

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

RAVIKUMAR P

(Reg. No. 37120058)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119. TAMILNADU.

JANUARY 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of RAVIKUMAR P (Reg. No. 37120058) who underwent the professional training in "COVID CHATBOT" under our supervision from JULY 2020 to AUGUST 2020.

Internal Guide

Dr. P.AJITHA 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, RAVIKUMAR P(Reg. No. 37120058) hereby declare that the Professional Training Report on “**COVID CHATBOT**” done by me under the guidance of Dr.P. AJITHA PhD., at Sathyabama institute is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering 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.P.AJITHA 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 covid 19- chatbot. discussion groups, blogs, wikis etc. 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. 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.

Chatbots come in two kinds:

- Limited set of rules
- Machine learning

Chatbot that uses limited set of rules

This kind of bots are very limited to set of texts or commands. They have ability to respond only to those texts or commands. If user asks something different or other than the set of texts or commands which are defined to the bot, it would not respond as desired since it does not understand or it has not trained what user asked. These bots are not very smart when compared to other kind of bots.

TABLE OF CONTENTS

CHAPTER NO :	TITLE	PAGE NO:
	ABSTRACT	5
	LIST OF FIGURES	
1	INTRODUCTION	
	1.1 CHAT BOT	9
	1.2 ARTIFICIAL INTELLIGENCE	9
	1.3 AI APPLICATION	12
2	NLP	
	2.1 NATURAL LANGUAGE PROCESS	14
	2.2 MACHINE LEARNING	14
3	COVID-19	
	3.1 COVID-19 CHATBOT	16
	3.2 FRAMEWORK AD FUNCTIONALITY	18
4	AIML	
	4.1 AI MARKUP LANGUAGE	19
	4.2 ENGINE DESIGN	22
	4.3 QUALITY EVOLUTION	23
	AND COMPARION	

	4.4 DISCUSSION	24
5	TECHNOLOGY	
	5.1 ABOUT PYTHON	26
	5.2 WHY IS PYTHON USED FOR?	27
	5.3 PYTHON LANGUAGE	28
	PROS AND CONS	
	5.4 PYTHON 2 VERSUS PYTHON 3	30
6	CONCLUSION	31
	REFERNCES	32
	APPENDIX	
	SCOURCE CODE	34

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
1	Classification	15
2	Regression	15
3	Symptoms of COVID-19	17
4	COVID-19 Epidemic Disease	18
5	Symptomatic inquiry dataset	21

CHAPTER 1

INTRODUCTION

1.1 CHATBOT

chatbots work 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 illustrates 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: "Great! Looking for flights."

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 enhance 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.

1.2 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. His categories are as follows:

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 decision-making 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.

EXAMPLES OF AI TECHNOLOGY

Automation is the process of making a system or process function automatically. Robotic process automation, for example, can be programmed to perform high-volume, repeatable tasks normally performed by humans. RPA is different from IT automation in that it can adapt to changing circumstances.

Machine learning is the science of getting a computer to act without programming. Deep learning is a subset of machine learning that, in very simple terms, can be thought of as the automation of predictive analytics. There are three types of machine learning algorithms: supervised learning, in which data sets are labeled so that patterns can be detected and used to label new data sets; unsupervised learning, in which data sets

aren't labeled and are sorted according to similarities or differences; and reinforcement learning, in which data sets aren't labeled but, after performing an action or several actions, the AI system is given feedback.

Machine vision is the science of making computers see. Machine vision captures and analyzes visual information using a camera, analog-to-digital conversion and digital signal processing. It is often compared to human eyesight, but machine vision isn't bound by biology and can be programmed to see through walls, for example. It is used in a range of applications from signature identification to medical image analysis. Computer vision, which is focused on machine-based image processing, is often conflated with machine vision.

Natural language processing (NLP) is the processing of human -- and not computer -- language by a computer program. One of the older and best known examples of NLP is spam detection, which looks at the subject line and the text of an email and decides if it's junk. Current approaches to NLP are based on machine learning. NLP tasks include text translation, sentiment analysis and speech recognition.

Pattern recognition is a branch of machine learning that focuses on identifying patterns in data. The term, today, is dated.

Robotics is a field of engineering focused on the design and manufacturing of robots. Robots are often used to perform tasks that are difficult for humans to perform or perform consistently. They are used in assembly lines for car production or by NASA to move large objects in space. More recently, researchers are using machine learning to build robots that can interact in social settings.

