customer service option. Because of this, interacting with a brand or a business has become easy for customers.

The purpose of chatbots is to scale business teams and help them in maintaining a cordial relationship with their customers. This helps businesses cut down on operational costs, save time and ensure worthwhile productivity as all the basic and monotonous requests are handled by the chatbot while complex queries are taken care of by the support team.

Chatbots can help customers make e-commerce purchases, answer customer service questions, monitor employee’s or customer’s satisfaction, improve response rate from customers, deliver a personalized experience, derive business intelligence and automate repetitive tasks.

In the development of a chatbot, Natural Language Processing (NLP) plays a vital role. NLP algorithms can process the received text, interpret, deduce, and determine what was meant and then specify a series of appropriate actions.

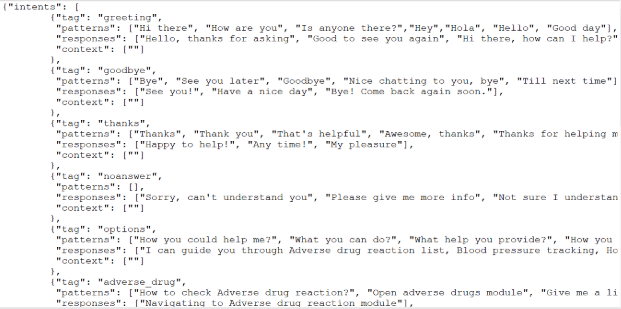
Machine Learning/Deep Learning and Neural Networks are generally used to make chatbots enterprise ready and intelligent enough to leave a lasting impression on the user.

**Analysis**

For this project, we have built a chatbot using deep learning techniques with the help of libraries such as Natural Language Toolkit (NLTK) and Keras in Python. The chatbot is trained on the dataset which contains categories, pattern and responses. Artificial neural network that has been used to classify which category the user’s message belongs to. This helps in choosing a random response from the list of responses.

The Dataset

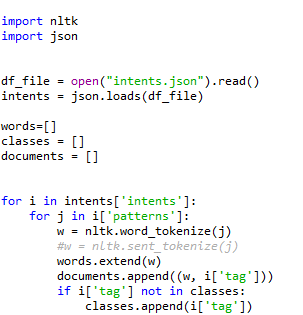
The dataset that we have used is ‘intents.json’. This is a JSON (JavaScript Object Notation) file that contains the patterns that are essential to find and the responses we want to return to the user. Here is our data file:



There are 6 steps that are required in order to create a chatbot from scratch:

1. Importing and loading the data file
2. Preprocessing the data
3. Creating training data
4. Building the model
5. Predicting the response
6. Testing our model

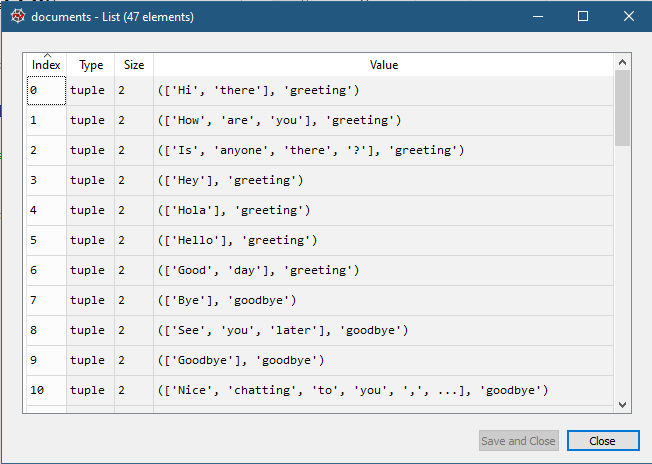
Importing and loading the data file



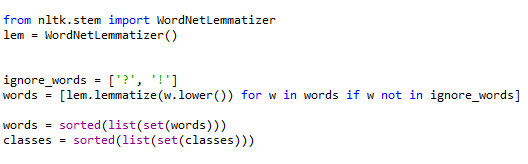
First, we will import the NLTK library by writing import nltk. JSON library is used to handle json file.

After loading the data, we created a list of all the words classes. All the words are tokenized. Another list ‘document’ is created which contain both, the words and their classes/intent with respect to that word.

This is our document list.



