KULLIYYAH OF INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

FINAL YEAR PROJECT REPORT

AI CHAT-BOT FOR CUSTOMER RECOMMENDATION

TABASSUM TAHERA
1413986

SUPERVISED BY

DR. RAWAD ABDULKHALEQ ABDULMOLLA

DECEMBER 2018
SEMESTER 1 2018/2019

FINAL YEAR PROJECT REPORT

AI CHAT-BOT FOR CUSTOMER RECOMMENDATION

by

TABASSUM TAHERA
1413986

SUPERVISED BY
DR. RAWAD ABDULKHALEQ ABDULMOLLA

In partial fulfillment of the requirement for the
Bachelor of Computer Science
Department of Computer Science
Kulliyyah of Information and Communication Technology
International Islamic University Malaysia

December 2018
Semester 1 2018/2018

ACKNOWLEDGEMENTS

Praise and thanks to Allah first and foremost whose blessing enabled me to accomplish this project.

I wish to express my deepest appreciation to my supervisor Dr. Rawad Abdulkhaleq Abdulmolla for relentless guidance, helpful suggestion, close supervision and moral encouragement to complete this task.

A special thanks to my parents and to all my teachers I have had. My sincerely thanks to all those whom directly or indirectly help me to complete this project.

ABSTRACT

As, an industry tremendous in IT products, Microsoft manufacture a huge number of electronic products, such as computers, tablets and so one. Many of these products are similar in respect of hardware like CPU power, RAM size, which is very tough to choose the desirable hardware for the customer who doesn’t have much knowledge. Although, they can’t directly talk to the salesman for clearing their confusions. To solve this problem, this project will help those customers by developing an online chatbot that will be able to talk to the customers and recommend the best electronic products they want.

TABLE OF CONTENTS

LIST OF TABLS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
CPU	Central Processing Unit
DBMS	Database Management System
LUIS	Language Understanding Intelligent Service
NLP	Natural Language Processing
RAM	Random Access Memory
SDK	Software Development Kit
SQL	Structured Query Language
SVM	Support Vector Machine
QnA	Question and Answer
ML	Machine Learning

LIST OF APPENDIXES

CHAPTER ONE

INTRODUCTION

1.1 Background

Natural Language Processing

Natural Language Processing (NLP) is the study of letting computers understand human languages [3]. Without NLP, human language sentences are just a series of meaningless symbols to computers. Computers don't recognize the words and don't understand the grammars. NLP can be regard as a "translator", who will translate human languages to computer understandable information.

Traditionally, users need to follow well-defined procedures accurately, in order to interact with computers. For example, in Linux systems, all commands must be precise. A single replaces of one character or even a space can have significant difference. However, the emergence of NLP is changing the way of interacting. Apple Siri [4] and Microsoft Cortana [5] have made it possible to give command in everyday languages and is changing the way of interacting.

In this project, we are going to use the API provided by Microsoft called Language Understanding Intelligent Service (LUIS)[6]. It contains a bunch of well-developed REST APIs. We will take time to read the documentations and more details will show in future reports.

Machine Learning

Machine Learning (ML) is an area of computer science that "gives computers the ability to learn without being explicitly programmed"[7]. The parameter of the formulas is calculated from the data, rather than defined by the programmer. Two most common usage of ML is Classification and Regression. As shown in figure1[8], Classification means to categorize different types of data, while Regression means to find a way to describe the data. Basic ML program will have two stages, fitting and predicting. In the fitting stage, the program will be given a large set (at least thousands) of data. The program will try to adjust its parameter based on some statistical models, in order to make it “fit” the input data best. In the predicting stage, the program will give a prediction for a new input based on the parameters it just calculated out. For example, the famous Iris flower dataset [9] contains the measurement of several features of three different species of flowers, such as the length of sepals and petals. A well-defined ML program can learn the pattern behind this feature and give prediction accordingly.

In this project, we will use the Microsoft Azure Machine Learning Studio [10]. It is a platform for implementing and testing ML models. The backend is on Microsoft’s cloud service, which make the calculation much faster than personal computers.

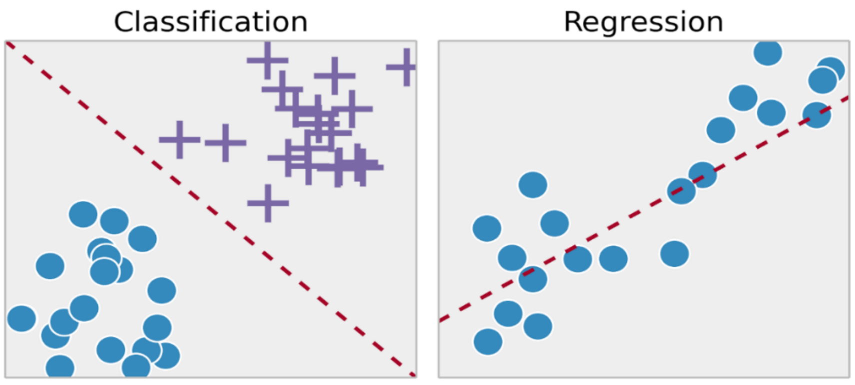


Figure – 1: Machine Learning

1.2 Problem Statement

The advancement of innovation provides people the more decisions with respect to different electronic devices, likewise it builds their confusions same time selecting the correct device. A lot of customers barely recognize the differences between tens and models and choose the best for them. Moreover, the online shopping site also decreasing the chance for them to look for help from shop assistants. To overcome from this problem, the project AI Chat-Bot for Customer Recommendation will help. The goal is to build an AI chatbot that help customer to choose their products. The bot will be accessible from Skype and Facebook online store. Customers can directly talk and acquire recommendation. They can feel like they are talking to a human shop assistant.

1.3 Project Objectives

We will do a prototype of a bot that tries to get all the information including the preference on CPU, tentative usage, expect price range, etc., and make recommendation if all the information is collected. This was a prototype that integrates all the APIs and web service, as a deployable web service. However, our requirement was changed to a more user-friendly way. Rather than asking all the requirements and information one by one to the customer, this time we just let the customer to directly point out their intent by sentence like “I want a computer to play large 3D games” and give recommendation sparsely. The devices are categorized into different groups, and the recommendation logic is just a simple link between intent and groups.

1.4 Project Scope

We will do a prototype of a bot that tries to get all the information including the preference on CPU, tentative usage, expect price range, etc., and make recommendation if all the information is collected. This was a prototype that integrates all the APIs and web service, as a deployable web service. However,
our requirement was changed to a more user-friendly way. Rather than asking all the requirements and information one by one to the customer, this time we just let the customer to directly point out their intent by sentence like “I want a computer to play large 3D games” and give recommendation sparsely. The devices are categorized into different groups, and the recommendation logic is just a simple link between intent and groups.

