

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(deemed to be university)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in
Information Technology

By

BENSON JOEL.B
(Reg. No. 37120014)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI- 600119 TAMIL NADU



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344

www.sathyabama.ac.in



AUGUST 2020

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonfide work of BENSON JOEL.B (Reg. No. 37120014) who underwent the professional training in "SIMPLE CHATBOT APPLICATION USING PYTHON" under our supervision from July 2020 to AUGUST 2020.

Internal Guide

DR.S.REVATHY,M.E., Ph.D

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I, BENSON JOEL .B (Reg. No. 37120014) hereby declare that the Professional Training Report on “SIMPLE CHATBOT APPLICATION USING PYTHON” done by me under the guidance of DR.S.REVATHY,M.E., Ph.D at Sathyabama Institute Of Science And

Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of technology degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. T.Sasikala M.E., Ph.D, Dean, School of Computing , Dr.SUBHASHINI M.E., Ph.D. , Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide DR.S.REVATHY,M.E., Ph.D,for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Information Technology who were helpful in many ways for the completion of the project.

ABSTRACT

A chatbot is a computer program that can converse with humans using artificial intelligence in messaging platforms. The goal of the project is to add a chatbot feature and API for You. discussion groups, blogs, wikis etc. Yioop provides all the basic features of web search portal. It has its own account management system with the ability to set up groups that have discussions boards. Groups are collections of users that have access to a group feed. The user who creates a group is set as the initial group owner. Posts are grouped by thread in a group containing the most recent activity at the top. The chatbot API for Yioop will allow developers to create new chatbots, powered by rules or artificial intelligence, that can interact like a human with users in a groups feed page. Example chatbots that can be developed with this API is weather chatbots or book flight chatbots. Over past few years, messaging applications have become more popular than Social networking sites. People are using messaging applications these days such as Facebook Messenger, Skype, Viber, Telegram, Slack etc. This is making other businesses available on messaging platforms leads to proactive interaction with users about their products. To interact on such messaging platforms with many users, the businesses can write a computer program that can converse like a human which is called a chatbot.



TABLE OF CONTENT

CHAPTER	TITLE	PAGE NO
1	1.1 INTRODUCTION	12
	1.2 CHATBOT AND MACHINE LEARNING	13
	1.3 ARTIFICIAL INTELLIGENCE	14
	1.4 AI APPLICATION	15
2	BACKGROUND AND LITERATURE REVIEW	16
	2.1 LITERATURE REVIEW	16
	2.2 NATURAL LANGUAGE PROCESSING	17
	2.3 MACHINE LEARNING	17
3	PROBLEM IDENTIFICATION	18
4	REQUIREMENTS AND SPECIFICATION	19
	4.1 HARDWARE REQUIREMENT	19
	4.2 SOFTWARE REQUIREMENT	19
5	MODULE DESCRIPTION	20
	5.1 AIML SCRIPTING	20
	5.2 CREATING A STARTUP FILE	21
	5.3 CREATING INTERFACE	22
	5.4 SPPEDING BRAIN LORD	23
	5.5 LOADING BRAIN	24
6	ADVANTAGES AND DISADVANTAGES	25
	6.1 ADVANTAGES	25
	6.2 DISADVANTAGES	27
7	CONCLUSION AND SCOPE FUTURE	29
	7.1 CONCLUSION	29

7.2 SCOPE FOR FUTURE	29
7.3 SOURCECODE	30
7.4 SCREENSHOT	31
REFERENCES	32

LIST OF ABBREVIATIONS

ACH	Automated Clearing House
ISP	Internet service provide
OCC	Open Cash Credit
ATM	Automatic Teller Machine
JDBC	Java Database Connectivity
JSP	Java Server Page
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
IDE	Integrated Development environment
DFD	Data flow Diagram
UML	Unified Modelling language
SQL	Structure Query Language
DML	Data Manipulation Language
GUI	Graphic User Interface
DBS	Database Systems
J2EE	Java 2 Platform Enterprise Edition
J2SDK D	JAVA 2 SDKD

LIST OF FIGURES

Figure No.	Caption
Figure 2.1	Regression and Classification
Figure 5.1	Aiml Script
Figure 5.2	Loading AIML
Figure 5.3	Creating Interface
Figure 5.4	Speeding Up brain
Figure 5.5	Loading Brain
Figure 7.4	Screenshot program
Figure 7.4.1	Screenshot output

LIST OF TABLES

Table No.	Caption	Page No.
Table 2.1	Literature Survey	7

CHAPTER 1

1.1 INTRODUCTION

A chatbot is an automated software program that interacts with humans. A chatbot is merely a computer program that fundamentally simulates human conversations. A chatbot that functions through AI and machine learning has an artificial neural network inspired by the neural nodes of the human brain. Chatbots are programs that can do talk like human conversations very easily. For example, Facebook has a machine learning chatbot that creates a platform for companies to interact with their consumers through the Facebook Messenger application. In 2016, chatbots became too popular on Messenger. By the consequences is noted that 2016 was the entire year of chatbots. The software industry is mainly oriented on chatbots. Thousands of chatbots are invented on startups and used by the businesses to improve their customer service, keeping them hanging by a kind communication. According to research, nowadays chatbots are used to solve a number of business tasks across many industries like E-Commerce, Insurance, Banking, Healthcare, Finance, Legal, Telecom, Logistics, Retail, Auto, Leisure, Travel, Sports, Entertainment, Media and many others. Thus that was the moment to look at the chatbots as a new technology in the communication field. Nowadays various companies are using chatbots to answer quickly and efficiently some frequented asking questions from their own customers. AIML and LSA are used for creating chatbots. Artificial Intelligence Markup Language (AIML) and Latent Semantic Analysis (LSA) are used for developing chatbots, which are used to define general pattern-based queries. This pattern can also be used to give random responses for the same query in the chatbot. LSA is a Latent Semantic Analysis technology in python, which is utilized to discover likenesses between words as vector representation. So that the unanswered queries by AIML will be viewed as a reply by LSA.

