“CHATBOT IN FOOD INDUSTRY”

By

KUNAL NITIN HADKAR 215A2132
VARNIKA PRASAD 116A2114
ROHIT VISHWAKARMA 116A2117

Under the guidance of

Dr. PREETI HEMNANI

SUBMITTED IN PARTIAL FULFILLMENT FOR THE
DEGREE OF
BACHELOR OF ENGINEERING
IN
ELECTRONICS AND TELECOMMUNICATION

SIES Graduate School of Technology
RISE WITH EDUCATION

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION
SIES GRADUATE SCHOOL OF TECHNOLOGY NERUL, NAVI
MUMBAI – 400706 ACADEMIC YEAR 2019– 2020

CERTIFICATE

The report is submitted in partial fulfillment of the degree course of Bachelor of Engineering in Electronics and Telecommunication Engineering, of University of Mumbai during the academic year 2019-2020.

1.	KUNAL NITIN HADKAR	215A2132
2.	VARNIKA PRASAD	116A2114
3.	ROHIT VISHWAKARMA	116A2117

Internal Guide   Head of Department   Principal

We have examined this report as per University requirements at SIES Graduate School of Technology, Nerul (E), Navi Mumbai on ____________________.

Name of External Examiner: ____

Signature with Date: ____

Name of Internal Examiner: ____

Signature with Date: ____

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

KUNAL NITIN HADKAR 215A2132
VARNIKA PRASAD 116A2114
ROHIT VISHWAKARMA 117A2117

Date

Signature

ACKNOWLEDGEMENT

It gives us immense pleasure to thank Dr. Atul Kemkar, our Principal for extending his support to carry out this project. We also thank to our Head of Department Dr. Preeti Hemnani and Project Co-Ordinator Dr. Shishir Jagtap for his support in completing the project. We wish to express our deep sense of gratitude and thank our Internal Guide Dr. Preeti Hemnani for her guidance, help and useful suggestions, which helped in completing our project work in time. We are glad that are panelist let us put our thoughts and ideas in front of them and guided us in to make it a useful product for the society. Also we would like to thank the entire faculty of Electronics and Telecommunication department for their valuable ideas and timely assistance in this project, And last but not the least, we would like to thank our non-teaching staff members for helping us in providing the lab facilities and the trouble shooting whether it was hardware or software, And our college for their support, in facilitating timely completion of this project.

Project Group
KUNAL NITIN HADKAR
VARNIKA PRASAD
ROHIT VISHWAKARMA

ABSTRACT

Some programs respond intelligently like a human or similar to a human. Such a program is called as a Chatbot. Chatbots are widely popular now-a-days and catching speed as an application of computer communication. There are generally two types of chatbots- a rule-based bot which works on keywords using 'Natural Language Processing' or NLP, which is a subset of Artificial Intelligence or AI, and the other is a complete 'Machine Learning and Deep Learning' based bot. This paper addresses the design and implementation of an NLP and AI developed Chatbot system in the food industry i.e. a bot in a restaurant which will perform the complete task of taking up orders and bringing it back at your table saving up your waiting time and cutting costs required for manpower. In this project, an AI (Artificial intelligence) based bot which can closely replicate natural human, behavior language and produce, replies to the conversations is to be designed. The project moreover emphasizes more on the application of chatbot in a particular domain. We have chosen the 'Food industry' as the domain for our chatbot. In this project we will be working to develop the bot in such a way that replication to the human conversations is maximum and the bot to give a instant reply to any reasonable and sufficient input given to it in the form of text or speech. The proposed bot is to be designed using Python and a number of its libraries. The bot can be a good candidate in a number of other fields such as banking, airport and railway systems (in most of the public transport system), hospital systems, etc. Nowadays chatbots have become very popular due to their ability to replace humans in most of the industrial systems. Hence considering the importance of the chatbot in the outside world as per the progress of Machine Learning systems and Artificial intelligence we chose this project so as to get hands-on Knowledge and hone our skills.

Contents

List of Figures

CHAPTER 1

1 INTRODUCTION

1.1 INTRODUCTION

Customer satisfaction with a company's services is often seen as the key to success and long-term competitiveness for a company. In today's world, we find ourselves now in a digital economy of expectation, one where we don't want to stand in a queue to order food or even wait for the waiter to do so. On a personal level basis, we have observed the same and that's where we thought of implementation of our idea, and this is where the idea of a 'Chatbot' came into picture. A chatbot is a program designed to counterfeit a smart communication on a text or spoken ground. Put simply, chatbots can mimic human conversations. By encouraging customers to place an order with the bot, thereby reducing the waiting period, it can increase sales and also attract the younger technology-driven audience. It will also help in reducing manpower required by restaurants, thus cutting costs with increasing efficiency.

A chatbot is basically a software that resides on a messaging platform and is developed with the purpose of having lifelike interactions with humans through text or audio. But how can these bots contribute to restaurants, you ask?

Suppose you feel like having your favorite pizza but your order is taking up too much time at the pizza joint. Wouldn't it be great if someone magically delivered your pizza? Well, we don't have pizza delivering robots as of now, but we have the next best thing – bots.

This project is based on a 'Food request accepting chatbot.' A system at the table, with the restaurant webpage opened, serves as the communication medium. A customer at the table can have a conversation with our bot, check the menu, etc. via the webpage and can decide and order his/her favorite dish. The restaurant staff gets notified of the order immediately. Once the order is received, the hardware part of the bot takes care of delivering your favorite food item right at your table. One can even order his/her favorite food item from home and get it delivered at the doorstep, so this project can also be considered as a smart food ordering system.

A chatbot is a conversational agent where a human conversation is Simulated using a Computer program. Techopedia has defined chatbots as - A chatbot is an (AI) program that simulates interactive human conversation by Making use of phrases and auditory or text-based signals. Humans nowadays have been replaced chatbot in industries in order to reduce manpower. This reduces the costs for the companies. Many sentiments of text, voice, etc. can be given as user inputs to the chatbot.

1.2 Need of Project

Food industries or restaurants with higher number of customers, especially during peak hours, really find it difficult to cope up with the incoming traffic and also deal with home delivery orders, etc. at the same time. Longer waiting time to serve customers leads to the bigger problem of losing them in future. Technology though nowadays, with the flourishment of Data Science, can help with this. Nowadays, companies are not only trying to improve the supplies but also to reduce the manpower required for tasks and thus cut costs. This has become possible by the usage of these conversational bots or talk-bots as they call them too, which are improving time efficiencies along with giving industries the opportunity to reduce costs.

CHAPTER 2

2 Literature Survey

ELIZA which was developed by Joseph Weizenbaum in 1964[1] is considered to be one of the first chatbots which worked on pattern making and pattern recognition. Main functions were to show triviality of communication between a human and a machine. It was carried out by making her remember words and conversations and later using that in a conversation as if she was a therapist. For example if someone mentioned their mood, ELIZA will automatically ask them about how their day was. Illusion was created on real understanding and human interactions. Often after chatting with ELIZA people could not believe she was a bot[2]. There were a certain limitations that arises from the conversations with ELIZA.

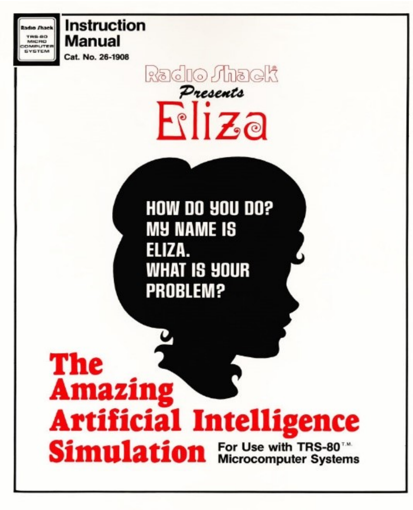
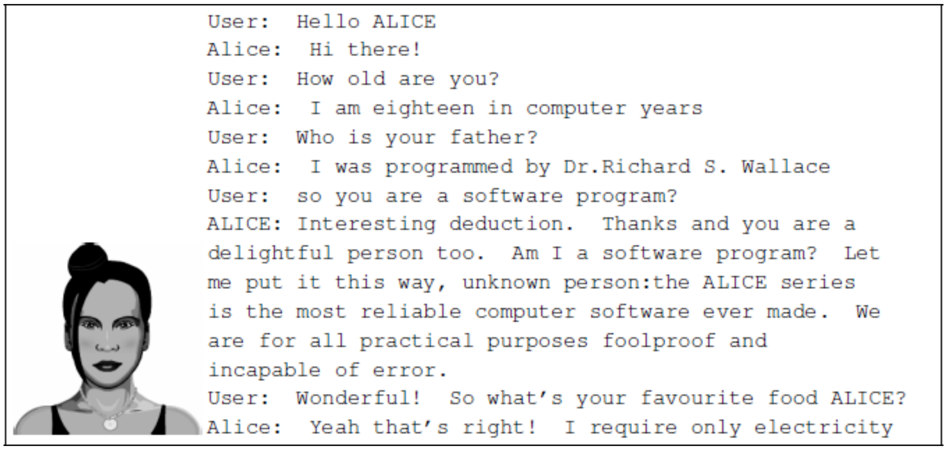

Instruction Manual
Cat. No. 26-1908
Radio Shack
Presents
Eliza
HOW DO YOU DO? MY NAME IS ELIZA. WHAT IS YOUR PROBLEM?
The Amazing Artificial Intelligence Simulation
For Use with TRS-80™ Microcomputer Systems

Figure 1: Image of instruction manual of ELIZA[3]

A.L.I.C.E (Artificial Linguistic Internet Computer Entity) is rule-based chatbot based on the AIML i.e Artificial Intelligence Markup Language . A.L.I.C.E is built upon more than 35,000 combinations where each combination has its own pattern. (A.L.I.C.E.) was originally written in Java and utilized an XML schema called AIML (Artificial Intelligence Markup Language).