1.3 AI APPLICATIONS

AI in healthcare. The biggest bets are on improving patient outcomes and reducing costs. Companies are applying machine learning to make better and faster diagnoses than humans. One of the best known healthcare technologies is IBM Watson. It understands natural language and is capable of responding to questions asked of it. The system mines patient data and other available data sources to form a hypothesis, which it then presents with a confidence scoring schema. Other AI applications include chatbots, a computer program used online to answer questions and assist customers, to help schedule follow-up appointments or aiding patients through the billing process, and

virtual health assistants that provide basic medical feedback. AI in business. Robotic process automation is being applied to highly repetitive tasks normally performed by humans. Machine learning algorithms are being integrated into analytics and CRM platforms to uncover information on how to better serve customers. Chatbots have been incorporated into websites to provide immediate service to customers. Automation of job positions has also become a talking point among academics and IT consultancies such as Gartner and Forrester.

AI in education. AI can automate grading, giving educators more time. AI can assess students and adapt to their needs, helping them work at their own pace. AI tutors can provide additional support to students, ensuring they stay on track. AI could change where and how students learn, perhaps even replacing some teachers. AI in finance. AI applied to personal finance applications, such as Mint or Turbo Tax, is upending financial institutions. Applications such as these could collect personal data and provide financial advice. Other programs, IBM Watson being one, have been applied to the process of buying a home. Today, software performs much of the trading on Wall Street. AI in law. The discovery process, sifting through of documents, in law is often overwhelming for humans. Automating this process is a better use of time and a more efficient process. Startups are also building question-and-answer computer assistants that can sift programmed-to-answer questions by examining the taxonomy and ontology associated with a database. AI in manufacturing. This is an area that has been at the forefront of incorporating robots into the workflow. Industrial robots used to perform single tasks and were separated from human workers, but as the technology advanced that changed.

CHAPTER 2

NLP

2.1 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 replacement 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 commands in everyday languages and is changing the way of interacting.

2.2 MACHINE LEARNING.

Machine Learning (ML) is an area of computer science that "gives computers the ability to learn without being explicitly programmed". The parameters of the formulas are calculated from the data, rather than defined by the programmer. Two most common usages of ML are Classification and Regression. As shown in figure 1[8], Classification means to categorize different types of data, while Regression means to find a way to describe the data. A 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 parameters 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.