1.2 Chatbot and Machine learning :

Machine learning chatbots works using artificial intelligence. User need not to be more specific while talking with a bot because it can understand the natural language, not only commands. This kind of bots get continuously better or smarter as it learns from past conversations it had with people.

Here is a simple example which illustrate how they work. The following is a conversation between a human and a chatbot: Human: "I need a flight from San Jose

to New York.” Bot: “Sure! When would you like to travel?” Human: “From Dec 20, 2016 to Jan 28, 2017.” Bot:

In order to achieve the ultimate goal, I have taken an iterative approach and divided my work into four major deliverables. These deliverables not only helped me in understanding the code structure of Yioop but also enhances Yioop’s functionality. In the rest of the report, I will be discussing about the four deliverables. To understand more on chatbot service, I had implemented a Facebook Messenger Weather Bot in deliverable 1, which is discussed in next section. The purpose of deliverable 2 is to introduce chatbots to the Yioop. I have added Bot Configuration settings which is used to add bot users in Yioop. In the next deliverable, I have added a functionality where the user will be able to call bots in a group thread. Activation of bots will happen by calling respective callback URL which is already configured that helps bots to have a conversation with users. More details on this is discussed in deliverable 3 section. As a deliverable 4, I have created a weather bot i.e, a web application in php that calls yahoo API to get weather information. The last section of the report contains the conclusion and future work.

I have implemented a Facebook Messenger Bot to get an overview of how chatbot is build. During this implementation, I understood the flow of control for a chatbot service with other services which is explained below.

In order to create a Facebook Messenger Bot, a developer needs to be authenticated and approved by Facebook to converse with the public and the web server for security reasons. For a Facebook Messenger Bot, I have created a simple web application using Node.js by installing the necessary dependencies using npm. I ran this locally. I also downloaded and installed ngrok and started it - npm run ngrok. This launched a Forwarding URL to the local running server, that means any requests to Forwarding URL will hit the locally running server. This url is used as a Callback URL in Facebook App which will be explained further. To set up the Facebook App, I have created a Facebook

Page and Facebook App using my Facebook account. While setting up a Webhook in the app settings, I have given the Forwarding URL as Callback URL and added code for verification. The access token in page settings is stored as environment

variable as it will be used in integration. In order to make webhook to receive messages from this page, the app is subscribed to the page created. To set up the bot to handle the POST calls at webhook, I have created a webhook endpoint in the sample application.

1.3 Artificial Intelligence

AI was coined by John McCarthy, an American computer scientist, in 1956 at The Dartmouth Conference where the discipline was born. Today, it is an umbrella term that encompasses everything from robotic process automation to actual robotics. It has gained prominence recently due, in part, to big data, or the increase in speed, size and variety of data businesses are now collecting. AI can perform tasks such as identifying patterns in the data more efficiently than humans, enabling businesses to gain more insight out of their data.

Types of artificial intelligence

AI can be categorized in any number of ways, but here are two examples.

The first classifies AI systems as either weak AI or strong AI. Weak AI, also known as narrow AI, is an AI system that is designed and trained for a particular task. Virtual personal assistants, such as Apple's Siri, are a form of weak AI.

Strong AI, also known as artificial general intelligence, is an AI system with generalized human cognitive abilities so that when presented with an unfamiliar task, it has enough intelligence to find a solution. The Turing Test, developed by mathematician Alan Turing

in 1950, is a method used to determine if a computer can actually think like a human, although the method is controversial.

The second example is from Arend Hintze, an assistant professor of integrative biology and computer science and engineering at Michigan State University. He categorizes AI into four types, from the kind of AI systems that exist today to sentient systems, which do not yet exist.

Type 1: Reactive machines.

An example is Deep Blue, the IBM chess program that beat Garry Kasparov in the 1990s. Deep Blue can identify pieces on the chess board and make predictions, but it has no memory and cannot use past experiences to inform future ones. It analyzes possible moves -- its own and its

opponent -- and chooses the most strategic move. Deep Blue and Google's AlphaGO were designed for narrow purposes and cannot easily be applied to another situation.

Type 2: Limited memory.

These AI systems can use past experiences to inform future decisions. Some of the decisionmaking functions in autonomous vehicles have been designed this way. Observations used to inform actions happening in the not-so-distant future, such as a car that has changed lanes. These observations are not stored permanently.

Type 3: Theory of mind.

This is a psychology term. It refers to the understanding that others have their own beliefs, desires and intentions that impact the decisions they make. This kind of AI does not yet exist.

Type 4: Self-awareness. In this category,

AI systems have a sense of self, have consciousness. Machines with self-awareness understand their current state and can use the information to infer what others are feeling. This type of AI does not yet exist.

CHAPTER 2

Background And Literature Review

2.1 Literature Review

Purpose	Papers	Year of publications	Pros	Cons
Usefulness of chatbots	Bayan Abu Shawar and Eric Atwell, 2007 "Chatbots: Are they Really Useful?"	2007	• Significance of Chatbot	Machine like Response
Natural language processing	Journal for Computational Linguistics and Language	2007	• Benefit of natural language processing in any AIML chatbot • Detailed overview of Natural language processing mechanism	• Technical and obsolete problems in Nlp mechanism
Real time solution for Open learner model	Towards natural language negotiation of open learner models	2009	• Use of technology to enhance response.	• Limited information • Problem in integration
Intelligent tutorial system	Whitepaper for the Army's Science of Learning Workshop, Hampton	2006	• Detailed description about Python and AIML	• Only for chatbot based on Aiml
Aiml and Brain loading	AI Chat Bot with AIML Submitted by NanoDano	2015	• Effective way of installation.	• Slow brain loading

2.2 Natural Language Processing

Natural Language Processing (NLP) is the study of letting computers understand human languages[3]. Without NLP, human language sentences are just a series of meaningless symbols to computers. Computers don't recognize the words and don't understand the grammars. NLP can be regarded as a "translator", who will translate human languages to computer understandable information.

