THE UNIVERSITY OF HONG KONG

DEPARTMENT OF COMPUTER SCIENCE

Final Report

[Industry-based Project]
AI Bot to Make the Best Decision for the Customer

Name: Huang Kai

UID: 3035086340

Course: COMP4801 Final Year Project

Supervisor: Dr. Kenneth Wong

Date of Submission: April 16th, 2017

Client: Microsoft Hong Kong Limited

Abstract

Microsoft is a worldwide company which produces a large number of competitive IT products catering to various needs of its customers. However, the fact is that nowadays many devices share a lot of similarities and common features, which makes customers quite confusing to select the most suitable model for daily use.

Microsoft Hong Kong Limited initiated this project in order to solve the aforesaid problem and better serve its customers. An AI Chat Bot has been successfully developed which can smoothly communicate with customers online in natural languages and recommend them with the products that meet their particular needs. Beyond that, the Bot also helps handle complaints from customers. Customers can easily get connected to the Bot through multiple channels.

The other two members of this project team are Chen Xusheng (Michael) and Tan Zhanwen (Francis). The supervisor of this project is Dr. Kenneth Wong in the Department of Computer Science, the University of Hong Kong.

Acknowledgements

Here I would like to extend my sincere gratitude to the following people for their kind and timely help throughout the whole project. It is their efforts that results in the successful completion of this project.

Dr. Kenneth Wong, my supervisor. Thank you for your time and guidance in this academic year. Your suggestions on design and implementation are really helpful to me.

Dr. Anthony Tam, second examiner of this project. Thank you for getting our team in touch with Microsoft and giving valuable feedbacks during the preliminary presentation.

Mr. Samson Lee, contact person from Microsoft. Thank you for providing useful documentations to start the project, keeping us updated with the requirements from the business team and helping me tackle many technical difficulties.

Michael and Francis, my teammates of this project. Thank you for your joint efforts and encouragement. It is my pleasure to cooperate with you.

Participants of the experiments. Thank you for taking time to try the Bot, providing real-life user input and giving practical comments to our team.

Table of Contents

1 Introduction

This section gives a general introduction and background of the whole project.

In the late twentieth century, when the development of Information and Technology just started, there were only limited choices of digital products. Nevertheless, things are totally different now. We are in the world filled up with choices. Needless to say, people tend to get lost when facing various options. For a lot of big companies such as Microsoft, they produce a large number of digital products that resemble in many aspects. On Microsoft Online Store, even though all the products have already been classified into different types, within each particular category, customers may still find it an unpleasant experience to distinguish the minor disparity among those models differing in CPU, RAM or Hard Disk etc. At the retail place, generally it is not possible for sales assistants to take care of every single customer on his or her special need and make recommendations correspondingly.

Thus, we have to admit that customers may be at the risk of failing to buy the right product they want and unwisely spending their money, especially those with little prior knowledge of the technical specifications of devices. Furthermore, it turns out that people have no idea of the exact performance they expect from a machine until they have actually used it in real life for a certain period of time.

A solution to the aforementioned problem is to develop an AI Chat Bot to make the best decision for customers. The Bot is supposed behave and react like a human being and is 24-hour online to entertain customers at any time of a day by immediate responses. The transformation from on-site shopping to online shopping makes the Bot competitive since it can be easily reached by customers and it can directly support their shopping online. It was really challenging to build a Bot several decades ago. But fortunately, machine learning technologies have seen rapid enhancement during the past few years and at the

same time many useful services and APIs emerge which we can make use of. Microsoft already provides the Bot Framework [1] as a starting point to build the Bot.

The remaining of this report proceeds as follows. I will first give a review of previous work related to this project in §2 followed by the project objectives in §3 and scope in §4. Afterwards, §5 provides some details of our communication with our client Microsoft. Most importantly, §6 elaborates the methodologies and technical details of this project. In §7, I present my own contributions to this project. §8 discusses the several obstacles and restrictions encountered throughout this project. §9 shows the deliverables of this project and illustrates a demo of the Bot. §10 is about the experiments we have conducted to evaluate the performance of the Bot and the subsequent results. I end this report by suggesting some possible future improvements and making a final conclusion.

2 Previous Work

This section discusses some previous work related to AI Bots and online customer service.

2.1 Existing AI Bots

Some existing AI Chat Bots could be found and used online.

Figure 1 shows a Chat Bot used by a retail store called '*1·800 flowers*' [2] to serve customers with the purpose of selling flowers. The Bot is hosted on Facebook Messenger, waiting for customers to order flowers. The Bot only provides predefined buttons for customers to click and select. Lack of artificial intelligence makes it less attractive and user friendly.

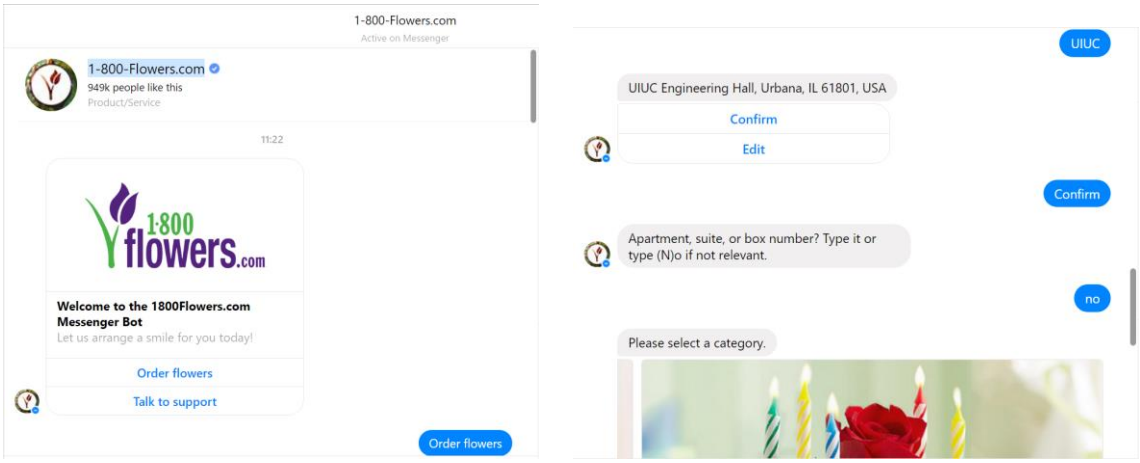


Figure 1: Bot for Ordering Flowers

Figure 2 below shows another AI Bot called '*cleverbot*' [3]. This Bot can understand human languages to some extent and ask user some questions. However, most questions asked by this Bot are about personal information (such as 'What's your name?', 'Where

do you come from?', etc.). English is the only language supported for this Bot. This Bot is not for the purpose of serving customers.

Figure 2: Clever Bot

2.2 Online Customer Service

Many companies do provide online support to give their customers more efficient and pertinent service. Figure 3 below shows the chat interface of Apple [4] and Figure 4 shows the online support of Microsoft [5].

Figure 3: Apple Online Support

Figure 4: Microsoft Online Support (1)

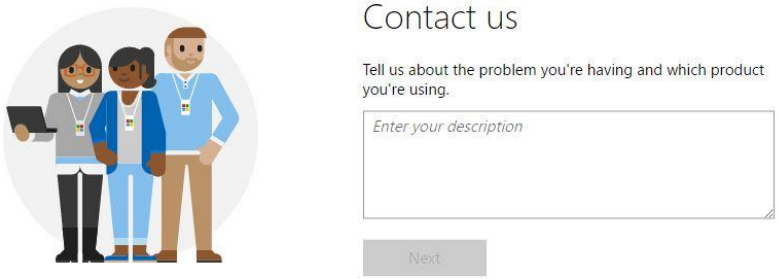


Figure 5: Microsoft Online Support (2)

It is worth mentioning that these online customer supports are operated manually. In other words, customers are actually talking to real persons behind and it is evitable that manual service will cause some delay in delivering responses. In the meanwhile, the service time of Apple and Microsoft are from 9:00am to 5:00pm and from 9:00am to 9:00pm respectively and thus customers can only be served during these time periods. It is true that customers can alternatively leave a message as shown in Figure 5 above, but not all the messages from customers are actually relevant. It may waste staff lots of time looking at those irrelevant messages.

