Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2020)
IEEE Xplore Part Number:CFP20K74-ART; ISBN: 978-1-7281-4876-2

Interactive Transport Enquiry with AI Chatbot

Dharani M
3rd Year B.Tech, Department of IT
VR Siddhartha Engineering College
Vijayawada, India
dharanis336@gmail.com

Jyostna JVSL
3rd Year B.Tech, Department of IT
VR Siddhartha Engineering College
Vijayawada, India
Jyostnavsl7@gmail.com

Sucharitha E
3rd Year B.Tech, Department of IT
VR Siddhartha Engineering College
Vijayawada, India
Sucharithaeeda926@gmail.com

Likitha R
3rd Year B.Tech, Department of IT
VR Siddhartha Engineering College
Vijayawada, India
Rajulapatilikitha4@gmail.com

Dr. Suneetha Manne
Professor and Head, Department of IT'
VR Siddhartha Engineering College
Vijayawada, India
Suneethamanne74@gmail.com

Abstract— **Public transportation is used efficiently by millions of people all over the world. People tend to travel to different places and at certain times they may feel completely lost in a new place. Our chatbot comes to rescue at this time. A Chatbot is often described as one of the most promising tools for communication between humans and machines using artificial intelligence. It is a software application that is used to conduct an online chat conversation via text by using natural language processing (NLP) and deep learning techniques. It provides direct contact with a live human agent in the form of GUI. This AI chatbot confirms the current location and the final destination of the user by asking a few questions. It examines the user's query and extracts the appropriate entries from the database. The deep learning techniques that are used in this chatbot are responsible for understanding the user intents accurately to avoid any misconceptions. Once the user's intention has been recognized, the chatbot provides the most relevant response for the user's query request. Then the user gets all the information about the bus names along with their numbers so that the person can travel safely to the desired location. Our chatbot is implemented in python's Keras library and used Tkinter for GUI.**

Keywords—artificial intelligence; chatbot; natural language processing; deep learning; Keras; GUI; Tkinter

I. INTRODUCTION

People often travel to different places and sometimes they may feel completely lost in a new place. They may not know the route and buses to reach their destination. They may face difficulty in asking the people around them to get into the bus which goes to their destination place. You arrive at the bus stop, ready to catch your bus and you don't have any idea which bus to take to reach your destination. This is the main problem we encounter in our daily life for many years while using public transport to reach the desired places. You come to an unknown place and you don't have any idea about which bus to take. There will be no information provided about the buses or about the bus numbers or the bus that takes you to the destination place. This is the outside world scenario when we go to a new place. When you come to the bus stop you will see many people waiting for the bus. Some of them know the exact bus to be taken to reach their destination and some of them are in a dilemma on which bus to be taken. So in that situation, you are completely stressed. Here came to know that most of the people use public transportation for their daily traveling to reach their destination. In that case, many of them don't have any idea of the bus data or about the bus numbers or timings. People face a problem when they are new to the place and don't know which bus to take. In that case, much time is wasted for waiting or for gathering the information about the bus that which they need to get in to reach their destination. There is one solution obtainable for this problem is that an interactive Chatbot that gives the entire data of the buses and their timings which makes it easier for people to use or to communicate at any place or any time. One main thing is that chatbots are essential for customer day to day life.

II. BASIC TERMINOLOGY

A. CHATBOT

A Chatbot is nothing but a software application that helps in conducting a conversation by using auditory or text-based methodologies. The programs in a Chatbot are developed to imitate human conversations. The use of Chatbot's for various

purposes include customer services, request routing, or for information gathering. Some chatbots are extensively used for word-classification processes by using Natural language processing.

B. ARTIFICIAL INTELLIGENCE

Artificial Intelligence simulates human intelligence power in various devices. These AI devices are programmed to reflect like human beings and mimic their actions. Artificial Intelligence represents the machines that exhibit natural traits that are similar to human minds such as Problem-solving and Comprehension.

C. NATURAL LANGUAGE PROCESSING (NLP)

Natural Language Processing is the program interface of how computers and humans interact with each other. This technique allows computer programs to analyze huge amounts of language data from various resources. The NLP developers can organize the knowledge to perform various tasks such as analyze text, text summarizing, topic extraction, stemming, text mining, automatic summarization, translation, speech recognizing, segmentation, and automated question answering.