In this project, we will be going to build a LUIS model for English language, because English is the most widely used language in the world. For this project, in order to improve accuracy, we are going to restrict the query types to *“Purchase”* and *“Complaint”*. They will be handled differently but be analyzed in the same language model. For a purchase intention, the bot will analyze what type of device the customer wants to buy and give recommendations. For a complaint recommendation, on the other hand, the bot will only save the contents of complaint into database for a specialist to handle. In this project, we are not going to give specific response to complaint. The device for recommendation is should be a dynamic database reflecting a real time price and specification of all the available devices. The bot will be going to be deployed in three different places, including Bot framework portal, Skype and Facebook messenger. Other platforms, e.g., Kik, Slack are optional and depends on future usage of Microsoft side.

1.5 Significance of the Project

Different from the bots widely used on the internet, ours will stand out because of three reasons. First, current bots usually communicate with users by giving multiple choices at each step and letting users to select from them. It is hard for a customer to clearly describe his requirement in this rigid way. Moreover, if a customer can answer each question, that means he already have a clear idea about what to buy, and thus no need to talk to the bot at all. Instead, we leverage the technology of natural language processing, such that the user can really “talk” to the bot and explain his requirement little by little.

Second, current bots usually have a pre-defined decision tree. It is hard for a designer to take every scenario into consideration because of the limitation of human mind and the increasingly fast evolution of technology. Instead, we leverage the technology of machine learning. The bot will learn to deal with different conditions from a huge amount of real-world data. The bot can also update itself while running, which can help it to keep it up to date.

Third, our bot can be accessed from various platforms, including Facebook Messenger, Skype and Microsoft online store. User can access the bot from anywhere using anyway. The purchase history and preference will be recorded for registered users in order to give customized recommendations.

The ultimate goal of the project is to build an easy to use talking bot that can talk with customer and provide recommendation based on their requirements. However, what is “easy to use” is negotiable. There is a tradeoff between the *recommendation accuracy* and the *concision of conversion*. If we want to make a more accurate recommendation, the only way is to ask more question to the customer and gather more requirements. Imagine when you go to a shopping mall and ask the shop assistant to recommend some items for you. She will ask a lot of questions about your usage, your expectation of price and even your preference of color before he can make recommendations for you.

This is the same for the bot. Although the machine learning behind the bot can extract useful information from past customers by learning the feature and relationships, the bot cannot give useful recommendation if it does not know about the customer. In order to make a good recommendation, besides a good machine learning algorithm and the abundant training data, the bot must have the data about the customer. For a registered
customer, some of the information may be already saved in the database, but for a new customer, the only way is asking questions.

It will be annoying if the customer simply wants some basic recommendation, but the bots asking a lot of lengthy questions, without prompt response, making the conversation last for minutes. Furthermore, the customer may not know how to answer some questions (e.g., “what CPU frequency you want, xx GHz or yy GHz”) or may not want to answer some questions (e.g., “what is your monthly income”) because of privacy reasons. An experienced shop assistant can discover the embarrassment immediately but the bot, on the other side in the network can never identify it, until the customer typed in “I don’t know” or “I don’t want to answer this question”.

CHAPTER TWO

REVIEW OF PREVIOUS WORKS

2.1 AI BOTS

AI bot (or Chatbot) is a computer program that conduct conversation via audio or textual messages and is widely used now. It typically uses Natural Language process techniques to analyze input sentence and generate outputs. There are currently two trends in the development of AI bots. The first one is to be as Human-Like as possible and try to pass the Turing Test. One of the most famous contests in this area is the annual Loebner Prize.

Another trend is trying to provide help to users, with users knowing that definitely they are talking to a bot and the bot can help them finish some specific jobs. Apple’s Siri, Microsoft’s Cortana and Google now can be categorized as this type. Because the conversation topics are limited in a relatively narrower area, they can achieve an acceptable accuracy and can be used in daily life.

Chatbots are increasingly being used to augment customer service. These technology applications use AI to process language, enabling them to understand human speech. They can decipher verbal or written questions and provide responses with appropriate information or direction. Many customers first experienced chatbots through dialogue boxes on company websites. Chatbots also interact verbally with consumers, such as Siri on the Apple iPhone or Amazon’s Alexa Voice Service.

Automated technology that recognizes and responds to speech has been around since the 1960s.2 Although chatbots don’t always get things right, their accuracy today is much improved from early versions. Recent advances in AI, data processing and machine learning have made chatbots much more effective. Using AI to analyze customer information and similar queries, chatbots today can “think” rather than simply provide answers from a pre-selected list of responses, making interactions with them more accurate and satisfying.

Customer interaction through chatbots is seen by many experts as the communication channel of the future. Chatbots are always “on” to provide immediate answers at any time of the day, making them quicker and easier than email or the telephone to contact a company.

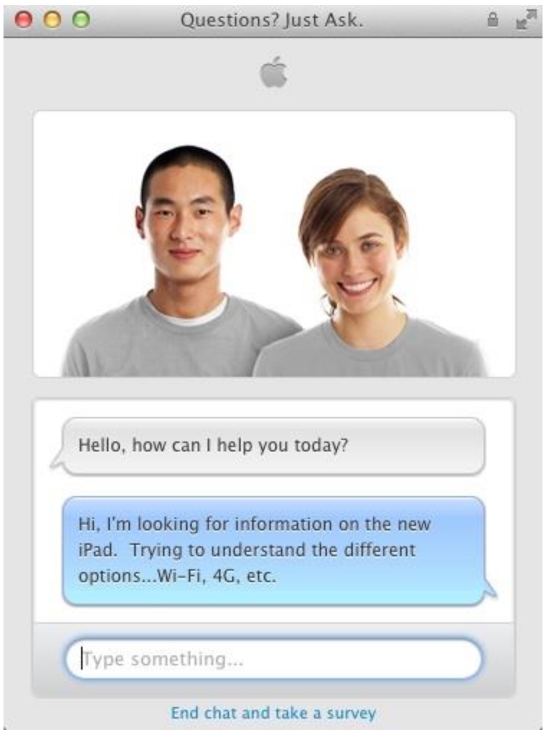
The technology analyst firm Gartner estimates that by 2020, more than 85% of customer interactions will be managed without human contact.4 Chatbots can interact over a range of technology, including laptops, computers, tablets and smartphones. Because they are able to listen and learn, they can analyze fine details and respond quickly. Their technology has advanced to take on more challenging issues to help build customer relationships or calm down an irate customer.

Chatbots can be customized to take on a company's brand personality and voice by linking to existing communications systems and corporate knowledge bases. They can be designed to interact with customers directly in everyday language.

A major benefit of integrating chatbots into an overall customer service framework is that they can allow human customer service representatives to have more time to address complex questions, problem issues and other "high-touch" interactions with consumers.