From the discussion above, it will be good if we can combine the advantages of existing Bots and online manual service to make an AI Bot that understands what people say and acts like a person. The creation of such a Bot can definitely relieve the workload of service staff to help customers make decisions on product purchase. In addition, the Bot stores a message into the database only if it confirms that the message is actually a complaint, nothing anything irrelevant.

3 Objectives

This section discusses the main objectives of this project.

The goal of this project is to produce an Artificial Intelligent Chat Bot to be used by Microsoft Hong Kong Limited to serve its potential customers. The Bot shall be able to understand sentences typed in natural languages and have conversations with customers in natural languages as well. The ultimate goal of this Bot is to recommend Microsoft customers with the certain kind of products meeting their demands best and at the same time enhance Microsoft's efficiency in handling complaints. The Bot ought to be deployed on several platforms that are popular in Hong Kong.

Our Bot will stand out for its distinct features. In the first place, besides leading customers to select from pre-defined choices all the time, as quite a lot of Bots in the market behave, our Bot allows customers to initiate the conversation and chat in natural languages. Customers will find the Bot more user friendly in this way. Secondly, customers can access the Bot from different channels depending on their own preferences. Thirdly, we fully take advantage of advanced machine learning techniques to make the Bot more intelligent. Last but not least, the inputs from all customers will be used as the

training data for the Bot in the future. That means even after the Bot has been put into use, its performance continues to be improved greatly.

4 Scope

This section discusses the project scope, which is defined in the following five aspects.

4.1 Interaction with the Bot

The Bot should be able to communicate smoothly with users online in natural languages and give immediate response to the user whenever a user input is received. The Bot ought to handle ‘Purchase’ and ‘Complaint’ queries from the customers and give replies according to the query type.

4.2 Pool of Recommendation Models

Candidate products that are to be recommended to the customers by the Bot come from the device catalogue provided by Microsoft. Not only products produced by Microsoft (such as Surface Pro, Surface Book, etc.) but also products from Microsoft’s business partners (such as ASUS, Samsung, HP, etc.) are in the pool of recommendation models.

4.3 Complaints Handling

The Bot should be able to handle complaints raised from customers. Complaints can be specific to a certain product or a certain service. Any other complaint that is neither about products nor about service will be regarded as a general complaint.

4.4 Languages Supported

Cantonese is chosen as the language supported by the Bot since it is most commonly used by the local people in Hong Kong. The responses of the Bot should be in Traditional Chinese as well.

4.5 Host Platforms

The Bot should be hosted on Skype and Facebook Messenger for the reason that these two are the most popular platforms in Hong Kong. The behaviors and performance of the Bot on both platforms should be more or less the same. Other platforms such as Slack are not required by Microsoft and thus not in the scope.

5 Communication with client Microsoft

This sections records our communication details with Microsoft.

5.1 Meetings in Person

We had three meetings in person with representatives from Microsoft. Details of these meetings are summarized as follows.

(1)
Date: Sept 8th, 2016
Venue: CYC Building, the University of Hong Kong
Attendees: Mr. Samson Lee (from Microsoft), Dr. Kenneth Wong, Dr. Anthony Tam, Chen Xusheng, Huang Kai and Tan Zhanwen.
Content:
Samson gave us a brief description of this project and the expectations of Microsoft on the project. Samson also introduced to us the main technologies we were going to use. We came out with a proposed schedule for the academic year.

(2)
Date: Nov 24th, 2016
Venue: Microsoft Office, Cyberport
Attendees: Mr. Samson Lee (from Microsoft), Chen Xusheng, Huang Kai and Tan Zhanwen.
Content:
We faced some difficulties in designing language models and Samson gave us some useful solutions. He advised to use one single language model so as not to increase the complexity of this project. He also suggested us to add some pop-up buttons to the Bot so that it can better guide the customers. In addition, we set the goals to be achieved in the next stage.

(3)
Date: Jan 17th, 2017
Venue: Microsoft Office, Cyberport
Attendees: Ms. Joelle Woo (from Microsoft), Ms. Anna Chow (from Microsoft), Huang Kai and Tan Zhanwen.
Content:
We gave a live demonstration of our Bot prototype to the technical representative and business representative from Microsoft. We got some comments from Anna on the user interface and flow of the Bot. Her comments include the following:

- Cantonese is of the highest priority in terms of languages supported by the Bot.
- Facebook Messenger is regarded as the most important platform for the Bot.
- Add some attachments such as images to make the conversation more vivid.
- Focus more on the usage of products rather than the technical specifications.

We further refined some of our scope according to new business requirements. Joelle and Anna said that they would help us contact the promoter to obtain some sample questions that customers would most likely ask.

Remark: We will have another meeting with Microsoft representatives to hand over the whole project in late April or early May, 2017.

5.2 Conversations via Email

After the first meeting, Samson provided us with many helpful documentations to start the project by email. Throughout the two semesters, we regularly email Samson to report our progress and ask him for feedbacks. We also sought him for help whenever we face technical difficulties. We are really thankful to Samson's quick response and timely help.

6 Methodologies and Implementation Details

This section elaborates in detail how we implemented this project, including the development tools we use, the application modules used by the Bot and the workflow of the Bot.

6.1 Bot Framework

Bot Framework is primarily employed as the fundamental framework of this project. Bot Framework generally facilitates to build high quality Bots with great user conversation experiences. The framework provides tools to easily build and connect Intelligent Bots including basic I/O, language and dialogue skills as well as user connection interfaces with promising, responsive and scalable features. Figure 6 below gives an overview of Bot Framework.

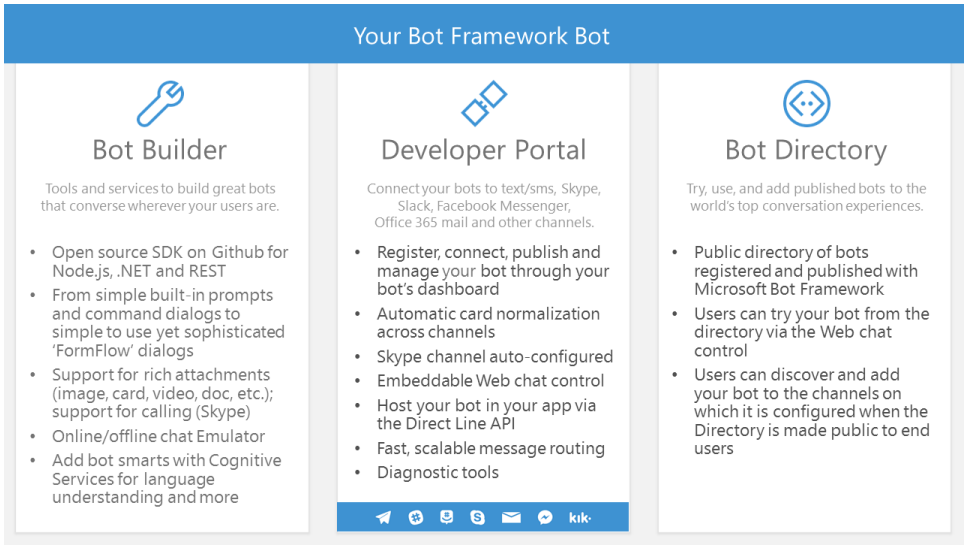


Figure 6: Bot Framework Overview

Bot Framework is composed of three components: Bot Builder, Developer Portal and Bot Directory.

6.1.1 Bot Builder

Bot Builder Software Development Kit (SDK) is an open source SDK hosted on GitHub which has a powerful dialogue system to handle interactions. The SDK provides libraries, samples and tools such as prompts or command dialogues for developers to set up the Bot and make it running. Attachments such as images, cards are supported as well. Bot Builders for Node.js, .NET or REST API are available to write Bots. There are already some sample Bot Builders with C# that aim at achieving different functionalities on GitHub [6].

6.1.2 Developer Portal

Bot Framework Developer Portal enables seamless connection of the Bot with popular service platforms such as Skype, Facebook Messenger, Office 365 mail and so forth. Developers need to first register the Bot on Bot Framework Site with the requisite details of the Bot. Then one can choose a desired channel and configure it with developer credentials. Afterwards, developers can publish the Bot and test its connection. Registered Bots can be managed via the Bot's dashboard in Developer Portal.

6.1.3 Bot Directory

Bot Directory is a public directory where developers can review and experience all the Bots registered through the Developer Portal. For each registered Bot, there's detailed information of the Bot and its publisher, along with the available channels. Users can try and discover these Bots using either the web chat control or any configured channel.

