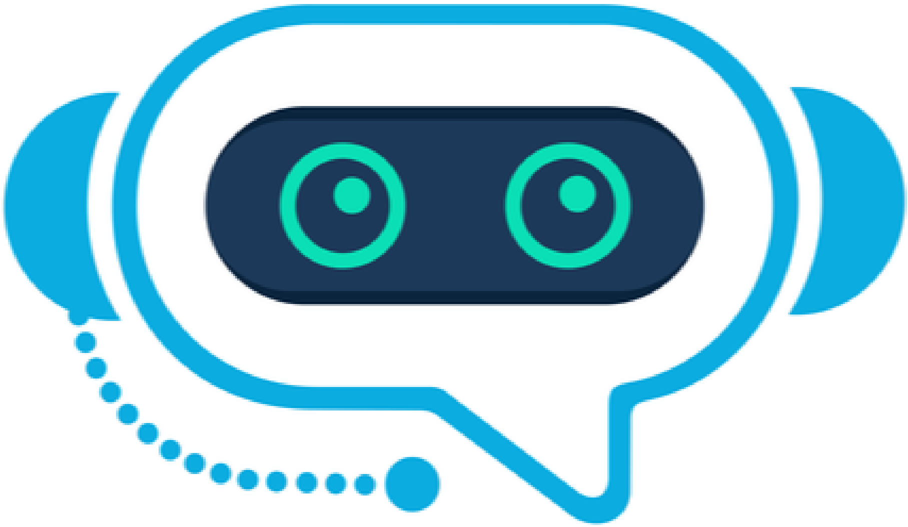


CHATBOT

PROJECT REPORT

SUBMITTED BY:
HITIN SARIN
101134217

Hitinsarin@cmail.carleton.ca

Problem Statement:

Most of the tasks these days are carried out using chatbots. Many companies have now incorporated chatbots for various routine services like customer support, information delivery, etc. Business owners want to stay connected to their clients 24/7 but it is virtually impossible. Either there is any task requiring customer support or finding new leads, business owners try to manage these channels of communication throughout their working day and still lag. The chatbot market size is now projected to grow from $2.6 billion in 2020 to $9.4 billion by 2024 at a compound annual growth rate (CAGR) of 29.7%. Hence, the dependency on AI technology like chatbots is growing remarkably.

Role in current scenario (Covid-19): Chatbots are reasonably working in dispersing the healthcare information in this current coronavirus crises. It is providing a strong prospective towards curated information. It conveys the response to specific questions in an interactive manner, much more efficiently than conventional online search methods.

Problem Analysis:

CHATBOT is an intelligent piece of software that helps in interacting with the customers, marketing on social network sites and promptly messaging the client. There are two basic types of chatbot models based on how they are built:

1. Retrieval based Chatbots - It uses predefined input patterns and responses to answer appropriately. It is commonly used for making goal-oriented chatbots in the industry.

2. Generative based Chatbots - These are based on sequence-to-sequence neural networks. It needs a large amount of data and it is based on Deep Neural networks.

We have built a retrieval oriented chatbot using deep learning techniques. This 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.

Covid19 - The two most authoritative voices of the pandemic, WHO and CDC, have also included chatbots in their websites to provide latest information on the spread of the disease and its symptoms. Many governments are also launching chatbots to provide validated information to their citizens.

Specifications:

This chatbot mainly uses the below libraries:

- **NLTK** - for symbolic and statistical natural language processing for English written.
- **Keras -** to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.
- **Tensorflow** - for dataflow and differentiable programming across a range of tasks.
- **Pickle -** for serializing and de-serializing a Python object structure.
- **Tkinter** – for creating interactive graphical user interface applications.

The software is segregated in parts/classes based on the functionality. The design of this chatbot is mainly divided into 5 parts:

1) Import and load the data file

Firstly, a file named as 'train_chatbot.py' was generated in which a code was written in order to train the chatbot. Then all the necessary packages were imported like 'nltk', 'keras', 'json' and 'pickle' for the chatbot and the variables were initialized.

2) Preprocess data

Before making a deep learning model, whole data was pre-processed by breaking it into words. A function named nltk.word tokenize() was used as well as each word in the words list was appended. A list of classes for our tags was also created. After sorting each word, the duplicate words were removed from the list created above. After all the words got converted into its lemma form, a pickle file was created to store the objects which were used later while predicting.