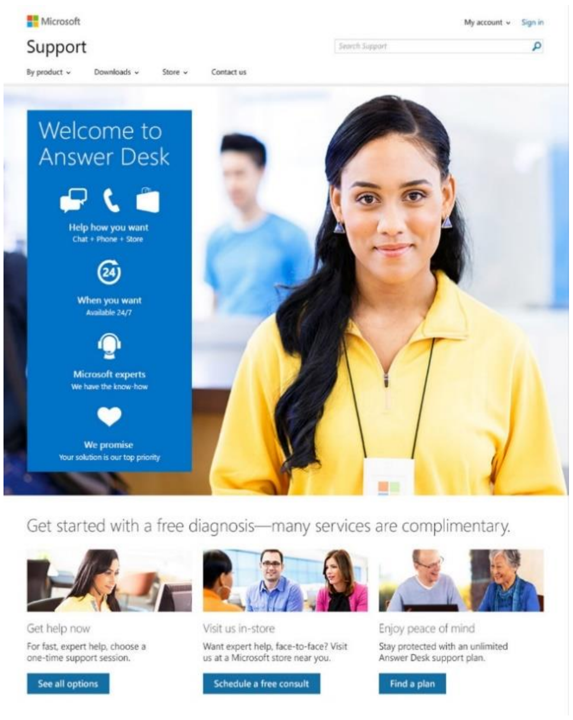
2.2 Customer Services

Currently, even big companies like Apple and Microsoft still haven't use AI bots in customer services. Figure 2 [2] and Figure 3 [3] shows the customer service interfaces for Apple and Microsoft. For both of them, they are actually annual work. There human beings on the other side of the chat box. As a result, the service hour is limited to regular office hours (9am-5pm and 9am-9pm for Apple and Microsoft accordingly). Moreover, one human being cannot serve multiple customers at a time and takes around 20 seconds latency for each question.



Hello, how can I help you today?

Hi, I'm looking for information on the new iPad. Trying to understand the different options...Wi-Fi, 4G, etc.



Get started with a free diagnosis—many services are complimentary.

For fast, expert help, choose a one-time support session.

Want expert help, face-to-face? Visit us at a Microsoft store near you.

Stay protected with an unlimited Answer Desk support plan.

Figure – 2 : Apple Online Support Figure – 3: Microsoft Online Support

AI bots are not widely used in customer service for several reasons. One of the most reasons are the accuracy. It will be annoying that one customer comes to ask for some emergency help, but the bot gives a wrong answer or cannot understand his idea after several rounds. Moreover, when people are talking, their expressions can vary tremendously, different from giving simple commands like "*set an alarm*". The project will be restricted in product recommendation and may have an acceptable accuracy is training data is enough.

CHAPTER THREE

METHODOLOGY

3.1 Design

This section covers the design, implementation and testing of different module of the bot, which contains the design of the Microsoft Bot Framework, Microsoft Azure QnA Maker, Microsoft Azure Bot Service and Microsoft Azure Machine Learning Studio.

Microsoft Bot Framework

Our bot is built on top of Microsoft's bot framework. It is a SDK provided by Microsoft to build general chat bots. With built-in classes to handle conversation. With the help of the SDK, we can focus on the logic of handling messages but forget about how the message is transferred between the chat box and the backend engine. It also defined a common gateway to connected with different front-end interfaces, such as Skype or Facebook.

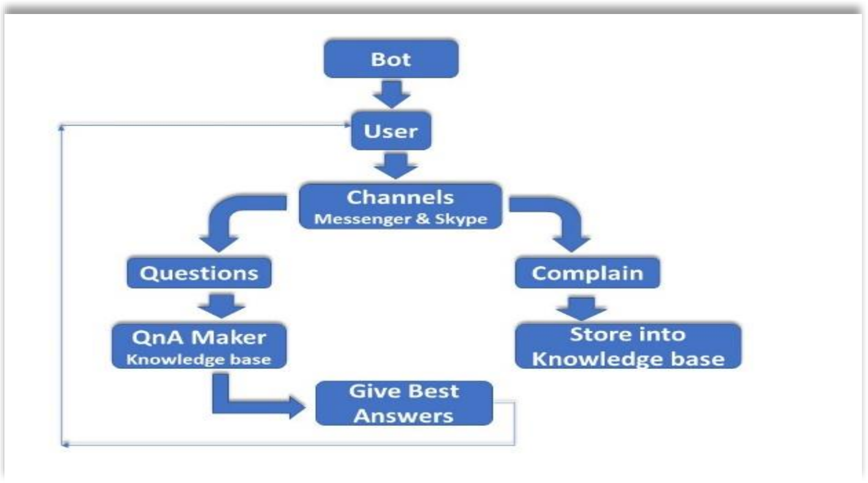


Figure – 4: Microsoft Bot Framework

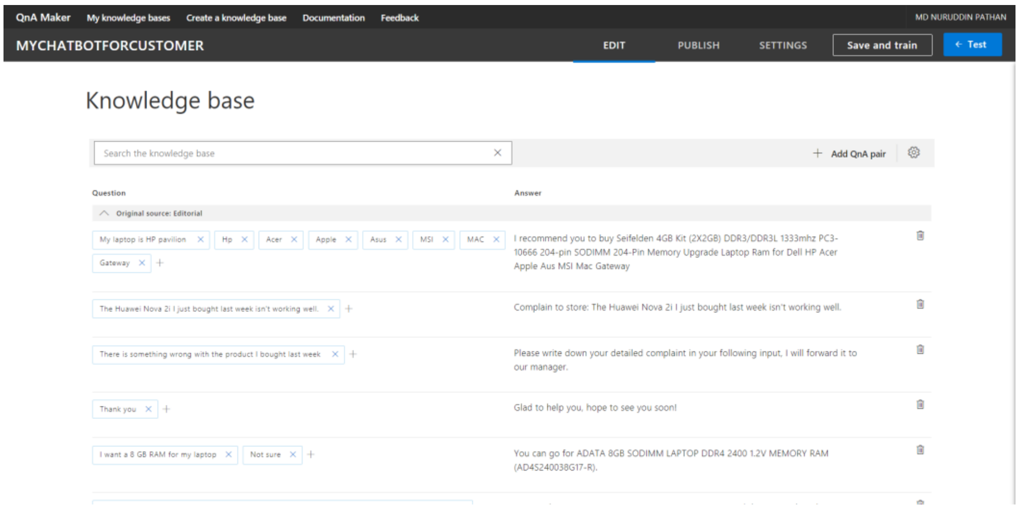
The framework will only responsible for transferring conversation contents. After receiving an object represents an input from user, I will be able to get the input string or images together with the metadata such as timestamp or user id. will freely analyze the input sentence and create another object representing output and send back to the user.

During our test in its own bot portal, the average latency is second level, a simple echo-server takes around 2-3 seconds to get response. While testing in Skype interface, the latency is around 2-6 seconds.

Microsoft Azure QnA Maker

QnA Maker is a cloud-based API service that creates a conversational question and answer layer over the data. QnA Maker enables to create a knowledge-base (KB) from the semi-structured content such as Frequently Asked Question (FAQ) URLs, product manuals, support documents and custom questions and answers. The QnA Maker service answers the user's natural language questions by matching it with the best possible answer from the QnAs from the knowledge base. After creating the KB, you must publish the KB so that the bot can interact with the KB and get the best answers for the users' questions. There are two important concepts in QnA Maker KB namely "question" and "answer". Questions describes what intention the speaker want to convey while saying the sentence. Answers are the useful information detected in the input sentence.