Traditionally, users need to follow well-defined procedures accurately, in order to interact with computers. For example, in Linux systems, all commands must be precise. A single replace of one character or even a space can have significant difference. However, the emergence of NLP is changing the way of interacting. Apple Siri and Microsoft Cortana have made it possible to give command in everyday languages and is changing the way of interacting.

2.3 Machine Learning.

Machine Learning (ML) is an area of computer science that "gives computers the ability to learn without being explicitly programmed". The parameter of the formulas is calculated from the data, rather than defined by the programmer. Two most common usage of ML is Classification and Regression. As shown in figure1[8], Classification means to categorize different types of data, while Regression means to find a way to describe the data. Basic ML program will have two stages, **fitting** and **predicting**. In the fitting stage, the program will be given a large set (at least thousands) of data. The program will try to adjust its parameter based on some statistical models, in order to make it "fit" the input data best. In the predicting stage, the program will give a prediction for a new input based on the parameters it just calculated out. For example, the famous Iris flower dataset [9] contains the measurement of several features of three different species of flowers, such as the length of sepals and petals. A well-defined ML program can learn the pattern behind this feature and give prediction accordingly.

CHAPTER 3

PROBLEM IDENTIFICATION

3.1 Problem Identification-

Changing of requirements As an industrial project to build a product, we must follow the requirement from the user. However, because the project's goal is to be used by the business team, but it is responsible by the technical team, the requirement changed a lot in the middle after a meeting with the business team. The business team want a simple bot that can give recommendation immediately. We had to archive what we had done before and build a new one.

- **Lacking of training data.** The quantity and quality of training data is critical to the performance to a machine learning model. However, because some confidential and privacy reasons, the business team cannot provide enough data for us, and we had to make up data by our own. For the machine learning model, we generate some fake data based on our daily life experience, which is really biased, although with a good accuracy on the fake dat.
- **Unstable API version.** Because API service we are using are still under development, and we cannot fix to a version for the API, the API may changes overtime. Moreover, there are inconsistencies between the APIs and their documents or sample codes.

CHAPTER 4

REQUIREMENTS AND SPECIFICATION

4.1 Hardware Requirements:

Processor : Pentium IV(minimum)
Hard Disk : 40GB
RAM : 256MB (minimum)

4.2 Software Requirements:

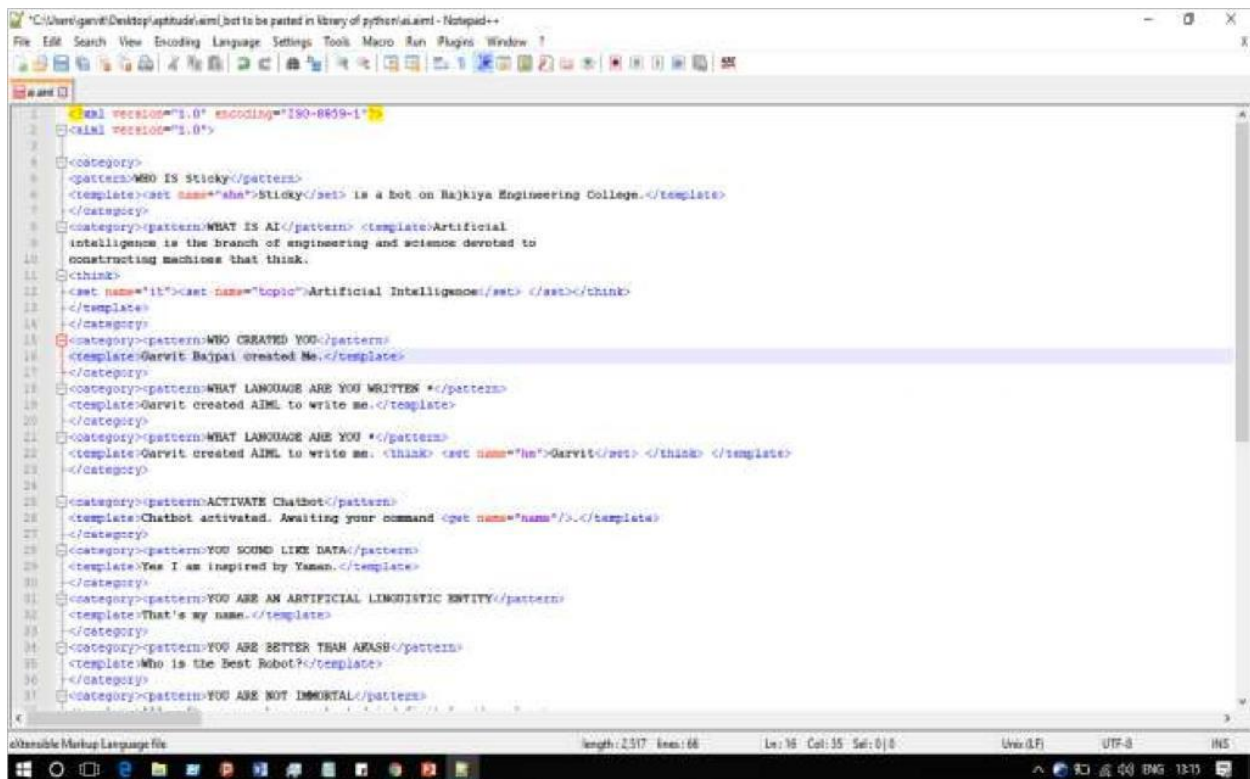
Operating System : Windows or Linux
Technology : PYTHON, AIML

CHAPTER 5

MODULE DESCRIPTION

5.1 AIML Scripting

we created the AIML file that only handles one pattern, load aiml b. When we enter that command to the bot, it will try to load basic_chat.aiml. It won't work unless we actually create it. Here is what you can put inside basic_chat.aiml. We will match two basic patterns and respond.