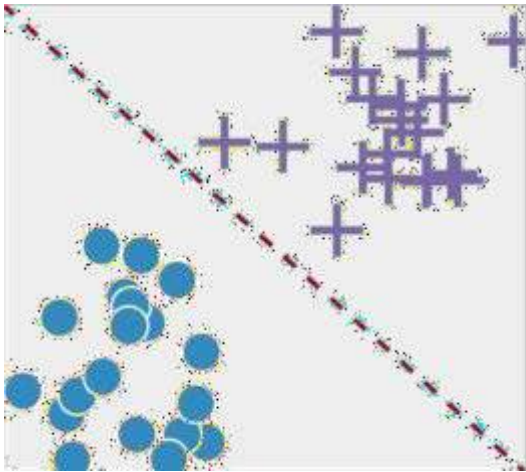


Fig1: classification

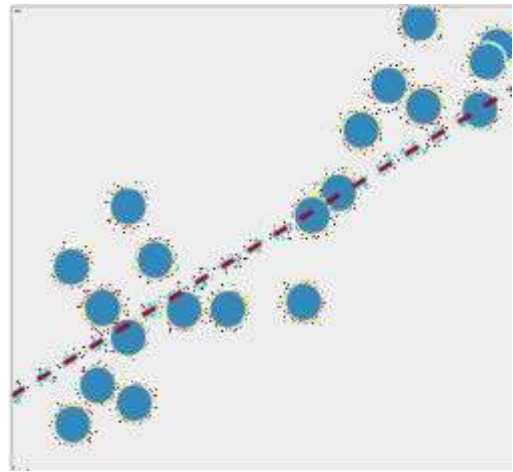


fig 2:regression

CHAPTER 3

COVID-19

3.1 COVID-19 CHATBOT

After the rise of the web and mobile apps, virtual chatbot applications are the latest invention of digital design . These applications are well known for automatic conversational agents that run on computer programming or a kind of artificial intelligence (AI) interaction between the users and machines with the intervention of natural language processing (NLP). Chatbots are potentially referred to as the most promising and advanced form of human-machine interactions]. Eventually, these virtual agents are getting involved in the main global sectors such as healthcare, banking, education, agriculture, etc. The healthcare sector is closely associated with human interaction, and it seems counterintuitive that conversational AI applications like chatbots are more prevalent . Hospital administrators a responding their day in appointment scheduling and answering routine questions of patients. Continuing or repeating the same actions and words is neither necessary nor productive. Such jobs can be easily done with bot applications. It is obvious that patient feedback assessments are also possible by collecting user responses to maintain good patient flow. In the occurrence of serious pandemics like novel Coronavirus (nCOV-19), health bots are beneficial as a supplement to personal clinical care or immediate medications. After nCOV-19 spread beyond China, it spread globally at a rapid pace and about six million cases have been confirmed [6]. Because of continuous patient flow, it has been a great challenge for national governments to supply enough medical specialists, resources, and equipment to hospitals or healthcare centers. Therefore, we designed an AI medical chatbot to ease the burden of healthcare



Fig 3:symptoms of COVID-19

systems by identifying infection severity. In addition, it has all the necessary preventative measures including interactions with live doctors. There are some established nCOV-19 virtual agents integrated with messenger applications. The World Health Organization

(WHO) has launched a dedicated messenger app in seven languages to keep the public safe from coronavirus infections [7]. Likewise, the German government developed a ‘fight COVID messenger bot’ [8], the Bangladesh-based SAJIDA Foundation developed an nCOV-19 information bot with a symptom checker and explanations of preventive measures [9]. While in India, the Aarogya Setu mobile app has been recently developed to create awareness of nCOV-19 with the parallel connection of a chatbot [10]. However, all of these bots are serving as medical consultants of the coronavirus, and none of them highlight the issues concerning remote patients in terms of the pandemic. Therefore, our proposed personal health chatbot for rural patients will act as a medical consultant, and also provides simple and relevant measures of not being infected by nCOV-19. Another advantage of this bot includes 24/7 accessibility and assesses the patient’s condition in a more human-like way. Due to the built-in backend logic function, it will detect the virus’s intensity and provide live interactions with doctors in the handling of dangerous conditions.



Fig 4: covid 19-epidermic disease

3.2 FRAMEWORK AND FUNCTIONALITY

The designed bot can handle user requests and identify message patterns with an artificial intelligence markup language (AIML). AIML is an XML-based markup dialect to create natural language software agents and gives the real human interactive experience to users. Depending on user responses, AIML logic retrieves symptomatic keywords to assess the existing user medical conditions. Ultimately, we aim to make sure the user feels like they are having a conversation with a health specialist. The functionality of chatbot is defined in two ways: request analysis or return response. At first, chatbot evaluates the severity of the virus through feedback from a predefined questionnaire. Simultaneously, if the user fails to acknowledge precise answers, the bot will fail to provide the correct response. In its response return, after the evaluation of a patient's condition, the chatbot provides an obvious response in the form of either generic text or text retrieved from the knowledgebase response. The designed framework of AI chatbot functionality. A render question could help chatbot precisely understand the user's request. To achieve the intended response, the fusion of AI technology with natural language programming (NLP) has been done, because NLP is a helpful technology to draw the relationship between natural language understanding and decides complementary response outcomes. The knowledge base is focused on user message response during the initiation of conversation—it should be natural at responding with a suitable back-end logic. The idea behind this is to identify preliminary symptoms of nCoV-19 from the user location. Thereafter, it will display whether the user is likely to be infected or not. After the user initiating the chat session, the bot engine starts questioning the person regarding symptomatic information. Once it has all of the necessary details, it finds the virus severity percentage that the user experienced and acts accordingly, by either making contact with health specialists or provides information regarding immediate preventive measures depicts the systematic functionality of the proposed chatbot application.