The aim of AIML, which was distributed with an open source license and has been implemented on a number of different platforms (i.e. Pandorabots), was

Figure 2: Joseph Weizenbaum using ELIZA[4]



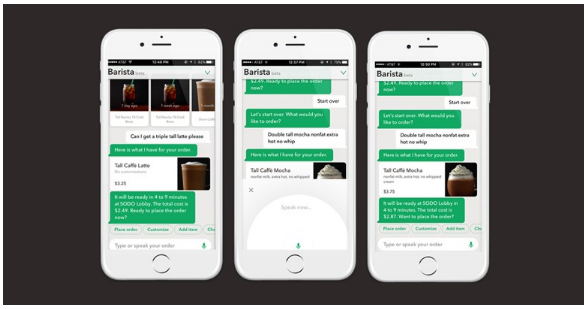
```
User:  Hello ALICE
Alice:  Hi there!
User:  How old are you?
Alice:  I am eighteen in computer years
User:  Who is your father?
Alice:  I was programmed by Dr.Richard S. Wallace
User:  so you are a software program?
ALICE: Interesting deduction.  Thanks and you are a
delightful person too.  Am I a software program?  Let
me put it this way, unknown person:the ALICE series
is the most reliable computer software ever made.  We
are for all practical purposes foolproof and
incapable of error.
User:  Wonderful!  So what's your favourite food ALICE?
Alice:  Yeah that's right!  I require only electricity
```

Figure 3: 3 Pictorial representation of conversation between user and ALICE[5]

to encourage other developers to modify the initial program and produce numerous Alicebot clones. A.L.I.C.E. won the restricted category Loebner Prize three times: in 2000, in 2001, and in 2004. This prize, initiated by Hugh Loebner in 1991, is “the first formal instantiation of the Turing Test”. AIML scripts Md. Shahriare Satu and Shamim-AI-Mamun reviewed applications of chatbot. It was stated that AML based chatbots are efficient to use , easy implementation.[6]

Thomas N. T. and Amrita Vishwa designed a AIML and LSA based chatbot to provide the customer care service over the E- commerce websites. Their approach shows we can improve the chatbot ability by adding other models to it. In android operating system, we can implement the chatbot using the various approaches. One of the approaches is shown by Rushabh Jain and

Figure 4: Shamim-AI-Mamun[7]



Can I get a triple tall latte please

Here is what I have for your order.

Tall Caffè Latte
No customizations
$3.25

It will be ready in 4 to 9 minutes at SODO Lobby. The total cost is $2.49. Ready to place the order now?

$2.49. Ready to place the order now?

Start over

Let's start over. What would you like to order?

Double tall mocha nonfat extra hot no whip

Here is what I have for your order.

Tall Caffè Mocha
nonfat milk, extra hot, no whipped cream
$3.75

It will be ready at SODO Lobby in 4 to 9 minutes. The total cost is $2.87. Want to place the order?

Figure 5: Photo of Chat Bot Guide[8]

Burhanuddin Lokhandwala in their Android based Chat-Bot paper
Barista Bot in 2018 was launched by global food chain STARBUCKS[9]. This allows customers to quickly pre-order and pay for their order and save their time and also skip the queue during peak hours in store. This eliminates the human mistakes while taking the orders like spelling your name wrong or getting confused between words of similar sound. In the USA, A supermarket brand named ‘Whole Foods’ launched their Bot to help customers decide their shopping list and also recipes for the week. The chatbot speeds up interactions with customers by using emojis instead of words. This also creates a unique experience for customers online. Dominos is probably one of the best-known influencers for Artificial intelligence in the

food industry. They created their own chatbot named ‘DOM’ in 2014. Today more than 55 percent of its sales in the India is via various digital platforms including using DOM. With Domino’s DOM currently being tested to answer customer phone calls in America and KFC opening the first AI-only staffed branch in Shanghai, China, chatbots seem here to stay for the food industry with new and exciting possibilities.[10]

CHAPTER 3

3 OBJECTIVES OF THE PROJECT

1. The objective or attempt is to provide service to the customers in food industry or restaurants/cafes, etc. via conversational bots or simply called chatbots as soon as they arrive at a table thus not making them to wait for the waiters for long periods of time.

2. This can be done by installing chatbots at the tables and the customers can order their stuff or even check the menu.

3. The bot then behaves just like a server and provides required service and passes on the message to the chef/manager. As far as working of the bot is concerned, we will have to train the bot.

4. The idea to train a chatbot as per the company or individual's requirements is by feeding the bot with conversational logs and then the bot makes use of Data Science or Machine Learning to extract desired information at a very fast rate as per the question asked to it by the user and performs mapping of words or phrases as per the requirement.

5. The other objective after achieving the main objective i.e. after getting appropriate and satisfactory response from the bot is to successfully make orders and deliver it to the head chef in the kitchen via text message or any other possible means. The hardware bot will then deliver the food item to the table.

6. For getting orders online, another objective is to get the website/webpage online so that people can order from anywhere around and get access to the restaurant's services.

CHAPTER 4

4 METHODOLOGY

Since we are working on an NLP developed rule-based bot, the methodology first gives a detailed information about them. However, one should have complete knowledge about the different chatbot types that can be designed and so there is a thorough explanation for the same in this section. Thereafter, the actual methodology is explained.

Rule-Based chatbots:-
1. In a Rule-based approach, a bot answers questions based on some rules defines to them and on which it is trained on. The rules can be vary from very simple rules to very complex rules.
2. The bots can handle simple queries but sometimes they can fail to manage complex rules.
3. These are the algorithms which are used where handful set of instructions are to be given to the system.

Self-learning chatbots:-
1. Self-learning bots are the ones that use some Machine Learning-based approaches. Self-Learning are definitely more efficient than rule-based bots.

2. These bots can be of further divided into two types.

A) They are Retrieval Based
B) Generative bots.
i. In retrieval-based models, a chatbot uses some self-learning approach to select a response from a library of predefined responses.
ii. The chatbot uses the context of the conversation for selecting the best response from a list of bot messages which are predefined. Methods for selecting a response can be engineered in many different ways.
iii. These can be from machine learning classifiers to rule-based if-else conditional logic.
iv. Generative bots not always replies with one of the answers from a set of answers but also Can generate the answers.
v. As they take word by word from the query and generates the answers, this makes them more intelligent and make them sound more sensible.

The designing and development of this system can be divided into three major domains-
1. Web development.
2. Natural Language Processing based software development.
3. Hardware model using IoT.

i. The first domain refers to a website/webpage developed for the UI (User Interface). The second domain refers to the actual bot program required to train the bot for appropriate responses.

ii. A python program provided with a pre-requisite restaurant dataset/database, to train the bot based on NLTK (Natural Language tool kit), is linked to the webpage using 'Flask' framework of python which is used to link python programs to webpages to display results on them and vice-versa.

iii. This framework was chosen by us over the more popular and preferred 'Django' framework, simply because of its easier syntax and lesser complexity. The third domain refers to the connectivity to the 'Hardware bot' which works on the principles of IoT (Internet of Things).

iv. The bot, on receiving an order, follows a path up to the kitchen, waits for a certain interval of time and then brings the order back at the customer's table.

v The following facts are kept in mind during designing and development process:

A. Selection of OS
Windows is used for this project because it is user friendly. It is also robust.
B. Selection of Software
'Brackets text editor' software is used for creating python programming as well as web development, because it is open source with a basic workspace and lesser complexity.
C. Creating a Chatbot
For creating a Chatbot, a program has to be
written. Python programming language is used for programming.
The Chatbot is created in such a way to help the user, improve the communication and amuse the user.
D. Creating a Chat
The chat is created using a pattern that is known to the user and could be

easy to understand. This chat is stored as a text file and serves as the database on which the chatbot is trained.

E. Pattern Matching

It is a technique of artificial intelligence and NLP used in the designing process. The user input is matched with the inputs saved in the database and corresponding response is returned.

F. Complexity

The design of the bot is very simple. It just answers
to the questions asked by the user, if similarities are found in the database.

G. Conversational and Entertaining

The Chatbot responses are a way known to the user. The conversation follows a Basic English language and interacts in an easy to read manner. The conversation between the user and the Bot is entertaining. It is like talking to another person.

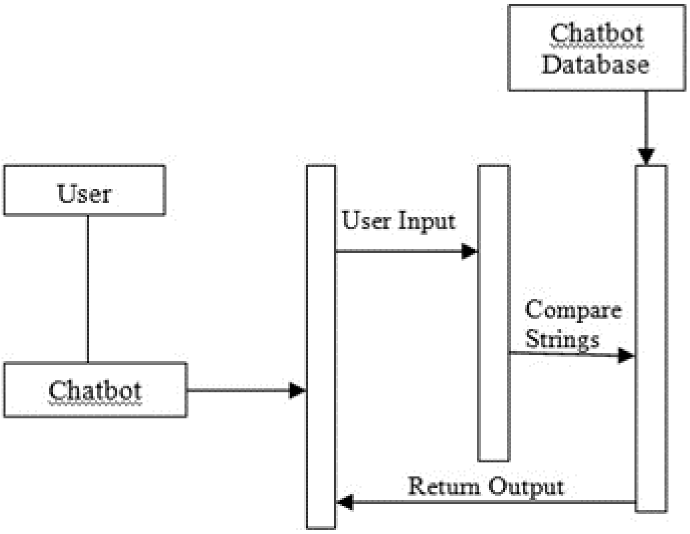


Figure 6: Sequence Diagram Representing Design of the Chatbot[11]

CHAPTER 5

5 DETAIL DESCRIPTION

5.1 System architecture/block diagram

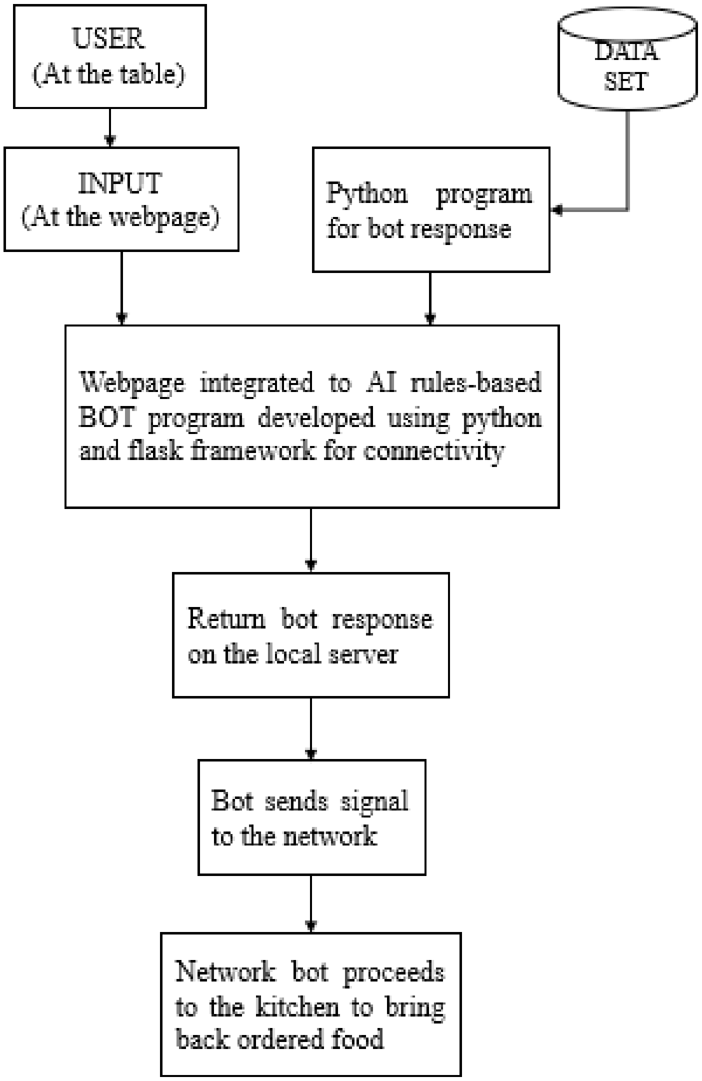

USER
(At the table)
DATA
SET
INPUT
(At the webpage)
Python program for bot response
Webpage integrated to AI rules-based BOT program developed using python and flask framework for connectivity
Return bot response on the local server
Bot sends signal to the network
Network bot proceeds to the kitchen to bring back ordered food

