ASD BOT: INTELLIGENT AI CHATBOT FOR AUTISM SPECTRUM DISORDER

A PROJECT REPORT BY

GROUP 10

APAR GARG (A0231539E)
GOPAN RAVIKUMAR GIRIJA (A0231541U)
YEONG WEE PING (A0231533R)

SUBMITTED TO

INSTITUTE OF SYSTEMS SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

21 LOWER KENT RIDGE RD, SINGAPORE 119077

in

Partial fulfillment of the requirements
for the degree of

MASTER OF TECHNOLOGY

NOVEMBER 2021

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Background

Pervasive Developmental Disorders (PDD) refers to a group of disorders characterized by delays in the development of socialization and communication skills. In 2013, researchers found that diagnosis of PDD which include –

- Autism
- Asperger Syndrome
- Childhood Disintegrative Disorder
- Rett Syndrome
- Not Otherwise Specified

wasn't consistent across clinics since they tend to have very similar signs and symptoms. So, they replaced all these medical terms with **Autism Spectrum Disorder (ASD)** [1] which encompasses all the PDD but uses a scale/spectrum that differentiates based on the severity of Social Communication deficits and Restrictive and Repetitive behavior/interests/activities. It's thought that using this scale of symptoms as opposed to differentiating between PDD will help in giving a more accurate and useful diagnosis. For example, those with what was previously described as Asperger Syndrome would now likely fall under severity level 1 (low) than in level 3 (high).

Approximately 1.8% of children in the U.S. have a diagnosis of ASD, a rate that has more than doubled over the past two decades. In 2021, people of all races and socioeconomic groups are impacted by the disorder. Worldwide, roughly 1 in 160 people is thought to have ASD [2]. The condition begins early in childhood and lasts throughout a person's life.

1.2. Business Case

Parents of children with atypical development can feel quite stressed and confused when the child is in the nascent stage. They may have a lot of questions in their mind regarding symptoms, screening, and treatment options, etc. Similarly, when faced with the diagnosis of ASD, parents can easily be overwhelmed by the news and may have a lot of questions regarding causes, and child's future, etc. In such difficult times, finding answers on search engines like Google, despite them being the de-facto places to get answers to all queries, can be very frustrating since the information is spread across different web pages. No single web source can provide the answers to all queries. In addition, with misinformation being rampant these days, checking the authenticity of information on web pages is next to impossible.

1.3. Solution

In this project, we have introduced a medical domain-specific chatbot system, which can answer frequently asked queries related to ASD by gathering and summarizing information from various authentic web sources. Moreover, it provides other useful features including the 'M-CHAT Screening test' and 'Special Needs Learning Facilities Nearby' to aid parents of autistic children. We have leveraged the natural language processing capabilities provided by Google's Dialogflow service to develop the chatbot and have later deployed it on Google Assistant, one of the most popular and widely used messaging services, to provide easy and quick access to users.

2. TOOLS AND TECHNOLOGIES USED

2.1. Google Dialogflow

Google Dialogflow is a natural language understanding platform that makes it easy to design and integrate a conversational user interface into your mobile app, web application, device, bot, interactive voice response system, and so on. Using Dialogflow, you can provide new and engaging ways for users to interact with your product.

Dialogflow can analyze multiple types of input from your customers, including text or audio inputs (like from a phone or voice recording). It can also respond to your customers in a couple of ways, either through text or with synthetic speech. The major components present in Dialogflow are Agents, Intents, Context, and Fulfilment [12].

2.1.1. Agent

A Dialogflow agent is a virtual agent that handles conversations with your end-users. It is a natural language understanding module that understands the nuances of human language. Dialogflow translates end-user text or audio during a conversation to structured data that your apps and services can understand.

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

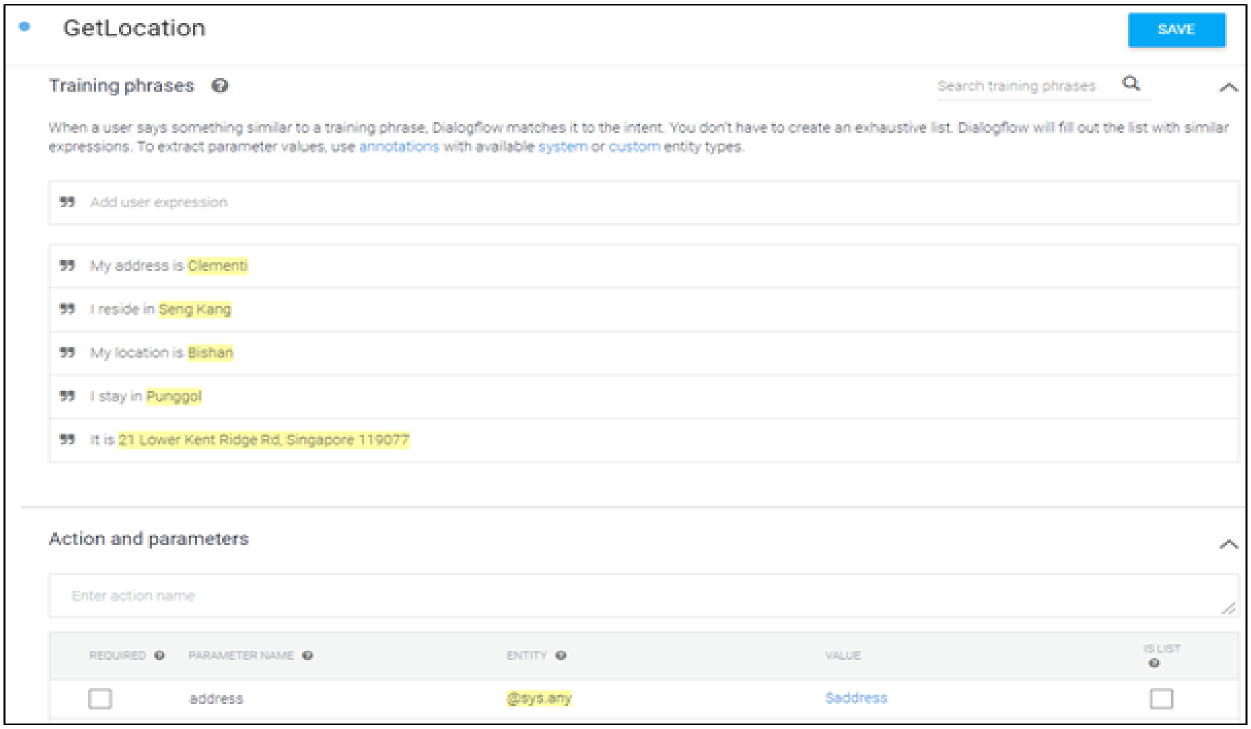
2.1.2. Intent

An intent categorizes an end-user's intention for one conversation turn. For each agent, you define many intents, where your combined intents can handle a complete conversation. When an end-user writes or says something, referred to as an end-user expression, Dialogflow matches the end-user expression to the best intent in your agent. Matching an intent is also known as intent classification.

For example, if the user wants to make queries related to ASD, then the Start Frequently Asked Question (FAQ) intent present in the agent is triggered and the corresponding response will be displayed. This process is called intent classification.

A basic intent contains the following:

- **Training phrases:** These are example phrases for what end-users might say. When an end-user expression resembles one of these phrases, Dialogflow matches the intent. You don't have to define every possible example, because Dialogflow's built-in machine learning expands on your list with other, similar phrases.
- **Action:** You can define an action for each intent. When an intent is matched, Dialogflow provides the action to your system, and you can use the action to trigger certain actions defined in your system.
- **Parameters:** When an intent is matched at runtime, Dialogflow provides the extracted values from the end-user expression as parameters. Each parameter has a type, called the entity type, which dictates exactly how the data is extracted. Unlike raw end-user input, parameters are structured data that can easily be used to perform some logic or generate responses.
- **Responses:** You define text, speech, or visual responses to return to the end-user. These may provide the end-user with answers, ask the end-user for more information, or terminate the conversation.



GetLocation

SAVE

Training phrases

Search training phrases

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use annotations with available system or custom entity types.

Add user expression

My address is Clementi

I reside in Seng Kang

My location is Bishan

I stay in Punggol

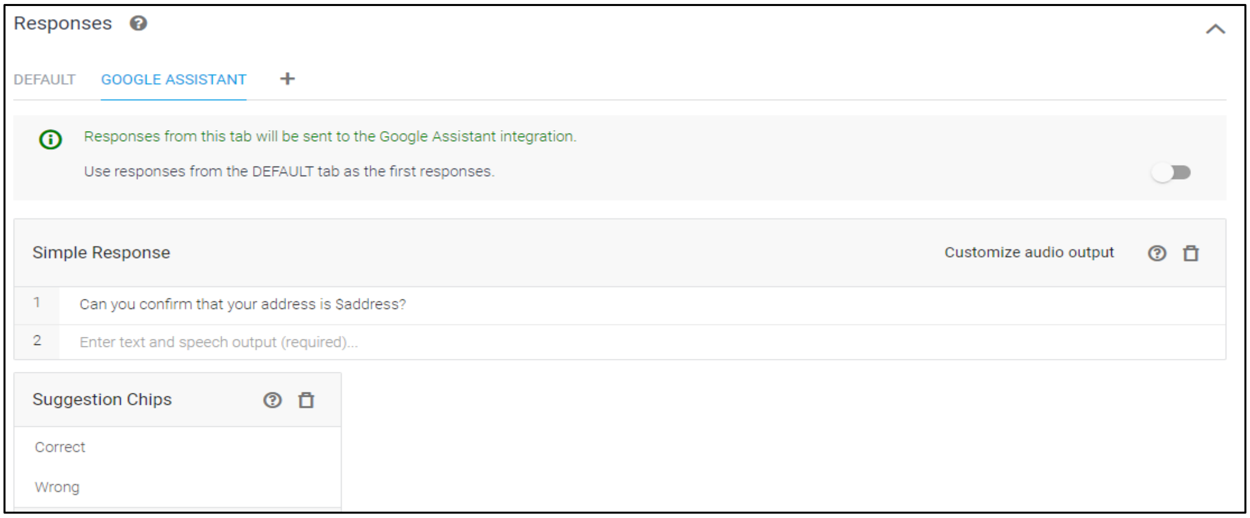
It is 21 Lower Kent Ridge Rd, Singapore 119077

Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
□	address	@sys.any	$address	□

Figure 1: Example of Intent, Training Phrases, Action and parameters



Responses from this tab will be sent to the Google Assistant integration.