CHAPTER 4

AIML

4.1 ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE

The AIML works according to stimulus-response methods and provides simple dialogue modeling. As mentioned, it is an XML-based markup language with a tag basis. These tags are identifiers that insert commands and make code snippets in the chatbots. AIML defines data object classes as objects responsible for modeling patterns in conversation. The general format of AIML command/tag/objects has a structure as

<Command>

List of parameters

</command>

With the integration of AIML in our chatbot, it can detect patterns from user messages and provides precise and meaningful answers. Our chatbot also figures out if the user wants to check the infection status or just wants to know basic measures and symptomatic behavior of the nCOV-19 virus. Given that a chatbot should understand the patterns of user's requests with predefined tags of the AIML component. In AIML, a predefined tag pattern (<pattern>) helps chatbot in recognition of virus symptoms and if it matches, the particular category of the questionnaire could be displayed. For instance, if the user suspects they have been infected, the following sample patterns In such the above-mentioned patterns, our chatbot looks for possible cases of being infected or if it crosses the threshold value immediately, it will connect to the health specialists. Any chatbot should be natural while responding to user responses and it needs to have a clear dataset and sustainable backend logic for outcome generation. At the Telemedicine department at the University of Camerino, we have developed a ten-question basic symptomatic questionnaire of COVID-19. This entire questionnaire is made of simple Yes/No type questions. If the user answered 'Yes', then the bot assigns the score of one, and if the user answered 'No', the bot gives a score of zero. The chatbot expects to use AIML logic to reply to user responses and sustain the input that could feed the machine.

Table 1 presents the initial symptomatic inquiries that the chatbot can identify whether the ut.

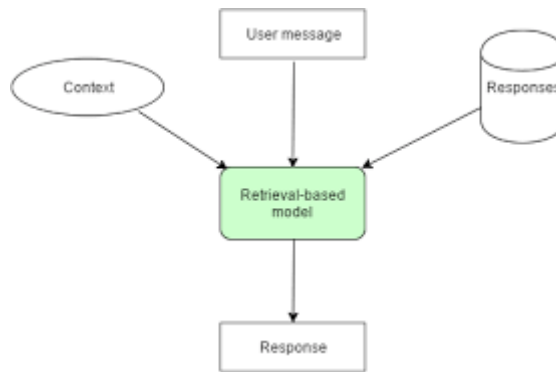


Fig 5: Symptomatic inquiry dataset.

No. Question Type

1. Do you have any kind of diabetes? Yes/No
2. Have you or your family been infected with nCOV-19? Yes/No
3. Do you know people in your immediate social environment who are or have been infected with the novel coronavirus? Yes/No
4. Are you closely interacting with other infected people in your region? Yes/No
5. Is your travel history associated with infected countries in the last two months? Yes/No
6. Have you used a bed or other premises previously used by someone who was infected by coronavirus? Yes/No
7. Have you been in contact with a suspected case-patient of COVID-19? Yes/No
8. Any shortness of breath? Yes/No
9. Are you facing a sudden rise in body temperature? Yes/No
10. Are you coughing often nowadays? Yes/No

Once the engine retrieves the user feedback for the given questionnaire, it will assess the symptomatic response score-defined AIML technique. We set a threshold value of AIML logic at 'seven', and if bot accumulated response value crosses this threshold limit, then the user will be connected directly to the doctor. Simultaneously, the bot will also anticipate immediate preventative measures to avoid direct contact with others. When the bot engine feels the

severity of the user's symptoms has reached the threshold, it automatically alerts the users to contact health specialists immediately. In this scenario, chatbot requests a doctor to get in touch with the user with useful clinical information. To trigger this procedure, the pandemic severity score has to touch or cross the threshold level [14].

total score of symptomatic information from user input, and the threshold value maximum limit of chatbot deciding if the user can handle the level of the condition. If $H \geq 1$, then chatbot triggers the infection severity and connects to the doctor immediately. Initially, we created a project folder with two directories called aiml_scripts for AIML scripts and configs, cd into the project given folder. We created an AIML script called user_pandemic_chat.aiml and placed it into the aiml_scripts folder with the following content:

```
<aiml version="1.01" encoding="UTF-8">
```

```
<!--user_pandemic_chat.aiml -->
```

```
<category>
```

```
<pattern>hello</pattern>
```

```
<template>
```

```
<random>
```

```
<li>hi (user name), I am a virtual health assistant to help you to identify whether you infected with NCOV-19 or not?You would like to know please press 'yes' or for preventive information press 'no'</li>
```

```
<li>yes</li>
```

```
<li>thank you for the response, please respond all the questions that displayed here</li>
```

```
</random>
```

```
</template>
```

```
</category>
```

```
<category>
```

```
<Pattern>What is your gender? (Male/Female) </pattern>
```