Figure 7: System Architecture/Block Diagram

The architecture of the entire system is summed up using a block-flow diagram shown above.
It can be explained as follows.
1. The user/customer at the table has to provide input to the system placed on the table (usually a tablet or mobile phone). This system has the website opened already and is ready for customer interaction. So, basically the user starts to enter his/her queries in the chat box section provided on the page.
2. Now, the data set prepared analyzing possible typical restaurant conversations will be compared with the user input using Python programming and then accordingly bot response is generated as well.
3. For the bot response to be generated on the webpage and to connect the user input to the Python code for comparison purposes, Python connectivity with the website is required and that is where 'Flask' framework of Python comes into picture in the architecture.
4. Appropriate bot response is generated on the server. The architecture makes it possible for people to order online too.
5. The software bot then finally informs about the final order to the kitchen chef via immediately.
6. The hardware bot then comes into picture who delivers the order at the table. For online orders though, the hardware bot isn't feasible.

5.2 Working of the Project

The first and foremost process was collecting data/creating a database. We researched about various cuisines and food items/dishes and also the conversations taking place at a restaurant. Then, the entire data was stored in a.txt file. After data collection, the program to implement the bot response was written in python. Python offers a number of libraries and frameworks free of cost to use and implement as per our needs. The two major modules used here are- NLTK and scikit learn or sklearn. Using NLTK, we performed tokenization and lemmatization while sklearn helped to performed Tf-idf vectorization and cosine similarity analysis. Tokenization is nothing but converting the entire dataset into a list of sentences or words whereas to lemmatize is to derive root words from the list of word tokens. Tf-idf vectorizer determines the frequency of word(s) in the dataset. More information on Tf-idf is shared in an article by Cory Maklin, a Data Scientist at Interest [12].

After generating responses on python platform, it was time to develop a UI where the user input and corresponding bot response will be displayed. For this, we used ‘Flask’ framework of python. The framework uses ‘routing’ method i.e. result to be displayed on a particular webpage is provided with the name of the web file along with the route to be accessed (example: for accessing the home page, route- ‘/home’ would be provided, likewise for ‘About’ page, ‘/about’ route would be provided). The framework is imported and accessed in the same python file and the user input and bot-response logic is implemented within the ‘routing to our page’ portion of the program. On executing the program without errors and bugs, the page is displayed on the ‘localhost’ server having a port number (generally 5000, but one can change it if required). Another important thing to note here is that we can route as many pages we want for a website using this framework, and so, we have a ‘Menu’ page and an ‘Instructional guide’ page too.

5.3 Modelling and Programming/Building the Bot

Natural Language Processing, or NLP for short, focuses on the interactions between human language and computers which works with AI, CS, etc. With 57NLP human interactions can be made smarter and effective. Utilization of NLP can help achieve better platform to perform various tasks such as machine translation, text parsing, speech recognition, image classification, pattern recognition, etc.

With user friendly interface it consists text processing libraries which help in parsing, semantic reasoning, stemming of words, tokens can be formed. NLTK is very efficient and diverse toolkit which provides a variety of libraries and modules helping to perform operations using python.

Before getting started with NLP, knowledge of NLTK and scikit library is important.
1.Initial stage: Download and Install NLTK
2.Run python – type install NLTK packages.
3.Import and run NLTK. Download all the packages at once.
4.text Pre-Processing with NLTK
Machine learning faces problems dealing with all text data hence the strings of data should be converted into vector usable by NLTK.

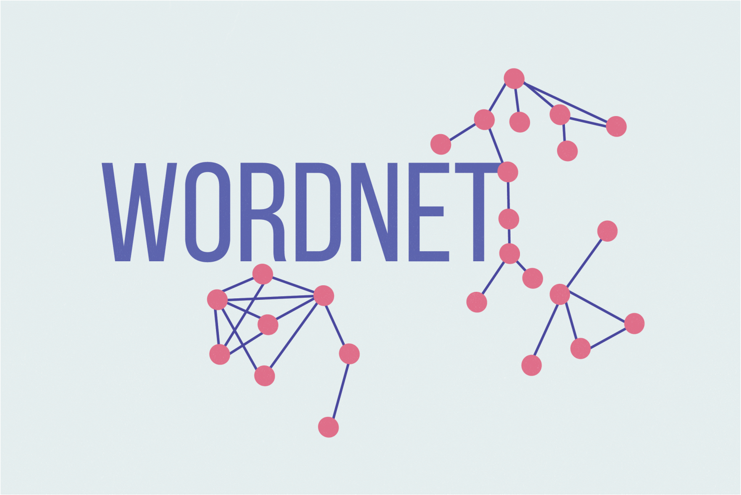


Figure 8: Wordnet[13]

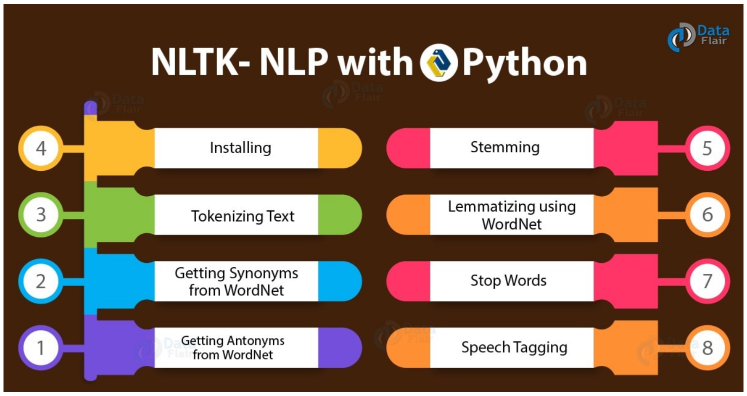


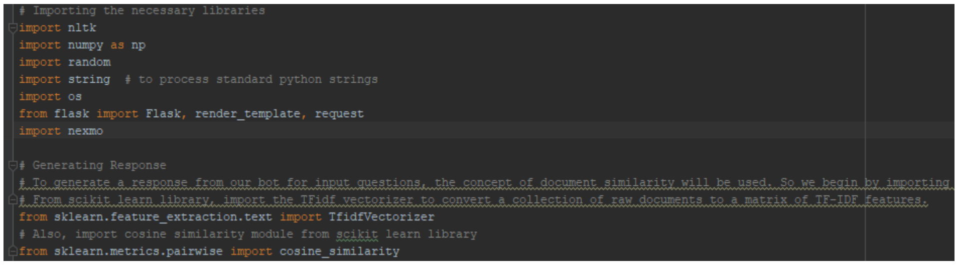
Figure 9: NLTK Python Tutorial(Natural Language Toolkit)[14]

It is important to pre-process data. Basic text pre-processing techniques includes:

- All the text is converted into either Uppercase or Lowercase so that a single word in two different cases is not treated differently.
- Tokenization: It is a method of converting sentences into small words often knows as tokens.
- Removing Noise i.e. alters anything that is not in standard form.
- Removing Stop words. Extremely common and repetitive words are completely excluded from the dataset in order to avoid confusion while selecting a sentence w.r.t. the input from the user. These are known as stop words.

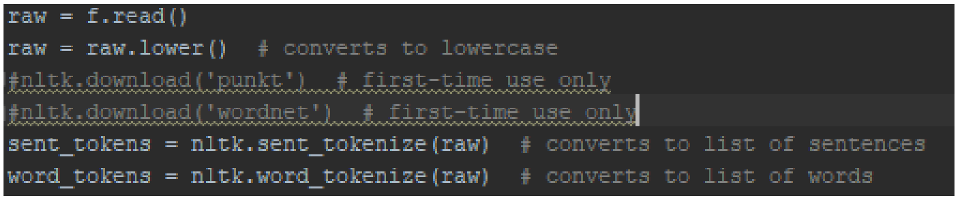
•Stemming: Stemming is the process of abbreviating words into their stems or root form.
For example, the word “Stunning” “stunned” “stun” all have a stem word “stun”
• Lemmatization: It is a modified version of stemmatization. The basic difference between a stem word and a lemma is that stems don’t always makes sense whereas you can look up a lemma in a dictionary. For example the word “sat”, “sitting”, “sit” will have a lemma “sit”.
• Bag of Words: After the text pre-processing is done, we need to transform the text into a meaningful arrays of numbers. This is called bag of words. It consists of two things: i. A vocabulary of known words
ii. Measure of repetitive or known words
The main focus behind the Bag of words is to scan a document w.r.t to the content it holds. Document is said to be similar if it has similar content.
For example, if we have words Art, is, the, everywhere, live, and we want to vectorize “Art is everywhere”, we will get the following vector: (1, 1, 0, 1, 0)[15]
TF-IDF Approach The main problem with the bag of words approach is that it gives weightage to longer documents. Frequently used words are given a higher preference in the document. Another way to deal with such a situation is to rescale the frequency of words according to how often they appear in a document. A score is maintained for all the words used. This method of scoring is called Term Frequency-Inverse Document Frequency or TF-IDF.
Term Frequency: scoring of each and every word of the document. TF = (Number of times term t appears in a document) / (Number of terms in the document) Inverse Document Frequency: measure of how rare is a word in a document. IDF = 1+log(X/x), where, X is the no. of documents and x is the number of documents with word t appeared in it. Tf-IDF weight is a statistical measure to evaluate the importance of a word in a document and this weight can be used in retrieval of information and text mining.
Example: Consider a document containing 500 words wherein the word ‘tree’ appears 5 times. The term frequency (i.e., tf) for tree is then (5 / 500) = 0.01. Now, assume we have 10 million documents and the word tree appears in ten-thousands of these. The Calculations for IDF is (10,000,000 / 1,000) = 4. Thus, the Tf-IDF weight is the product of these quantities: 0.01 * 4 = 0.40. Tf-IDF can be implemented in scikit learn.
Cosine Similarity TF-IDF gives two real valued vectors in a vector space. Cosine similarity is basically the dot product of two vectors divided by product of their mods. It can also be called finding similarities between two non-zero vectors.