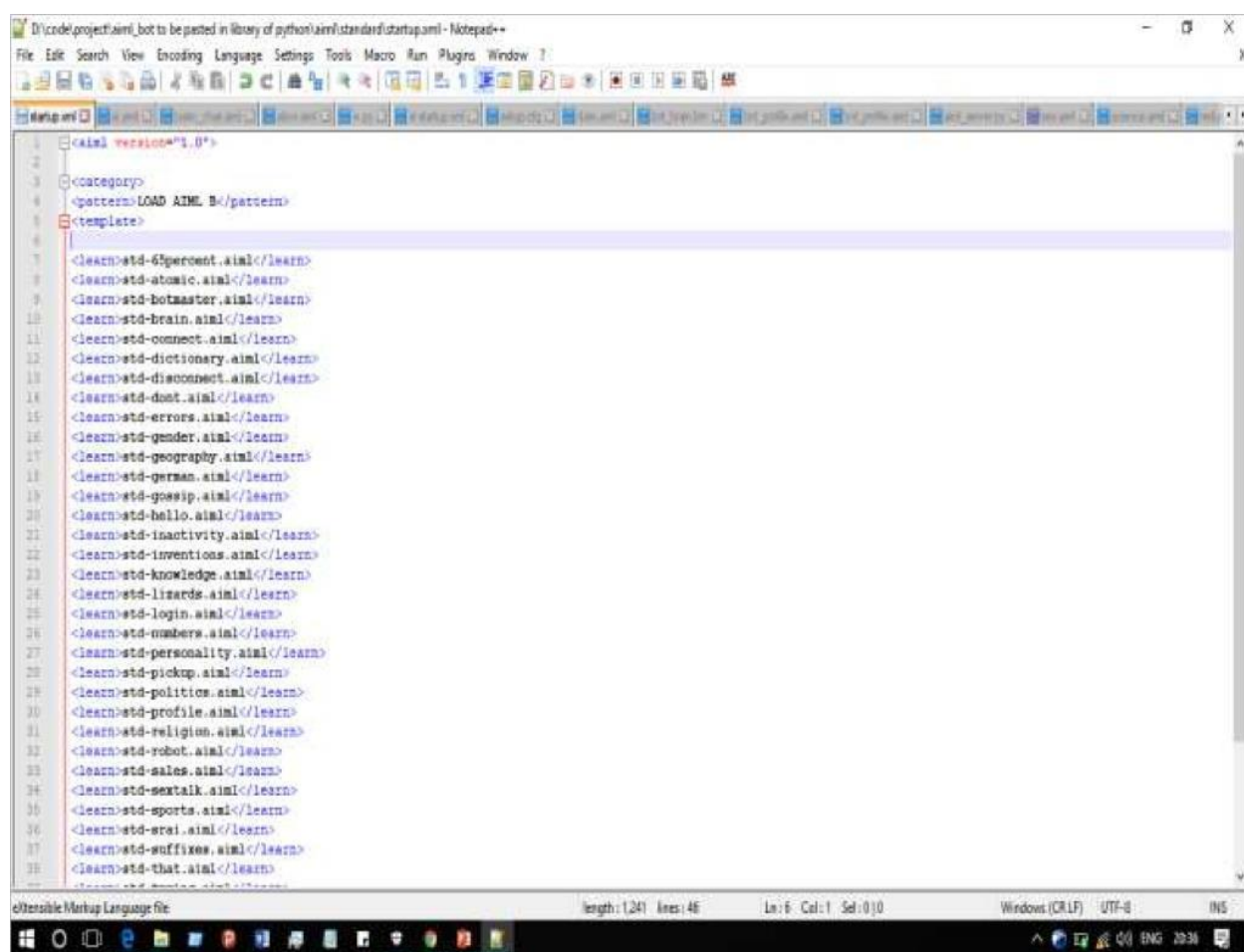
```
1 <aiml version="1.0" encoding="ISO-8859-1">
2 <load version="1.0">
3
4 <category>
5 <pattern>WHO IS Sticky</pattern>
6 <template><set name="aha">Sticky</set> Is a bot on Rajkya Engineering College.</template>
7 </category>
8
9 <category><pattern>WHAT IS AI</pattern> <template>Artificial
10 intelligence is the branch of engineering and science devoted to
11 constructing machines that think.
12 </template>
13 <think>
14 <set name="it"><set name="topic">Artificial Intelligence</set> </set></think>
15 </template>
16 </category>
17
18 <category><pattern>WHO CREATED YOU</pattern>
19 <template>Garvit Bajpai created Me.</template>
20 </category>
21
22 <category><pattern>WHAT LANGUAGE ARE YOU WRITTEN *</pattern>
23 <template>Garvit created AIML to write me.</template>
24 </category>
25
26 <category><pattern>WHAT LANGUAGE ARE YOU *</pattern>
27 <template>Garvit created AIML to write me. <think> <set name="he">Garvit</set> </think> </template>
28 </category>
29
30 <category><pattern>ACTIVATE Chatbot</pattern>
31 <template>Chatbot activated. Awaiting your command <get name="name"/>.</template>
32 </category>
33
34 <category><pattern>YOU SOUND LIKE DATA</pattern>
35 <template>Yes I am inspired by Yamen.</template>
36 </category>
37
38 <category><pattern>YOU ARE AN ARTIFICIAL LINGUISTIC ENTITY</pattern>
39 <template>That's my name.</template>
40 </category>
41
42 <category><pattern>YOU ARE BETTER THAN ARASH</pattern>
43 <template>Who is the Best Robot?</template>
44 </category>
45
46 <category><pattern>YOU ARE NOT IMMORTAL</pattern>
47 </category>
```

Fig3.1 AIML Script

5.2 Creating a Startup File-

It is standard to create a startup file called std-startup.xml as the main entry point for loading AIML files. In this case we will create a basic file that matches one pattern and takes one action. We want to match the pattern load aiml b, and have it load our aiml brain in response.

We will create the basic_chat.aiml file in a minute.



