**DEEP LEARNING CHATBOT**

A Project Report submitted in partial fulfillment for the award of

MBA (Fintech)

**Submitted by**

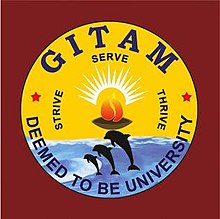
A.BALAJI

Regd No: 121923901002

Under the guidance of

Mr. Divakar Allavarapu

Assistant Professor



**DECLARATION**

We, the undersigned, hereby declare that, the project titled “AI Project report on School Education Management System” submitted to GITAM Institute of Management, GITAM University for the award of the Masters of Business Administration (Fintech) is the original work done by us, under the guidance of Mr. Divakar Allavarapu, Assistant Professor, GITAM Institute of Management. The empirical findings in this report are based on the data collected by us. It was not copied from any other project. The project has not been submitted to any Substitute/University for the award of any Diploma/Degree.

Date: 18-04-2020

Place: Visakhapatnam A.BALAJI (121923901002)

CERTIFICATE

This is to certify that the project entitled, “AI Project report on School Education Management System”, is a bonafide work done by A.BALAJI(121923901002) is submitted in partial fulfilment for the Masters of Business Administration (Fintech) of GITAM University. It has not been submitted for the award of any diploma/degree in any other Institution/University.

Date: 18-04-2020 Mr. Divakar Allavarapu

Place: Visakhapatnam Assistant Professor

**ACKNOWLEDGEMENT**

We thankful to Prof. Sheela, Principal, GITAM Institute of Management, GITAM University, Vishakhapatnam, for giving us an opportunity to learn and have a study on very important topic of management. We are thankful to Mr. Divakar Allavarapu the project guide who helped us and guided us in completing this project. We are immensely thankful to friends and the respondents who have guided us at every stage in preparing and finishing this project with their valuable suggestions.

A.BALAJI (121923901002)

**CONTENTS Page No**

1. Introduction 06
2. Executive Summary 07
3. PREPROCESSING DATA 07-11
4. MODELLING & COMPLING 11-13
5. CHATBOT 14-15
6. Conclusion 15
7. References 15

CHATBOT

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 they are as follows.

1) Retrieval based

2) 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.

**Executive Summary:**

In this project we are going to create a conversation chatbot using NLTK, Keras (LSTM), Python. First we pre-process the data then we will create the model, save the model and predict the out puts of the model and make a function to interact with the chatbot.

**Preprocessing of Data:**

The process of converting data to something a computer can understand is referred to as **pre-processing.**One of the major forms of pre-processing is to filter out useless data.

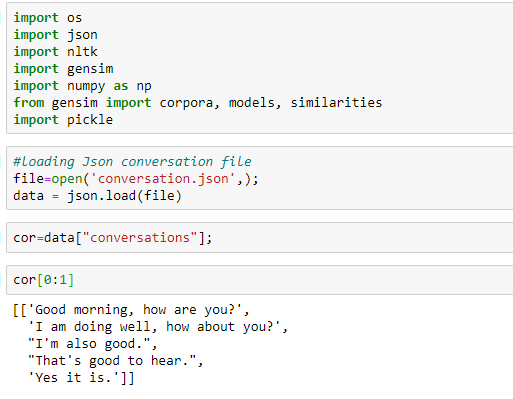
Preprocessing of data is nothing but massaging the data means making the data that is convenient for giving inputs to the neural network.

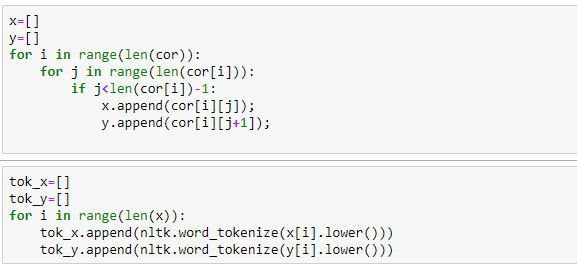
In this I have taken a common conversation dataset in a json format containing conversations.

The imported data contains lot of things that are not useful for example the special characters we have to clean these by using Regex functions.

Then we will do some preprocess process as follows,

**Loading data and Separating QUESTIONS and ANSWERS:**



****

**Tokenization:**

Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences. In the process of tokenization, some characters like punctuation marks are discarded. The tokens become the input for another process like parsing and text mining.

**Removing Stop word:**

In natural language processing, useless words (data), are referred to as stop words.

**Stop Words:** A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

We would not want these words taking up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to be stop words. NLTK (Natural Language Toolkit) in python has a list of stop words stored in 16 different languages. You can find them in the nltk\_data directory.

# Stemming:

In the areas of Natural Language Processing we come across situation where two or more words have a common root. For example, the three words - agreed, agreeing and agreeable have the same root word agree. A search involving any of these words should treat them as the same word which is the root word. So it becomes essential to link all the words into their root word. The NLTK library has methods to do this linking and give the output showing the root word.