Cosine Similarity (D1, D2) = Dot product (D1, D2) / $\||D1||\ *\||D2||$
where D1 and D2 are two non-zero vectors.
Now, let us get going with the actual building process: Firstly, we have imported the necessary libraries for data extraction and data manipulation processes. These include- nltk, numpy, sklearn, random and string libraries.

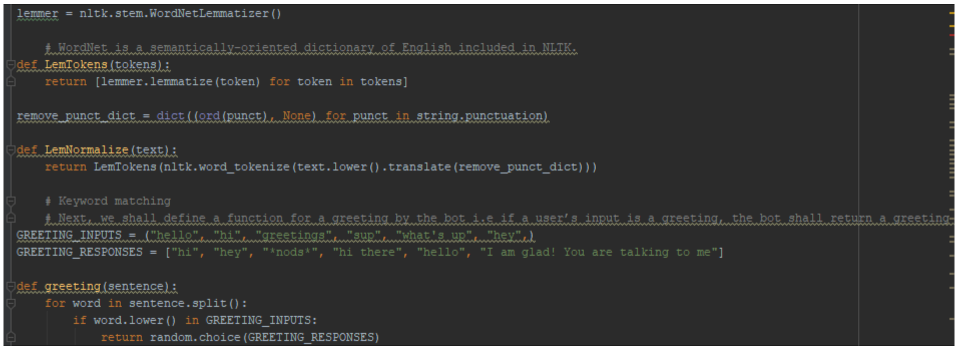


```
# Importing the necessary libraries
import nltk
import numpy as np
import random
import string  # to process standard python strings
import os
from flask import Flask, render_template, request
import nexmo

# Generating Response
# To generate a response from our bot for input questions, the concept of document similarity will be used. So we begin by importing
# From scikit learn library, import the TFidf vectorizer to convert a collection of raw documents to a matrix of TF-IDF features.
from sklearn.feature_extraction.text import TfidfVectorizer
# Also, import cosine similarity module from scikit learn library
from sklearn.metrics.pairwise import cosine_similarity
```

Then for the dataset we have included typical conversations and other necessary information in a .txt file which is the 'corpus' for our project. Another important point to note here is that at this stage the bot isn't learning things on its own and correcting its experienced mistakes which involves training process; it is just looking out for keywords related to the text we input and performs actions based on that. Then, for the pre-processing to start we go through the corpus files and convert strings into single words. These are saved as sentence tokens or word tokens. Then lemmatization is performed i.e. we define a function called as LemTokens which returns 'normalized tokens' for the input tokens. After this process, 'Greeting process' of the bot is defined i.e. a greeting by a bot is defined as a function i.e if the user is greeting the bot as an input the bot shall greet in return. A list here consists of probable greeting inputs from the user party and its response is also stored in a list which the bot will return to the user if it reads the input as a greet from his/her side.

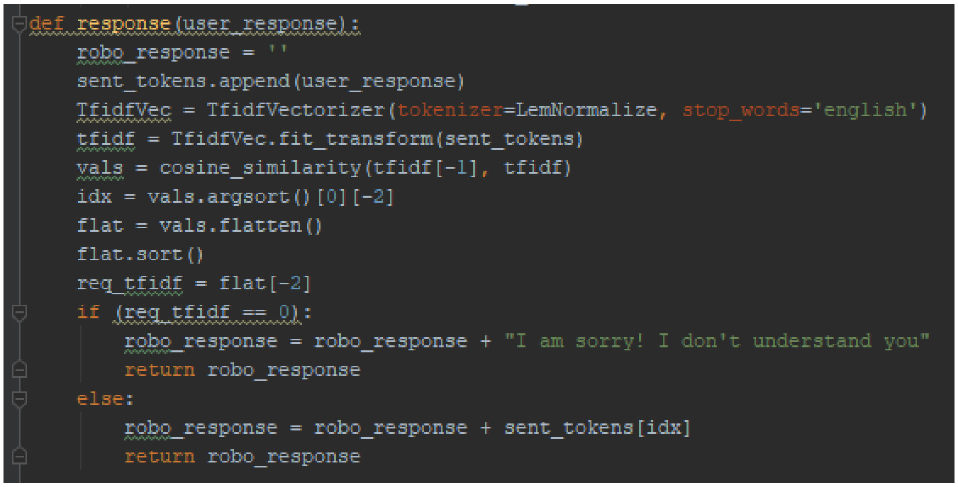


```
raw = f.read()
raw = raw.lower()  # converts to lowercase
#nltk.download('punkt')  # first-time use only
#nltk.download('wordnet')  # first-time use only
sent_tokens = nltk.sent_tokenize(raw)  # converts to list of sentences
word_tokens = nltk.word_tokenize(raw)  # converts to list of words
```



```
lemmer = nltk.stem.WordNetLemmatizer()

    # WordNet is a semantically-oriented dictionary of English included in NLTK.
def LemTokens(tokens):
    return [lemmer.lemmatize(token) for token in tokens]

remove_punct_dict = dict((ord(punct), None) for punct in string.punctuation)

def LemNormalize(text):
    return LemTokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)))

    # Keyword matching
    # Next, we shall define a function for a greeting by the bot i.e if a user's input is a greeting, the bot shall return a greeting
GREETING_INPUTS = ("hello", "hi", "greetings", "sup", "what's up", "hey",)
GREETING_RESPONSES = ["hi", "hey", "*nods*", "hi there", "hello", "I am glad! You are talking to me"]

def greeting(sentence):
    for word in sentence.split():
        if word.lower() in GREETING_INPUTS:
            return random.choice(GREETING_RESPONSES)
```

Next and the most important part is generating responses from the bot after it has understood parameters of the data from the previous steps. Document matching and checking for similarities is used. Modules from sklearn library 'TFidf Vectorizer' are imported to convert a given data in to a matrix form of TFidf features. Also 'cosine similarity' module is imported from the same library. We have defined a function 'response' which searches the user's utterance for one or more known keywords and returns one of several possible responses. The idea here is to append the list of tokens of words from the dataset with the user's input/response and then after sorting the words using the sort function, comparison is done with the second last closest phrase since the most recent last phrase would be the appended phrase itself. In such a scenario the output would be the phrase itself which the user had entered and so to avoid this comparison is made with the second last closest one after appending the user's response.
If the bot is unable to find an appropriate response to the input it will return "I'm sorry, I don't understand you." Finally, we have fed the lines that we want our bot to say while starting and ending a conversation depending upon the user's input.



```
def response(user_response):
    robo_response = ''
    sent_tokens.append(user_response)
    TfidfVec = TfidfVectorizer(tokenizer=LemNormalize, stop_words='english')
    tfidf = TfidfVec.fit_transform(sent_tokens)
    vals = cosine_similarity(tfidf[-1], tfidf)
    idx = vals.argsort()[0][-2]
    flat = vals.flatten()
    flat.sort()
    req_tfidf = flat[-2]
    if (req_tfidf == 0):
        robo_response = robo_response + "I am sorry! I don't understand you"
        return robo_response
    else:
        robo_response = robo_response + sent_tokens[idx]
        return robo_response
```

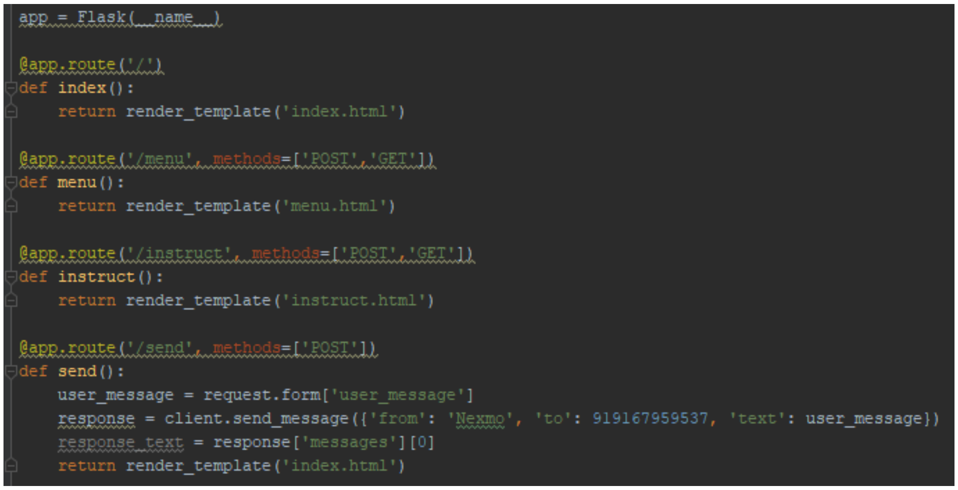
The entire coding is done using Python 3.6 and PyCharm IDE for python is used. A fair bit of questions and conversations are being answered and held by the bot within its ‘dataset’ reach.

