

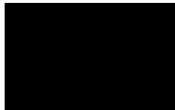
PLAGIARISM DECLARATION

1. I confirm that this assignment is my own work and is not copied from another person's work.
2. I acknowledge that copying someone else's assignment, or part of it, constitutes a form of plagiarism.
3. I have not allowed anyone to copy my work or part of it, with the intention of passing it off as their own work.

Name: Dennis Ng Han Jie

Admin: 204461H

Signature:



Date: 09/12/22

Industry: Logistics & Transportation

Model: <https://teachablemachine.withgoogle.com/models/nzNheZokO/>

Problem Definition

Problem

Singapore is known as one of the top logistic hubs in the world. Thus, drivers in the Logistics and Transportation industry are a core component in ensuring that goods are being transported in time. However, these drivers are often overworked even despite being given normal working hours, where it is from 8am to 5pm. Additionally, they are underpaid, leading to a rise of drivers needing to work overtime to earn more money to support themselves and their family. Thus, they will still be driving on the road until 10pm or even past midnight instead.

In addition to these drivers, there are also taxi drivers or private-hire drivers. Like the drivers in the Logistics and Transportation drivers, they would be working during the wee hours of the night or morning. This is because they are incentivized to do so since there will be lesser competition, and their fares will also be higher.

As a result, these drivers will be fatigued due to the lack of sleep and not having sufficient rest time. And therefore, they will not be as alert as before and will be unable to concentrate on the road, which is as dangerous as drunk driving. Thus, this has led to many road accidents caused by drowsy driving, some were even fatal as well.

Impact

As a result of these road accidents, many lives have been lost. And as for those who were lucky enough to survive the accident, they would need time to recover and will be unable to work for a while. Thus, there will be a loss in revenue for both the survivors and their employers.

They could also potentially be disabled for life and will be unable to carry on with their current lifestyles.

There may also be cases where survivors may develop psychological trauma from the accident, resulting in Vehophobia, the fear of driving, and will therefore need to go through therapy which may be costly and time-consuming.

Additionally, the companies that they are working for will be impacted as well as they may need to cover the costs of their hospital bills. There will also be lesser manpower while these survivors are still recovering, resulting in a loss in productivity for the company.

Furthermore, the goods that were being transported may have also been damaged in the accident which needs to be replaced but may be costly, leading to a loss in revenue for them.

Solution Formulation

Thus, to solve this problem, a computer vision AI that can detect if drivers are drowsy driving should be implemented into future cars. Doing so will help prevent future accidents caused by drowsy driving.

Dataset

The dataset we will be using will consist of pictures of humans with their eyes closed, and pictures of humans with their eyes opened. This allows the AI model to recognize whether the user have their eyes opened or closed, and by doing so, we will be able to determine if the driver is falling asleep on the wheel.

The dataset also needs to consist of people from different races, and those with different iris colours, and of different genders. This is to ensure that the AI model will remain effective regardless of their race, iris colours, and gender.

The pictures will then be labelled accordingly depending on which category it belongs to. This is to let the AI model know the exact output we are looking for. For instance, referring to the figures below, Figure 1 will be labelled with 0 and Figure 2 will be labelled with 1. This is because supervised learning will be used to train the AI model, thus it needs to know what the correct output is to allow it to correct itself and adjust accordingly.



Figure 1 Drowsy Driving (Label = 0)



Figure 2 Driving (Label = 1)

Computer Vision / NLP

Since we will be using images and classifying into two classes, eyes open and eyes closed, Computer Vision will need to be used. This is because Computer Vision has the ability to extract features from images and recognize patterns in them, which we will need in order to classify the images into the two categories mentioned earlier.

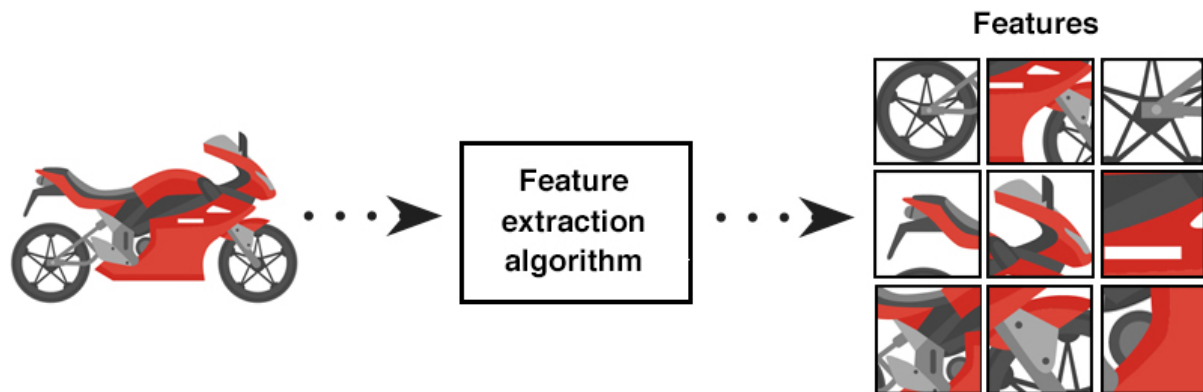


Figure 3 Computer Vision Feature Extraction

Development Process & Testing

Google Teachable Machine Model: <https://teachablemachine.withgoogle.com/models/2CK505Ny3/>

Google Teachable Machine will be used to train the AI model that will be used to solve our problem. After our dataset has been collected, it will first be split into a ratio of 70:30, where 70% of the dataset is used for training, and the rest will be used for testing. This is to check whether there will be an issue of overfitting, where the AI model is only able to determine the correct output on pictures that it has been trained on before.

The following datasets shown below are used for training and testing the model.