# Lemmatization:

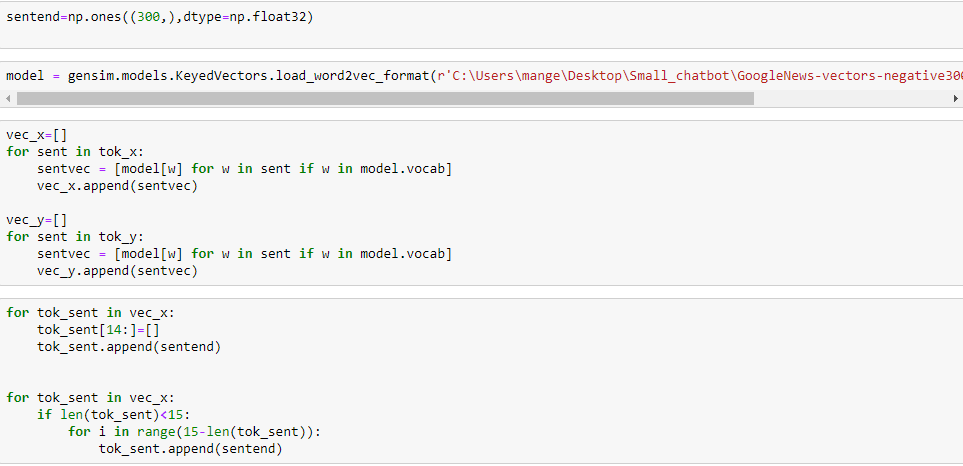
Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meaning to one word.

Text preprocessing includes both [Stemming](https://www.geeksforgeeks.org/introduction-to-stemming/) as well as Lemmatization. Many times people find these two terms confusing. Some treat these two as same. Actually, lemmatization is preferred over Stemming because lemmatization does morphological analysis of the words.

**CODE For tokenization and converting it into Word2Vec form:**

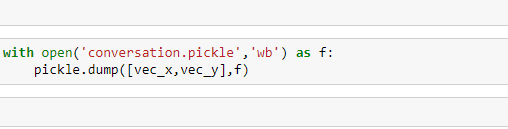
In the below code for converting word 2 vector I taken Google-news-vector which is a pre trained word2vec model

I created sentend variable which is an array containing all ones we will use it to attached that at the end.

****

# 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.

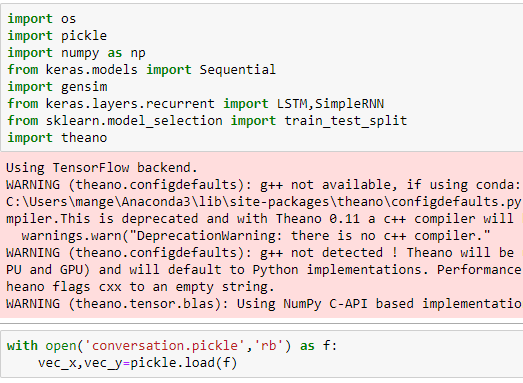


**STEP-2 MODELLING CHATBOT and COMPILING CHAT BOT:**

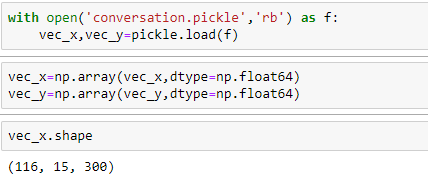
In this section we will create a model

First we read the pickle file which we saved in the first section. Then we will convert it into nuppy array with float dtype

And we will load the relevant libraries



Code for converting list vex\_x, vec\_y into numpy arrays



Know we have create a neural network model using LSTM

**LSTM:**

## Long Short-Term Memory Network

The Long Short-Term Memory network, or LSTM network, is a recurrent neural network that is trained using Backpropagation Through Time and overcomes the vanishing gradient problem.

As such, it can be used to create large recurrent networks that in turn can be used to address difficult sequence problems in machine learning and achieve state-of-the-art results.

Instead of neurons, LSTM networks have memory blocks that are connected through layers.

A block has components that make it smarter than a classical neuron and a memory for recent sequences. A block contains gates that manage the block’s state and output. A block operates upon an input sequence and each gate within a block uses the sigmoid activation units to control whether they are triggered or not, making the change of state and addition of information flowing through the block conditional.

There are three types of gates within a unit:

* **Forget Gate**: conditionally decides what information to throw away from the block.
* **Input Gate**: conditionally decides which values from the input to update the memory state.
* **Output Gate**: conditionally decides what to output based on input and the memory of the block.

Each unit is like a mini-state machine where the gates of the units have weights that are learned during the training procedure.

You can see how you may achieve sophisticated learning and memory from a layer of LSTMs, and it is not hard to imagine how higher-order abstractions may be layered with multiple such layers.