D. NATURAL LANGUAGE TOOLKIT (NLTK)

Natural Language Toolkit is mainly programmed for building and applying symbolic and statistical NLP in python. It is a suite of libraries including text processing for tokenization, parsing the text, classification of text, stemming of words, and reasoning of semantics.

E. KERAS

Keras is a neural network API library, in python. It runs on top of It is effective in running on top of R, PlaidML, TensorFlow, Microsoft Cognitive Toolkit.

F. TKINTER

Tkinter is an interface that develops the GUI (Graphical User Interface) applications.

G. TensorFlow

Tensorflow is a software library framework that mainly focuses on machine learning that uses dataflow graphs and differentiates programming over various number tasks to build models. It is used by the developers to create large-scale development applications including the neural networks. It is mainly applied for Classification, Understanding, Prediction, and Creation.

III. PROPOSED WORK

PROJECT DESIGN METHODOLOGY:

The user gives the queries to the Chatbot asking for the bus number that goes from source to destination. The Chatbot consists of all the intents that have the bus numbers along with their routes. The user input is then compared with the strings in the intents file which consists of the Chatbot database. If the input values match the database values the Chatbot will return the output by specifying the bus number. Then the output is displayed to the user.

It checks if the query matches the chatbot database entry. If the exact entry is found in the database, the closely related response from the data is given by the chatbot. If it does not find any entry in the database, the bot asks the user to train it accordingly and records the response as an equitable response to the question. Chatbots are often used for customer service.

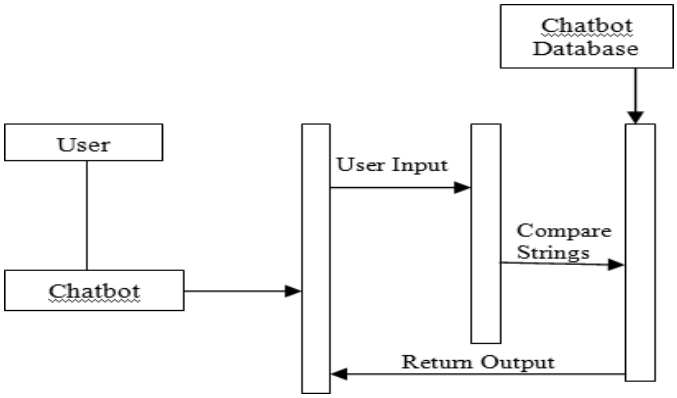


Figure 1: UML Diagram of Chatbot

They deconstruct the sentence or query inputted by the user by following the NLP techniques and mechanisms and determine the type of sentence for better accuracy. Our model is built on supervised learning techniques with already pre-loaded data to extract and build efficient models against the training set.

IV. ARCHITECTURE DIAGRAM

Natural language processing (NLP) is a combination of natural language understanding (NLU) and natural language generation (NLG). NLP is one of the forms of AI that allows chatbots to understand the user's messages and respond accordingly. AI is nothing but the field of making devices do the tasks humans do. NLP plays a super important role when it comes to building chatbots. NLP has the following layers:

1. Application.
2. A platform for Data Lake.
3. NLP Engine.
4. Data Storage.

NLP is separated into two important parts:

a) Natural Language Understanding: NLU maps the inputs given by the user to their useful depictions. It analyzes unusual appearances of the language.

b) Natural Language Generation: NLG is used for text planning, text mining, sentence planning, and text realization.

Python consists of a library named NLTK (Natural Language ToolKit) which implements NLP. The following are the techniques present in NLP:

a) TOKENIZATION: It is the action that breaks an unstructured sentence into simple understandable words. It also produces a useful and structural derivation of an inputted sentence by the user.

b) STEMMING: The procedure in which the given words are normalized into its basic form or root form is called Stemming.

c) LEMMATIZATION: In this process, different inflected forms of a word are put together in the form of a group. The output obtained is a perfect word after combining all the words.

d) STOP WORDS: Stop words are used to make a sentence to extract the exact meaning.

e) PARTS OF SPEECH: It is a pre-defined library that contains the parts-of-speech in python.

f) CHUNKING: It uses chunks of words from a sentence and groups them to make bigger chunks.