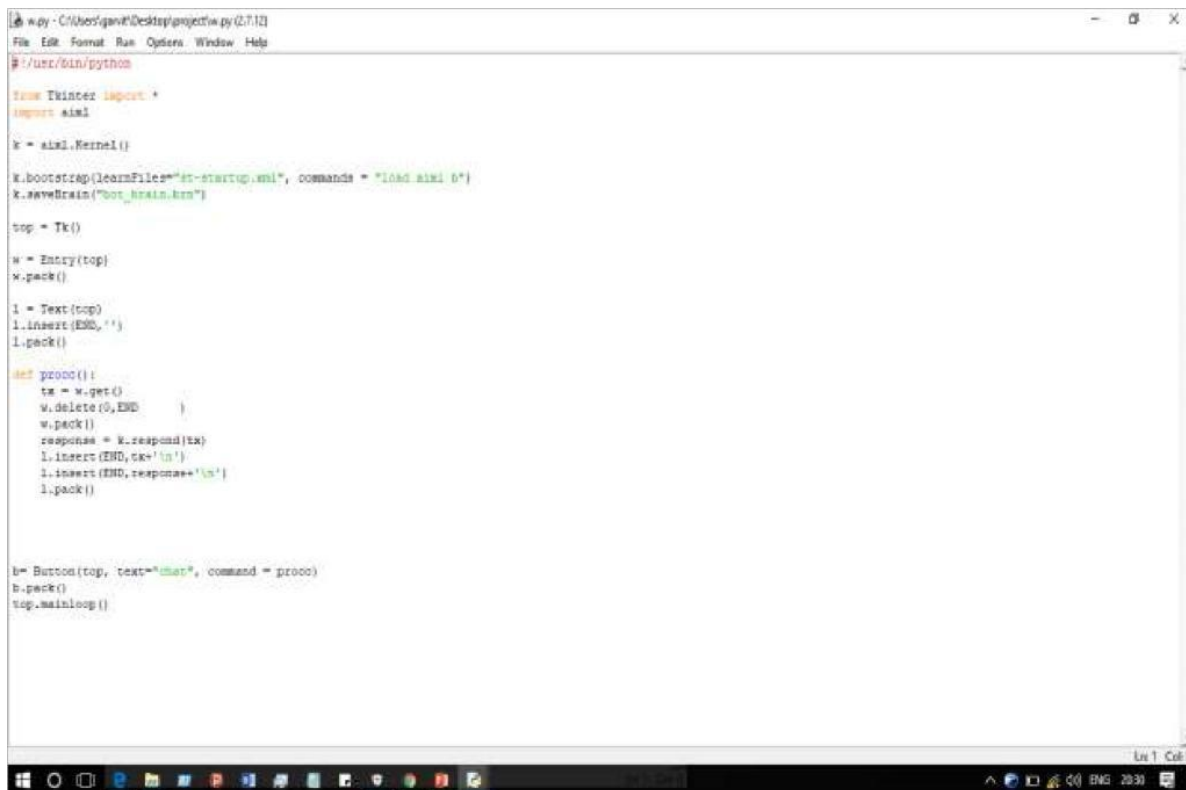
```
<aiml version="1.0">
  <category>
    <pattern>LOAD AIML B</pattern>
    <template>
      <learn>std-6percent.aiml</learn>
      <learn>std-atomic.aiml</learn>
      <learn>std-botmaster.aiml</learn>
      <learn>std-brain.aiml</learn>
      <learn>std-connect.aiml</learn>
      <learn>std-dictionary.aiml</learn>
      <learn>std-disconnect.aiml</learn>
      <learn>std-dust.aiml</learn>
      <learn>std-errors.aiml</learn>
      <learn>std-gender.aiml</learn>
      <learn>std-geography.aiml</learn>
      <learn>std-german.aiml</learn>
      <learn>std-gossip.aiml</learn>
      <learn>std-hello.aiml</learn>
      <learn>std-inactivity.aiml</learn>
      <learn>std-inventions.aiml</learn>
      <learn>std-knowledge.aiml</learn>
      <learn>std-lizards.aiml</learn>
      <learn>std-login.aiml</learn>
      <learn>std-numbers.aiml</learn>
      <learn>std-personality.aiml</learn>
      <learn>std-pickup.aiml</learn>
      <learn>std-politics.aiml</learn>
      <learn>std-profile.aiml</learn>
      <learn>std-religion.aiml</learn>
      <learn>std-robot.aiml</learn>
      <learn>std-sales.aiml</learn>
      <learn>std-sextalk.aiml</learn>
      <learn>std-sports.aiml</learn>
      <learn>std-urai.aiml</learn>
      <learn>std-suffixes.aiml</learn>
      <learn>std-that.aiml</learn>
    </template>
  </category>
</aiml>
```

Fig5.2 Loading Aiml

5.3 Creating Interface-

As shown in the figure, there are six components in our project. The **interfaces** are the front end chat box for user to talk to the bot, which can be the Bot Portal, Skype, Facebook, etc. The **connector** works as a common gateway for all the interfaces. The outbound side calls different APIs to different front end, but the inbound APIs kept the same for our bot to connect. Fortunately, this connector has already been implemented by the bot framework SDK, we only need to rightly configure them.

The **bot** part contains the main flow control of our project. It is responsible for redirect the input to different models, parse the return values, and determines what to do next. It is also connected to the database to retrieve and update values.

