

COVID-19 SCREENING AND FAQs CHATBOT

Thesis submitted in fulfillment of the requirements for the Degree of

Bachelor of Technology

By

DEEPESH PANDEY (Enrolment No. 17102183)



Department of Electronics and Communication Engineering

**JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY
(Declared Deemed to be University U/S 3 of UGC Act)
A-10, SECTOR-62, NOIDA, INDIA**

June 2020

TABLE OF CONTENTS

DECLARATION BY THE SCHOLAR.....	iv
SUPERVISOR'S CERTIFICATE.....	v
ACKNOWLEDGEMENTS.....	vi
ABSTRACT.....	vii
LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix-x
 CHAPTER 1	 1-7
1.1 MOTIVATION.....	1
1.2 INTRODUCTION.....	1-2
1.3 LITERATURE SURVEY.....	2-3
1.4 PROJECT IDEA.....	3
1.5 METHODOLOGY.....	4-7
 CHAPTER 2	 8-14
NATURAL LANGUAGE UNDERSTANDING (NLP).....	8-10
2.1.1 NATURAL LANGUAGE UNDERSTANDING (NLU).....	8
2.1.2 NATURAL LANGUAGE GENERATION (NLG).....	9-10
2.1.3 DIALOGUE PROCESSING.....	10
2.2 TYPES OF CHATBOT MODELS.....	11
2.3 CHATBOT PROCESSING MECHANISM.....	12
2.4 CHATBOTS.....	12
2.5 COMPARISON CHART.....	13-14

2.6 HEALTHCARE CHATBOTS.....	14
 CHAPTER 3	 15-17
3.1 DIALOGFLOW.....	15
3.2 FLASK.....	16
3.2.1 WHAT IS FLASK ?.....	16
3.2.2 WHAT IS HTTP AND WHAT IT HAS TO DO WITH FLASK ?.....	16
3.2.3 HOW IS FLASK INVOLVED ?.....	16
3.3 NGROK.....	17
 CHAPTER 4	 18-27
4.1 IMPEMENTATION.....	18-27
4.1.1 USING DIALOGFLOW TO BUILD THE CHATBOT.....	18-23
4.1.1.1 AGENTS.....	19
4.1.1.2 INTENTS.....	19-22
4.1.1.3 ENTITIES.....	22
4.1.1.4 CONTEXTS.....	23
4.1.2 BUILDING APP USING FLASK.....	23-24
4.1.3 USING NGROK TO EXPOSE THE LOCALHOST.....	24-27
 CHAPTER 5	 28-31
5.1 LEARNING OUTCOME.....	28
5.2 IMPORTANCE OF CHATBOTS IN INDUSTRY.....	29-31
 REFERENCES.....	 32

DECLARATION BY THE SCHOLARS

I hereby declare that the work reported in the In-House Internship thesis entitled “**COVID-19 SCREENING AND FAQs CHATBOT**” submitted at **Jaypee Institute of Information Technology, Noida, India**, is an authentic record of our work carried out under the supervision of **Dr. Gaurav Verma**. I have not submitted this work elsewhere for any other degree or diploma. I am fully responsible for the contents of my In-House Internship.

(Signature of the Scholar)

DEEPESH PANDEY

DATE:.....

SUPERVISOR’S CERTIFICATE

This is to certify that the work reported in the final report for In-House Internship entitled “**COVID-19 SCREENING AND FAQs CHATBOT**”, submitted by **DEEPESH PANDEY** at **Jaypee Institute of Information Technology, Noida, India**, is a bona fide record of their original work carried out under my supervision. This work has not been submitted elsewhere for any other degree or diploma.

(Signature of the Supervisor)

(Dr. Gaurav Verma)

Assistant Professor (Senior Grade), ECE Department,
JIIT, Noida (U.P.) INDIA

Date:.....

ACKNOWLEDGEMENT

Without the guidance and motivation of my mentor, this project would not have achieved as much as it did. These groups of people include my advisors, mentor and other students. This project would not exist without the support of my mentor, Assistant Professor (Senior Grade), Dr. Gaurav Verma.

Throughout the project they kept me motivated, and whenever we had a problem, they always had plenty of suggestions about how to solve it. Other faculty of JIIT helped us with specific challenges.

ABSTRACT

Artificial Intelligence (AI) is a field of study in intelligent agents and has been in existence since the Turing machine and much beyond. There have been great advancements in AI over the past few decades. One of the major areas of AI is concerned with the interaction between computers and human (natural) languages. Most natural language processing (NLP) systems were based on complex sets of hand-written rules up to the 1980s. Starting in the late 1980s, however, there was a revolution in NLP with introduction of machine learning (ML) algorithms for language processing. A **chatbot** (talkbot, chatterbot, bot or interactive agent) is a computer program which conducts a conversation via auditory or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test. Most chatbots are either accessed via virtual assistants such as Google Assistant, Amazon Alexa, Apple Siri or via individual organizations' apps and websites. There are various platforms available to create a chatbot such as msg.ai, wit.ai, Dialogflow, Microsoft Language Understanding Intelligent Service (LUIS) etc. **Dialogflow** was chosen for this project because, the platform is very intuitive, the ease of creation of the chatbot was better when compared to other platforms, provides support for large set of languages and it provides integrations to various third-party apps. But most of the platforms follow the same concepts and terminologies.

Chatbot creation can be divided into design, building, analytics and maintenance. The chatbot *design* is the process that defines the interaction between the user and the chatbot. We as the chatbot designers defined the chatbot personality, the questions that will be asked to the patients (users), and the overall interaction. An important part of the chatbot design is also centered around user testing. Dialogflow provides out-of-the box techniques for testing and training. The process of *building* a chatbot involves two tasks: understanding the user's intent and producing the correct answer. In our case, the first task was to understand the user input and we have narrowed our inputs to a small set of questionnaires to demonstrate the NLP techniques. The responses are generated depending on the first task. Dialogflow provides very good *analytics* to monitor the usage of the chatbot and helps us improve accuracy and the overall user experience. We leveraged most of the features that Dialogflow provides and we even developed our own website and embedded our chatbot in it.

I thoroughly enjoyed building our own chatbot from scratch and learning the advancements in the chatbots, NLP, machine learning and in general AI.

LIST OF TABLES

TABLE 2.1 TABLE SHOWING TYPES OF CHATBOTS	11
TABLE 2.2 COMPARISION CHART	13

LIST OF FIGURES

FIGURE 1.1 STRUCTURE OF THE FLASK APP	4
FIGURE 1.2 NGROK PROMPT DISPLAYING THE WEB URL	5
FIGURE 1.3 NGROK URL IN WORKING STATE	6
FIGURE 1.4 CHATBOT INTEGRATED WITH FACEBOOK MESSENGER	7
FIGURE 2.1 A FLOWCHART OF THE PROCESSING MECHANISMS FOR A CHATBOT	12
FIGURE 2.2 WIT.AI CONSOLE	14
FIGURE 2.3 LUIS.AI CONSOLE	14
FIGURE 3.1 DIALOGFLOW LOGO	15
FIGURE 3.2 FLASK LOGO	16
FIGURE 3.3 NGROK LOGO	17
FIGURE 4.1 CHATBOT IS ABLE TO IDENTIFY THE INTENT DESPITE THE TRAINING PHRASE DOESN'T MATCHES	19
FIGURE 4.2 TRAINING PHRASES	19
FIGURE 4.3 INTENTS	20
FIGURE 4.4 SEVERAL TRAINING PHRASES	21
FIGURE 4.5 ACTIONS AND PARAMETERS	21

FIGURE 4.6 RESPONSES FOR OUTDOOR ACTIVITIES INTENT	22
FIGURE 4.7 ENTITES FOR MY CHATBOT	22
FIGURE 4.8 NGROK PROMPT SHOWING THE WEBSITE URL	25
FIGURE 4.9 FINAL WEBSITE FOR MY CHATBOT	25
FIGURE 4.10 REPLYING SHOWING LOW RISK	26
FIGURE 5.1 CHATBOT APPLICATIONS	29

CHAPTER 1

1.1 MOTIVATION

COVID-19 is a pandemic that threatens global health. Now, the impacts are not even only in the health sector; economic, politic, national defense and security, etc are also threatened. And my country, Indonesia, is very slow in dealing with the spread of this virus. A lot of people don't know whether they are infected by this virus or not.

So, I wanted to give all the help in my capacity I could. So, I decided to build a Chatbot that could answer all the major FAQs regarding the Novel Coronavirus so that people become more aware of its dangers and could better understand how they can prevent themselves from getting infected.

1.2 INTRODUCTION

Human-Computer Interaction (HCI) researches the design and use of computer technology, focused on the interfaces between people (users) and computers. The notion of an open-ended dialog between the user and computer likens human-computer interaction to human-to-human interaction. This is exactly what we achieved in our chatbot by taking mental illness as a use case and building a conversation with the user (patient). An important facet of HCI is the securing of user satisfaction. **Human-computer interface** can be described as the point of communication between the human user and the computer. The flow of information between the human and computer has several aspects: visual based, audio based, machine environment, input flow, output flow and feedback.

A **dialog system** or **conversational agent (CA)** is a computer system intended to converse with a human, with a coherent structure. **Chatbots** are typically used in dialog systems for various practical purposes and use sophisticated NLP systems. The criterion of intelligence depends on the ability of a computer program to impersonate a human in a real-time written conversation with a human judge, sufficiently well that the judge is unable to distinguish reliably between the program and a real human.

Dialogflow (formerly api.ai) is Google-owned developer of human-computer interaction technologies based on natural language conversations. Dialogflow helps us build natural and rich conversational experiences that gives users new ways to interact with the product by building engaging voice and text-based conversational interfaces powered by AI.