The NLP Engine's main functionality includes Natural Language Understanding.

NATURAL LANGUAGE UNDERSTANDING (NLU):

NLU is the action of producing significant phrases in the form of natural language. It includes −

a) Text planning: It retrieves important content from the database.

b) Sentence planning: It chooses the needed words, forms relevant phrases with those words, and gives a meaningful sentence.

c) Text realization: It is nothing but mapping the flow of the sentence plan corresponding to the sentence structure.

The natural language understanding is harder than natural language generation.

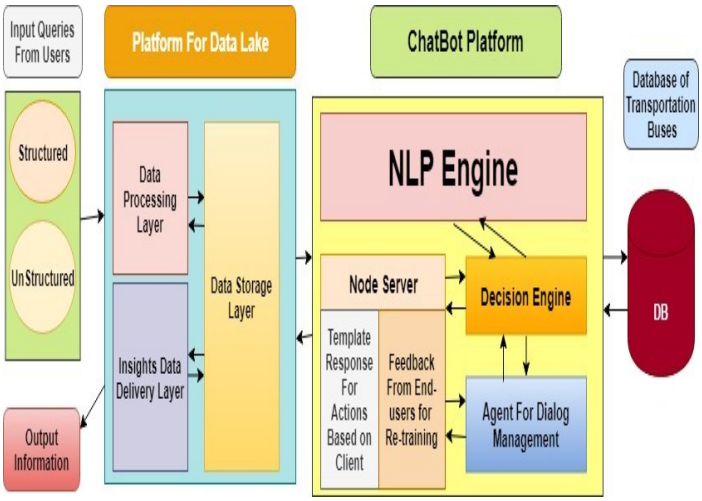


Figure 2: Chatbot Architecture

V. OVERVIEW OF ALGORITHM USED

The algorithm we used in building this chatbot is an RNN (Recurrent Neural Network) algorithm called "LSTM" which is used to classify the category for which the user's message is belonged to and it will give an arbitrary response from the record of responses using text classification.

One of the most powerful artificial neural networks and is a well- known subset of RNN is "Long Short-Term Memory" which is designed to recognize patterns in sequences that are present in numerical time series data emerging from government agencies, stock markets which also includes text, handwriting, and the spoken word.

LSTM's have a temporal dimension. LSTM's usually take less time and ordered into an accountable way when compared to other networks. Some errors can revert through back-propagation by time and layers, LSTMs help preserves those errors. They tend to maintain a persistent error rate and allows the recurrent networks to continue the learning process over several time steps and opens a channel to connect sources and their corresponding results remotely. LSTMs comprises of gated cells in which the details collected from the resources are stored.

Similar to the computer memory, the information can be stored in, or manipulated from a cell in LSTMs. The gates have the functionality of opening and closing which enables the cell to take meaningful decisions about the storage of information whether to read or write the data into the cell. The gates present in LSTM's are analog which are executed with element-wise multiplication by sigmoids. These sigmoids are present in the range of 0-1. Analog is always advantageous over digital because of differentiable nature and suitability for back-propagation. These gates receive the signal and decide whether to pass the information or block the information and filter them according to their own set of weights. The weights which are used for modulating the input and hidden states are modified via the RNN training process.

These cells comprehend when to allow the data to join, desert, or be deleted by the recurrent process of making random guesses, back-propagating the errors occurred, and altering weights via gradient descent.

The following figure shows the information about how the data is flowing through the memory cell and the way it is managed by the gates.

Starting from the base of the diagram, the three-way arrows show where the data flows into the cells at various plots. The merging of current input and previous cell state is fed to the cell and also fed to all the three gates. These gates decide the way of handling the input.