A screenshot of a Python IDE window titled 'w.py - C:\Users\gan\Desktop\project\w.py (2.7.12)'. The window contains a Python script for a chat interface using Tkinter. The script imports Tkinter and a custom 'aiml' module. It initializes an 'aiml.Kernel()' object 'k', boots it with a 'learnFile' and 'commands', and saves the brain. A Tk window 'top' is created with an Entry widget 'w' and a Text widget 'l'. A 'proc()' function is defined to handle button clicks: it gets the text from 'w', deletes the current content in 'l', gets a response from 'k', and inserts the response into 'l'. A Button widget 'b' is added to 'top' with the text 'chat' and the 'proc()' command. Finally, 'top.mainloop()' is called. The Windows taskbar is visible at the bottom.

```
w.py - C:\Users\gan\Desktop\project\w.py (2.7.12)
File Edit Format Run Options Window Help

# /usr/bin/python

from Tkinter import *
import aiml

k = aiml.Kernel()

k.bootstrap(learnFile="st-startup.xml", commands = "load aiml 0")
k.saveBrain("bot_brain.knn")

top = Tk()

w = Entry(top)
w.pack()

l = Text(top)
l.insert(END, '')
l.pack()

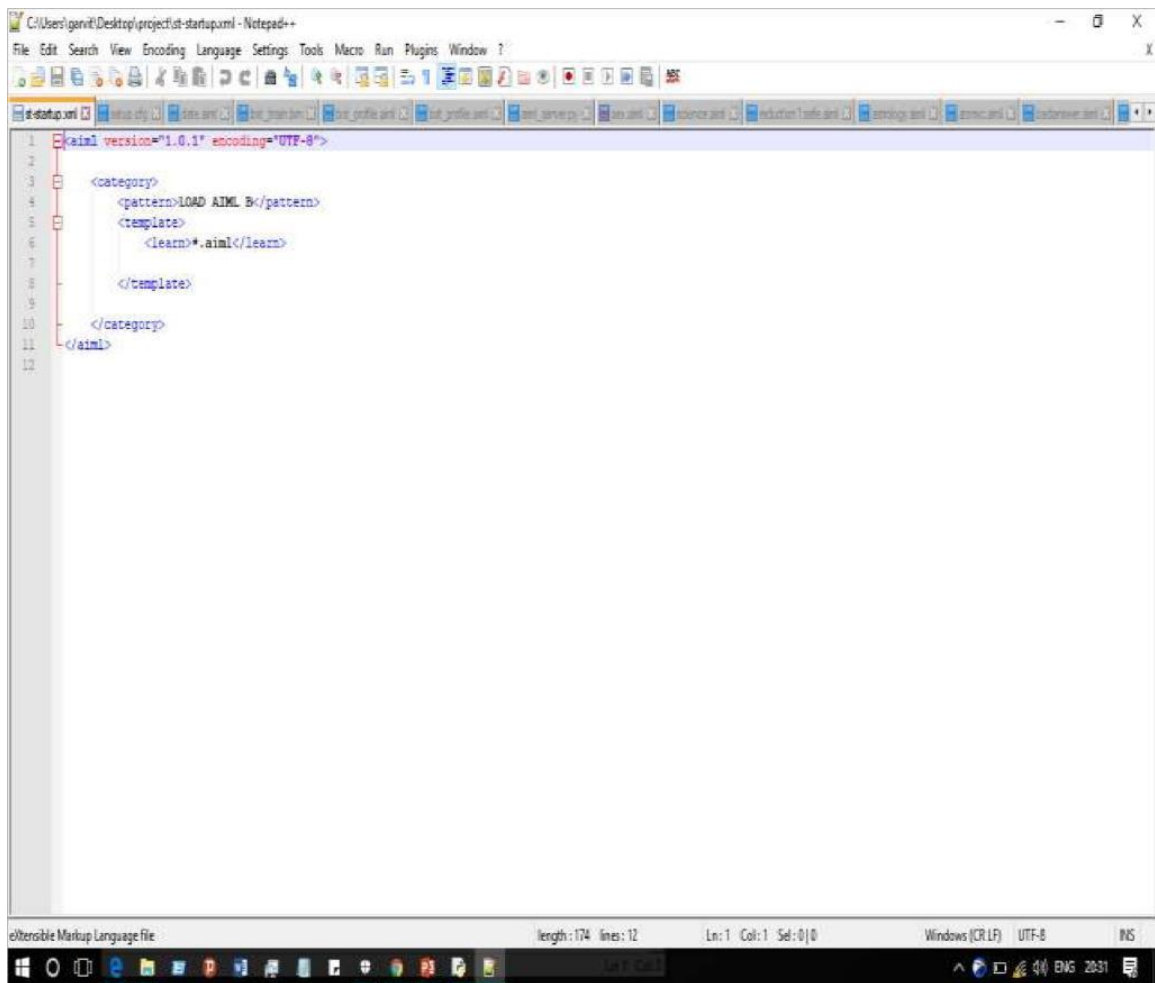
def proc():
    tx = w.get()
    w.delete(0,END)
    w.pack()
    response = k.respond(tx)
    l.insert(END, tx+'\n')
    l.insert(END, response+'\n')
    l.pack()

b= Button(top, text="chat", command = proc)
b.pack()
top.mainloop()
```

Fig 5.3 creating Interface

5.4 Speeding up Brain Load

When you start to have a lot of AIML files, it can take a long time to learn. This is where brain files come in. After the bot learns all the AIML files it can save its brain directly to a file which will drastically speed up load times on subsequent runs.