The QnA Maker has some limitation for adding database. To solve this problem, we are using Microsoft Azure Machine learning Studio to store more real-world data.



Knowledge base

Original source: Editorial

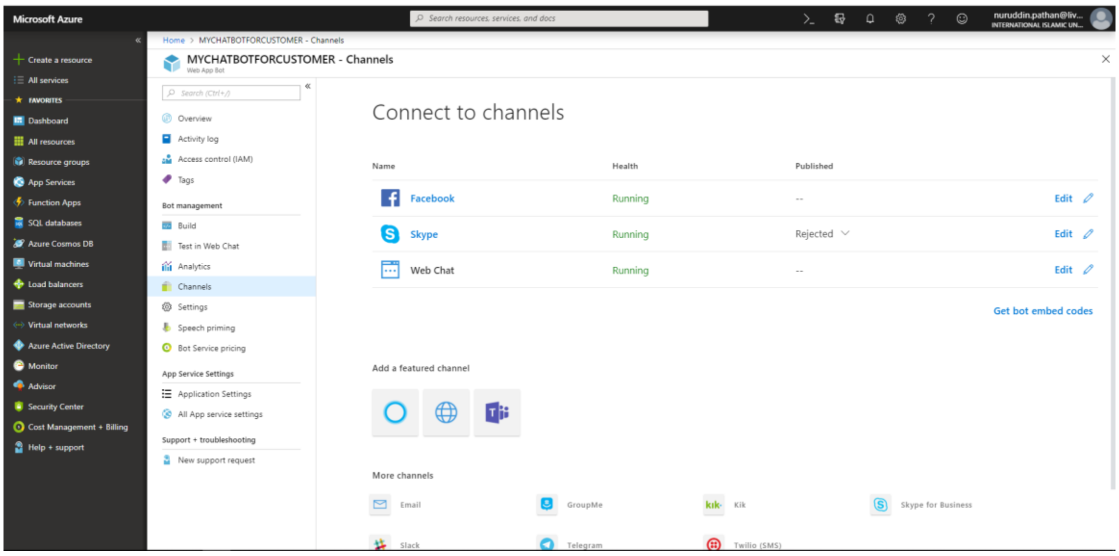
Question	Answer
My laptop is HP pavilion; Hp; Acer; Apple; Asus; MSI; MAC; Gateway	I recommend you to buy Seifelden 4GB Kit (2X2GB) DDR3/DDR3L 1333mhz PC3-10666 204-pin SODIMM 204-Pin Memory Upgrade Laptop Ram for Dell HP Acer Apple Aus MSI Mac Gateway
The Huawei Nova 2i I just bought last week isn't working well.	Complain to store: The Huawei Nova 2i I just bought last week isn't working well.
There is something wrong with the product I bought last week	Please write down your detailed complaint in your following input, I will forward it to our manager.
Thank you	Glad to help you, hope to see you soon!
I want a 8 GB RAM for my laptop; Not sure	You can go for ADATA 8GB SODIMM LAPTOP DDR4 2400 1.2V MEMORY RAM (AD4S240038G17-R).

Figure – 5: QnA Maker Knowledge Base

Microsoft Azure Bot Service

Azure Bot Service speeds up development by providing an integrated environment that's purpose-built for bot development with the Microsoft Bot Framework connectors and Bot-Builder SDKs. Developers can get started in seconds with out-of-the-box templates for scenarios including basic, form, language understanding, question and answer, and proactive bots.

For the bot I used Microsoft Azure Bot Service to make the bot. It has lot of channels that can connect to the bot and users can use the bot from different channels. For this bot I used Facebook Messenger and Skype. It has the web chat to test the bot whether it is working or need modifications.



Connect to channels

Name	Health	Published	
Facebook	Running	--	Edit
Skype	Running	Rejected	Edit
Web Chat	Running	--	Edit

Figure – 6: Microsoft Bot Service

Microsoft Azure Machine Learning Studio

Microsoft Azure Machine Learning Studio is a platform for users to build machine earning predictive models. It has a drag-n-drop graphical interface to represent then logic flow of a machine learning model. A trained model can be deployed as a web service, taking data as input and give predication based on the trained model. For example, the figure above is a sample model used to compare two models for detecting hand-writing English Characters.

The Azure Machine Learning Studio Model contains a trained model on a recommendation engine. After meeting with the Microsoft, they asked us to implement a recommendation engine based on the customer’s user profile, such as age, gender or occupation. The training data is the preference of different person’s preference of different type of devices, and I am going to use the profile of a new user to give recommendations.

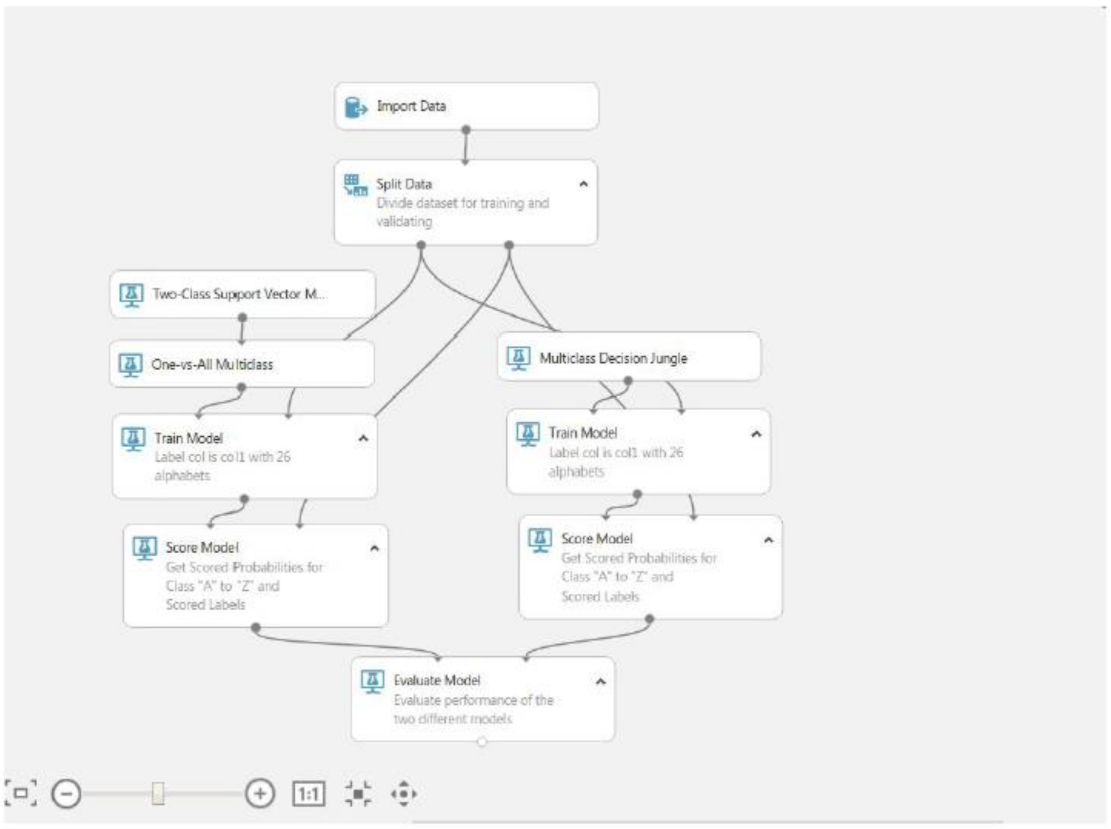

Import Data
Split Data
Divide dataset for training and validating
Two-Class Support Vector M...
One-vs-All Multiclass
Multiclass Decision Jungle
Train Model
Label col is col1 with 26 alphabets
Train Model
Label col is col1 with 26 alphabets
Score Model
Get Scored Probabilities for Class "A" to "Z" and Scored Labels
Score Model
Get Scored Probabilities for Class "A" to "Z" and Scored Labels
Evaluate Model
Evaluate performance of the two different models
1:1