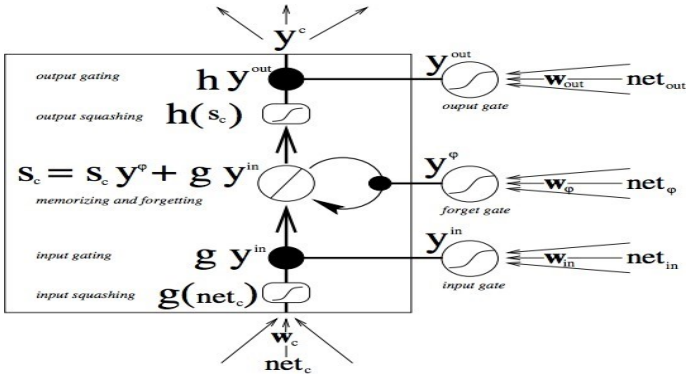


Figure-3: LSTM Representation

The diagram consists of the gates in the form of black dots which determine individually whether to take new input in or to delete the current cell state. They also help in determining the state impact of the network's output at the current time step. In the diagram, S_c is the present state of the memory cell, and g_y_in is taken as input to it. Every gate can be opened or closed, and they will recombine their open and closed/shut states at every step. The flows of the diagram represent that the cell can fail to remember its state or be written to or be read from, or not, at every time step. The result of each operation is given to us by large bold letters. There are dissimilar sets of weights that filtrate the input for its corresponding input, output, and forgetting. The forget gate is always portrayed as a linear naming function, for the reason that in case if the gate opens, the present state of the memory cell will be multiplied by one and propagates forwards to the next step.

In the below diagram, the gates are presented in their working model which has straight lines. These lines represent the closed gates and blank circles that represent open ones. The forget gates are represented by the lines and circles which are running horizontally down the hidden layer of the below figure.

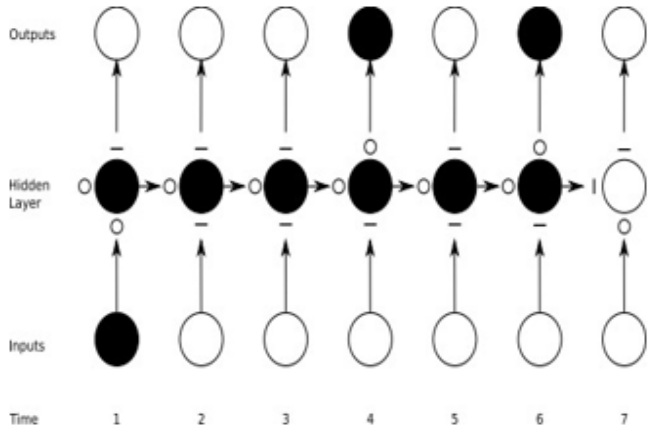


Figure-4: Neural Network Diagram

The main function that is performed by the feed-forward networks is to map one input to the corresponding output. But the recurrent networks are capable of mapping one input to many outputs that are related to it.

VI. DATASET AND TOOLS

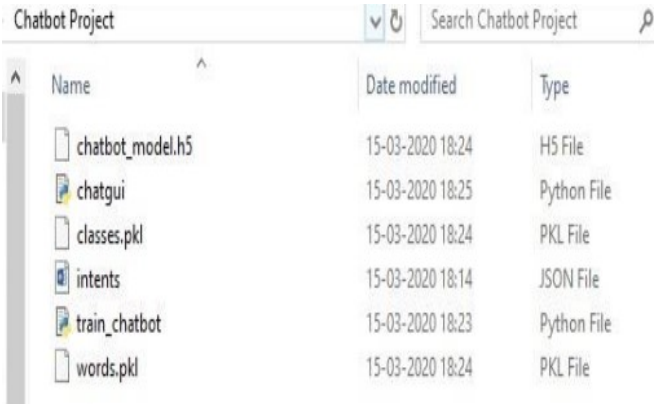
The bus dataset is taken from the Vijayawada city buses information database and formatted into the intents.json file. This JSON file comprises the patterns we need to find and the responses that it will deliver to the user. The tool used for running the chatbot is Python IDLE.

VII. RESULTS & OBSERVATIONS

A. DESCRIPTION OF RESULTS & OBSERVATIONS

We used deep learning techniques like LSTM to build our chatbot. Our chatbot is trained recursively on the bus dataset which comprises of intents, patterns, and response. The libraries that are imported to create customer service chatbot are NLTK, TensorFlow, Keras, tflearn.