Use responses from the DEFAULT tab as the first responses.

Simple Response — Customize audio output

1 Can you confirm that your address is $address?

2 Enter text and speech output (required)...

Suggestion Chips

Correct

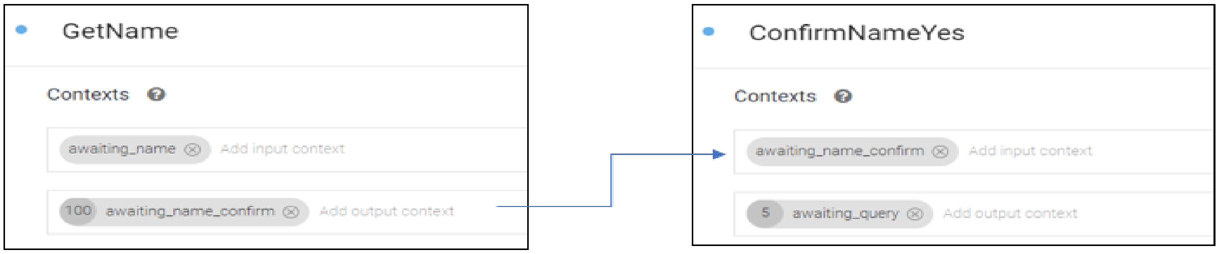
Wrong

Figure 2: Example of Responses in Intent

2.1.3. Context

Dialogflow contexts are similar to natural language contexts. If a person says to you "they are orange", you need context to understand what "they" is referring to. Similarly, for Dialogflow to handle an end-user expression like that, it needs to be provided with a context to correctly match an intent.

Using contexts, you can control the flow of a conversation. You can configure contexts for intent by setting input and output contexts, which are identified by string names. When an intent is matched, any configured output contexts for that intent become active. While any contexts are active, Dialogflow is more likely to match intents that are configured with input contexts that correspond to the currently active contexts.



GetName

Contexts

Input context: awaiting_name — Add input context

Output context: 100 awaiting_name_confirm — Add output context

ConfirmNameYes

Contexts

Input context: awaiting_name_confirm — Add input context

Output context: 5 awaiting_query — Add output context

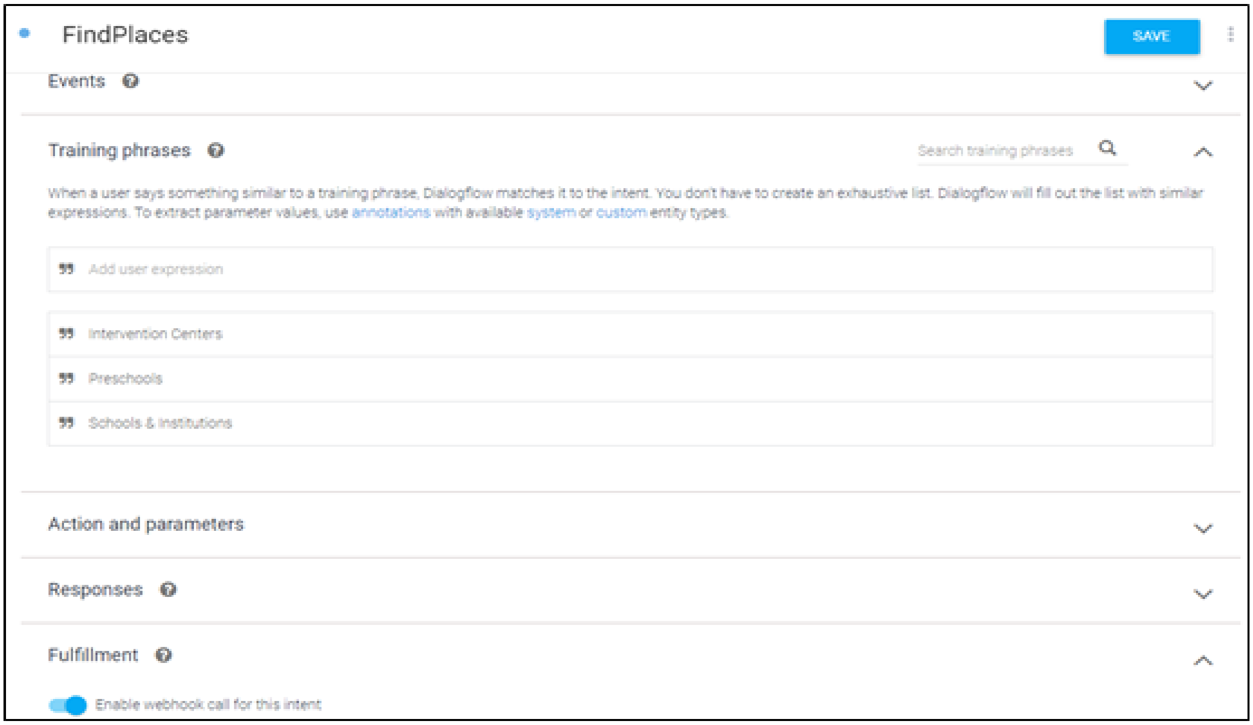
Figure 3: Example to show how 2 Intents connect via Input and Output Context

2.1.4. Fulfillment

By default, your agent responds to a matched intent with a static response. If you are using one of the integration options, you can provide a more dynamic response by using fulfillment. When you enable fulfillment for an intent, Dialogflow responds to that intent by calling a service that you define. For example, if a user wants to look for a nearby school or institution, the user will provide the current location and your service will return the name and distance of the search result.

Each intent has a setting to enable fulfillment. If an intent requires some action by your system or a dynamic response, you should enable fulfillment for the intent. If an intent without fulfillment enabled is matched, Dialogflow uses the static response you defined for the intent.

When an intent with fulfillment enabled is matched, Dialogflow sends a request to your webhook service with information about the matched intent. Your system can perform any required actions and respond to Dialogflow with information on how to proceed.



FindPlaces

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use annotations with available system or custom entity types.

Add user expression

- Intervention Centers
- Preschools
- Schools & Institutions

● Enable webhook call for this intent

Figure 4: Example of Intent which calls Webhook

2.2. Google Assistant

It is a virtual assistant developed by Google and it is available on mobile platforms and smart home devices. It interacts with the user in the form of natural voice, textual input, and visual images. The main functionalities included are searching the internet, scheduling events and alarms, adjusting hardware settings on the user’s device, and showing information from the user’s Google account [10].

2.3. Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics developed by Guido van Rossum. It was originally released in 1991. Python has a reputation as a beginner-friendly language, replacing Java as the most widely used introductory language because it handles much of the complexity for the user, allowing beginners to focus on fully grasping programming concepts rather than minute details [9].

2.4. NGROK

NGROK is a web tunneling tool that can be used for testing webhooks and local servers [11].

2.5. Application Programming Interface (API)

API enables companies to open up their application(s) data and functionalities securely to developers. Developers can easily leverage data and functionalities through a documented interface [3].

Let's consider that we as Customer are sitting in a restaurant for dinner. The Waiter takes our Order and asks the Chef who is in the kitchen to prepare Food for us. After preparing the Food, Chef gives the Food to the Waiter and he forwards it to us.

This is exactly how an API works. We can say that Client/Front-end Application (Customer) sends Request (Order) to API (Waiter). API takes that Request and forwards it to the Backend Server (Chef). Then, the Server takes action according to Request and sends the Response (Food) to API. In the end, API forwards the Response to the Client/Front-end Application.

2.6. Flask

Flask is a framework of Python that allows us to build up web applications. Flask is based on WSGI (Web Server Gateway Interface) toolkit and Jinja2 template engine. [13].

2.7. Heroku

Heroku is an elegant, flexible, and easy-to-use container-based cloud Platform as a Service (PaaS). Developers can use it to deploy, manage, and scale modern apps [4].