Figure – 7: A sample of model on Machine Learning Studio

Unfortunately, because of the privacy reasons, I cannot get the data during the development phase. I had to make up some fake data to demonstrate the workflow, and it may get connected to the real data set if the business team accepted our bot. As a result, the recommendation of the Machine Learning Model seems very biased. For example, it will recommend a young boy with a gaming device, not matter he really like playing games of not.

Database

The database contains all the device models for recommend by the bot. As mentioned previously, because I am still implementing a prototype, the database is not dynamic.

The catalog and prices will be fixed through all time. However, I try to make the codes to connect with database as an independent module, if the bot is actually connected with a dynamic later, the code logic to be changed will be kept minimal. The database also holds the complaint and feedback from a user. The complaint is used for the customer service people to follow up. The feedback actually means the whether the customer likes the recommendation. It can be used as training data to the machine learning module to improve the accuracy.

3.2 Implementation

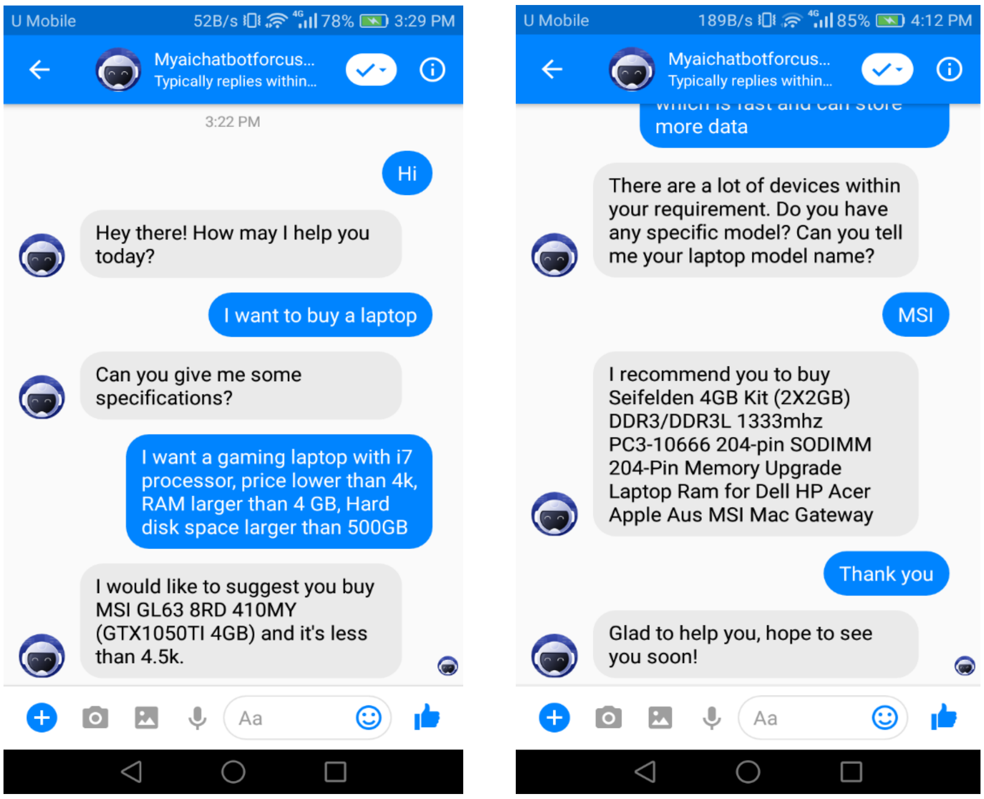
When a user inputs something to the bot through Messenger and Skype, the input will be firstly sent to the knowledge base for its labelling. The knowledge base will classify the input from its database within the input sentence. Then the labelled result will be returned to the bot hosted on Azure in a JSON format.

The JSON string from knowledge base will be received by the bot developed using Bot Framework. The JSON string is parsed by the bot and the questions and answers are extracted for further processing. Corresponding response will be made to user based on the intent. For example, if the question is “greeting”, simply a predefined sentence “Hey there! How may I help you today?” will be returned. If the question is “purchase”, the bot will guide the user through the purchase by giving them recommendations.

Finally, the bot will interact with the database, either query the product catalogue for the products that meets user requirement then return the query result to the user or insert a record for the user to be handled later.

I have implemented a functional bot prototype which is able to understand the intent and entities in a user input by using QnA, ask follow-up questions, search the SQL database based on user demands and finally recommend the product that meets these requirements to the customer. One can try talking to the bot in similar manners illustrated by the following screenshots (figure – 8,9).

The bot handle complaints also. If any user has any issue with their products, they can send their complaints and the bot will store it for the future solutions (figure - 10).



3:22 PM

Hi

Hey there! How may I help you today?

I want to buy a laptop

Can you give me some specifications?

I want a gaming laptop with i7 processor, price lower than 4k, RAM larger than 4 GB, Hard disk space larger than 500GB

I would like to suggest you buy MSI GL63 8RD 410MY (GTX1050TI 4GB) and it's less than 4.5k.

more data

There are a lot of devices within your requirement. Do you have any specific model? Can you tell me your laptop model name?

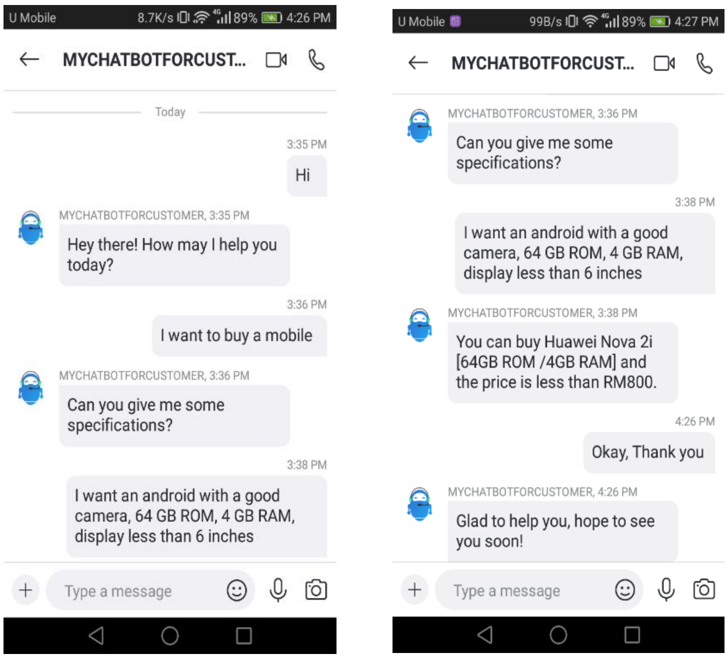
MSI

I recommend you to buy Seifelden 4GB Kit (2X2GB) DDR3/DDR3L 1333mhz PC3-10666 204-pin SODIMM 204-Pin Memory Upgrade Laptop Ram for Dell HP Acer Apple Aus MSI Mac Gateway

Thank you

Glad to help you, hope to see you soon!