Our chatbot's structure of files is as follows:



A) Intents.json – This file consists of all the data that we make use of for training the model. The data file contains a collection of tags that have already defined patterns, responses in it.

B) train_chatbot.py – This file is comprised of a script that is used in building the model. It trains the model by using deep learning techniques to classify and identify what the user is asking the bot.

C) words.pkl – All the distinct words are stored in this pickle file and it comprises the records of our vocabulary.

D) Classes.pkl – The classes.pkl is also a pickle file that comprises the list of all categories that can be used to store all the tag names to classify when we are predicting the message.

E) Chatbot_model.h5 – This model is trained which comprises a hierarchical data format and has stored weights and the architecture of our training model.

F) Chatgui.py – The Chatgui.py is a script file where we will build a GUI (Graphical user interface) that is required for the better user experience to use our chatbot.

We will build this Chabot by following these steps:

1. Import the required libraries and load the data –

Create a new python file named train_chatbot where all the required modules and packages are imported. To read the JSON data file in our python program we imported the JSON package. This package has the functionalities that are used in parsing.

2. Preprocessing the data -

Working with raw data is conflicting, so data has to go through a lot of pre-processing techniques for the machine to easily understand. We use the tokenizing technique to break the sentences into words. Our intents file contains a list of patterns and responses, which are tokenized individually and appends the words in a list of records.

Our tags are placed in a list of classes to add all the intents associated with patterns. Next, we will lemmatize each word i.e, we convert words into the lemma form and remove

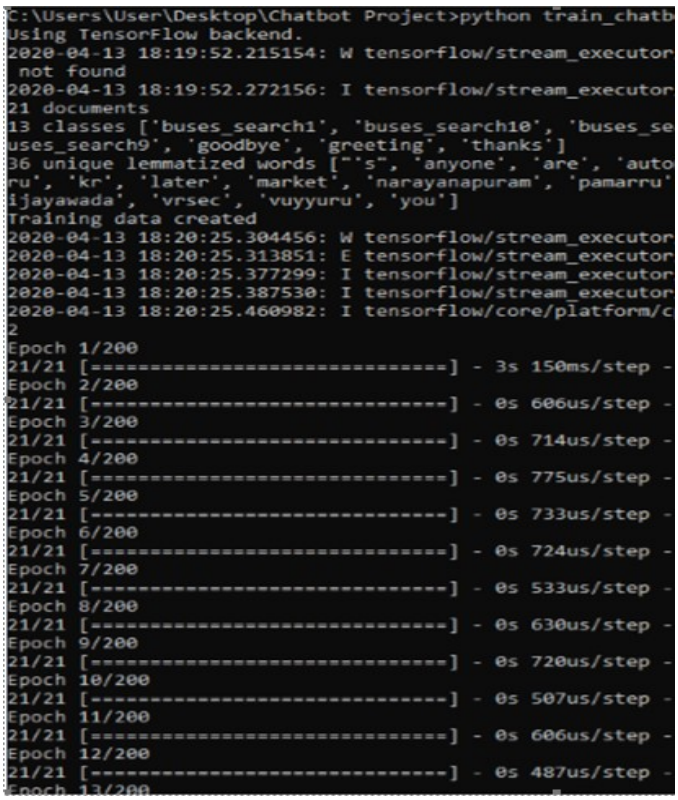
duplicate words from the list of words so that we can reduce all the canonical words. The words contain the vocabulary of our chatbot and classes contain the entities to classify. We create a pickle file by using the pickle.dump () method. This file stores all the objects obtained while running the python file which is used in the process of prediction.

3. Create training and testing data –

The training data is created which takes the input entry and gives the output response.