6.2 Development Tools

We choose Visual Studio [7] as the IDE and C# [8] as the programming language of this project. We use Bitbucket [9] for team collaboration and version control.

6.2.1 Visual Studio

Visual Studio is an IDE from Microsoft. Its highlighted features include great IntelliSense, easy code navigation, fast builds, and quick deployment. Visual Studio also has advantage of increasing productivity and making it easy to do work alone or as part

of a team. The template for Bot Application can be directly imported into Visual Studio for development. We can test the Bot locally on Visual Studio by using Bot Framework Emulator [10] and we can publish the Bot directly on Visual Studio.

6.2.2 C#

C# is a general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative. Bot Builder for .NET leverages C# to provide a natural way to write Bots, which is comfortable and convenient to use.

We choose Bot Builder for .NET using C# as our programming language instead of Node.js-based or REST API-based Bots due to two main reasons. First of all, C# is a more mature language with a well-developed community. The second reason is that the documentations for writing Bots using C# provided by Microsoft are much more detailed and the APIs are well tested. We need to highly rely on the APIs to control the Bot's conversational logic.

6.2.3 Bitbucket

Bitbucket is a web-based hosting service that uses Git revision control systems. Visual Studio supports Bitbucket for teamwork and the source code hosted on Bitbucket can be set in private domain.

6.3 Microsoft Azure

Microsoft Azure [11] is Microsoft's cloud computing platform for building, deploying and managing applications and services through Microsoft-managed data centers. Different subscription plans are available to fit individual needs. Most of the application modules of this project are hosted on Azure.

6.4 Language Understanding Intelligent Service (LUIS)

Microsoft's Machine Learning API – Language Understanding Intelligent Service (LUIS) [12] is applied to understand languages contextually. LUIS enables the Bot to interpret the meanings of human languages and accordingly react to user requests. LUIS models are created as domain-specific applications. LUIS supports multiple languages including English, Chinese, etc. Note that each LUIS model only supports one language.

Two main components of a LUIS model are *intent* and *entity*. Developers can design intents and entities to achieve the language understanding purposes for each model. Intents are the intentions or desired actions of a user that are conveyed through utterances (sentences). Entities are the key data that ought to be retrieved from utterances, describing information closely related to the intent.

For example, as to the given utterance 'I want to book a flight from Hong Kong to Shanghai', we can define intent as 'Flight Booking' and define entities 'Departure' ('Hong Kong') as well as 'Destination' ('Shanghai'). After defining and labelling the intent and entities, all the necessary information in this sentence is captured.

Each LUIS application has a predefined intent 'None', which simply means that the model fails to get any real meaning from the given utterance. LUIS also has some prebuilt entities such as 'datatime', etc. One can also define hierarchical entities and composite entities if necessary.

In our project, we have defined 9 intents: '*Notebook*', '*Desktop*', '*Tablet*', '*2in1*', '*Gaming Device*', '*Greeting*', '*Satisfied*', '*Unsatisfied*' and '*Complaint*'.

The first five intents are the five categories of products. LUIS identifies these intents by the specified usage of a product in the utterance. 'Greeting' is the intent that mostly happens when a user first initiates a conversation with the Bot. 'Satisfied' and 'Unsatisfied' are used to detect a customer's state of mind on the recommended products. 'Complaint' is for detecting a customer's intention to complaint about something. We further define two entities in order to classify the complaints: 'Product' and 'Service'.

The following table on the next page gives some sample utterances provided by Microsoft and the corresponding intents, entities:

Utterance	*Intent*	*Entity*
你好，有無人應答？	Greeting	
我通常只會係屋企用電腦	Desktop	
我通常只會用電腦瀏覽網站同睇電郵	Tablet	
我想搵部電腦打機	Gaming Device	
我想搵部電腦做文書工作	2in1	
我會上網，同埋用 Photoshop, Illustrator	Notebook	
多謝你的建議	Satisfied	
唔好意思你有無其他建議？	Unsatisfied	
點解宜家微軟服務電話無台灣？	Complaint	Service
Surface Pro3 電池異常	Complaint	Product

Table 1: Sample Training Data for LUIS

We have tried to rephrase and add variations to each sample sentence in order to provide more training data for the LUIS model. LUIS provides means to train and test the model easily after it is fed with data.

After the completion of the LUIS model, it can be published as an application and the Bot can get connected to it using a subscription key. Bot can send requests to the LUIS model and receive responses through an HTTP endpoint.

6.5 Recommendation Engine: Machine Learning Studio

Microsoft Azure Machine Learning Studio [13] is an interactive and visual workspace to easily build and test predictive analysis models. One can drag and drop datasets, preprocessing and training modules onto an interactive canvas, then connect them together to form an experiment. The trained model of a Machine Learning Studio experiment serves as the recommendation engine of our Bot.

Figure 7 below shows the interactive canvas of our experiment. The training data we use include the user profile (containing age, gender and occupation) and their ratings on the five categories of products (namely Notebook, Desktop, 2in1, Tablet and Gaming Devices). The idea of the model is that people with similar age, same sex or similar occupations tend to favour the same kind of digital products.

Data tables are first joined together and duplicated rows are removed before running the machine learning algorithm. We choose the Naïve Bayes algorithm to train the model under the assumption that all customers submit their ratings independently. After the training process, if a profile of a new customer is given, the model will return the product category with the highest possible rating according to the ratings of previous customers possessing similar personal characteristics.

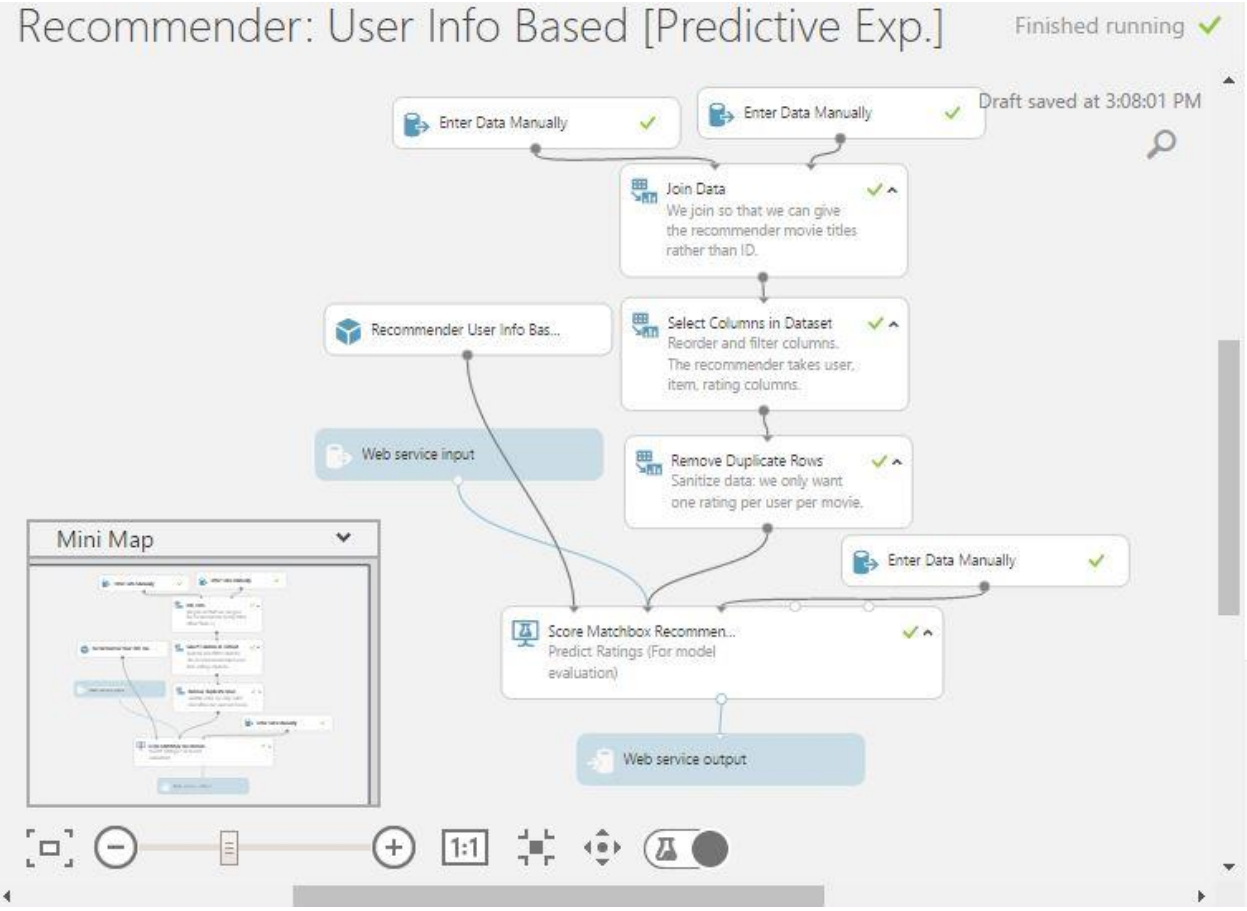


Figure 7: Machine Learning Studio Experiment

After being published as a web service, the model can be accessed by the Bot using an API key and request URI in the form of requests and responses. The Bot encapsulates the request in JSON format and send the request to the web service. The web services also gives back the response in JSON format and the Bot can parse the JSON string to get the recommendation result.