Figure – 8: Messenger conversation with Bot


U Mobile 8.7K/s 89% 4:26 PM

MYCHATBOTFORCUST...

Today

3:35 PM

Hi

MYCHATBOTFORCUSTOMER, 3:35 PM

Hey there! How may I help you today?

3:36 PM

I want to buy a mobile

MYCHATBOTFORCUSTOMER, 3:36 PM

Can you give me some specifications?

3:38 PM

I want an android with a good camera, 64 GB ROM, 4 GB RAM, display less than 6 inches

Type a message

U Mobile 99B/s 89% 4:27 PM

MYCHATBOTFORCUST...

MYCHATBOTFORCUSTOMER, 3:36 PM

Can you give me some specifications?

3:38 PM

I want an android with a good camera, 64 GB ROM, 4 GB RAM, display less than 6 inches

MYCHATBOTFORCUSTOMER, 3:38 PM

You can buy Huawei Nova 2i [64GB ROM /4GB RAM] and the price is less than RM800.

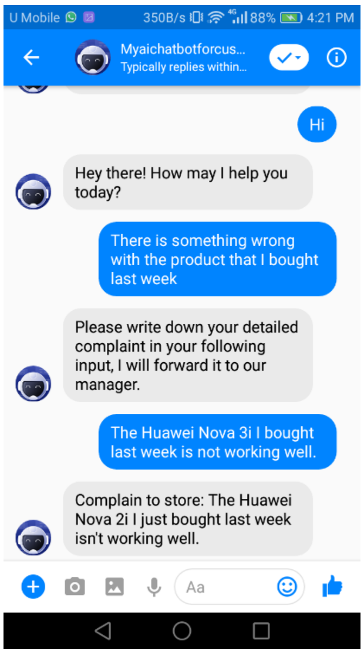
4:26 PM

Okay, Thank you

MYCHATBOTFORCUSTOMER, 4:26 PM

Glad to help you, hope to see you soon!

Type a message


U Mobile 350B/s 88% 4:21 PM

Myaichatbotforcus...
Typically replies within...

Hi

Hey there! How may I help you today?

There is something wrong with the product that I bought last week

Please write down your detailed complaint in your following input, I will forward it to our manager.

The Huawei Nova 3i I bought last week is not working well.

Complain to store: The Huawei Nova 2i I just bought last week isn't working well.

Aa

Figure – 9: Skype Conversation with Bot

Figure – 10: User Complaint

The Machine Learning Model

This section introduced how we found used the Microsoft Azure Machine Learning Studio to create a recommendation engine based on user profile.

First, I generate some fake data for the training. The data is divided into three tables. The first one represents the registered users, with one column representing the userID, and several columns representing profile of the user, such as gender, age, occupation, hobbies. The second table are the five types of devices provided by the Microsoft. This can be replaced by a catalog with all the deviceIDs and their features. However, because lack of data, I only use the five types for demonstration usage.

The third table is the preference of the users to different types. To qualify the preference, I let each user give a rating, from 1 to 10 to each device. The three columns are userID, deviceID, and ratings respectively. The ratings are generally based on our everyday experience, which can be really biased but can show that the module is really working. We only generated the ratings for the first half users as training data and assumes that the other half are the registered user that will talk to our bot and ask for recommendation.

I also monic the process of tuning learning parameters, but because the data is really biased, it can always give good performance. The two figures below (figure – 11) show the workflow of the training phase of the machine learning model. (Sorry but the picture is too long and cannot be put into one page).

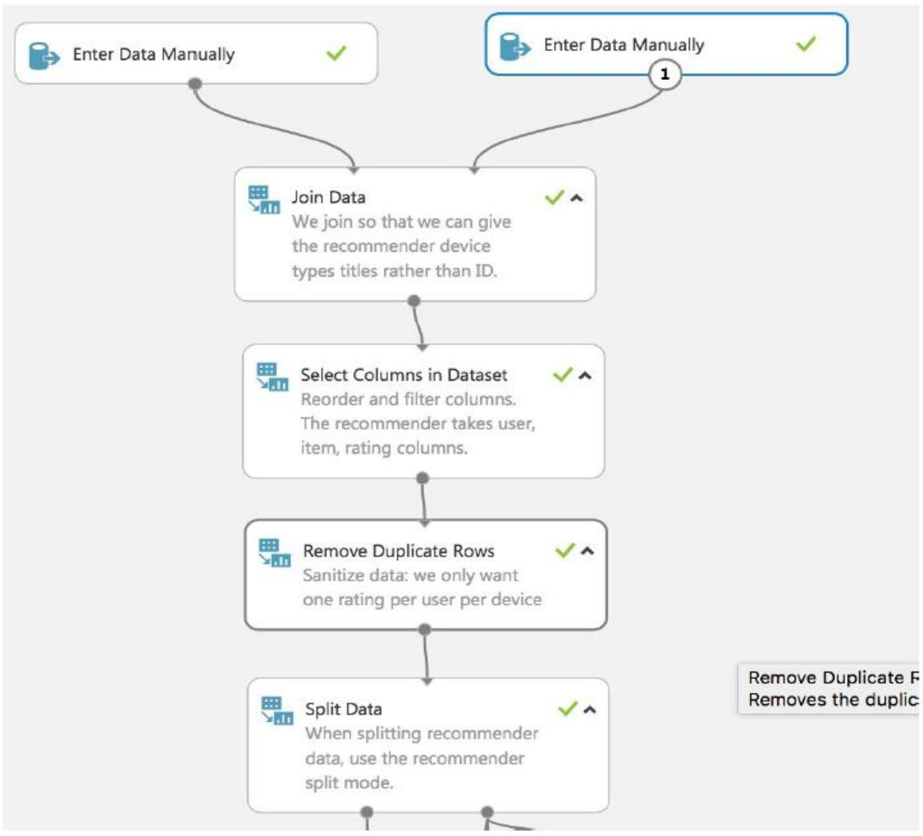

Enter Data Manually
Enter Data Manually
1
Join Data
We join so that we can give the recommender device types titles rather than ID.
Select Columns in Dataset
Reorder and filter columns. The recommender takes user, item, rating columns.
Remove Duplicate Rows
Sanitize data: we only want one rating per user per device
Split Data
When splitting recommender data, use the recommender split mode.
Remove Duplicate F
Removes the duplic

Figure – 11: The first half of the recommender engine.

