Chatbots in COVID-19 Pandemic: Challenges and Solutions

A Project Thesis

Submitted to the Department of Computer Science

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

in

Computer Science

University of Regina

By

Jay Manishkumar Joshi

Regina, Saskatchewan

March, 2022

ABSTRACT

A chatbot is an intelligent piece of software able to interact and execute identical acts to a human being. Research indicates 60% of call to doctors clinic are for small-scale diseases, 80% of which can be easily cured using simple home remedies. Due to the unprecedented effects of the COVID-19 pandemic, World Health Organization has encouraged people to reduce unnecessary hospital visits and utilize remote healthcare facilities. In this scenario, chatbots can be effectively used to deliver digital health care solutions. Hence, this research is focused on discussing novel methods using advanced cloud services to build and deploy easy to use, cost-effective, scalable, AI-enabled smart chatbots to provide digital health care solutions to mitigate the global effects of the COVID-19 pandemic. We have developed chatbots to support three different use cases related to COVID-19 pandemic. Firstly, a chatbot model is trained using Azure cloud services to answer user queries related to novel coronavirus. Secondly, we have closely work with domain experts at MD Canada to design a chatbot architecture to collect patient health information. Lastly, we have identified stacked deep learning models for user emotion detection in chatbots.

ACKNOWLEDGEMENT

First and foremost, I would like to thank my project supervisor Dr. Alireza Manashty. Without his sincere involvement and assistance in every single step throughout the process, this project would have never been accomplished.

This project work has received financial support in parts by Mitacs through the Mitacs Accelerate Program with their partner Organisation MD Canada Wellness Solutions. I would like to mention Dr. Walter and Mr. Steven Xue at MD Canada Wellness Solutions for their active contributions in guiding me through to this noble cause.

POST DEFENSE ACKNOWLEDGEMENT

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AI Artificial Intelligence

API Application Interface

CD Continuous Deployment

CI Continuous Integration

FAQ Frequently Asked Question

GRU Gated Recurrent Unit

HTTP Hypertext Transfer Protocol

IoT Internet of Things

JSON Java Script Object Notation

KB Knowledge Base

LSTM Long Short Term Memory

LUIS Language Understanding

NLP Natural Language Processing

NLU Natural Language Understanding

PHQ Public Health Questionnaires

QnA Question and Answers

RNN Recurrent Neural Network

SDK Software Development Kit

WHO World Health Organisation

YAML Yet Another Markup Language

Chapter 1

Introduction

1.1 Introduction to Chatbots

Artificial Intelligence (AI) involves a complex decision-making process to mimic the learning and problem-solving abilities of the human mind. AI has made a strong impact on how one engages itself with day-to-day activities by designing and developing intelligent virtual assistants, which can perform many routine life activities easily [14]. A chatbot is an intelligent piece of software able to interact and execute identical acts to a human being. Chatbots or virtual agents are mainly designed to carry out interactions between humans and computers.

In the year 1950, the famous mathematician and computer scientist "Alan Turing" wondered if a computer program can chat with a human being. The first-ever chatbot that passed the Turing test was ELIZA created by Joseph Weizenbaum in 1964 [56]. ELIZA was developed for imitating psychologists, which can conduct conversations with pattern matching techniques that made users feel that ELIZA can comprehend user text in an intelligent manner. ELIZA had many limitations, but it was responsible for evolutionary development in the field of conversational AI. The sample chat script from ELIZA is shown in Figure 1.1 below.

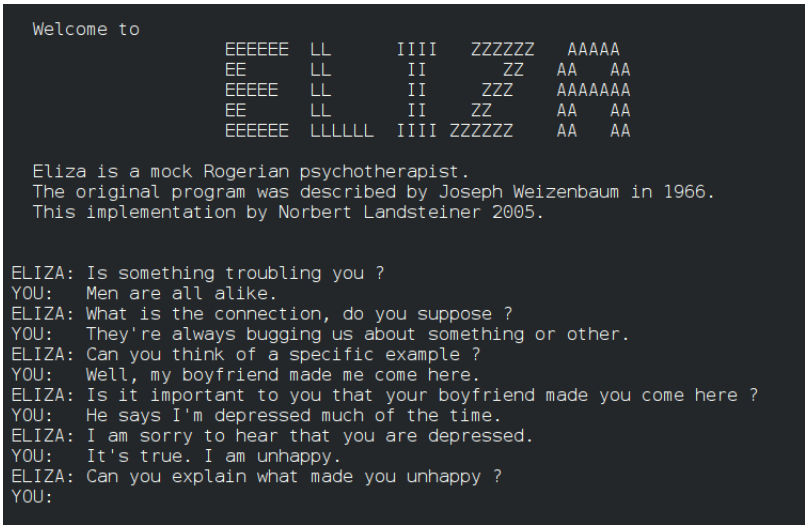

Figure 1.1: ELIZA Sample Chat Script [56]

In 1972, a chatbot titled PARRY was developed and was used extensively for the patients of schizophrenia [50]. It is considered more intelligent than ELIZA because it had a personality characteristic as humans and a better operating structure. In 1988, the Jabberwacky was the first scripted program to use AI in the domain of chatbots. Jabberwacky was designed and developed from CleverScripts, a programming language that is solely based on spreadsheets. Jabberwacky used contextual pattern matching and response generation techniques based on the history of chats [14].

The development of AI based chatbots made a tremendous rise in 2016 with the launch of virtual assistants which can be used in multiple devices like mobile phones and home speakers, which can identify and comprehend human and digital voices and can execute day to day tasks such as monitoring smart home devices, managing appointments, alarms, calendars, emails, etc. In addition, they can easily get connected to the internet and can respond to any of queries smartly.

Microsoft XiaoIce [60], is another AI-enabled chatbot that can fulfil human requests and assist them in socializing with the outside world. This was the first-ever chatbot deployed which had a comprehensive ability to generate an emotionally intelligent response to the end-user.

At the beginning of 2016, there was an evolutionary change in the field of AI and natural language processing that significantly changed how companies designed their business models and customer services. Social media platforms like Twitter, Facebook and KIK allowed developers to build and deploy chatbots to provide customers with a simple way to communicate their daily requirements via online messaging platforms. Nearly 34,000 chatbots were deployed in the domain of eCommerce, marketing, health care and education. In addition, the Internet of things (IoT) introduced a new digital era of connecting smart objects where conversational agents have proved to be effective in providing communication among them [14]. Also, thousands of chatbots with specific applications were developed and deployed on social messaging platforms, industries, and research schools.

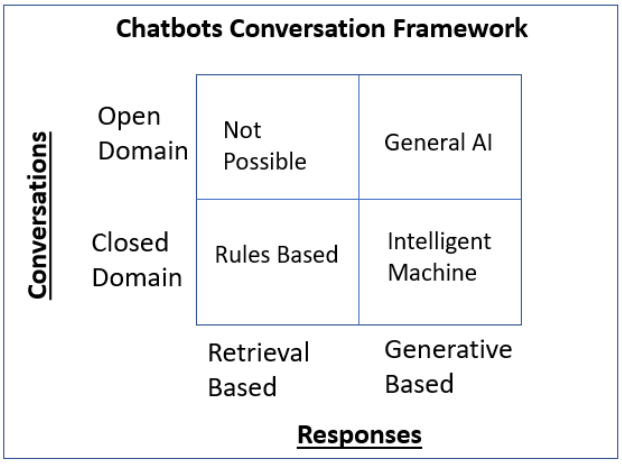

Figure 1.2: Types of chatbots [20]

Based on conversations and dialogue systems, chatbots can be divided broadly into two categories: open domain and closed domain systems. Open domain systems can conduct conversations about various general topics, while closed domain systems can only chat about a fixed type of information. For example, a chatbot that can order a pizza or restaurant menu is a closed domain, while a bot that can reply to general queries on Facebook, Google or Twitter is an open domain chatbot. In addition, a chatbot that can comprehend and respond to mostly all types of health

queries falls somewhere between open domain and closed domain systems.

Chatbots can also be classified as either retrieval or Generative-based [20]. Retrieval-based chatbots are similar to a rule-based system that can only answer user questions from well-defined datasets. Generative-based chatbots generate contextual responses by applying advanced machine learning and deep learning algorithms. Figure 1.2 represents different types of chatbots in a framework.

The primary aim of developing chatbots was to mimic human conversations. But nowadays, chatbots are useful in many different domains like eCommerce, education, marketing, health care, entertainment, and various other fields. Chatbots in eCommerce can be useful to provide product recommendations, online customer services, and many other additional functionalities. In education, significant studies are conducted to implement pedagogical agents and advanced tutoring systems.

There are several benefits that contribute to the fact that chatbots are relevant nowadays in almost every domain. Firstly, the implementation of chatbots can reduce customer service costs by replacing humans with virtual intelligent assistants. By the end of 2022, chatbots can reduce customer service cost to $22 billion [5]. In addition, chatbots can leverage customer engagement and satisfaction by significantly reducing the response time. Moreover, chatbots with advanced artificial intelligence techniques can provide precise information to customer queries and problems.

1.2 Conversational AI in Health Care

In the 21st century, since technology has taken leaps and bounds, we do not accept anything around us less than rapid, cost-effective, result-oriented, and accurate. Especially this becomes challenging in the large domain like healthcare as all of us requires equal attention and immediate care as easy as possible. Physicians and healthcare staffs have fewer resources and thus they are not able to manage the needs of health-

care in everyday life. Nevertheless, due to the rise in cloud services and digital health intervention, healthcare chatbots can be designed effectively to support patients need in day-to-day life beyond the on-site diagnosis and consultations.

To deliver the above-mentioned solution, chatbots are growing huge popularity. In the last few years, AI-based conversational agents have exemplified multiple advantages for diagnosis, treatments, and medical support. Researchers have analyzed a variety of healthcare applications implemented through chatbots. From providing guidance about sex, drugs, and alcohol abuse to teenagers [26], self-cure to therapy patients, healthy lifestyle coaching, spreading awareness on smoking and many more [48].

A specific research group discussed mimicking human conversations in the context of health care. The agent can surpass the limitations of human-machine interactions and allow users to communicate freely with chatbots. The researchers have implemented Health On-Line Medical Suggestions (HMOLeS) which can adapt and quickly learn dynamic medical scenarios, thereby making chatbots robust in providing modular healthcare suggestions [16].

Benilda Eleonor [50] has implemented a chatbot called "A Pharmabot" which can aid parents regarding general medicines for their children. The chatbot provided medical solutions with pictorial representation to increase user engagement. The potential of chatbots to connect with users by comprehending their conditions and situations is a crucial part of AI development. K-bot was introduced as AI-based knowledge enabled conversation agent to provide a health routine to asthma patients [31].

Tanioka's review of chatbots indicates that humanoid chatbots are feasible options to provide healthcare if they can understand user emotions, empathize with the user, as well as if standardise the debate on ethical concerns . The newly developed chatbots trained using deep neural networks can generate real humanized responses

because their dialogue management follows a hybrid category of rule-based as well as generative based approach [52].

Mental health chatbots have gained popularity and many fortune companies have identified it as the "future of therapy". A telehealth agent was developed to reduce the risk of hospitalization, support nurses and has achieved some promising results [21].

Thus, conversational agents are used in many applications in healthcare which primarily opens the discussion on technologies and proposals used in this area for research and engineering.

In conclusion, the medical industry is widely adopting chatbots in order to automate healthcare solutions by optimizing cost and resources. Hence, the best practices incorporated for designing and developing chatbots will help the medical domain to achieve the higher objective of providing full potential healthcare services through conversational agents. The high rise in acceptance of AI-based chatbots will increase end-user engagement by boosting the overall market growth. The following factors have significantly contributed to the rise in the growth of the chatbot market.

1. 24/7 availability
2. Decrease in patient waiting times for primary care
3. Medical cost savings and demand for mobile health applications
4. AI-based bots leveraging customer engagement

The only concern which obstructs the growth of healthcare chatbots is patients' data privacy issue which needs to be addressed in a very guided manner. The global chatbot market in 2018 was nearly $ 116.9 million and is expected to increase by $ 345.3 million by 2026, registering a CAGR of 14.5% from 2019 to 2026 [7].

1.3 Organisation of Report

This report is organised in appropriate divisions and sub-divisions so that readers can comprehend this report very easily. The very next Chapter 2 will walk you through the motivation of this research, challenges, and problem statements we are trying to achieve. Chapter 3 of this report is focused on providing background knowledge of chatbot frameworks and cloud services. Chapter 4 and Chapter 5 will discuss some methods to develop dual purpose chatbots using Microsoft Bot Framework cloud services. Chapter 6 will discuss some machine learning and deep learning models used for emotion identification and contextual language generation in health-related chatbots. Finally, we conclude our methods, results with key messages and provide some future work in this domain.

Chapter 2

Motivation, Problem Statement and Challenges

2.1 Motivation

Coronaviruses are large RNA groups that are responsible for respiratory tract infections whose symptoms closely resemble the common cold. In November 2019, the first case of Coronavirus was reported in the city of Wuhan in China. After then, from January to March 2020, the number of infection cases increased throughout the globe and on 11^{th} March 2020, WHO has declared coronavirus a global pandemic. As of now, over 100 million cases of coronavirus is reported across the globe. Global prevention and control of COVID-19 disease posed a serious number of challenges that needed to be addressed immediately [15].

The unprecedented effects of a novel COVID-19 pandemic have caused major impacts on the world economy, medical care facilities and other parts of society. To lessen the effects of disease spread, special quarantine and social distancing measures have been implemented all over the globe. In specific, U.S. Centers for disease control and prevention has guided citizens to reduce unnecessary hospital and healthcare

visits and people can utilize alternatives to medical facilities like telephonic conversations, video calls with doctors etc. Similarly, quarantine measures suggested by WHO made digital health consultations gain immediate popularity all over the world. Despite this, during a global pandemic, there were choked emergencies in hospitals, and they struggled to manage the balance between restricted capacities and high demand. Medical Professionals had made a conscious discussion on how to provide healthcare consultations online via digital platforms. In addition, the major concern during the pandemic was to keep doctors safe and isolated while assisting patients with the best healthcare utilities possible [17].

Therefore, medical professionals and the government are relying on digital healthcare as support to resolve the COVID-19 outbreak. Hence, the novel coronavirus pandemic had significant potential for conversational agents to mitigate the effects on healthcare by providing mental health support, providing online differential diagnosis, appointment bookings etc.

2.2 Problem Statements

A COVID-19 global pandemic has some unique potential characteristics whose solution can be delivered online via chatbots. Some of the problem statements addressed in this project are mentioned below.

1. How can we provide medically precise information through chatbots?

   During the global pandemic, new guidelines were regularly updated from the Centers for Disease Control and Prevention (CDC), WHO regarding COVID-19 infections, emergency rooms, travelling etc. But a visit to a doctors' clinic with mild symptoms can even overwhelm health clinics and can significantly contribute to increasing the number of COVID-19 cases. Hence, it was necessary to prevent misinformation, because, during the Zika virus spread in 2016, a lot

of misinformation was reported through social media websites like Facebook, Twitter [38]. Since chatbots can deliver a single answer to user queries, it can be easy to guide patients by providing single information from credible sources.

2. How can we collect patient health data through chatbots and narrow down the scope of clinical diagnosis?

   Due to emergency visits, hospitals were overwhelmed by the number of COVID-19 infected patients regularly. Chatbots can be used to mimic the onsite diagnosis and collect patient health information to narrow down the scope of clinical diagnosis.

3. How can we identify user emotions in healthcare chatbots?

   Due to the sudden effects of the COVID-19 pandemic, the mental health issues are still not addressed correctly. Frontline health workers and nurses are not trained to provide immediate mental health support. Hence, this issue required immediate attention. Emotional support online through chatbots can reduce psychological harm due to isolation and loneliness. As per statistics of the SARC pandemic in 2003, WHO reported that more than 30% of patients were affected by mental health and psychological issues such as anxiety, depression, loneliness and many other psychiatric symptoms. It took more than 30 months for each of them to completely recover from the mental health effects of the outbreak [35]. Primary evidence indicates that chatbots can be designed effectively to abate the psychological effects on the human brain.

2.3 Solution Overview

In summary, we propose a chatbot framework using advanced cloud services and deep learning models to provide digital healthcare solutions. We worked on building

a chatbot that is focused on the three problem statements mentioned earlier.

1. How can we provide medically precise information through chatbots?
2. How can we collect patient health data through chatbots and narrow down the scope of clinical diagnosis?
3. How can we identify user emotions in healthcare chatbots?

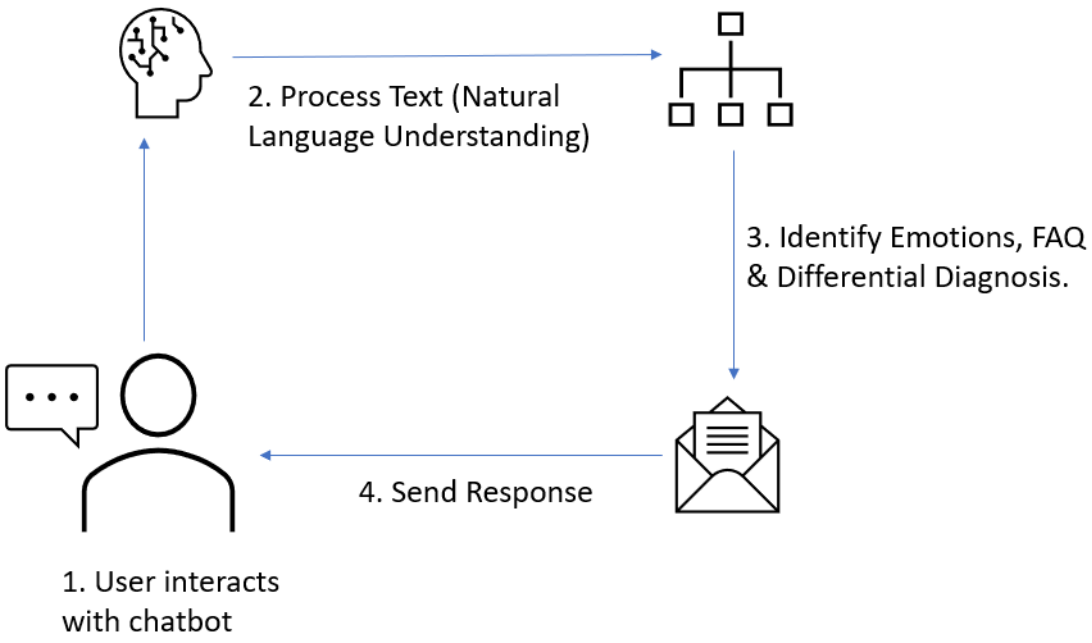

Figure 2.1: Solution Overview

The above Figure 2.1 shows the working of a proposed chatbot framework. The chatbot framework can answer user queries related to the novel coronavirus pandemic. In addition, a chatbot is programmed to provide differential health diagnoses beyond onsite consultations. Further, a chatbot model can also be trained to identify user emotions and to generate emotionally intelligent responses back to the user.

2.4 Challenges

Results from the literature review suggested that the usage of healthcare bots can pose many challenges at social and technical levels. At the social level, medical bots can become inactive due to a lack of usage from the community. There is a huge gap between how patients perceived these technologies compared to robust health care solutions delivered through chatbots. Acceptance of chatbots is hugely affected in medical care because of users' negative notions about chatbot integrity, health data privacy, and ability to provide precise information.

From the technical perspective, since plenty of information is updated on daily basis, it is extremely challenging for chatbots to provide accurate information. Information retrieved from multiple sources may not be precise and coherent and it can leave users in a confused state of mind. Moreover, most of the chatbots deployed currently are not robust to assist users with sensitive topics like mental health and psychological issues. For example, the use of emotional context in natural language generation is not yet considered a sophisticated solution, since the reliability of medical chatbots in handling nervous breakdown or suicidal tendencies is not yet explored to its full potential. Since these mental health issues are more conspicuous during pandemic, the inability of chatbots to provide empathetic responses may leave the patients with a high risk of suicide or unnecessary worries. Also, the lack of emotional data available significantly reduces the number of conversational models deployed for emotion detection in chatbots. This novel pandemic can lead end users to ask chatbots about different scenarios which can cause misunderstanding and can misguide users to take inappropriate actions.