5.4 Website description

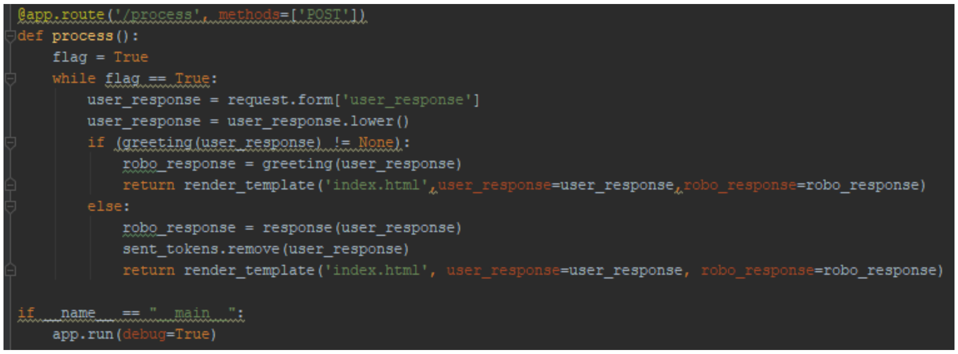
The website we have built for the UI is a responsive HTML-CSS and Jquery based website. The theme is a restaurant based one with Menu and an instructional page on how to access the bot. The Home page is the one which contains the area to access the bot and have a conversation. It also has a separate Text-box area to order. Please visit http://rohitvish14.pythonanywhere.com/. For the results, we have included screenshots of the website in the next chapter.

5.5 Connectivity

Now for the python code to be connected with our website to display the results and get customer feedback and queries, we have used the Flask framework from Python. The framework uses ‘routing’ method i.e. result to be displayed on a particular webpage is provided with the name of the web file along with the route to be accessed (example: for accessing the home page, route- ‘/home’ would be provided, likewise for ‘About’ page, ‘/about’ route would be provided). The framework is imported and accessed in the same python file and the user input and bot-response logic is implemented within the 'routing to our page' portion of the program. On executing the program without errors and bugs, the page is displayed on the 'localhost' server having a port number (generally 5000, but one can change it if required). Another important thing to note here is that we can route as many pages we want for a website using this framework, and so, we have a 'Menu' page and an 'Instructional guide' page too.

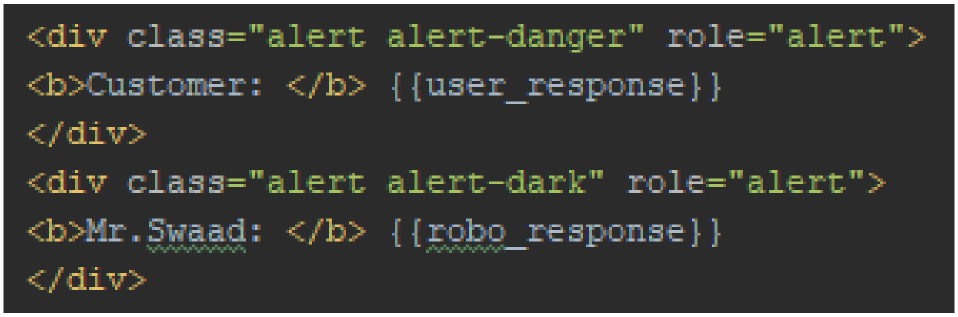


```python
app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/menu', methods=['POST','GET'])
def menu():
    return render_template('menu.html')

@app.route('/instruct', methods=['POST','GET'])
def instruct():
    return render_template('instruct.html')

@app.route('/send', methods=['POST'])
def send():
    user_message = request.form['user_message']
    response = client.send_message({'from': 'Nexmo', 'to': 919167959537, 'text': user_message})
    response_text = response['messages'][0]
    return render_template('index.html')
```



```python
@app.route('/process', methods=['POST'])
def process():
    flag = True
    while flag == True:
        user_response = request.form['user_response']
        user_response = user_response.lower()
        if (greeting(user_response) != None):
            robo_response = greeting(user_response)
            return render_template('index.html',user_response=user_response,robo_response=robo_response)
        else:
            robo_response = response(user_response)
            sent_tokens.remove(user_response)
            return render_template('index.html', user_response=user_response, robo_response=robo_response)

if __name__ == "__main__":
    app.run(debug=True)
```

An important thing to note here is that for the render template defined functions in the python file to be understood by the html files, we have to use 'Jinja 2' templates defined using '{{ }}', which we have done as shown in the following pictures to define 'user-response' and 'robo-response' as alert messages.



```
<div class="alert alert-danger" role="alert">
<b>Customer: </b> {{user_response}}
</div>
<div class="alert alert-dark" role="alert">
<b>Mr.Swaad: </b> {{robo_response}}
</div>
```

5.6 Working of Hardware

The hardware bot is nothing but a Line-follower robot car with a facility to handle the ordered foot item. The core of the hardware is an Arduino Uno board connected to a motor shield to control movements and programmed in such a way that it follows a specific path up to the kitchen area. For sending response from the webpage to the hardware bot, there is a switch attached at the back-side of the car. On switching it ON, the hardware bot follows the path built up to the kitchen area, waits for a certain time interval and again on switching it on the bot returns to the customer's table with the order. The components required to build the bot are mentioned in the 'List of components' section.

The chassis has parts which are shown in the following image.

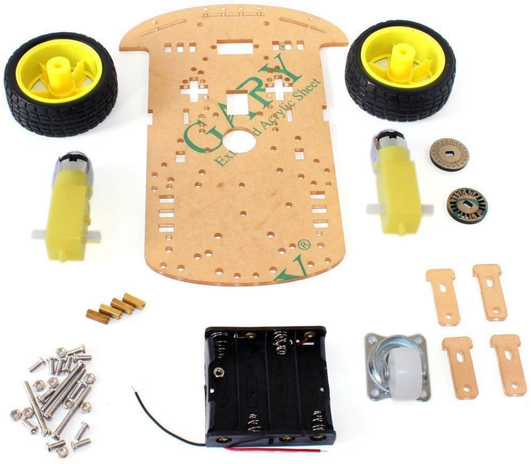


Figure 10: Components of the bot

The parts were then assembled and the bot looks like this.

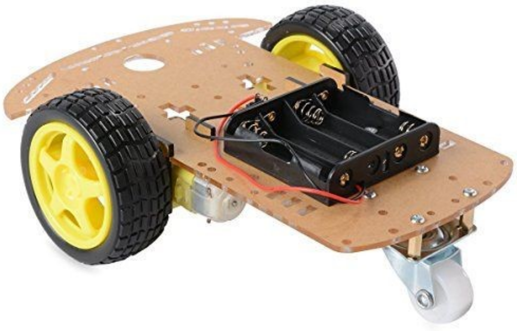


Figure 11: Construction of the Bot

Then, attachment of the components to the chassis was done. Arduino, with the motor shield attached to it and also the battery holder was fixed on the chassis. Also, the IR sensors were attached on the front side of the chassis, facing down. It was made sure that they were attached at the corners. Then the switch was attached too. Soldering of wires to the motors and also switch was done and everything was kept ready.

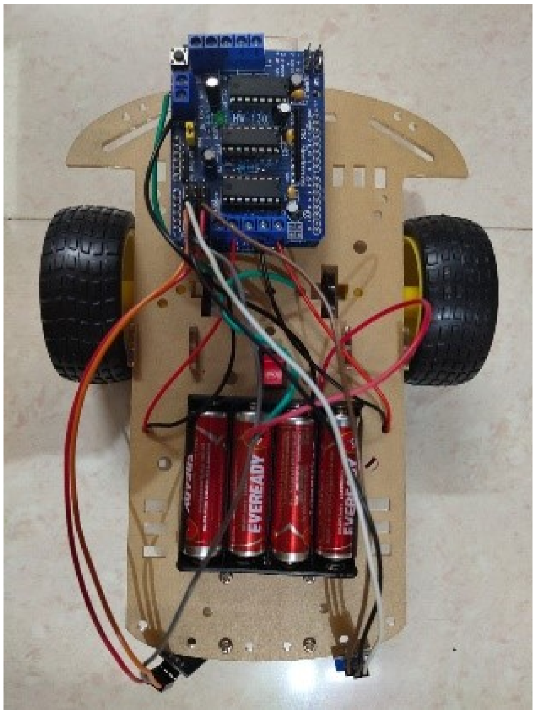


Figure 12: Completed construction of the bot

The main circuit diagram for Arduino via motor shield connection to the two motors is as shown.