This figure represents the preprocessing of the training data. First, I join the tables, remove duplicate keys and then split the data into training data and test data, with a rate of 7:3.

The figure below is the logic for training and testing of the recommender. The rating, together with the user's features are fitted in the recommender to do the training. And the three testing methods are used to evaluate the training model.

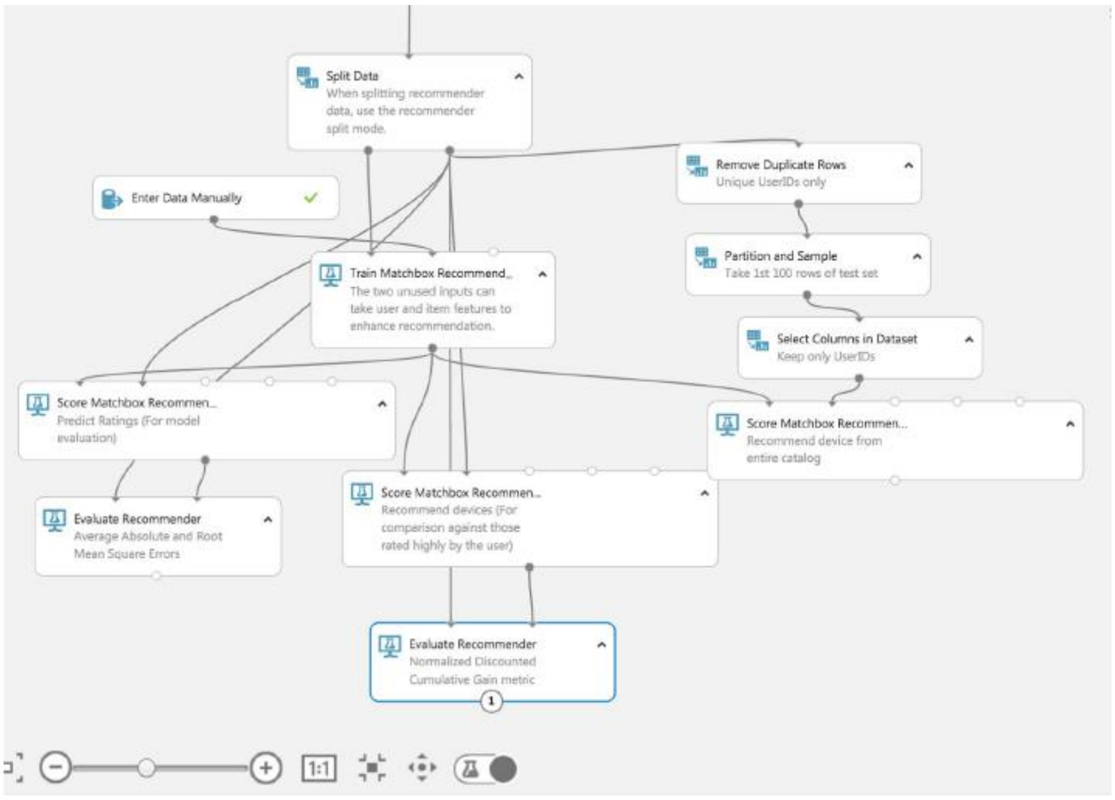

Split Data
When splitting recommender data, use the recommender split mode.
Remove Duplicate Rows
Unique UserIDs only
Enter Data Manually
Partition and Sample
Take 1st 100 rows of test set
Train Matchbox Recommend...
The two unused inputs can take user and item features to enhance recommendation.
Select Columns in Dataset
Keep only UserIDs
Score Matchbox Recommen...
Predict Ratings (For model evaluation)
Score Matchbox Recommen...
Recommend device from entire catalog
Evaluate Recommender
Average Absolute and Root Mean Square Errors
Score Matchbox Recommen...
Recommend devices (For comparison against those rated highly by the user)
Evaluate Recommender
Normalized Discounted Cumulative Gain metric
1:1

Figure – 12: The second half of the recommender engine.

After training the model, I deployed the machine learning model as a web service and define the input and outputs. The trained is as an input to the recommender in the left bottom corner. I also include the training data, which means if a user has already rated the devices, I do not need to do predictions but follow her preference.

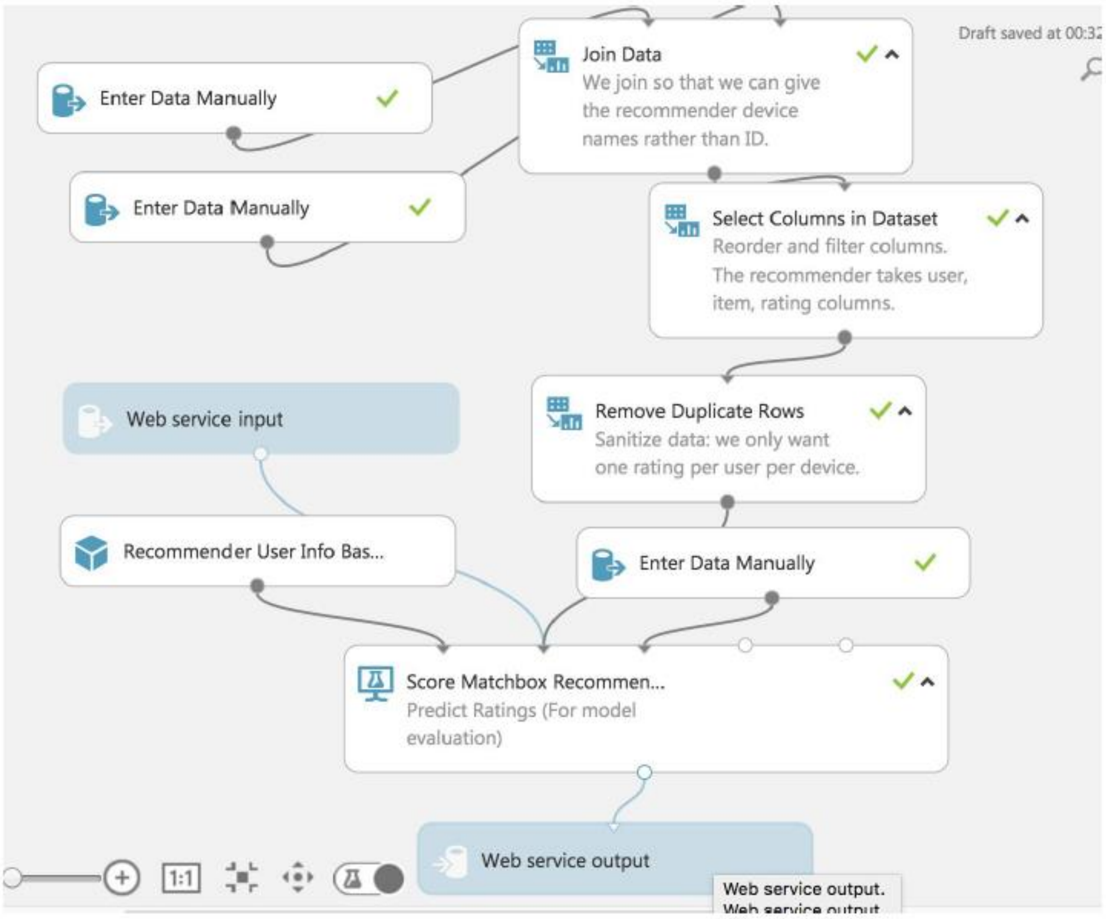

Join Data
We join so that we can give the recommender device names rather than ID.
Draft saved at 00:32
Enter Data Manually
Enter Data Manually
Select Columns in Dataset
Reorder and filter columns. The recommender takes user, item, rating columns.
Web service input
Remove Duplicate Rows
Sanitize data: we only want one rating per user per device.
Recommender User Info Bas...
Enter Data Manually
Score Matchbox Recommen...
Predict Ratings (For model evaluation)
1:1
Web service output
Web service output.
Web service output

Figure – 13: The deployed recommender engine.