Dialogflow incorporates Google's machine learning expertise and products such as Google Cloud Speech-to-Text and is backed by Google Cloud Platform which can easily scale millions of users. Using years of domain knowledge and NLU, Dialogflow analyzes and understands the user's intent and responds in the most useful way.

1.3 LITERATURE SURVEY

Smart chatbots made up of NLU, NLG and ML engines have the following components: Fig. 1. Architecture of Chatbot System

3.1 A review on Chat Interface This unit is the front end of the system. It is responsible for collecting the user queries from the user which are the input to the system. It is also responsible for displaying the system generated results to the user. Therefore, it can be said that the chat interface is the face of the system through which the entire communication takes place. It is the mediator of conversation between the system and the user. The query that user fires on the chat interface is passed on to the chatting backend which acts as a message delivering system between the Chat interface and the Machine Learning Layer. This interface can be accessed either as a website or as a smart phone app. The type of interface depends on the requirements of the user that are to be satisfied by the system. If the system is accessed from a smartphone, the interface will be in the form of an app and if the system is accessed from a website, then the interface will be in the form of a website. For building apps on the smartphone, it will require to use android for android phones or Swift for iOS. In this case, only the interfacing platform will be programmed on android and the complete backend processing of the system will take place on a server on which the system will be deployed. For making a website, either Java or Python web frameworks can be used. Java provides Spring and Struts as the most advanced and latest web frameworks. Similarly, Python allows usage of Django and Flask frameworks for building of a website. The criteria for selection of the programming language depends upon the functionalities that the system intends to provide, the requirements of the users that will use the system, the algorithms that are to be used by the system, etc. Selection of an appropriate programming language makes it simpler for developers to develop a system which provides maximum functionality to the user with high accuracy and minimum complexity

A review on NLU Engine

NLU i.e. Natural Language Understanding is a subpart of NLP (Natural Language Processing) which enables the system to understand the natural language or the conversational language spoken by the users. The conversational language used by humans for day to day conversations is not as perfect as the formal language. It does not focus much on the vocabulary and the grammar. Hence, it becomes difficult for a system to understand the intent of the sentence.

The input received from the user is in unstructured text format which cannot be understood by the system directly. It understands input only in structured formats. The unstructured text received from the user is converted to structured format by extracting important words and patterns from the user text using the NLU techniques. Humans are capable of understanding any mispronunciations, homophones, swapped words, shortened form of words (like „it“s“ for „it is“), slang words or phrases and also words which are not used in formal vocabulary but exist in regular conversations. NLU techniques enables the system to identify these twerks if the user makes use of them while conversing with the chatbot, so as to make the user feel that the conversation is taking place between two humans and not between a human and a bot. NLU systems do not directly understand the meaning of the user sentences. It involves a sequence of processes to derive the actual intent of the sentence. To understand a complete sentence, the NLU system needs to understand each word of that sentence. It means that the initial task is the segmentation of the sentences into individual words. Next, to understand the word, the system needs to understand the grammar of the sentence. This can be done by knowing the parts of speech of each word in that sentence. Here comes the POS (Parts-Of-Speech) tagger into picture. After knowing the grammatical weightage of each word, all of them are parsed to know the dependency among them. This is the most important step wherein the word with the highest dependency is extracted, from which the intent of the system can be known. It is not possible that the knowledge base would contain the exact sentence that the user has sent. It might contain a sentence with the same intent but with different words used in it. To match these types of synonymic sentences, synonym determination and sentence matching are required. The different tasks to be implemented under the NLU Engine and the methods to do the same have been discussed further.

1.4 PROJECT IDEA

Idea for this project basically arises from the currently prevailing pandemic situation all around the world. I wanted to build a chatbot for the solving the common FAQs of the public regarding the Novel Coronavirus and to allow them to have a platform for their screening through a series of questions so that people can know how severe is their risk to the Novel Coronavirus. Also, I wanted to make use of the powerful technologies like Natural Learning Processing and Artificial Intelligence in this course of the project, thus going forward with this idea of building a chatbot seemed rather ideal as it serves a cause and also makes use of the state-of-the-art technologies I wanted to try my hands on for a long time. This led to the birth of my project “COVID-19 SCREENING AND FAQs CHATBOT”.

1.5 METHODOLOGY

For the completion of this project a well established blueprint was formulated by me and I tried to complete it accordingly. The crux and the main focus of the project is building a chatbot using Dialogflow which is a powerful tool developed by Google for developers to make their agents and also integrate them inside their website or other third-party websites like Facebook Messenger, Slack, Telegram, etc.

Using this tool I developed my chatbot over a period of time, the entire process has been described in chapter 4.

Then I build a website using Flask on my local machine. This website makes use of several python libraries and its structure looks like this.

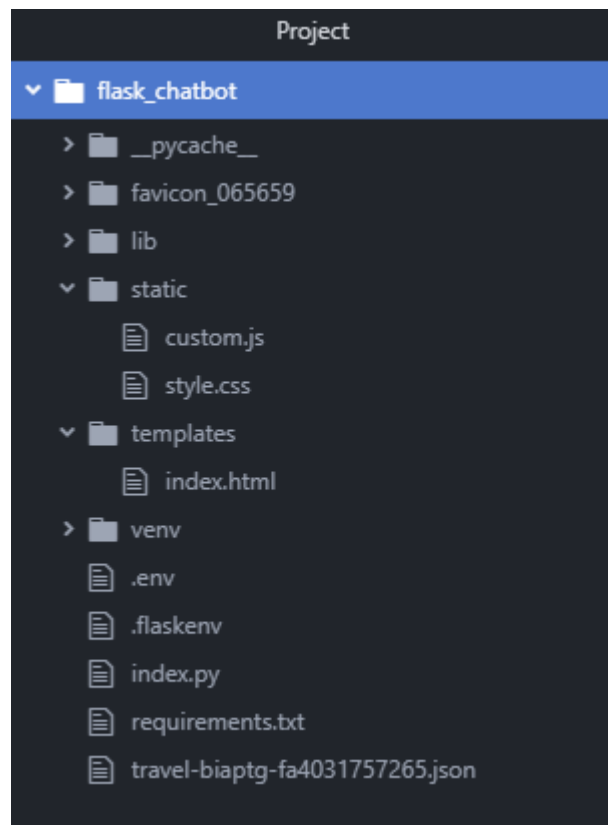


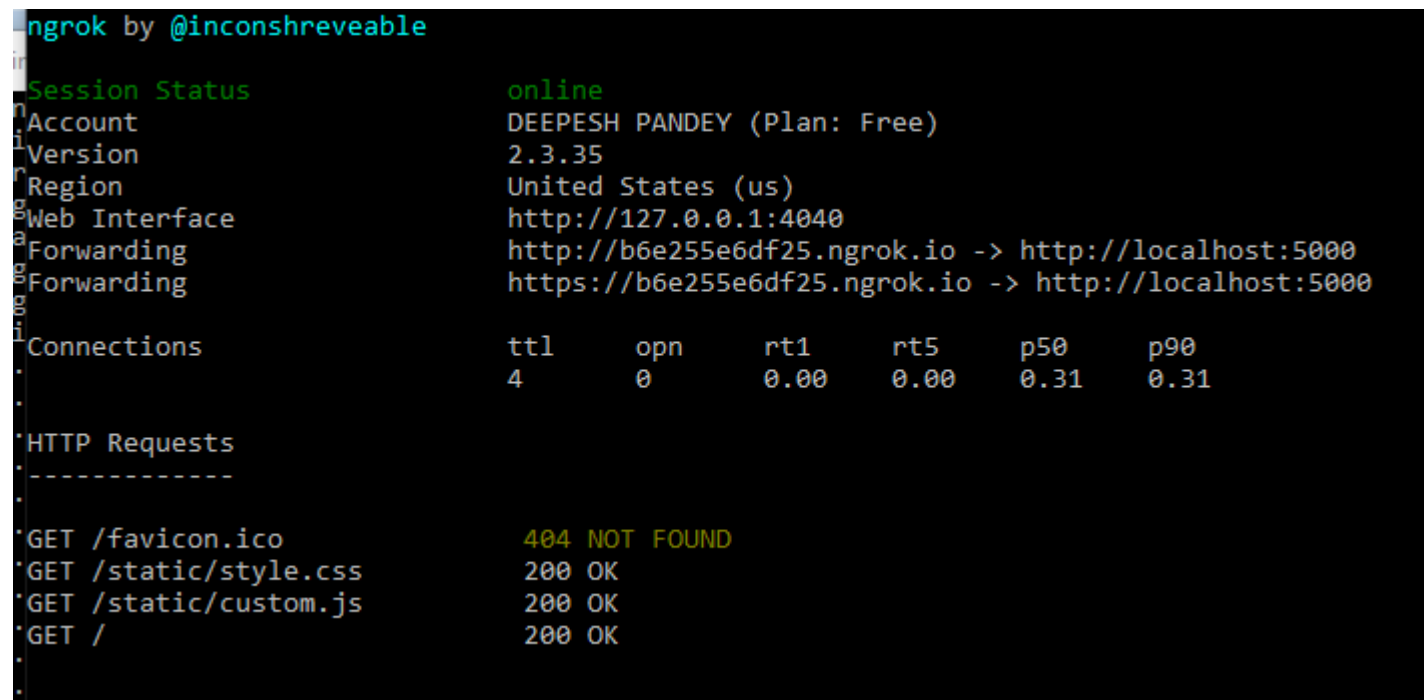
FIGURE 1.1 STRUCTURE OF THE FLASK APP

There is an index.html file which contains the code which describes the layout of the website on which my chatbot is deployed. The .json file at the end of the folder is the file needed to enable my project on dialogflow to be integrated in my website. The style.css file is yet another file which is used in styling of the website, the layout of the chatbot display and other small entities as well.