2.5 Summary

In this chapter, we discussed mainly about motivation, problem statements and challenges of this project. In the next chapter, we will introduce the basics of chatbot frameworks and cloud services.

Chapter 3

Chatbot Frameworks and Cloud Services

3.1 Introduction

There is an enormous rise in digitalization and eCommerce platforms in recent past years. In a recent survey, it was derived that 53% of buyers will look for buying products from a company that can provide services via messaging platforms [5]. In this scenario, many business firms are looking to provide product recommendations and customer support online to enhance customer engagement and experience in a much more effective way. Hence, businesses are adopting chatbots to understand conversations in a contextual way, to personalized services to every customer so that they feel connected, and they can engage customers in a humanizing manner.

But due to a lack of technical skills, infrastructure and domain expertise, many businesses face difficulties in building and deploying Natural Language Understanding (NLU) based chatbots which can hinder their business and economic growth. So, to get quickly started many technical giant companies like Google, Amazon, Microsoft Azure provide ready to build chatbot frameworks with advanced NLU and Machine

Learning capabilities. The below section provides the basic introduction to chatbot frameworks and their components.

3.2 Basics of Chatbot Frameworks

Chatbot frameworks are easy to use built-in templates provided by cloud service providers where you can use predefined classes and functions to create and deploy conversational agents using NLU and machine learning capabilities. The basic component of the chatbot frameworks is described below [19].

3.2.1 Natural Language Understanding

NLU is a method by which a machine can understand natural language text in chatbots. It is a process of transforming natural language into a machine-readable language.

Lexical Analysis: It is process involving breaking of sentences into tokens of words and characters. Commonly, this process is known as tokenization.

Syntactic Analysis: The language structure of the information string alludes to the sequence of words in a sentence, so they linguistically bode well. NLP utilizes syntactic investigation to assess whether the regular language lines up with linguistic or other consistent principles. With this, we can easily extract meaningful sentences explicitly.

Lemmatization/Stemming: It diminishes the structural variation in the sentences and simplifies it. Lemmatization is extremely useful in the case of chatbots to comprehend and understand user queries. It conducts a morphological analysis of words to better understand the context of each word. Stemming again reduces its words to their root. It considers only the prefix and suffix of a given word. For example, the word 'studying' is reduced to 'study' after study. This process is useful

in many scenarios despite its limitations that the stem of many words is similar and indicates similar meaning which is problematic sometimes.

Word Segmentation: It is a common way of segmentation of different words especially in the English language because words are separated by spaces, and it is easy to group similar words into one segment.

Parsing: It is a process to analyze grammatical errors in a sentence and the most common way is known as Dependency Parsing. Dependency Parsing analyzes grammatical structures based on dependency among the words.

Semantic Analysis: It verifies whether the text is either meaningful or not and it explicitly extracts the meaningful mappings from syntactic constructions. It will reject the sentences like 'fire in water' which has no meaning at all. This method is useful while translating text from one form into another form. Also, it extracts meaningful phrases and categories from the sentence like time, place, etc.

3.2.2 Natural Language Generation

It is a process of contextual text formation to create a comprehensive response. This is mainly used in chatbots. In its simple form, natural language generation is responsible for the arrangement of phonetically right sentences and expressions. The main challenge in this task is to comprehend the complicated formations of natural language. The construction of language is itself exceptionally ambiguous regarding punctuation, lexis, and different parts of discourse like comparisons and illustrations. A solitary word can be taken as a thing or an action word, a solitary sentence can be passed in numerous ways, additionally, solitary information might have various implications, and so forth.

3.2.3 Automatic Speech Recognition

This terminology falls under computational linguistics, which is responsible to build advanced technologies and methods that are useful for the translation of human speech into text with the assistance of computers and mobile devices. It is often interchangeably used with various terms like Speech to Text translation, Computer Speech Recognition (CSR), Automatic Speech Recognition (ASR). From the perspective of innovation, the historical backdrop of speech recognition is exceptionally tremendous, having surprising advancements. As of late, this field has advanced a great deal by the progression in enormous information and profound learning. The progression not just demonstrated by the distributed scholarly papers yet in addition by the assortment of itemized learning strategies taken on by industry in planning and sending of discourse acknowledgement frameworks. It includes research and development in computational linguistics and related computer science domain.

3.2.4 Basic Working of a Chatbot

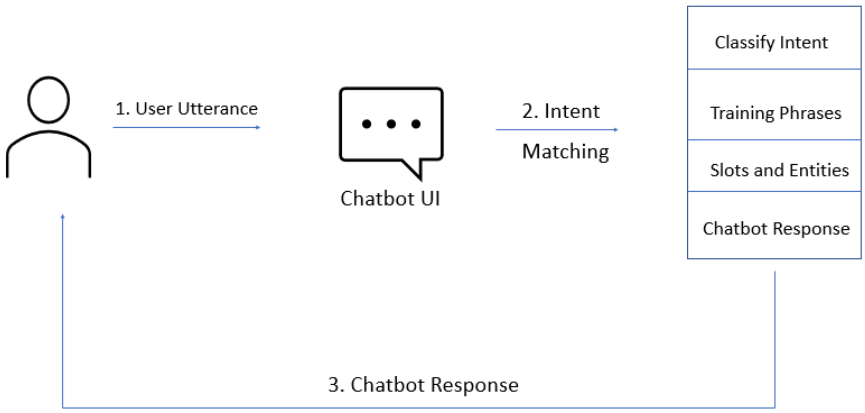

Figure 3.1: Basic Working of Chatbot

The user types in a message through the client interface. The chatbot receives this user utterance in a form of natural language and converts this utterance into the machine-readable form by using NLP techniques. From the user utterances, In-

tents are matched and classified based on the phrases used during training of the chatbot model. The chatbot returns the response to serve the user request based on responses used during training for the respective matched intent. The pre-defined action is returned from the bot based on the intent matched with the highest confidence interval. The bot uses the confidence interval technique to match the best intent with user utterance.

- Intents

  Intents are the most basic way of understanding what the user wants the bot to do. It is an action user wants to perform, or the information one wants to retrieve. For example, if a user types in or says, "I want to book a flight", then the intent of the user is classified as "booking a flight".

- Entities

  Entities are the parameters required by the bot to fulfil the end-user requests. For example, if the user types in or says "I want to book a flight to California for tomorrow". Here the parameters: California and tomorrow are entities that are required to complete the user requests.

- Slots

  Slots are the values you require to collect in order to satisfy the user request. For example, if a user types in "I want to book an appointment", bot response will be "what is the preferred date and time?"

- Training phases

  Training phrases are used to train the chatbot model to understand the user text accurately. During the training of the model, the user will provide multiple training phrases with the possible key-value pairs where a single key is a possible text from the end-user, and value is the response of the bot for the corresponding

text. The bot is trained on many possible examples so that chatbot can assist many different forms of user queries.

- Conversation Flow

  A flow is a sequence of dialogues defined in the database to execute the custom business logic. The flow of dialogues is explicitly programmed to serve the primary purpose of the bot. Errors and exceptions are handled at run time by defining dynamic responses if the flow of the conversation is compromised.

- Multi-turn Conversations

  The multi-turn conversation is the way of chatbots to get involved in a conversation with the end-user. Multi-turn conversations allow one by one conversation of chatbots with the user. For example, a chatbot collecting information of booking flights after identifying the user intent asks users a series of questions one by one to complete user requests.

- Fulfillment

  Fulfilment allows you to generate dynamic responses based on the intent classified by the model. You can easily integrate services with specific intents. For example, if the end-user wants to schedule an appointment, you might want to check in the database if there are free slots available or not.

3.3 Cloud Services

This section will briefly talk about chatbot services provided by top cloud service providers like Google, Amazon, and Microsoft.

3.3.1 Google Dialog Flow

Google Dialog Flow chatbot framework comes with built-in AI and NLP capabilities by which you can create chatbots in multiple languages, and you can easily deploy them on multiple platforms. The main benefit of using Google Dialog Flow is it provides client libraries in multiple platforms and easy to use interfaces for creating chatbots for cars, speakers, and many other use-cases. It provides lifelike conversational AI with state-of-the-art virtual agents [6].

Available in two editions: Dialogflow CS (advanced), Dialogflow ES (standard).

- Dialog Flow ES
- Dialog Flow CS

Dialogflow CS is the new technique to develop chatbot agents, applying a state machine approach for the agent design. This will provide complete control over a conversation, an effective user experience and robust development of the workflow.

3.3.2 Amazon Lex Services

Amazon Lex is a generalised framework service for creating and deploying conversational agents into many android applications, web-based frameworks using advanced voice and text capabilities. It provides many advanced deep learning framework capabilities of ASR for conversion of speech into text, NLU to recognize the intent of the user utterance, also provide services to enable you to build applications with highly engaging user experiences and lifelike dialogue management. Using amazon lex services, we can use the same deep learning functionalities that power Amazon Alexa and is now available to any developer, providing us with capabilities of building sophisticated, natural language and easy deployable conversational bots [4].

Amazon lex chatbots can be scaled automatically, so enterprise developers do not need to be worried about infrastructure engineering. It is integrated with AWS

Lambda [4], that you can implement to trigger complex functionalities for executing your backend business logic for data analysis and updates. Once our chatbot is deployed on the cloud, we can directly publish it to chat platforms, mobile applications and IoT services. We can also analyze our chatbot consumption by the outside world by tracking built-in metrics for our chatbot.

Amazon Comprehend is a NLP-based web service that facilitates developers to extract built-in entities and relationships in the text with a single API call. Amazon Comprehend Medical service uses machine learning in the backend to identify health-related information from the text. You can easily extract information like diseases, medicines, symptoms, and dosages without training a single machine learning model [1].

Thus, AWS lex service provides us with easy-to-use, secure mode of building, and deploying chatbots with advanced analytics and monitoring.

3.3.3 Azure Bot Services

With Microsoft Bot Framework, software developers can develop enterprise-grade conversational AI experiences for users while maintaining complete control of data storage and accessibility. Developers can easily build multilingual to multimodal bots for any possible use cases like customer support, sales, and employee productivity [3].

Enterprise developers can incorporate their chatbots with high-quality AI capabilities with Azure cognitive services. You can create a bot to interact with natural language and a recognizable unified service like your brand. With the bot framework, we can publish our bot to as many as thirteen channels and hence we can reach as many as customers possible. We can publish our bot to Interactive Voice Response (IVR) and digital platforms, including web applications, mobile apps, Microsoft Teams, Facebook Messenger, and Slack.

Microsoft provides health chatbot frameworks that enable developers in healthcare

firms to build and deploy intelligent and high AI-powered conversational agents along with built-in health intelligence tools and extensible tools to enhance the process, mitigate the costs by learning patient outcomes [3].

Azure Text Analytics is a NLP-based service to extract entities from the text. Text Analytics medical API service can be used by developers to identify health-related keywords from the unstructured text within the medical documents [2].During the COVID-19 outbreak, many customers were using Providence–an Azure healthcare framework to manage hospital resources. The Azure COVID-19 templates were used by thousands of medical firms, the US Centers for Disease and Prevention (CDC), and many other global healthcare authorities. CDC used a chatbot called 'Clara' to provide online COVID-19 assessments and to answer COVID-19 related FAQs. Providence templates were serving to thousands of patients locally within a few months [9].

Since Azure bot templates were extensively used to deliver COVID-19 health solutions, we were adopting some of the individual components from the Microsoft bot framework to build chatbot for our use cases.

3.4 Summary

This chapter briefly discussed the basics of chatbot frameworks and cloud services. The next chapter will discuss the methods we have used in building QnA chatbots.

Chapter 4

QnA Chatbots

4.1 Introduction

The fast-evolving science about novel COVID-19 disease and virus has created unparalleled medical information needs and noticeable policy changes across the globe. Precise and well-timed health information is globally required across governments and stakeholders. Many public health organizations have been asked to deliver precise information about novel coronavirus, symptoms, pandemic instructions, travel restrictions, etc. In-campus reduction in staff has weakened the process of delivering accurate information throughout the world. The dominance of pandemic and COVID-19 effects has made organisations acquire new roles in order to disseminate the public health information [36].

Since there is enormous demand for information on COVID-19, many health care organisations have leveraged the potential of conversational agents using cloud services to automate the response of frequently asked questions. We have tested a proof of concept using cloud technologies to build QnA chatbots. Our main contribution to this chapter lies in discussing various novel methods to leverage the power of Microsoft bot framework cloud services to build, deploy and test easy to use, time and

cost-effective, scalable, AI-enabled smart QnA chatbots. QnA chatbot can update its knowledge base frequently and retrain itself to be in sync with the rapidly evolving science about novel coronavirus. Since the knowledge base is trained from multiple trusted sources, it is trustworthy and we can provide timely responses to the users. Thus, we have exemplified an end to end solution that automates the COVID-19 FAQ response with chatbots using Azure QnA bot architecture. Our methods indicate that AI-enabled intelligent healthcare solutions can be quickly deployed to disperse accurate information.

The beginning of the next section will discuss QnA chatbot architecture and later we discuss steps to build a QnA chatbot on the Azure cloud platform.

4.2 QnA Maker Architecture

Figure 4.1 shows various components used for building QnA chatbots. Azure web application will run inside the Azure app instance. Azure web app will make an API call to QnA maker service to fetch the appropriate answer related to user queries. User chatlogs can be saved in Azure Cosmos DB.

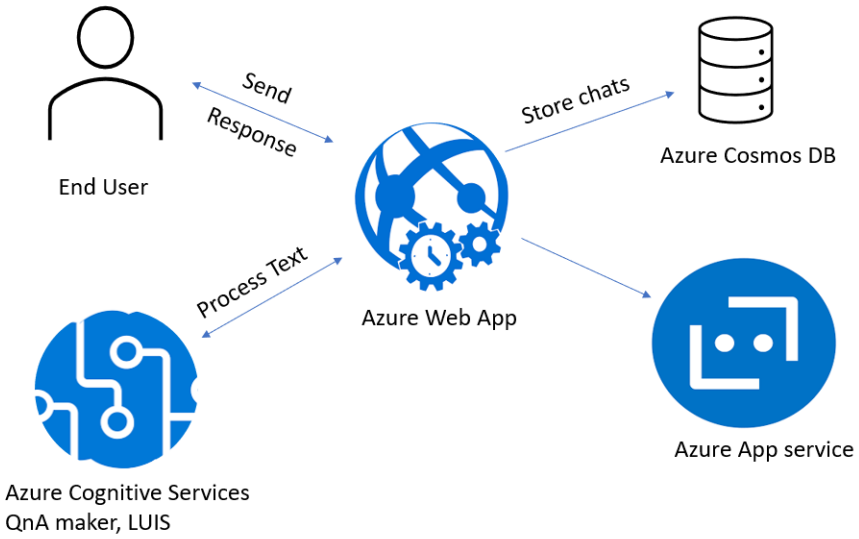

Figure 4.1: QnA Maker Architecture

4.2.1 Azure Cognitive Services

Azure Cognitive Services brings built-in AI services without the need for any machine learning expertise. It just takes an API call to enable the ability to search, comprehend, speak and empower the decision-making process with ease into applications.

1. QnA Maker

   Azure QnA maker service is a cloud-based NLP framework that enables you to develop a natural conversational layer over your data. It can be applied to find the most suitable answer for any inputs from the customized knowledge sources. The QnA bots are primarily designed to respond to any of the users' questions. For example, the user might ask a bot: "what are symptoms of the COVID-19 disease?". The QnA maker service is primarily applied to develop conversational client applications.

2. Language Understanding Intelligent Service (LUIS)

   LUIS is an AI-based cloud service that can be easily trained to apply customized machine learning to the user utterances to identify the overall contextual meaning and extract relevant, detailed information. Microsoft natural language service will disburden oneself from in-house domain expertise related to AI or deep machine learning algorithms. In a few moments, we can easily build and use our own AI-enabled conversational agents. The service offers enterprise-grade security, compliance, and data privacy. It provides encryption of our data in storage. LUIS can be easily integrated with other Azure cloud services like bot framework, and QnA maker.

   QnA maker and LUIS API services are designed for a completely different purpose. The language understanding model determines and classify the intent of the end users' utterance while the QnA maker identifies the best possible response based on the user's query from the knowledge base. To identify the

correct set of API services to call, we had comprehend the user text coming from the client interface, and the type of information the application requires to get from the Azure cognitive services.

4.2.2 Azure Cosmos DB

We have stored the user chat logs into the Azure Cosmos DB database. To maintain low latency and high availability of our application, these instances need to be deployed in cloud data centres that are nearest to our end-users. These applications need to be highly responsive in near real-time handling dynamic requests at peak hours of usage, storing data and making the data available to end-users in microseconds. Azure Cosmos DB is a fully automated managed NoSQL database designed for exactly two above mentioned purposes. It can be scaled automatically resulting in single-digit millisecond response time and provides instant dynamic scalability. Azure Cosmos DB automates the process of data management and administration process with frequent updates and patching. It has built-in capabilities to handle capacity management along with optimized cost, serverless, and auto-scaling choices that are highly responsive and can match its performance efficiency with the demand.

4.2.3 Azure Web App

Azure web applications built using multiple platforms like bot composer, bot framework SDK, and powerful virtual agents can be easily deployed using Azure app service plan.

4.2.4 Azure App service

Azure App Service is an HTTP-based service for managing REST API's, web applications, and servers. It provides the flexibility in writing codes in any programming

language from C, Java, Python, and PHP. Azure web app service is easy to scale on windows based as well as Linux based environments.

4.3 Steps to Deploy QnA Maker on Cloud Instance

This section explains the step by step procedure of deploying QnA chatbots on Azure cloud instances.

4.3.1 Resource Manager on Cloud Instance

Azure resource group service is a virtual collection of all the services provided by the Azure cloud environment. It is a collection of virtual machines, database servers, web app services along with continuous integration, deployment and other storage accounts etc. Before initiating a QnA maker service in our Azure environment, we have launched an Azure resource group in the Azure cloud environment. Once we have launched Azure resource group, we initiated QnA maker service. We have used the free Azure default subscription for launching a resource group in the Azure environment. Figure 4.2 shows the Azure UI with resource instances deployed to run our instances.

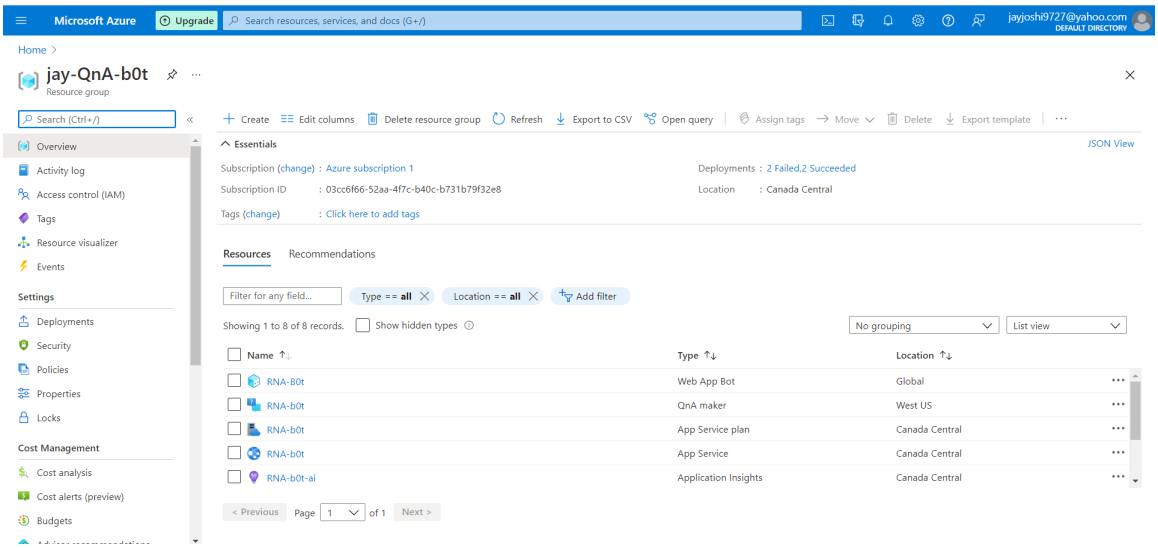

Figure 4.2: Resource Instance

4.3.2 QnA Maker Service

After creating the resource manager, we have registered for the QnA maker service to establish a QnA maker bot interface in the Azure bot application. QnA maker free tier is available which allows free 100 transactions per minute and 50K transactions per month which is sufficient to test our QnA bot. Once we have registered with the QnA maker service, we have accessed the QnA maker information from the Azure web interface. Now we have access to different information such as API keys, endpoints, resource group, metrics dashboard for monitoring, billing, and other additional information of QnA maker service can be easily managed from Azure portal. Figure 4.3 exemplifies QnA maker service deployed in the Azure cloud instance.