**BELOW IS THE CODE FOR MODELING AND COMPILING:**

model =Sequential()

model.add(LSTM(output\_dim=300,input\_shape=vec\_x.shape[1:],return\_sequences=True,init='glorot\_normal',inner\_init='glorot\_normal', activation='sigmoid'))

model.add(LSTM(output\_dim=300,input\_shape=vec\_x.shape[1:],return\_sequences=True,init='glorot\_normal',inner\_init='glorot\_normal', activation='sigmoid'))

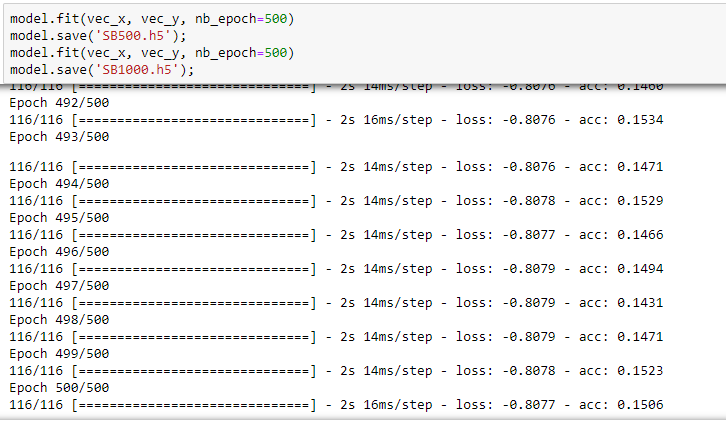
model.add(LSTM(output\_dim=300,input\_shape=vec\_x.shape[1:],return\_sequences=True,init='glorot\_normal',inner\_init='glorot\_normal', activation='sigmoid'))

model.add(LSTM(output\_dim=300,input\_shape=vec\_x.shape[1:],return\_sequences=True,init='glorot\_normal',inner\_init='glorot\_normal', activation='sigmoid'))

Compile code we use loss function cosine proximity and optimizer adam

model.compile(loss='cosine\_proximity', optimizer='adam', metrics=['accuracy'])

Fitting the model:

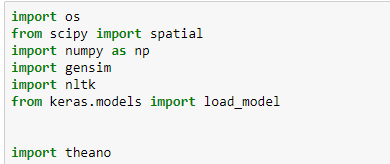


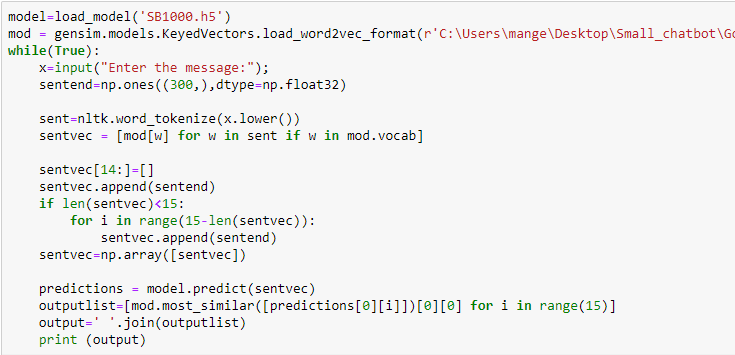
In the above code we run the model with 1000 epoch and we saved the model results are shown above.

**Creating a bot:**

The third step and final step is creating a chatbot nothing but creating a function that is used to give reply for the user

Below is the import required to create a chatbot function

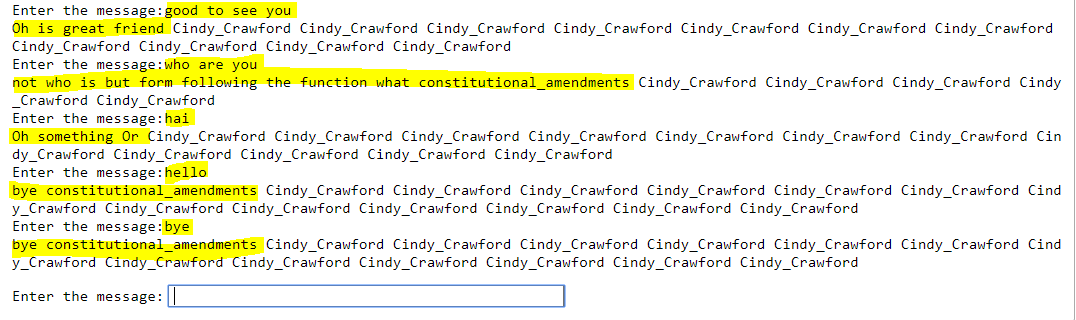




If we see the code clearly the input taken from the user is processed and given the data to the predicted model so that the response can be displayed as output.

But because of the **low accuracy that chatbot did not performed well and the test results are also not good.**

**This is the output snippets of the chatbot:**

****

**CONCLUSION:**

The chat bot was created by using a little conversation data set which contains questions and answers. While creating it I used both NLTK and machine learning algorithms as the model build was not properly designed because of the low accuracy, the chat bot predictions are not good as you can see the output.

**Reference:**

Geeks for geeks

Keras documentation