3) Create training and testing data

After preprocessing the data, training data was created in which both the input and the output was provided. Input is the pattern and output are the class to which the input pattern belongs.

4) Build the model

After training the dataset, a deep neural network was built which was divided into three layers (Keras sequential API was used for this). After training the model for 200 epochs, 100% accuracy was achieved on model.

5) Predict the response (Graphical User Interface)

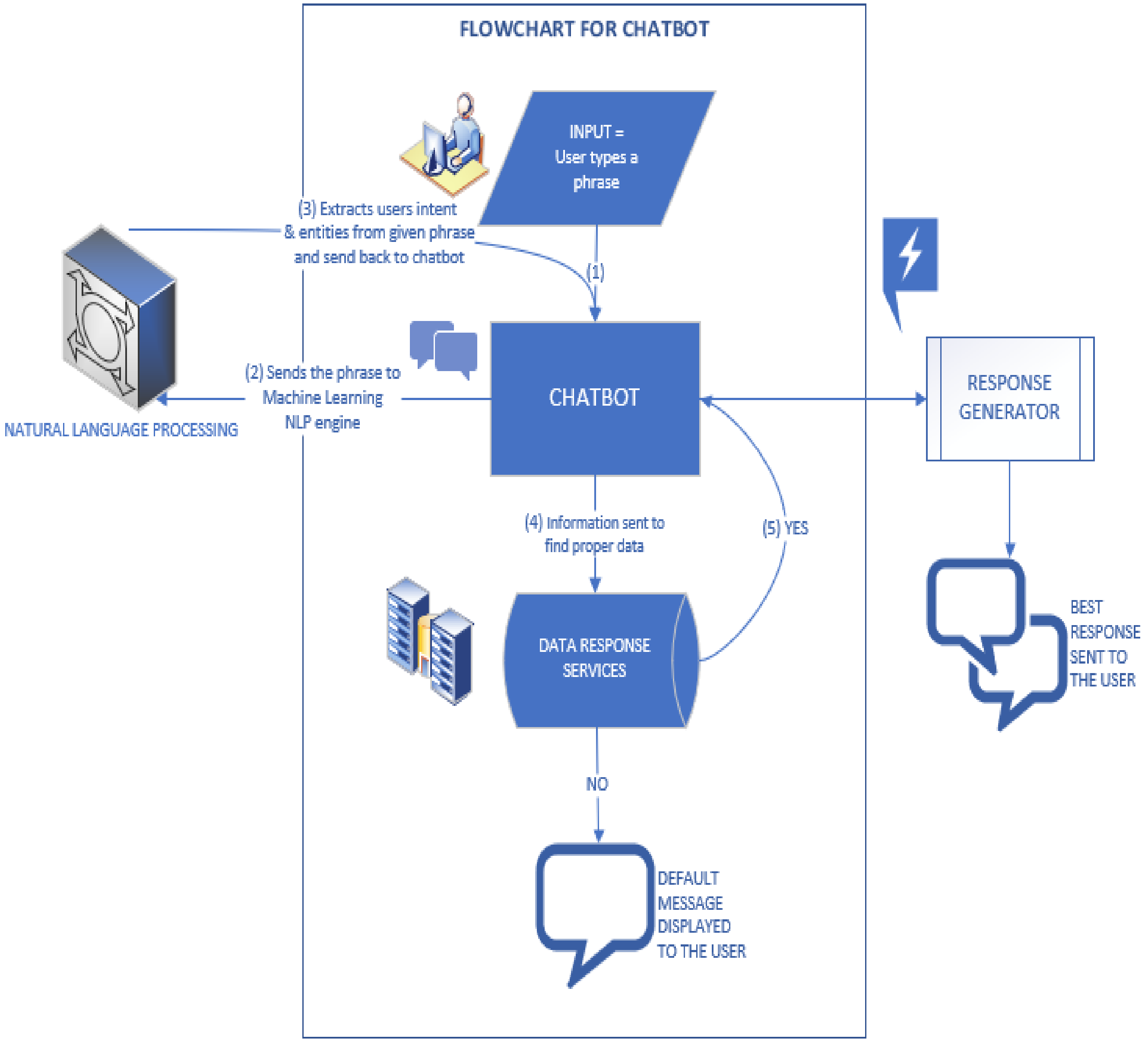
Then, a graphical user interface that predicts the response from the bot was created. The functions to identify the class were implemented and then retrieved a random response from the list of responses. Later, GUI was created using Tkinter library (inbuilt in python) in order to start the interaction with the bot.