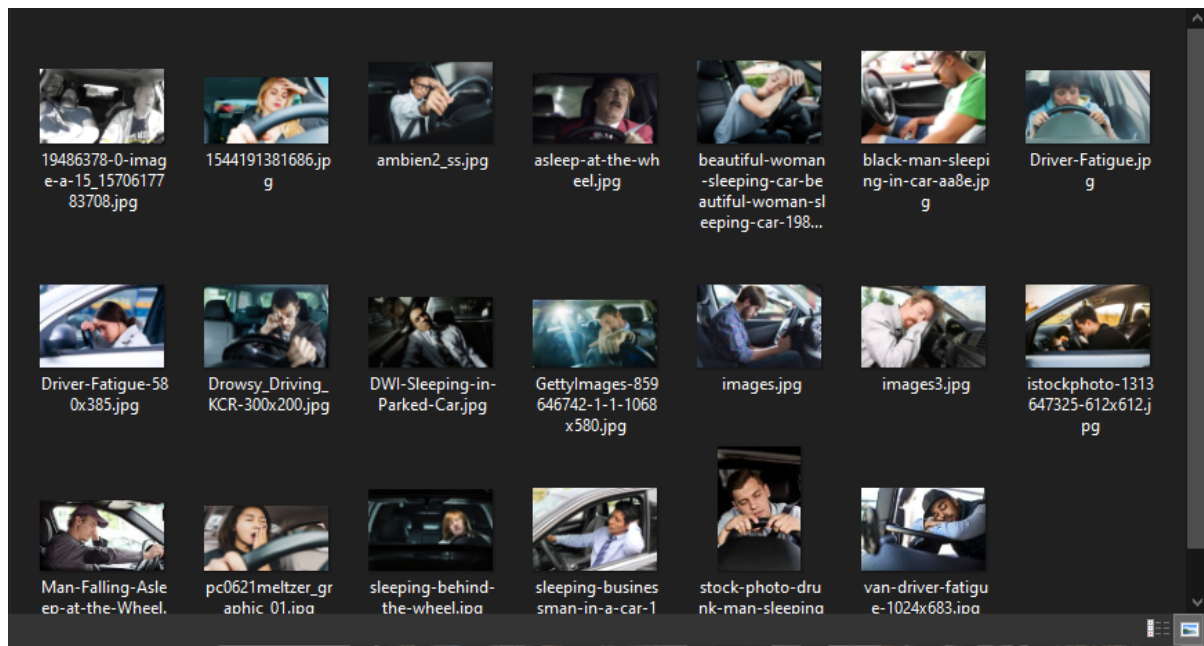


Figure 4 Drowsy Driving dataset



Figure 5 Driving dataset



Figure 6 Test dataset

The accuracy per epoch and loss per epoch values will also be observed during training. This is to check if the AI model is indeed learning to recognize the features and can accurately determine the correct output. A good example would be like Figure 4 and 5, where the accuracy per epoch should be increasing, and the loss per epoch decreasing as the number of epochs increases.

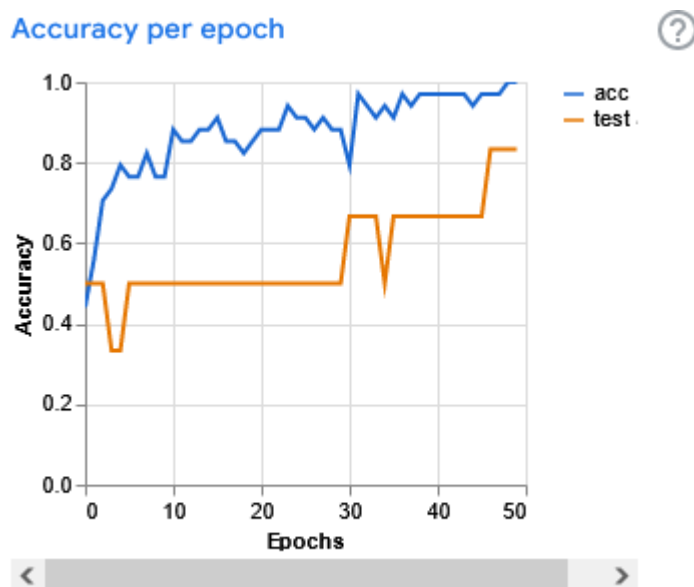


Figure 7 Accuracy per epoch for trained AI model

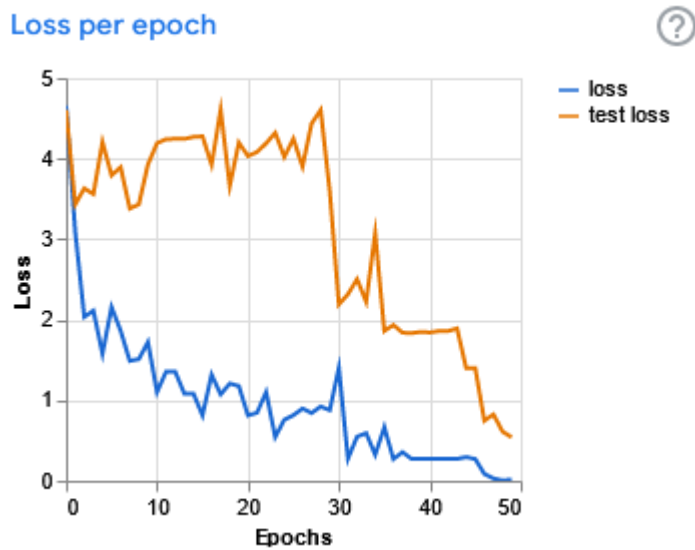


Figure 8 Loss per epoch for trained AI model

A confusion matrix can also be calculated to determine which category the AI model has issues with. By doing so, we can feed in more data for that category, allowing the AI to train more and recognize them better. An example of how a confusion matrix looks like is Figure 6. In this example, the AI model determined some of the images to be of category 5, when it was supposed to be of category 2 and 3.

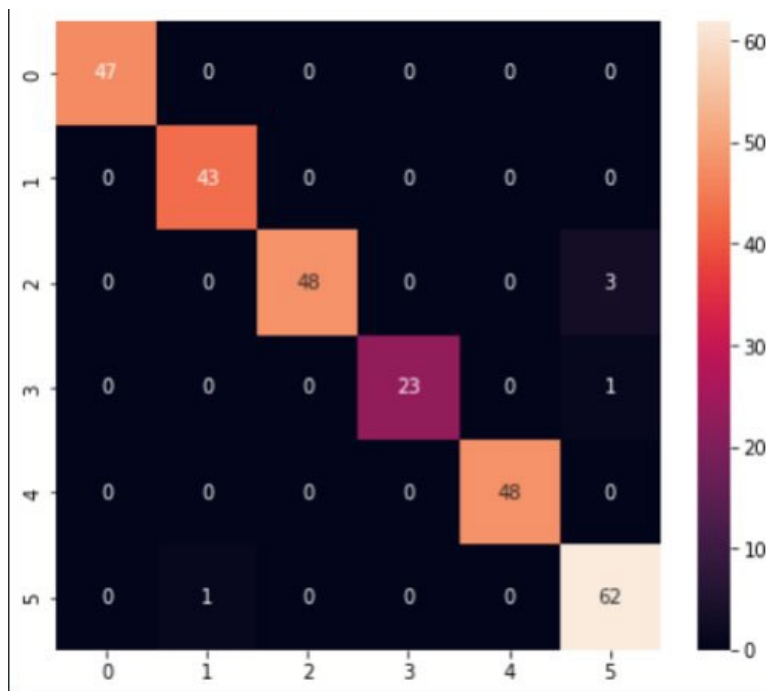


Figure 9 Confusion Matrix example

In the case where the AI model can precisely determine all categories, it would look like Figure 7.

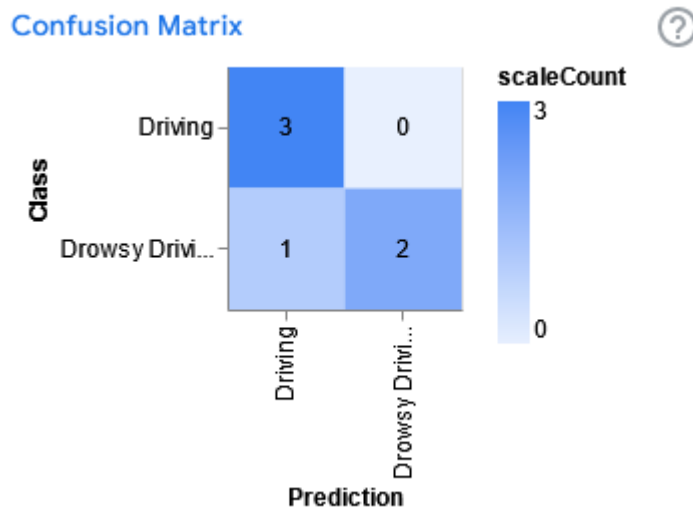


Figure 10 Confusion matrix for trained AI model

Thus, judging from these figures, we can tell that the model trained with the Google Teachable Machine in one of the tests, it determined that it was Driving when it was in fact Drowsy Driving. Thus, more images on drowsy driving will need to be inputted to allow it to recognize it better. However, from these results, it shows that this solution is feasible in recognizing whether the driver is fatigued.

Deployment

After our AI model has been trained using Google Teachable Machine, it will need to be placed on a dashcam that is able to view the driver, so that it will be constantly capturing the driver's eyes.

It will also need to be integrated with the car's electronics such that if the AI were to detect that the driver's eyes were closed for longer than 3 seconds, it will sound an alarm through the car's speakers to wake the driver up which the driver along with turning on the hazard lights. The car will also not be able to accelerate and will slowly decelerate, forcing the driver to a complete stop. If the alarm has been going off for more than 10 seconds, when the car is stopped, the driver will not be able to start the car until a few hours have gone by.


Prototype & Test Cases


Test case 1


The first test case will be a driver that is fatigued. This is to determine if the AI model is able to detect if the user is indeed drowsy driving.





Figure 11 Drowsy driving test


Preview  **Export Model**

Input ☒ **ON** File 


Choose images from your files,
or drag & drop here


Import images from
Google Drive





Output

Driving

**Drowsy
Drivi...**

100%

Figure 12 Test case 1 output

Test case 2

The second test case will be of a driver that is not drowsy. This is to ensure that the AI model is able to determine accurately determine if the user is drowsy or not drowsy, as it may lead to a bad user experience if there are a lot of false positives or worse, not being able to solve our problem which was to prevent accidents from happening caused by drowsy driving. The below figure will be used for this test.



Figure 13 Driving test

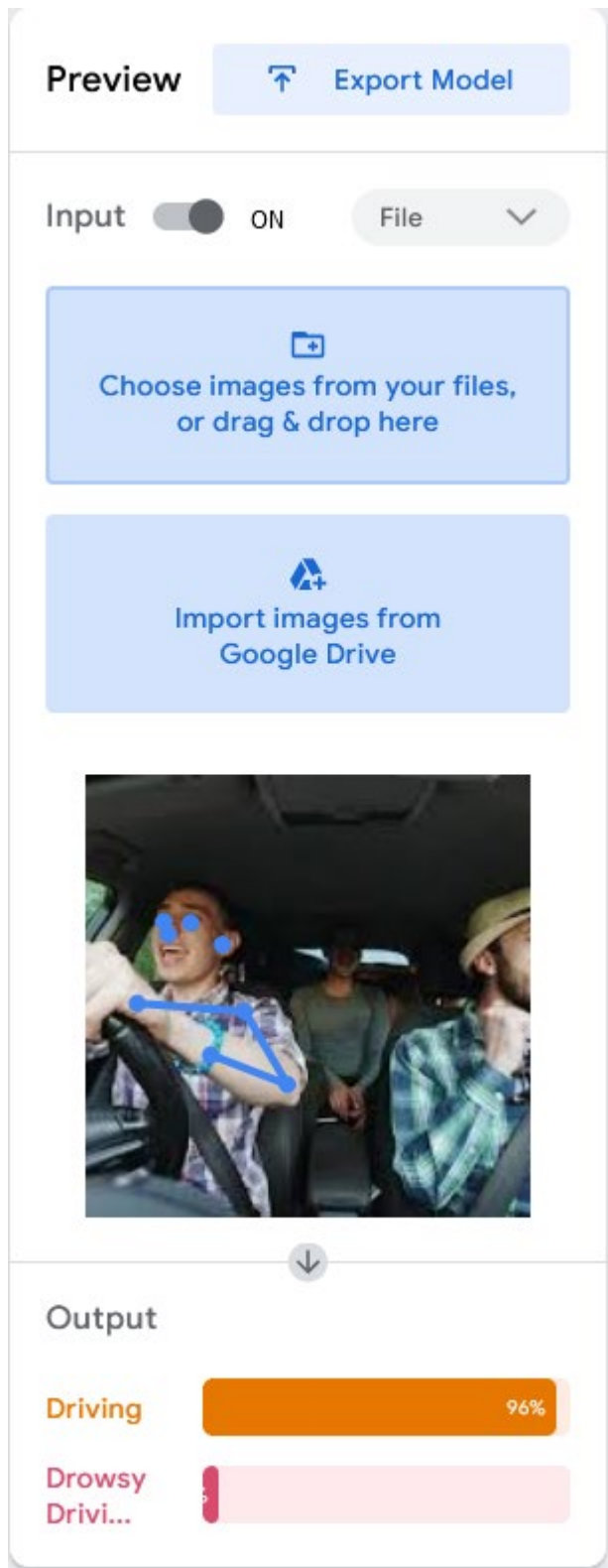


Figure 14 Test case 2 output

As you can see from the output, the AI model is able to accurately determine if the user was drowsy driving or not.

Thus, by integrating this AI model into cars, it will help to prevent accidents that are caused by drowsy driving. This is because if the AI model detects the driver has fallen asleep on the wheel, the car will not be able to accelerate anymore and slowly decelerating to a stop. The car will also sound an alarm that may wake the driver up and turn on the hazard lights. Turning on the hazard lights helps to warn other drivers to be wary of the vehicle so that they will remain alert. Additionally, if the driver does not stop the alarm within 10 seconds, after the car has stopped, they would not be able to start the car again until a few hours has gone by. This forces the driver to rest first before continuing driving.

Therefore, I believe that there will be fewer accidents that are caused due to drowsy driving as a result of implementing this AI model into future cars.

Industry: Healthcare

Model: <https://bot.dialogflow.com/1b030206-c7e5-4677-801a-4ac7ea2c4bfc>

Problem Definition

Problem

In recent years, the healthcare industry in Singapore has become increasingly important, not only because of COVID-19, but also due to the increasing aging population. As a result, many people are hospitalized and needs to be looked after, and thus many workers in the healthcare industry are being stretched thin and are on the verge of burnout. To resolve this, advancements in the healthcare industry should be made to help reduce the workload on these healthcare workers.

Another reason for the never-ending workload is due to many Singaporeans preferring to immediately see a doctor even for the merest problems. One such example would be going straight to the Accidents & Emergency department when they only have a common flu because they fear that it might be COVID-19. As a result, many healthcare workers need to see to each and every one of them when it could have been easily taken care of by resting at home or by seeing a General Practitioner at a nearby clinic instead.

Impact

As a result, many healthcare workers have to work longer hours, and sometimes only being able to have breaks for meals. Some may even feel that they are being underpaid for the amount of work they have to do every single day. This has led into a vicious cycle where healthcare workers are quitting their jobs and passing on their workload onto another, causing the next worker to feel overloaded. Eventually, there will not be enough manpower to tend to all the patients in the healthcare industry.

During these trying times, it is especially important to ensure that there is more than enough manpower. This is because having lesser manpower would lead to workers becoming more fatigued due to having heavier workloads, which may eventually result in them making mistakes. A minor mistake in the healthcare industry can be disastrous. For instance, a nurse may administer their patients an incorrect dosage for their medicine, or worse, a surgeon making a mistake when performing surgeries due to the lack of sleep and being unable to concentrate.

Solution Formulation

Thus, there is a need to make advancements in the healthcare industry. And one of the advancements that I believe will be helpful is making a chatbot. This will allow users to ask the chatbot questions regarding their symptoms, allowing them to determine if they should seek a specialist or not.

Datasets

For our dataset, we will be using phrases of questions that would be commonly asked by a general practitioner, along with a list of symptoms for common illnesses. This is to determine what symptoms the user has and whether they should seek for a specialist, or it is just a mild illness.

Computer Vision / NLP

Natural language processing (NLP) is a must in order to create a chatbot. Only with NLP would the computer be able to understand the text that will be inputted, only then can the AI determine what is needed from the user. Else, it would not be able to comprehend what is it the user wants to know from the chatbot. Furthermore, if the text inputted were to differ from each other a little, through the implementation of NLP, the AI will be able to know that they are of the same intention. For instance, "I got a sore throat" and "My throat hurts", to a human we would know that they both mean the same thing, however without NLP, a computer would not be able to know if they are related. This is possible because of word embedding, where the words are placed into a vector space, and the closer they are to each other, they share similar meanings.

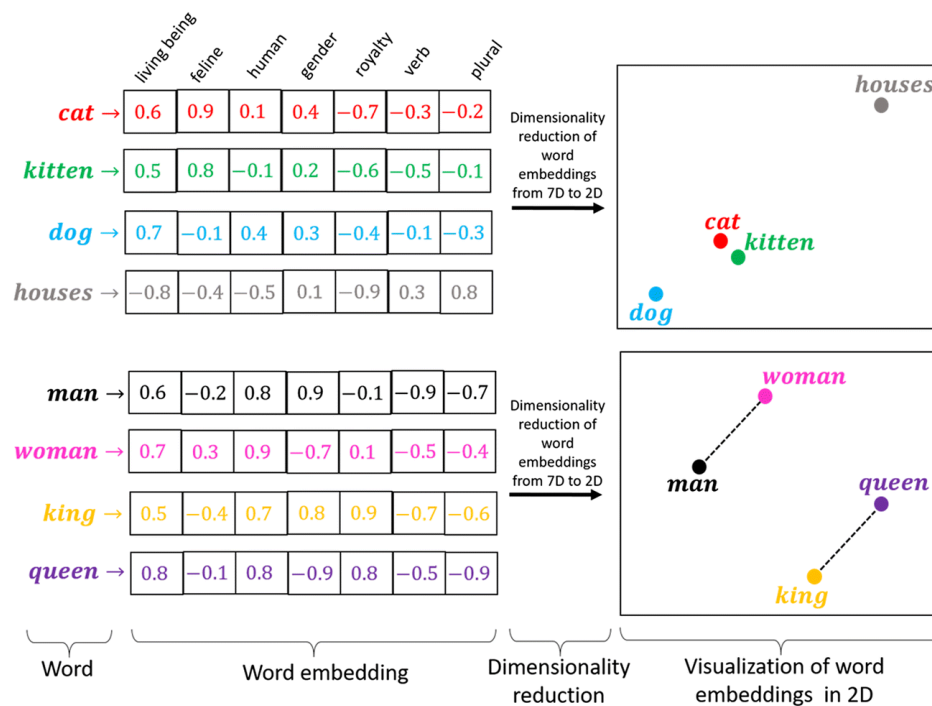


Figure 15 Word embedding

Development Process & Testing

Google DialogFlow will be used to create our chatbot. After creating our agent, we will need to create three additional intents other than the default ones, “Intro”, “Major Illness”, and “Mild Illness”.

In each intent, training phrases will need to be added accordingly along with the appropriate responses. By doing so, it allows the chatbot to know what the appropriate response should be depending on the user’s input. For instance, if the user were to input “I need help”, the chatbot will reply with “What are your symptoms?”. Refer to the figures below for more examples.

Intro SAVE

Contexts ? ▼

Events ? ▼

Training phrases ? Search training phrases Q ^

⚠ Template phrases are deprecated and will be ignored in training time. More details [here](#).

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

☞ I need help

☞ I am injured

☞ I don't feel well

☞ I feel sick

Figure 16 Intro intent - Training phrases

Intro SAVE

Action and parameters ^

Enter action name

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

[+ New parameter](#)

Responses ? ^

DEFAULT +

Text Response 🗑

- Can you please describe your symptoms?
- What are your symptoms?
- Enter a text response variant

ADD RESPONSES

☐ Set this intent as end of conversation ?

Figure 17 Intro intent – Responses

Major illness

SAVE

Events

Training phrases

Search training phrases

⚠ Template phrases are deprecated and will be ignored in training time. More details [here](#).

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

» I can't breathe

» I am coughing up blood

» My body temperature is 39 degrees

» I have a high fever

» I feel nauseous

» I am unable to move

Figure 18 Major illness intent – Training phrases

Major illness

SAVE

Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	major-symptoms	@major-symptoms	\$major-symptoms	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Responses

DEFAULT

+

Text Response

1 Please seek for medical attention immediately. Please call 995 if it is an emergency

2 Enter a text response variant

ADD RESPONSES

☒ Set this intent as end of conversation

Figure 19 Major illness intent – Response

Mild illness
SAVE

Contexts ⓘ

Add input context

2 Mildillness-followup ⓘ Add output context

Events ⓘ

Training phrases ⓘ

Search training phrases 🔍

⚠️ Template phrases are deprecated and will be ignored in training time. More details [here](#).

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

I have a mild headache

I am coughing

I keep shivering

I feel cold

My body temperature is 37.5 degrees

Figure 18 Mild illness - Training phrases

Mild illness
SAVE

Action and parameters

Enter action name

REQUIRED ⓘ	PARAMETER NAME ⓘ	ENTITY ⓘ	VALUE	IS LIST ⓘ
<input type="checkbox"/>	mild-symptoms	@mild-symptoms	\$mild-symptoms	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Responses ⓘ

DEFAULT +

Text Response

1 Do you have any other symptoms?
2 Enter a text response variant

ADD RESPONSES

☐ Set this intent as end of conversation ⓘ

Figure 19 Mild illness – Responses

Entities will also need to be created to recognize the which parts of the sentence the chatbot should be looking out for to determine what symptoms the user has. Thus, two entities labelled “major-symptoms” and “mild-symptoms” will be created

And in these entities, we will be adding entries of symptoms depending on which entity it is, for instance, in the major-symptoms entity, we will be adding in “nauseous” and “can’t breathe” as entries. We will also be adding synonyms for each of the entries in both entities such that the chatbot will be able to recognize the symptoms even if the wording is different.

major-symptoms SAVE

☒ Define synonyms ☐ Regexp entity ☐ Allow automated expansion ☐ Fuzzy matching

high fever	high fever, 39 degrees, 40 degrees
unable to move	unable to move, leg broken, ankle sprain, sprained ankle
nauseous	nauseous, feel like vomiting, world spinning, fainting
can't breathe	can't breathe, difficulty breathing, chest tight
Click here to edit entry	

[+ Add a row](#)

Figure 20 major-symptoms entity

mild-symptoms SAVE

☒ Define synonyms ☐ Regexp entity ☐ Allow automated expansion ☐ Fuzzy matching

cold	cold, shivering
37.5	37.5, 38
cough	cough, mild cough
sore throat	sore throat, throat hurts
fatigue	fatigue, muscle pain, muscle aches
Click here to edit entry	

[+ Add a row](#)

Figure 21 mild-symptoms entity

Follow-up intents should also be created. This is to ensure that the user has told the chatbot all of their symptoms and confirm that they do not have any major symptoms.



Figure 22 Follow-up intents for mild illness

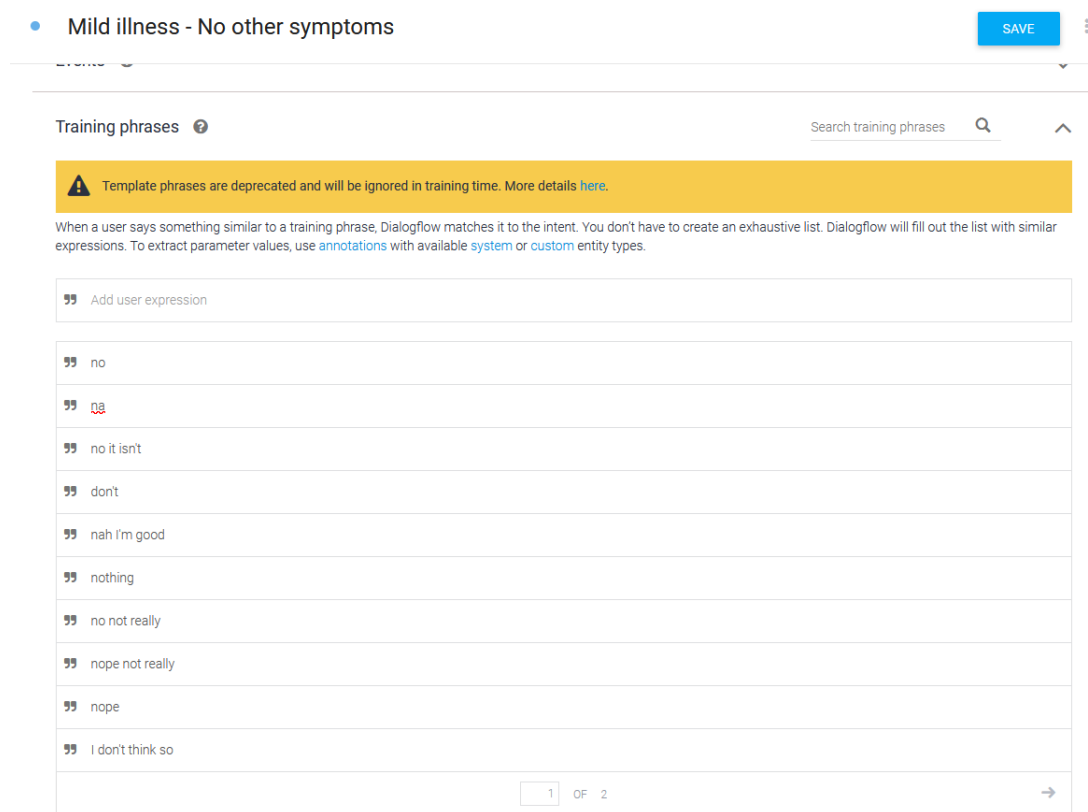


Figure 23 Mild illness - No other symptoms - Training phrases

• Mild illness - No other symptoms

SAVE

Action and parameters

MildIllness.MildIllness-no

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Responses

DEFAULT +

Text Response

- I suggest that you take some medicine and to take some time off to rest at home. If you still feel that your condition is not recovering or is worsening even after resting for a few days, please seek medical attention immediately.
- Enter a text response variant

ADD RESPONSES

☒ Set this intent as end of conversation

Figure 24 Mild illness - No other symptoms – Response

• Mild illness - more symptoms

SAVE

Events

Training phrases

Search training phrases

⚠ Template phrases are deprecated and will be ignored in training time. More details [here](#).

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

” yes

” yes I do

” yeah

” yup

” yep

” absolutely

Figure 25 Mild illness - more symptoms - Training phrases

• Mild illness - more symptoms

SAVE

Action and parameters

MildIllness.MildIllness-yes

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Responses

DEFAULT +

Text Response

- 1 What other symptoms do you have?
- 2 Can you please describe your symptoms?
- 3 Enter a text response variant

ADD RESPONSES

☐ Set this intent as end of conversation

Figure 26 Mild illness - more symptoms – Responses

A fallback should also be added. Fallbacks are what the chatbot will respond with if it does not know what the user has inputted. In the example below, a fallback was added as a follow-up intent for the mild illness intent. Thus, as a safety precaution, the chatbot will advise the user to seek for medical help as it is unable to determine if their symptom is major or not.

• Mild illness - fallback

SAVE

Feedback Intent training phrases are "negative examples" the agent will not match to any other intent. [Read more in documentation](#)

Add user expression

Action

MildIllness.MildIllness-fallback

Responses

DEFAULT +

Text Response

- 1 Sorry, I didn't quite get that. Please seek medical attention if you find that it may not be just a mild illness. Thank you
- 2 Enter a text response variant

ADD RESPONSES

☒ Set this intent as end of conversation

Figure 27 Mild illness - fallback - Responses

Deployment

After training the chatbot, it should be deployed on the SingHealth website that remains on the side of the screen while also being minimizable, similar to the figure shown below. This is to let users to easily find and also allow them to notice this chatbot.

Furthermore, SingHealth already has a page that allows users to find conditions based on symptoms, however not everyone knows about it and knows how to use it effectively due to the website's user interface, leading to inaccurate results. Thus, by implementing it there, users will be able to easily find out more about their conditions based on their symptoms, allowing them to determine if they should seek for medical help or not. In turn, it will lead to lesser people visiting the hospital for minor illnesses. Additionally, with the database used for the Conditions & Treatments and Medicines page, it can be fed into the chatbot to train it, allowing it to better determine which symptoms are minor or major.

It should also be deployed on the HealthBuddy app to make it more accessible without needing to visit the website. More and more Singaporeans are using the HealthBuddy app as well due to the ability to make appointments on the app. Thus, they will be able to know about the chatbot before they even make an appointment, reducing the number of visitors to the clinics and hospitals that do not have to.

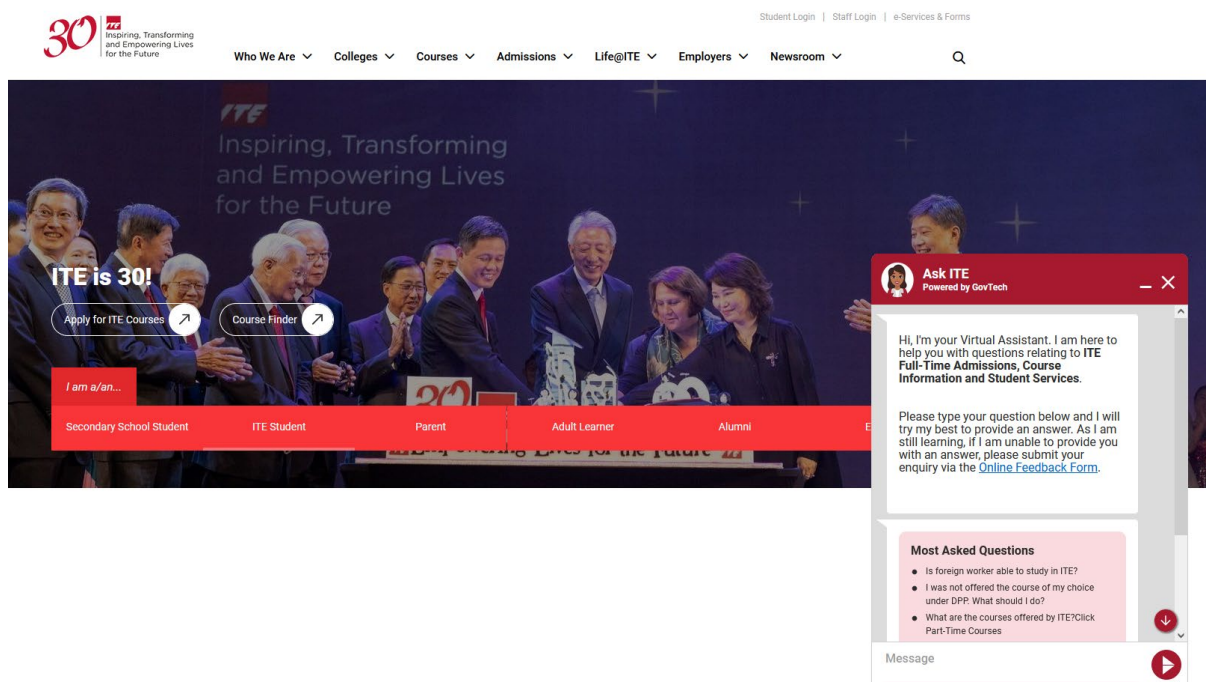


Figure 28 Chatbot example from ITE website

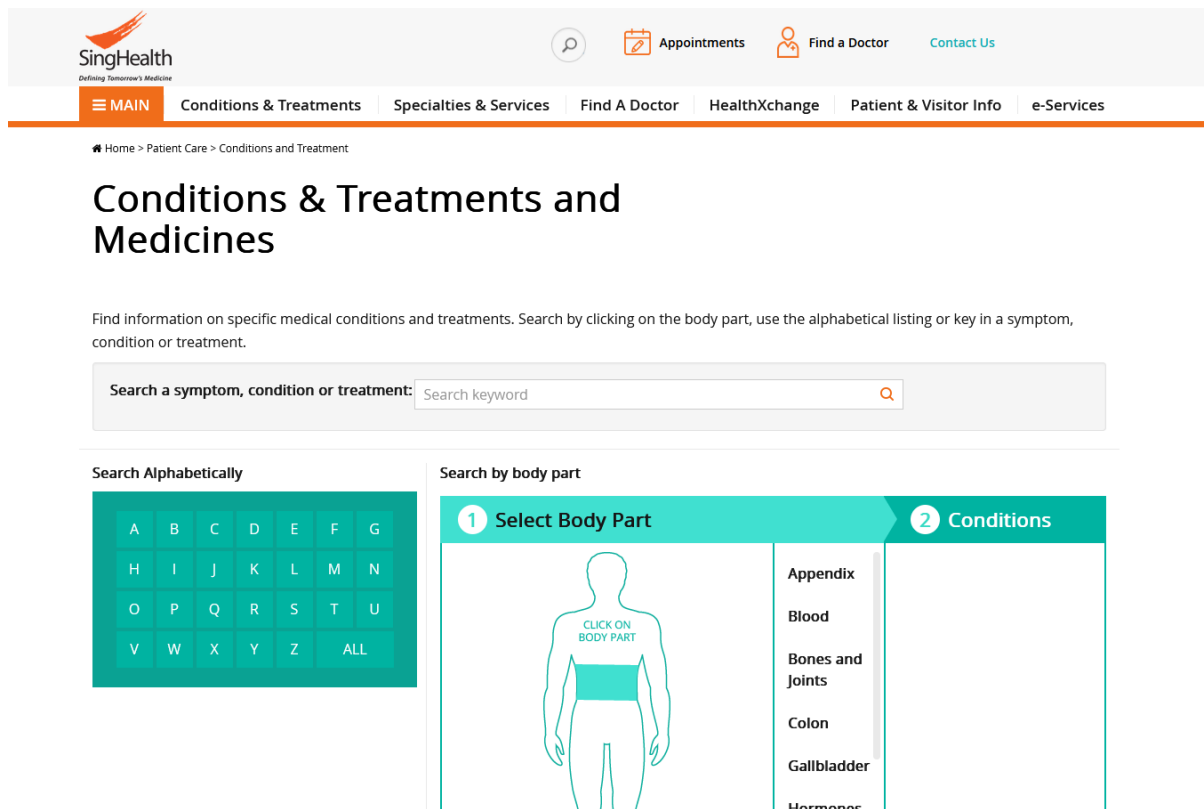


Figure 29 SingHealth Conditions & Treatments and Medicines

Prototype & Test Cases

Test case 1

The first test case would be testing whether it would be to tell the user to rest at home if their symptoms were only minor. This needs to be ensured as the problem was that many Singaporeans were visiting the clinics and hospitals when it was only a minor problem.

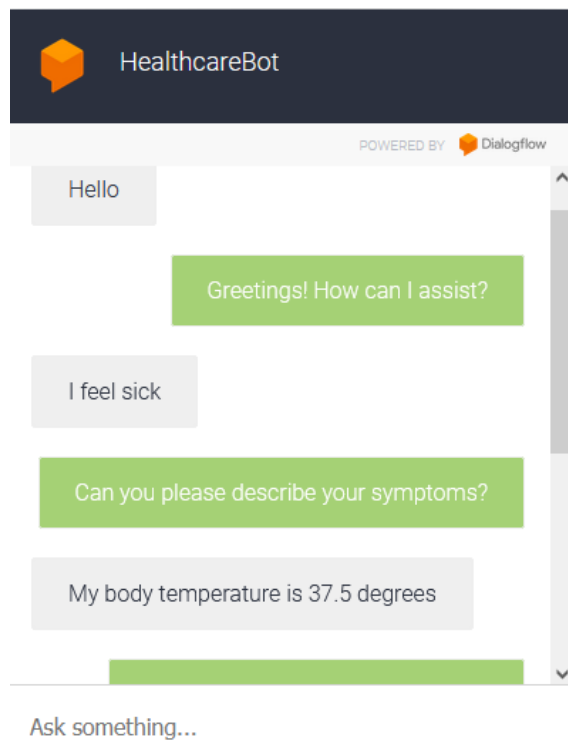


Figure 30 Test case 1 (Part 1)

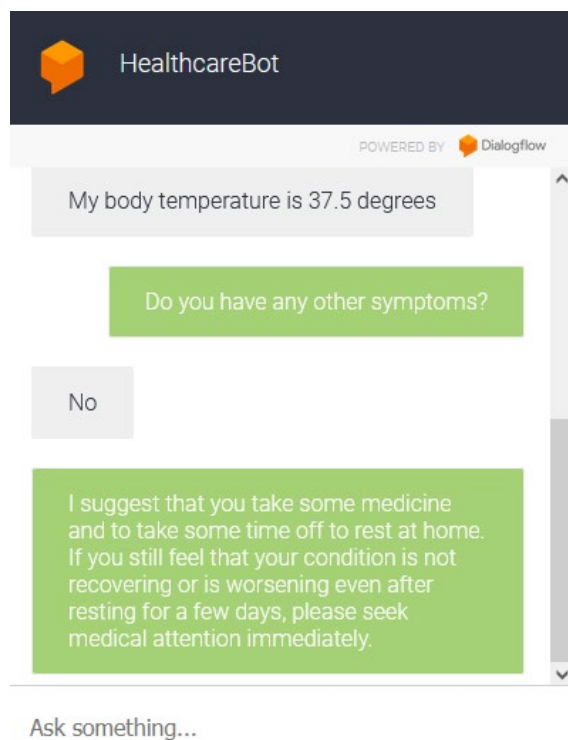


Figure 31 Test case 1 (Part 2)

Thus, as shown in the two above figures, it is able to do so. Therefore, there will be a decrease in the number of visitors to the clinics and hospitals since they would now be able to easily determine if they could just rest at home to recover, without needing to see a specialist.

Test case 2

The other test case would be to test if the chatbot will be able to determine if there was a major symptom even if the user inputs minor symptoms beforehand. This is because the users may not input the major symptoms first, or they could not know which symptoms are major or minor, thus they would instead key in all their symptoms one by one. Therefore, to ensure that the chatbot will still be able to respond appropriately, this should be tested.

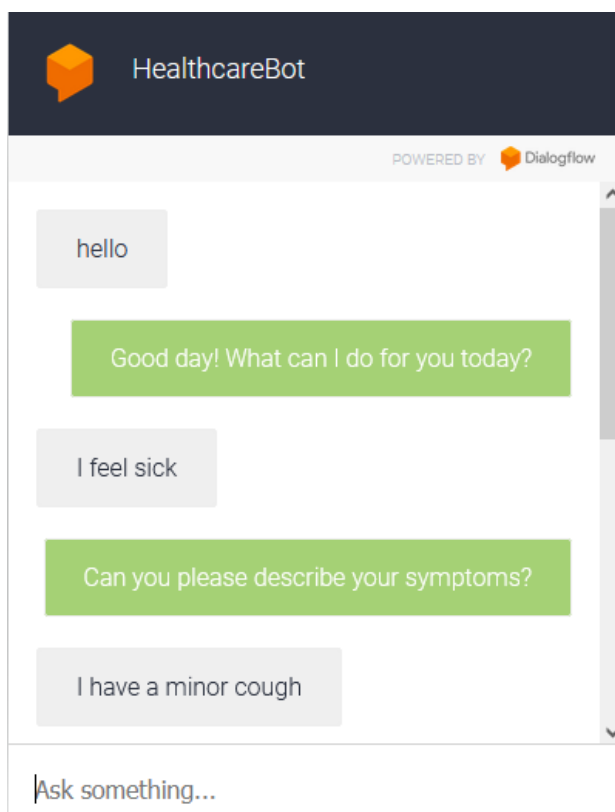


Figure 32 Test case 2 (Part 1)

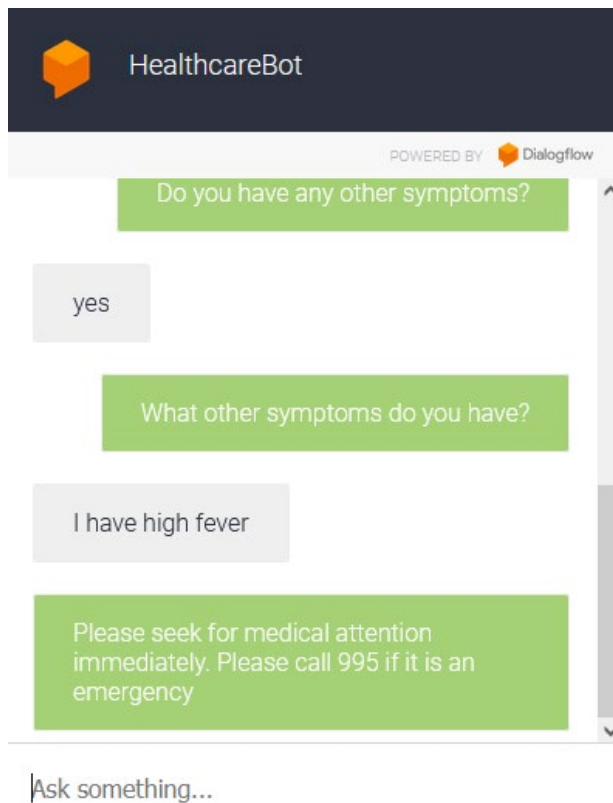


Figure 33 Test case 2 (Part 2)

As shown in the above two figures, it is able to tell the user to seek medical attention immediately at any sign of a major symptom. Thus, it would ensure those users who are in urgent need of medical help will still seek a doctor rather than the chatbot telling them to stay at home to recover.

Therefore, I believe that this solution will be feasible and also be able to help in lowering the workload of the workers in the healthcare industry. With proper implementation, the chatbot will be able to determine if users should seek for medical help or they could just rest at home, helping to decrease the number of patients they would need to care for. Furthermore, by feeding more data to the chatbot along with the ease of training, it will be easy to train the chatbot to recognize even more symptoms. It may also eventually help users in identifying their conditions even without needing to see a doctor.