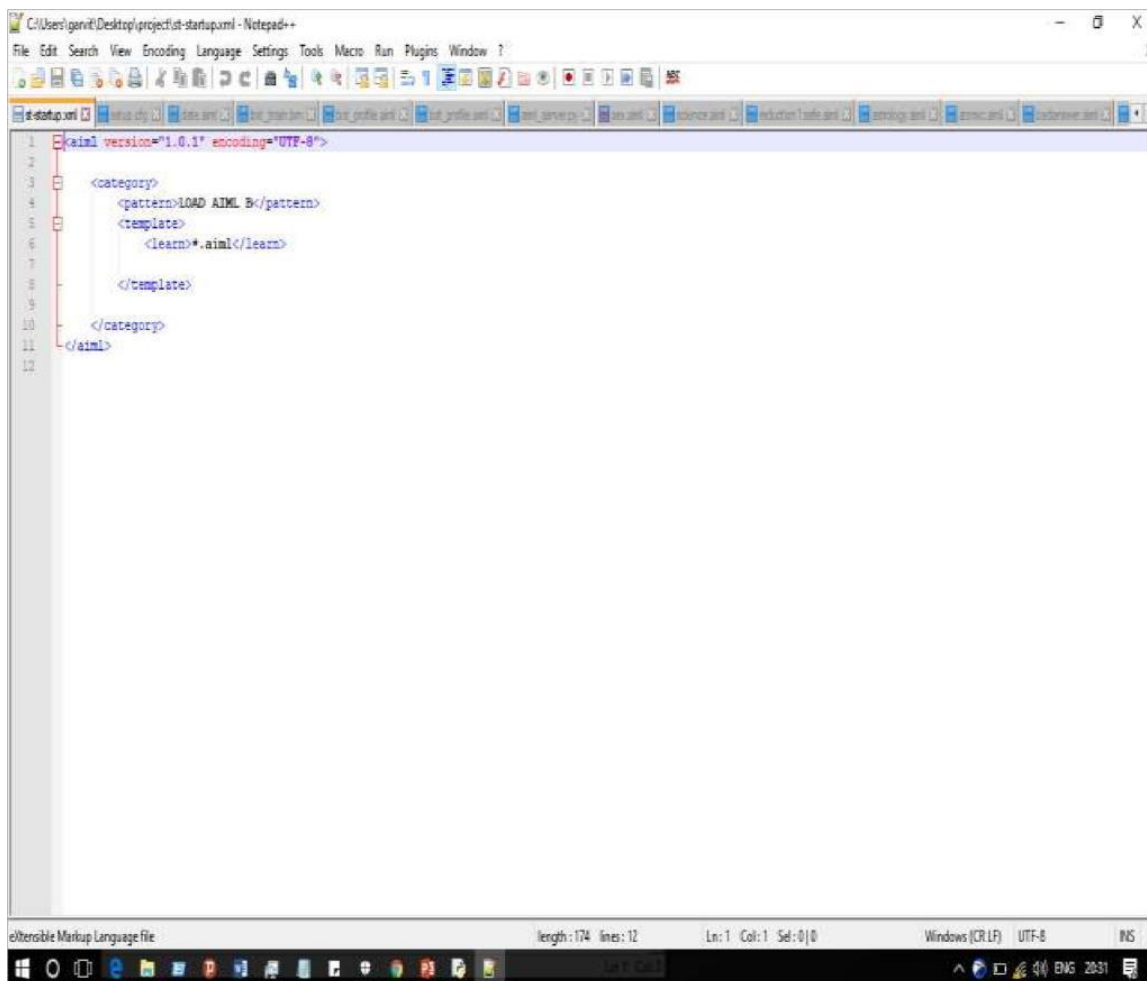


```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <category>
4   <pattern>LOAD AIML B</pattern>
5   <template>
6     <learn>*.aiml</learn>
7   </template>
8 </category>
9
10 </aiml>
```

Fig 5.4 Speeding up Brain

5.5 Loading Brain

This is the simplest program we can start with. It creates the aiml object, learns the startup file, and then loads the rest of the aiml files. After that, it is ready to chat, and we enter an infinite loop that will continue to prompt the user for a message. You will need to enter a pattern the bot recognizes. The patterns recognized depend on what AIML files you loaded. We create the startup file as a separate entity so that we can add more aiml files to the bot later without having to modify any of the programs source code. We can just add more files to learn in the startup xml file.



```
1 <aiml version="1.0.1" encoding="UTF-8">
2
3   <category>
4     <pattern>LOAD AIML B</pattern>
5     <template>
6       <learn>*.aiml</learn>
7     </template>
8   </category>
9
10 </aiml>
```

Fig 5.5 loading Brain

CHAPTER 6

ADVANTEGES & DISADVANTEGES

6.1 Advantages

1. Accessible anytime:

I'm sure most of you are always kept on hold while operators connect you to a customer care executive. On an average people spend around 7 minutes until they are assigned to a person. Gone are the frustrating days of waiting in a queue for the next available operative. They are replacing live chat and other forms of slower contact methods such as emails and phone calls. Since chatbots are basically virtual robots they never get tired and continue to obey your command. They will continue to operate every day throughout the year without requiring to take a break. This improves your customer UX and helps you rank highly in your sector. Another advantage of this instant response is that you can also skillfully craft your chatbot to maintain your image and brand.

2. Handling Capacity:

Unlike humans who can only communicate with one human at a time, chat bots can simultaneously have conversations with thousands of people. No matter what time of the day it is or how many people are contacting you, every single one of them will be answered immediately.

Imagine you own a restaurant, and you have a good reputation for your food of which most of your revenues come from delivery. As the demand keeps rising, you will have more customers to take orders from but very few staff to attend them all. Having a chatbot would eliminate such problem and cater to each and every person and ensure that no order is missed.

3. Flexible attribute:

Chatbots have the benefit that it can quite easily be used in any industry. Unlike other products where you have to do a lot of development and testing to change platforms, chatbots are relatively easy to switch. One has to just train the bot by giving the right conversation structure and flow to switch its current field or industry.

Or if there is a lot of back and forth between two sections of the industry say customer support and sales, then you could have custom built presets which would already have the conversation flow and structure to carry out the interactions with the user.

4. Customer Satisfaction:

Humans are bound to change of emotions. Chatbots, on the other hand, are bound by some rules and obey them as long as they're programmed to. They will always treat a customer in the perfect way no matter how rough the person is or how foul language the person uses.

Not everyone orders the same food everyday, people's choices may change everyday. In this case, it can use your order history to make suggestions for the next order, learn your address details and much more. Customers love this smooth interaction and want all their transactions to be as simple as possible.

5. Cost Effective:

Hiring a human for a job is never a cheap affair, and it will be expensive if your revenue are not high or sales targets are not met and would create havoc in the business. Due to the boundaries of human beings, a single human can only handle one or two people at the same time. More than that would be extremely tough for the employee. Chatbots could help solve this age-old problem. As one chatbot is equal to loads of employees, it can easily communicate with thousands

6. Faster Onboarding:

Before you want to accomplish a task, you first must learn how to work on the task and complete it. Only then will they be considered fit for the job. There is a continuous teaching involved in every level of hierarchy the employee will go through. Also, there will be a lot of change in the employees, some stay, some get fired, some more join in etc.

What we want to say is, employees will change; it's a fact. And this would require you to allot a lot time of your employees into grooming the new joiners. Chatbots could eliminate that time to almost zero, but provide a very clean and easy to understand conversation flow and structure that needs to be maintained by the chatbot. No doubt there will be changes in this too, but it will rather take a fraction of your time to resolve as compared to human employees.