6.6 SQL Database

SQL Database [14] is created on Azure and used for the Bot to store and access data. The Bot can be configured to get connected to the SQL Database via a connection string at the time when it is published to Azure. Figure 8 below shows the connection string of the SQL Database I created on Azure with Server name 'fyp16004.database.windows.net' and Database name 'CustomerBotDB'.

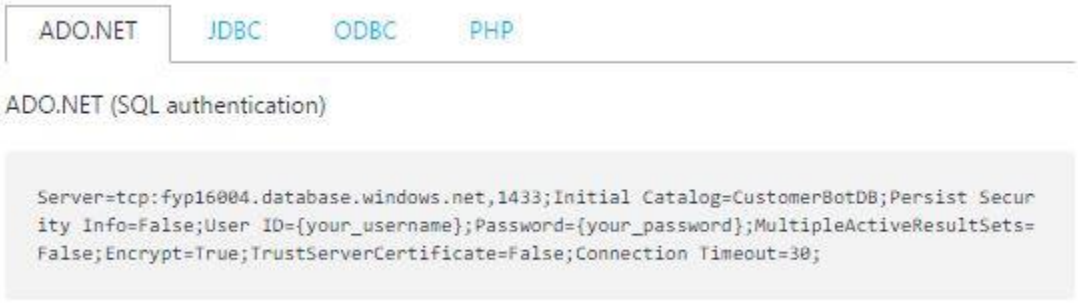


Figure 8: SQL Database Connection String

The SQL Database can be accessed and operated by queries either using Microsoft SQL Server Management Studio (SSMS) [15] or using SQL Server Object Explorer in Visual Studio. Figure 9 below gives a view of SSMS.

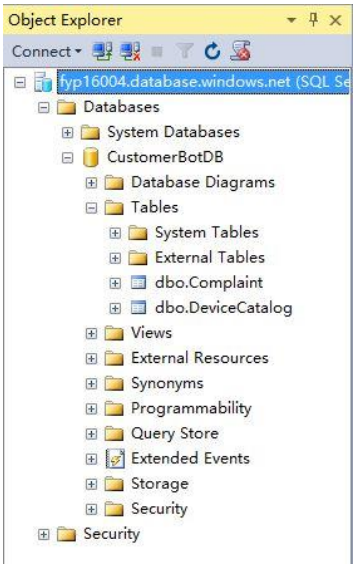
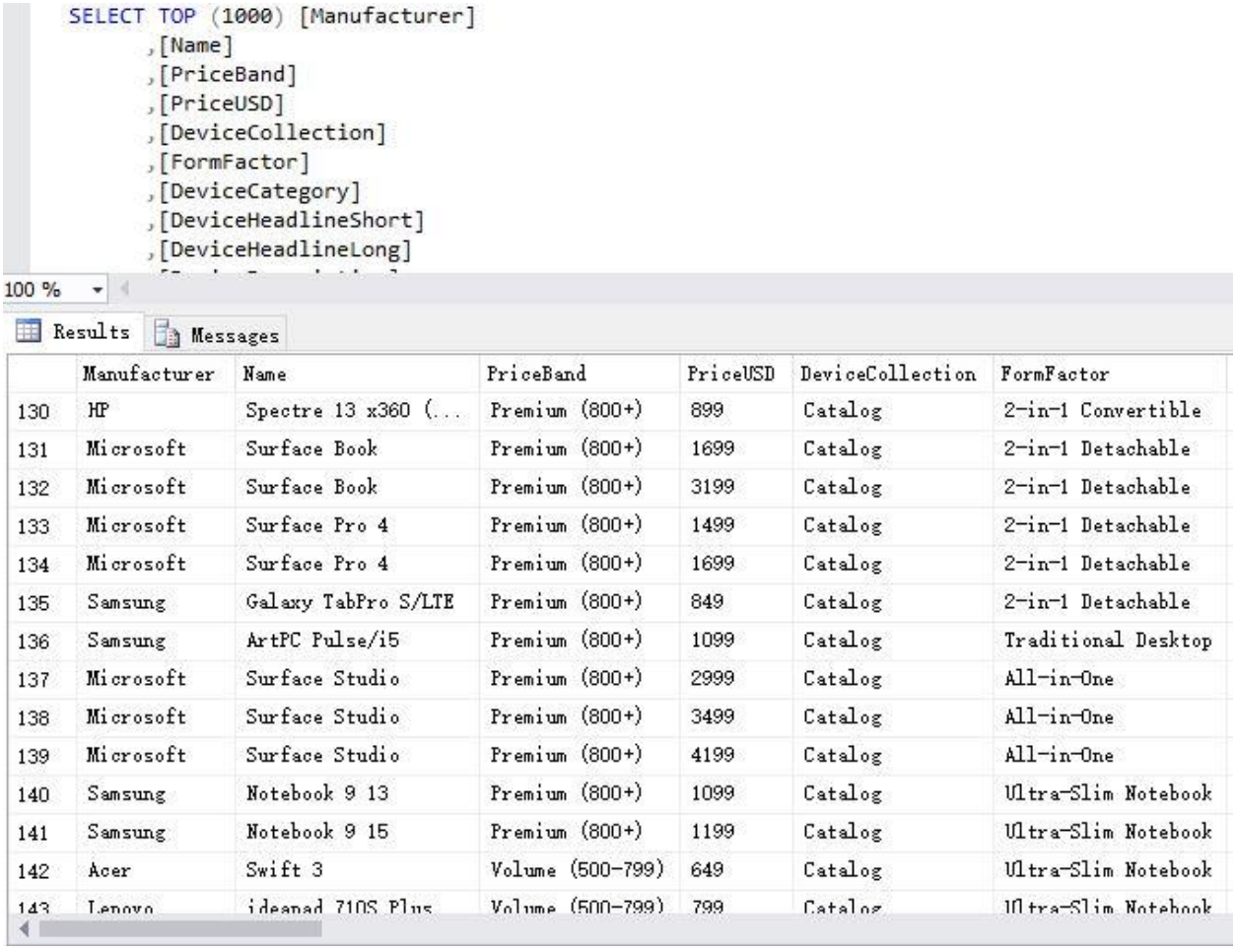


Figure 9: Microsoft SQL Server Management Studio

There are two tables in the database. ‘DeviceCatalog’ table contains the data of all the products provided by Microsoft. Its fields contain ‘Name’, ‘Manufacturer’, ‘FormFactor’, ‘Price’, etc. Sample products can be viewed in Figure 10 below. Another table ‘Complaint’ stores all the complaints made by customers. Its fields include ‘UserID’, ‘UserName’, ‘Channel’, ‘Time’, ‘Type’ (of the complaint) and ‘Content’.



```
SELECT TOP (1000) [Manufacturer]
      ,[Name]
      ,[PriceBand]
      ,[PriceUSD]
      ,[DeviceCollection]
      ,[FormFactor]
      ,[DeviceCategory]
      ,[DeviceHeadlineShort]
      ,[DeviceHeadlineLong]
```