<template>

 male

 great, what's your age (20-35), (36-55), (55-75), >75

 36-55

are you face any of following symptoms (cough/ Fever/Breath difficulties)

None

Nice, have you ever had any of the following (heart or lung diseases/hypertension/None of the above)

None of the above

it seems that your non infective member on board, however please follow the preventive measures that suggested and

don't hesitate contact us any time

preventive measures document

good day

</random>

</template>

</category>

</aiml>

4.2 ENGINE DESIGN

The conversational agent engine allows the chatbot to identify patient preferences, problem areas, and patient conditions. It identifies the behavioral patterns and autonomously learns from the conversation. The chatbot engine is a principal back-end logic that validates input using the method from Web API. After successful identification of infection severity, the engine maintains two different methods, such as contact with the doctor and immediate follow-up of household preventive measures. Each symptom defines with an integer variable

to assess the seriousness score. Once all the symptomatic information is collected, the bot triggers a specialist consultation. The engine prepares a solution in the XML format for this action. AIML uses <template> and <sraix> tags to parse the XML feedback and user display.

4.3 QUALITY EVOLUTION AND COMPARISON

We considered ten different functional aspects of existing individual bots of nCOV-19 that were included in this study to evaluate each performance (Table 2). Primarily, each bot examines the visual look and implementation form on the website and the speech synthesis units. Simultaneously, to evaluate the knowledge base of reliably established bots, a similar line of questioning was asked for each virtual assistant. This enables us to understand language skills, conversation abilities, and text context of chatbots. Table 2. Performance evolution and comparison of NCOV-19 bots (1: poor and 5: very satisfied). Functionality WHO nCOV-19 Launched Bot Fight COVID Messenger Bot Germany

SAJIDA Corona

Bot Application in

Bangladesh

Aarogya Setu

in India Our Bot

Visual Look 3 3 3 4.5 4.5

Form of Implementation on the

Website 3.5 3.5 4 3.5 5

Knowledge Base 4.5 4 4 4.5 5

Speech Synthesis Unit 3 3.5 3 3 4.5

Knowledge presentation 5 4.5 4.5 4.5 5

Conversational Abilities,

Language Skills

and Context Sensitivity

Personalization 4 3.5 3 4 5

Personality Traits 4.5 4.5 4 4.5 4.5

Emergency Responses in

Unexpected 4.5 4 3 4 5

Situations

Possibility of Rating Chatbot 3.5 3 3 4 5

Overall Average 4 3.7 3.55 4.05 4.8

In addition, we explore other features like personality traits, personalization, and emergency responses with possible user ratings. To assess the quality components of each bot, we used a standard measurement tool rating of 1 to 5 with denotations like 1: very poor, 2: poor, 3: average, 4: satisfied, 5: excellent. Transparent and feasible measures are used, as there is no need to include advanced statistical analysis [15]. Scores of each functionality were given based on user feedback and online patient ratings. In the end, we consider the average score of ten functionalities to gain quality insight into the mentioned nCOV-19 chatbots. From Table 1, it is clear that present bots have some functional limitations. We considered all of these problems while developing our new conversational agent. The WHO launched bot (4.0), and the Indian bot application Arogya Setu (4.05) had satisfactory ratings, followed by the other two bots, such as Fight nCOV-19 from Germany (3.7), and the SAJIDA corona bot from Bangladesh ().

4.4DISCUSSION

Artificial intelligence consists of a heterogeneous set of techniques and methods for building intelligent systems with cognitive capabilities of recognizing, classifying, reasoning diagnosing, or having at least some of these properties [16,17]. A list (incomplete) of the main problems covered by this discipline includes automatic reasoning, planning, diagnosis and automatic classification, understanding of natural language, learning, and robotics. NLP is an artificial intelligence tool that is well-suited for clinical diagnostic issues and the development of symptomatic description pathways when in need of private medical care. It will be started

by identifying the most common patient pathologies who live in remote areas like mountains, forests, and slums [18]. Because of the nonavailability of nCOV-19 clinical information from medical