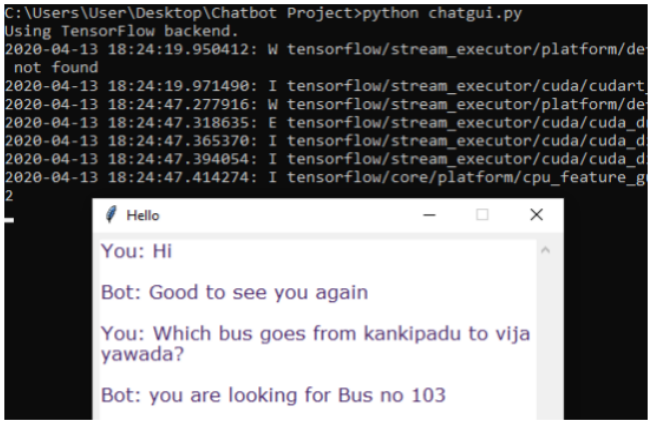
4. Training the model –

Our deep learning models architecture consists of 3 dense layers. The Keras sequential API is used for this. The first layer consists of 128 neurons, the second layer has 64 and the last layer has the same neurons as that of the number of classes. The remaining dropout layers are introduced to reduce the over-fitting of the model. We have used the SGD optimizer and fit the data to start the training process of the model. After the training of 200 epochs is finished, the model's accuracy reached 100%, then we saved the trained model using Keras model.save ("chatbot_model.h5") function



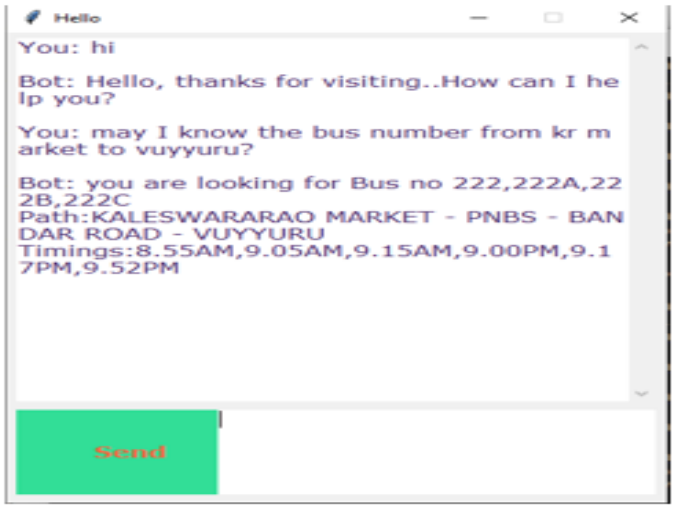
5. Interacting with the chatbot

Our model is built successfully and is ready to chat, we now create a nice graphical user interface for our chatbot in a new file name "chatgui.py". We are using the Tkinter module to build the structure of the desktop application and then we will receive the user message and perform the preprocessing techniques before we input the message into our trained model. The model then takes the input and will predict the tag of the user's message and will randomly select the response from the list of responses available in our intents file.



B. TEST CASE RESULTS:

In our test case, we communicated with the bot by asking a few queries to know the route from Kr market to Vuyyuru. As our chatbot is already trained with the intents file that consists of the information related to the query, it checks the tags, classes, and pickle files for the data and gives the response accordingly in the given fashion. A chatbot can be able to detect any query irrespective of the grammar used by different users.



The above image is the GUI representation of our chatbot. It consists of a send button that is given the functionality to send the query to the chatbot. Our chatbot's GUI representation is named 'Hello' which is represented at the top. It retrieves the text from the intents file based on the question asked.

C. PERFORMANCE ANALYSIS:

Long Short term memory algorithm is used in our chatbot to ensure that the accuracy increases when compared to other algorithms in use. In this approach, the words given by the user are clearly understood by the recurrent training of the input context. In our work on identifying the user context, the accurate results given by our algorithm are more important.

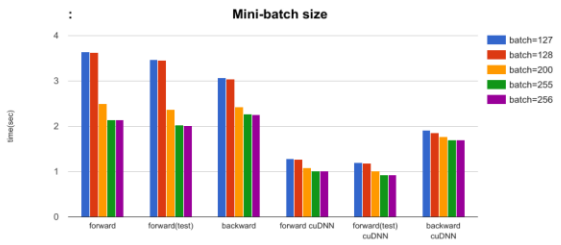


Fig 6: Performance comparison of LSTM with various batches

D. COMPARATIVE STUDY:

When coming to traditional existing algorithms like Multinomial Naïve Bayes and Restricted Boltzmann machine learning algorithm, the performance level is very low. The multinomial Naïve Bayes algorithm demonstrated the accuracy of 76 on the evaluation set. Eventually, for the Restricted Boltzmann algorithm, we found that the accuracy is 79 for CD at a learning rate of 0.003, 5000 iterations for a mini-batch size of 200, and a K-value of 1.