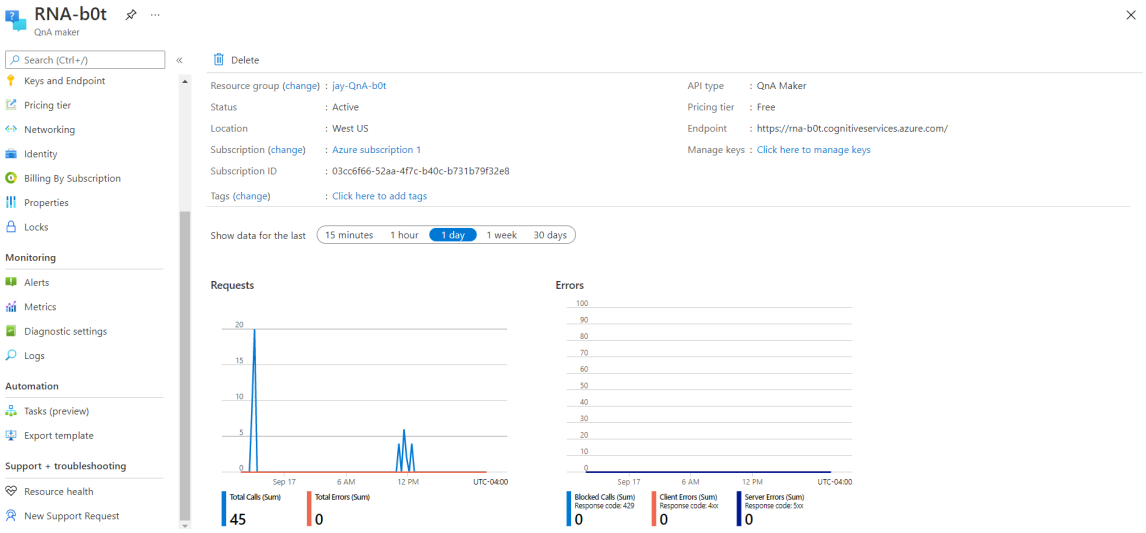

Figure 4.3: QnA Maker Service

We have analyzed metrics related to our resources on the Azure instances. For analyzing usage of QnA chatbot, we can easily view dashboard metrics like the number of calls to API, the number of failures, how much data was used per day, per hour, per month. Figure 4.4 shows the example of a number of API calls at each time daily for QnA maker service.

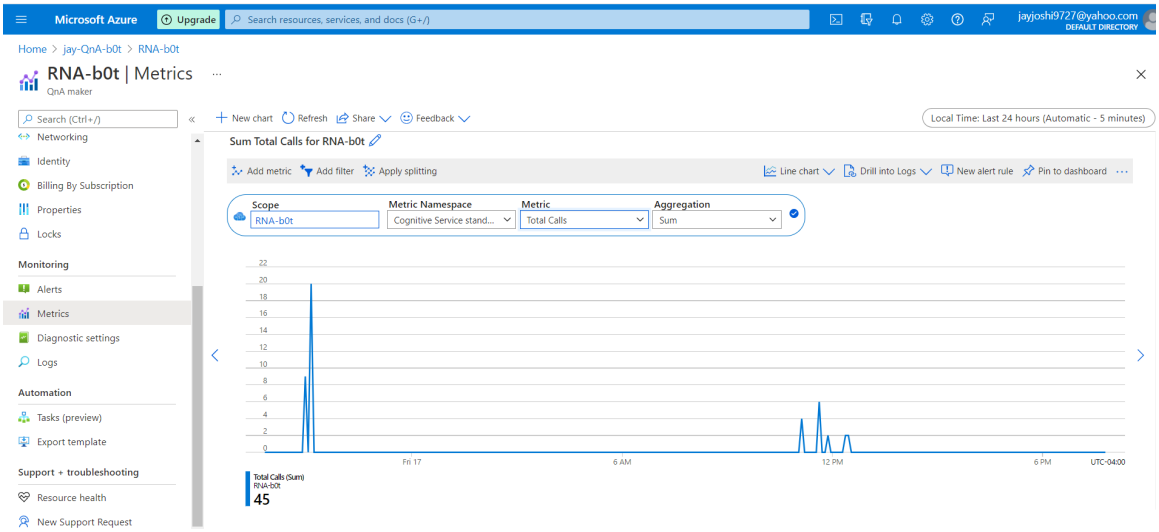

Figure 4.4: QnA Maker Metrics

4.3.3 Training Knowledge Base

For building a chatbot to provide accurate information, we have trained knowledge base using Azure knowledge base and LUIS services. Since the main aim of building the QnA maker was to provide the accurate information we have included information from multiple sources as shown in Figure 4.5. So, globally people can get the best information on COVID-19 help centres, food resources, testing, quarantine, travel, and many more. Overall, more than 550 QnA pairs were extracted from the trusted sources for training our QnA chatbot.

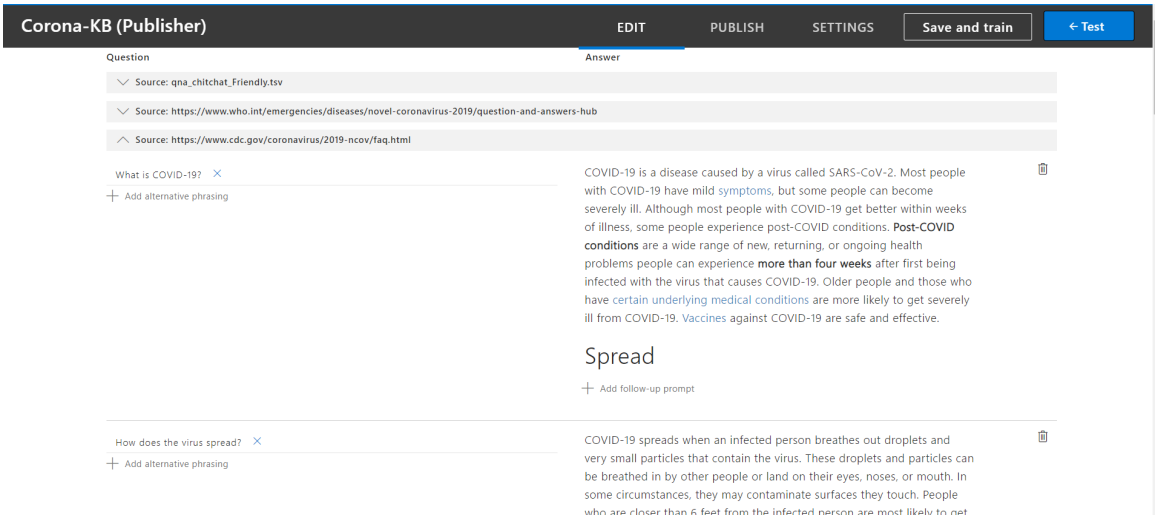

Figure 4.5: Knowledge Base

Once the QnA pairs are extracted by the Azure knowledge base, we have added alternative phrases for some common and most frequently asked QnA pairs so that the response from the QnA chatbot exactly matches the user query as shown in Figure 4.6.

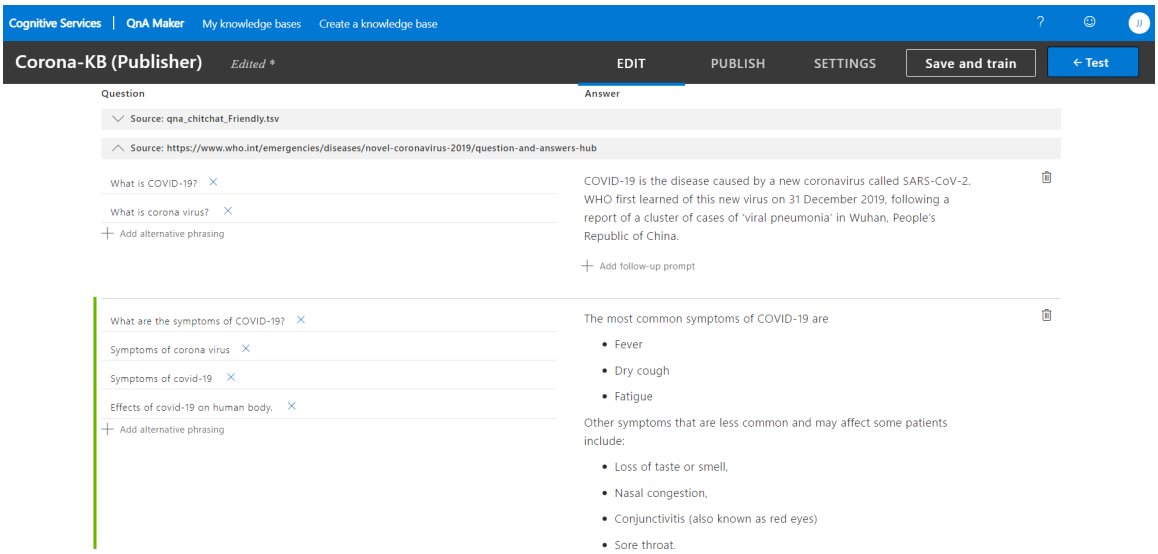

Figure 4.6: Alternative Phrases in QnA maker

For example, as shown in Figure 4.6, the alternative phrases are added to the most common question "What is COVID-19?" The added phrase is "what is corona virus?". Also, there are many additional phrases added to the question "What are symptoms of corona virus?" like "Symptoms of corona virus", and "Effects of COVID-19 on the human body". This method will make sure that people asking for information in multiple ways will get the most accurate responses from the knowledge base.

We have also added personality to our QnA chatbot by training a separate knowledge base for the most usual questions. This allows our chatbot to express boredom, greet, laugh or giggle at times with some unified answers to those usual set of questions. Figure 4.7 indicates how we can add personalities to our chatbot using Azure bot service.

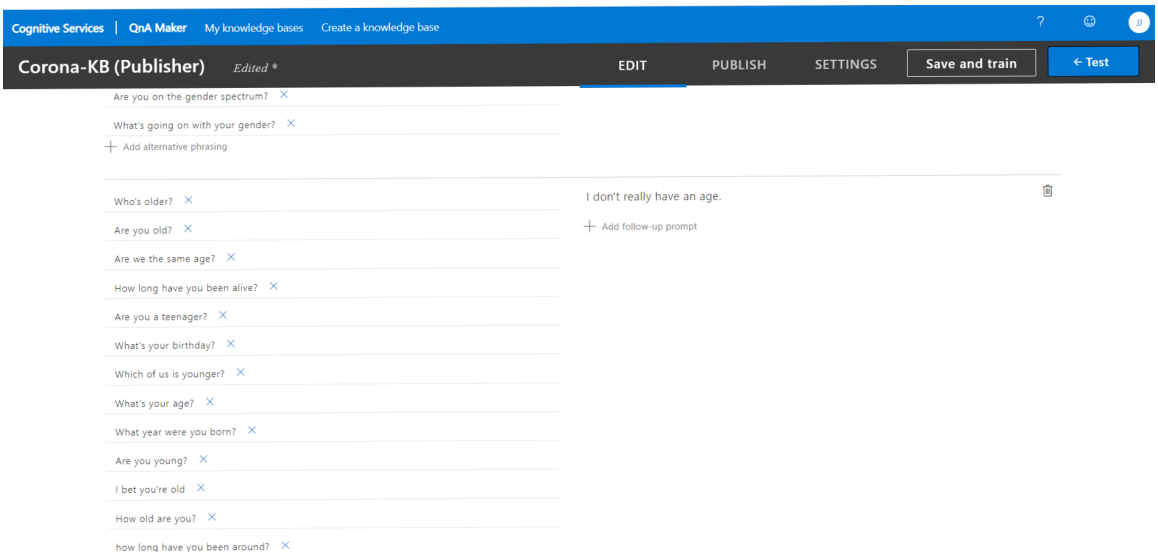

Figure 4.7: Personality in Chatbots

After adding the training phrases for the QnA pairs, we have saved and trained our QnA chatbot model with a simple click on the save and train button. The training of the bot usually takes up to five minutes and once the bot QnA pairs are trained we have quickly tested our knowledge base before publishing our knowledge base to Azure cloud instance. Figure 4.8 indicates the testing of the knowledge base in the webchat interface. You can type in the question and can inspect the response from the trained model. For every response, the bot replies with the confidence score labelled with it. The confidence score indicates that how much the bot is confident of its response to the users' query. In the below example, you can easily analyze the confidence score is returned for the bot's response to the question "What is coronavirus"? The value of the confidence score is 100.

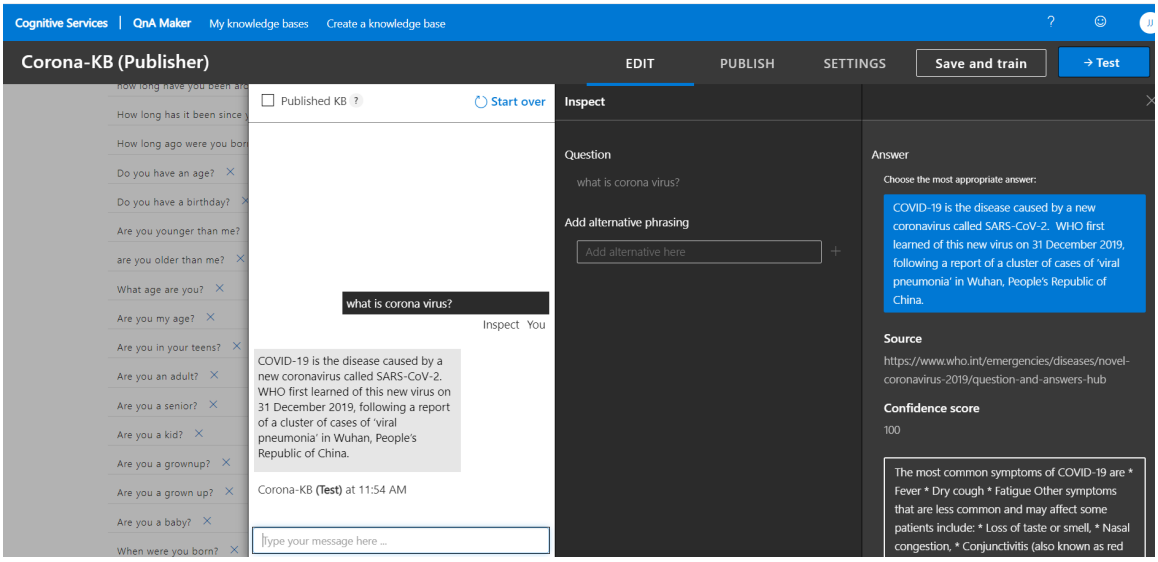

Figure 4.8: Training Knowledge Base-1

Now if the confidence score does not match with some common threshold value, we have added the additional phrases for those questions from the webchat interface and we have retrained our model. For example, in Figure 4.9, "do I need to wear a mask at home?" returns the response with a confidence score of 81.05. In this scenario, if the response matches the user queries, we have retrained the bot by adding the additional phrase "do I need to wear a mask at home?"

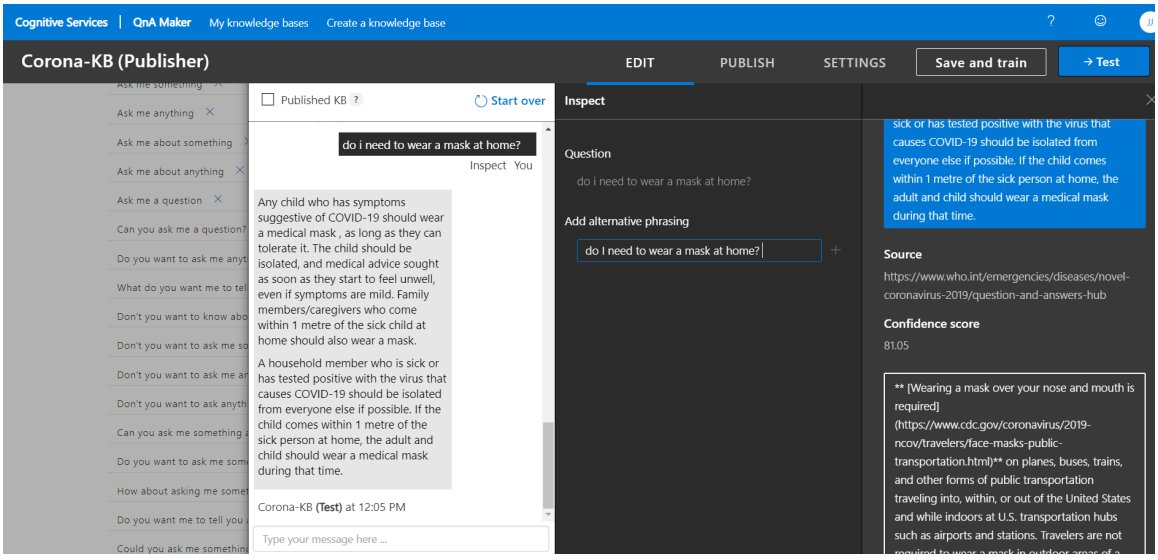

Figure 4.9: Training Knowledge Base-2

Now, once our knowledge base was tested in the web chat completely, we have published our knowledge base with a simple click on publish button. We will get the JSON response of the URL which we have used for publishing our QnA maker service. It publishes the knowledge base to the specified QnA maker established in the Azure cloud instance.

4.3.4 Continuous Integration and Deployment

For CI/CD, we have easily configured our Github repository with an App service instance. App service is like a compute resource that will run QnA maker API for the web applications.

By configuring our Git repository with the Web application service, it has generated a Github YAML file which will define a workflow to deploy our code changes in the master branch to the App service repository. So, any future changes to the Github repository will make the changes to our web app service automatically. Every time we do not need to update those code changes manually. The YAML file generated after adding the Git CI/CD configuration is shown in Figure 4.10 below.