3.3 Testing/ Evaluation

This section shows the evaluation of our project, mainly focus on two aspects. The performance of the QnA Maker Knowledge Base, the performance of the profile-based recommendation.

The performance of the QnA Maker Knowledge Base

The QnA Maker Knowledge Base is not giving 100% correct output because of the limitation of dataset and the bot is still learning. When testing the bot with Facebook Messenger it takes 3 - 5 minutes but for Skype it takes only 2 - 3 minutes maximum for replying. The Knowledge Base has only limited data so if users ask any irrelevant questions to the bot it will not give the right answer that user wants. It will take time to understand every single question and give the best answer.

Machine Learning Studio Model Evaluation

Because the data are made up by my selves, with limited quantity and quality, I have already foreseen a bad performance on the model. However, the result matched our assumption while generating the data. For example, if I give a young boy, it will recommend gaming devices. I also assumed that programmers want a high-performance desktop, the recommender also gives the right recommendation. Because I have no real data, I cannot test whether the model can be generalized to unseen data.

The table below shows the result of the machine learning model, it can be seen that the model fit the training data well, no matter on predicting the ratings given by a user to an item, or on predicting the rank of preference of a user. That is because the data is biased. Although we tried to add some disturb in the data, we were still following some rules while generating them. That means actually the model must be overfitted and can hardly extend to real world. However, this can be solved if the bot is maintained by Microsoft and they can use the real data to train the model.

Mean Absolute Error (test size = 60)	1.83
Root Mean Square Error (test size = 60)	2.32
Normalized Discounted Cumulative Gain (test size = 15)	

Table 1: The Performance of Machine Learning Model

CHAPTER FOUR

ANALYSIS OF RESULT AND OUTCOME

User Study

I found 20 students from HKU as users of our bot, and let them give rates to the bot, on the conversation logic, recommendation satisfactory rate, and overall rating, from 1 to 9. I also showed them the five categories of devices and let them determine whether they thought they have been recommended with the right types, as an evaluation of the performance of Bot. The table below shows the average of the ratings.

Rating on interaction logic	8.75
Rating on recommendation result	5.45
Overall Rating	6.6
Accuracy of category	0.6

Table 2: The User Study Result

From the result, the interaction logic is well accepted by the users because of the concision of the words. However, the accuracy is not satisfactory.

This can be largely improved if more data can collect in real world as training data. The current training dataset provided by the Microsoft is too small for a machine learning program. Moreover, the data are actually created by the business team, rather than from real world, which makes the data are also biased as that for machine learning module. Furthermore, the lacking in data size makes it hard to find relation between each sample. The Bot can work perfectly if the input is similar to a previous training data but will be panic if the data is not similar to anyone and give some random answer.

CHAPTER FIVE

CONCLUTION AND FUTURE WORK

Machine Learning and AI bots are the most heated topic in the current Computer Science world. However, in order to have good performance, other than a good algorithm, the training data is also critical. In this project, I built a Machine Learning based bot prototype as a standalone web service to give recommendation of electronic device for customers. If the bot is continually developed and maintained by the Microsoft, I want to give the following suggestions which I believe can make the bot much better. I hope this bot will be continually developed by Microsoft and be online someday.

REFERENCES

[1] Quora [Internet]. USA: Why do PC laptop manufacturers have such a confusion of differently specified models? [updated 2012 Mar 17, cited 2016 Oct 22]. Available from:
https://www.quora.com/Why-do-PC-laptop-manufacturers-have-such-a-confusion-of-differently-specified-models .

[2] Kissmetrics [Internet]. USA: Are you Losing Sales by Giving Customer Too Many Choices? [updated 2012, cited 2016 Oct 22]. Available from:
https://blog.kissmetrics.com/too-many-choices .

[3] Lexalytics [Internet]. USA: NLP in 5 Minutes. [updated 2016 April 27, cited 2016 Oct 21].
Available from: https://www.lexalytics.com/lexablog/2016/nlp .

[4] Apple [Internet]. USA: Siri. [cited 2016 Oct 21]. Available from:
http://www.apple.com/ios/siri .

[5] Microsoft [Internet]. USA: What is Cortana. [updated 2016 Sep 10, cited 2016 Oct 21].
Available from: https://support.microsoft.com/en-hk/help/17214/windows-10-what-is .

[6] Microsoft [Internet]. USA: Language Understanding Intelligent Service. [cited 2016 Oct20]. Available from:
https://www.microsoft.com/cognitive-services/en-us/language-understanding-intelligent

[7] Simon P. Too Big to Ignore: The business Case for Big Data. USA: Wiley; 2013 Mar 18.

[8] Gitbooks [Internet]. Artificial Intelligence: Machine Learning. [cited 2016 Oct 20].Available from:
https://leonardoaraujosantos.gitbooks.io/artificialinteligence/content/

[9] Fisher RA. The use of multiple measurements in taxonomic problems. Annals of Eugenics.1936; 7(2): 179-188.
[10] Microsoft [Internet]. USA: Microsoft Azure. [cited 2016 Oct 20]. Available from:
https://www.microsoft.com/cognitive-services/en-us/language-understanding-intelligent

[11] Microsoft [Internet]. USA: Microsoft Bot Framework. [cited 2016 Oct 20]. Available from:
https://dev.botframework.com/ .

Appendix A

Gant Chart

Dates	Descriptions	Status
Feb 15 – Feb 22	- Got the idea of the project - Confirm the scope of the project - Find the title of the project	Completed
Feb 23 - March 23	- Collected information about the project and do some research on few papers.	Completed
March 24 – March 30	- Came up with a report for the project	Completed
March 31 – April 6	- Ready the report based on previous work - Tried to find out the limitation of the past bots - Made a demo poster for the project	Completed
April 7 – May 14	- Made the final report and the final poster and finished FYP showcase	Completed
Sep 15 – Nov 15	- Design the bot and train the language model with data - Develop and connect the bot with database	Completed
Oct 10 – Nov 25	- Hosted the bot on Facebook - Develop the Machine Learning Studio Model - Develop the demo Bot	Completed