3. PROPOSED METHODOLOGY

3.1. Workflow

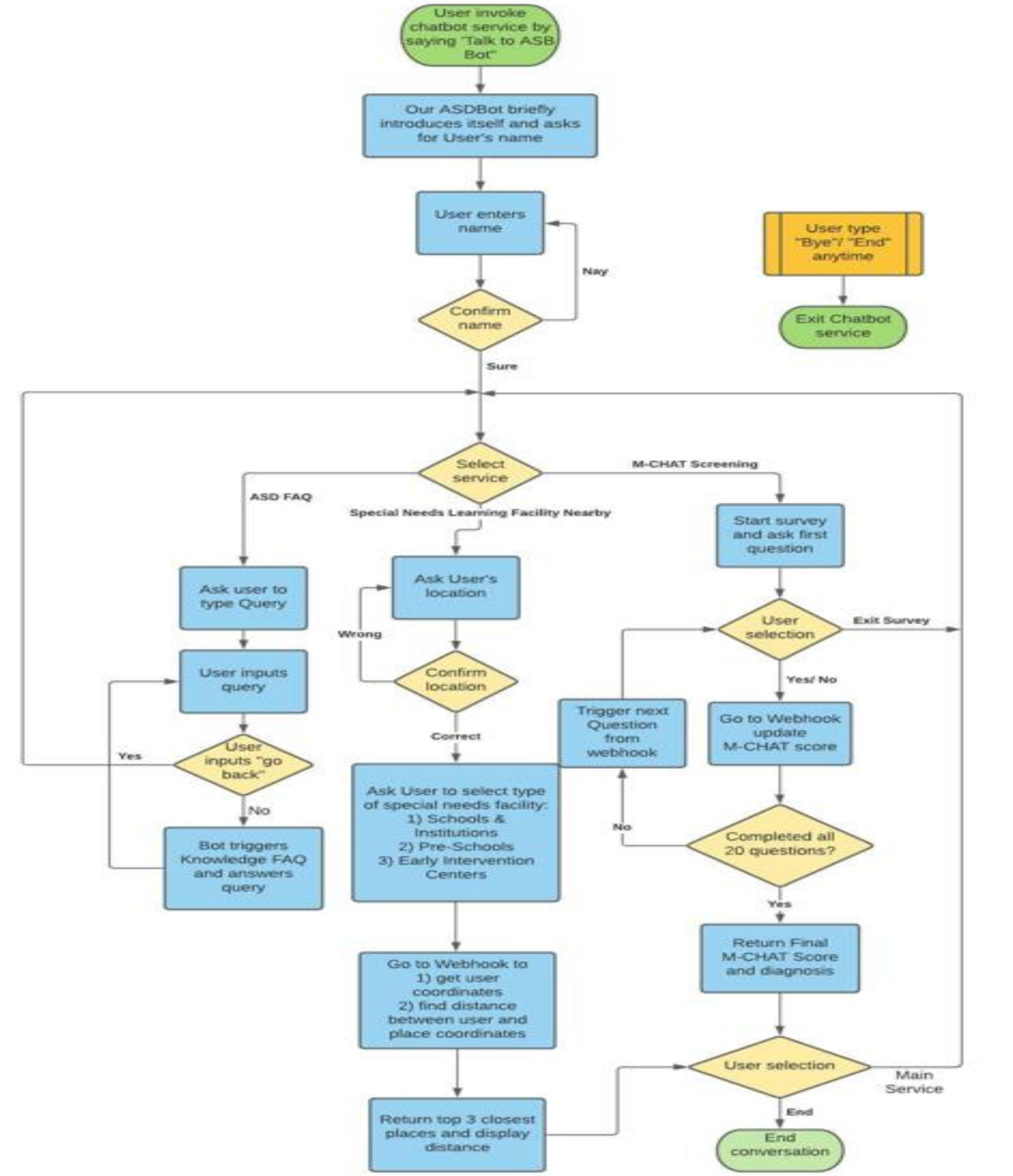

User invoke chatbot service by saying "Talk to ASB Bot"
Our ASDBot briefly introduces itself and asks for User's name
User enters name
Confirm name
Nay
Sure
User type "Bye"/ "End" anytime
Exit Chatbot service
Select service
ASD FAQ
Special Needs Learning Facility Nearby
M-CHAT Screening
Ask user to type Query
User inputs query
User inputs "go back"
Yes
No
Bot triggers Knowledge FAQ and answers query
Ask User's location
Wrong
Confirm location
Correct
Ask User to select type of special needs facility:
1) Schools & Institutions
2) Pre-Schools
3) Early Intervention Centers
Go to Webhook to
1) get user coordinates
2) find distance between user and place coordinates
Return top 3 closest places and display distance
Start survey and ask first question
User selection
Exit Survey
Yes/ No
Go to Webhook update M-CHAT score
Trigger next Question from webhook
No
Completed all 20 questions?
Yes
Return Final M-CHAT Score and diagnosis
User selection
Main Service
End
End conversation

Figure 5: ASD Bot flowchart

3.2. Knowledge Base FAQ

A Knowledge Base represents a collection of knowledge documents that we provide to Dialogflow. These knowledge documents contain information that is useful during conversations with end-users, particularly when a user makes queries on ASD-related questions and our ASD Bot provides answers in the form of an FAQ.

3.2.1. Web Scraping

To build our Knowledge FAQ related to ASD, our team identified more than 30 online websites from different countries and crawled their FAQ sections. To achieve this, we needed to explore a few types of python packages for parsing the pages of websites. Beautiful Soup and Selenium were selected as our choice for doing this scraping task, due to their ease of use and simplicity.

We split this task equally between 3 of our team members and then collated all our results to come up with the merged raw FAQ pairs in the form of a single .csv file. During the scraping process as we build up our Knowledge FAQ, we also utilized Jaccard Similarity in our python code to assign scores to questions that are similar to our existing list so that any new answers can be appended together.

3.2.2. Data Preparation

We gathered a total of 250 questions with 450 unique responses. Then these were organized and curated into a large dataset so that our ASD Bot can make good use of this wealth of knowledge to provide a cognitive response to the end-user in the form of a chatbot FAQ.

First, we utilized the keyword classification technique in python code and grouped these into 3 general categories:

1. Causes and Symptoms
2. Treatment and diagnosis
3. Family and relationship

Then we split this equally and carried out curation on each category to make our FAQ pairs simple and easy to understand. We did this by grouping similar types of questions and then extracting the important points from the list of answers to form a concise answer. The relevant website URL was also included at the end of the answer text so that users can find out more by visiting the website if they wish. FAQ pairs that are out of local context were also removed. An example of an unrelated contextual question will be “What is Medicare?” (which is in the USA).

After completing this, the final version of our knowledge base consisted of a total of 246 FAQ pairs.

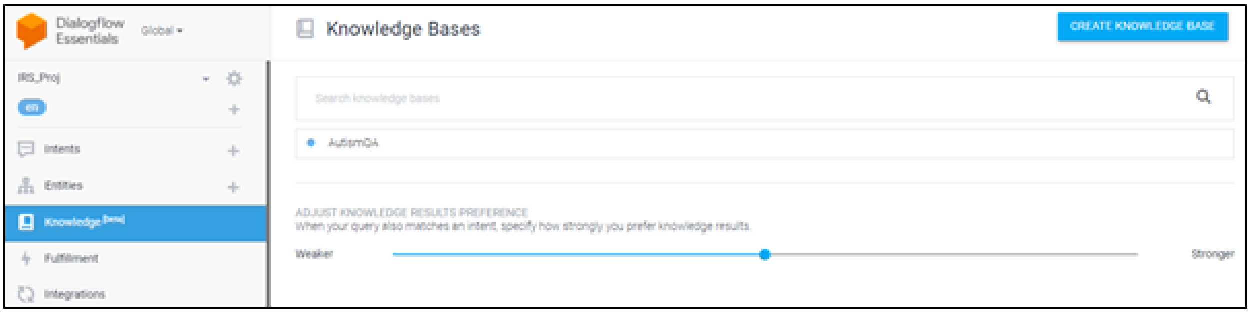
3.2.3. Upload

Once our curated knowledge base consisting of 246 FAQ pairs was ready, we uploaded these into Dialogflow's Knowledge Base which is currently a Beta feature.

The following knowledge document types are supported by Dialogflow:

1. **FAQ:** The document content contains the question and answer pairs as either HTML or CSV. Typical FAQ HTML formats are parsed accurately, but unusual formats may fail to be parsed. CSV must have questions in the first column and answers in the second, with no header. Because of this explicit format, they are always parsed accurately.
2. **Extractive QA:** Documents for which unstructured text is extracted and used for question answering.

Since the bulk of our work was done on scraping the FAQ sections from multiple websites and then curating the information gathered to form our knowledge FAQ, the first type was suitable for our needs.



Knowledge Bases

AutismQA

ADJUST KNOWLEDGE RESULTS PREFERENCE

When your query also matches an intent, specify how strongly you prefer knowledge results.

Figure 6: Beta feature of Dialogflow Knowledge Base which supports FAQ format

3.3. Fulfillment

3.3.1. Special Needs Learning Facilities Nearby

Singapore's education system is often regarded as one of the best in the world. Nevertheless, it's not just about mainstream education. Singapore has several inclusive schools, early intervention centers, and special needs schools to educate differently-abled children.

One of the features of our chatbot is to help parents locate the nearest learning facilities for their child in Singapore from any specified location. Learning Facilities include:

- Special Needs Schools and Institutions
- Special Needs Preschools
- Early Intervention Centers

The process followed to implement this functionality in python is mentioned below:

Database Creation

1. Manually get the names of all learning facilities in Singapore from [5]
2. Extract the precise address of each learning facility from Google Search using Web Scraping.
3. Get geocoordinates for each learning facility address using OpenCage Geocoding API [6]. The API provides forward (text to latitude/longitude) geocoding via a RESTful API.
4. Store (name, geocoordinates) pair for all learning facilities.