Finally after completion of the chatbot and enabling the integration on my website, I had the challenge to make it accessible to everyone on the internet. Here is where ngrok comes in the picture. Ngrok enables me to expose my local host to the web and accessible to anyone using a secured tunnel.

When I run the following command on the command prompt I am provided a secured url which exposes my local host to the web on which my flask app is running.

```
ngrok http 5000
```



```
ngrok by @inconsreveable
Session Status      online
Account            DEEPESH PANDEY (Plan: Free)
Version            2.3.35
Region             United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding          http://b6e255e6df25.ngrok.io -> http://localhost:5000
Forwarding          https://b6e255e6df25.ngrok.io -> http://localhost:5000

Connections
  ttl    opn    rt1    rt5    p50    p90
    4      0    0.00   0.00   0.31   0.31

HTTP Requests
-----
GET /favicon.ico      404 NOT FOUND
GET /static/style.css 200 OK
GET /static/custom.js 200 OK
GET /                  200 OK
```

FIGURE 1.2 NGROK PROMPT DISPLAYING THE WEB URL

This led to successful completion of my project and this URL can be shared with anyone and accessed to screen themselves for COVID-19. This is how the final website appears.

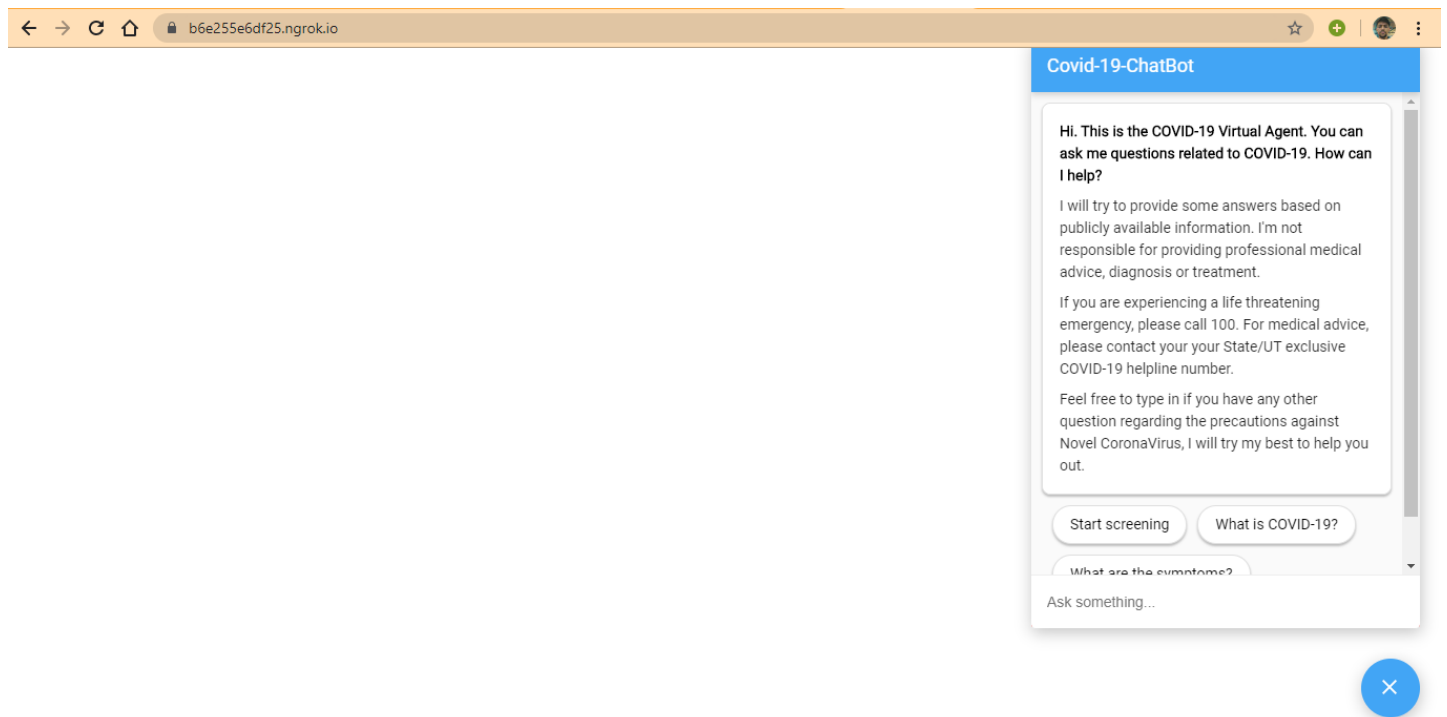


FIGURE 1.3 NGROK URL IN WORKING STATE

Also, this chatbot can also be integrated to make it accessible to anyone using Facebook Messenger like third party services.

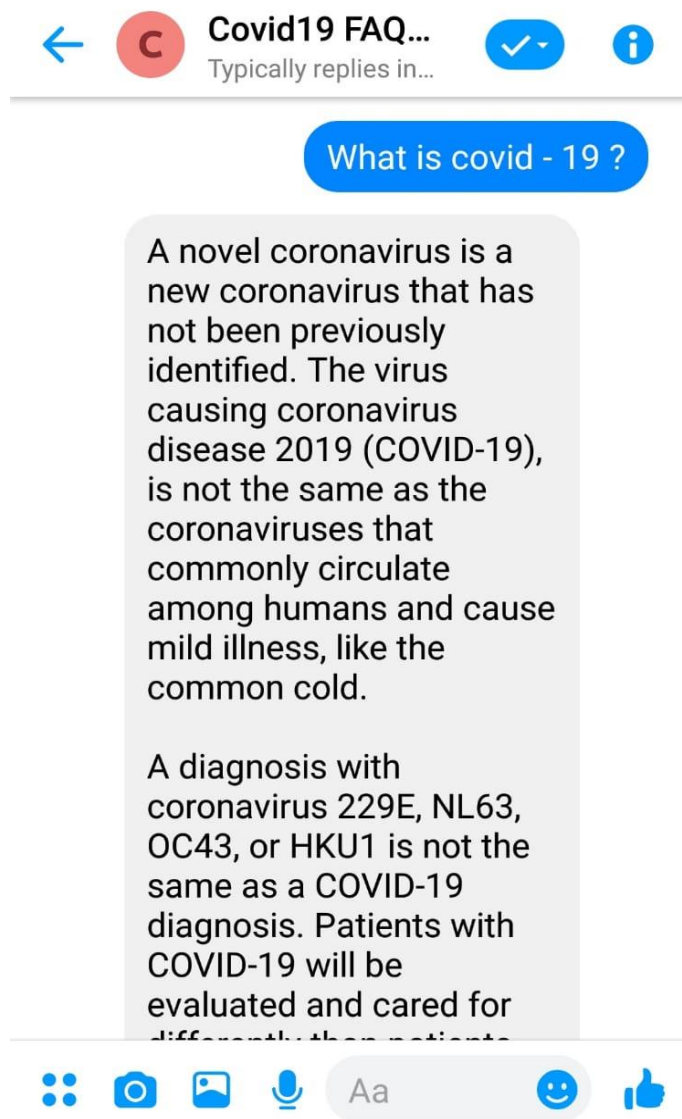


FIGURE 1.4 CHATBOT INTEGRATED WITH FACEBOOK MESSENGER

CHAPTER 2

2.1 NATURAL LANGUAGE PROCESSING (NLP)

Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software.

The study of natural language processing has been around for more than 50 years and grew out of the field of linguistics with the rise of computers.

The three important NLP viewpoints with respect to chatbots and other dialogue systems are:

1. Natural Language Understanding (NLU)
2. Natural Language Generation (NLG)
3. Dialogue Processing

2.1.1 NATURAL LANGUAGE UNDERSTANDING (NLU)

NLU involves the process of extracting meanings from text inputs. In this direction, the basic design steps include:

- **Syntactic Parsing** - determines the function of each word (part-of-speech), the way words are related to each other, how they are grouped into phrases and how they can modify each other. The context-free grammar (CFG) definition and parsers implementation are the common NLP techniques used in this step.
- **Semantic Parsing** - The role of a semantic parser is to extract the context-independent meaning of a written sentence. The discriminative methods such as support vector machines (SVM) and statistical methods such as decision trees and classification and regression trees (CART) are used to find the most probable parse tree that fits the sentence.
- **Contextual Interpretation** - refines the semantic interpretation by taking advantage of information at the discourse level and, removing remaining ambiguities such as anaphors, pronouns and ellipses. Discourse entity (DE) list maintains a set of constants referring to objects that have been evoked in previous sentences and can subsequently be referred implicitly.

2.1.2 NATURAL LANGUAGE GENERATION (NLG)

NLG involves the building of responses for the chatbots based on the processing done in the NLU step.

- **Artificial Intelligence Markup Language (AIML)** – an XML-compliant language designed to develop the AI flows in dialogue systems. Its purpose is to simplify the job of conversation modelling. The general structure of the AIML objects looks as follows:

```
<category>
  <pattern> User input </pattern>
  <template> Corresponding response to the user input </template>
</category>
```

The important elements of AIML are:

- **Categories** – basic units of knowledge consisting of patterns and templates
- **Recursion** – used to recursively match the other categories and thereby, simplify the complex grammatical forms. It is represented as *<srai>* tag.
- **Context** – the category tag uses the context *<that>* tag to refer to the previous input of the user.
- **Variables** – used to support the getting and setting of commonly used texts using *<get>* and *<set>* tags that are generally used to store proper nouns.
- **Pronoun Swapping** – used for replacement of pronouns such as ‘you’re’ with ‘I’m’ and ‘your’ with ‘my’ etc.
- **Pattern matching** – mainly used in question-answering chatbot systems for simple statements and natural language enquiring.
- **Response generation**
 - Document planning – brakes the high-level communicative goals into structured representations of atomic communicative goals
 - Micro-planning (Sentence planning) – is the phase where the number of generated clauses is decided in order to produce language with improved naturalness. Methods such as semantic grammars – reverse parsing are used in generation of unnatural proto-phrases.
 - Surface realization – the process of transforming the abstract structure obtained in the micro-planning stage into surface linguistic structures by adding function words, inflecting words, determining word order etc.