100 %

Results Messages

	Manufacturer	Name	PriceBand	PriceUSD	DeviceCollection	FormFactor
130	HP	Spectre 13 x360 (...	Premium (800+)	899	Catalog	2-in-1 Convertible
131	Microsoft	Surface Book	Premium (800+)	1699	Catalog	2-in-1 Detachable
132	Microsoft	Surface Book	Premium (800+)	3199	Catalog	2-in-1 Detachable
133	Microsoft	Surface Pro 4	Premium (800+)	1499	Catalog	2-in-1 Detachable
134	Microsoft	Surface Pro 4	Premium (800+)	1699	Catalog	2-in-1 Detachable
135	Samsung	Galaxy TabPro S/LTE	Premium (800+)	849	Catalog	2-in-1 Detachable
136	Samsung	ArtPC Pulse/i5	Premium (800+)	1099	Catalog	Traditional Desktop
137	Microsoft	Surface Studio	Premium (800+)	2999	Catalog	All-in-One
138	Microsoft	Surface Studio	Premium (800+)	3499	Catalog	All-in-One
139	Microsoft	Surface Studio	Premium (800+)	4199	Catalog	All-in-One
140	Samsung	Notebook 9 13	Premium (800+)	1099	Catalog	Ultra-Slim Notebook
141	Samsung	Notebook 9 15	Premium (800+)	1199	Catalog	Ultra-Slim Notebook
142	Acer	Swift 3	Volume (500-799)	649	Catalog	Ultra-Slim Notebook
143	Lenovo	ideapad 710S Plus	Volume (500-799)	799	Catalog	Ultra-Slim Notebook

Figure 10: Device Catalogue Table in the Database

Each table is converted to an ADO.NET Entity Data Model [16], which is essentially a class object in C#, to allow programmatic access to the database. Visual Studio can help generate the data models automatically.

Language-Integrated Query (LINQ) [17] is used to query the database. The syntax of LINQ is quite similar to SQL. ‘Where’ clause is supported so that the Bot can filter the products by given requirements. Some sample queries can be found in Figure 11 below.

```
using (ServiceContext svcContext = new ServiceContext(_serviceProxy))
{
 var query_where1 = from a in svcContext.AccountSet
                    where a.Name.Contains("Contoso")
                    select a;
 foreach (var a in query_where1)
 {
  System.Console.WriteLine(a.Name + " " + a.Address1_City);
 }
}
```

Figure 11: LINQ Query Examples

6.7 Text Translator

Text Translator is part of Microsoft's Cognitive Services [18], which provides abundant APIs for vision, speech, language, etc. We use Text Translator API to translate Cantonese into Mandarin. The reason why we do this is that LUIS doesn't support Traditional Chinese but supports Simplified Chinese. If we intentionally use Traditional Chinese to train the model, the accuracy turn out to be extremely low. So we first use Text Translator to translate Cantonese into Mandarin and then pass the translated text to the LUIS to detect intent.

6.8 Workflow

The workflow of the Bot can be summarized by the flow chart on the next page (Figure 12).

Whenever a customer inputs a sentence in Cantonese, it is first translated into Simplified Chinese and then forwarded to the LUIS model for intent detection. If the intent is 'Greeting', the Bot will greet the customer and asks what kind of product he or she wants to buy. If the intent is a product category (namely 'Notebook', 'Desktop', '2in1', 'Tablet', 'Gaming Device'), the Bot will query the database and recommend some products within the specified category. If the user is 'Satisfied', then the conversation comes to an end. If the user is 'Unsatisfied' with the recommendation, the Bot will let the customer select some features to filter the products and query the product catalogue again. If the intent is 'Complaint', the Bot will store the complaint into the database for future handling.

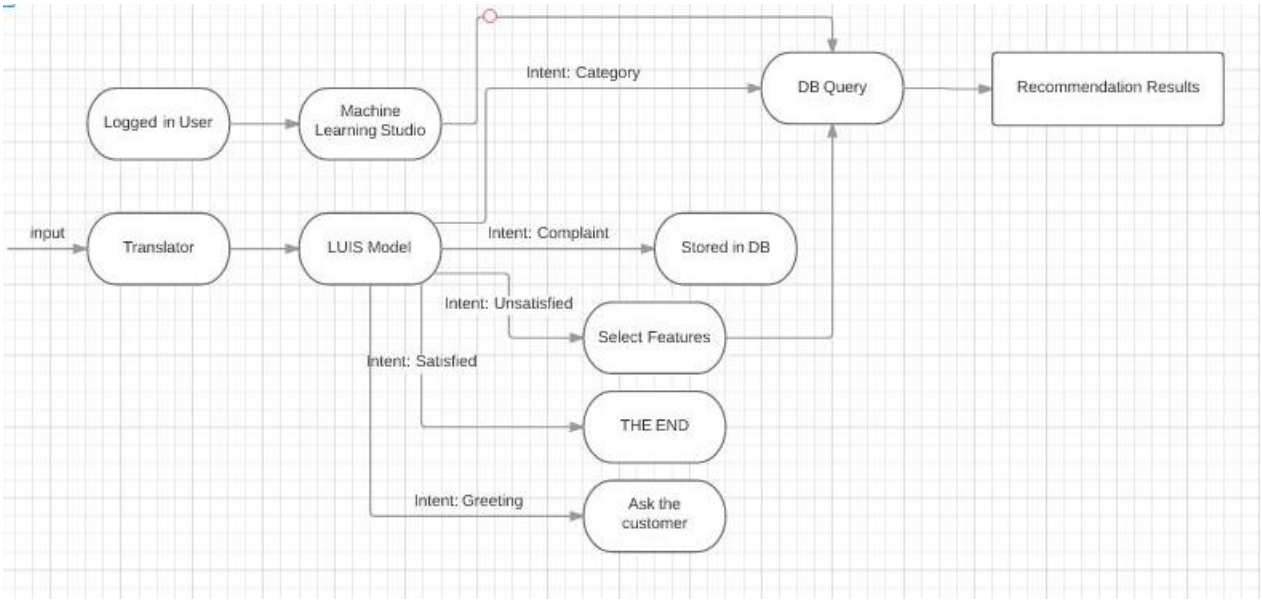


Figure 12: Workflow of the Bot

If the customer is a logged in user with personal profile, the Bot will then first of all run the recommendation engine and recommend the customer with the products having highest ratings from other customers with similar characteristics.

The Bot Demo in §9.3 will further illustrate the workflow above by screenshots.

7 Individual Contributions

This sections lists my main contributions to the whole project and my self-evaluation concerning what I have achieved.

7.1 Database Management and Query

I am fully responsible for managing everything related to the database. I set up the service of SQL Database on Azure, created tables in the Database, imported the device catalogue into the Database and connected the Bot to the Database. I also wrote all the

queries to store complaints in to the Database and to retrieve the products that meet customers' requirements. Details of SQL Database are elaborated in §6.6.

7.2 Design and Training of the LUIS model

I was actively involved in the design of the LUIS model. I came up with a certain amount of data to train the language model and did some testing on the model. Details of LUIS are elaborated in §6.4.

7.3 Connection of Machine Learning Studio to the Bot

I created a workspace for Machine Learning Studio on Azure and deployed the experiment model on Machine Learning Studio as a web service. Furthermore, I wrote the code to send response from the Bot to the web service, get back the response and parse it to retrieve the prediction. Details of Machine Learning Studio are elaborated in §6.5.

7.4 Bot Registration on Developer Portal

I registered the Bot on Developer Portal of Bot Framework. The Bot is identified with a messaging endpoint, a Microsoft App ID and its password. I also published the Bot to Azure by creating an Azure app service to host it.

7.5 Bot Deployment of Facebook

I deployed the Bot onto Facebook. To achieve this, I created a Facebook App and a Facebook Page to host the Bot as an app. One thing to mention is since Skype itself is a Microsoft platform, the Bot can be automatically deployed to Skype at the time when it is registered on Developer Portal. However, as Facebook is a third-party platform, it takes a few more steps to deploy the Bot onto Facebook. Screenshots of the Bot on Facebook can be found in §9.3.5.

7.6 Project Webpage Implementation and Management

I independently set up the project webpage using HTML and CSS in order to deliver information and progress of this project. I also constantly update the documentations and the Bot demo on the website. More information about the project webpage can be found in §9.1.

7.7 Self-Evaluation

Personally speaking, I have done a substantial part for this project. The process of creating and configuring applications on platforms is quite tedious and I have to carefully read the documentations before I start. Actually I did encounter some problems when setting up the apps on Azure and doing operations with the database but fortunately all the problems were successfully solved in the end. From this project, I have obtained some experience for using Microsoft platforms and services. I enjoy the teamwork with peer students and the cooperation with Microsoft.