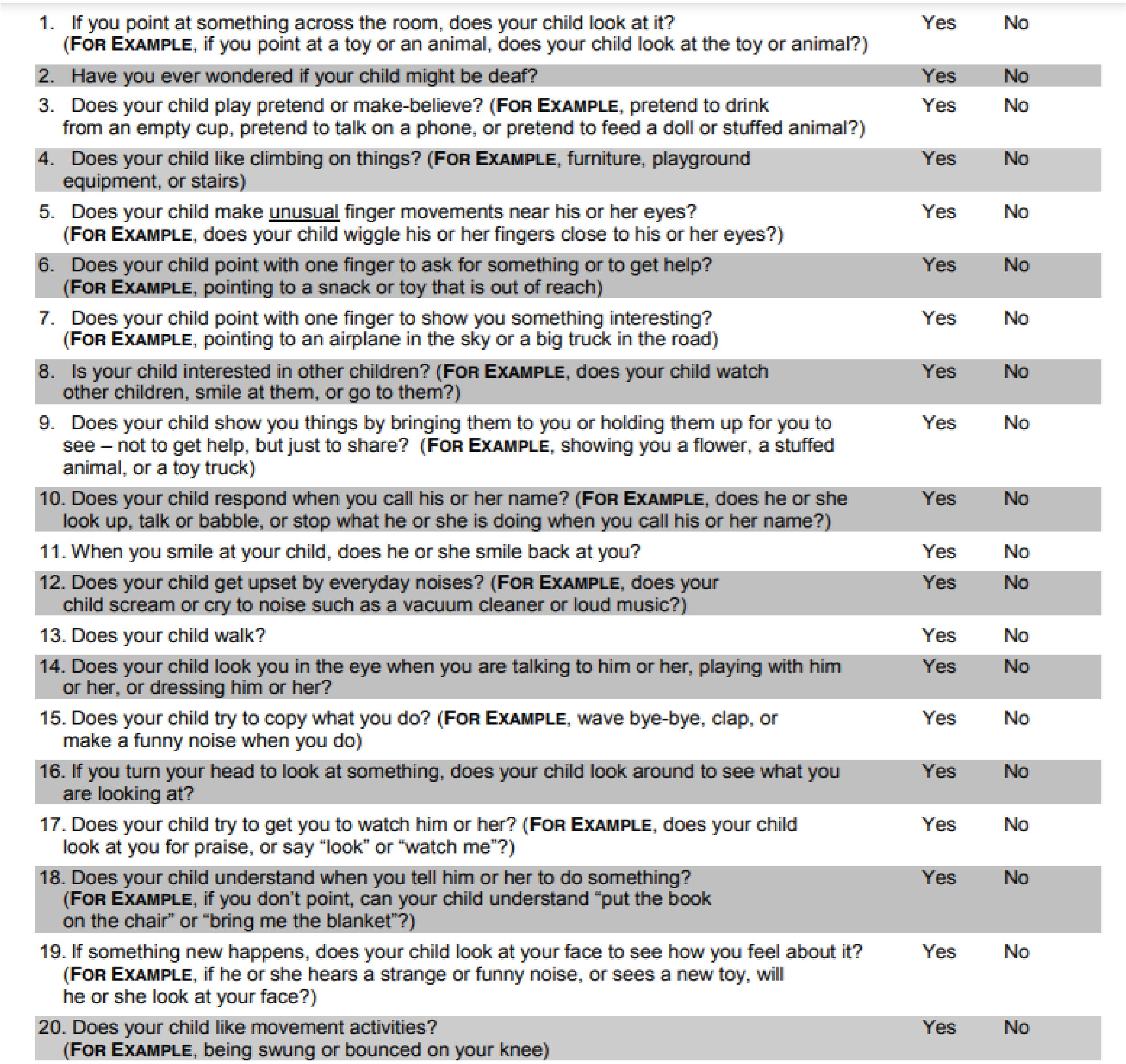
Fulfill user request

1. Get geocoordinates for the given user address using OpenCage Geocoding API.
2. Calculate orthodromic (shortest) distance between user coordinates and each learning facility coordinates using the Haversine formula [7].
3. Filter out locations that are more than 50 km away from the user's location.
4. Get nearest 3 locations from the user's location.

3.3.2. M-CHAT: Autism Screening

The M-CHAT (Modified Checklist for Autism in Toddlers) is a parent-report screening tool to assess the risk for ASD for a child in the age group 16 to 30 months. The M-CHAT screening test consists of a Questionnaire with 20 yes/no questions. Based on the answers provided by the parent we calculate an M-CHAT score and based on this score we categorize the risk factor into 3 (low-risk, medium-risk, high-risk). Based on this score the parent can plan their future steps [8].

Our M-CHAT ASD screening tool is only meant to be the first step in a diagnostic process. A high-risk score by itself does not mean your child will be diagnosed with ASD. The primary goal of M-CHAT is to detect as many cases of ASD as possible ('maximizing sensitivity' in scientific parlance), so there will be some false positives: cases where a child is assessed as being 'at-risk' but in fact will not be diagnosed with ASD. However, any child that receives a 'high-risk' score should receive an evaluation from an ASD specialist.



1. If you point at something across the room, does your child look at it? (**For Example**, if you point at a toy or an animal, does your child look at the toy or animal?)	Yes	No
2. Have you ever wondered if your child might be deaf?	Yes	No
3. Does your child play pretend or make-believe? (**For Example**, pretend to drink from an empty cup, pretend to talk on a phone, or pretend to feed a doll or stuffed animal?)	Yes	No
4. Does your child like climbing on things? (**For Example**, furniture, playground equipment, or stairs)	Yes	No
5. Does your child make unusual finger movements near his or her eyes? (**For Example**, does your child wiggle his or her fingers close to his or her eyes?)	Yes	No
6. Does your child point with one finger to ask for something or to get help? (**For Example**, pointing to a snack or toy that is out of reach)	Yes	No
7. Does your child point with one finger to show you something interesting? (**For Example**, pointing to an airplane in the sky or a big truck in the road)	Yes	No
8. Is your child interested in other children? (**For Example**, does your child watch other children, smile at them, or go to them?)	Yes	No
9. Does your child show you things by bringing them to you or holding them up for you to see – not to get help, but just to share? (**For Example**, showing you a flower, a stuffed animal, or a toy truck)	Yes	No
10. Does your child respond when you call his or her name? (**For Example**, does he or she look up, talk or babble, or stop what he or she is doing when you call his or her name?)	Yes	No
11. When you smile at your child, does he or she smile back at you?	Yes	No
12. Does your child get upset by everyday noises? (**For Example**, does your child scream or cry to noise such as a vacuum cleaner or loud music?)	Yes	No
13. Does your child walk?	Yes	No
14. Does your child look you in the eye when you are talking to him or her, playing with him or her, or dressing him or her?	Yes	No
15. Does your child try to copy what you do? (**For Example**, wave bye-bye, clap, or make a funny noise when you do)	Yes	No
16. If you turn your head to look at something, does your child look around to see what you are looking at?	Yes	No
17. Does your child try to get you to watch him or her? (**For Example**, does your child look at you for praise, or say “look” or “watch me”?)	Yes	No
18. Does your child understand when you tell him or her to do something? (**For Example**, if you don’t point, can your child understand “put the book on the chair” or “bring me the blanket”?)	Yes	No
19. If something new happens, does your child look at your face to see how you feel about it? (**For Example**, if he or she hears a strange or funny noise, or sees a new toy, will he or she look at your face?)	Yes	No
20. Does your child like movement activities? (**For Example**, being swung or bounced on your knee)	Yes	No

Figure 7: M-CHAT ASD Screening Questions

Scoring Mechanism

For all questions except 2, 5, and 12, the response “NO” indicates ASD risk; for items 2, 5, and 12, “YES” indicates ASD risk.

- **Low-Risk Category:** Final M-CHAT score will be in the range of 0-2
- **Medium Risk Category:** Final M-CHAT score will be in the range of 3-7
- **High-Risk Category:** Final M-CHAT score will be in the range 8 or above

M-CHAT Python Webhook

When the proper intent is matched. The function of the python webhook is to trigger an event that will trigger the next question. In addition to that, based on the answers provided by the user the python webhook will also keep track of the current score and once the user finishes answering all questions the python function will categorize the child into high-risk, medium-risk, or low-risk based on the final score and then it will send a response back to the user about the future steps to be considered based on the score.

3.4. Workflow Integration and Testing

Once all the features - Knowledge FAQ and Fulfilments; were completed, the next important step was to do the integration and ensure these separate features work in our ASD Bot. One important step here was to utilize the Input and Output context between Intents and design the flow so that our ASD Bot functions smoothly and properly.

Extensive testing to cover corner cases was then done to make sure the pipeline (Figure 5) works.

4. CONCLUSION AND FUTURE WORK

In this work, we have developed a medical domain-specific chatbot system called ASD Bot. The ASD Bot is capable of answering queries related to Autism Spectrum Disorder (ASD). In addition to that, it also provides services like 'M-CHAT Screening test' and 'Special Needs Learning Facilities Nearby'. This report provides a detailed explanation of all these features and includes the design and implementation steps.

Due to time and resource constraints, there are a few limitations to this chatbot. A cognitive system requires many training phrases to be interactive. Due to the limited amount of training phrases and entities available, not everything could be matched to the real-life scenario. By introducing more training phases, we can improve the quality of interaction between chatbot and user. Currently, ASD Bot provides only two additional features apart from answering FAQ questions. In addition to this, including features like hospital, appointment booking will help the parents to a great extent. Currently, the 'Special Needs Learning Facilities Nearby' works only in Singapore, adding information about facilities from other countries to the knowledge base would help in improving the functionality of this feature.

REFERENCES

1. https://www.cdc.gov/ncbddd/autism/facts.html
2. https://www.elemy.com/studio/autism/statistics-and-rates
3. https://www.ibm.com/cloud/learn/api
4. https://www.heroku.com/about
5. https://www.sassymamasg.com/learn-special-needs-school-singapore-preschools-early-intervention-therapy-centres-international-schools/
6. https://opencagedata.com/
7. https://www.geeksforgeeks.org/program-distance-two-points-earth/
8. https://www.cpqcc.org/sites/default/files/M-CHAT-R_F_1.pdf
9. https://www.teradata.com/Glossary/What-is-Python
10. https://en.wikipedia.org/wiki/Google_Assistant
11. https://ngrok.com/
12. https://cloud.google.com/dialogflow/docs
13. https://www.geeksforgeeks.org/python-introduction-to-web-development-using-flask/

APPENDIX OF REPORT A

Project Proposal

GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)
PRACTICE MODULE: Project Proposal

Date of proposal:

7 November 2021

Project Title:

IRS Project – Intelligent AI Chatbot for Autism Spectrum Disorder

Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)*

Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore
NATIONAL UNIVERSITY OF SINGAPORE (NUS)
Contact: Mr. GU ZHAN / Lecturer & Consultant
Telephone No.: 65-6516 8021
Email: zhan.gu@nus.edu.sg

Background/Aims/Objectives:

Autism Spectrum Disorder (ASD) is often a difficult condition to diagnose. Although there are many online websites and resources that provide information on ASD, it can be confusing and a hassle for anxious parents to navigate all these different resources which are scattered all over the place.