- **Chat script** - the technique that helps when no matches occur in AIML. It concentrates on the best syntax to build a sensible default answer.
- **Markov chain** - used to build responses that are more probabilistically applicable and consequently are more correct. The idea of Markov chains is that there is a fixed probability of occurrences for each letter or word in the same textual data set.
- **Language tricks** - includes the sentences, phrases, paragraphs available in chatbots in order to add variety to the knowledge base and make it more convincing. The types of language tricks are: canned responses, typing errors and simulating key strokes, model of personal history and Non-Sequitur (not a logical conclusion). Each of these language tricks is used to satisfy a specific purpose and to provide alternative answers to questions.
- **Ontologies/Semantic networks** – consists of set of relationally and hierarchically interrelated concepts. The aim of using ontologies in a chatbot is to compute the relation to synonyms, hyponyms and other relations which are natural language concept names. The interconnection between these concepts can be represented in a graph enabling the computer to search by using particular rules for reasoning.

2.1.3 DIALOGUE PROCESSING

Aims at building a man-machine dialog based on turn-taking process wherein the information is transferred from one participant – the user and the other participant – dialog manager (DM). The process of dialogue management involves implementation of the interaction strategy to organize the sequence of system dialogues to achieve the common goal of the user and chatbot. The level of user satisfaction of a chatbot is heavily influenced by the concept of “Degree of Initiative” –

- **System-Led** – an initiative that is completely controlled by the system wherein it asks precise questions to the user and expects information or answers from the user.
- **User-Led** – an initiative that is led by the user while the system is expected to provide information to the user queries without asking for more details.
- **Mixed-Initiative** – an initiative in which both the participants interact cooperatively to achieve the conversation goal. For example, our health chatbot shares the control between the user and system.

Relational databases enable the chatbot to remember past conversations and make the conversation more continuous and meaningful by building knowledge bases. The Structured Query Language (SQL) allows generation of queries and query block nesting to save the conversation history. This mainly makes the search of a word and phrase match easier.

2.2 TYPES OF CHATBOT MODELS

In general, we have two types of chatbot models:

	Retrieval - Based	Generator - Based
Description	Simple and pre-defined heuristic model	Smart model generating new response from scratch
Grammatical errors in responses	Rarely involves errors in grammar	Can be prone to grammatical errors

Behavior towards unseen cases	Not handled	Handles efficiently
Contextual responses	Not considered	Considers context for response generation
Length of conversation	Long	Short
Domain	Open	Closed

TABLE 2.1 TABLE SHOWING TYPES OF CHATBOTS

2.3 CHATBOT PROCESSING MECHANISM

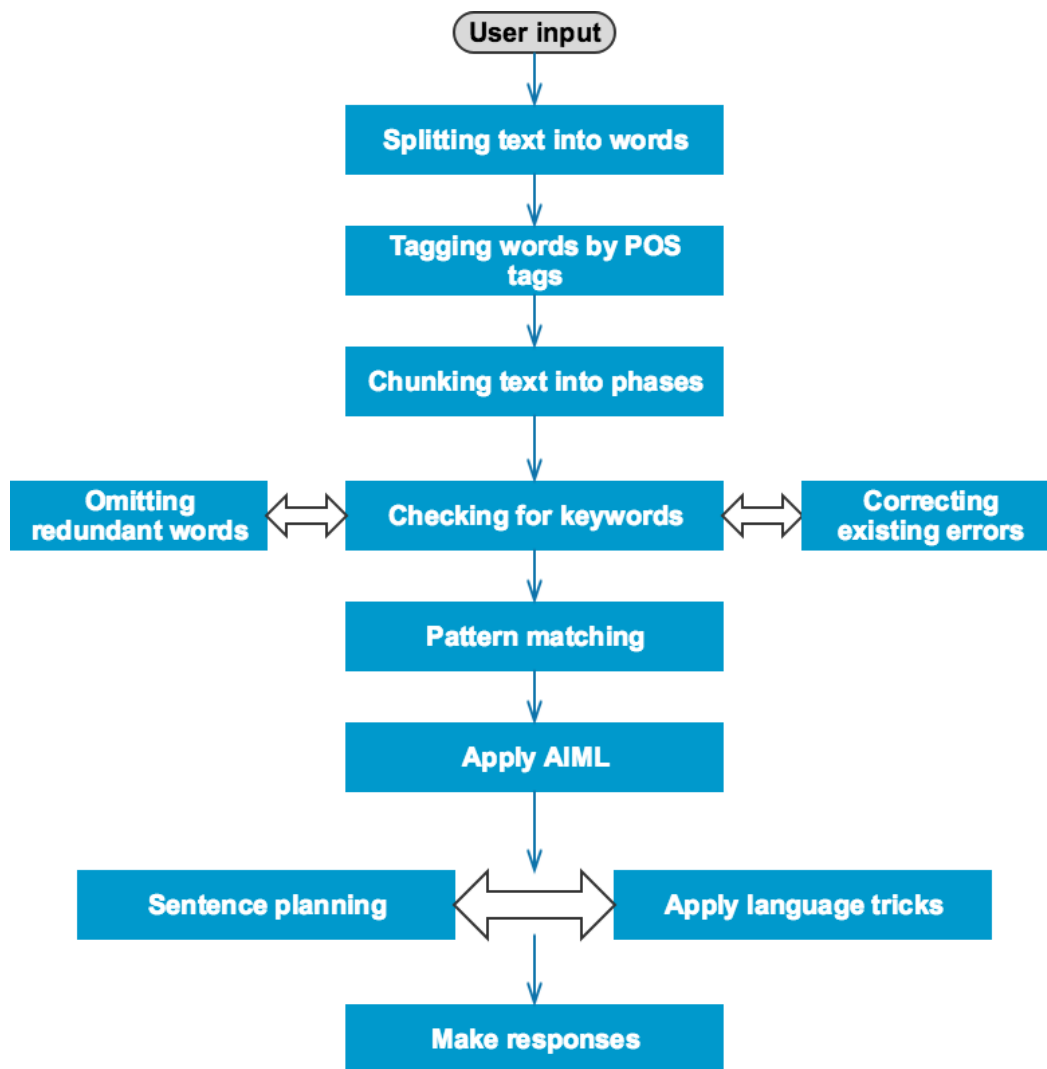


FIGURE 2.1 A FLOWCHART OF THE PROCESSING MECHANISMS FOR A CHATBOT

2.4 CHATBOTS

Chatbots and virtual assistants are becoming part of our daily life and they are also becoming smarter every day. There is a lot of scope, many inventions to be made in this domain and the use cases are plenty. This was one of our motives to choose this topic. We explored the various platforms and tried our hand in three of the famous ones: wit.ai, luis.ai and Dialogflow. Table-1.1 summarizes these three platforms.

2.5 COMPARISION CHART

Platforms	wit.ai	luis.ai	Dialogflow
Description	<ul style="list-style-type: none"> Allows to easily create text or voice-based bots that humans can chat with on their preferred messaging platform. It allows to understand user input after certain training, identify intents, extract entities, and predict what the bot should do based on the current context and user query. 	<ul style="list-style-type: none"> Introduced by Microsoft. The Microsoft Bot framework helps us to build, test and deploy bots for many well-known platforms such as Facebook, Slack etc. There are some pre-built domains that we can import to our chatbot together with its entities, intents and utterances. 	<ul style="list-style-type: none"> Formerly known as api.ai, this Google acquired platform is popular for building conversational interfaces. Offers more than 30 pre-built agents such as navigation, hotel booking, small talk, weather, news etc.
Pros	<ul style="list-style-type: none"> The concept of story is really powerful and allows controlling the conversation flow using branches and conditions on actions. An “Inbox” exists, where the requests that could not be processed by the chatbot are listed, so the developers can train the bot. 	<ul style="list-style-type: none"> More feasible for professional purposes and .NET developers. The enterprise version can be integrated with other application provided by Microsoft Azure to enhance functionalities. 	<ul style="list-style-type: none"> One-click integration with several platforms like Slack, Telegram, Facebook Messenger, Twitter, Amazon Alexa, Google Assistant etc. Built-in voice interface and support for Node.js, Ruby, Android, iOS, Python etc. SDKs.
Cons	<ul style="list-style-type: none"> wit.ai has integration with Facebook messenger only. Stories are in beta. There are cases where it is difficult to control the flow of the conversation and the bot tends to misunderstand the user requests. 	<ul style="list-style-type: none"> luis.ai has 10,000 transactions for each month and is priced after that. 	<ul style="list-style-type: none"> Priced for enterprises requiring high utilization of Dialogflow. It is impossible to block the matching of an intent if a context is present.