6.2 Disadvantages-

1. Too many functions

Most of developers strive to create a universal chatbot that will become a fully-fledged assistant to user. But in practice functional bots turn out not to cope with the majority of queries. They often do not understand the user, forget what they were told 5 minutes earlier, and have many other disadvantages. And that is no wonder, as the development of a universal bot, which would understand natural language and could evaluate context,

takes years of hard work for a team of experienced programmers. And even in this case, such programs should be constantly improved while in service.

However, modern technologies allow building rather useful bots to perform specific actions such as booking train tickets or providing support to bank customers.

2.Primitive algorithms There are two types of bots: based on artificial intelligence, being able to learn in the process of communication; programmed for specific behavior scenarios.

Artificial intelligence chatbots are considered to be better, as they can respond depending on the situation and context. But the development of complex algorithms is required for this purpose. Meanwhile, only IT giants and few developers possess such powerful technological base.

Therefore, it would be better for ordinary companies to focus on the second variant of bots, as they are more reliable and simpler. Namely for the reason they do not possess intelligence, they will not be able to adopt rude communication patterns and get beyond the control of creators.

3.Complex interface

Talking to a bot implies talking in a chat, meaning that a user will have to write a lot. And in case a bot cannot understand the user's request, he will have to write even more. It takes time to find out which commands a bot can respond to correctly, and which questions are better to avoid. Thus, talking to a chatbot does not save time in the majority of cases.

Perhaps the efficiency of virtual assistants will increase due to the implementation of voice recognition function in the future. But for the time being their functional capabilities are very restricted, and they can be truly useful only in a few business areas.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusions

Chatbots are the new Apps! As we have discussed in the above deliverables, this project brings the power of chatbots to Yioop and enriches its usability. Chatbots in Yioop can give a human like touch to some aspects and make it an enjoying conversation. And they are focused entirely on providing information and completing tasks for the humans they interact with. The above mentioned functionality in all the deliverables is implemented and pushed in to Yioop code. By implementing the above mentioned deliverables I was able to add a basic chatbot functionality in to the Yioop. I.e., configuring and creating accounts for bot users with bot settings which is mentioned in deliverable 2, activating a bot whenever a user asks for it via post in a thread which is discussed in deliverable 3 and as I discussed in deliverable 4, I have implemented a simple weather chatbot that gives weather information whenever a user ask and Fig. 3 tells that I was also able to converse with the bot in Yioop. I intend to enhance the system developed so far in CS298.

7.2Future Scope

There are limitations to what has been currently achieved with chatbots. The limitations of data processing and retrieval are hindering chatbots to reach their full potential. It is not that we lack the computational processing power to do so. However, there is a limitation on “How” we do it. One of the biggest examples is the retail customer market. Retail customers are primarily interested in interacting with humans because of nature of their needs. They don’t want bots to process their needs and respond .

APPENDIX

7.3 SOURCE CODE

```
from tkinter import *
def send():
    send = "You:" + e.get()
    text.insert(END, "\n" + send)
    if(e.get()=='hi'):
        text.insert(END, "\n" + "Bot: hello")
    elif(e.get()=='hello'):
        text.insert(END, "\n" + "Bot: hi")
    elif (e.get() == 'how are you?'):
        text.insert(END, "\n" + "Bot: i'm fine and you?")
    elif (e.get() == "i'm fine too"):
        text.insert(END, "\n" + "Bot: nice to hear that")
    else:
        text.insert(END, "\n" + "Bot: Sorry I didnt get it.")
text = Text(root,bg='light blue')

text.grid(row=0,column=0,columnspan=2)
e = Entry(root,width=80)
send =
Button(root,text='Send',bg='blue',width=20,command=send).grid(row=1,column=1)
e.grid(row=1,column=0)
root = Tk()
root.title('SIMPLE CHATBOT')
```

root.mainloop()

7.4 Screenshots

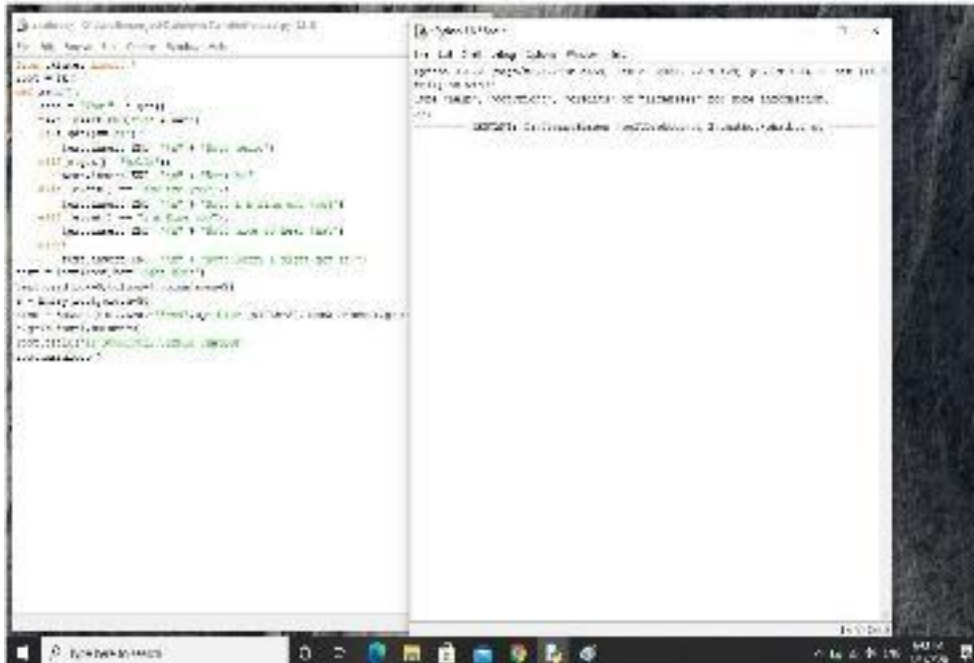


FIG: 7.4.1 PROGRAM