Our proposed intelligent Chatbot aims to provide a one-stop platform to parents with children who are autistic or suspected of having ASD or just want to know more about ASD. Our Chatbot intends to provide the following services:

1. A concise and comprehensive FAQ knowledge base to answer any queries related to ASD
2. M-CHAT screening to help advise the parent if a child is at risk of being diagnosed with ASD
3. Help locate nearby facilities and institutions to help their child

Requirements Overview:

- Research ability:
    - Researched ASD related topics
    - Researched on pros and cons of Google's Dialogflow and Amazon's Lex
- Programming ability:
    - Performed scraping using Beautiful Soup and Selenium on more than 30 websites, and employed Jaccard Similarity to help judge the similarity between questions

    - Performed keyword classification via python code to help categorize the FAQ pairs
    - Fulfillment tasks are done via webhook

- System integration ability:
    - Integrated Dialogflow Knowledge Base and Fulfilments into a cognitive ChatBot
    - Deployed ChatBot into Google assistant

Resource Requirements (please list Hardware, Software and any other resources)

Software proposed for consideration:

- Google's Dialogflow for building our chatbot
- Google's Colab for testing and developing our python codes
- Google Assistant for frontend deployment
- Heroku-flask web framework to host our backend python codes for fulfillment tasks

Number of Learner Interns required: (Please specify their tasks if possible)

Our team consists of 3 members

1. Apar Garg (web scraping, find facilities nearby fulfillment, Heroku deployment, report, and documentation)
2. Gopan Ravikumar Girija (web scraping, M-CHAT screening fulfillment, Google Assistant integration, report and documentation, video making)
3. Yeong Wee Ping (web scraping, Knowledge base FAQ, overall chatbot workflow integration, Google Assistant integration, report, and documentation)

Methods and Standards:

Procedures	**Objective**	**Key Activities**
Requirement Gathering and Analysis	The team should meet with ISS to scope the details of the project and ensure the achievement of business objectives.	1. Gather & Analyze Requirements 2. Define internal and External Design 3. Prioritize & Consolidate Requirements 4. Establish Functional Baseline
Technical Construction	· To develop the source code in accordance with the design. · To perform unit testing to ensure the quality before the components are integrated as a whole project	1. Setup Development Environment 2. Understand the System Context, Design 3. Perform Coding 4. Conduct Unit Testing

Integration Testing and acceptance testing	To ensure interface compatibility and confirm that the integrated system hardware and system software meets requirements and is ready for acceptance testing.	1. Prepare System Test Specifications 2. Prepare for Test Execution 3. Conduct System Integration Testing 4. Evaluate Testing 5. Establish Product Baseline
Acceptance Testing	To obtain ISS user acceptance that the system meets the requirements.	1. Plan for Acceptance Testing 2. Conduct Training for Acceptance Testing 3. Prepare for Acceptance Test Execution 4. ISS Evaluate Testing 5. Obtain Customer Acceptance Sign-off
Delivery	To deploy the system into the production (ISS standalone server) environment.	1. Software must be packed by following ISS's standard 2. Deployment guidelines must be provided in ISS production (ISS standalone server) format 3. Production (ISS standalone server) support and troubleshooting process must be defined.

Team Formation & Registration

Team Name: Group 10
Project Title (repeated): IRS Project – Intelligent AI Chatbot for Autism Spectrum Disorder
System Name (if decided): ASD Bot
Team Member 1 Name: Apar Garg
Team Member 1 Matriculation Number: A0231539E
Team Member 1 Contact (Mobile/Email): e0703571@u.nus.edu
Team Member 2 Name: Gopan Ravikumar Girija
Team Member 2 Matriculation Number: A0231541U
Team Member 2 Contact (Mobile/Email): e0703573@u.nus.edu
Team Member 3 Name: Yeong Wee Ping
Team Member 3 Matriculation Number: A0231533R
Team Member 3 Contact (Mobile/Email): e0703565@u.nus.edu

For ISS Use Only		
Programme Name:	**Project No:**	**Learner Batch:**
Accepted/Rejected/KIV:		
Learners Assigned:		
Advisor Assigned: Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg		

APPENDIX OF REPORT B

Mapped System Functionalities against knowledge, techniques, and skills of modular courses

Modular Courses	System Functionalities / Techniqe Applied
Machine Reasoning (MR)	• **Knowledge Elicitation and extraction**: Web crawling from websites & repositories, manual extraction from websites • **Knowledge Representation**: Knowledge Base to store all the queries and answers related to ASD
Cognitive System (CGS)	• **Cognitive System**: Dialogflow's default Named Entity Recognition, Dialogfow's default NLU, Dialogflow's default Rule-based system • **Intent Training & Entity Definition:** Training phrases for all intents. • **Event:** Triggering next questions in M-CHAT.

APPENDIX OF REPORT C

Installation and User Guide

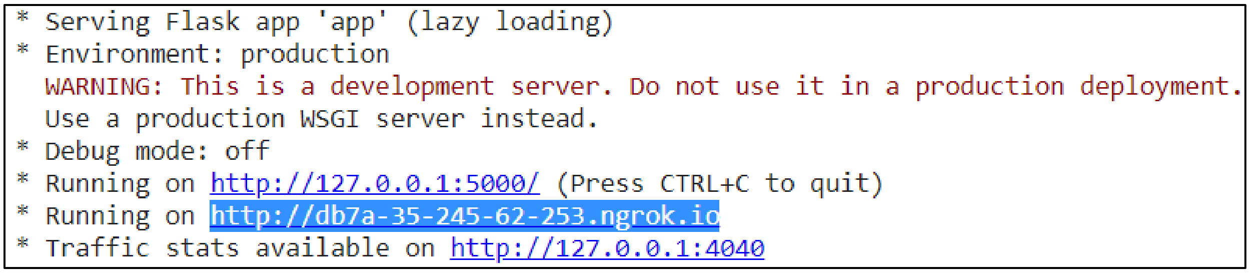
Deployment

Before beginning, download our GitHub repository from https://github.com/AparGarg99/ASD-Bot

Server

We will be using NGROK's local server for webhook integration.

1. Go to Google Colab.
2. Upload and execute Webhook.ipynb present in downloaded repository to Colab.
3. Take note of the temporal forwarding URL, as shown highlighted in the example screenshot below:

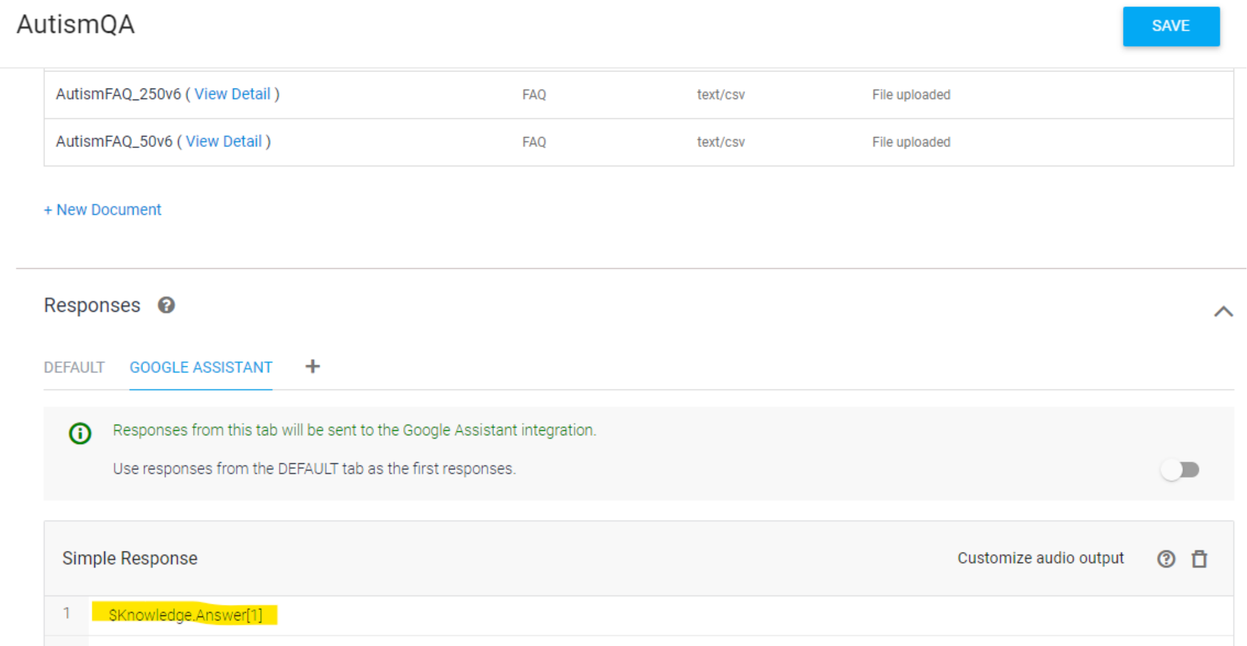


```
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Running on http://db7a-35-245-62-253.ngrok.io
 * Traffic stats available on http://127.0.0.1:4040
```

4. Modify the temporal forwarding URL. Suppose the original URL was http://db7a-35-245-62-253.ngrok.io, then change it to https://db7a-35-245-62-253.ngrok.io/webhook/.
5. Store this new temporal forwarding URL in a notepad file.