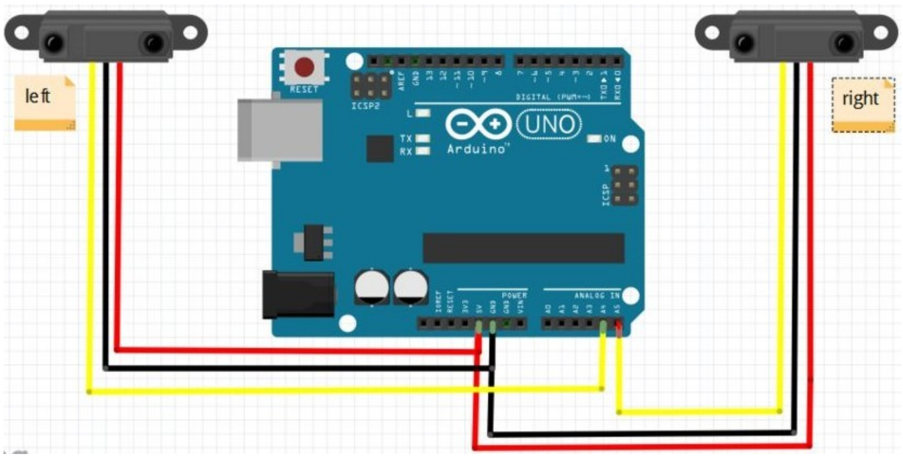


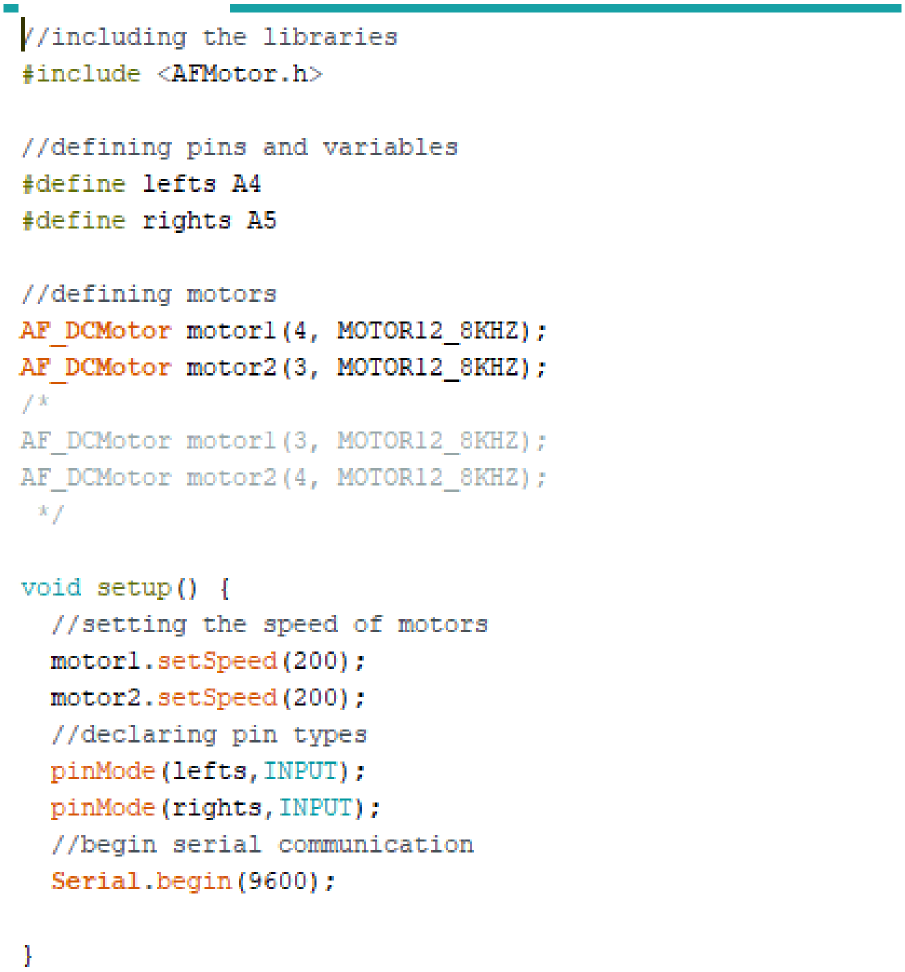
Figure 13: Circuit diagram for Arduino

Left Sensor to Arduino:
VCC > > 5V
Gnd > > Gnd
Out > > A4

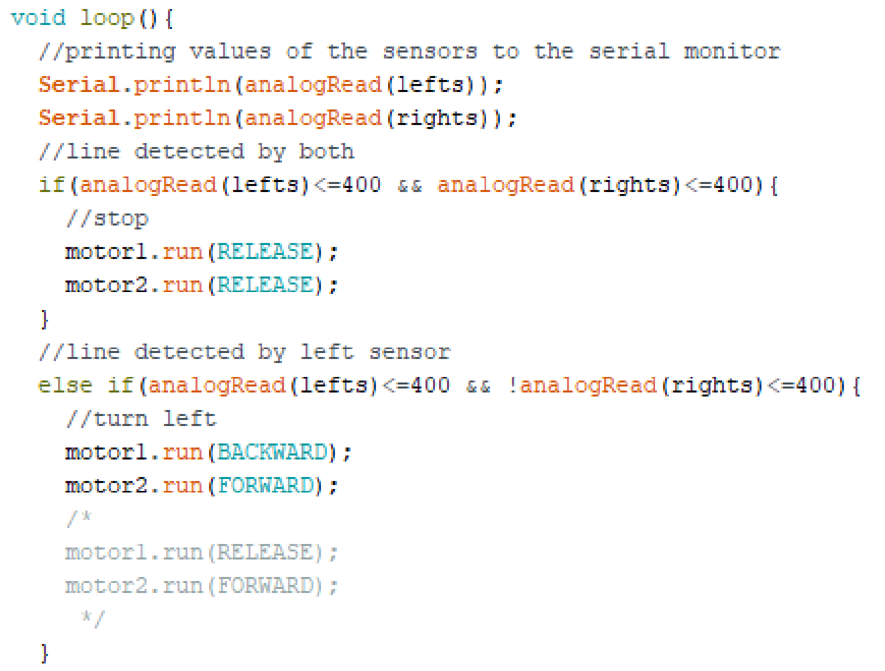
Right Sensor to Arduino:
VCC > > 5V
Gnd > > Gnd
Out > > A5

There are 4 sensor outcomes as follows:
Case 1:-
In this case, both the sensors don't detect the line. Both the motors rotate forward. As a result, the car moves forward.
Case 2:-
In this case, only the left sensor detects the line which means that the car requires to turn in the left direction. The left motor rotates backward and the right motor rotates forward. As a result, the car turns left.
Case 3:-

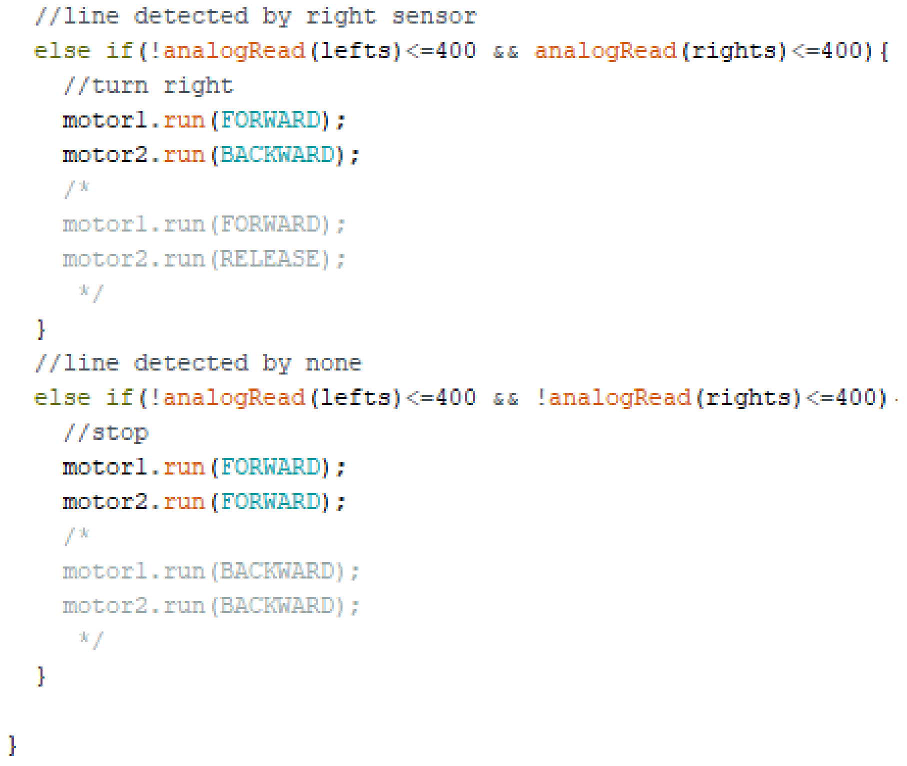
In this case, only the right sensor detects the line which means that the car requires to turn in the right direction. The left motor rotates forward and the right motor rotates backward. As a result, the car turns right.
Case 4:-
In this case, both the sensors detect the line. This means that the end has come. Both the motors stop rotating. As a result, the car stops.
The Arduino code for the functionality of the bot:



```
//including the libraries
#include <AFMotor.h>

//defining pins and variables
#define lefts A4
#define rights A5

//defining motors
AF_DCMotor motor1(4, MOTOR12_8KHZ);
AF_DCMotor motor2(3, MOTOR12_8KHZ);
/*
AF_DCMotor motor1(3, MOTOR12_8KHZ);
AF_DCMotor motor2(4, MOTOR12_8KHZ);
 */

void setup() {
  //setting the speed of motors
  motor1.setSpeed(200);
  motor2.setSpeed(200);
  //declaring pin types
  pinMode(lefts,INPUT);
  pinMode(rights,INPUT);
  //begin serial communication
  Serial.begin(9600);

}
```



```
void loop(){
  //printing values of the sensors to the serial monitor
  Serial.println(analogRead(lefts));
  Serial.println(analogRead(rights));
  //line detected by both
  if(analogRead(lefts)<=400 && analogRead(rights)<=400){
    //stop
    motor1.run(RELEASE);
    motor2.run(RELEASE);
  }
  //line detected by left sensor
  else if(analogRead(lefts)<=400 && !analogRead(rights)<=400){
    //turn left
    motor1.run(BACKWARD);
    motor2.run(FORWARD);
    /*
    motor1.run(RELEASE);
    motor2.run(FORWARD);
     */
  }
```



```
  //line detected by right sensor
  else if(!analogRead(lefts)<=400 && analogRead(rights)<=400){
    //turn right
    motor1.run(FORWARD);
    motor2.run(BACKWARD);
    /*
    motor1.run(FORWARD);
    motor2.run(RELEASE);
     */
  }
  //line detected by none
  else if(!analogRead(lefts)<=400 && !analogRead(rights)<=400)
    //stop
    motor1.run(FORWARD);
    motor2.run(FORWARD);
    /*
    motor1.run(BACKWARD);
    motor2.run(BACKWARD);
     */
  }

}
```

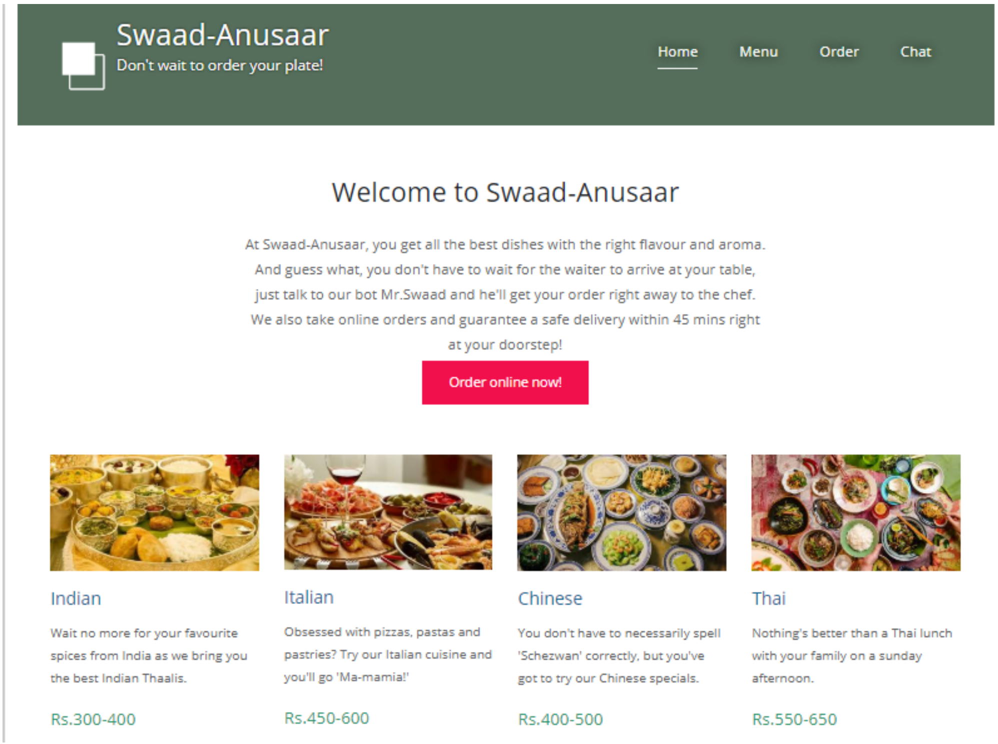
5.7 LIST OF COMPONENTS

SR. NO.	Name of Components	Requiremen
1	Software- Pycharm IDE, Brackets Text Editor, Internet	-
2	Robot car chassis with standard car-kit items	1
3	Arduino Uno board	1
4	Motor shield with driver IC L293D	1
5	Infrared Sensors	2
6	Switch	1
7	1.5V battery	4
8	Connection wires and support items	-

CHAPTER 6

6 RESULTS

6.1 Implementation of Results:



Swaad-Anusaar

Don't wait to order your plate!