6) Run the chatbot

The model was trained by running the train_chatbot.py file. Once, the bot was trained, chatgui.py file was executed to interact with the bot using GUI window.

Design:

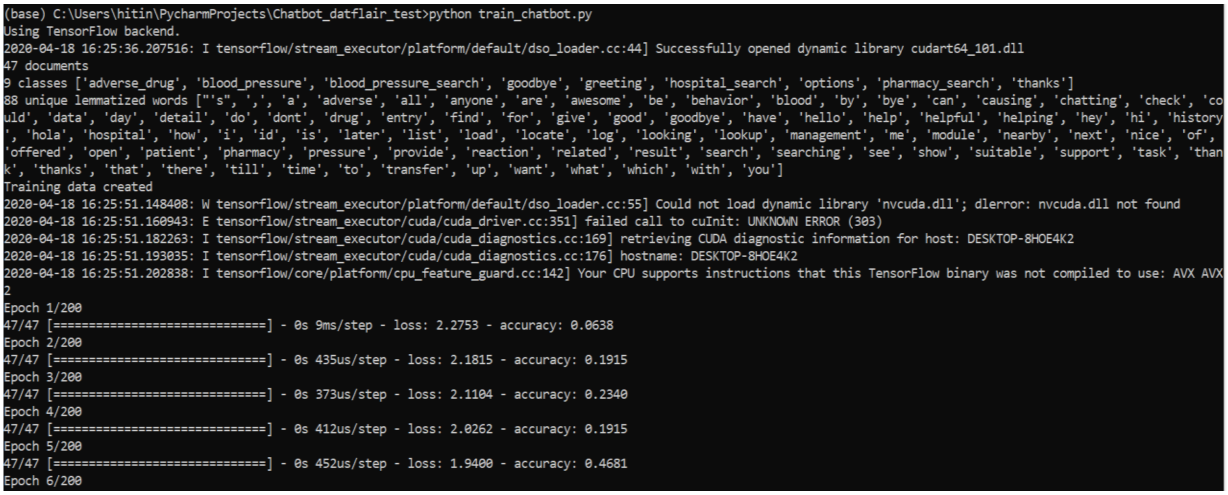
Following is the flowchart stating the Working Mechanism of Retrieval-based Chatbot. The diagram explains how the input from user is processed, retrieved and the best suited response is generated.


FLOWCHART FOR CHATBOT
INPUT = User types a phrase
(1)
(3) Extracts users intent & entities from given phrase and send back to chatbot
CHATBOT
(2) Sends the phrase to Machine Learning NLP engine
NATURAL LANGUAGE PROCESSING
RESPONSE GENERATOR
BEST RESPONSE SENT TO THE USER
(4) Information sent to find proper data
(5) YES
DATA RESPONSE SERVICES
NO
DEFAULT MESSAGE DISPLAYED TO THE USER

Testing:

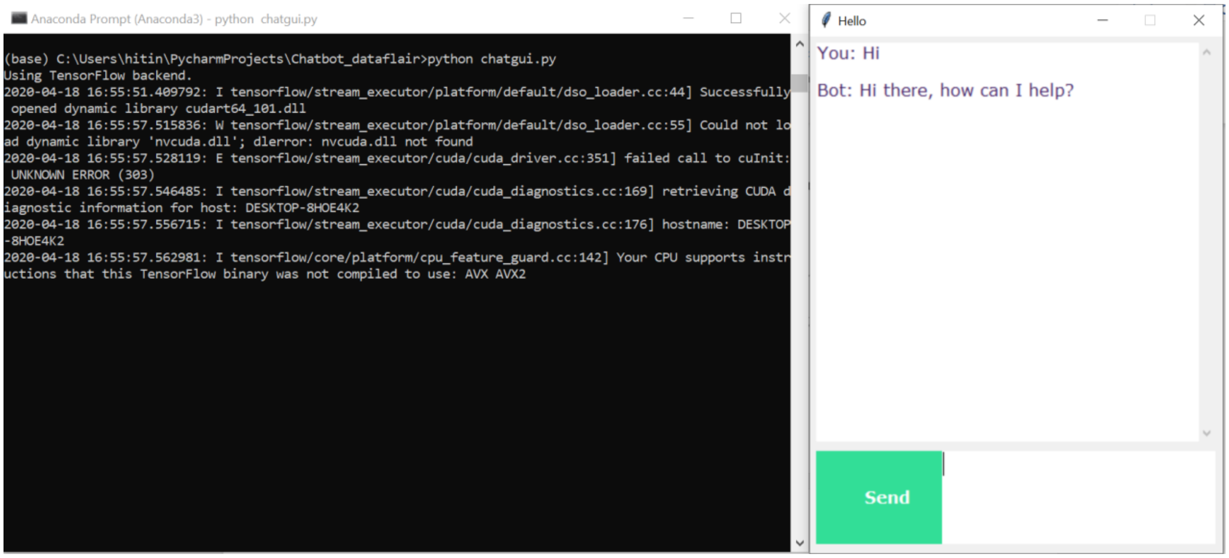
Let us now run all the functionalities one by one and test for some real time cases.