Dialogflow

1. Go to Dialogflow at https://dialogflow.com/.
2. Register for a Dialogflow account using a set of Google user credentials.
3. Create a new agent. Once created, click on the Export and Import tab in the settings of your agent and click Import From ZIP.
4. Follow the onscreen instructions in Dialogflow to upload the ASDBot_dialogflow.zip folder present in downloaded repository.
5. Go to the Knowledge[beta] option on the menu access on the left in Dialogflow. Click CREATE KNOWLEDGE BASE at top right of page, name it as AutismQA and start uploading the .csv files containing the FAQ pairs. Keep the default setting as 0 for ADJUST KNOWLEDGE RESULTS PREFERENCE. Next, tick the box next to AutismQA and click ENABLE.


AutismQA

AutismFAQ_250v6 (View Detail)	FAQ	text/csv	File uploaded
AutismFAQ_50v6 (View Detail)	FAQ	text/csv	File uploaded

Responses from this tab will be sent to the Google Assistant integration.

Use responses from the DEFAULT tab as the first responses.

Simple Response

1 $Knowledge.Answer[1]

The last step will be to key in $Knowledge.Answer[1] under the GOOGLE ASSISTANT's Simple Response field

6. Go to the Fulfillment option on the menu access on the left. Ensure the Webhook is enabled.
7. Paste the new temporal forwarding URL in the notepad file in Webhook and click Save.

Get Started

Congratulations, you have successfully deployed ASD Bot!! You can now test the bot via the test console.

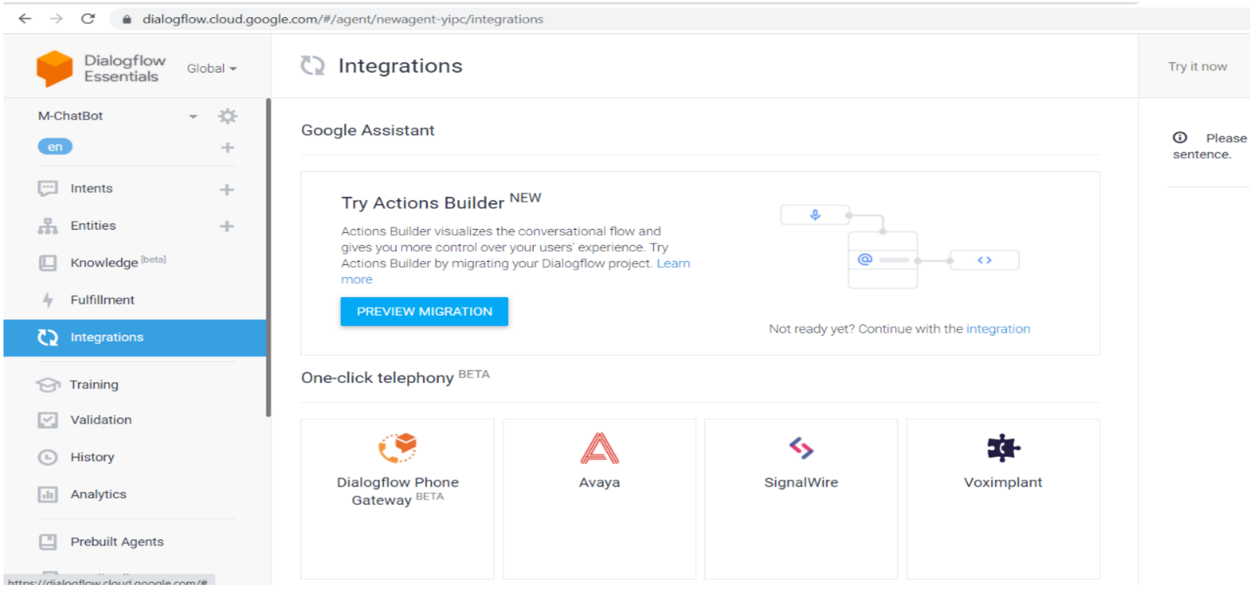
Additional Note

Instead of ngrok, you may consider using other PaaS (Platform as a Service) such as Heroku, AWS, Azure, etc, that better suit your needs.

Google Assistant Integration

Dialogflow provides an option to integrate the developed chatbot to Google Assistant. To integrate our ASD Bot to Google Assistant we followed the below-mentioned steps.

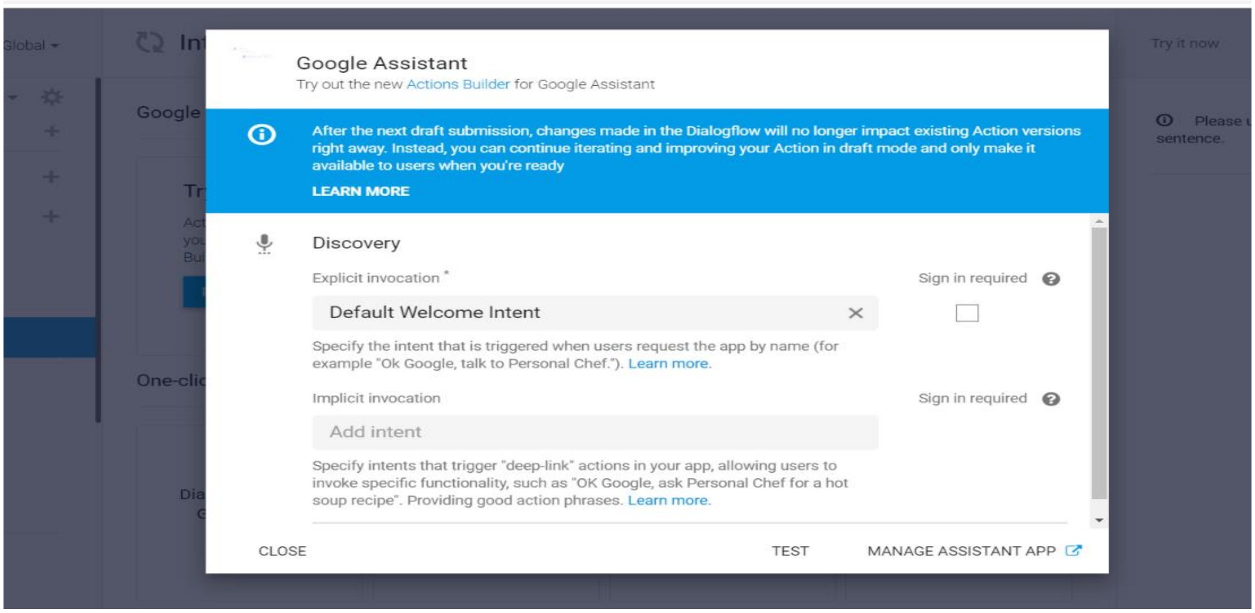
1. To start, choose integration and click on integration present on the right side of the screen.



Actions Builder visualizes the conversational flow and gives you more control over your users' experience. Try Actions Builder by migrating your Dialogflow project. Learn more

Not ready yet? Continue with the integration

2. On clicking, you will see two fields i.e., explicit, and implicit invocation. You can specify your explicit and implicit invocation intents here. We kept Explicit invocation as Default welcome intent. Click on Manage Assistant App.



Try out the new Actions Builder for Google Assistant

After the next draft submission, changes made in the Dialogflow will no longer impact existing Action versions right away. Instead, you can continue iterating and improving your Action in draft mode and only make it available to users when you're ready

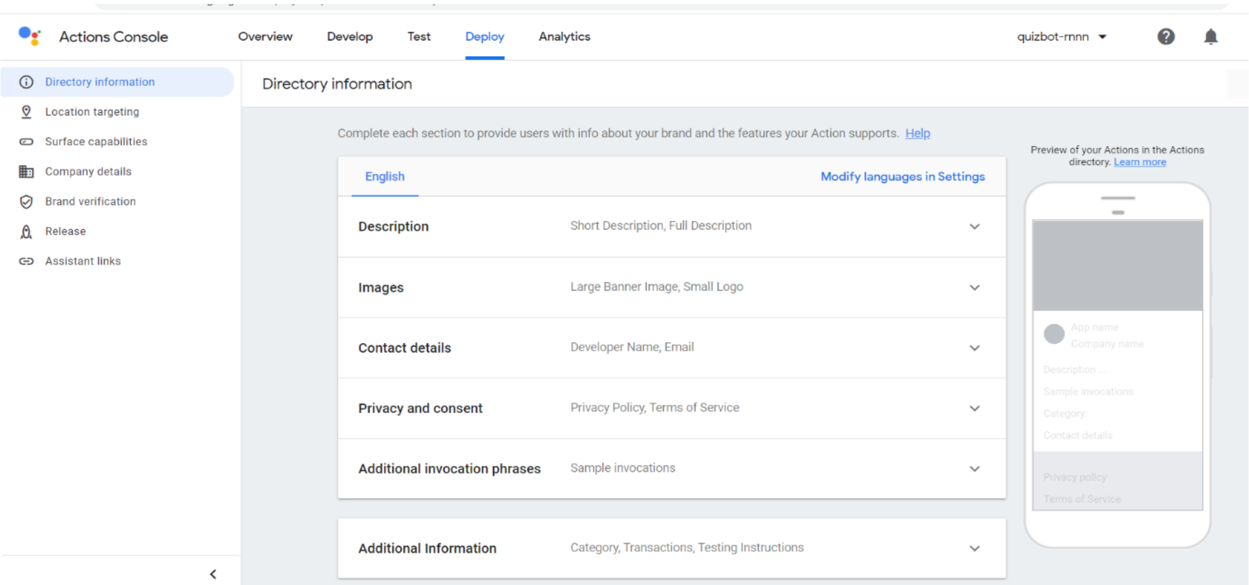
LEARN MORE

Default Welcome Intent

Specify the intent that is triggered when users request the app by name (for example "Ok Google, talk to Personal Chef."). Learn more.