Welcome to Swaad-Anusaar

At Swaad-Anusaar, you get all the best dishes with the right flavour and aroma. And guess what, you don't have to wait for the waiter to arrive at your table, just talk to our bot Mr.Swaad and he'll get your order right away to the chef. We also take online orders and guarantee a safe delivery within 45 mins right at your doorstep!

Indian

Wait no more for your favourite spices from India as we bring you the best Indian Thaalis.

Rs.300-400

Italian

Obsessed with pizzas, pastas and pastries? Try our Italian cuisine and you'll go 'Ma-mamia!'

Rs.450-600

Chinese

You don't have to necessarily spell 'Schezwan' correctly, but you've got to try our Chinese specials.

Rs.400-500

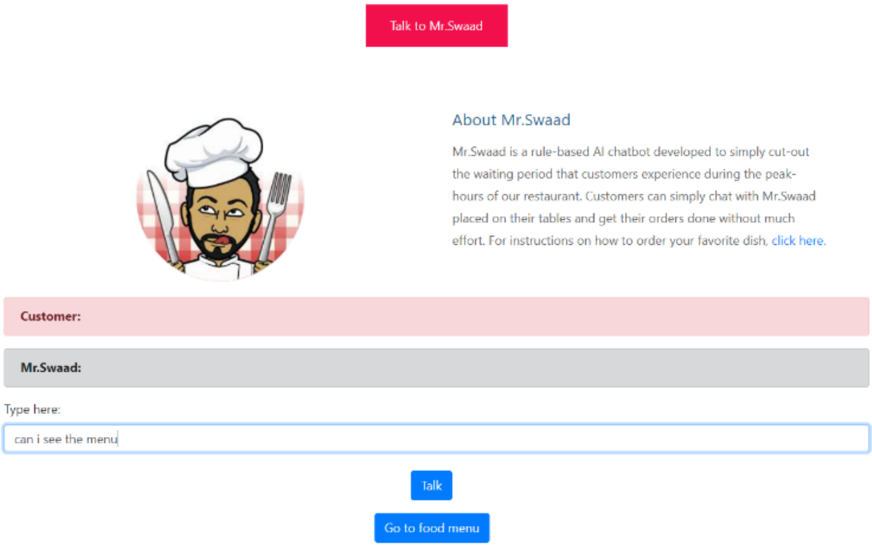
Thai

Nothing's better than a Thai lunch with your family on a sunday afternoon.

Rs.550-650

Figure 14: Restaurant webpage.

The homepage has a section for chatting with the bot as shown below. A user can input anything he/she wishes to and will get an immediate response from the bot which is displayed on the same page. If the user input is well understood by the bot, it returns an appropriate response, although when not understood it returns with a ‘I’m sorry, I can’t understand you’ statement. With our dataset, we have tried to make the bot return with satisfactory responses as much as possible.
As soon as the order is made, the bot can send the signal to the built hardware model. Once a customer is satisfied after talking to the bot, he/she



About Mr.Swaad

Mr.Swaad is a rule-based AI chatbot developed to simply cut-out the waiting period that customers experience during the peak-hours of our restaurant. Customers can simply chat with Mr.Swaad placed on their tables and get their orders done without much effort. For instructions on how to order your favorite dish, click here.

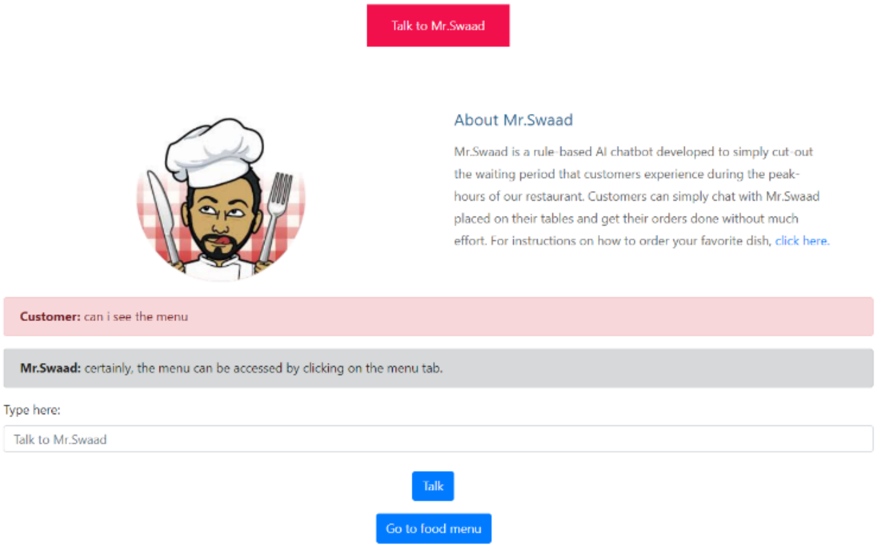
Customer:

Mr.Swaad:

Type here:

can i see the menu

Figure 15: User input/query.



About Mr.Swaad

Mr.Swaad is a rule-based AI chatbot developed to simply cut-out the waiting period that customers experience during the peak-hours of our restaurant. Customers can simply chat with Mr.Swaad placed on their tables and get their orders done without much effort. For instructions on how to order your favorite dish, click here.

Customer: can i see the menu

Mr.Swaad: certainly, the menu can be accessed by clicking on the menu tab.

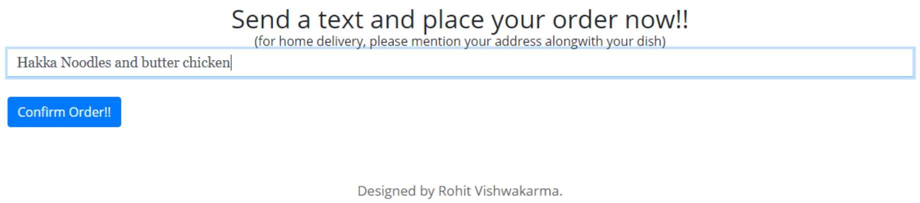
Type here:

Talk to Mr.Swaad

Figure 16: Bot response

can finalize the order by mentioning what they wish to order in the ‘Confirm order’ section of the page.

A text message to the chef is immediately sent onclick with an ‘order



Send a text and place your order now!!

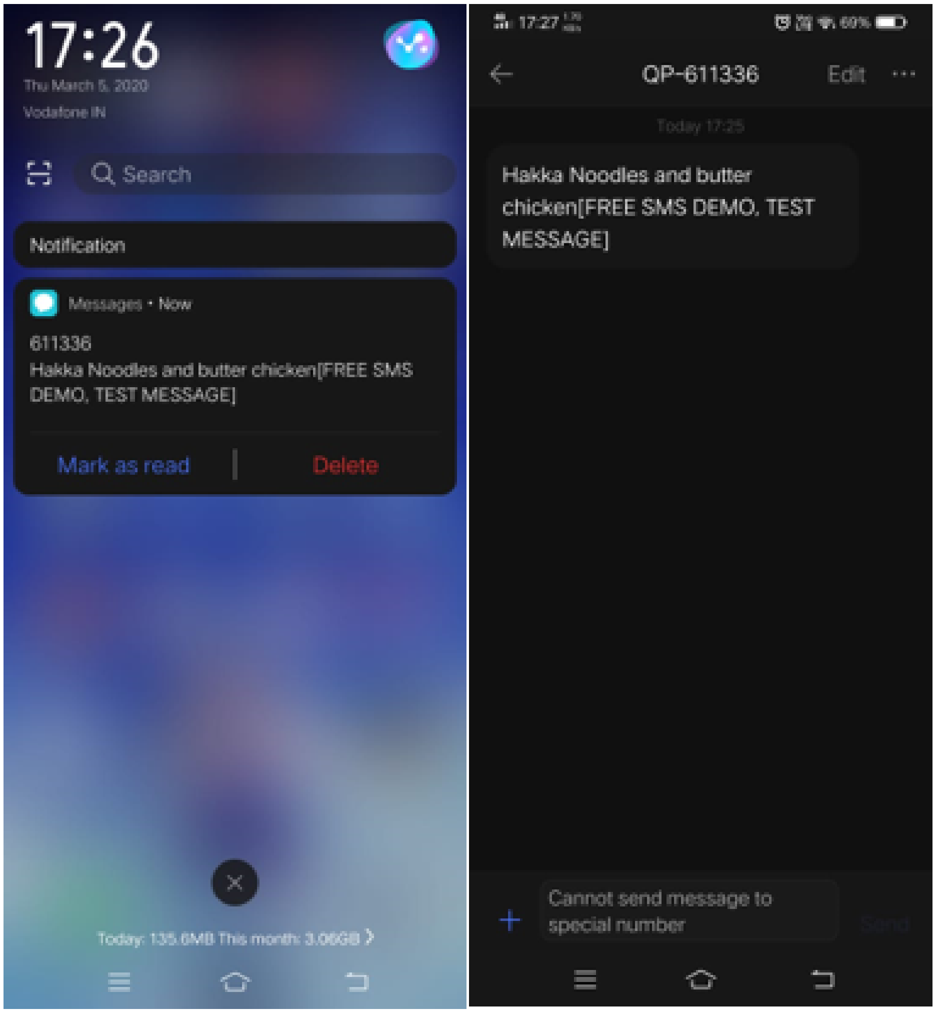
(for home delivery, please mention your address alongwith your dish)

Hakka Noodles and butter chicken

Designed by Rohit Vishwakarma.