On the other hand, more functionalities of the Bot could have been achieved more if I had spared more time and efforts on this project and if the task allocation had been performed more efficiently within our team.

8 Difficulties and Limitations Encountered

This section discusses the difficulties and limitations we faced during the project and our solutions.

8.1 Connection Error to SQL Database

After setting up the SQL Database on Azure, when I tried to connect to the Database, there was always an error saying the following:

A network-related or instance-specific error occurred while establishing a connection to SQL Server.
The server was not found or was not accessible. Verify that the instance name is correct and that SQL Server is configured to allow remote connections.
(provider: Named Pipes Provider, error: 40 - Could not open a connection to SQL Server)

It had been a long time before I found a solution to this error even though I searched for a lot of posts discussing this issue. This severely dragged the progress of the project since if the Bot couldn't be connected to the Database, it had no data to serve the customer.

Finally, it turned that the connection error came from the Wi-Fi. If I use any of *'Universities WiFi'* or *'HKU'* or *'Wi-Fi.HK via HKU'*, such connection error always arises. Only using *'eduroam'* Wi-Fi can eliminate such error. It seems that Wi-Fi actually can block the connection of the SQL Database. But such circumstance is still quite strange. It remains to be discovered why certain Wi-Fi connections would cause error.

8.2 Expiration of Subscriptions on Azure

All the accounts used for this project were newly created by our team when we started the project. For Microsoft Azure Platform, it only gives a free trail to each account for one month. Samson also provided us with one *Azure Pass* [19] subscription for one month. But this is still not enough to sustain the whole project period. Unfortunately, we failed to subscribe for *Visual Studio Dev Essential* [20] which could give us 25USD per month for free on Azure.

Eventually, we used *Microsoft Imagine* [21], which provides a one-year subscription for university students. Since Translator and Machine Learning Studio Web Service couldn't use Imagine, we use *Pay-As-You-Go* [22] subscription for these two applications.

However, all the apps already created on Azure using free trial and Azure Pass couldn't be moved to other subscriptions directly. Thus, we had no choice but to repeat the tedious process and re-create all these apps again, which was rather time and energy consuming.

8.3 Design of LUIS Models

The primary difficulty of designing LUIS models comes from the complexity of natural languages. A sentence can be extremely complicated even for human beings to understand. Also, each sentence may also have multiple layers of meanings. The second difficulty comes from the limitations of LUIS itself. For each LUIS model, we can only define at most 10 entities. Even though each entity can be hierarchal, LUIS only supports two levels of hierarchy and a parent entity can only have 10 child entities. Thirdly, the training algorithm behind the LUIS model is a black box to us and we don't know what exactly happens when LUIS trains the model. There is no other choice of machine learning algorithms and if the training result turn out not to be that satisfactory, what we can do is to make modifications to the design of intents and entities or give more training data for the model to learn from. Last but not least, whenever the business requirements from Microsoft changes, most likely we have to change the design of the LUIS model and do the labelling of intents and entities again. Thus it became quite challenging for us to tackle these obstacles and at the same time retrieve as much information from an utterance as possible.

In the first semester, we tried to retrieve every possible feature of a digital device in a user input, such as Brand, CPU, RAM, Hard Disk and Price, etc. For those features having numeric values (i.e. RAM, Hard Disk and Price), we even extract the relationship (i.e. larger than, smaller than and equal to). This worked to some extent, but it made the model really complex and the accuracy of detection is far from satisfactory.

After having some further discussion and negotiation with Microsoft, we all agree the point that we need to mainly focus on the usage of the product instead of the technical specifications as not that many customers have technical backgrounds. This makes the design easier and more straightforward. We define the intent as the target category of products and use entities to classify whether the complaint is related to products or service. Details can be found in §6.4.

8.4 Lack of Business Input and Training Data

One huge limitation of this project is that we lack of business input and training data, which is quite critical for the training of LUIS model and the recommendation engine. What Microsoft provides us with is mostly high-level ideas and instructions instead of real business data. We only obtained a digital device catalogue and some sample questions that customers are likely to ask from them. Frankly speaking, these data are far from adequate.

As a consequence, to train the language model, we had to imagine how customer would ask when making a purchase and come up with as many variations of sentences as possible by ourselves. To train the recommendation engine, Microsoft won't disclose the confidential personal information of its customers to us and therefore we have to produce some test data by ourselves for the demonstration purpose. Still, the amount of training data is not enough. After we hand over this project to Microsoft, if Microsoft can feed the recommendation engine with sufficient real user information, then hopefully the result will be quite good.

9 Deliverables and Results

This section presents the major deliverables and results of this project.

9.1 Project Webpage

We have established the project webpage [23] for the update of project progress. The website provides general information of this project including a brief introduction, team members and contact details. All the documentations of this project and the link of Bot to Skype are also available from the website. Figure 13 on the next page gives a screenshot of the project webpage.

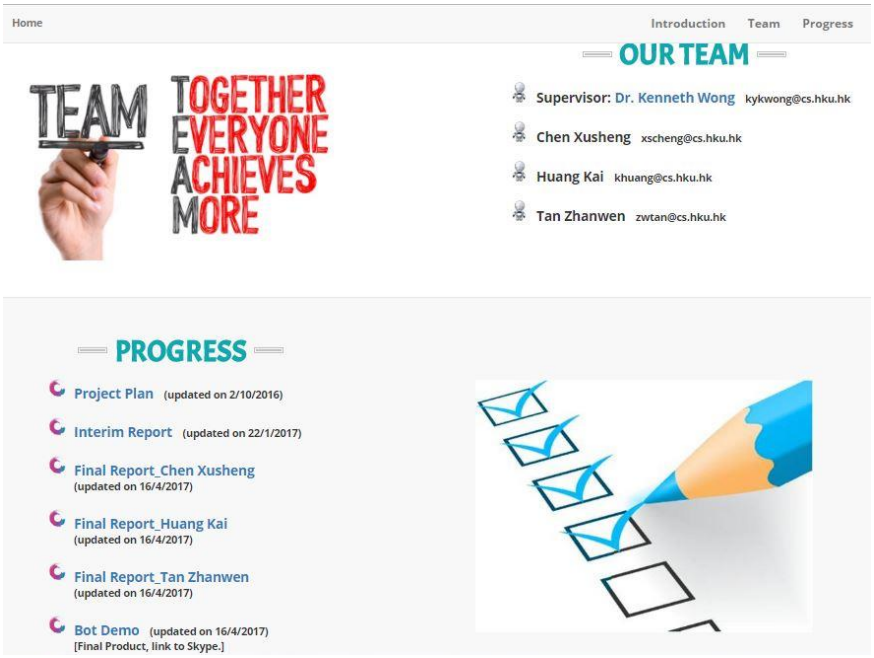


Figure 13: Project Website

9.2 Documentations

There are three documentations for this project, namely project plan, intermediate report and final report.

The project plan discusses the background, objectives, scope of this project and proposes the methodologies, division of labor and schedule.

The intermediate report mainly serves to present our initial implementation of the Bot prototype, supplementing with technical details, difficulties encountered and plans for the next stage.

The final report, namely this document, gives a detailed review and conclusion of the whole project.

9.3 Bot Demo

Our final product is a fully functional AI Chat Bot that supports online conversations with users using natural languages. A user can expect the immediate reply from the Bot after he or she sends a message to it. The Bot can give recommendation on products or

handle customer complaints. This subsection gives a vivid demo of the Bot to show the result of this project.