experts, our designed conversational agent with an elaborating questionnaire could enable us to address general questions. The questionnaire threshold outcome is a simple statistical test that represents the origin of developing the AI system symptomatology section. In this study, we randomly selected four international chatbots for nCOV-19, including three messenger bots and one Indian application with the built-in chat feature. A comparison of each application was done with ten relevant functional aspects mentioned in Table 1. Each bot had some limitations and two of them only reached satisfactory outcome ratings. To overcome this, we designed a sophisticated chatbot application that surpassed the mentioned bots. During implementation, the appearance of a chatbot would play a vital role in measuring quality standards. The bot's visual look resembles the targeted users and enhances their involvement or desire

graphics of a low quality. However, we maintain a good visual look to attract more users, and the knowledge base stores patient symptomatic information. Speech synthesis units have a special feature of converting written text into a synthetic speech. We felt that no existing chatbot possessed this function with a unique custom voice coupled with a shutdown option. Conversational ability, personalization, and personality traits will be vital to identify symptomatic behavior of the nCOV-19 pandemic. Single embedded links can present preventive measures, and these links work as phrased tag words to the user's input for ongoing conversation. In an emergency, the chatbot provides patient location, symptomatic data, and the infection severity score to a doctor or healthcare organization's automatic voice message alert.

user. Similarly, it offers a chance to speak with a doctor 24/7 with a live chat feature; the user can feel confident in sharing their personal information of mental health or insight symptomatic behavior of the nCOV-19 virus.

CHAPTER 5

TECHNOLOGY

5.1 ABOUT PYTHON:

Dating from 1991, Python is a relatively new programming language. From the start, Python was considered a gap-filler, a way to write scripts that “automate the boring stuff” (as one popular book on learning Python put it) or to rapidly prototype applications that will be implemented in one or more other languages.

However, over the past few years, Python has emerged as a first-class citizen in modern software development, infrastructure management, and data analysis. It is no longer a back-room utility language, but a major force in web application development and systems management and a key driver behind the explosion in big data analytics and machine

Perfect for IT, Python simplifies many kinds of work, from system automation to working in cutting-edge fields like machine learning.

Python is easy to learn. The number of features in the language itself is modest, requiring relatively little investment of time or effort to produce one’s first programs. Python syntax is designed to be readable and straightforward. This simplicity makes Python an ideal teaching language, and allows newcomers to pick it up quickly. Developers spend more time thinking about the problem they’re trying to solve, and less time thinking about language complexities or deciphering code left by others.

Python is broadly used and supported. Python is both popular and widely used, as the high rankings in surveys like the Tiobe Index and the large number of GitHub projects using Python attest. Python runs on every major operating system and platform, and most minor ones too. Many major libraries and API-powered services have Python bindings or wrappers, allowing Python to interface freely with those services or make direct use of those libraries. Python may not be the fastest language, but what it lacks in speed, it makes up for in versatility. Python is not a “toy” language. Even though scripting and automation cover a large chunk of Python’s use cases (more on that below), Python is also used to build robust, professional-quality software, both as standalone applications and as web services.

5.2 WHAT IS PYTHON USED FOR?

The most basic use case for Python is as a scripting and automation language. Python isn't just a replacement for shell scripts or batch files, but is also used to automate interactions with web browsers or application GUIs or system provisioning and configuration in tools such as Ansible and Salt. But scripting and automation represent only the tip of the iceberg with Python.

- Python is used for general application programming. Both CLI and cross-platform GUI applications can be created with Python and deployed as self-contained executables. Python doesn't have the native ability to generate a standalone binary from a script, but third-party packages like `cx_Freeze` or `PyInstaller` can be used to accomplish that.
- Python is used for data science and machine learning. Sophisticated data analysis has become one of fastest moving areas of IT and one of Python's star use cases. The vast majority of the libraries used for data science or machine learning have Python interfaces, making the language the most popular high-level command interface to for machine learning libraries and other numerical algorithms.
- Python is used for web services and RESTful APIs. Python's native libraries and third-party web frameworks provide fast and convenient ways to create everything from simple REST APIs in a few lines of code, to full-blown, data-driven sites. Python's latest versions have powerful support for asynchronous operations, allowing sites to handle up to tens of thousands of requests per second with the right libraries.
- Python is used for metaprogramming. In Python, everything in the language is an object, including Python modules and libraries themselves. This allows Python to work as a highly efficient code generator, making it possible to write applications that manipulate their own functions and have the kind of extensibility that would be difficult or impossible to pull off in other languages.