Table 1: Comparison of the proposed algorithms

Algorithm	**Accuracy**
Multinomial Naïve Bayes	76.57
Restricted Boltzmann	79.3
LSTM	95.02

E. FIELDWORK DETAILS

The main aim of the chatbot is to clarify the queries of users that they have regarding the bus number details. As a part of fieldwork, we have gone to different bus stops and enquired about the users about the problem they are facing in recognizing the buses. The conversation built paved a way for a better understanding of the expectations. The expectations are the requirements of the project development. The following inputs are given by the users at the time of fieldwork. Our chatbot helps society in a well-versed way where the users will feel satisfied with the information they obtained from our chatbot.

VIII. CONCLUSION & FUTURE WORK

CONCLUSION:

The main objective of the paradigm is to prosper a chatbot. This idea was developed by considering the problems faced by travelers. A broad spectrum of the literature review is done to clinch unison in all the chores, where a chatbot is considered the best model to satisfy the requirements. Research on the chatbot draws a nexus to perceive more about the burning temporal technologies and incompatible algorithms like Artificial Intelligence, Machine Learning, Python, and NLP.

FUTURE WORK:

These following are the future extensions to the work:

1. Dragging out this application by furnishing authorization services.

2. Initiate database and perpetuate users.

3. Surge the efficacy of application by providing voice chat.

4. Enlarging it to web support

IX. REFERENCES:

[1] Feng Liao, Liyuan Shang, Wendong Wan, "Design and Implementation of Public Transportation Enquiry System Based on J2EE" in Atlantic Press, 2018.
[2] Venkataramanan.S, Suriya.D, Mr. K.R.MohanRaj," Voice Based Transport Enquiry System" in Int. Jnl. Of Advanced Networking & Applications (IJANA), 2018.
[3] Ashay Argal, Siddharth Gupta, Ajay Modi, Pratik Pandey, Simon Shim, Chang Choo," Intelligent travel chatbot for predictive recommendation in echo platform", in IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC),2018
[4] Qin M, Zhang N, Wang Z. Research on bus travel decision system based on generalized cost calculation[C] International Conference on Logistics, Informatics and Service Sciences. IEEE, 2017:1-4.
[5] Agachai Sumalee, Hung Wai Ho, "Smarter and more connected: Future intelligent transportation system" in International Association of traffic and safety services (IATSS), 2018
[6] Darius Zumstein and Sophie Hundertmark, "Communicating and transacting with Chatbots: Insights from Public Transport" in International Conferences WWW/ Internet and Applied Computing, 2017
[7] Sahaya sakila, Akshat Shrivastava, Md Arman ansari, Lakshmi Kanta Kar, Mukul Kumar, "RAILBOT: A Railway (IRCTC) Chatbot", in International Journal of Engineering Science and Computing (IJESC), 2018
[8] R. Yan, Y. Song and H. Wu, "Learning to Respond with Deep Neural Networks for Retrieval Based Human-Computer Conversation System," in Proceedings of the 39th International ACM SIGIR conference on Research and development in information retrieval, 2016
[9] Salim Akhtar Sheikh, suneetha bansal, vineeta Tiwari "Artificial Intelligence Based Chatbot for Human Resource Using Deep Learning" published in 2019 in Journal of Emerging Technologies and Innovative Research
[10]Andrew Pulver, Siwei Lyu,"LSTM with Working Memory" published in 2017 International Joint Conference on Neural Networks (IJCNN)
[11] Miss. Amruta, P. Deshmukh, Prof. V. B. Bhagat," Voice Based Retrieval for Transport Enquiry System", In International Research Journal of Engineering and Technology (IRJET), 2019
[12] Shafquat Hussain, Omid Ameri Sianaki, Nedal Ababneh, "A Survey on Conversational Chatbots Classification and Design and Techniques" in 2019