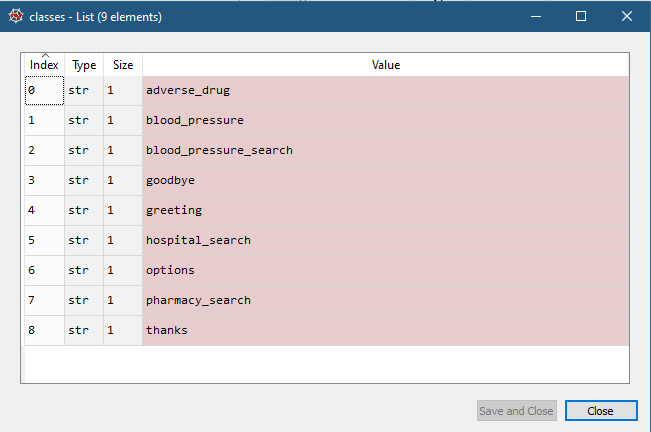
Preprocessing the data



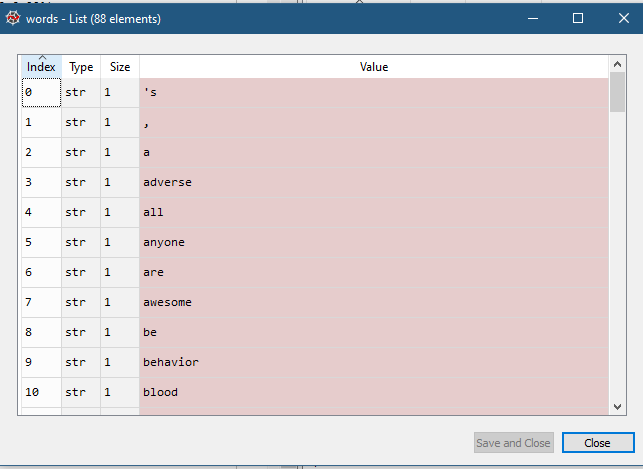
In above codes we are preprocessing the data. Following things are done:

* We have cleaned the data by removing special characters like ‘?’ and ‘!’ which are there in our raw data. These words do not have a meaning when we train the data because every question will have ‘?’, which can lead to wrong predictions. So, technically we are removing the bias.
* Before lemmatizing the words, we have transformed all the words to lower case. This is done because machine will read the words for example, ‘Hello’ and ‘hello’ as a two different word.
* We lemmatize the words, this mean we have converted a word to its base form. We have used lemmatization over stemming because we wanted the converted words to have meaning. Stemming may or may not convert to a base form which has a meaning.
* After converting all the words in ‘words’ list, we have sorted them in an order for ease of understanding later. We have also sorted list ‘classes’.

This is our class list with 9 classes.



This is our words list.



Creating training data

Now we have to create a training data for our Artificial Neural Network. Our input is the words/sentences and the out are the class/intent it belongs to. But the machine doesn’t understand text so we will convert text into numbers.

We have created a list which has all the words. Now we will convert this list to an array for each word. This will be done by, if the sentence has those words, we will prepare a list which will show 1 for all those words and rest all will be 0.

**Explanation of working of this model with an example**

**For example: let say our all the words list is [‘hello’, ‘hi’, ‘bye’, ‘thank’, ‘you’]**

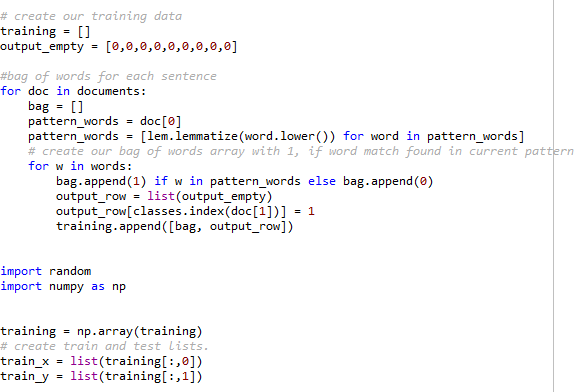
**And our classes list is [‘greeting’, ’goodbye’, ‘thanks’]**

**So, if the input text is ‘Thank you’. Our input for model will be a list [0,0,0,1,1].**

**Our output of model will be [0,0,1]**

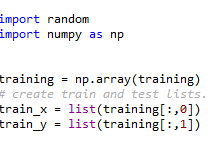
**Now our model will understand that it has to give a response from goodbye class. It will select a random response from the goodbye class in our data.**

So, we are creating a Bag of word model which count the number if the word is there in a sentence which is in our all the words list.