9.3.1 Recommendation for Product Purchase

Figure 14: Bot Demo (1)

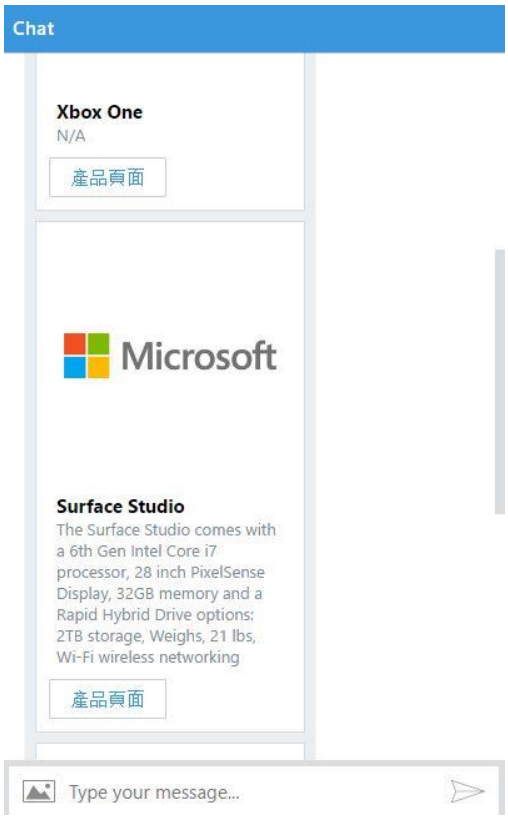


Figure 15: Bot Demo (2)

As shown in Figure 14 and Figure 15 above, when a potential customer starts a conversation by typing in something like '你好', from the response of the LUIS model, the Bot will know that the user intent is 'Greeting' and then respond by saying hi as well and asking the customer what kind of product he or she wants. If the customer tells the Bot a specific usage of the product he or she intends to purchase, the Bot will get the target category that serves well for this certain usage from the LUIS model and recommend several products from this category for the customer to select.

For example, if the customer replies '我鐘意買來打機', the Bot interprets the target category as 'Gaming Devices' and thus recommend the customer with Xbox and some other machines with high computing power (such as Surface Studio). More details of corresponding usage and categories are discussed in §6.4.

An image of each recommended product, a brief introduction and a link to the product purchase page will be provided to the user for his or her convenience. In case there is no product image in the database, a default image (i.e. logo of Microsoft) will be shown. In case there is no description of the product in the database, the Bot will omit this field. In case there is no link of the product in the database, a link to Microsoft online store will be an alternative.

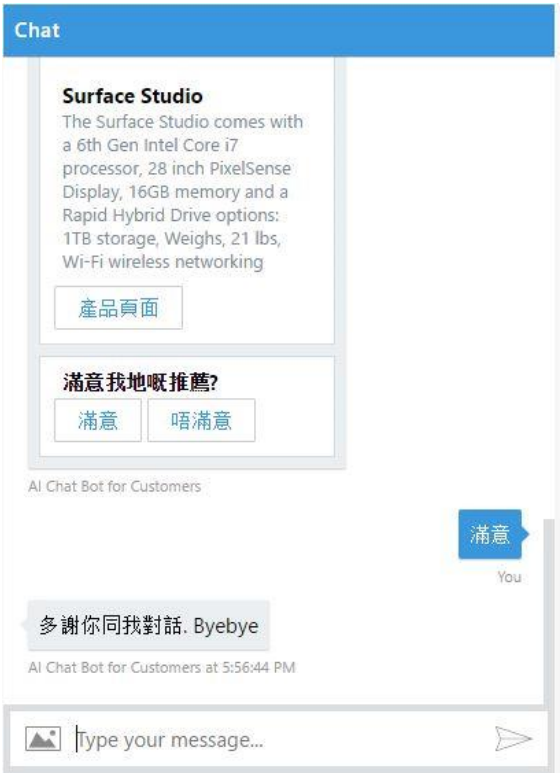


Figure 16: Bot Demo (3)

Afterwards, the Bot will proactively ask whether the customer is satisfied with the recommendation results by popping up buttons. If yes (the case demonstrated in Figure 16 above), the Bot will treat it as the end of the conversation, then say thank you and goodbye to the customer.

However, if the customer is satisfied with none of the products recommended by the Bot (the case demonstrated in Figure 17-18 below), the Bot will provide the customer with several options to select so as to filter the products. The customer can specify whether he or she has requirements on brand, price or category by clicking buttons. If the customer has no requirement on a feature, he or she can simply click ‘No preference’.

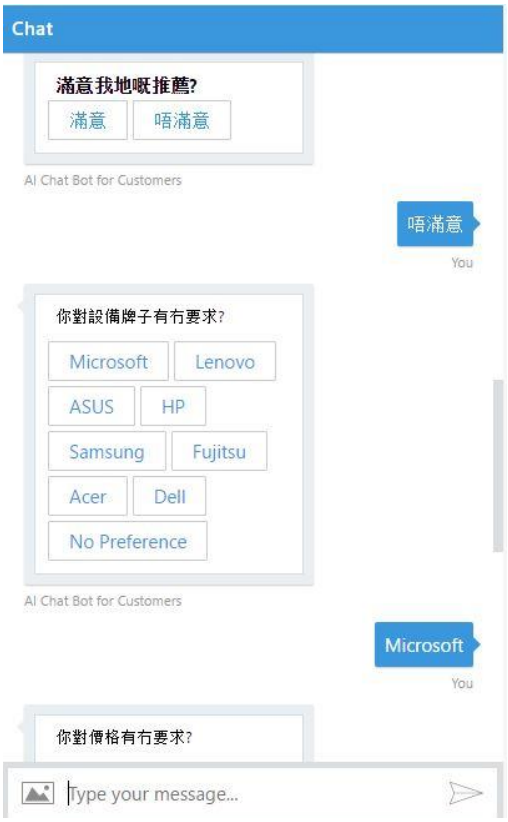


Figure 17: Bot Demo (4)

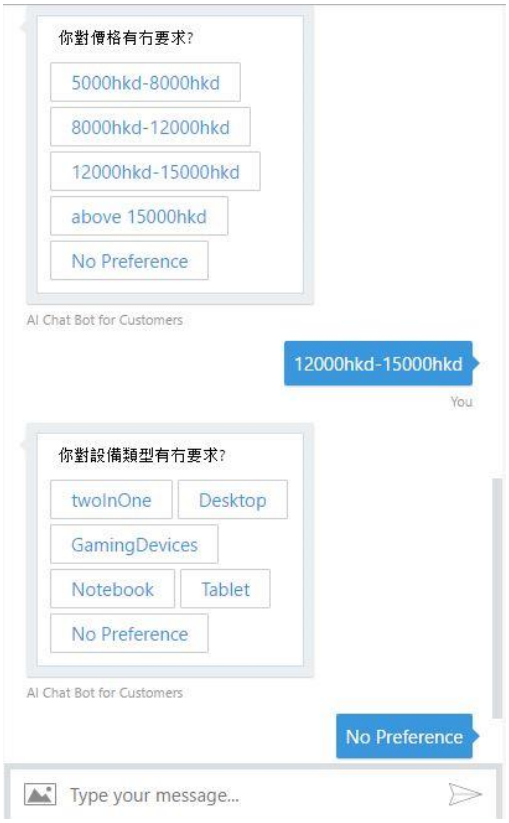


Figure 18: Bot Demo (5)

After the customer finishes choosing preferences for these three features, a confirmation message will be prompted by the Bot, as shown in Figure 19-20 on the next page. The customer can make any modification to the previous choices if he or she wishes. After confirmation, the Bot will search the product category and return several products satisfying these requirements to the customer and this conversation comes to an end.

Figure 19: Bot Demo (6)

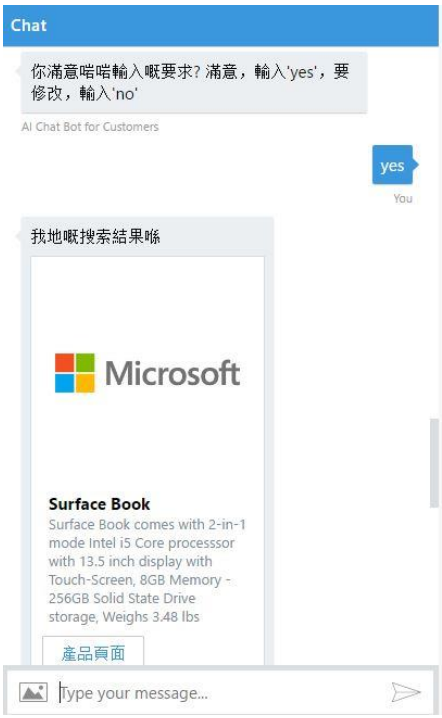


Figure 20: Bot Demo (7)

9.3.2 Customer Complaint

Besides product purchase, the customer can also make a complaint to the Bot as well as shown in Figure 21 on the next page. Any general inquiry or complaint about a particular product or service will be stored into the database for future processing. When the customer inputs a complaint, the Bot will tell him or her that the complaint is well recorded and Microsoft staff will handle it as soon as possible.