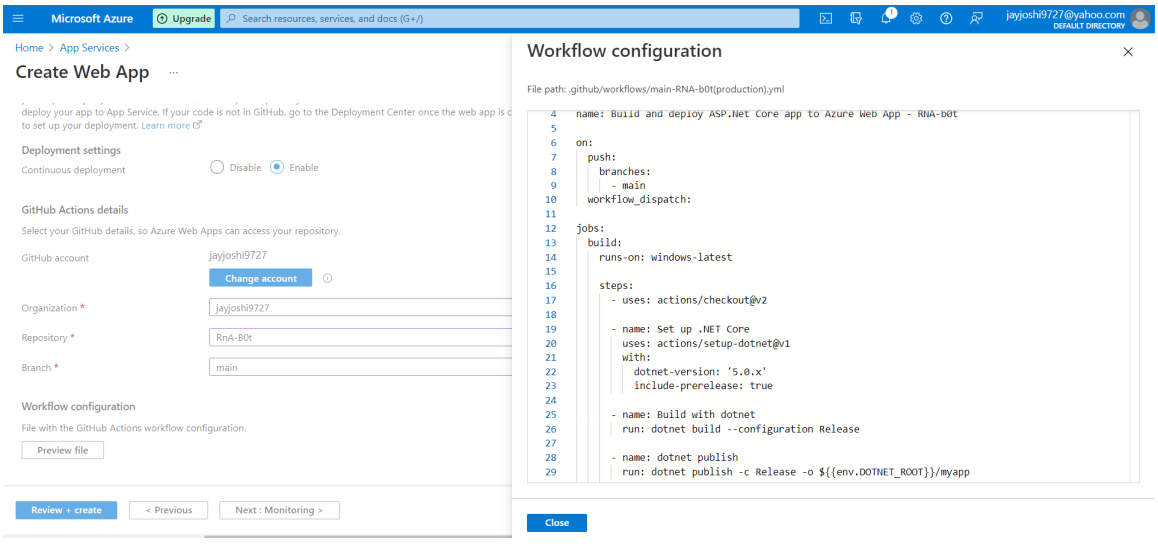

Figure 4.10: Git CI/CD Pipeline

4.3.5 Testing Chatbot in WebChat

After following the above-mentioned process, we have successfully tested our chatbot by querying the knowledge base with a different set of alternative phrases. We have tested our chatbot in the web chat interface in the Azure cloud environment. Azure web chat interfaces is a fully customizable client interface to connect with chatbots. Figure 4.11 and 4.12 shows the web chat interface for testing QnA chatbots.

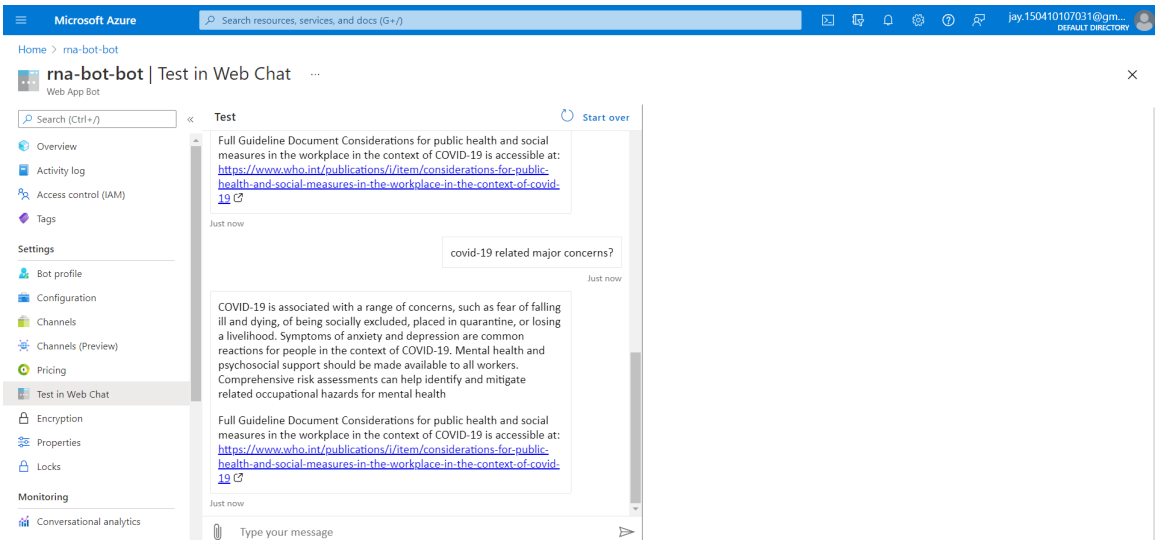

Figure 4.11: Testing QnA Chatbots-1

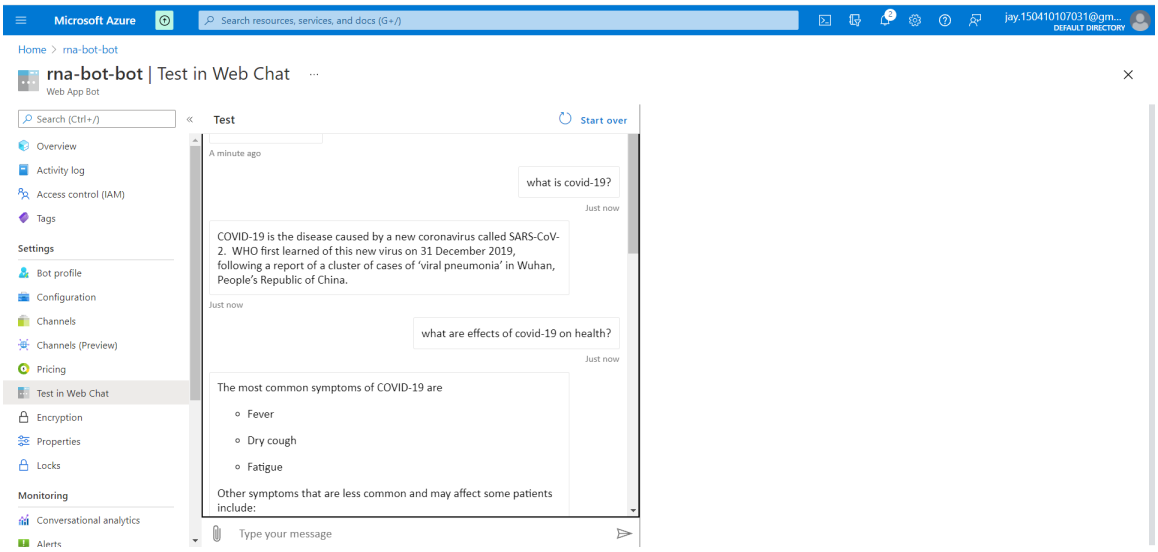

Figure 4.12: Testing QnA chatbots-2

4.4 Summary

Our main aim with this chapter was to explore, how we could quickly deploy chatbots using advanced cloud services to answer user queries related to COVID-19. Thus, we have briefly discussed methods to build AI-enabled smart QnA chatbots using components of the azure bot service. The next chapter will discuss methods for developing chatbots to collect patient health information.

Chapter 5

Pre-consultation Health Services Through Chatbots

5.1 Introduction

During the COVID-19 pandemic, hospitals and clinics are overcrowded due to a rapid increase in the number of infections daily. Most hospitals and clinics do not have the required staff and resources by which they can provide healthcare to patients infected with the novel coronavirus as well as disinfectants with other serious medical illnesses. During these stressful times, a pre-consultation and differential diagnosis chatbot can save a lot of time for doctors in the clinic and it is effective especially when clinics and hospitals are overwhelmed by COVID-19 infected patients. It can improve work time efficiency as well as reduce the burden of clinical checkups for physicians.

Also, it is believed that the high cost of our medical facilities can often be subjected to the lack of patient interaction once they left the clinical trials. Chatbots can provide healthcare treatments at low costs and even improve treatment if the doctor and the patient keep in touch before and after the patients' medical checkup.

Pre-consultation chatbots mostly are designed to collect patients' medical history

and symptoms before the patient is appointed to meet the doctor. In an ideal scenario, doctors' and nurses do not take much time in collecting symptoms and medical history before providing treatment to patients. So, pre-consultation chatbots are developed as quick decision flow to collect patients' health data to realize the optimization of diagnosis and treatment.

When a patient presents themselves to the physician, they "present" symptoms. These symptoms or "presentations" form the basis of the clinical inquiry. However, symptoms tell only part of the story of the clinical encounter. The other essential elements of a clinical encounter include:

1. Presentation - symptoms that present to the physician
2. Medical History - what ailments have they had in the past (injuries, disease, etc.)
3. Family History - what ailes family members, past and present
4. Medication - what drugs are they currently taking and for what
5. Differential Diagnosis - what may be some clues to narrow down the scope of the clinical inquiry

Hence, the role of the chatbot will be to collect the information above using machine learning where appropriate and summarise it for the physician prior to the consultation.

To narrow down the scope of clinical diagnosis, we have discussed a novel "Emotibot" architecture in collaboration with domain experts at MD Canada Wellness Solutions to maximize the amount of health information collected in an optimized duration of time. We have closely worked with the team of doctors at MD Canada to design the generalized decision flow to optimize the information collected during COVID-19

pandemic. The advanced flow of medical questionnaires can also assist doctors in collecting emotional data from the user.

Some famous pre-consultation chatbots are listed below along with their functionalities.

1. Left Hand Doctor [57]

    - Auto collection of past medical record: It can assist patients to provide their own medical conditions and save disease information record in a guided manner. This can improve user participation, help people know their medical conditions, and prepare themselves for the next patent-doctor in-person visit.
    - It assists users to know patient conditions in advance: The information collected is matched against hospitals electronic record to provide primary medical history of patient to doctors. Hence, when doctors ask patients for actual diagnosis, they need to collect little information from the user. It can save a lot of time for physicians and can increase clinical efficiency of patients' diagnosis. The users' responses will be generated and saved as a standard medical record which requires little modification to complete the medical notes.
    - How successful it is: Since it is released, it is open for the left-hand doctors and has covered more than 6000 common diseases into 35 different departments and it mainly has five main functions not limited to intelligent self-diagnosis, guidance, pre-diagnosis, QA's, etc. It has served over 350 industry customers, with actual diagnosis of nearly one million people everyday.

2. Lantone [57]

- Guided Questions and Answers: If you are uncomfortable and want to have a medical checkup, you can use this chatbot system first and it will provide you basic guidance to finish some questions about your symptoms. Your symptoms are logged and send to the doctor before you go for a visit to the hospital.
- Provide existing medical test and results. You can easily upload some diagnostic medical results to make sure doctor knows your situation better.
- Review Patients diseases: Patient can view their own medical results after seeing a doctor and once the results are saved and recorded in the database.
- Success: It is applied to more than 150 communities of health care, to the five best hospitals of China, more than 300 private clinic and 3000 community centers in China.

5.2 Designing Flow of Medical Questionnaires

The main purpose of designing "Emoti-bot" was to propose a generalised decision flow of medical questionnaires to collect patient health data and narrow the scope of clinical diagnosis during a COVID-19 pandemic. Whenever a patient arrives at the clinic, doctors and nurses mainly aim to collect two different types of information i.e., current concerns and past health records. Doctors and nurses are trained to ask a specific set of questions to extrapolate maximum health information from the patients. In an ideal scenario, this procedure does not take more than five minutes. "Our Emoti-bot" decision flow exactly aims to replicate the above clinical process and aims to collect maximum health information from the patients. We have designed a hierarchical decision flow and divided various medical questions into different question sets each belonging to a different category. The sequence of question sets is designed in a flow based on the type and category of medical information required prior to

in-person visit. Patients medical record is main branch of the decision flow. At the highest level, patients' health record can be divided into two different categories.

1. Current Reasons for Consultation

2. Past Medical Record

The above-mentioned question sets aim to collect health information from the patient at two different levels: present and past.

Figure 5.1 shows the generalised decision flow to mimic the onsite consultation.

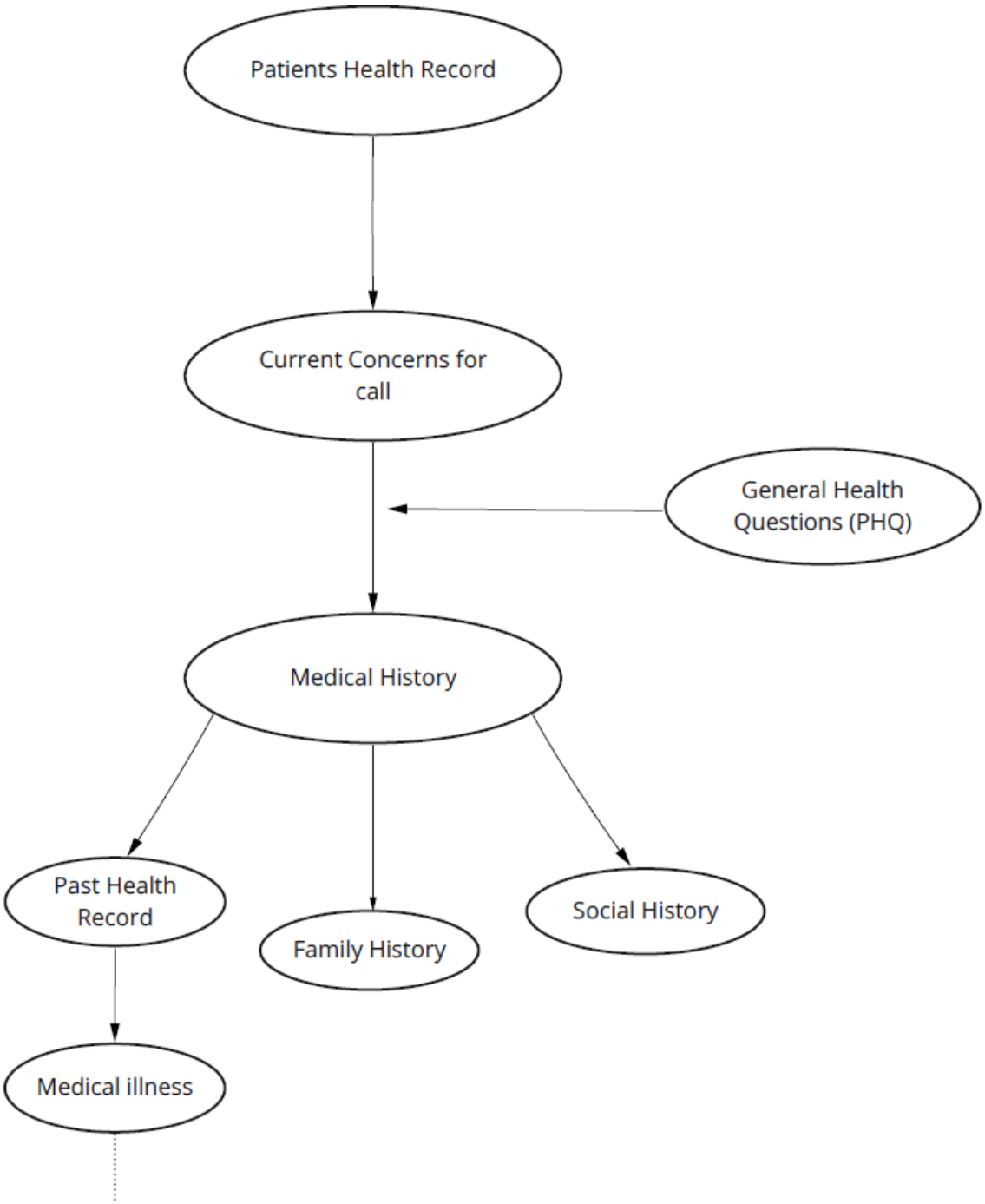

Figure 5.1: Emotion Question Set

According to domain experts, 80% of the time patients are not willing to discuss their past health records. But to collect the health record, the past medical record is divided into multiple categories, each category belonging to one question set and each question set has some very specific set of questions. This gives flexibility to patients

and they can limit the amount of health information provided to chatbot yet at the same time we can optimize the type and amount of information collected through the chatbots. Hence, the past medical record is divided into three specific categories.

1. Past Health Record
2. Family History
3. Social History

Based on information collected from domain experts at MD Canada, user emotions variably depend upon their health symptoms and conditions. Hence, some unique set of question sets can be incorporated into the above decision flow in Figure 5.1 to identify end-user sentiments. But more often users have trust issues with chatbots and they do not provide medically precise information. So, it was challenging to add additional question sets in the above decision flow which can provide patient emotional data before the actual clinical trial.

The sentiments of end-user are more dependent on the history of present illness instead of the past health record. Up to 90% of emotional health data can be collected from the history of present illness because those questions are considered as mind-body aligned questions [34].

However the history of present illness does not always contribute to providing emotional data, but heart, lungs or other related issues have more emotional data embedded. The perfectly normal people more often have one problem to solve. Since the amount of information collected is limited in this case, it will have a less statistical significance which implies low emotional data. Desperate people will not give more data from which emotion can be identified easily.

To update the decision flow, we have incorporated the set of general questions responsible to identify user emotions between current concerns and medical history. This set of general questions are directly taken from PHQ questionnaires [34].

At end of each question set, users are provided with the option if they want to willingly end their consultation or if they want to continue to the next question set. In addition, users are provided with the summary of their responses at end of each question set and they can choose to update their responses at any point in time. The addition of a general question set after history of present illness can leverage patient engagement with the Emoti-bot.

5.3 Emoti-bot Architecture

The Emoti-bot architecture is shown in Figure 5.2. MongoDB NoSQL database is used to retrieve multiple question sets at run time and to store the user response of each question. Interface web apps can be executed in the Azure app service as an HTTP service its configured with Azure DevOps pipeline and GitHub repository for CI/CD to the Azure app service instance. We can also use a container application running as a docker instance in the Azure environment to get rid of errors due to environmental conditions. Containers provide a robust way to tie up your application as a single package and deploy it to the cloud. Environment standardisation and shared compute resources are the two major benefits of using docker instances in the run time. CI/CD pipeline is configured with Azure DevOps pipelines, so all the visual studio code changes are deployed to our cloud environment.

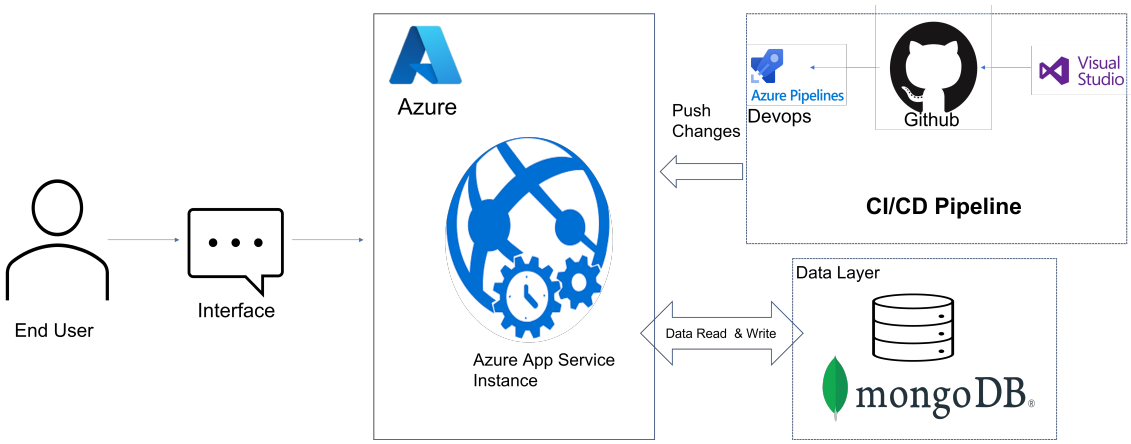

Figure 5.2: Architecture of Emoti-bot

5.4 Methods

This section briefly describes end to end methods identified for building and testing Emoti-bot.

5.4.1 Data Modeling in NoSQL

Data modeling is a standardised process of defining data requirements for our custom enterprise application logic and preparing standard templates to store data into databases. The main challenges in data modeling are to maintain a balance between your application requirements, the performance capabilities of your backend database engine and data access patterns. While designing data models and classes we have considered the application requirements for usage of data (like queries, data processing, data accessibility, etc.). Hence we have created a data model which can answer the most basic questions like

1. Is our application data read operation heavy?
2. Is our application data write heavy?

We have created our data model in NoSQL databases because NoSQL databases

have a lot of advantages over SQL databases. Like SQL database, we do not need to define fixed schema like number of columns and tables. NoSQL documents in a single group are unstructured and hence each document can have a different set of fields and data type. Documents in a single group of collections in NoSQL databases do not need to follow fixed schemas unless we restrict our database. Schemas are flexible, you can easily add new fields to the documents and remove some existing fields by changing the structure of documents in the database.