TABLE 2.2 COMPARISION CHART

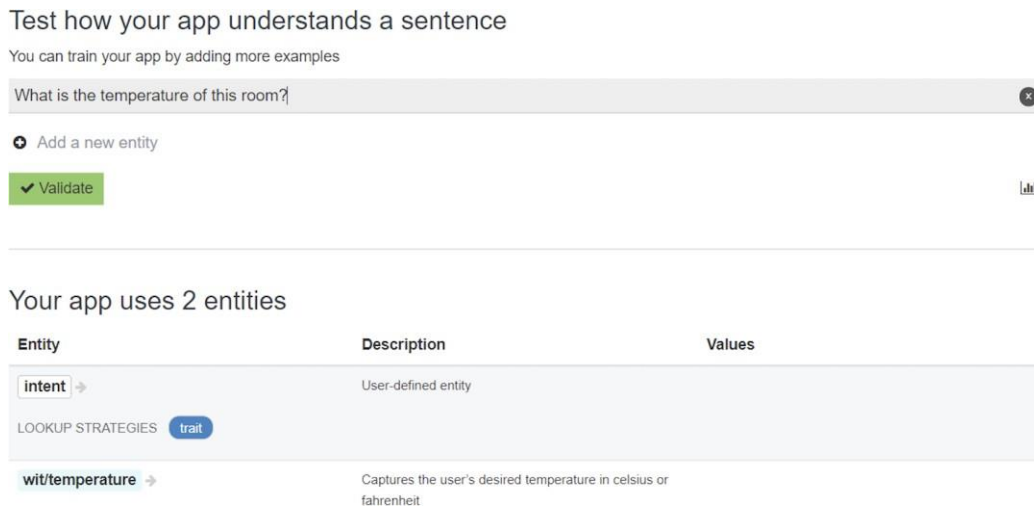


FIGURE 2.2 WIT.AI CONSOLE

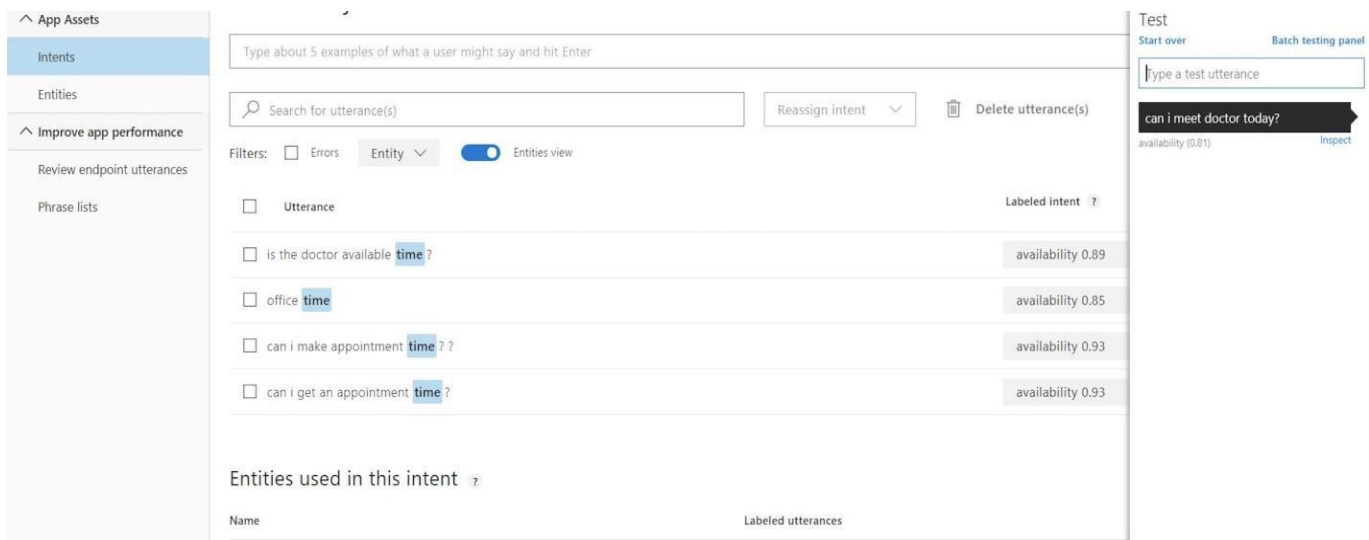


FIGURE 2.3 LUIS.AI CONSOLE

2.6 HEALTHCARE CHATBOTS

Healthcare domain was an interesting and challenging area for me to target. There are already chatbots available in areas like flight booking, hotel management, food ordering services, home automation etc. Healthcare is one of the areas where there is a lot of research and opportunities with respect to utilizing chatbots. Most of the chatbots available in this domain deal with hospital management like scheduling an appointment, getting to know the availability of the doctors, managing prescriptions, heart rate and blood pressure monitoring, reporting symptoms and illness etc. Some of the healthcare chatbots in these domains are Your.MD, Florence and GYANT etc.

CHAPTER 3

3.1 DIALOGFLOW



FIGURE 3.1 DIALOGFLOW LOGO

Dialogflow, now owned by Google, is a conversational UX platform enabling brand unique, natural language interactions for devices, applications, and services. In other words, Dialogflow is a framework which provides NLP / NLU (Natural Language Processing / Natural Language Understanding) services. More than that, Dialogflow supplies one-click integration with most of the popular messaging platforms such as Facebook, Telegram, Twitter, Viber, Kik, etc. and also supports voice assistants such as Google Assistant and Amazon Alexa.

Dialogflow is a natural language understanding platform used to design and integrate a conversational user interface into mobile apps, web applications, devices, bots, interactive voice response systems, and so on.

A Dialogflow agent is similar to a human call center agent. You train them both to handle expected conversation scenarios, and your training does not need to be overly explicit.

I leveraged most of what Dialogflow has to offer to build my “COVID-19 Screening And FAQs Chatbot”. The scope of the problems is vast, but we have narrowed to a small set such as getting answer some of the most crucial FAQs related to precautions and research regarding Novel Coronavirus, apart from the screening.

3.2 FLASK



FIGURE 3.2 FLASK LOGO

3.2.1 WHAT IS FLASK ?

It makes the process of designing a web application simpler. Flask is a web framework. This means **flask** provides you with tools, libraries and technologies that allow you to build a web application.

This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

3.2.2 WHAT IS HTTP AND WHAT IT HAS TO DO WITH FLASK ?

HTTP is the protocol for websites. The internet uses it to interact and communicate with computers and servers. When you type the name of a website in the address bar of your browser and you hit enter. What happens is that an HTTP request has been sent to a server.

For example, when I go to my address bar and type google.com, then hit enter, an HTTP request is sent to a Google Server. The Google Server receives the request and needs to figure how to interpret that request. The Google Server sends back an HTTP response that contains the information that my web browser receives. Then it displays what you asked for on a page in the browser.

3.2.3 HOW IS FLASK INVOLVED ?

I wrote a code that will take care of the server side processing. My code will receive requests. It will figure out what those requests are dealing with and what they are asking. It will also figure out what response to send to the user. To do all this I have used Flask.

3.3 NGROK



FIGURE 3.3 NGROK LOGO

Ngrok is a cross-platform application that enables developers to expose a local development server to the Internet with minimal effort. The software makes your locally-hosted web server appear to be hosted on a subdomain of ngrok.com, meaning that no public IP or domain name on the local machine is needed. Similar functionality can be achieved with Reverse SSH Tunneling, but this requires more setup as well as hosting of your own remote server.

Ngrok is able to bypass NAT Mapping and firewall restrictions by creating a long-lived TCP tunnel from a randomly generated sub-domain on ngrok.com (e.g. 3gf892ks.ngrok.com) to the local machine. After specifying the port that your web server listens on, the ngrok client program initiates a secure connection to the ngrok server and then anyone can make requests to your local server with the unique ngrok tunnel address.

Various tunnel servers are available around the world and locations include: US (Ohio), Europe (Frankfurt), Asia (Singapore), and Australia (Sydney). Alternatively, the ngrok server software can be self-hosted on a VPS or dedicated server.

By default, ngrok creates both HTTP and HTTPS endpoints, making it useful for testing integrations with third-party services or APIs that require valid SSL/TLS domains. Other use cases include: quickly showcasing local demos to clients, testing mobile application backends, and running personal cloud services from your home PC.

One praised feature of ngrok is the ability to track and replay HTTP requests via ngrok's web console (accessible on <http://127.0.0.1:4040/>). The replay functionality is highly useful when testing API calls or webhooks as one can easily inspect all header content and request/response data in one place via the console UI.

CHAPTER 4

4.1 IMPEMENTATION

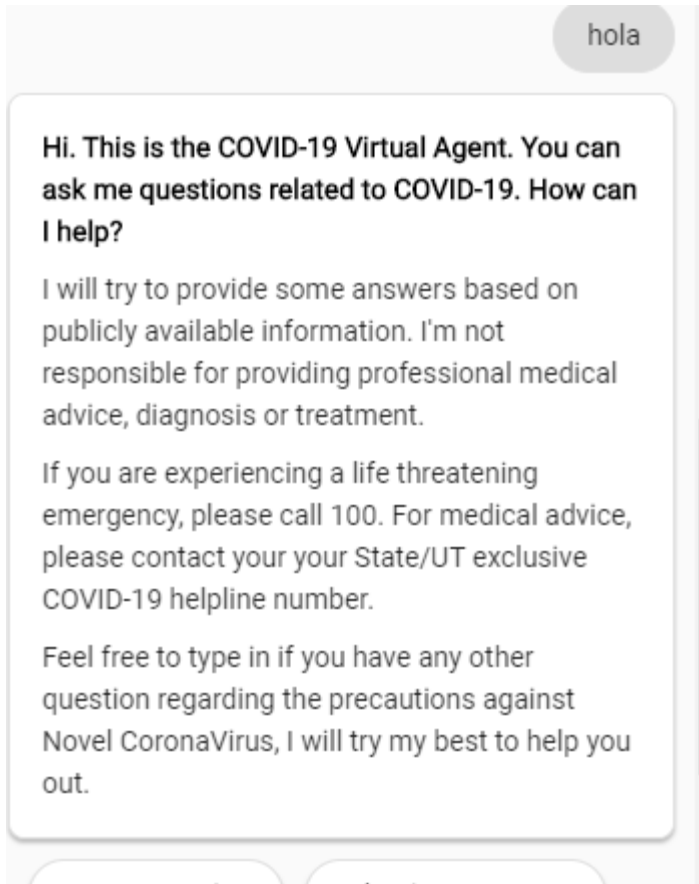
4.1.1 USING DIALOGFLOW TO BUILD THE CHATBOT

All of the above software and technologies combined together lead to the successful completion of this project. Here I will display the intents and entites used to build the chatbot and how it is able to answer a vast variety of questons.