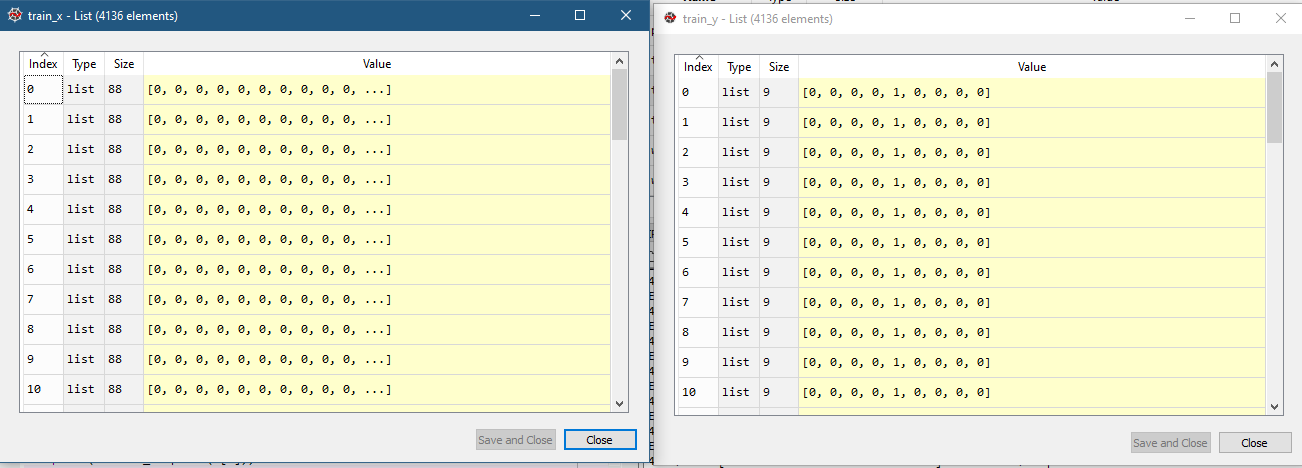


In above codes, we are matching our words list with document list(after applying the same data preprocessing to document list).

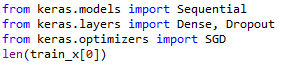
We are creating bag of word list for each index i.e each sentence and creating output list.



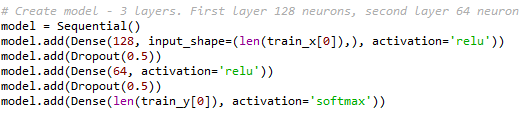
Now we are converting our list to NumPy array. Our input variable is the first column and we have saved it as train\_x. Our output column is second column which is saved as train\_y.

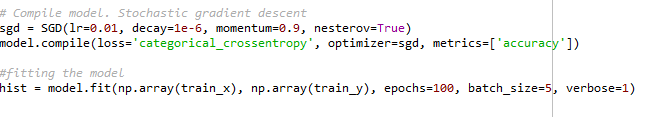


Building the model

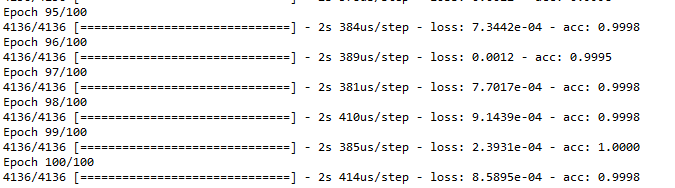


We have used Keras library to train our Neural Network. We will build a deep neural network that has 3 layers. First layer 128 neurons, second layer 64 neurons and 3rd output layer will predict the class. So output will list of 9 values for 9 classes.





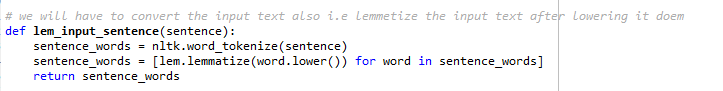
We have used stochastic gradient descent which will send input in batches. We are calculating Crossentrophy loss function and also will be seeing the accuracy with every epochs. Model is trained with 100 epochs with batch size of 5.



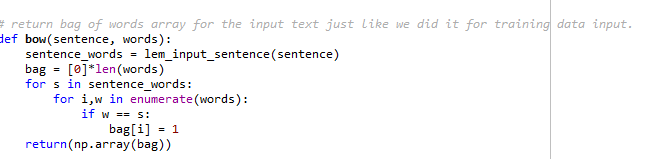
We can see once the model is built, our accuracy is 99.9% and loss function is very close to 0. If we increase the epochs, I feel we can reach accuracy to 100%.