- Python is used for glue code. Python is often described as a “glue language,” meaning it can allow disparate code (typically libraries with C language interfaces) to interoperate. Its use in data science and machine learning is in this vein, but that’s just one incarnation of the general idea.

Also worth noting are the sorts of tasks Python is not well-suited for. Python is a high-level language, so it’s not suitable for system-level programming—device drivers or OS kernels are straight out. It’s also not ideal for situations that call for cross-platform standalone binaries. You could build a standalone Python app for Windows, Mac, and Linux, but not elegantly or simply. Finally, Python is not the best choice when speed is an absolute priority in every aspect of the application. For that you’re better off with C/C++ or another language of that caliber.

5.3 THE PYTHON LANGUAGE’S PROS AND CONS

Python syntax is meant to be readable and clean, with little pretense. A standard “hello world” in Python 3.x is nothing more than:

- `print("Hello world!")`
- Python provides many syntactical elements that make it possible to concisely express many common program flows. Consider a sample program for reading lines from a text file into a list object, stripping each line of its terminating newline character along the way:
- `with open('myfile.txt') as my_file:`
- `file_lines = [x.strip('\n') for x in my_file]`
- The `with/as` construction is a “context manager,” which provides an efficient way to instantiate a given object for a block of code and then dispose of it outside of that block. In this case, the object in question is `my_file`, instantiated with the `open()` function. This takes the place of several lines of boilerplate to open the file, read individual lines from it, then close it up.

- The `[x ... for x in my_file]` construction is another Python idiosyncrasy, the “list comprehension.” It allows a given item that contains other items (here, `my_file` and the lines it contains) to be iterated through, and to allow each iterated element (that is, each `x`) to be processed and automatically appended into a list.
- You could write such a thing as a formal `for... loop` in Python, much as you would in another language. The point is that Python has a way to economically express things like loops that iterate over multiple objects and perform some simple operation on each element in the loop, or work with things that require explicit instantiation and disposal. Constructions like this allow Python developers to balance terseness and readability.
- Python’s other language features are meant to complement common use cases. Most modern object types—Unicode strings, for instance—are built directly into the language. Data structures—like lists, dictionaries (i.e., hashmaps), tuples (for storing immutable collections of objects), and sets (for storing collections of unique objects)—are available as standard-issue items.
- Like C#, Java, and Go, Python has garbage-collected memory management, meaning the programmer doesn’t have to implement code to track and release objects. Normally garbage collection happens automatically in the background, but if that poses a performance problem, it can be triggered manually or disabled entirely.
- An important aspect of Python is its dynamism. Everything in the language, including functions and modules themselves, are handled as objects. This comes at the expense of speed (more on that below), but makes it far easier to write high-level code. Developers can perform complex object manipulations with only a few instructions, and even treat parts of an application as abstractions that can be altered if needed.
- Python’s use of significant whitespace has been cited as both one of Python’s best and worst attributes. The indentation on the second line shown above

isn't just for readability; it is part of Python's syntax. Python interpreters will reject programs that don't use proper indentation to indicate control flow.

- Syntactical white space might cause noses to wrinkle, and some people do reject Python out of hand for this reason. But strict indentation rules are far less obtrusive in practice than they might seem in theory, even with the most minimal of code editors, and the end result is code that is cleaner and more readable.

5.4 PYTHON 2 VERSUS PYTHON 3

- Python is available in two versions, which are different enough to trip up many new users. Python 2.x, the older "legacy" branch, will continue to be supported (i.e. receive official updates) through 2020, and it might even persist unofficially after that. Python 3.x, the current and future incarnation of the language, has many useful and important features not found in 2.x, such as better concurrency controls and a more efficient interpreter.

Python 3 adoption was slowed for the longest time by the relative lack of third-party library support. Many Python libraries supported only Python 2, making it difficult to switch. But over the last couple of years, the number of libraries supporting only Python 2 has dwindled; most are now compatible with both versions. Today, there are few reasons against using Python 3.