The process that Dialogflow follows from invocation to fulfillment is similar to someone answering a question, with some liberties taken of course. In order to start a conversation with an agent, the user needs to invoke the agent. A user does this by asking to speak with the agent in a manner specified by the agent's developer. For the agent to understand what the user said, it needs examples of how the same text can be said in different ways. These are added to training phrases.

Figure-4.1 shows the user expressions that we have configured for the welcome intent. Note that, we do not have "Hola" as training phase. But the agent is trained using NLP and machine learning techniques such as word embedding, user data, lexical synonyms, bag- of-words, synonym detection, regular expressions, tokenization, tags, supervised learning etc. and hence was able to detect what the user said.

We do not need to input all possible user inputs, but only need to put in few sentences or words and the powerful algorithms in Dialogflow takes care of the rest. We observe a lot of other techniques here in this small example. The case of the words is not taken into consideration, but punctuation and special characters are. The Dialogflow agent needs to know what information is useful for answering the user's request.



Ask something...

FIGURE 4.1 CHATBOT IS ABLE TO IDENTIFY THE INTENT DESPITE THE TRAINING PHRASE DOESN'T MATCHES

Training phrases ?

”	Add user expression
”	Hi
”	yo
”	Hey
”	Hello there
”	Good evening
”	Good afternoon
”	Good morning
”	Hello

FIGURE 4.2 TRAINING PHRASES

4.1.1.1 AGENTS

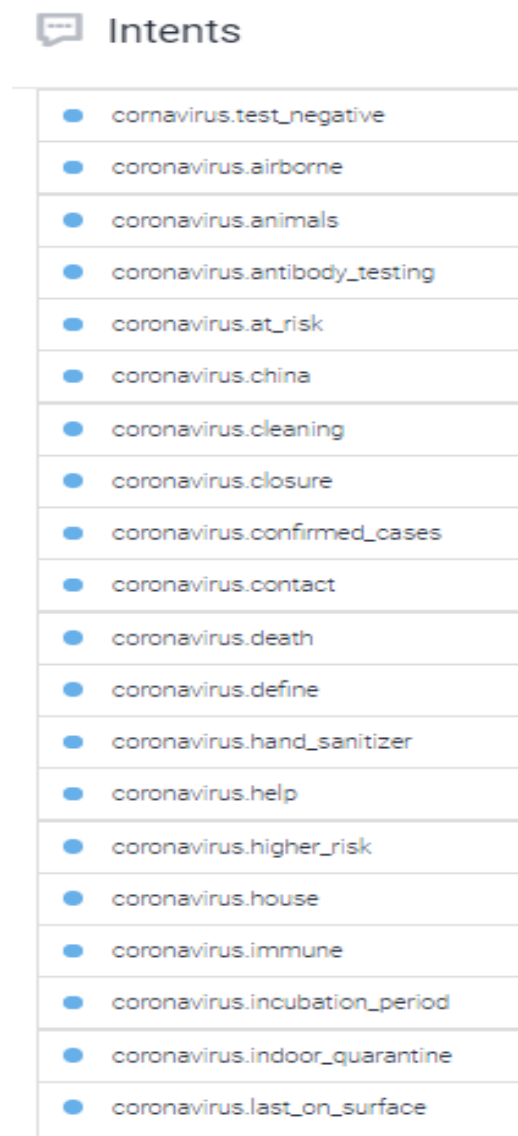
Agents are best described as NLU modules. These are included in the app to transform natural user requests into actionable data. Agents can also be designed to manage a conversation flow in a specific way. There are a number of pre-built agents available in Dialogflow.

4.1.1.2 INTENTS

An intent represents a mapping between what a user says and what action should be taken. Intent interfaces have the following sections:

- Training Phrases
- Action
- Response
- Contexts

The reason is, getting the mood of a person is a very diverse question. And the possibility on how a user might respond depends on each individual. Hence, we have a lot of phrases here and the NLP and ML algorithms train these sentences and can understand similar sentences from the user even if there are typos. More the training phrases, more the agent learns. The system detects the correspondence between the words (or phrases) and existing developer and system entities and highlights such words and phrases. It also automatically assigns a parameter name to each detected entity. Figure-1.7 shows the action and parameters for getMood intent. The colors in Figure-1.6 correspond to the entities shown in Figure-1.7. For example, the word awful is negative and is tagged with negativeFeelings entity.



The image shows a screenshot of a chatbot interface. At the top, there is a heading 'Intents' with a speech bubble icon. Below this, there is a list of 20 intents, each preceded by a blue circular bullet point. The intents are listed in a table-like structure with horizontal lines separating them.

Intents
coronavirus.test_negative
coronavirus.airborne
coronavirus.animals
coronavirus.antibody_testing
coronavirus.at_risk
coronavirus.china
coronavirus.cleaning
coronavirus.closure
coronavirus.confirmed_cases
coronavirus.contact
coronavirus.death
coronavirus.define
coronavirus.hand_sanitizer
coronavirus.help
coronavirus.higher_risk
coronavirus.house
coronavirus.immune
coronavirus.incubation_period
coronavirus.indoor_quarantine
coronavirus.last_on_surface

FIGURE 4.3 INTENTS

Add user expression

Can I go out yet

can i go out

What does shelter in place mean?

What is shelter in place?

Can I still go to work?

Can I still take a walk in San Francisco?

Can I still take a walk in the neighborhood?

still jog

Can I still go take a hike?

shelter in place

1 OF 5

FIGURE 4.4 SEVERAL TRAINING PHRASES

Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	geo-city	@sys.geo-city	\$geo-city	<input type="checkbox"/>
<input type="checkbox"/>	outdoor_activities	@outdoor_activities	\$outdoor_activities	<input type="checkbox"/>
<input type="checkbox"/>	age	@sys.age	\$age	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

FIGURE 4.5 ACTIONS AND PARAMETERS

Figure-1.8 shows the agent’s responses which will be provided by the agent when the intent is triggered. One of the responses is randomly selected and this in turn keeps the conversation going. Different responses will make the agent more human-like.

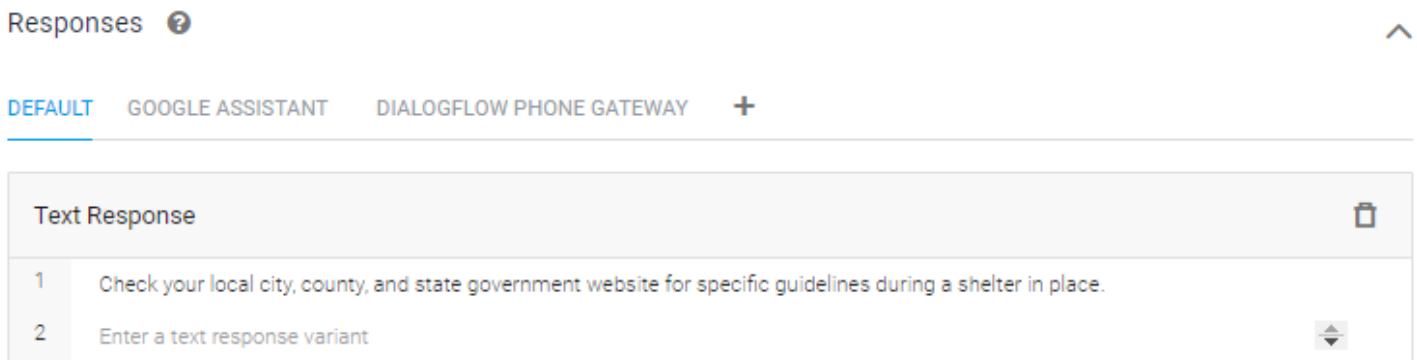


FIGURE 4.6 RESPONSES FOR OUTDOOR ACTIVITIES INTENT

4.1.1.3 ENTITIES

Entities are powerful tools used for extracting parameter values from natural language inputs. Any important data we want to get from a user’s request, will have a corresponding entity. There are three types of entities: system (defined by Dialogflow. Example: given names, address, phone numbers, color, temperature etc.), developer (defined by a developer), and user (built for each individual end-user). Figure-1.9 shows the entities for “activities”. When the bot wants to learn about the activities that the user likes to do, there are innumerable possibilities and we have defined a few as shown in the figure. But, Dialogflow is intelligent enough to understand all the synonyms and continuously learns more synonyms.