Each document in a NoSQL database represents a single entity or object. Each entity or object can match its corresponding fields in their respective documents even if it variates from other entities and objects inside the group of documents. The most guiding principle when creating data models in SQL databases is normalizing your data for avoiding redundancy in the database and refer the data. Data normalization refers to taking a single entity and breaking down it into multiple components. For example, field address can be broken down into home address and business address.

Due to this normalization in the SQL databases, updating or retrieving a single row in the database requires multiple join operations which may not be time and cost effective. Instead, we could define a single embedded document in the NoSQL database (basically denormalizing the data). For example, for address field we can define a single embedded document with fields home address and business address. This can reduce the subsequent number of read and write operations in the databases. Now retrieving a single entity from a NoSQL database is a single read-write operation. Hence by denormalization we can assist our applications to induce less cost of read and write operations in the database using NoSQL.

We are using MongoDB NoSQL databases to store and retrieve application data at run time. We have designed two different data models, one for retrieving the questions dynamically at run time and one for storing the user answer for each question. The data model diagram in Figure 5.3 indicates the structure of documents for storing

question details in the database. The Flow class in the database is designed to store the question sets in the respective order in which we ask users each question. So, list of question_set_id's is ordered data structure to store the flow of question sets in the decision tree. The other two documents respectively store details at different granularity levels. Question Set document stores the order of questions in that set which is same as that in decision tree. Question Details document consists of details of each question like question text, input choices, category, etc.

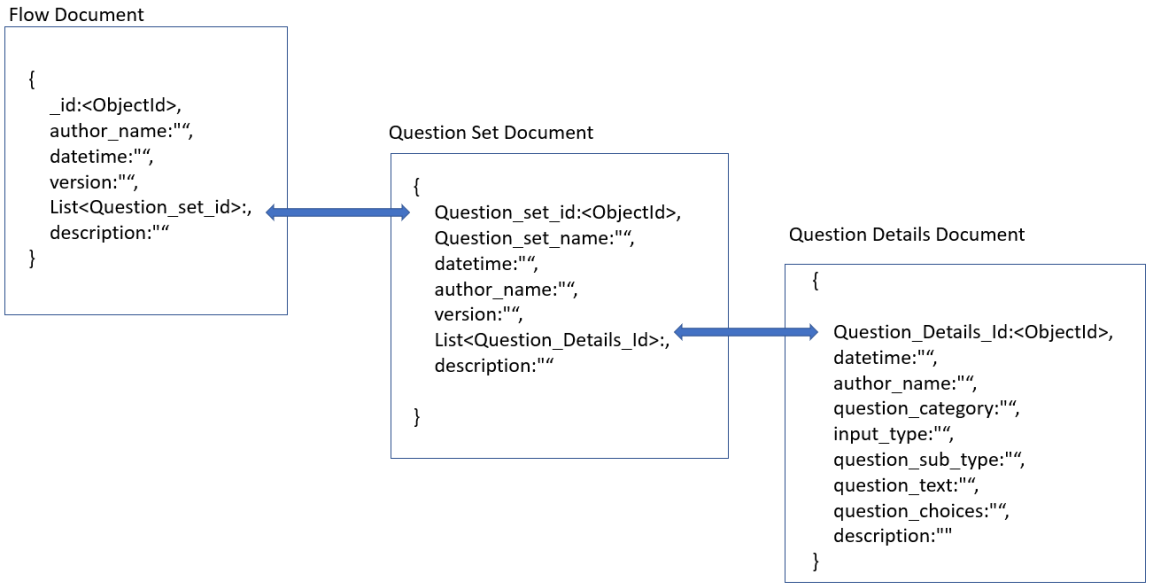

Figure 5.3: Data Model for Question Sets

The data model diagram in Figure 5.4 indicates the structure of embedded documents to store the user records and user answers to each question. The documents Answer Set and Answer Details are embedded documents in the User Record class which provides reference to store user answer to each question at the user level.

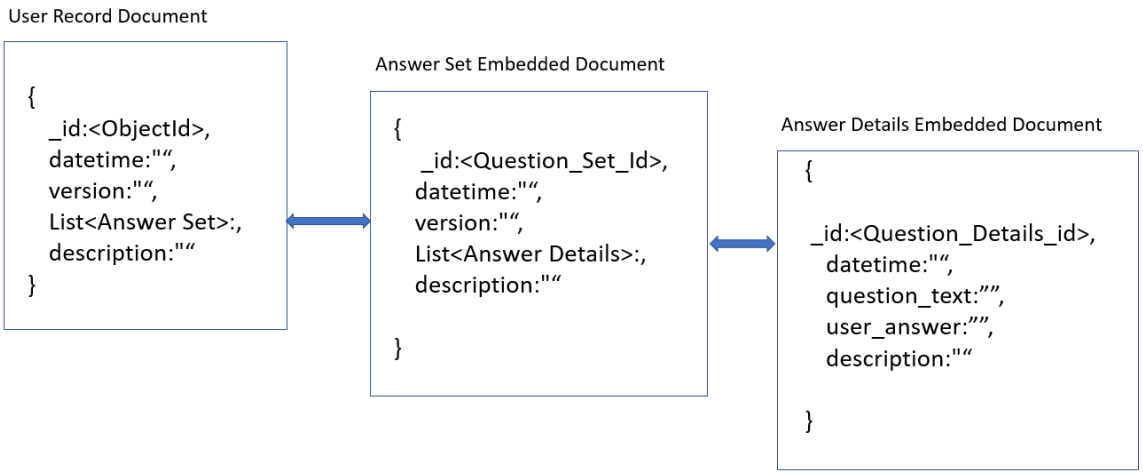

Figure 5.4: Data Model for User Response

5.4.2 Emoti-bot Using Bot Framework Composer

Microsoft bot framework composer is based on a bot framework SDK published as an open-source IDE for engineers to build dynamic conversational chatbots. It provides developers with support of NLP based models, QnA maker service, knowledge bases and dynamic response generation and enables these services with the code to handle complex tasks such as system integration. The conversational interfaces can be tested within the composer framework and deployed into Azure while maintaining any dependencies. Composer application is available as a desktop client and as cloud component [8].

Dialogs in the composer framework provide a unified way for chatbots to manage and handle conversational flow with the end-users. Event models and adaptive dialog frameworks can ease up the complex conversation modelling, by which we have build more dynamic, natural dialog flow along with context switching. Using, composer we need to focus on the conversational model rather than the mechanisms inside the dialog management.

Composer has enabled us to author dialog experiences with a visual designer which is a more efficient, easier approach to modelling for more complicated dialog

flows where context switching, interruption and management are prominent. [8].

As shown in Figure 5.5, with a simple visual editing canvas interface we have easily created complex decision flows and customized our bot for language understanding and generation. We have easily created the conditional branches to create a customized sequential flow of dialogues.

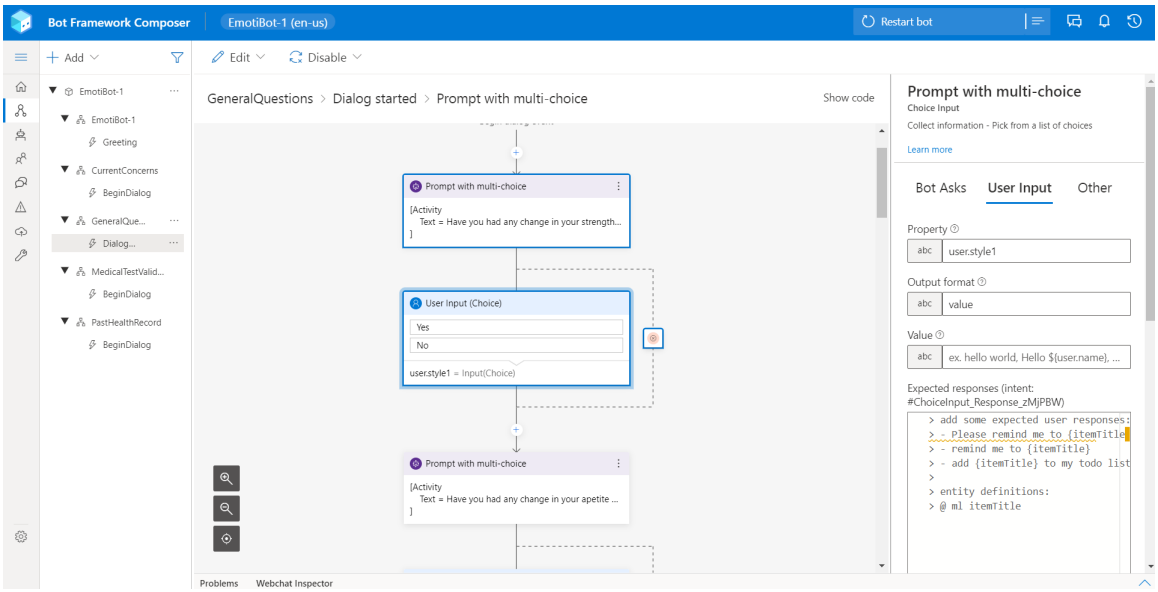

Figure 5.5: Bot Framework Composer Canvas

We have tested our chatbot once constructed in the webchat as shown in Figure 5.6. A bot is configured to take users through a series of general questions discussed in the decision tree.

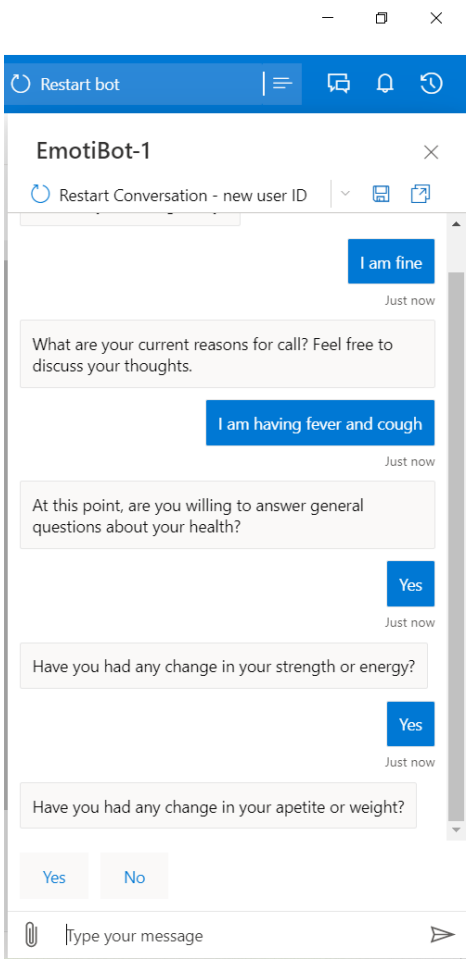

Figure 5.6: Testing bot with Composer

In order to get more control over dynamic flow in a programmatic way, we have used Microsoft bot Framework SDK. With SDK, we have complete control to configure dialogue flows from the database.

5.4.3 Emoti-bot using with Microsoft Bot SDK

The Microsoft bot framework SDK with Azure bot service provides built-in abstractions and templates to develop, test and deploy chatbot in the cloud and manage AI based development using Azure cognitive services. Microsoft bot SDK can be easily extended to use related templates, databases, and Azure AI services.

The **Azure Bot Object** has the dialog reasoning and logic for each turn, and

it exposes the turn handler method that can accept the forthcoming activities from the Bot Adapter. The Microsoft bot framework SDK provides multiple models for maintaining turn logic for the chatbot [3].

Activity Handlers class provides a built-in event-driven model in which the forthcoming activity types are identified as the different events. With this mechanism, we can easily handle conversational models with short interactions with the users. In the bot logic,

- We have implemented handlers for each activity identified using Activity Handler class and those handlers are triggered for appropriate response from the bot.

- For long running conversations, Dialogs library are the key to implement state-based model with the users. Hence, for large running sequential flows, we have used Dialogs library to manage the complex flow of questionnaires.

- We have implemented activity handler and component dialog class for dynamic management of event-based models.

- For each turn, we have designed our own custom logic for handling complex flow by implementing the bot class. For handling wide range of user interactions, we have used dialog manager and adaptive dialog classes.

Activities come from bot framework service through HTTP requests. The bot replies to inbound activity with a 200 HTTP status code. The response sent from the bot to the channel is from different HTTP post to the bot framework service. The bot response is also acknowledged with 200 status codes via HTTP response [3].

The HTTP based protocols will not specify the order in which post requests are made along with their acknowledgements. But in order to fit with HTTP based services, nested requests are used which implies that outbound activities are made

within the scope of inbound requests. For back-to-back unified HTTP connections, security models should provide acknowledgement for both requests.

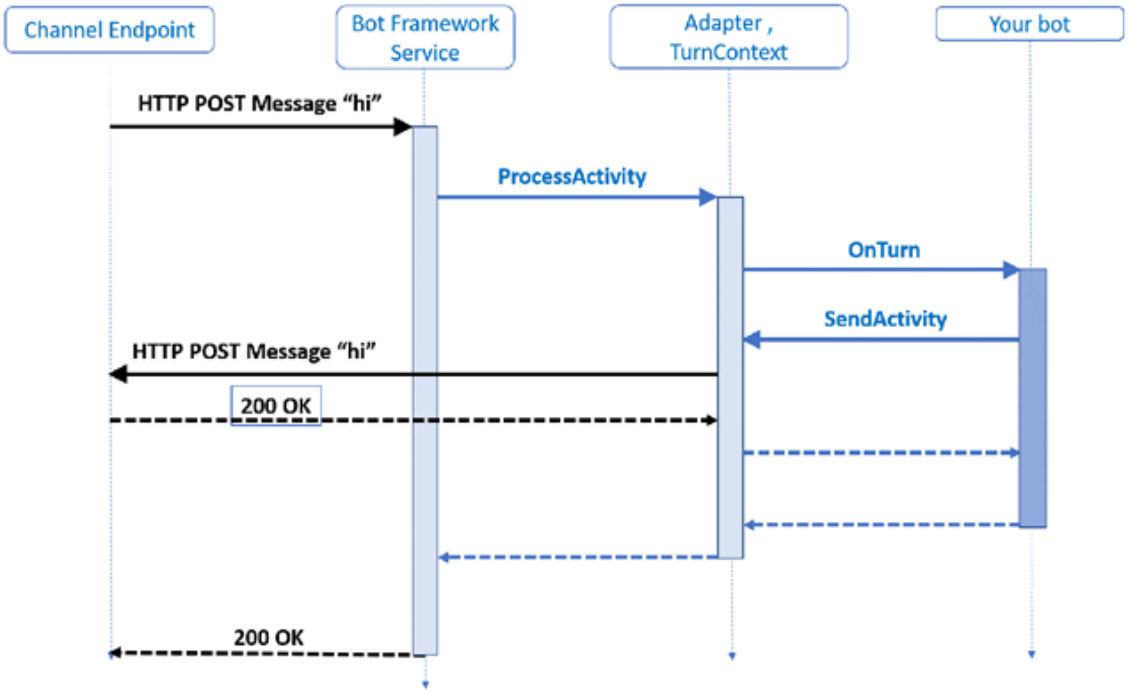

Figure 5.7: Activity Diagram of Microsoft Bot [3]

The activity diagram of a chatbot using a bot framework is shown in Figure 5.7. The message activity from the bot is the same as that of the user. The process begins with an HTTP post request, with information passed as JSON load arriving at the webserver. Since we are programming the bot in C#, this is an ASP.NET framework.

Adapter is the primary component of Microsoft framework and is a core component during the run time. The activity is submitted over as JSON in the HTTP POST request. Activity object is later initiated by deserializing the JSON request and is passed to adapter through its built-in function i.e., process activity. As soon as this activity is received at the server, the adapter creates a turn context and calls the middleware object [3].

Turn Context object is responsible to provide the mechanism for the chatbot to send outbound messages and activities, mostly in response to inbound messages.

In order to implement this mechanism, turn context provides methods such as update,send, delete, and response. All the response functions run in an asynchronous process [3].

Azure chatbots are inherently stateless similar to modern web applications. The internal state in the bot is managed by storage layer abstractions provided by the Azure bot toolkit to make state management easier.

Activity related information is also stored and managed by turn context. It provides additional information during each turn across layers of the bot. It is one of the prominent abstractions and is also configured to carry inbound, outbound activities to middleware components as well as the bot.

A **Waterfall Dialog** is implemented in such a way that it is mostly used for collecting information from the end-user in a very guided and logical manner. Each singular waterfall step is implemented as an asynchronous function that accepts the waterfall step context parameter as its argument. At each particular step, the chatbot asks the user for input and can initiate a child dialog which itself is a prompt dialog, then it waits for user response and passes the user response to the next step. Hence, user response is preserved from one function to another and so on [12]. The Figure 5.8 shows a series of waterfall steps and stack execution.

We have implemented the class of Waterfall Dialog library to execute the steps in a sequential manner. At each waterfall step, the bot will retrieve the next question in a flow from the database and store the user response. After completion of each question set, users are provided with the option of whether they want to change the response or not. Later the updated response is changed in the database.

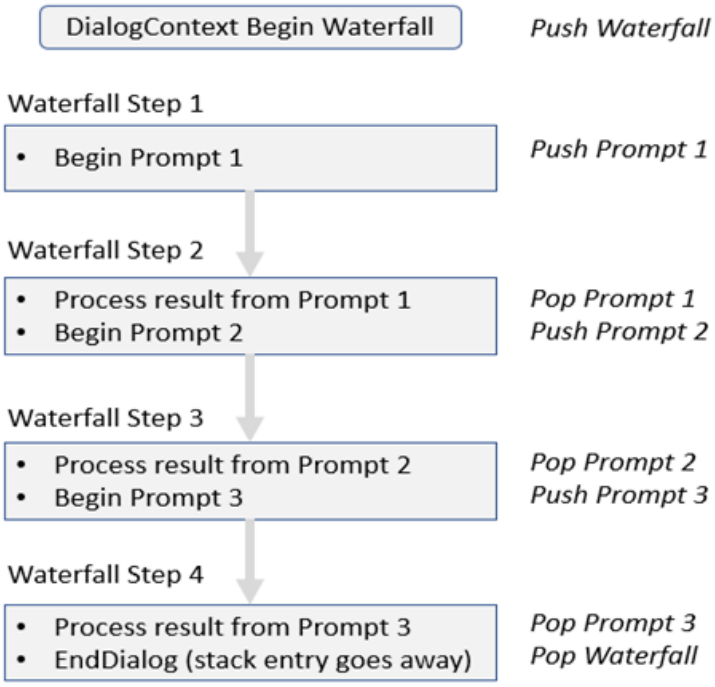

Figure 5.8: Waterfall Dialogs [12]

5.4.4 Testing Bot with Bot Framework Emulator

Once programming is completed, we can easily test the bot source code and run the bot locally for inspection and error handling. We can test the bot source code by running the bot with the Bot framework emulator. The emulator is a client application that will provide mechanism to test chatbots locally as well as online on Azure web service.