Specify intents that trigger "deep-link" actions in your app, allowing users to invoke specific functionality, such as "OK Google, ask Personal Chef for a hot soup recipe". Providing good action phrases. Learn more.

3. On clicking that you will reach the Deployment stage. In this window, you will be asked to provide a series of information including your bots name, description, privacy policy, etc. After filling in all the required fields click on release present on the left side of the screen.



Directory information

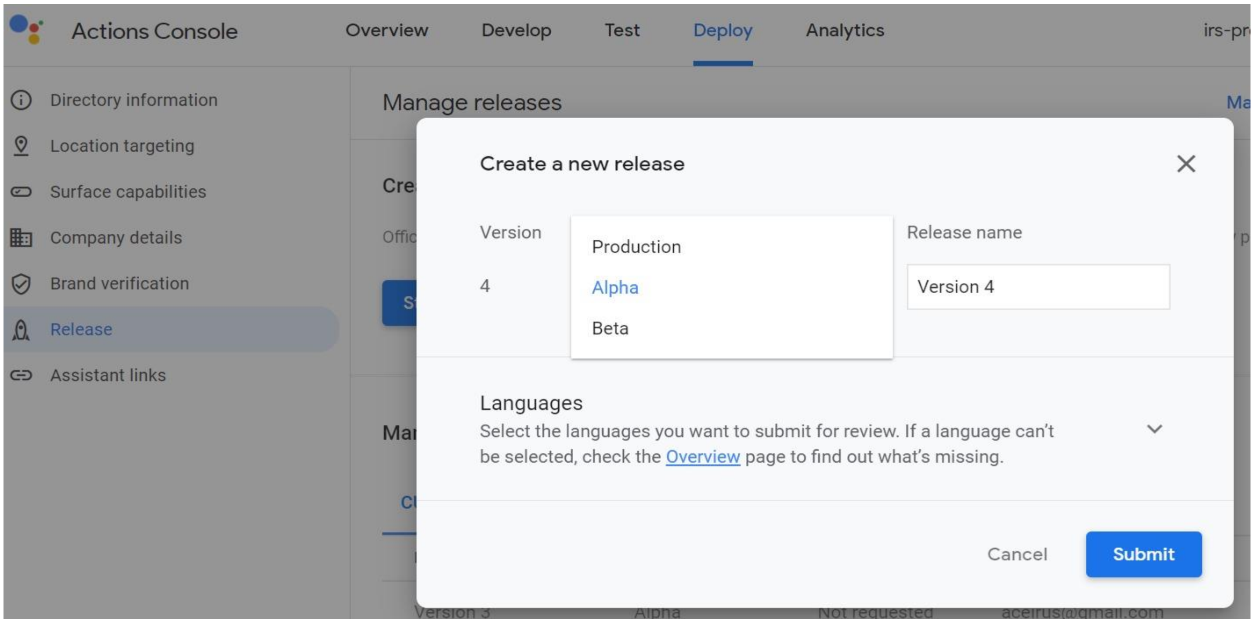
Complete each section to provide users with info about your brand and the features your Action supports. Help

English | Modify languages in Settings

Description	Short Description, Full Description
Images	Large Banner Image, Small Logo
Contact details	Developer Name, Email
Privacy and consent	Privacy Policy, Terms of Service
Additional invocation phrases	Sample invocations
Additional Information	Category, Transactions, Testing Instructions

Preview of your Actions in the Actions directory. Learn more

4. Here you will be asked to choose what type of release are you going to perform. There are 3 options to choose from. The alpha release will allow max up to 20 people to access the bot, the beta release will allow up to 200, and the production release will allow anybody using Google assistant to access it. Choose your preferred option and click submit.



Create a new release

Version: 4

Production
Alpha
Beta

Release name: Version 4

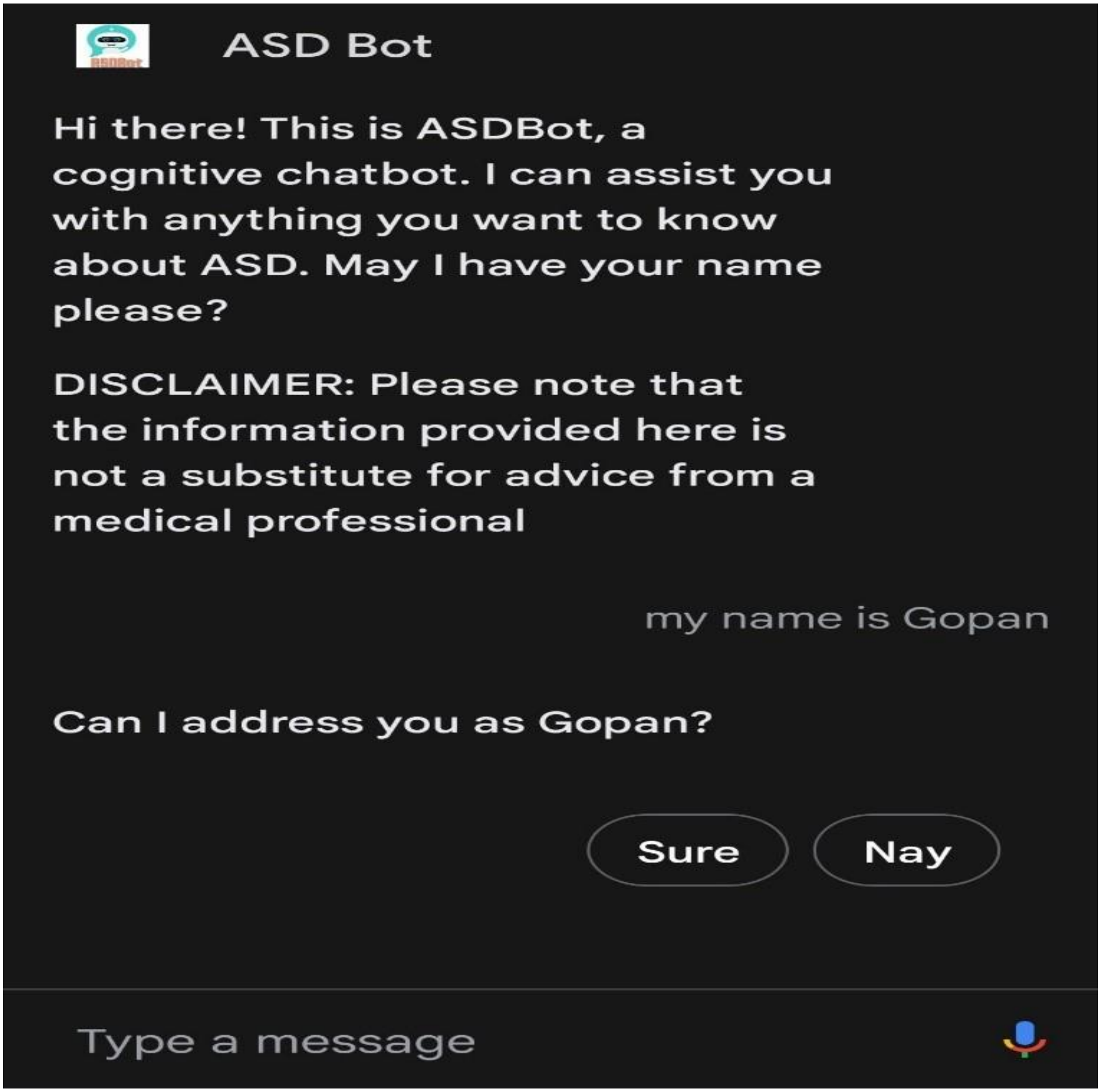
Languages

Select the languages you want to submit for review. If a language can't be selected, check the Overview page to find out what's missing.

For production release Googles testing team will review your bot and if you have followed all their requirements, they will deploy it else they will send us an error report and we need to make changes according to that before giving the next release. The alpha release does not require review from the Google test team and you can immediately access your bot integrated to Google Assistant but only limited people to whom permission was given beforehand can access the bot.

Running ASD Bot on Google Assistant

1. Open Google Assistant on your mobile phone.
2. Type the key phrase "Talk to ASD Bot" on the keyboard. ASD Bot is linked to google assistant and can be activated by uttering its name or typing ("Talk to ASD Bot") in text format. An image of the ASD Bot page integrated into Google assistant is shown below.



ASD Bot

Hi there! This is ASDBot, a cognitive chatbot. I can assist you with anything you want to know about ASD. May I have your name please?

DISCLAIMER: Please note that the information provided here is not a substitute for advice from a medical professional

my name is Gopan

Can I address you as Gopan?

APPENDIX OF REPORT D

Individual Reports

APAR GARG (A0231539E)

Personal Contribution

I have made 5 contributions to the project:

1. Performed Project management, which includes project planning, deciding the scope of our project, scheduling of tasks and events for each team member.
2. Developed web scraping scripts to extract Frequently Asked Questions (FAQ) regarding ASD and their answers from web pages of well-known agencies like the Centers for Disease Control and Prevention (CDC) and Ministry of Health of Singapore (MOH).
3. Implemented 'Special Needs Learning Facilities Nearby' functionality using Python and Flask which could help parents of autistic children find Schools and Institutions, Pre-Schools, and Early Intervention Centers nearest to the given location.
4. Deployed the flask app on the Heroku web server.
5. Carried out extensive integration testing of the final chatbot to ensure that there are no gaps/bugs.