1) After running ‘python train_chatbot.py’ we were successful in training the model.



```
(base) C:\Users\hitin\PycharmProjects\Chatbot_datflair_test>python train_chatbot.py
Using TensorFlow backend.
2020-04-18 16:25:36.207516: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library cudart64_101.dll
47 documents
9 classes ['adverse_drug', 'blood_pressure', 'blood_pressure_search', 'goodbye', 'greeting', 'hospital_search', 'options', 'pharmacy_search', 'thanks']
88 unique lemmatized words ["'s", ',', 'a', 'adverse', 'all', 'anyone', 'are', 'awesome', 'be', 'behavior', 'blood', 'by', 'bye', 'can', 'causing', 'chatting', 'check', 'co
uld', 'data', 'day', 'detail', 'do', 'dont', 'drug', 'entry', 'find', 'for', 'give', 'good', 'goodbye', 'have', 'hello', 'help', 'helpful', 'helping', 'hey', 'hi', 'history
', 'hola', 'hospital', 'how', 'i', 'id', 'is', 'later', 'list', 'load', 'locate', 'log', 'looking', 'lookup', 'management', 'me', 'module', 'nearby', 'next', 'nice', 'of',
'offered', 'open', 'patient', 'pharmacy', 'pressure', 'provide', 'reaction', 'related', 'result', 'search', 'searching', 'see', 'show', 'suitable', 'support', 'task', 'than
k', 'thanks', 'that', 'there', 'till', 'time', 'to', 'transfer', 'up', 'want', 'what', 'which', 'with', 'you']
Training data created
2020-04-18 16:25:51.148408: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2020-04-18 16:25:51.160943: E tensorflow/stream_executor/cuda/cuda_driver.cc:351] failed call to cuInit: UNKNOWN ERROR (303)
2020-04-18 16:25:51.182263: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: DESKTOP-8HOE4K2
2020-04-18 16:25:51.193035: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: DESKTOP-8HOE4K2
2020-04-18 16:25:51.202838: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX
2
Epoch 1/200
47/47 [==============================] - 0s 9ms/step - loss: 2.2753 - accuracy: 0.0638
Epoch 2/200
47/47 [==============================] - 0s 435us/step - loss: 2.1815 - accuracy: 0.1915
Epoch 3/200
47/47 [==============================] - 0s 373us/step - loss: 2.1104 - accuracy: 0.2340
Epoch 4/200
47/47 [==============================] - 0s 412us/step - loss: 2.0262 - accuracy: 0.1915
Epoch 5/200
47/47 [==============================] - 0s 452us/step - loss: 1.9400 - accuracy: 0.4681
Epoch 6/200
```

2) Run ‘python chatgui.py’ to launch GUI of chatbot to chat with the bot.