CHAPTER 6

CONCLUSION

Our idea behind this study is to present sophisticated AI medical chatbots for users, especially during unknown pandemics like nCOV-19. The presented AI chatbot will have a large impact on patient life during serious epidemics. It would provide the advantage of putting access to virtual doctors into their hands. We bring health specialists and professionals into our platform to feed medical information into a bot engine, also to the availability of every user whenever the possibility of infection is detected.\ At present, the proposed chatbot is in the design phase, which will be followed up by total design into code soon, with plans to launch this app in the next few months. Initially, we would like to release the basic version soon, especially in the context of the severity of the present pandemic of the novel coronavirus. We are developing this chatbot engine in Python, and Watson as the AIML platform. After releasing this bot into the market, based on user feedback, further updates will be possible for at least 3–4 months. Once this COVID-19 pandemic is over, we plan to reuse this conversational agent and make it compatible with other epidemics or other services with individual APIs. .

REFERENCES

1. Jadhav, K.P.; Thorat, S.A. Towards Designing Conversational Agent Systems. In Advances in Intelligent Systems and Computing; Springer: Berlin, Germany, 2020.
2. Battineni, G.; di Canio, M.; Chintalapudi, N.; Amenta, F.; Nittari, G. Development of physical training smartphone application to maintain fitness levels in seafarers. *Int. Marit. Health* 2019, 70, 180–186. [CrossRef] [PubMed]
3. Yan, R. “Chitty-chitty-chat bot”: Deep learning for conversational AI. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), Stockholm, Sweden, 13–19 July 2018.
4. Luo, X.; Tong, S.; Fang, Z.; Qu, Z. Frontiers: Machines vs. humans: The impact of artificial intelligence chatbot disclosure on customer purchases. *Mark. Sci.* 2019. [CrossRef]
5. Chung, K.; Park, R.C. Chatbot-based healthcare service with a knowledge base for cloud computing. *Cluster Comput.* 2019, 22, 1925–1937. [CrossRef]
6. Sohrabi, C.; Alsafi, Z.; O’Neill, N.; Khan, M.; Kerwan, A.; Al-Jabir, A.; Iosifidis, C.; Agha, R. World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19). *Int. J. Surg.* 2020, 76, 71–76. [CrossRef] [PubMed]
7. WHO Health Alert Brings COVID-19 Facts to Billions Via WhatsApp. Available online: <https://www.who.int/news-room/feature-stories/detail/who-health-alert-brings-covid-19-facts-to-billions-via-whatsapp> (accessed on 13 April 2020).

8. How Governments Worldwide are Using Messaging Apps in Times of COVID-19'. Available online:

<https://www.messengerpeople.com/governments-worldwide-covid-19/#Germany>

(accessed on 6 May 2020).

9. SAJIDA Foundation and Renata Ltd. Team up to Tackle the COVID-19 Pandemic| Dhaka Tribune'. Available

online: [https://www.dhakatribune.com/feature/2020/04/06/sajida-foundation-and-renata-ltd-team-up-to-](https://www.dhakatribune.com/feature/2020/04/06/sajida-foundation-and-renata-ltd-team-up-to-tackle-the-covid-19-pandemic)

[tackle-the-covid-19-pandemic](https://www.dhakatribune.com/feature/2020/04/06/sajida-foundation-and-renata-ltd-team-up-to-tackle-the-covid-19-pandemic) (accessed on 6 May 2020).

10. Aarogya Setu Mobile App| MyGov.in. Available online:

<https://www.mygov.in/aarogya-setu-app> (accessed

on 6 May 2020).

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access

APPENDIX

(A)SOURCE CODE:

```
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer
import os

def setup():
    chatbot = ChatBot('Bot',
        storage_adapter='chatterbot.storage.SQLStorageAdapter',
        trainer='chatterbot.trainers.ListTrainer')
    for file in os.listdir(r'C:/Users/SALEEM/data/'):
        convData = open(r'C:/Users/SALEEM/data/' + file,encoding='latin-1').readlines()
        chatbot.set_trainer(ListTrainer)
        chatbot.train(convData)
        #print("Training completed")

setup()

from SimpleWebSocketServer import SimpleWebSocketServer, WebSocket
from chatbot import get_response

class ChatServer(WebSocket):

    def handleMessage(self):
        # echo message back to client
        message = self.data
        response = get_response(message)
        self.sendMessage(response)

    def handleConnected(self):
        print(self.address, 'connected')

    def handleClose(self):
        print(self.address, 'closed')

server = SimpleWebSocketServer("", 8000, ChatServer)
server.serveforever()
```

(B)SCREENSHOTS:

