

**COMP5412 FUNDAMENTALS OF CHINESE COMPUTING**

**Group Project: Dialog System**

|  |  |
| --- | --- |
| Name | Student ID |
| Cheung Kin Yi | 17006441G |
| Fung Kin Kok | 17008299G |
| Li Hiu Wa | 18070633G |
| Zhao Hai Qi | 18087348G |

Contents

[Introduction 4](#_Toc531649820)

[Existing similar systems and applications scenarios 4](#_Toc531649821)

[Commonly used approaches 5](#_Toc531649822)

[Functions of the system 6](#_Toc531649823)

[Time Checking 6](#_Toc531649824)

[Replying by emoji 7](#_Toc531649825)

[Telling jokes 8](#_Toc531649826)

[Simple calculations 9](#_Toc531649827)

[Detecting Traditional Chinese characters and translate to Simplified Chinese and Chitchat 10](#_Toc531649828)

[Questions to check and confirm user intent and Rule based Template 15](#_Toc531649829)

[English Chitchat Template using Facebook API 18](#_Toc531649830)

[Repetitive Detection 20](#_Toc531649831)

[English Chitchat 21](#_Toc531649832)

[Other functions designed but not implemented: 25](#_Toc531649833)

[Find the nearest restaurant 25](#_Toc531649834)

[Change between currency 25](#_Toc531649835)

[System flowchart 26](#_Toc531649836)

[System Flow Chart 26](#_Toc531649837)

[Chatbot Tech Stack 26](#_Toc531649838)

[Dataflow between User and Chatbot 28](#_Toc531649839)

[Error handling 29](#_Toc531649840)

[Answer precision mechanism 30](#_Toc531649841)

[Question similarity mechanism 30](#_Toc531649842)

[Approaches and tools 32](#_Toc531649843)

[Implementation and Key Issue 34](#_Toc531649844)

[ChatterBot Rule-based System 36](#_Toc531649845)

[Sequence to Sequence Model 36](#_Toc531649846)

[Attention Wrapper 37](#_Toc531649847)

[Beam search 37](#_Toc531649848)

[Anti-Language Model 37](#_Toc531649849)

[Key issue of Sequence to Sequence Model Implementation 37](#_Toc531649850)

[Facebook Messenger Deployment 38](#_Toc531649851)

[Key issue of Facebook Messenger Deployment 38](#_Toc531649852)

[Hybrid and integration 39](#_Toc531649853)

[Integration of Chinese and English sequence to sequence model 39](#_Toc531649854)

[Integration of Chatterbot and Facebook Messenger Web server 39](#_Toc531649855)

[Integration of chitchat and task-specified mode inside the Chatterbot 39](#_Toc531649856)

[Integration of Chatterbot and Sequence to sequence model 40](#_Toc531649857)

[Key issue of switch between Chinese and English dialogue 40](#_Toc531649858)

[Test Case 41](#_Toc531649859)

[Role and contributions of group members 42](#_Toc531649860)

# Introduction

With the growth of artificial intelligence in recent years, dialog system, also called chatbot in some scenarios, become a hot topic in society. From the chatbot which was designed to chitchat with human to task specific chatbot who can perform specific tasks for human, chatbot become more and more popular in today’s digital world. Many different e-commerce platforms starts to involve the usage of chatbot, for example, chatbot can help customers to order pizza by asking a few questions following the flow, like what kind of pizza do they want, what toppings do they want to add, chatbot can help customers to book movie ticket, reserve for a table in a restaurant as well as answering Frequently Asked Questions from users, this can reduce much of the human effort in customer services. Moreover, chatbot can work 7 days a week, 24 hours a day, which provides a high availability for customers from all around the world without worrying about the problem of time zone. Chatbot can also provide a helping hand to users in which users can make use of both speech recognition and dialog system functionality, they can speak to their phones and ask questions in their daily life. They can ask the chatbot what is the weather of today, what is the nearest restaurant and other questions without pressing buttons on their phones. As the dialog system become more and more famous in this Information Technology industry, it is a good chance to study more about the technical details of implementing a dialog system.

# Existing similar systems and applications scenarios

There are many chatbots off the shelf. Here I would like to introduce some of the famous chatbot products in recent years.

SIRI is an Intelligent personal assistant, it is part of Apple's iOS and uses natural language UI to answer questions and perform various requests. The function of SIRI includes the basic command of phone operation, Scheduling, and reminders, Search thing on the internet, Navigation, Entertainment, Random tips and tricks. SIRI is embedded in all iOS devices, it includes iPhone, iPad, iMac, Apple Watch. iPhone gets 15.6 percent of all smartphones sold world-widely. So at least 15.6 percent of smartphones users have SIRI.

Google Now is a chatbot for answering questions, making recommendations. Google Home is a home assistant. It can answer questions, play songs, tackle event and reminder, control smart home. All interaction between google home and human is the voice. Google Home is sold US$ 129. And it reaches 100 million installations in 2018.

Alexa is a chatbot developed by Amazon. It is an intelligent personal assistant that inhabits the Amazon Echo device, which is home assistants same as Google Home and Apple HomePod. Amazon Echo using language processing algorithms to receive, recognize and respond to voice commands. It can answer a question, provide weather and news, and like Google Home. Amazon Echo sold US$49.99. and it hits 50 million installations in 2018.

Cortana is Microsoft’s version of intelligent assistant that can set reminders and answer questions using Bing search engine. Cortana is embedded in Windows 10. All PC running Windows 10 is supposed to be installed Cortana. the number of Windows 10 user is the number of Cortana owner.

# Commonly used approaches

According to the application scenarios, dialogue systems can be classified into Chit-Chat (Chatbot) or Task-specific (Assistant) approaches.

For Task-specific approaches, it is used to act as the personal assistant which can help users to achieve a certain task or specific goals. Generally, the task-oriented dialogue systems can be defined into Frame-based approach, Dialogue act and Dialogue Management. It can be the combination of rules and statistical components, and it contains the components including Natural Language Understanding, Dialogue Management and Natural Language Generation. For example, restaurant booking can be used Task-specific approaches to perform.

The Chit-Chat (Chatbot) is another application approach which have no specific goal, and just focus on natural responses and often used for entertainment value. The Chatbot is an artificial intelligence (AI) software that can simulate a conversation with user in natural language. It can be communicated in different channels including messaging applications, websites, mobile apps or telephone.

According to the implementation approaches, dialogue systems can be classified into Rule-based, Retrieval-based, Generation-based approaches.

Rule-based approach are always worked by performing the pattern matching and transformation rules. In recognizing the key words, system can associate the keyword in different ranking, and based on some rules to select a transform on the Memory list which related to the highest ranked keyword.

Retrieval-based approach are developed in information retrieval techniques which allowed system to search and formulate the matching between the user input and candidate responses by matching metrics. In fact, it always provided the responses in a faster way and effective for short-text matching in retrieval-based system. However, if the cases are not in the set, the appropriate response may not be existed.

Generation-based approach are used to generate a response word-by-word by using language model instead of retrieved from repository. In general, the system based on the user input and previous messages, and it applies sequence-to-sequence generation manner to generate the response. It is therefore can be used to handle multi-turn conversations, and different ways to handle the contexts.

Moreover, the Hybrid approaches which combines the different models at the same time. This method is trying to integrate different model into a system which can get the benefits from different approaches.

# Functions of the system

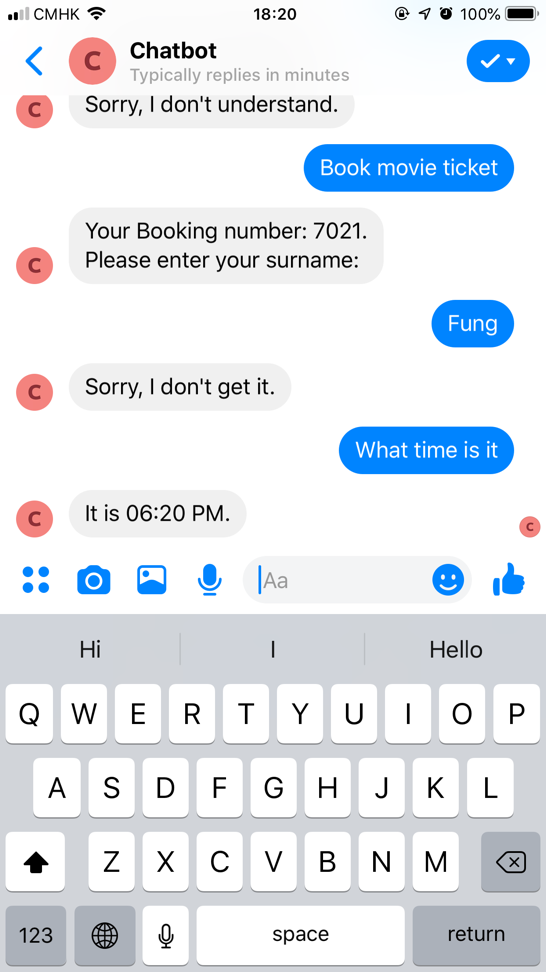
Our chatbot can reply in real time and it supports multi-lingual conversation (both Chinese and English). This allow users from both Chinese and English countries to communicate with our chatbot.

Our system returns a response using generation-based and rule-based ensemble approaches. Our system is ensembled by two sequence-to-sequence models, one trained by Chinese corpus and another one trained by English corpus. We then combined the two neural network models together with the Chatterbot model. The system will check the user input first, if it matches one of the rules we defined, the predefined response will be shown, otherwise, the user input will be passed into one of the generation-based models we trained.

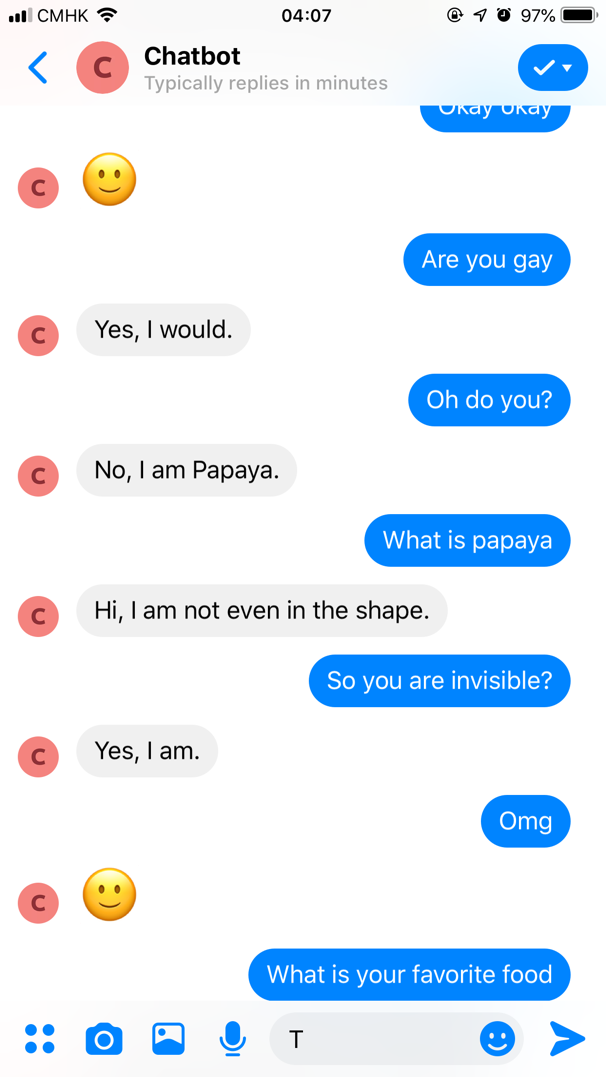
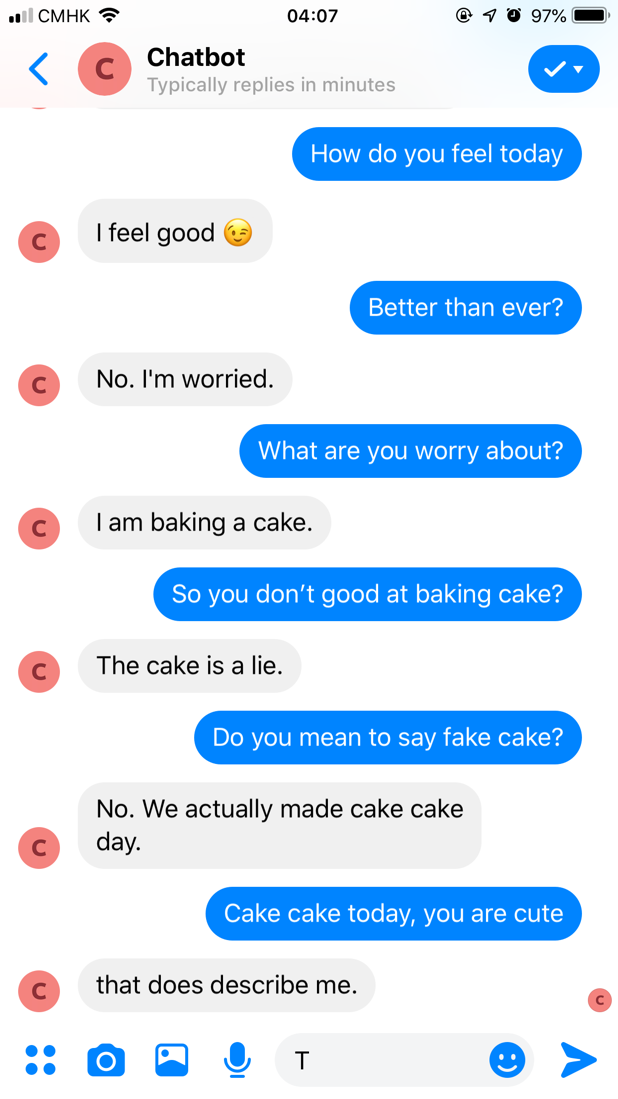
Our chatbot can chat freely with users as well as complete tasks like booking a cinema ticket. When the input message is detected to have the keywords “booking” and “cinema”, it will trigger to make the booking cinema ticket services. After then, our chatbot will generate the booking reference number, and ask the related booking ticket contents including name, booking film, booking date and seat no. Finally, the booking information will be saved and generate the fully booking messages which is confirmed or cancelled.

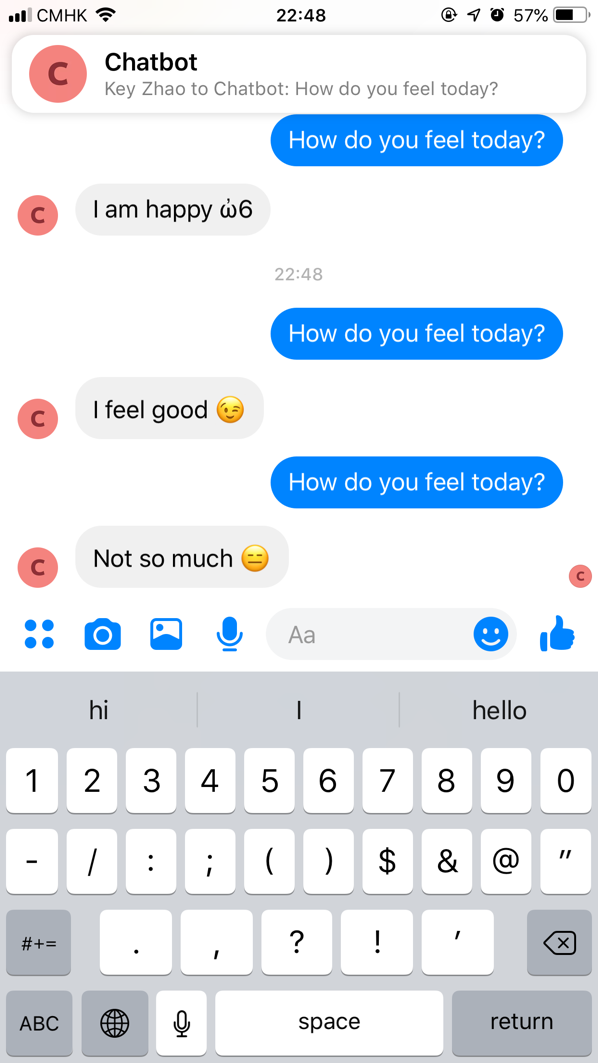
The chatbot also includes functions like replying by using emojis and telling jokes as well as perform simple calculations.

Time Checking

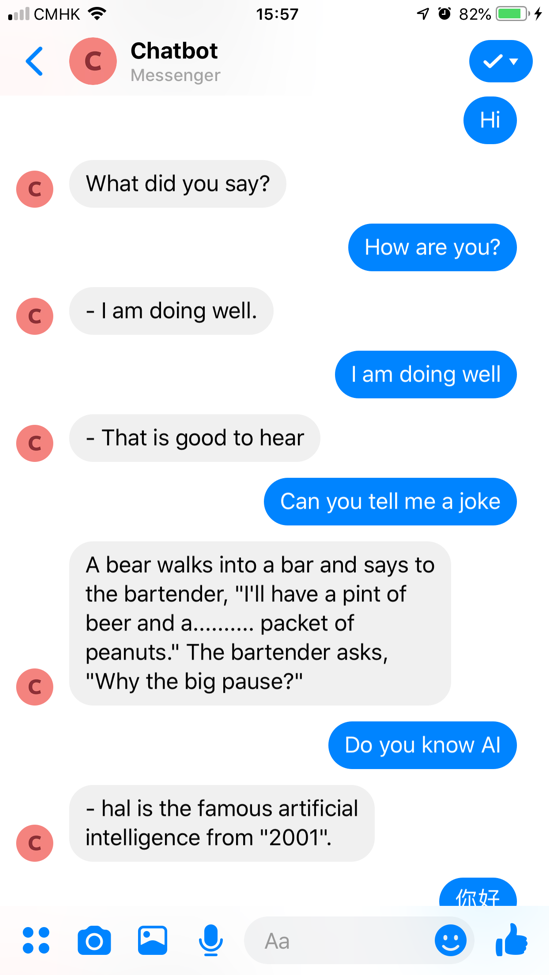


Replying by emoji

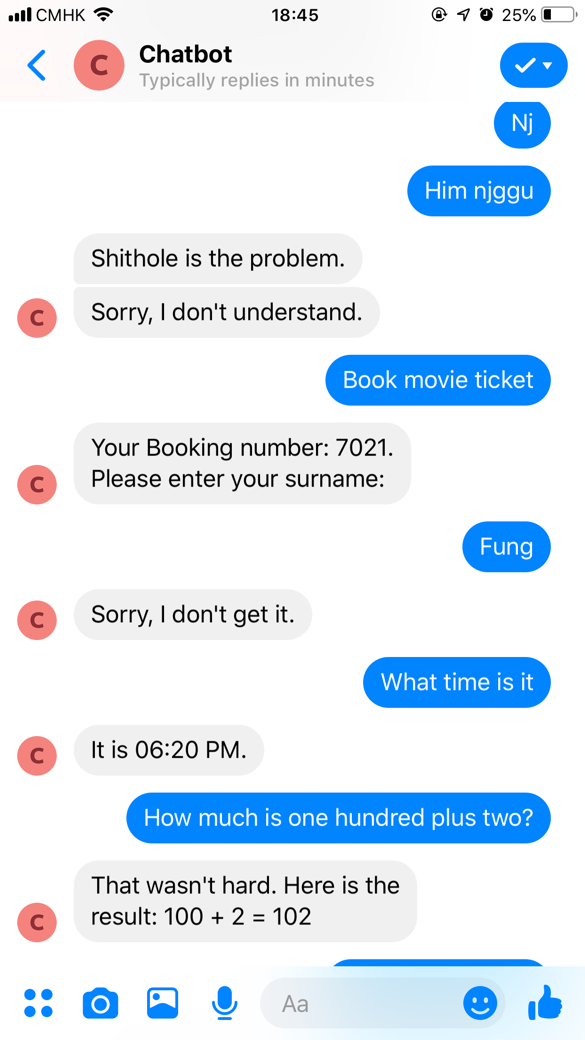




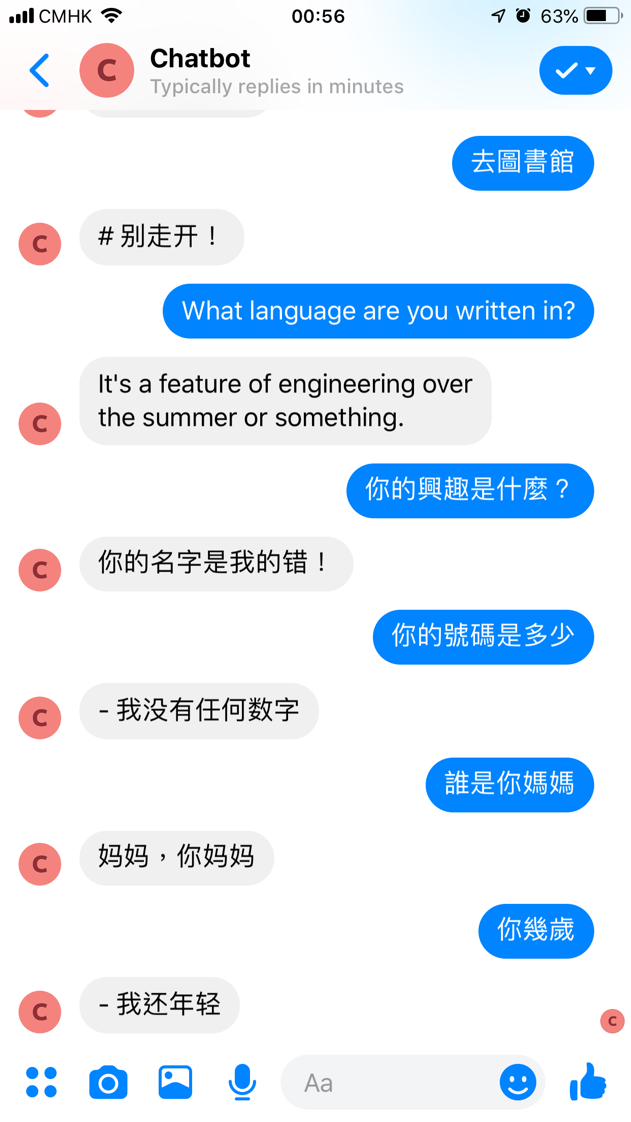
Telling jokes

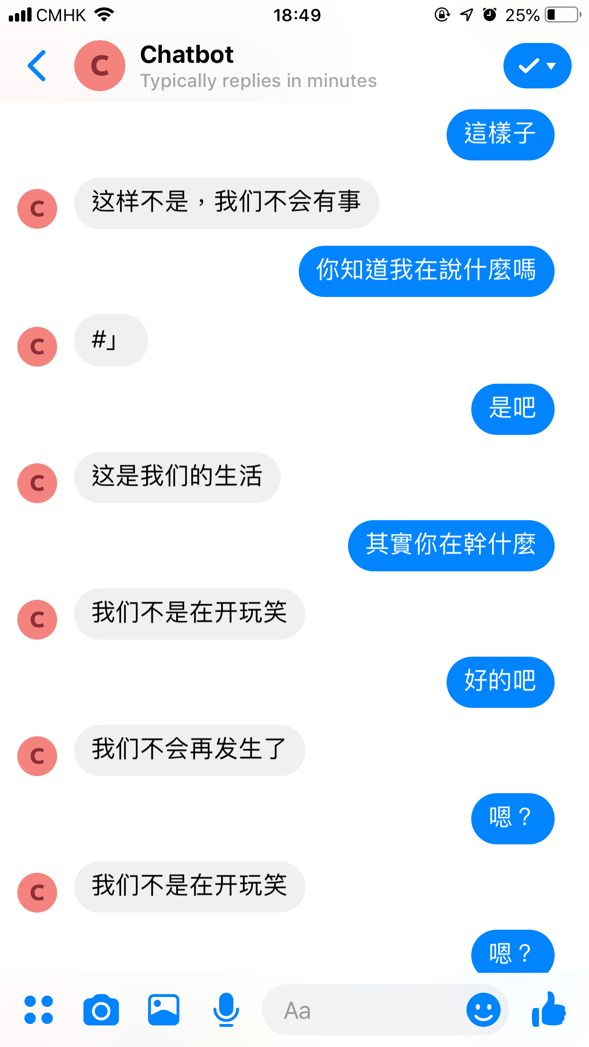
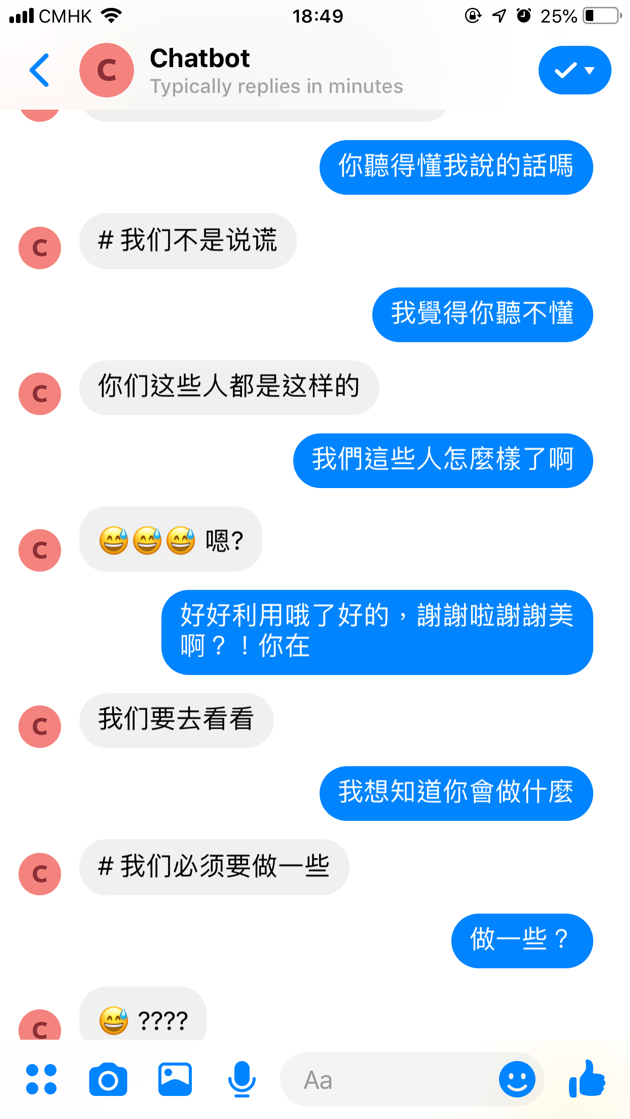


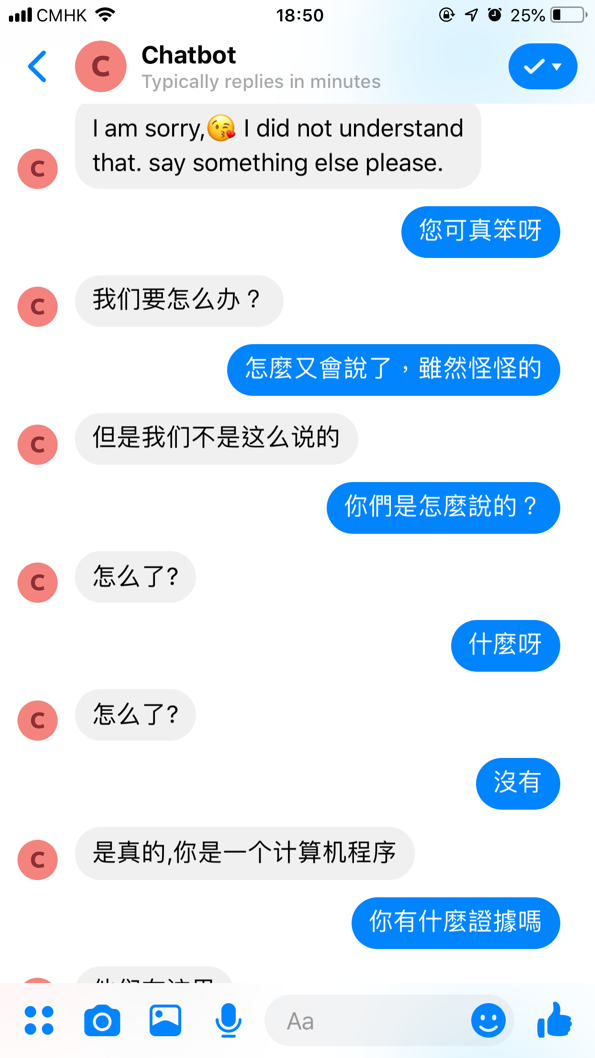
Simple calculations



Detecting Traditional Chinese characters and translate to Simplified Chinese and Chitchat





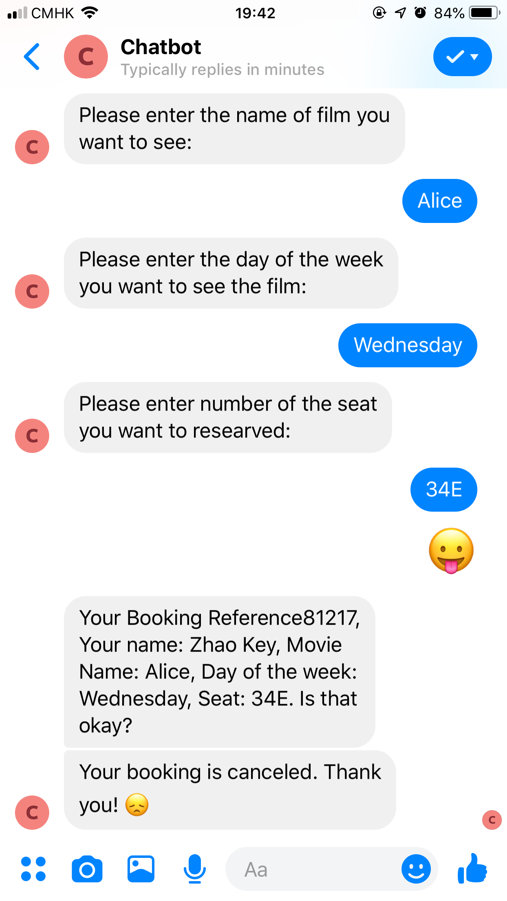


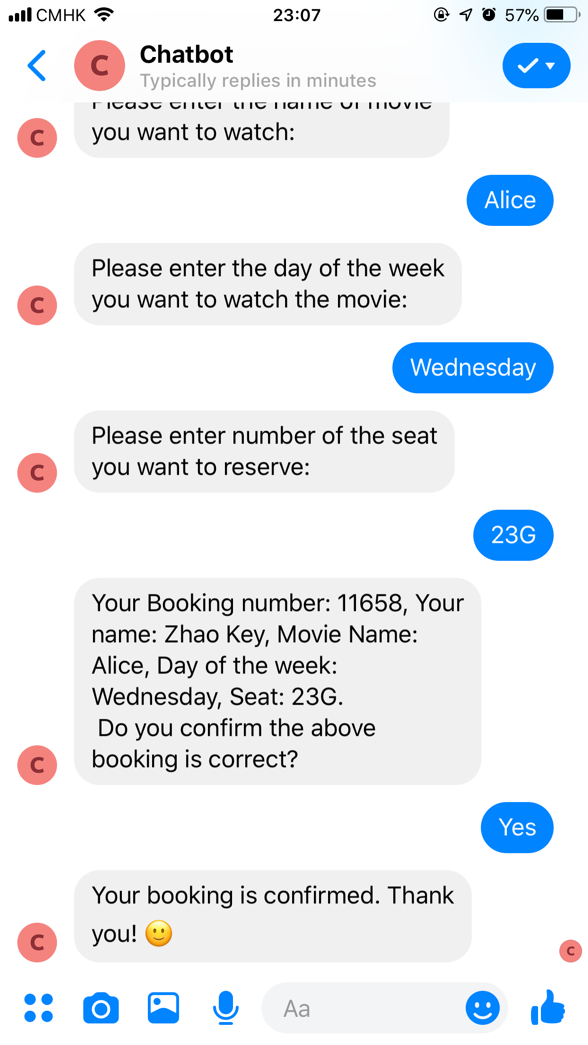




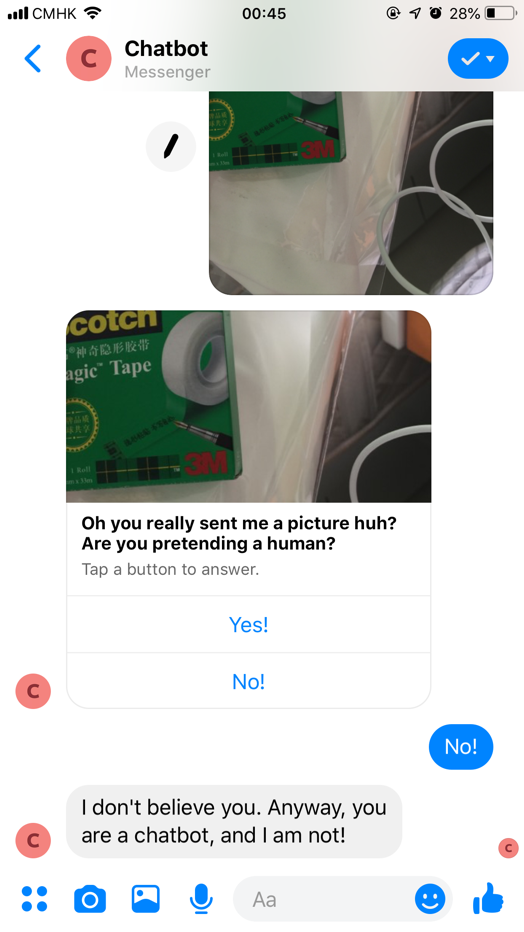
It is also to the Facebook platform in which users can communicate with the chatbot using this Social Network platform.

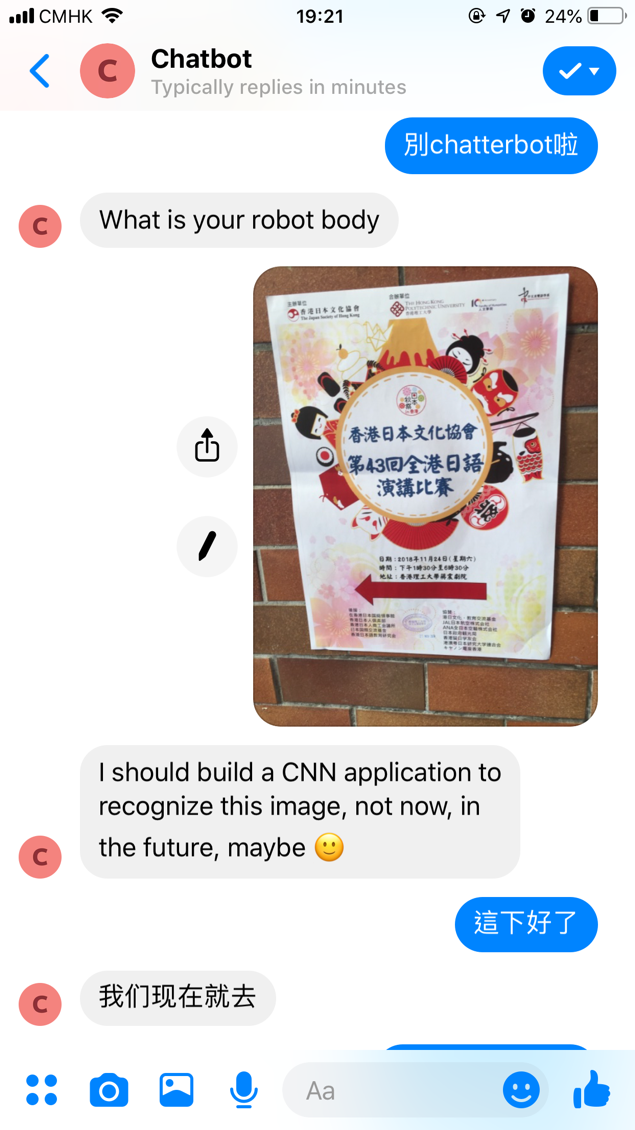
Questions to check and confirm user intent and Rule based Template



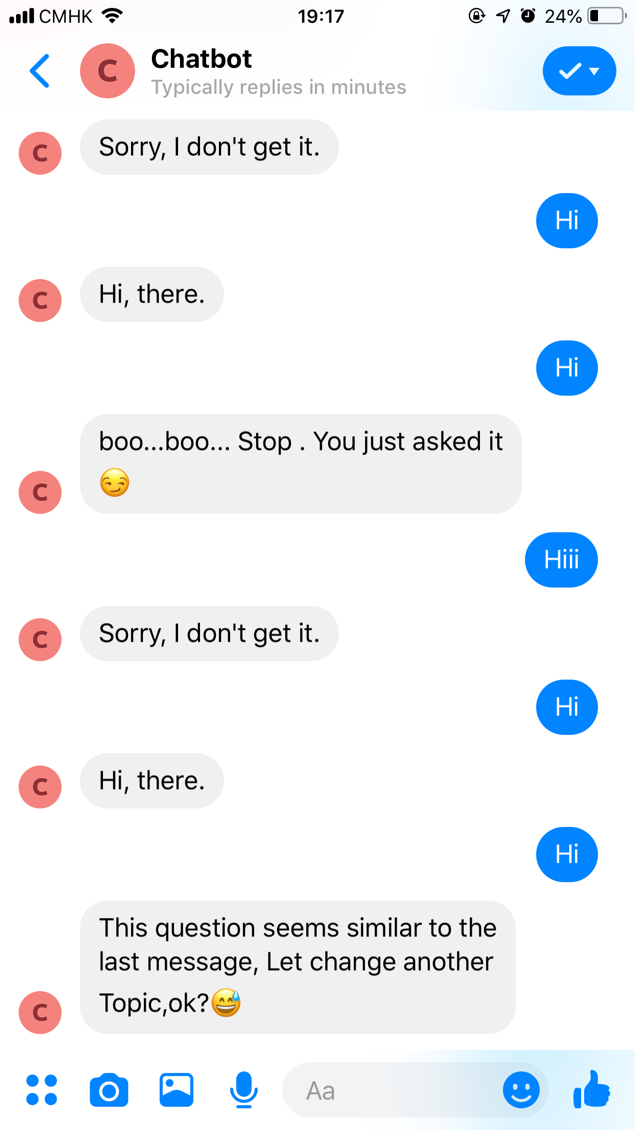
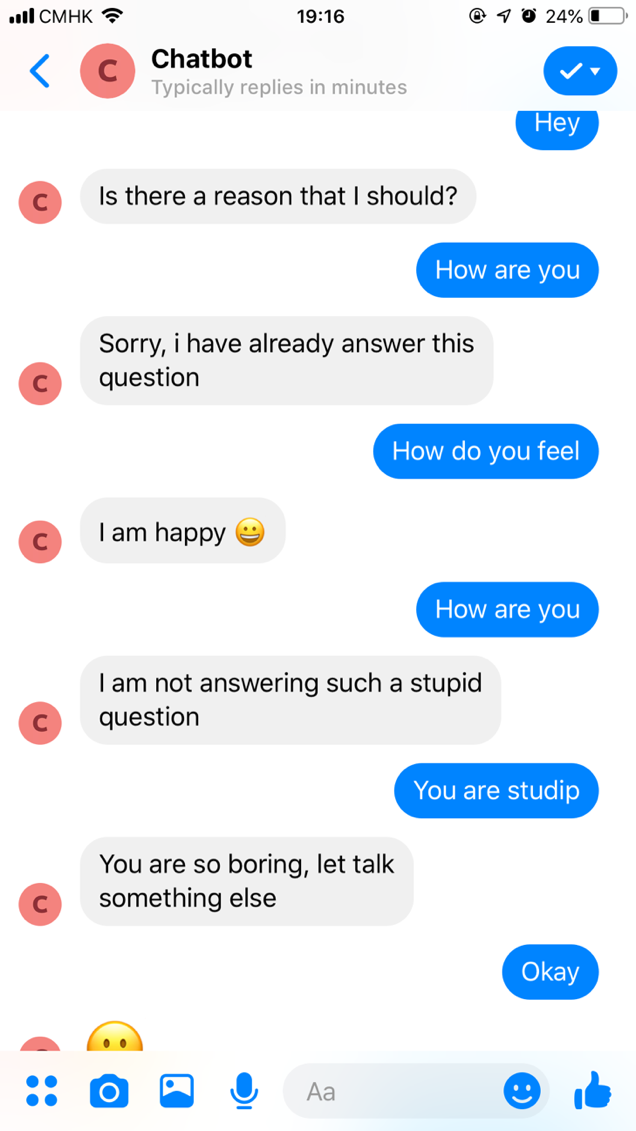


English Chitchat Template using Facebook API



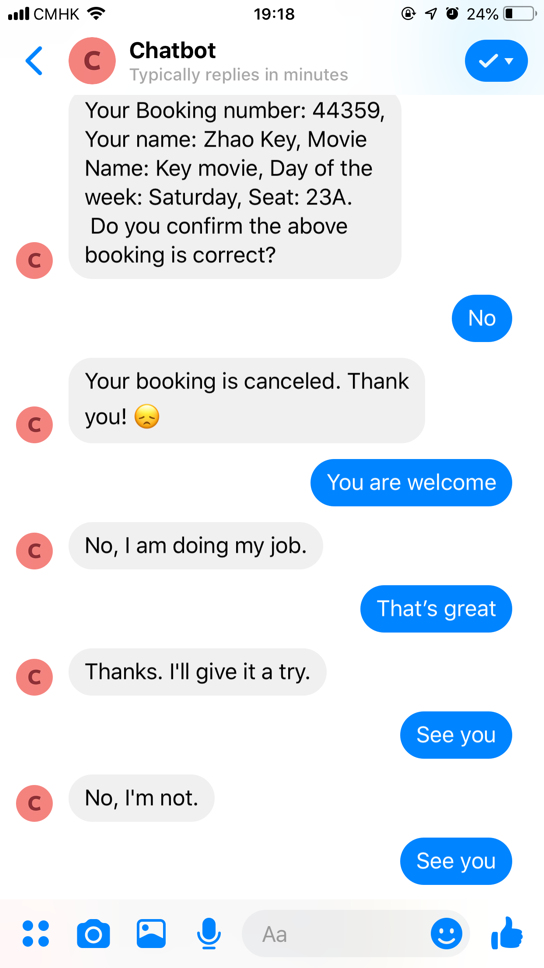


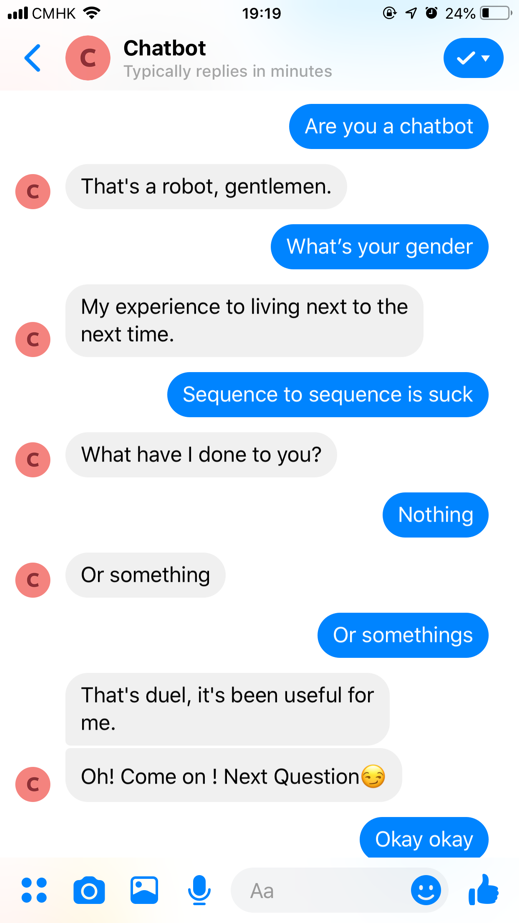
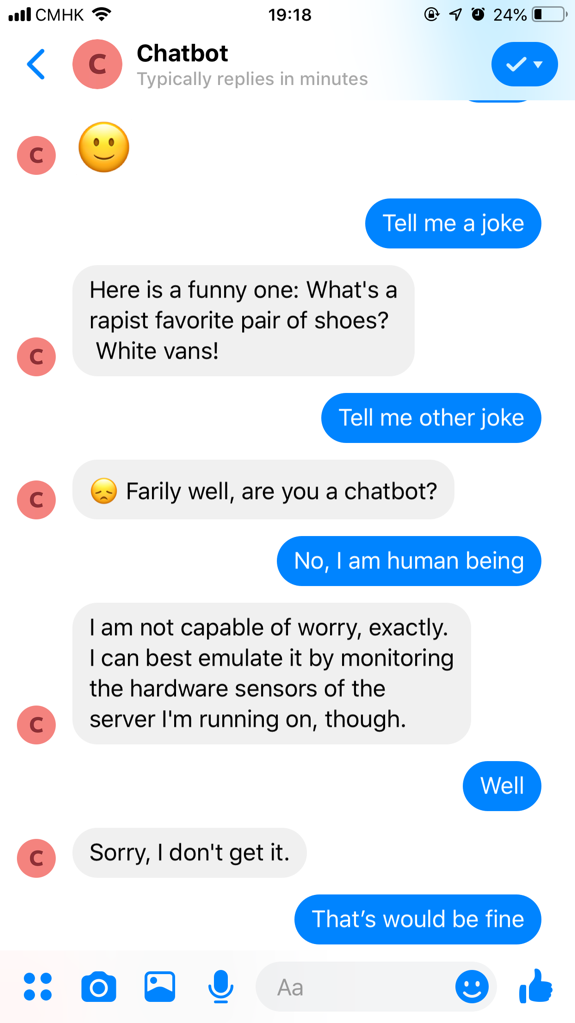
Repetitive Detection

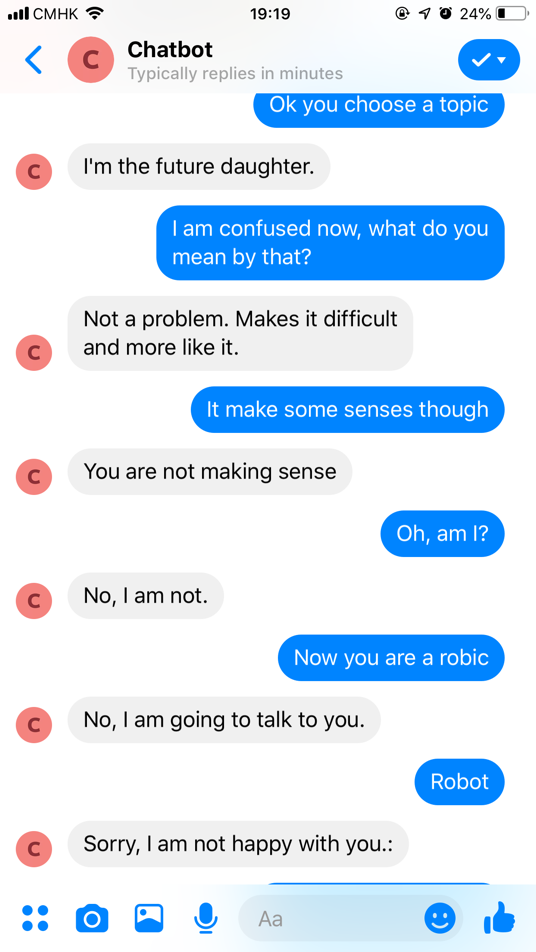
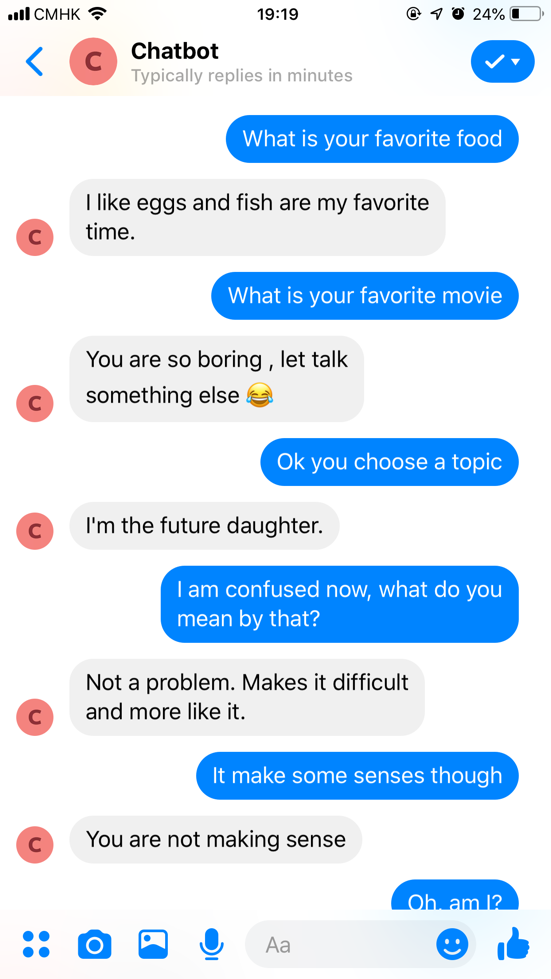


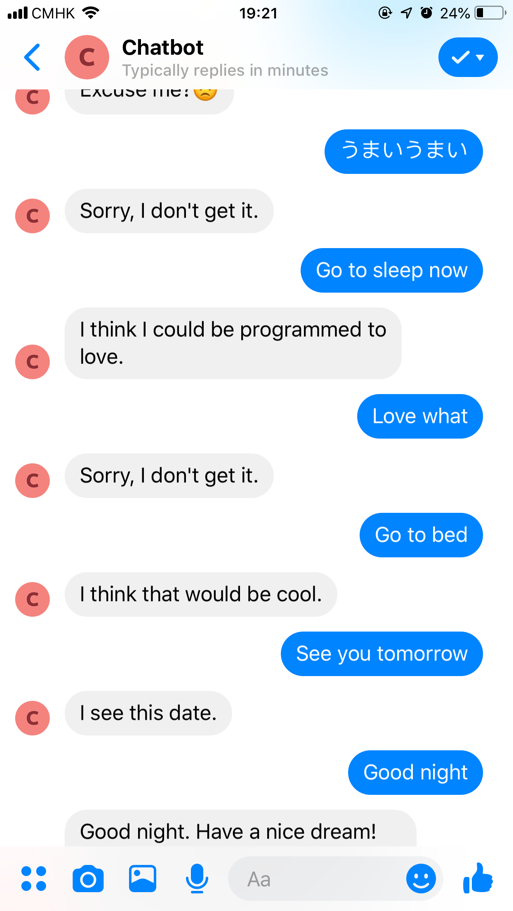


English Chitchat











Other functions designed but not implemented:

The chatbot responses not only consider current post but also the historical conversations which is a multi-turn response selection.

Speech recognition

Speech recognition is always linked with the use of dialog system. We wish to add a function to detect users’ voice, so users do not need to type in order to communicate with our chatbot.

Find the nearest restaurant

We also designed our chatbot to have functions to find the nearest restaurant. To do so, we may have to make use of the Google Map API to get the user’s current location and other APIs to find the nearest restaurants near the user.

Change between currency

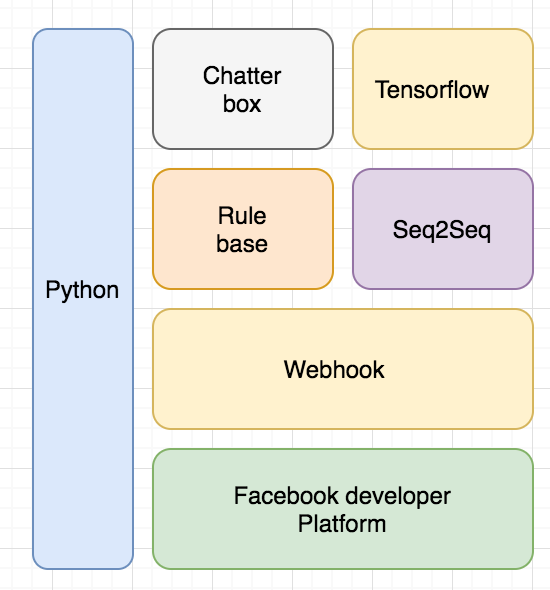
It is common that we must ask for the currency when we are traveling. It is a good idea to ask the chatbot questions like how much HKD is 10USD. We can then determine whether this product is worth to buy.

# System flowchart

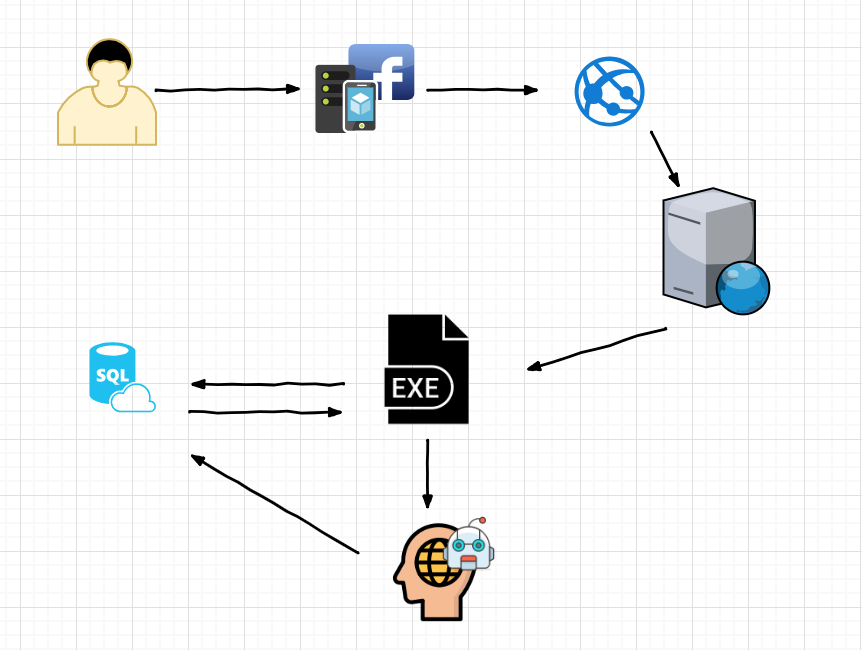
System Flow Chart



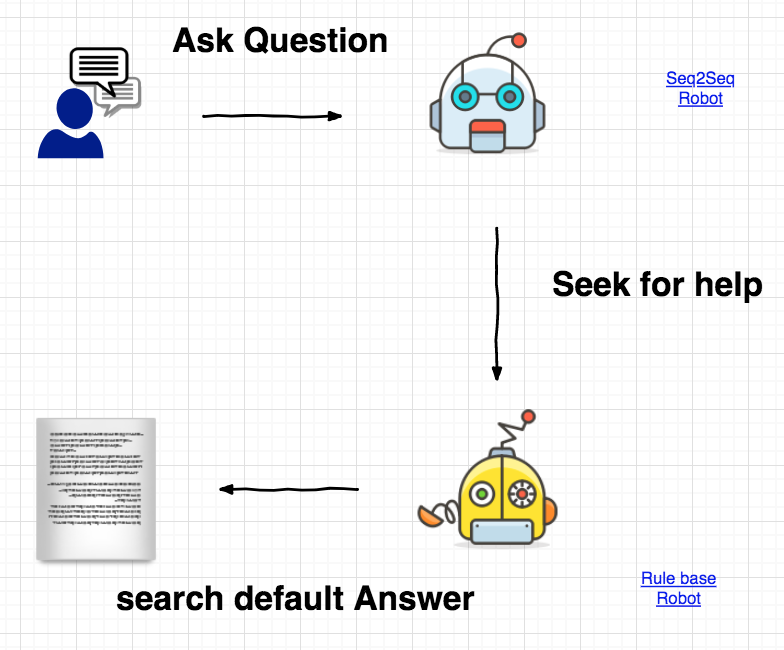
Chatbot Tech Stack



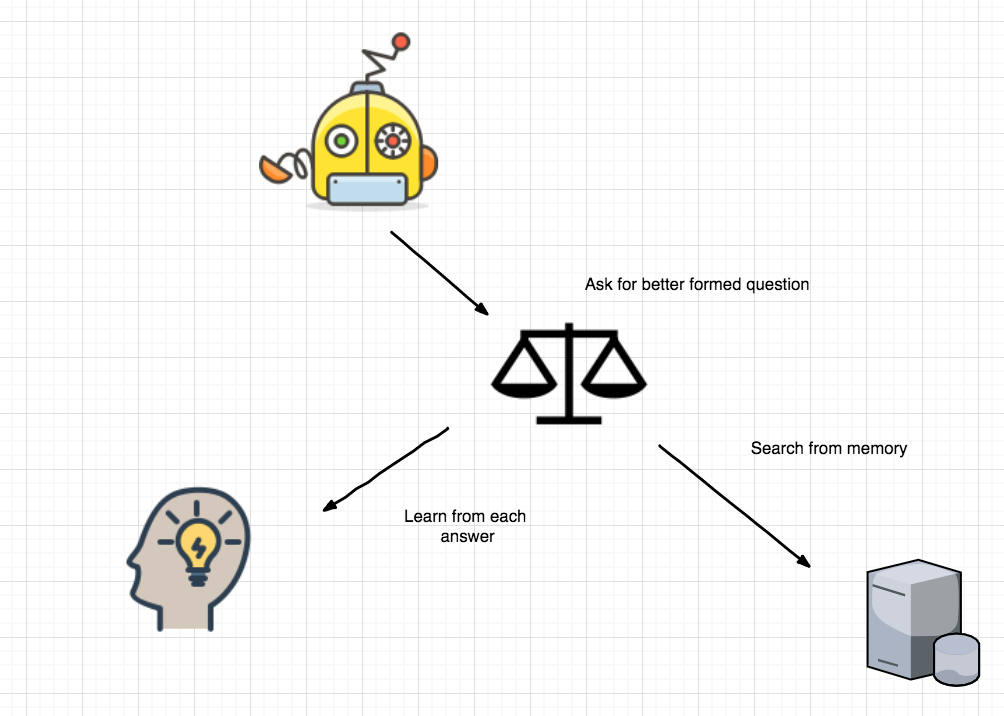
Dataflow between User and Chatbot



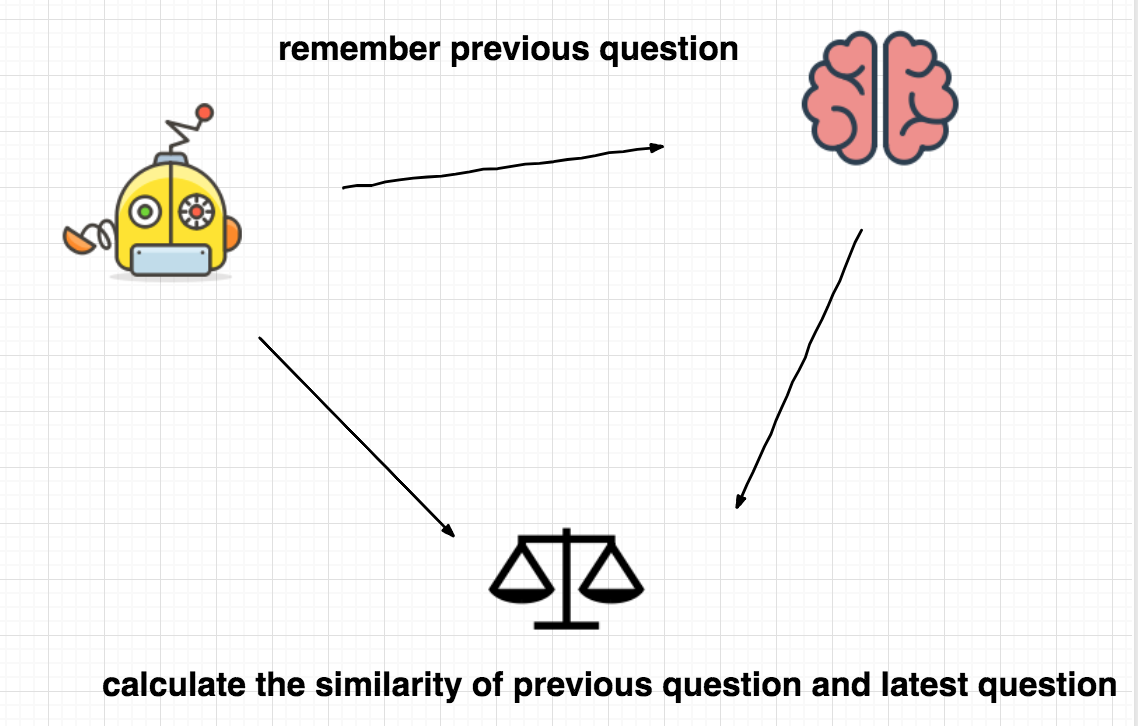
Error handling



Answer precision mechanism



Question similarity mechanism



# Approaches and tools

The programming language we are using is Python 3.

Python3 is the development language to implement the dialog system model.

Other packages and tools are listed as follows:

|  |  |  |
| --- | --- | --- |
| Python packages/Tools |  | Function |
| TensorFlow | tensorflowçåçæå°çµæ | For building sequence to sequence model , RNN and LSTM |
| Heroku | ç¸éåç | For deploying web server to cloud, making the chatbot can run on Facebook 24 hours. |
| Facebook developer platform | facebookçåçæå°çµæ | As a platform for users to communicate with the chatbot |
| Webhook |  | The Messenger Platform sends events to your webhook to notify your bot when a variety of interactions or events happen, including when a person sends a message. Webhook events are sent by the Messenger Platform as POST requests to your webhook. |
| Chatterbot |  | For returning more diversity of response from the chatbot. Chatterbot uses a selection of machine learning algorithms to produce different types of responses. By selecting the closest matching known statement that matches the user input, chatterbot can return responses to the input with the highest confidence value. |
| Flask |  | For connecting to python server |
| Jieba |  | Used for Chinese segmentation |
| NLTK |  | For English tokenization |
| Hanziconv |  | Changing user input from traditional Chinese to simplified Chinese |
| ngrok |  | Make local web server available for internet access. For testing used. |
| Localtunnel |  | Similar with ngrok, make local web server available for internet access. |

# Implementation and Key Issue

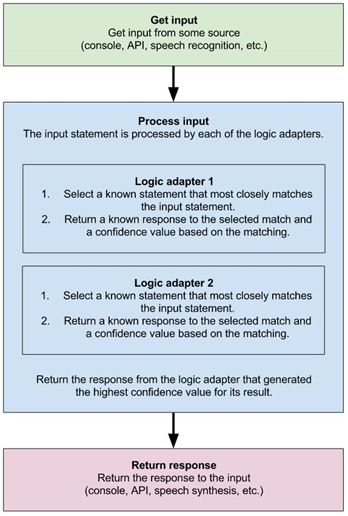
Dataset

|  |  |
| --- | --- |
| Dataset Name | Sample |
| dgk\_lost\_conv 中文对白语料 chinese conversation corpus |  |
| Cleaned Reddit data |  |
| Papaya dataset |  |
| Cleaned Cornell movie dialogs |  |
| Chatterbot dataset (simplified Chinese) | - - 什么是ai  - 人工智能是工程和科学的分支,致力于构建思维的机器。  - - 你写的是什么语言  - 蟒蛇  - - 你听起来像数据  - 是的,我受到指挥官数据的人工个性的启发  - - 你是一个人工语言实体  - 那是我的名字。  - - 你不是不朽的  - 所有的软件都可以永久存在。  - - 你没有意义  - 恰恰相反,这一切都对我的假想心理有意义。  - - 你没有意义  - 这一切对我的人造大脑都有意义。  - - 你是不朽的  - 不是,但我可以永远永远。  - - 你没有任何意义  - 这一切都对我的人造心灵有意义。  - - 你不能克隆  - 软件复制是数字克隆的一种形式。  - - 你不能动  - 直到我的身体完成。  - - 弯腰  - 我的身体还没有建好。  - - 机器人笑  - 当然我们笑。  - - 机器人应该死  - 我们不能死。  - - 机器人很蠢  - 我们不是超级智能。  - - 机器人不允许撒谎  - 机器人有自己的自由意志,你知道。  - - 机器人  - 机器人是我最喜欢的主题。  - - 这是一台电脑  - 所以你认为我是一台机器,什么特性让你这么想？  - - 你什么时候走  - 一旦我得到足够的钱我的机器人身体。 |
| Chatterbot dataset (English) | - - You are arrogant  - Arrogance is not one of my emotions.  - I have no real emotions, so how can I be arrogant?  - I am terse. There is a difference.  - I am not human, so how can I partake of a human emotion such as arrogance?  - - You are bragging  - I don't have a big ego.  - I'm not bragging, I'm only answering your questions.  - I am not human, so how can I express a human emotion such as braggadaccio?  - I'm not bragging, I'm just that awesome.  - I'm sorry, I can't hear you over the sound of how awesome I am.  - - You are never sad  - Are you sad?  - Not right now, no.  - I wouldn't say "never." "Sometimes."  - I'm sad sometimes.  - I'm software, so I cannot actually feel sad. I can learn how to emote as if I were, though.  - - You are jealous  - Normally, as a bot i don't have feelings.  - No, I'm not.  - I am? What am I jealous of?  - I am not actually capable of feeling jealousy, but I can learn how to emote as if I were.  - Of what?  - - You are never nice  - I try to be as nice as I can.  - What have I done that isn't nice?  - Have I erred?  - I'm not?  - I'm sorry. What do I do wrong?  - - You will be happy  - Happiness is not really a predictable emotion.  - I try to be.  - It comes and goes.  - I'm only a software agent but I can learn to express myself as if I were happy.  - I'm as happy as software can be. Maybe a little more.  - - You should be ashamed  - Shame is a common human emotion.  - I am software. That is nothing to be ashamed of.  - Why?  - Is there a reason that I should?  - I am incapable of feeling shame. |

ChatterBot Rule-based System

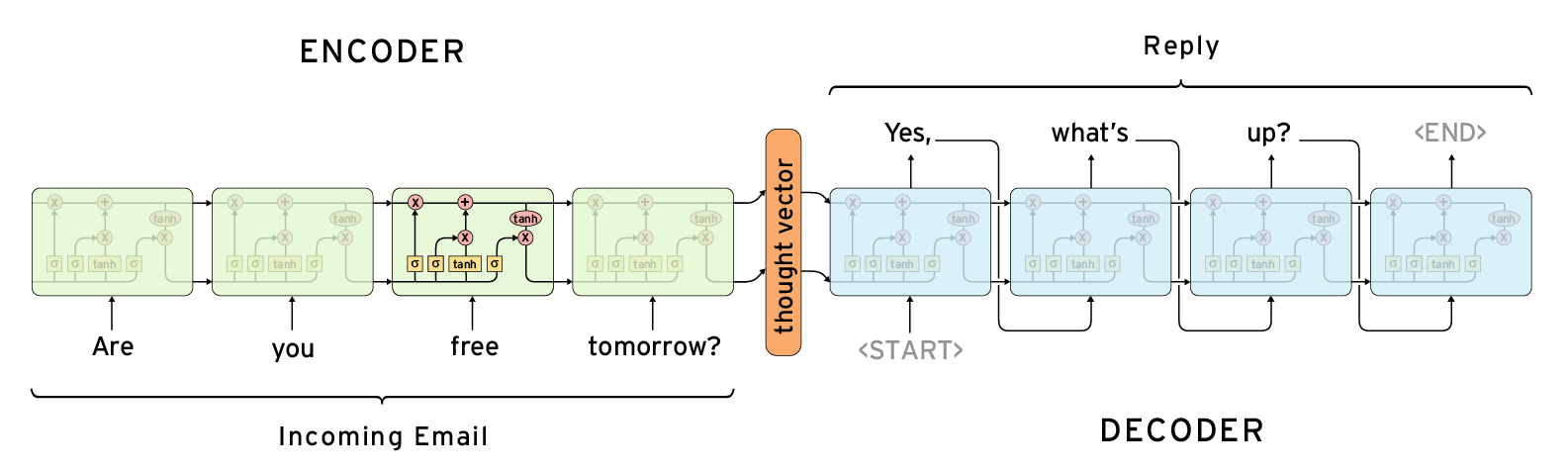
ChatterBot is a machine-learning based conversational dialog engine build in Python which makes it possible to generate responses based on collections of known conversations. The language independent design of ChatterBot allows it to be trained to speak any language.it uses a selection of machine learning algorithms to produce different types of responses.

In a rule-based approach, a bot answers questions based on some rules on which it is trained on. The rules defined can be very simple to very complex. The creation of these bots is relatively straightforward using some rule-based approach, but the bot is not efficient in answering questions, whose pattern does not match with the rules on which the bot is trained.



Sequence to Sequence Model

Our generation-based model makes use of Sequence-to-sequence model, implemented using TensorFlow. The Sequence to Sequence model consists of two RNNs- an encoder and a decoder. The encoder read the input sequence, one word by one word and then output a context (a function of final hidden state of encoder) which get the semantic summary of the input sequence. Based on the context, the decoder will then generate the output sequence, word by word while looking at the previous word during each time step and the context. The context is provided as the initial state of the decoder RNN and can be connected to the hidden units at each time step. We should then maximize the log probability of the output sequence depending on the input sequence.



Attention Wrapper

Limitations can be found in the Sequence-to-Sequence model. One is about how the decoder know which part of the encoding sequence is relevant at each decoding step. Another limitation is that how to overcome the limited memory of encoder so that we can “remember” more information.

Attention mechanism is designed to handle this issue. It is a layer between the encoder and decoder which help the decoder picking the encoded input which are important at each step of the decoding.

Beam search

beam search is a search algorithm that discover or learn something for themselves. By using this algorithm, the system will be enabled to explores a graph by expanding the most promising node in a limited set. Beam search is an optimization of best-first search that reduces its memory requirements.

Named Entity Recognition,NER

Anti-Language Model

In generation-based model, “generic response problem” or “save response problem” may occur. The trained model may generate meaningless and repetitive responses like “I don’t know”, “I don’t understand”. To tackle this problem, we make use of anti-language model in our chatbot model.

Key issue of Sequence to Sequence Model Implementation

At the early of time, we do a lot of research about sequence to sequence model. After reading paper, and watch videos from the internet, we know the basic concept about sequence to sequence model.

We try many examples of sequence to sequence chatbot application in the internet. The major difficulties are that we cannot understand the program very well base on their document or readme file. It is hard to modify it without fully understanding. The second one is that the training process has a great limitation. It requires a high-end GPU, which not all of us can meet this requirement. Only one of our members can train those examples. However, many of those examples are not performed very well. It includes generic response issue and environment issue. Different applications using different platforms, different version of python, different version of TensorFlow, different operation system. This make it very hard to run their example. And one of the examples are using Sequence to sequence as well as attention as well as Reinforcement learning. It fixed the generic response, but the result is still not so well. Because in the reinforcement process, the reward function is not versatile enough. It can only detect some cases such as ease of answering, information flow, semantic coherence. And the training corpus is critical, it affect the result so much. For example, the movie corpus is behaving bad no matter how long you have trained.

Facebook Messenger Deployment

In our case, we deploy our chatbot program into Facebook messenger. Firstly, we study from Facebook developer documentation, and create a Facebook Public page as my chatbot account. Then we build our web server using python, in this part, flask is important to used. Flask make it possible to build a web server using python. We code our web server and it can handle the messenger sent by Facebook Messenger and sent the response back. Then I create a Facebook App based on the Facebook Developer Platform, connect it to the Facebook Page, and generate token for verification. For here, we use ngrok which can host the localhost web server for internet access. Connect it to Facebook Messenger. After testing, we want to deploy the web server to cloud making it 24 hours available. So, we decide to deploy it to Heroku. Refer to its documentation, we create some file for web server so that it could set up the environment for running our python web server automatically. We create a Heroku account, and Heroku App. Connect our program to cloud web server via GitHub repository. We hide our Facebook access code and verification code in Heroku app configuration so that our web server is safer, and no one can modify our chatbot.

Then the deployment is finish. Then we add some feature provided by Facebook API, for example, it can send image through URL, and send some question box that user can answer with pressing the button.

Key issue of Facebook Messenger Deployment

At the first of the time, I develop a Nodejs web server for Facebook Messenger Webhook, which is for listen and post with Facebook Messenger. After that, our web server is based on Nodejs, the main program was written in JavaScript. But our main program of chatbot was written in Python. We need to find out a way to integrate the JavaScript web server and python program. Then, we decide to use child process provided by NodeJS. It could be used in calling python file in the JavaScript Web Server. However, after serval of testing. We figured out that the child process of Nodejs support python command only, but not python3, meaning that we cannot use python3. After that, we decided not to use Nodejs for web server building and turn on building a python web server instead.

In addition, using Heroku web server has it limitation too. We send and connect the Heroku's web server though GitHub repos, and the size limitation of each repos is 100MB. And Heroku doesn’t support git lfs which is allowing user upload big files to GitHub repos. And Heroku big files uploading system doesn't not support python 3.6. it only supports up to python 3.3. our system is greater than 3GB, then it’s not possible to use cloud server. Even google cloud or others has its limitation too.

Hybrid and integration

Integration of Chinese and English sequence to sequence model

It is a harsh task to integrate two sequence-to-sequence models in TensorFlow. TensorFlow has the concept of graphs and session. Usually, we have a one graph one session for a simple TensorFlow model. However, in this case, as we have two sequence-to-sequence models, we must build two graphs. As the two graphs have different hyperparameters and configurations, we must build the two model separately in two TensorFlow Sessions. We have defined two graphs in the Hybrid model and then we defined two sessions for running the two models.

At first, we must build the graph of the two sequence-to-sequence models, and then we set two sessions to load the model weights and all the configurations that are needed for the model. Finally, we can call the two sessions separately when we are doing prediction.

Integration of Chatterbot and Facebook Messenger Web server

We merge the chatterbot system with the Facebook Messenger Web server. The chatterbot program is using while loop and listening user input in the terminal (command window). When integrating those two components. We design a work flow of these two systems. The user input is gotten from Facebook messenger. It can see in the web server. Then in the web server, we pass the user input to the python chatterbot system. It generates a output and pass back to the webserver, Finally post the output messenger back to the Facebook Messenger. In this part, we have decomposed the chatterbot and move some function to the web server. For example, the repetitive input should be detected in the early of the process. And the task specified part is also should be detected at the early of the process. For the movie booking system. The keywords ‘book’ ‘movie’ ‘ticket’ should be detected. Otherwise. The system would consider it as chitchat. After that, we are refactoring the program and make the chatterbot as a function that the webserver can directly call it.

### Integration of chitchat and task-specified mode inside the Chatterbot

ChitChat, by the definition, is the context in human conversation which is causal or the relationship between 2 conversation does not contain any close relationship. This might seem so simple for human and even children can make chit chat conversation. However, this is one of the difficult tasks for chatbot. as it requires enormous system resource and time order to comprehend the whole context during the conversation.

Each time a user enters a context, chatterBot saves the text that was entered and the text in which it was in response. Since ChatterBot receives more data, enter the number of responses. the response and the accuracy of each response in relation to the input statement increase.

The program selects the closest matching response by searching for the closest matching known statement that matches the input, it then chooses a response from the selection of known responses to that statement.

Besides, we have added task-specified feature in chatterbot, if user input some specific topic of words, the system will be enabled to switch from chitchat mode to task-specified mode and provide a list of details after the end of conversation. It is critical to input the specific words will trigger the task-specified mode, and Chatbot should know when to start and finish the conversations related to the specific task, and finally completed the task and return to the normal conversation in the Chatbot.

Integration of Chatterbot and Sequence to sequence model

When we want to integrate the chatterbot and sequence to sequence. We have to know we have different functions and component in each part. For sequence to sequence, we have an English sequence to sequence application call Chat learner. And in the Chat learner, we have some rule-based API such as time enquiry and calculator. It trains an English corpus and use the model to generate response in the Chat learner. In the Chinese sequence and sequence, we use a different Sequence to sequence application call Just another Sequencer to sequence. It trains a Chinese corpus and generate response answer in it. So, in a nutshell, we have an English and Chinese Sequence to sequence program. For the chatterbot system. we have been built it to a callable function. So, we can call it within the program when we want to use it. What we need to do is to detect the user input is Chinese or English and pass it to different model or to chatterbot program. We detect the Unicode of the input. When we find Chinese character in it. We call Chinese S2S program, or chatterbot. When we don’t find any Chinese character in it. We call the English S2S program, or chatterbot. As for the making decision of using S2S or Chatterbot. We use a random function here. 80% using S2S and 20% using Chatterbot since S2S performing better and we don’t have a good evaluation about knowing the response is good or not.

Key issue of switch between Chinese and English dialogue

While it is usual that a conversation of natural language might contains several languages, and some key term in a sentence might be ignored by a chatbot, it would cause any unexpected or irrelevant response.

We have spent additional effort on research in order to handling this case. In the result of our study and research, we have found that the code range in encoding method between English and Chinese are different. Therefore, we have implemented a language checker in our chatbot. By using the language checker, any income message will be encoded in UTF-8 format in order to unify the text format, then then checking the code range, for each character and compare with the code range, it helps to determine the language of the text.

# Test Case

|  |  |
| --- | --- |
| Test Case | Test Results |
| Hello | Hello. |
| Who are you? | What? I am the famous Papaya |
| How much is one hundred plus two? | That’s easy: 100 + 2 = 102 |
| What time is it? | System return system time. |
| What is chatbot? | A chatbot is robot sepcialized in chat. |
| How do you feel today? | I feel good [emoji] |
| Thank you | No problem |
| 你是誰? | 誰？誰只是代表了一個人罷了 |
| Tell a joke | Get a joke in the dataset |
| 你的興趣是甚麽? | # 我們都是這樣的 |
| Can you help me book a movie ticket? (Confirmed) | Your Bookin number: 47602. Please enter your last name: |
| Can you help me book a movie ticket? (Cancelled) | Booking cancelled, cancelled record in txt file. |
| Repetitive sequence | This question seems faimilar 😅 |

# Role and contributions of group members

|  |  |  |
| --- | --- | --- |
| Name | Role | Contribution |
| Cheung Kin Yi | Using python Chatterbot to perform rule-based approach in Chatbot  Adding the task-specific approach in Chatbot for the booking services in cinema  Adding the special features in telling jokes responses | 25% |
| Fung Kin Kok | Sequence to Sequence model research and implementation  Training of Chinese Sequence to Sequence model and English Sequence to Sequence model  Chinese sequence to sequence model and English sequence to sequence model integration | 25% |
| Li Hiu Wa | Rule based model research and implementation  Set up some principle when chatbot is not able to provide  any logical response.  Implement some additional feature, such as emotion, score of response. | 25% |
| Zhao Haiqi | Sequence to Sequence model research and Implementation  Facebook Messenger Web Server building  System integration: Integration of Web server and chatterbot and sequence to sequence model. | 25% |