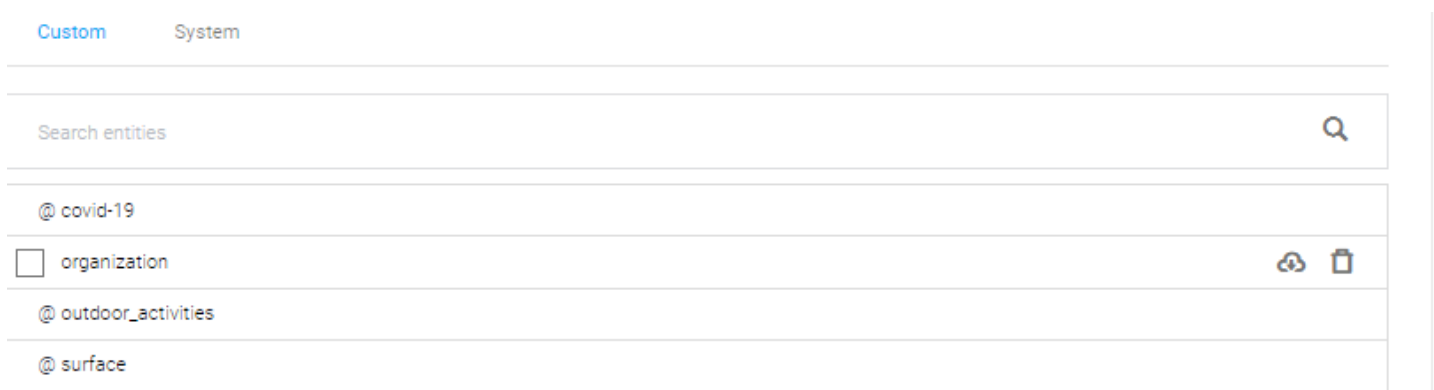


FIGURE 4.7 ENTITES FOR MY CHATBOT

Figure-2.0 shows the “activities” entity usage in the conversation. Although, there is no such activity as “I like playing computer games” in our entity list, Dialogflow recognizes that “playing computer games” is an activity that a person would like to do.

4.1.1.4 CONTEXTS

Contexts represent the current context of a user’s request. This is helpful for differentiating phrases which may be vague or have different meanings depending on the user’s preferences, location or the topic of conversation. Contexts are designed for passing on information from previous conversations and can be used to manage the conversation flow. Each intent can have several input and output contexts. The output contexts from an intent will act as an input context to another intent.

4.1.2 BUILDING APP USING FLASK

The following script snippet in the index.html file allows us to embed the chatbot we build earlier in our website.

```
<script src="https://www.gstatic.com/dialogflow-  
console/fast/messenger/bootstrap.js?v=1"></script>  
<df-messenger  
  intent="WELCOME"  
  chat-title="Covid-19-ChatBot"  
  agent-id="8545fa13-f407-4e55-8162-535d96e4f943"  
  language-code="en"  
></df-messenger>
```

This will display a pop up message whenever the user access the website and clicking that pop up message initiates the chat with the chatbot.

The index.py is the main script which runs our flask app. The snippet for it is as follows. Here we are importing the Flask module and creating a Flask web server from the Flask module.

This will run the application. Having debug=True allows possible Python errors to appear on the web page. This will help us trace the errors.

```
from flask import Flask, request, jsonify, render_template
import os
import dialogflow
import requests
import json
import pusher
app = Flask(__name__)

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

# run Flask app
if __name__ == "__main__":
    app.run()
```

4.1.3 USING NGROK TO EXPOSE THE LOCALHOST

Ngrok is a cross-platform application that enables developers to expose a local development server to the Internet with minimal effort. The software makes your locally-hosted web server appear to be hosted on a subdomain of ngrok.com, meaning that no public IP or domain name on the local machine is needed.

```
ngrok by @inconsreveable

Session Status      online
Account             DEEPESH PANDEY (Plan: Free)
Version             2.3.35
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://b6e255e6df25.ngrok.io -> http://localhost:5000
Forwarding           https://b6e255e6df25.ngrok.io -> http://localhost:5000

Connections         ttl      opn      rt1      rt5      p50      p90
                   4        0        0.00     0.00     0.31     0.31

HTTP Requests
-----
GET /favicon.ico     404 NOT FOUND
GET /static/style.css 200 OK
GET /static/custom.js 200 OK
GET /                 200 OK
```

FIGURE 4.8 NGROK PROMPT SHOWING THE WEBSITE URL

On visiting the https link we will be redirected to our website where our localhost is securely exposed to the internet users.

The final website after successful completion looks like this in working stare.

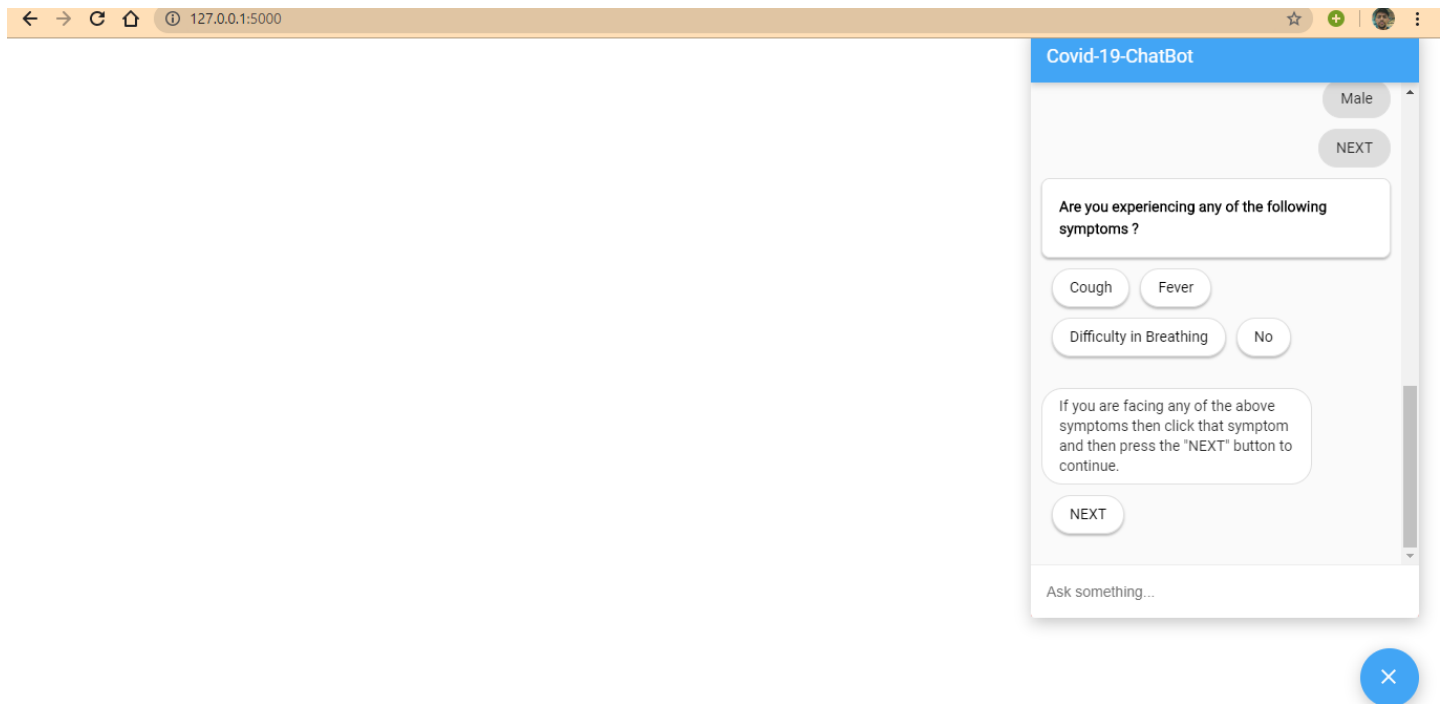


FIGURE 4.9 FINAL WEBSITE FOR MY CHATBOT

We can see that the bot is asking the series of questions to determine how severe is the risk of Coronavirus to the person taking the screening test. At the end it will display the severity level and the level and what should he/she do to remain safe.