Predicting the response

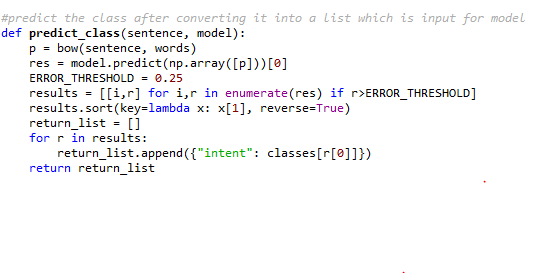
This is most challenging part for our project. Our main goal is to predict the class of the sentence which user has typed. Our model is ready, so now we will do all the preprocessing and data cleaning for the text user is typing. We have to do it for every input, so best way is to build a function for this process.



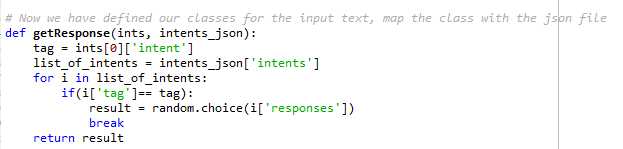
Above code shows the same data cleaning as mentioned above in preprocessing section.



Next thing is to create a bag of word which is the input for our model. Again, we are doing the same like preparing training set. It will use our lem\_input\_sentence() function which will clean and lemmatize the words. Then it will return bag of words array: 0 or 1 for each word in the bag that exists in the sentence.



The next thing is to predict the class of the input text it belongs to. This function takes the input sentence and the model which we have built. It will use our bow() function to create a input list, then it will give an input to the model, model will predict output class. Once the class is predicted (the output is number), we will match the output to the original class and get the text of the class. It will create a list which can use to select response.

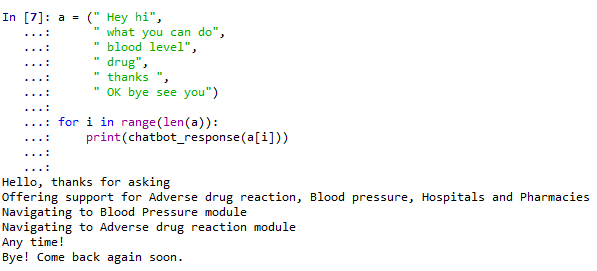


Now we have built a function to get a random response from the file for a class.

First, we will map the tag with the predicted class. Then using random library, we will select a random response from the data file.

Testing our model

I have given a tuple which is user input and iterate the items in the function chatbot\_response() to get a response.



We can see it is working and giving response which are relevant.

Scope of improvement

For making a chatbot, you need a large amount of data in your data file to make it more like interactive. This chatbot is built on a small data which means it will only work if you give somewhat similar input. If chatbots are built on large data, we can make it more interactive. This is for a healthcare industry so you can see when user is giving input like drug, it is responding about the adverse drug reaction module. This shows how it is dependent on data.

Another scope of improvement or I would say next step will be create a GUI to create a pop-up window to accept a input from user and display as a response.

**Conclusion**

From this project, we understood about chatbots and implemented a deep learning of a chatbot in Python which is accurate. We can customize the data according to business requirements and train the chatbot with great accuracy. We can build responses as per the industry like we did it for health care industry where it gives information regarding blood pressure and adverse drug reactive module. Chatbots are used everywhere and all businesses is looking forward to implementing bot in their workflow. This is a great AI application and still have a great scope of research.

*References*

Shaikh, B. (2019, April 11). An Introduction to Chatbots. Retrieved from <https://chatbotslife.com/an-introduction-to-chatbots-8ca7d7ad2eb2>

llSourcell/How\_to\_make\_a\_chatbot. (2020). GitHub. Retrieved 26 March 2020, from <https://github.com/llSourcell/How_to_make_a_chatbot>

Team, D. (2019). Python Chatbot - Learn to build your first chatbot using NLTK & Keras - DataFlair. DataFlair. Retrieved 26 March 2020, from <https://data-flair.training/blogs/python-chatbot-project/>

Advantages and Disadvantages of Chatbots You Need to Know: Blog. (2019, December 5). Retrieved from <https://www.aivo.co/en/blog/advantages-and-disadvantages-of-chatbots-2/>