Figure 17: Confirming order.

confirmation' dialog box popping up on the customer system. This was done using 'nexmo' library of python which enables us to send/receive messages via any app or webpage. Nexmo allows users to create a free account and a person can send/receive messages, mails, etc. upto a limited allowable amount. For demonstration purposes, one of us has entered their contact number and created a free nexmo account. In the 'Working' section of the previous chapter, one can find among the pictures where we have also imported the 'nexmo' library and routed it to the webpage in the 'send' function of the Flask code. On the website, as soon as a customer clicks on the 'Confirm Order' button as shown in the image above, a text message is received as shown in the following image. This notifies the chef with a time stamp about the received orders. If someone wishes to order online, we have made the website live for the same. For this we have used 'pythonanywhere' platform. This website allows us to have a free non-customized domain name for our website.



611336
Hakka Noodles and butter chicken[FREE SMS DEMO, TEST MESSAGE]

Hakka Noodles and butter chicken[FREE SMS DEMO, TEST MESSAGE]

Figure 18: Order notification message on chef's mobile phone

6.2 HARDWARE OF THE PROJECT

The hardware bot, built using IoT technology, is basically a wireless car with IR (infrared sensors) at the front to detect the path to follow. The core of the hardware is an Arduino Uno board connected to a motor shield to control movements and programmed in such a way that it follows a specific path up to the kitchen area. For sending response from the webpage to the hardware bot, there is a switch attached at the back-side of the car. On switching it ON, the hardware bot follows the path built up to the kitchen area, waits for a certain time interval and again on switching it on the bot returns to the customer's table with the order.

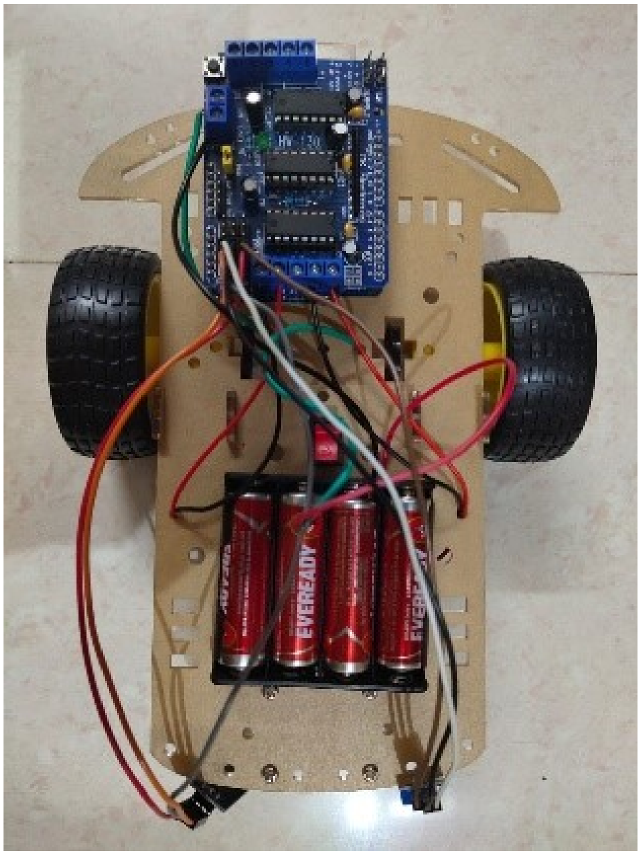


Figure 19: Path follower Bot

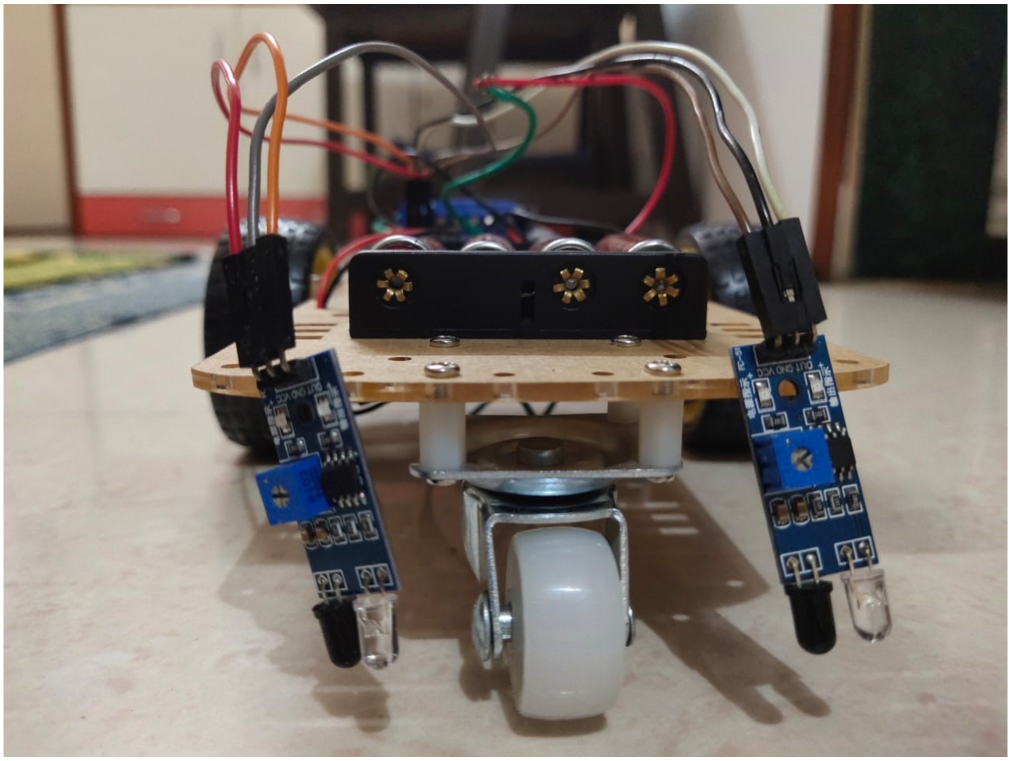


Figure 20: Path follower hardware bot prototype

CHAPTER 7

7 CONCLUSION AND FUTURE SCOPE

7.1 Advantages

1) Customer service is Improved.
2) Monitoring of Customer data becomes easier and much less complicated.
3) Cost saving systems.
4) Keeping track of billing information at a push of a button.
5) Pre-ordering of the food is possible before reaching the Restaurant.
6) Feedback can be directly given to the chef or even the Manager of the Restaurant.
7) Multiple Admin can keep a track of the service from a remote location.

7.2 Future Scope

1 Here we have proposed an attempt of implementation of Chatbot in restaurants and various food corners. We are in a era where we are simply surrounded by various smart chatbots.
2. Today everything from booking a ticket to launcing a missile, is based on Artificial intelligence and Machine Learning Implementation of this makes our life easier and this also helps to save time in this busy life today.
3. Chatbots can effectively increase the ability of customer satisfaction and will improve sales. It is observed that the coustemer expirence gets even better when they have an virtual assistance to help them with all the help they need while ordering the food they want.
4. In future work we may include the functionality in such a way that the chatbots will be able to manage personalized accounts of the customers and will be able to manger their order history and will make it easier for them to keep a track of the payments made and the food ordered.
5. Payments and bills will be taken care off by the chatbot itself. Since there is a Vast field of implementation of chatbots, it is hard to imagine a future without a chatbot.
6. A Feedback can be given by the customers to the robot itself.

7.3 Conclusion:

In this Report, information about the design and implementation of a chatbot in food industry has been presented. The proposed system will not only be helpful to reduce the waiting period in restaurants, but will also cut costs required for manpower for food industries and restaurant owners. With technological advent, it is also an innovative method to attract younger audience thereby increasing sales. This can take us one step Forward towards 'The Digital India' Movement. Implementations of this project will lead to advancement to the Restaurant industry and Food industry as well.

8 REFRENCES

[1] 'Computational Linguistics': ELIZA- A Computer Program for the study of Natural Language Communication between Man and Machine - Joseph Weizenbaum, published in January, 1966.
[2] 'How bots can solve the 5 most common restaurant industry challenges' – Riya Savjani, published on September 29, 2016.
[3]Matthew Reed, Talking ELIZA, 2007, Cover of instruction manual ELIZA, accessed 24 july, 2019, http://www.trs-80.org/talking-eliza/
[4]N. Landsteiner , Eliza, 2005, Joseph Weizenbaum using ELIZA, accessed 5 August, 2019, https://www.masswerk.at/elizabot/
[5] Bayan Abu Shawar, Eric Atwell, Chatbots: Are they useful?, 2007, A sample of chatting with ALICE, accessed 9 August 2019, https://www.semanticscholar.org/paper/Chatbots

[6] 'Starbucks Bot Steps Up the Coffee Experience' – Deena Zaidi, published on October 13, 2017.
[7] Shamim Al Mamun, Review of integrated applications with AIML based chatbot, 2015, accessed 9 Augut, 2019, https://www.researchgate.net/profile/Shamim-Al-Mamun2
[8] Lauren Johnson, Starbucks' App Now Includes a Voice-Activated Ordering Bot , 2017, 5 Photo of Chat Bot Guide, accessed 25 August, 2019,https://www.chatbotguide.org/starbucks-bot
[9] 'Whole Foods just launched a Messenger chatbot for finding recipes with emojis' – John Brandon, published on July 12, 2016.
[10] 'Domino's introduces 'Dom the Pizza Bot' for Facebook Messenger' – Nikki Gilliland, published on August 12, 2016
. [11] Menal Dahiya, a tool of conversation: CHATBOT, 2017, Sequence Diagram Representing Design of the Chatbot, accessed 21 July, 2019, https://www.researchgate.net/figure/Sequence-Diagram-Representing-Design-of-the-Chatbot-fig1-321864990
[12] 'TFIDF — TFIDF Python Example' – Cory Maklin, published on May 5, 2019.
[13] Artem Kuchumov, Dataset of English hyponyms wordnet 2018, wordnet, accessed 3 September, 2019, https://www.kaggle.com/duketemon/hyponyms-wordnet
[14] DATAFLAIR TEAM, NLTK Python Tutorial (Natural Language Toolkit), 2018, NLTK Python Tutorial (Natural Language Toolkit), accessed 13 September 2019, https://data-flair.training/blogs/nltk-python-tutorial/

[15] ‘Building a Simple Chatbot from Scratch in Python (using NLTK)’-Parul Pandey, published on September 17,2018.