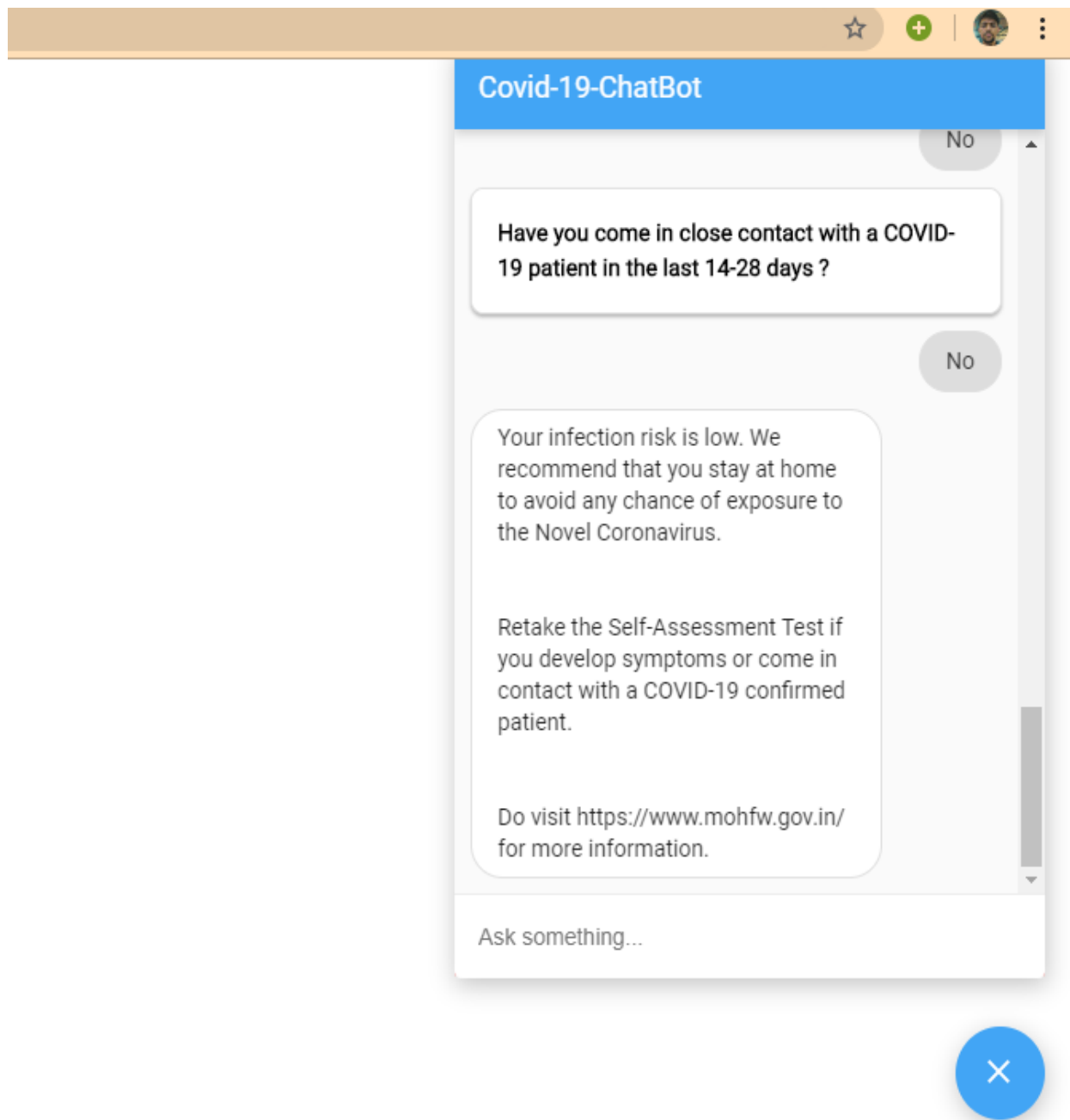


FIGURE 4.10 REPLYING SHOWING LOW RISK

It provides appropriate steps to be followed by the by the user in case he/she has a high risk of getting infected as well.

It can as well answer hundred of questions related to COVID-19 which are essential information. Some of the questions are as follows:

Is the virus airborne?, Can I be tested for antibodies?, What should I do if I've met someone with coronavirus?, What should I do if I have risk factors?, What is the Wuhan virus?

If I have recovered from COVID-19, Will I be immune to it?, What cleaning products should I use to protect against COVID-19?, I'm definitely sick and still waiting to be tested. What else should I do?, Does the weather have an impact on the virus? , Will coronavirus harm the fetus in pregnant women? , Can my pets get the virus? , Why is social distancing so important?

These are a few questions of the many and the questions can be rephrased as well and still the bot will be able to identify the similarity and answer those questions with equal efficiency.

CHAPTER 5

5.1 LEARNING OUTCOME

First, I learned how to make efficient chatbots with state-of-the-art technologies like NLP which makes use of machine learning using Dialogflow.

- Making chatbots to identify the query of the user by identifying the keywords in the message as I explained earlier using the flowchart which my chatbot uses to scoop out the essential information and separate it from the other useless information.
- Using contexts and entites to better interact with the user and call the exactly appropriate intent which should be triggered and reply to the user most appropriate response. Contexts help to trigger the correct intent using certain keywords extracted from the message entered.
- Integrating chatbots in websites using features of DialogdfLOW to enable everyone to use my chatbot.

Second, I learned about the difference between a static website and a web application.

Static Websites:

- Means that the server is serving HTML, CSS, and JavaScript files to the client. The content of the site does not change when the user interacts with it.

Web Applications

- A web application or dynamic website generates content based on retrieved data (most of the time is a database) that changes based on a user's interaction with the site. In a web application, the server is responsible for querying, retrieving, and updating data. This causes web applications to be slower and more difficult to deploy than static websites for simple applications.

Server Side and Client Side.

- I learned that a web application has two sides. The client side and the server side. The client side is what the user interacts with and the server side is where the all the information that the user inputted is processed.

Third, I learned about Cloud Services.

Using Google Cloud Platform (GCP) I also learnt how to deploy my application to the cloud and get a permanent we address for my website which would remove the hassles to use a random ngrok url everytime.

Fourth, I learned how to use Python as a Server Side Language.

To create the server side of the web application we had to use a server side language. I learned that I could use the framework called Flask to use Python as the Server Side Language.

5.2 IMPORTANCE OF CHATBOTS IN INDUSTRY

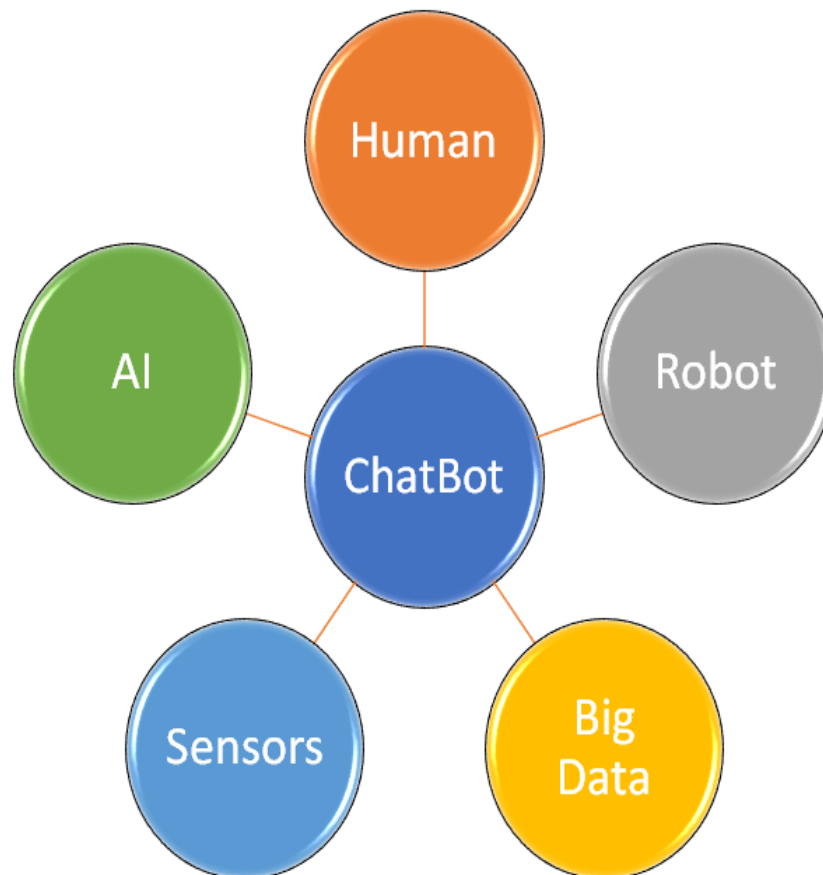


FIGURE 5.1 CHATBOT APPLICATIONS

HOW CHATBOTS ARE DOMINATING IN DIFFERENT FIELDS

1. Improve Customer Service

- Extensive Customer Assistance:

83% of online shoppers need support during shopping. So your customers may require help trying to understand which product fit their needs/budgets anytime of the day. Some get lost due to navigation issues on your site. Some get mixed up. Sometimes there may be missing information about a product. In these situations, chatbots can provide assistance in real-time like a sale person in a real store. Chatbots can present it customers rich content with product pages, images, video, price and can even asked questions.

- Always -Available Customer support

Customer support process can be improved with the help of chatbots. They can be programmed to give automatic answers to repetitive questions and forward the request to a real person when a more complicated action is needed. So the agent focus more on important cases rather than on the easy question.

- Proactive Customer Interaction

Companies apply a “passive customer interaction” which means they only responds to customers when they are connected and not initiate the communication. In 2018, no company has the manpower to initiate a communication with all their customers periodically, chatbots allow you to start a conversation with each customer regarding any issue, anytime of the day. This will make your clients feel your brand is one step ahead and improve your brand perception in the long run.

My project uses the AI which is a real game-changer in the industry. Now, I will review some of its applications in different industries.

Here are some of the significant AI applications in major industries

1. Healthcare

The entry of the technology giants like Microsoft, Google, Apple and IBM in the healthcare sector holds significant importance for the industry. AI is currently being applied for a range of healthcare needs, including data mining for identifying patterns and then carrying out the more accurate diagnosis and treatment of medical conditions, medical imaging, medication management, drug discovery and even robotic surgery.

A case in point in this regard is the application of IBM Watson (an AI tool) to derive the meaning and context of a set of structured and unstructured data that might be critical for selecting a treatment plan, and then analyze the patient's medical record to identify a potential treatment plan. In other words, IBM Watson functions like a human doctor.

Similarly, a platform called Artificial Intelligence for Drug Discovery (AIDD), developed by the biopharma company NuMedii leverages big data and AI to detect the link between diseases and drugs at the systems level.

2. Retail and E-commerce

Retail and E-commerce is perhaps the only space where the application of AI is the most discernible to the majority of end-users. Being a competitive space, organizations are always looking out for techniques to find patterns in consumer behavior and thereby align their strategy to outsmart their competitors.

AI has certainly found a sweet spot in the entire scheme of things. The product recommendations on your Amazon account are nothing but a real-time application of complex AI algorithms to determine which products you are more likely to buy.

AI applications are also increasingly being used to enhance the customer experience. For example, many of the chatbots available on websites are powered by AI and programmed to provide instant answers to a range of common customer queries.

REFERENCES

- <https://cloud.google.com/dialogflow/docs>
- <https://ngrok.com/docs>
- <https://flask.palletsprojects.com/en/1.1.x/>
- https://www.researchgate.net/publication/334836867_A_Literature_Review_On_Chatbots_In_Healthcare_Domain
- <https://machinelearningmastery.com/natural-language-processing/#:~:text=Natural%20Language%20Processing%2C%20or%20NLP,with%20the%20rise%20of%20computers.>
- <https://developers.google.com/assistant/conversational/rich-responses>