Learning Outcomes

I can broadly classify my learnings under the following two heads:

1. Technical Learnings
   - **Google Dialogflow:**The most useful thing I learned during this project is how to build workable and efficient chatbot systems using Google Dialogflow. I also got to explore the pros and cons of using Dialogflow w.r.t other Conversational AI services like Microsoft Azure Bot and Amazon Lex. Working on Dialogflow has certainly improved my understanding of various terminologies like entities, intents, context, events, and training phrases and also their inter-dependencies.
   - **Heroku:** I learned how to host the source code on Heroku remote server. Multiple other hosting platforms were tried and tested, but Heroku seemed to be the most intuitive platform and easy to use.
   - **Flask:** I learned how to make a web application using flask in python.
   - **API:** While developing the backend python code, I learned how to fetch data from a website to python code using an API service.

2. Non-Technical Learnings
   - I learned different methods to calculate the orthodromic (shortest) distance between two points on earth.
   - Being from North India, I learned how to work and build strong relationships with teammates from other cultures namely South India and Singapore.
   - I learned the fundamentals of Autism Spectrum Disorder (ASD) – causes, symptoms, screening, and treatment options.

Application of gained knowledge and skills

Last year I interned for a company called D Cube Analytics. During the internship, I developed a medical domain-specific knowledge-based QA system, which can answer the queries related to the sales and marketing of drugs from a given database. The QA system provides a real-time response to the common business queries of any employee in the company and hence removes unnecessary dependencies on the analytics team.

The system was developed from scratch using Python and SpaCy. Although the system performed fairly well on small and simple user queries, it didn't work well for long and more contextual queries. It failed to establish dependencies between entities in complicated queries. Moreover, integrating the backend python system with the Slack messaging service was a cumbersome process.

Learning how to create a functional chatbot using Dialogflow and webhook integration has been a fruitful experience. With the knowledge that I have gained in this project, I would like to extend my QA system functionality to handle much more complicated and contextual queries by the employees in the company. With the help of Dialogflow, I will also be able to easily integrate the system with several messaging services like Slack, Google Assistant, and Facebook Messenger. Such a system would bring significant time savings to the analytics team.

Read more about my QA System: https://github.com/AparGarg99/BI-Bot

GOPAN RAVIKUMAR GIRIJA (A0231541U)

Personal Contribution

I have made 6 contributions to the project:

1. Developed web scraping scripts and performed web scraping to extract Frequently Asked Questions (FAQ) regarding ASD and their answers from web pages of well-known agencies like Autism Resource Center (Singapore) and Centers for Disease Control and Prevention. Our overall knowledge base was created from 30 web resources out of which I handled 10.
2. Structured the answers collected into a concise and readable format. Combined multiple answers from multiple websites into a single answer after doing a fact check.
3. Implemented ‘M-CHAT screening test’ webhook functionality using Python and Flask which could help parents of kids to do an initial screening test to decide future actions.
4. Carried out extensive integration testing of the final chatbot to ensure that there are no gaps/bugs.
5. Part of the group that integrated Dialogflow bot with Google Assistant.
6. Prepared project presentation videos. In addition to that, I was responsible for handling multiple sessions in Report.

Learning Outcomes

I got to learn a lot during the project.

1. Technical Learnings

- **Google Dialogflow:** I learned how to build workable and efficient chatbot systems using Google Dialogflow. I got an opportunity to understand various concepts related to Dialogflow like context, events, entities, intents, and training phrases and the relationship between them. In addition to that, I also explored available alternatives for Dialogflow like Amazon Lex and understood the pros and cons of each approach.
- **Flask:** I learned how to make a web application using flask in python. During the development of the M-CHAT webhook, I learned how to control the flow of Dialogflow intents using events triggered from the webhook. In addition to that, I have learned how to pass values between different intents. This knowledge helped me to store M-CHAT score values without the help of a database.
- **Web Scraping:** I got an opportunity to learn how to use python packages like Beautiful Soup to extract relevant information from different web resources.

2. Non-Technical Learnings

- I learned how to generate animation videos while creating project presentation videos.

- I learned a lot about ASD. From the perspective of technical knowledge, I learned many medical and legal terminologies related to autism. In addition to that, I learned a lot about what preparation and requirements are needed to improve the quality of life of people with an ASD. Furthermore, I learned about different diagnosis methods along pros and cons associated with them.
- Being from South India, I learned how to work and build strong relationships with teammates from different age groups and diverse cultures namely North India and Singapore.

Application of gained knowledge and skills

The Indian government is providing many scholarships and financial assistance to students belonging to a different religion, caste, and economic background. But many students are unaware of these scholarships and currently, there is no efficient way to inform people about financial assistance schemes provided by the government. Even if the student knows about these scholarships currently there are no efficient methods to answer their queries related to these scholarships, for example, many students have doubts related to eligibility criteria, where to apply, and what all documents are required to apply for a scholarship.

Learning how to create a functional chatbot using Dialogflow and webhook integration has been a good experience. With the knowledge that I have gained in this project, I would like to develop a QA system that can answer every query related to all scholarship schemes provided by the government. Chatbots can conversationally put them so that students can easily understand all the complicated documentations. In addition to that, the chatbot will also be able to help the student to understand if he/she is eligible for a particular scholarship. Dialogflow has provided functionalities to integrate chatbots to many social media platforms like messenger, google assistant, slack, etc. This will in turn help, students to access my bot from all these platforms.

YEONG WEE PING (A0231533R)

Personal Contribution

In this project, my contributions can be broadly categorized into the following:

1. **Research on ASD:** Consulted a personal friend who works as a professional child psychologist and got to understand the needs of parents concerned about ASD and tried to address such concerns into our system. For example, ensuring the usual questions that are often asked are provided with answers that are concise and easy to understand into our FAQ knowledge base.
2. **Web Scraping:** Performed web scraping using Beautiful Soup to extract useful information from Frequently Asked Questions (FAQ) related to ASD and their answers from more than 10 web pages of well-known agencies like Autism Recovery Network (Singapore) and Amaze Australia (which is highly recommended by my child psychologist friend).
3. **Knowledge FAQ:** Collated all the FAQ pairs obtained from web scraping done by all team members and curated all these into a concise and easy-to-understand Knowledge FAQ. I was also responsible for enabling the Knowledge base FAQ (beta feature) in Dialogflow and making it work in our chatbot.
4. **Overall Integration:** I was tasked with the important role of integrating all the work from the whole team, making adjustments, and designing the entire system workflow to deliver a seamless experience for end-users using our chatbot. I also carried out extensive testing after integration to detect and correct corner cases to ensure a fully functional cognitive chatbot.
5. **Google Assistant Deployment:** Set up Google Assistant for deployment, completed the final round of extensive testing and review with the team and was responsible for releasing our ASDBot to production.

Learning Outcomes

During the course of this project, I gained valuable exposure and knowledge pertaining to what is required of a good cognitive chatbot system and end-to-end development.

1. Technical Learnings
   - **Google Dialogflow:** At the beginning, I got to better understand the different interfaces to build a chatbot system like Google's Dialogflow and Amazon's Lex. After understanding how each platform works, I managed to come to a conclusion and suggested to the team to decide on Dialogflow due to its ease of use, automatic agent build, and ability to engage users via more interactive voice or text-based conversation. Being responsible for the integration task, I also developed a much better understanding of the overall workings and different components (like intents, events, context, and lifespan, as well as rich response such as suggestion chips) in Dialogflow and how to make use of them to design a smooth functional workflow.
   - **Web Scraping:** In order to efficiently extract relevant useful information from online web resources, I realized that there are quite a few python packages, like Beautiful Soup, Selenium, and AutoScraper, that can help achieve this. Through this process, I also got to understand better how websites are designed and structured using HTML tags.

2. Non-Technical Learnings
    - Since our chatbot is designed to help anyone better understand ASD, I managed to gain a deeper understanding of ASD and other related medical terms. Being a father for a 2-year-old child myself, I even applied the M-CHAT screening tool developed by my other team member to access my child's behavior and was glad the diagnosis outcome was 'low risk'! Before embarking on this project, I did not know such a convenient tool exist.
    - During the process of setting up our chatbot for integration with Google Assistant, I was made aware of the importance of personal privacy protection as one of the requirements is to come up with our own Privacy Policy before allowing us to proceed. Most importantly, I also learned that we have to be sensitive to the type of information we provide to the end-user. In this case, we are dealing with medical and health-related information but since none of our team members are medically trained, we have to be careful and state such a disclaimer.
    - Thanks to this project, I got the opportunity to work and build strong relationships with teammates from another country, India. They are also of different age groups, professional backgrounds, and cultures, which makes it more interesting interacting and learning about them.

<u>Application of gained knowledge and skills</u>

Being in the semiconductor industry for the past 15 years, I never had the chance to be involved in and build such a fun and cognitive chatbot system. I was amazed and impressed every time I came across a smart Chatbot from any particular website, but now I feel that I had gained very useful experience and the confidence to develop one myself. This skill I believe will contribute a lot to my next career path, as I always notice fresh engineers who joined my company tends to have a lot to learn and thus a lot of questions. I believe this is the same for any industry. Although what I usually do was to maintain a traditional FAQ page hosted somewhere within the company system that they can assess, but I am sure if this was done in the form of an interactive Chatbot, I am sure the learning will be much more fun for them.

In addition, during my work on building the Knowledge base FAQ beta feature in Dialogflow, I am particularly interested in the other type which is 'Extractive QA' whereby the developer just needs to provide any unstructured text in the form of plain text or HTML and this tool will help intelligently extract the relevant information and form the FAQ. However, since this feature is currently experimental and from my own online research, I realized many developers claimed that this is still unstable and needs further development. Therefore I decided against trying this out for our project in favor of functional stability. But I am definitely going to keep track of this development and hope to use this in my future work.