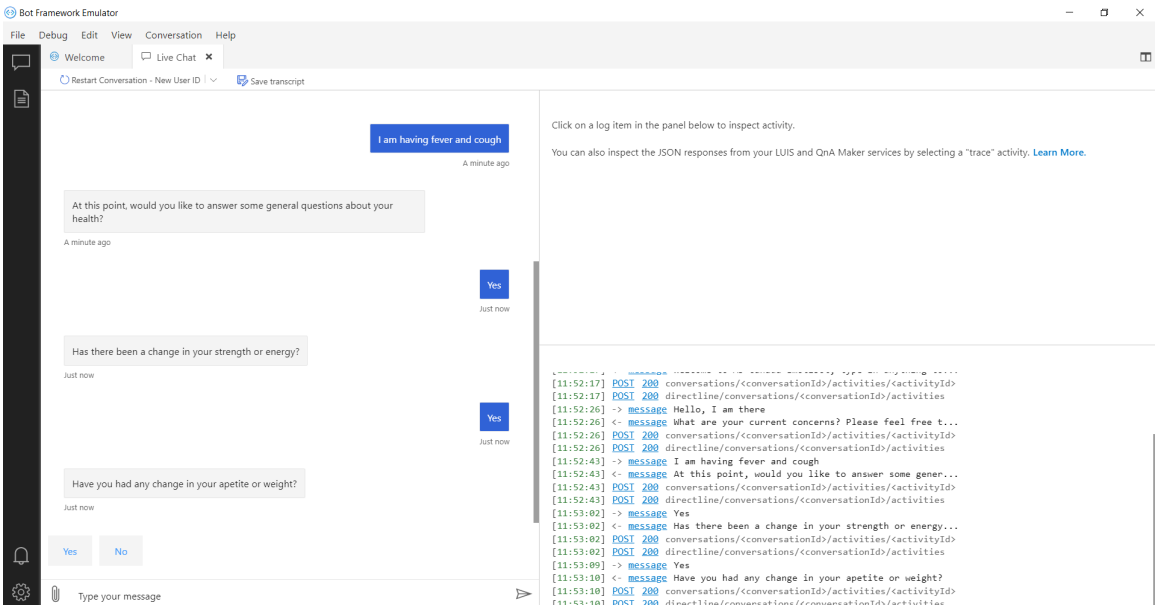

Figure 5.9: Testing Emoti-bot with Emulator-1

With the emulator, we have inspected and analyzed messages that our chatbot sends and receives. Messages are displayed in the emulator as they would appear in the web interface. It logs requests and responses in JSON format.

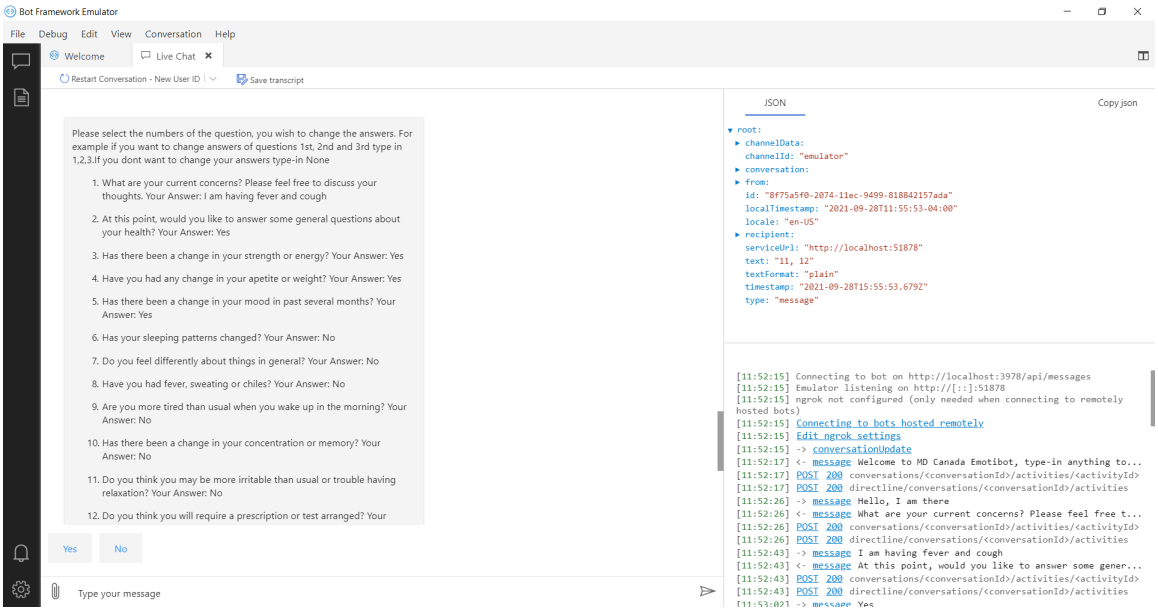

Figure 5.10: Testing Chatbot with Emulator-2

5.5 Summary

This chapter was mainly focused to discuss generalised decision flow to imitate the clinical trials and narrow the scope of clinical diagnosis during a COVID-19 pandemic. We discussed various novel methods to build the Emoti-bot architecture by closing working with domain experts at MD Canada. The next chapter will discuss emotion modelling in chatbots.

Chapter 6

Emotion Modeling in Chatbots

6.1 Introduction

WHO defines mental health as "a state of well-being in which a person feels his or her own capabilities, can cope with regular stress of human life, can work effectively and is capable to contribute to his or her own community". Though it has a direct impact on one's physical well-being and behaviour, emotional health is one of the most neglected areas of healthcare. Globally, our society still stigmatizes emotional health-related issues even in this fast-forwarded modern era. The new coronavirus pandemic had a significant effect globally and had inflicted havoc in urban and rural areas. The never seen effects of pandemic hold the capacity to affect the psychological well-being of the global population [18].

Firstly, individuals who are affected by the novel virus, those who are at high risk of carrying the virus, frontline health workers, and doctors who are in immediate contact with patients can end up having mental disorders like anxiety, stress, loneliness, or depression. The isolation norms can enhance the concerns of individuals in need of affection because of limitations in physical contact. In addition, the global economical downfall can contribute to emotional instability.

These unprecedented times have seen a massive increase in the number of physical violence and suicidal victims. Hence, a mental health issue is a most widely discussed topic nowadays.

Recent research and developments are significantly directed towards chatbots being emotionally intelligent [41]. The novel pandemic has significantly increased the need for emotionally smart chatbots. People still owe to social stigmas about mental healthcare and fear to reach out to others for emotional support. This is where a chatbot comes into the picture as a therapeutic agent. The anonymous behaviour of chatbots allows users to be more outspoken about their feelings and emotions. Most of the technical chatbots in recent years are capable to reciprocate the end-user sentiment. If we design chatbots effectively they can provide emotional support and can decrease a load of therapists at the same time.

Hence our main contribution is to explore end to end process of identifying deep learning models that can be used for emotion detection from the user utterance in chatbots. This chapter introduces some empirically proved deep learning architectures used to identify emotions expressed in the text in chatbots. Later, we discuss some future works on how to generate empathetic responses well suited to user utterance without losing the context of the conversation in chatbots.

6.2 Background and Literature Review

Emotion models are the basis of the process following emotion detection. These models describe the way for representation of emotions. We have briefly discussed some of the emotion models below.

Categorical Emotion model is popularly defined as a discrete emotion model. The categorical emotion model assumes that few emotions are universally accepted and are independent of each other. These sets of emotions are joy, sadness, anger, fear,

disgust, and surprise. The model categorizes emotions expressed into distinguishable classes. Popularly known models are the Plutchick model, OCC model and Ekman Model. Robert Pluthick model defines that emotions are expressed in contrary notions like joy vs sadness, anger vs fear, trust vs disgust, and surprise vs anticipation [45]. Paul Ekman model explains that emotions are expressed from neural systems and how every individual perceives a particular situation [33]. The Ekman model differentiated emotion into six different classes i.e. anger, fear, joy, sadness, surprise, and disgust.

The OCC model apposed the basic foundation theory of independent emotions by Ekman and Pluthick model. OCC model describes emotions expressed on the basis of intensity and how the human beings perceived certain events. The model added 16 additional emotion classes to the Ekman model of emotions like shame, admiration, envy, gloating, and many more [25].

Dimensional Emotion Model describes that emotions are dependent of each other. Emotions are represented in a multi-dimensional space and the model shows how emotions are related to each other focused on the occurrence of severity levels of certain events i.e. high or low [23]. The dimensional space is most commonly described as below.

1. Valence: This dimension describes emotions as positive or negative emotions
2. Arousal: This dimension describes emotions as apathetic or exited
3. Dominance. In this dimension emotion is expressed by certain degree.

Componential Emotional Model extends the basic foundation of dimensional theory and states that an individual may feel an emotion is extracted from certain events. Appraisal theory is a popularly used componential model. Appraisal theory defines that an individual can only express emotion if it is created by appraisal of entity and event outcome is dependent on individual's experience, goals, and probabilities of actions [27].

Only a few research groups have studied the potential of chatbots for providing mental health support. There are multiple ways to provide mental health support through chatbots. From psychiatric counseling [40], behaviour change [44], Cognitive Behavioural Therapy (CBT) [30], all the approaches are delivered through chatbots. We have mainly focused on data sets with emotion labels following the categorical emotional theory from the famous Ekman and Pluthick model.

A deep learning-based emotion identification model was trained using LSTM architecture. Experimental results exemplified the effectiveness of LSTM based architectures in the classification of six different emotion labels [32].

Gated Recurrent Unit (GRU) architecture was applied for the classification of more than ten emotion labels. The state of art model exceeded 90% accuracy [13].

An emotional chatting architecture was studied to generate emotional response after classifying emotions. The architecture is trained with emotion label and response data set along with additional concepts of internal and external memory. Emotions are expressed in response from the advanced vocabulary stored in external memory [53].

6.3 Methods

In this section we briefly discuss end to end process of training deep learning models.

6.3.1 Dataset Description

Source	Dataset Description
Emotions Dataset of Tweets to Study Interactions between Affect Categories [39]	Dataset Consists of 10,983 tweets labelled with 11 emotions .
ISEAR [49]	Dataset Consists of around 8000 sentences labeled with 7 emotions
Kaggle Contextualized Affect Representations for Emotion Recognition [47]	Dataset Consists of around 20,000 sentences labeled with 6 emotions
Emobank [22]	More than 10000 sequences labeled with VAD(Valence, Appraisal and Dominance) technique
Goemotions: Dataset of Fine graned emotions [28]	More than 50,000 samples with over 28 emotion classes

Table 6.1: Open Source Datasets

The above Table 6.1 describes some of the publicly available datasets for emotion detection. Some of the datasets follow the classification theory of Ekman and Pluthicks' emotion model while some of datasets are described as multi-dimensional models. Since our project is mainly focused on classification models, we have used two of open source datasets for our experiments. Firstly, we have used ISEAR emotion dataset with six emotion labels namely disgust, anger, sadness, shame, fear, joy and guilt. Distribution of each emotion in the dataset is almost equal. Secondly, we have used Kaggle emotion dataset with six different emotion categories namely, joy, sadness, anger, fear, love and surprise. The distribution of labels in the dataset is unbalanced but each label has more counts compared to ISEAR emotion dataset.

6.3.2 Dataset Cleaning and Processing

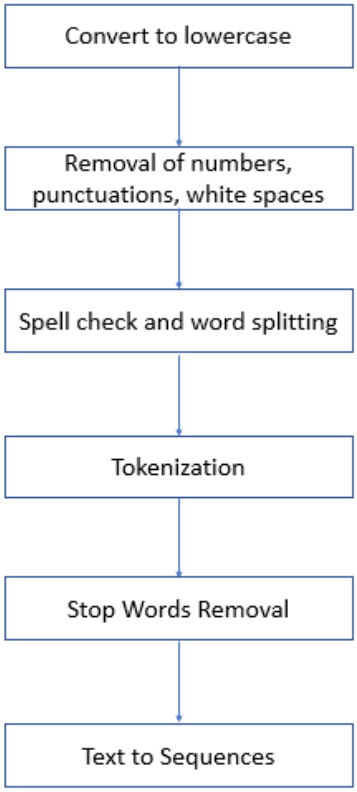

Figure 6.1: Data pre-processing steps

The data set in NLP needs to be pre-processed and cleaned before it is used for training deep learning algorithms. The above flow chart describes multiple steps executed to clean and pre-process the data. Each word in the data set is converted into lower case. Numbers, punctuation and white spaces are removed programmatically from the text. Also, we have used some open-source python libraries like Pyspeller, Autocorrect to correct spelling errors in the text. Combined words are broken down into meaningful words using dynamic programming algorithms like the Viterbi segment algorithm [11]. We have tokenized input sentences as a sequence of words and removed stop words from each sentence in the dataset. Finally, each word is represented by its index in the word dictionary created from the data set after the removal of stop

words.

6.3.3 Word2Vec Algorithms

Machine learning algorithms cannot accept the raw text as input and they only work with specific input and output formats. Hence, we need to convert raw text into integers or vectors of numbers into a specific format. Word Embeddings is the popular technique by which we can covert sentences into vectors of numbers. Some of the techniques used traditionally are described in this section below.

Bag of Words is the simplest linguistic model to represent text as numbers. Each sentence is represented with the length of vector equal to the size of vocabulary by indicating the presence or absence of words with 1s and 0s inside the vector. Bag of Words suffers from many limitations. If the size of vocabulary increases, the size of vectors also increases which results in sparse matrices. Sparse matrices require large memory and powerful machines for processing. Bag of Words cannot retain the semantic or linguistic meaning of words from the sequence [58].

TF-IDF is a mathematical approach designed for information retrieval. It evaluates the relevance of words inside the document. The term increases significantly with the occurrence of words inside the document. The term is offset by the number of documents that contain the word.

CBOW and Skip-Grams are two popular methods to learn word embeddings by using neural network models. In the CBOW method, the centre word is predicted based on context (surrounding words). While in Skip-gram word model predicts the surrounding words based on the centre (context) words. Both the methods are accurate in capturing word semantics and meanings from the word corpus [37].

Glove Vectors operates on major limitations of word2vec algorithms mentioned previously. Firstly, the Latent Matrix Factorization method popularly used for learning word vectors perform well to collect statistical information but lacks on word

analogy task. Secondly, the skip-gram model might do well on the word-analogy tasks, but it lacks in utilizing statistical information.

We are using Glove vectors which is a large pre-trained set of word vectors to represent each of the words in our corpus as a word vector of 300 dimensions. The concept of Glove vectors is, "Judge the word by the company it keeps", Glove basically is an unsupervised algorithm for creating vector representation of words. Training is performed on global co-word occurrence statistics from a large corpus which represents interesting substructures in word vector space [43].

The cosine similarity between two-word vectors provides an effective method for measuring linguistic and semantic similarity between two words. This similarity metric used for nearest neighbour evaluations gives a non-scalar number that quantifies the relatedness of the words. This is mainly helpful because they are so many pairs of words which represent the same level of information like man and woman, girl, and boy, etc. Since Glove vectors is an effective way to capture semantic and linguistic information between two words, we are using pre-trained Glove word vectors as an embedding layer to our deep learning architectures.

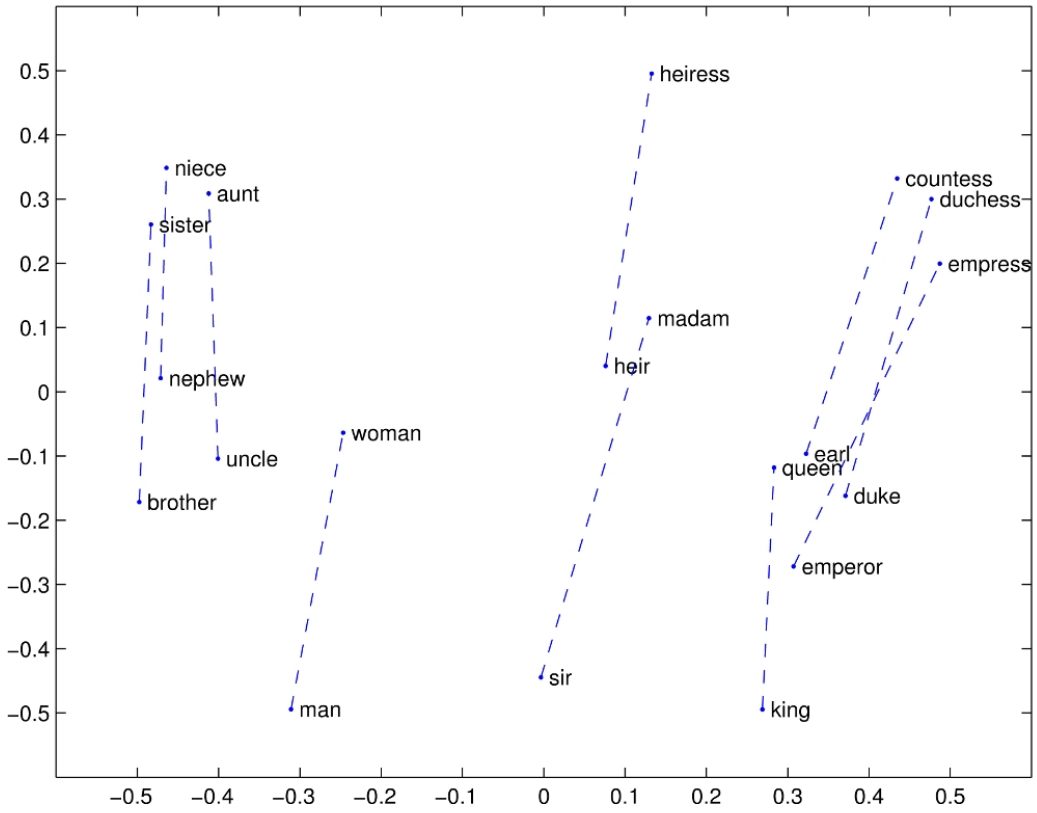

Figure 6.2: Glove Vectors [43]

6.3.4 Deep Learning Architectures

In this section we have defined some of sequence to sequence learning architectures useful for text classification.

The human brain is not persistent enough to start thinking from scratch whenever required. Traditional neural networks are unable to perform this task, which is one of the major limitations. Recurrent Neural Network (RNN) is capable to address this issue, they work in loops which allows them to persist the previous information. These loops make recurrent neural networks seem to be the unique way of modelling things, but they are like any other traditional neural network.

As shown in Figure 6.3 below, RNN can be considered as multiple instances of same network as passing messages to its successors. This chain-like behavior naturally reveals that a RNN is related to chain like sequence of inputs and data.

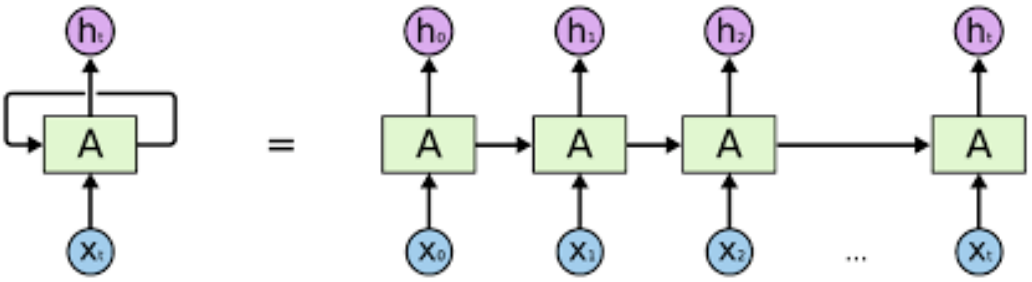

Figure 6.3: Recurrent Neural Networks (RNN) [10]

Usually, we only need to look at recent information to do the present task. Suppose RNN is designed to predict the next word in sentences based on previous inputs. For example, we need to predict the last word in the sentence "The stars are in the sky", we don't need to look at previous information it's obvious that last word will be the sky. Thus, in such examples where gaps between previous information are less, RNN's are best to learn past information. But as the gap between previous and present information starts increasing RNN's are unable to connect the long chain of information. Hence, for handling long term dependencies LSTM is used [10].