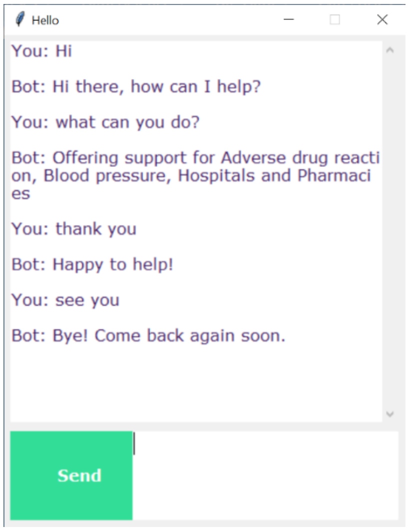


```
(base) C:\Users\hitin\PycharmProjects\Chatbot_dataflair>python chatgui.py
Using TensorFlow backend.
2020-04-18 16:55:51.409792: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully
 opened dynamic library cudart64_101.dll
2020-04-18 16:55:57.515836: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not lo
ad dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2020-04-18 16:55:57.528119: E tensorflow/stream_executor/cuda/cuda_driver.cc:351] failed call to cuInit:
 UNKNOWN ERROR (303)
2020-04-18 16:55:57.546485: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA d
iagnostic information for host: DESKTOP-8HOE4K2
2020-04-18 16:55:57.556715: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: DESKTOP
-8HOE4K2
2020-04-18 16:55:57.562981: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instr
uctions that this TensorFlow binary was not compiled to use: AVX AVX2
```

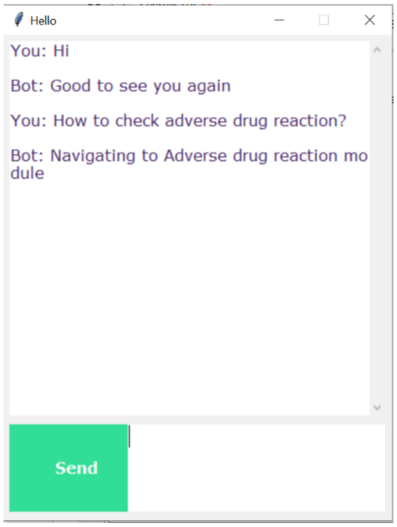
You: Hi

Bot: Hi there, how can I help?

3) Some responses related to the data on which chatbot is being trained:



```
You: Hi

Bot: Hi there, how can I help?

You: what can you do?

Bot: Offering support for Adverse drug reacti
on, Blood pressure, Hospitals and Pharmaci
es

You: thank you

Bot: Happy to help!

You: see you

Bot: Bye! Come back again soon.
```



```
You: Hi

Bot: Good to see you again

You: How to check adverse drug reaction?

Bot: Navigating to Adverse drug reaction mo
dule
```

Impact:

Microsoft invented new Healthcare Bot service which is especially designed for answering queries of public in this COVID-19 pandemic situation. Microsoft worked with the Centers for Disease Control and Prevention for inculcating this new bot service as a self-screening tool for people who are facing the ambiguity of whether they need treatment for COVID-19 or not.

Among all the Microsoft users, Healthcare Bot service is now handling more than one million messages per day approximately from the public concerned about COVID-19 infections [8].

Other impacts are:

- Chatbot designed assists the user to answer a query on a particular subject.
- It automatically reduces the manual botheration and speeds up the work.
- It is a high cost-effective software providing relevant information to the users.

- This software accurately solves problem statement thus making the life of a user a lot easier.

Challenges:

Where chatbots have numerous benefits, there are some deficiencies that must be addressed for maintaining the public's trust in the information provided by the chatbots. For instance, it is possible to receive illogical and contrary responses to the symptom checking through the chatbot hosts for COVID-19. Some chatbots deliver the inconsistent results when presented with identical symptoms. Some chatbots urge the users to get immediate care, while others suggest taking rest at home.

The reason behind this inconsistency could be rooted from a variety of sources for example, some chatbots might rely on only single source of information. In other chatbots, the back-end repository might not be frequently updated from where the chatbots retrieve information in order to respond effectively.

References:

1. https://www.businessinsider.com/chatbot-market-stats-trends
2. https://www.weforum.org/agenda/2020/04/chatbots-covid-19-governance-improved-here-s-how/
3. https://likegeeks.com/python-gui-examples-tkinter-tutorial/
4. https://github.com/tensorflow/tensorflow/issues/22794
5. https://www.youtube.com/watch?v=wypVcNIH6D4
6. https://www.w3schools.com/
7. https://github.com/parulnith/Building-a-Simple-Chatbot-in-Python-using-NLTK
8. https://www.geekwire.com/2020/microsoft-countering-coronavirus-pandemic-chatbots-virus-trackers/