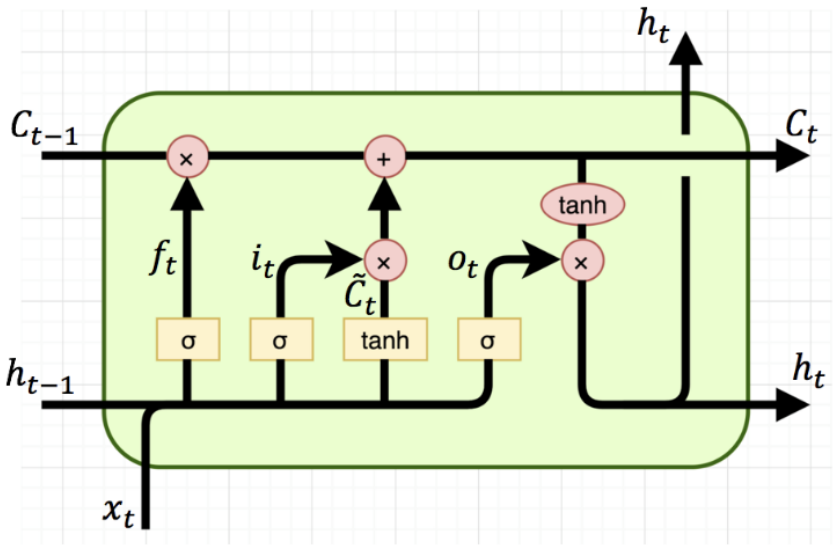

Figure 6.4: Long Short Term Memory(LSTM) [10]

The main part of the LSTM architecture is the cell state. The horizontal line

upwards decides what information we can eliminate and what new information we can add to the cell state. Gates are the way how LSTM can add and remove information from the cell state. As shown in the below Figure 6.5, gates consist of the combination of sigmoid activation function and point wise multiplication operation. Sigmoid activation function transforms the given input into ranges from 0 to 1; where 0 indicates no information to add in the new cell state, while 1 indicates adding all the information in the updated cell state.

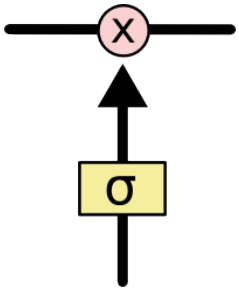

Figure 6.5: LSTM Gates [10]

The equation for forget gate is stated as

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{6.1}$$

The next step performs an operation to add new information to the cell state. Mathematical operations defined to add new input and cell state are given below

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{6.2}$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{6.3}$$

Further step updates the new cell state. Point wise multiplication of forget gate (with information discarded) is added to the new information calculated by equation

6.2 and 6.3. Hence the updated cell state is

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6.4}$$

The final output state in LSTM is calculated as

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_0 \tag{6.5}$$

$$h_t = o_t * \tanh(C_t) \tag{6.6}$$

The more simplified version of LSTM is Gated Recurrent Unit (GRU). It combines the forget gate and input gate into one update gate. It also combines cell state and hidden state into one cell state called hidden state. The resulting architecture of GRU is less complex compared to LSTM as shown below in Figure 6.6.

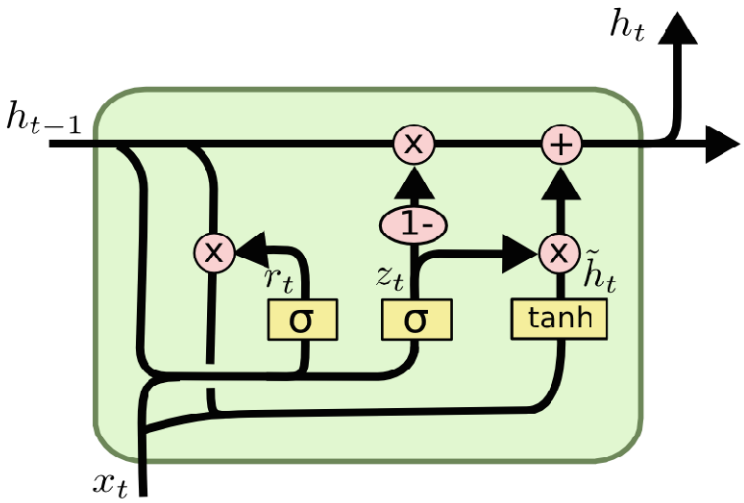

Figure 6.6: Gated Recurrent Unit (GRU) [10]

Equations defining GRU model [24] are stated below

$$Z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{6.7}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{6.8}$$

$$\tilde{h_t} = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{6.9}$$

$$h_t = (1 - Z_t) * h_{t-1} + Z_t * \tilde{h_t} \tag{6.10}$$

6.4 Experiments

We have explored some of the deep learning architectures to identify emotions from the text.

6.4.1 Overfitting LSTM

In the very first experiment, the LSTM model was trained on the ISEAR data set. Since the data set was balanced we trained LSTM architecture by randomly splitting the data set into the train, validation, and testing. The splitting ratio was 70%, 10% and 20% respectively. As shown in Figure 6.7, while training we found that the LSTM was overfitting after a certain epoch i.e. the model was not performing better on unseen data.There can be many reasons for LSTM to overfit.

- Complexity of the model
- Large training parameters
- Dataset is relatively small to fit all possible examples from natural language

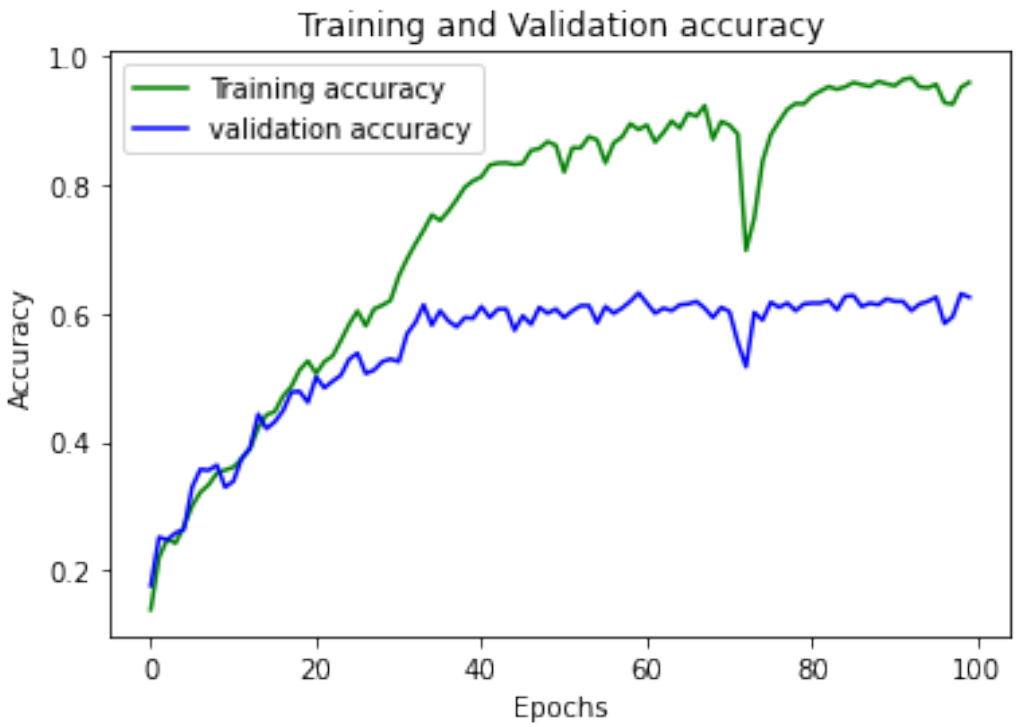

Figure 6.7: LSTM Overfitting

Since we have used dropout in LSTM layers which can randomly drop LSTM units, complexity and large training parameters do not seem to be the obvious reasons. Hence, we concluded the main reason for model overfitting to be a small amount of training data for each label.

6.4.2 Stacked Deep Learning Models

Since the simple LSTM model trained on the ISEAR dataset was overfitted, we tried a set of different deep learning architectures for evaluation. The next experiment was conducted by combining datasets from ISEAR and KAGGLE. The stacked deep learning architecture is described in Figure 6.8 with labelled input and output. The architecture shown below is empirically derived by evaluating the model on unseen data. Under the hood, we have tried many different architectures with varying input and output sizes. We have defined two cells on top of the embedding layer from glove

vectors for classification. We have finally trained to variants of RNN i.e. LSTM and GRU.

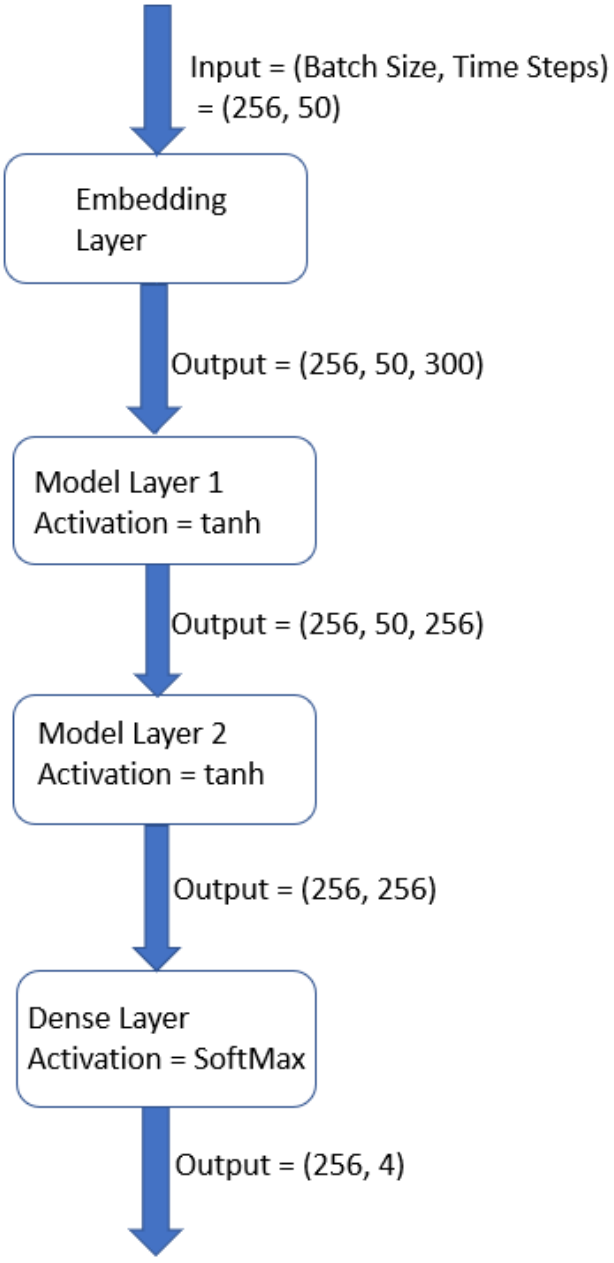

Figure 6.8: Stacked Architecture

6.4.3 Metrics Definitions

For each stage of the experiment, we are reporting four metrics namely precision, recall, f1-score and accuracy. Generally, metrics of interest are identified by domain experts based on their domain knowledge and objectives specified. The main purpose of reporting metrics is to measure "How successful the outcome is based on the objectives specified?". In the healthcare domain mainly for classification, domain experts preferably report recall compared to precision. The main aim of a chatbot is to identify emotions expressed in user utterances i.e. chatbot effectively needs to identify a varied range of emotions to generate an emotionally intelligent response. Hence, we would preferably identify, Weighted F1-Scores as an objective to evaluate different deep learning models.

1. Precision is considered when reducing number of false positives.

$$precision = \frac{TP}{TP + FP} \tag{6.11}$$

   In the above equation 6.11, TP is True Positive. FP is False Positive.

2. Recall is considered when reducing number of false negatives.

$$recall = \frac{TP}{TP + FN} \tag{6.12}$$

   In the above equation 6.12, FN indicates False Negative.

3. F1-score is the harmonic mean of precision and recall.

$$f1 - score = \frac{2 * precision * recall}{precision + recall} \tag{6.13}$$

4. Accuracy is defined as percentage of instances correctly identified out of total

instances in database.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6.14}$$

6.4.4 Results

Both variants of RNN (i.e. LSTM and GRU) show convincing results when trained as sequence of stacked layers. The results of stacked LSTM and stacked GRU is reported in Table 6.2 and 6.3 respectively. Both the models are effective when trained on combination of large datasets.

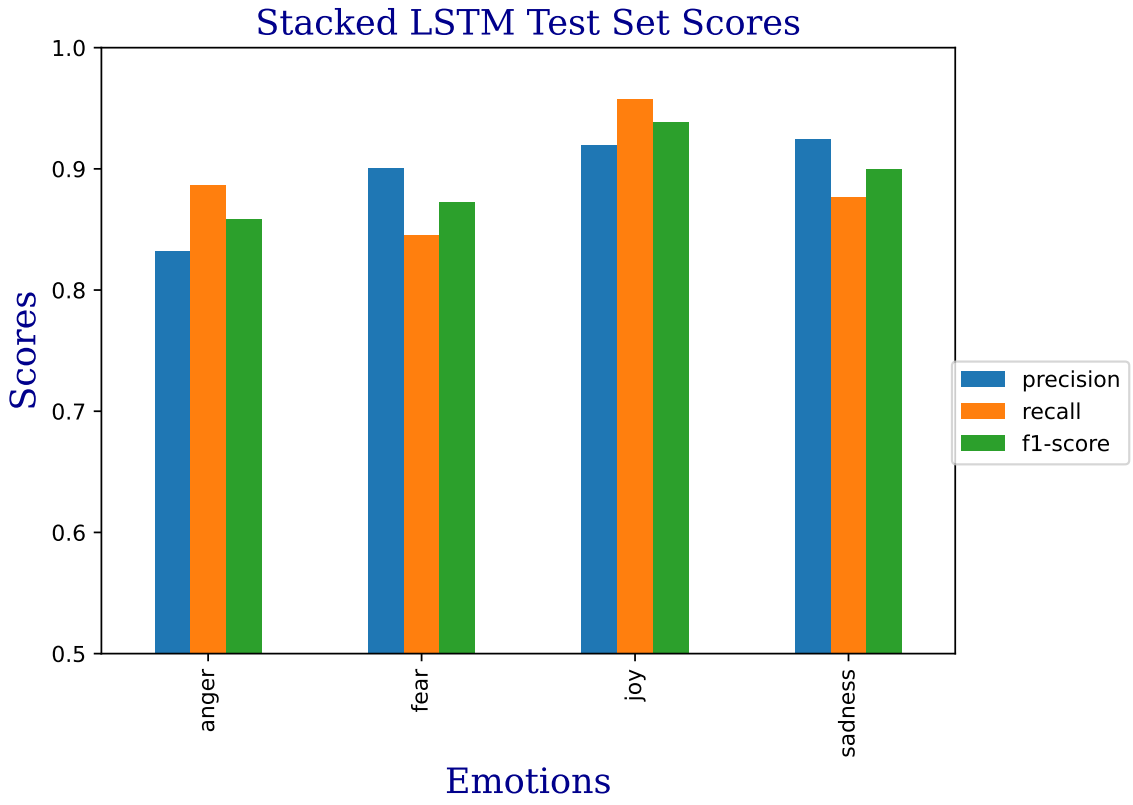

Figure 6.9: Stacked LSTM Test Set Scores

Label	Precision	Recall	F1-Score
anger	0.83	0.89	0.86
fear	0.90	0.85	0.87
joy	**0.92**	**0.96**	**0.94**
sadness	0.92	0.88	0.90

Table 6.2: Stacked LSTM Test Set Scores

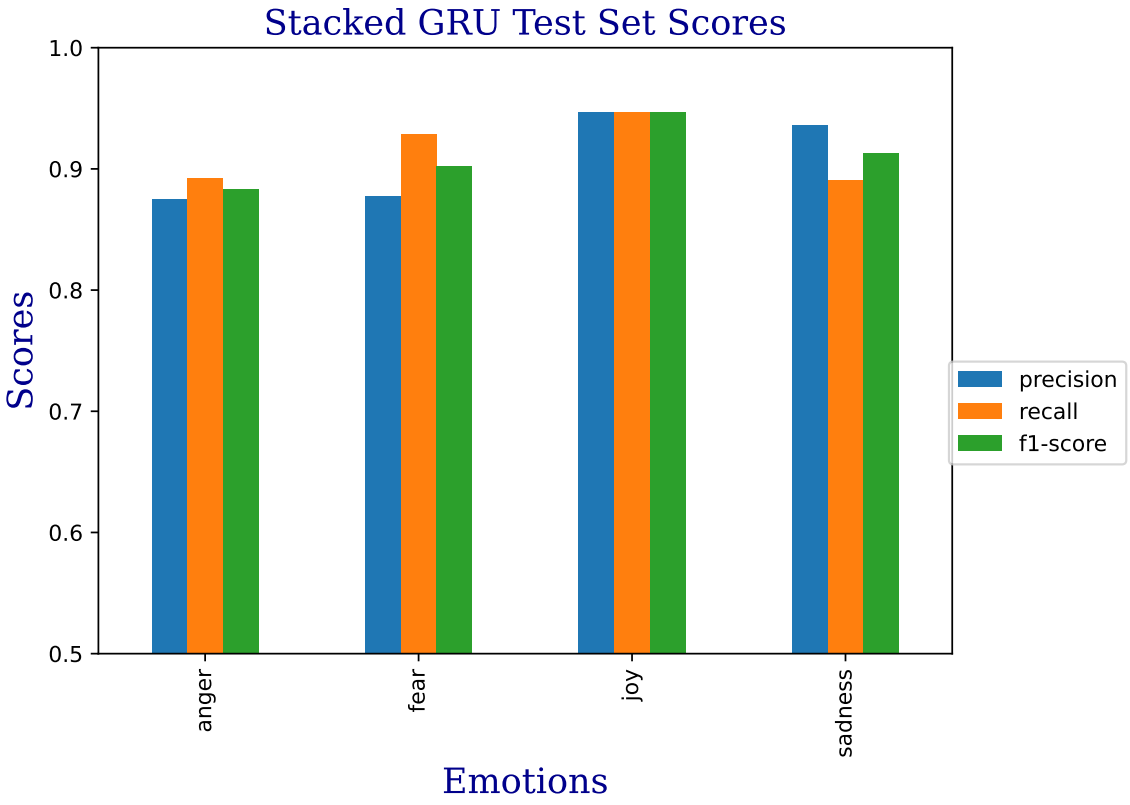

Figure 6.10: Stacked GRU Test Set Scores

Label	Precision	Recall	F1-Score
anger	0.88	0.89	0.88
fear	0.88	0.93	0.90
joy	**0.95**	**0.95**	**0.95**
sadness	0.94	0.89	0.91

Table 6.3: Stacked GRU Test Set Scores

In summary, we conducted three experiments to identify emotions from the text in chatbots. Firstly, we trained a simple LSTM model on ISEAR unbalanced data set and as a result, the model was overfitting. The stacked deep learning models convincingly provided better results on the combined dataset. Finally, we are reporting two metrics i.e. Accuracy and Weighted f1-score, from the training validation and testing set of each experiment in Figures 6.11 and 6.12 respectively.

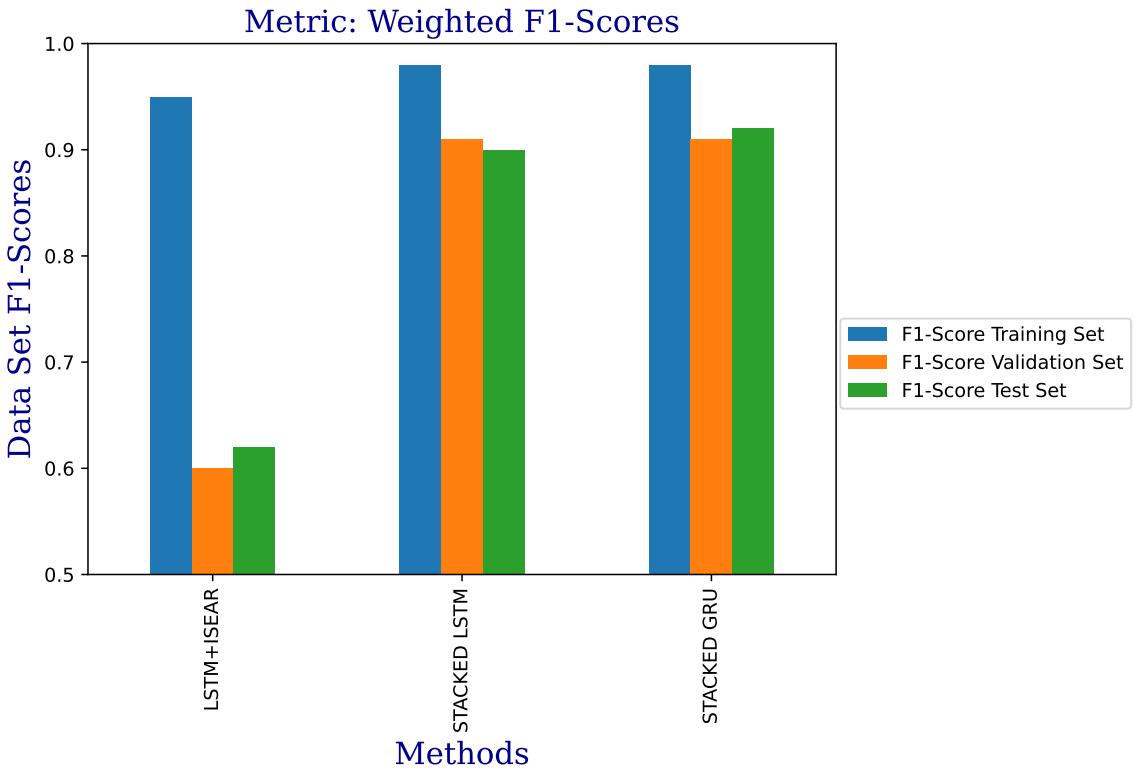

Figure 6.11: Metric:Weighted f1-score

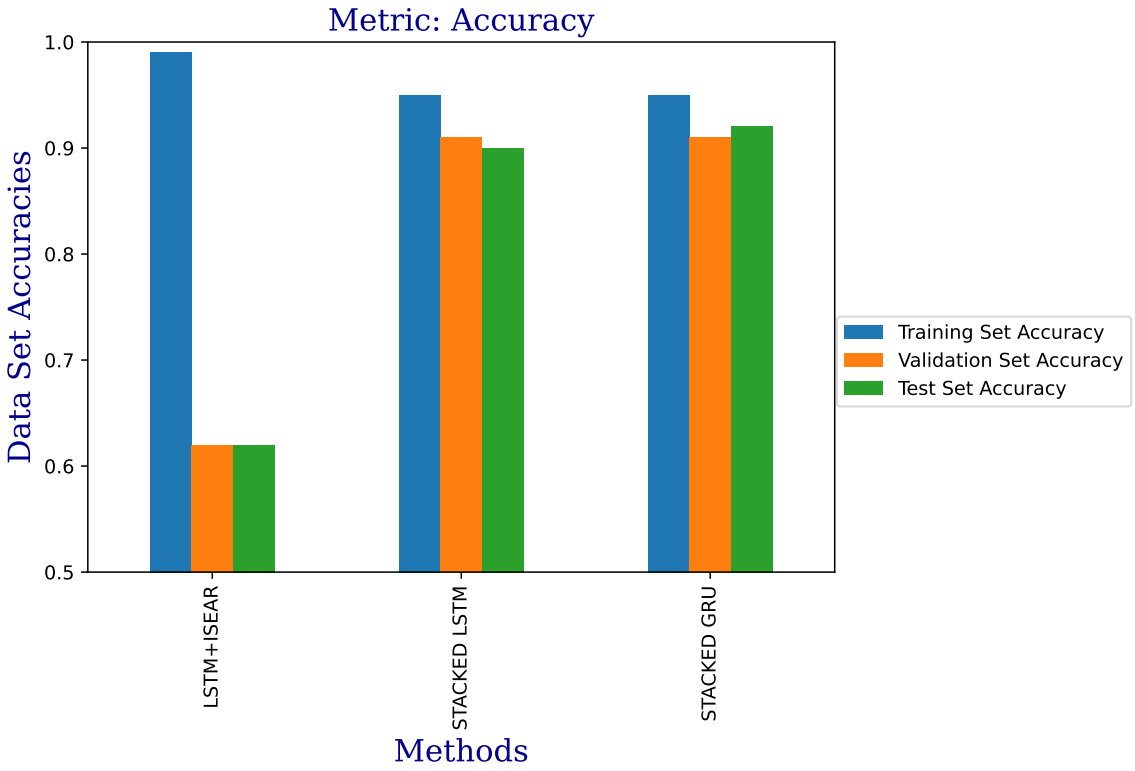

Figure 6.12: Metric:Accuracy

Method	**Training Set Accuracy**	**Validation Set Accuracy**	**Test Set Accuracy**
LSTM + ISEAR	0.99	0.62	0.62
STACKED LSTM	0.94	0.91	0.9
STACKED GRU	**0.95**	**0.91**	**0.92**

Table 6.4: Summary Scores: Accuracy

Method	Training Set Weighted F1-Score	Validation Set Weighted F1-Score	Test Set Weighted F1-Score
LSTM + ISEAR	0.95	0.6	0.622
STACKED LSTM	0.95	0.91	0.9
STACKED GRU	**0.95**	**0.92**	**0.92**

Table 6.5: Summary scores: Weighted f1-score

6.5 Summary

The main purpose of this chapter was to explore deep learning models to identify end-user emotion in chatbots. Hence, we have identified end to end process of training deep learning models in NLP. The next chapter will identify ethical concerns for conversational agents in healthcare.

Chapter 7

Ethical Chatops in Healthcare

In the last few years, health ethicists have tried to unify the framework for the ethical deployment of evolutionary technologies, especially for AI in healthcare [42]. Due to the rapid increase in Conversational AI applied to healthcare during the COVID-19 pandemic, currently, there are no established standards to evaluate chatbots from an ethical perspective. The Conversations per Session (CPS) measure was defined to quantify the metric for ethical evaluation [51]. But these quantitative metrics ignore the social, political and ethical challenge these chatbots accrue, especially in the domain of healthcare.

The ethical practices considering beneficence, non-maleficence, explicability, justice, and autonomy as themes were discussed for future developments in Cognitive Behavioral Therapy (CBT) mental health chatbots to guide ethical research and policies in the post-COVID-19 society [55]. The World Economic Forum's RESET (Reveal, Escalate, Substitute, Explain, and Track) was launched in January 2020 to develop a governance framework for the ethical use of chatbots in the health domain by combining folks from multiple disciplines of chatbot developers, healthcare workers and academia [54]. The research forum bought the global experts to make a positive impact and minimize the negative consequences of using chatbots in healthcare. As

a result of this, the governance framework consists of ten principles and seventy-five recommendation actions.

In order to effectively apply chatbots in healthcare, clinical practices should be incorporated which are actually being applied for mental health in clinics. Based on previous literature's, we have identified ethical concerns related to the use of chatbots in healthcare. Firstly, patients' sensitive data should be kept private and confidential. In patient-doctor interaction where patients' privacy is protected, conversational agents generally tend to ignore these aspects. Safety is another concern while using chatbots. This can be subjected to the inability of chatbots to comprehend users' utterances and tackle emergency situations. Currently, chatbots lack the ability to remember past conversations and identifying emotional cues from utterances. Imitating therapists in chatbots can create an impression of talking to real humans. This deception can be considered unethical from the healthcare perspective. Finally, there is a potent risk of misuse especially to replace already established services [29].

Physicians think that chatbots are unable to provide effective healthcare because they lack the intelligence of assessing patients and their inability to reciprocate user emotions [29]. The impact of chatbots on the patient-doctor relationship is still unclear.Long term effects of chatbots in healthcare are still unclear but many recent studies have achieved short term advantages that had a positive impact on digital healthcare solutions.

Lastly, the next chapter will summarize the project with some future work in this direction.

Chapter 8

Conclusion

8.1 Summary

In conclusion, chatbots can be used to deliver digital health care solutions if designed appropriately. This project research mainly discussed the role of chatbots in mitigating the global effects of the COVID-19 pandemic on the healthcare domain. We have studied problem statements on three different levels. Firstly, we have designed QnA chatbots using advanced cloud services which can be frequently updated with continuous integration and deployment. To develop chatbots for online consultation health services, we have created a generalised decision flow to mimic the onsite consultations. Lastly, we have explored deep learning models which can be used for emotion identification from the text in chatbots. Hence, chatbots are significantly contributing to improving health services and reducing the healthcare system operating costs. We need to constantly evaluate to ensure reliable operation, by keeping the trust between patients and medical staff.

8.2 Future Work

MD Canada Emoti-bot framework was mainly designed to collect health information from patients before onsite consultations. Currently, the functional and quality testing of Emoti-bot is completed successfully. The next step is to perform effectiveness testing. In addition, the next goal is building Content Management System(CMS) for chatbots. With that, we can dynamically create chatbots, dialogue contexts and services (customer services etc.) by providing a user-friendly UI and decision tree support. Users' only need to focus on creating their context content via the decision tree and no technical expertise is required. This is not limited to medical chatbots but generic chatbot services as well.

The existing research and developments on emotionally smart chatbots are expanded to interdisciplinary research, combining research from artificial intelligence, computational linguistics, and mental health. Deep Learning algorithms for emotion classification and response generation requires a large amount of labelled training data. Currently available open-source data sets are unbalanced and chatbot models trained on such datasets will affect model performance on unseen data. Open-source datasets available from social media has been effective sometimes, but it too suffers from unbalanced distribution of emotion classes. So, creating more human-like labelled data sets to train more effective deep learning chatbot models is the future to precise emotion detection.

In order to design an effective empathetic chatbot framework, we need to consider the impacts of all the stages of chatbots.

1. Emotions expressed by user through user utterance
2. Emotion classified by deep learning algorithms
3. Empathetic response generation back to the user.

Our experiments indicate that deep learning models can be trained to identify emotions in chatbots at the sentence level. As a future step we need to implement advanced transformer-based models with attention mechanisms like BERT, GPT, Roberta, Encoder-Decoder for empathetic response generation based on the emotion identified. There are some open-source data sets available for generating the contextually coherent responses.

A new benchmark dataset called "empathetic dialogues" is described in [46]. The data set consists of more than 25k conversations. Experimental results suggest that dialogue models trained with the following dataset are more real in comparison with models trained on data from the internet. DialoGPT: a large scale generative pre-training set for conversational response generation is presented in [59]. The data model is trained on more than 147m dialogue pairs from the Reddit discussion thread.

Thus by considering a full end to end chatbot experience, from emotion classification to empathetic response generation chatbots can be enhanced to be more natural and empathic in conversations.

REFERENCES

[1] AWS Comprehend Medical API. https://aws.amazon.com/comprehend/medical/. Accessed:2020-06-20.

[2] Azure Text Analytics. https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/#overview. Accessed:2020-07-20.

[3] Basics of Microsoft Bot Services. https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-basics?view=azure-bot-service-4.0. Accessed:2020-06-20.

[4] Build Conversational AI with AWS Lex. https://aws.amazon.com/lex/. Accessed:2020-06-20.

[5] Chatbot Statistics. https://www.smallbizgenius.net/by-the-numbers/chatbot-statistics/#gref. Accessed:2020-06-05.

[6] Google Dialogflow Service. https://cloud.google.com/dialogflow. Accessed:2021-06-20.

[7] Healthcare Chatbots Market Outlook 2026. https://www.alliedmarketresearch.com/healthcare-chatbots-market. Accessed:2021-09-20.

[8] Introduction to Microsoft Bot Framework Composer. https://docs.microsoft.com/en-us/composer/introduction?tabs=v2x/. Accessed:2020-09-28.

[9] Providence Chatbots. https://customers.microsoft.com/en-gb/story/1402681562485712847-providence-health-provider-azure-en-united-states. Accessed:2020-06-20.

[10] Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed:2020-09-30.

[11] Viterbi Algorithm. https://en.wikipedia.org/wiki/Viterbi_algorithm. Accessed:2021-05-5.

[12] Waterfall and Component Dialogs. https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-concept-waterfall-dialogs?view=azure-bot-service-4.0. Accessed:2020-09-28.

[13] M. Abdul-Mageed and L. H. Ungar. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In *ACL*, 2017.

[14] E. Adamopoulou and L. Moussiades. Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006, 2020.

[15] N. Alballa and I. Al-Turaiki. Machine learning approaches in covid-19 diagnosis, mortality, and severity risk prediction: A review. *Informatics in Medicine Unlocked*, page 100564, 2021.

[16] F. Amato, S. Marrone, V. Moscato, G. Piantadosi, A. Picariello, and C. Sansone. Chatbots meet ehealth: Automatizing healthcare. In *WAIAH@AI*IA*, 2017.

[17] B. Anthony Jnr. Implications of telehealth and digital care solutions during covid-19 pandemic: A qualitative literature review. *Informatics for Health and Social Care*, 46:68–83, 11 2020.

[18] C. Antony, B. Pariyath, S. Safar, A. Sahil, and A. R. Nair. Emotion recognition-based mental healthcare chat-bots: A survey. In *International Conference on IoT based Control Networks and Intelligent Systems (ICICNIS 2020)*, page 12, January 2021.

[19] S. Ayanouz, B. A. Abdelhakim, and M. Benhmed. A smart chatbot architecture based nlp and machine learning for health care assistance. New York, NY, USA, 2020. Association for Computing Machinery.

[20] M. Balaji and Y. N. Intelligent chatbot model to enhance the emotion detection in social media using bi-directional recurrent neural network. *Journal of Physics: Conference Series*, 1362:012039, 11 2019.

[21] N. Bott, S. Wexler, L. Drury, C. Pollak, V. Wang, K. Scher, and S. Narducci. A protocol-driven, bedside digital conversational agent to support nurse teams and mitigate risks of hospitalization in older adults: Case control pre-post study. *J Med Internet Res*, 21(10):e13440, Oct 2019.

[22] S. Buechel and U. Hahn. EmoBank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 578–585, Valencia, Spain, Apr. 2017. Association for Computational Linguistics.

[23] R. A. Calvo and S. Mac Kim. Emotions in text: dimensional and categorical models. *Computational Intelligence*, 29(3):527–543, 2013.

[24] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[25] B. N. Colby. *Contemporary Sociology*, 18(6):957–958, 1989.

[26] R. Crutzen, G.-J. Y. Peters, S. D. Portugal, E. M. Fisser, and J. J. Grolleman. An artificially intelligent chat agent that answers adolescents' questions related to sex, drugs, and alcohol: An exploratory study. *Journal of Adolescent Health*, 48(5):514–519, 2011.

[27] T. Dalgleish and M. Power. *Handbook of cognition and emotion*. John Wiley & Sons, 2000.

[28] D. Demszky, D. Movshovitz-Attias, J. Ko, A. S. Cowen, G. Nemade, and S. Ravi. Goemotions: A dataset of fine-grained emotions. *CoRR*, abs/2005.00547, 2020.

[29] K. Denecke, A. Abd-alrazaq, and M. Househ. *Artificial Intelligence for Chatbots in Mental Health: Opportunities and Challenges*, pages 115–128. 08 2021.

[30] S. Jang, J.-J. Kim, S.-J. Kim, J. Hong, S. Kim, and E. Kim. Mobile app-based chatbot to deliver cognitive behavioral therapy and psychoeducation for adults with attention deficit: A development and feasibility/usability study. *International Journal of Medical Informatics*, 150:104440, 2021.

[31] D. Kadariya, R. Venkataramanan, H. Y. Yip, M. Kalra, K. Thirunarayan, and A. Sheth. kbot: Knowledge-enabled personalized chatbot for asthma self-management. *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 138–143, 2019.

[32] M. Karna, D. S. Juliet, and R. Joy. Deep learning based text emotion recognition for chatbot applications. In *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, pages 988–993, 2020.

[33] M. Kowalska and M. Wróbel. *Basic Emotions*, pages 1–6. Springer International Publishing, Cham, 2017.

[34] K. Kroenke, R. L. Spitzer, and J. B. Williams. The phq-9: validity of a brief depression severity measure. *Journal of general internal medicine*, 16(9):606–613, 2001.

[35] I. Mak, C.-M. Chu, P. Pan, M. Yiu, and V. Chan. Long-term psychiatric morbidities among sars survivors. *General hospital psychiatry*, 31:318–26, 07 2009.

[36] M. McKillop, B. South, A. Preininger, M. Mason, and G. Jackson. Leveraging conversational technology to answer common covid-19 questions. *Journal of the American Medical Informatics Association*, 28, 12 2020.

[37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[38] A. Miner, L. Laranjo, and A. B. Kocaballi. Chatbots in the fight against the covid-19 pandemic. *npj Digital Medicine*, 3:65, 05 2020.

[39] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko. Semeval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, 04 2018.

[40] K.-J. Oh, D. Lee, B. Ko, and H.-J. Choi. A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence

generation. In *2017 18th IEEE International Conference on Mobile Data Management (MDM)*, pages 371–375, 2017.

[41] E. W. Pamungkas. Emotionally-aware chatbots:A survey. *CoRR*, abs/1906.09774, 2019.

[42] J. Parviainen and J. Rantala. Chatbot breakthrough in the 2020s? an ethical reflection on the trend of automated consultations in health care. *Medicine, Health Care and Philosophy*, 09 2021.

[43] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[44] J. Pereira and O. Díaz. Using health chatbots for behavior change: A mapping study. *Journal of Medical Systems*, 43, 04 2019.

[45] R. Plutchik. Chapter 1 - a general psychoevolutionary theory of emotion. In R. Plutchik and H. Kellerman, editors, *Theories of Emotion*, pages 3–33. Academic Press, 1980.

[46] H. Rashkin, E. M. Smith, M. Li, and Y. Boureau. I know the feeling: Learning to converse with empathy. *CoRR*, abs/1811.00207, 2018.

[47] E. Saravia, H.-C. T. Liu, Y.-H. Huang, J. Wu, and Y.-S. Chen. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

[48] T. Schachner, R. Keller, and F. v Wangenheim. Artificial intelligence-based conversational agents for chronic conditions: Systematic literature review. *J Med Internet Res*, 22(9):e20701, Sep 2020.

[49] K. R. Scherer and H. G. Wallbott. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66 2:310–28, 1994.

[50] H. Shum, X. He, and D. Li. From eliza to xiaoice: Challenges and opportunities with social chatbots. *CoRR*, abs/1801.01957, 2018.

[51] H.-Y. Shum, X. He, and D. Li. From eliza to xiaoice: Challenges and opportunities with social chatbots. *Frontiers of Information Technology and Electronic Engineering*, 19, 01 2018.

[52] S. Siddique and J. Chow. Machine learning in healthcare communication. *Encyclopedia*, 1:220–239, 02 2021.

[53] T. Spring, J. Casas, K. Daher, E. Mugellini, and O. A. Khaled. Empathic response generation in chatbots. In *SwissText*, 2019.

[54] V. Sundareswaran, A. Sarkar, Weltwirtschaftsforum, and M. C. Corporation. *Chatbots RESET: A Framework for Governing Responsible Use of Conversational AI in Healthcare*. World Economic Forum, 2020.

[55] G. N. Vilaza and D. McCashin. Is the automation of digital mental health ethical? applying an ethical framework to chatbots for cognitive behaviour therapy. *Frontiers in Digital Health*, 3:100, 2021.

[56] J. Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, Jan. 1966.

[57] H. Zhang, J. Zheng, et al. The application analysis of medical chatbots and virtual assistant. *The Frontiers of Society, Science and Technology*, 3(2), 2021.

[58] Y. Zhang, R. Jin, and Z.-H. Zhou. Understanding bag-of-words model: A statistical framework. *International Journal of Machine Learning and Cybernetics*, 1:43–52, 12 2010.

[59] Y. Zhang, S. Sun, M. Galley, Y. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. *CoRR*, abs/1911.00536, 2019.

[60] L. Zhou, J. Gao, D. Li, and H. Shum. The design and implementation of xiaoice, an empathetic social chatbot. *CoRR*, abs/1812.08989, 2018.