In Figure 22 on the next page, we can see that in the database, the complaint '你好，我部 Windows10 今天突然無法使用，開機時某幾個鍵打不了字，該點解決？' is classified as a complaint for Product. We can also view the user id, the name of the customer, the channel he or she is using and the time when the complaint is received from the table. Such customer information will be useful to the staff for future contact.

Figure 21: Bot Demo (8)

UserID	UserName	Channel	Time	Type	Content
29:1MPH7iKQ-yik3UhHCK_UUOl3KOHVySCgKKACNYqmEeXI	Xusheng Chen	skype	2017-03-20 14:49:49.603	Product	Surface Pro 3 电池异常
29:1BU2zfqDNakc0qhU5_Gwa-nWxd9N3Jh___OMmLjhs8Uo	Francis Tan	skype	2017-03-29 03:31:33.107	Product	windows phone打電話無聲音
29:1HK1128VSatk1fhU5_Gwa-nWxd9N3Jh___OMmLjhs8Uo	Kai Huang	skype	2017-03-29 04:55:38.220	General	手機無法安裝outlook點算?
29:1XC013UTfKL3Jb1qhzjGsRs1ru1QobJkGffgAJbf38Lz...	Samson Lee	skype	2017-04-03 01:02:59.107	Service	點解宜家微軟服務電話無台灣?
2UTxbhIPeDv	You	webchat	2017-04-13 10:40:20.533	Product	你好，我部Windows10今天突然無法使用，開機時某幾個鍵打不了字，該點解決?

Figure 22: Complaint Table in the Database

9.3.3 Recommendation Engine for logged-in users

We simulate the case of logged-in users by the special user input in the format of '@+the id of the customer'. When the customer inputs '@id', the Bot will identify that the user has logged in, retrieve the user profile including the customer's age, gender and occupation information. Then the Bot will run the recommendation engine and recommend him or her with the category of products with the highest ratings from those having similar personal characteristics with the current customer. The recommendation engine is trained using the ratings of previous customers and their personal information. More details of the recommendation engine can be found in §6.5.

For example, in Figure 23 on the next page, the customer with id 70 is 19-year-old student and according to the ratings of customers with similar characteristics, gaming devices have the highest rating and thus some products that provides good experience for playing games are recommended to this customer.

Figure 24 below gives another example of a customer with id 113, who is a 27-year-old office lady and according to the ratings of customers with similar characteristics, tablets are preferred and thus some devices in the category of tablet are recommended to this customer.

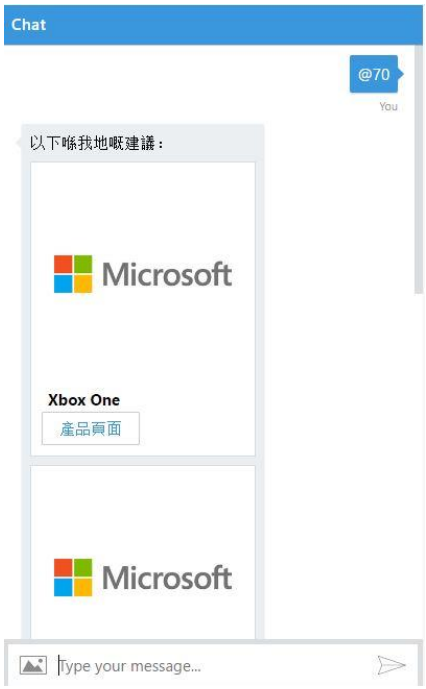


Figure 23: Bot Demo (9)

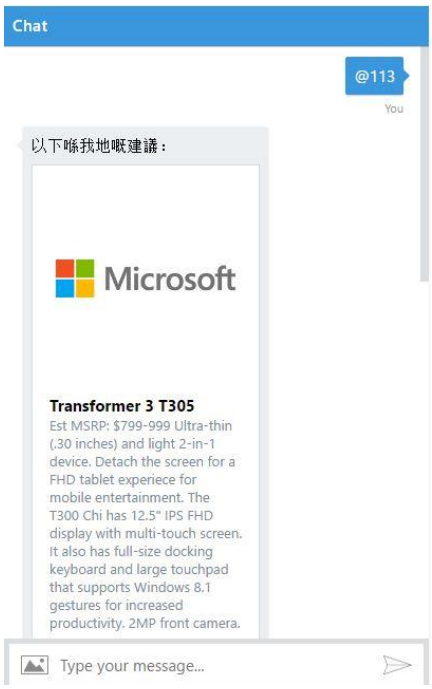


Figure 24: Bot Demo (10)

9.3.4 Bot on Skype

The Bot has been deployed on Skype. Figure 25 on the next presents the conversation interface on Skype when a user chats with the Bot. The interface is exactly the same as chatting with other friends. One can add Bot to the contact list if he or she wishes and talk to the Bot at any time.

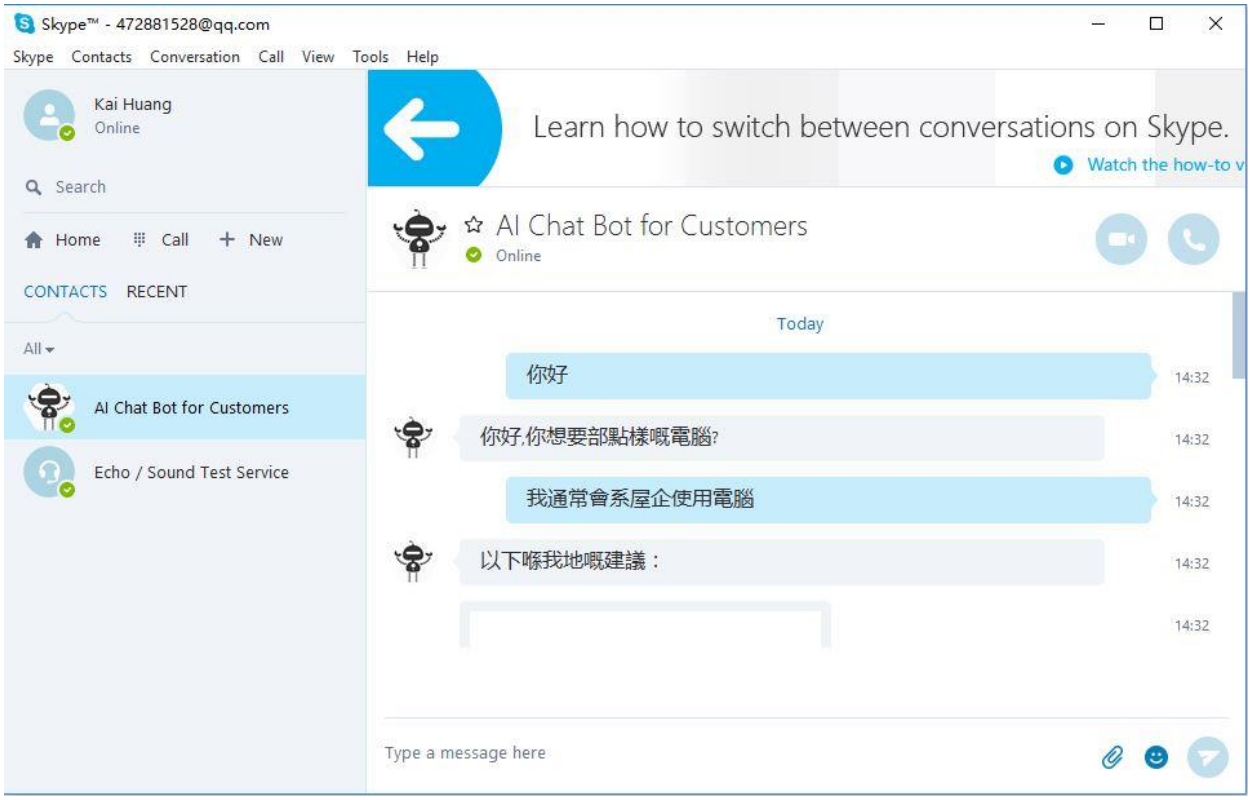


Figure 25: Bot on Skype

9.3.5 Bot on Facebook

The Bot has also been deployed on Facebook. Figure 26 on the next page is a screenshot of the Facebook Page to host the Bot and Figure 27 on the next page presents the conversation interface with the book on Facebook. Users can either access the Bot using Web Facebook or Facebook Messenger.

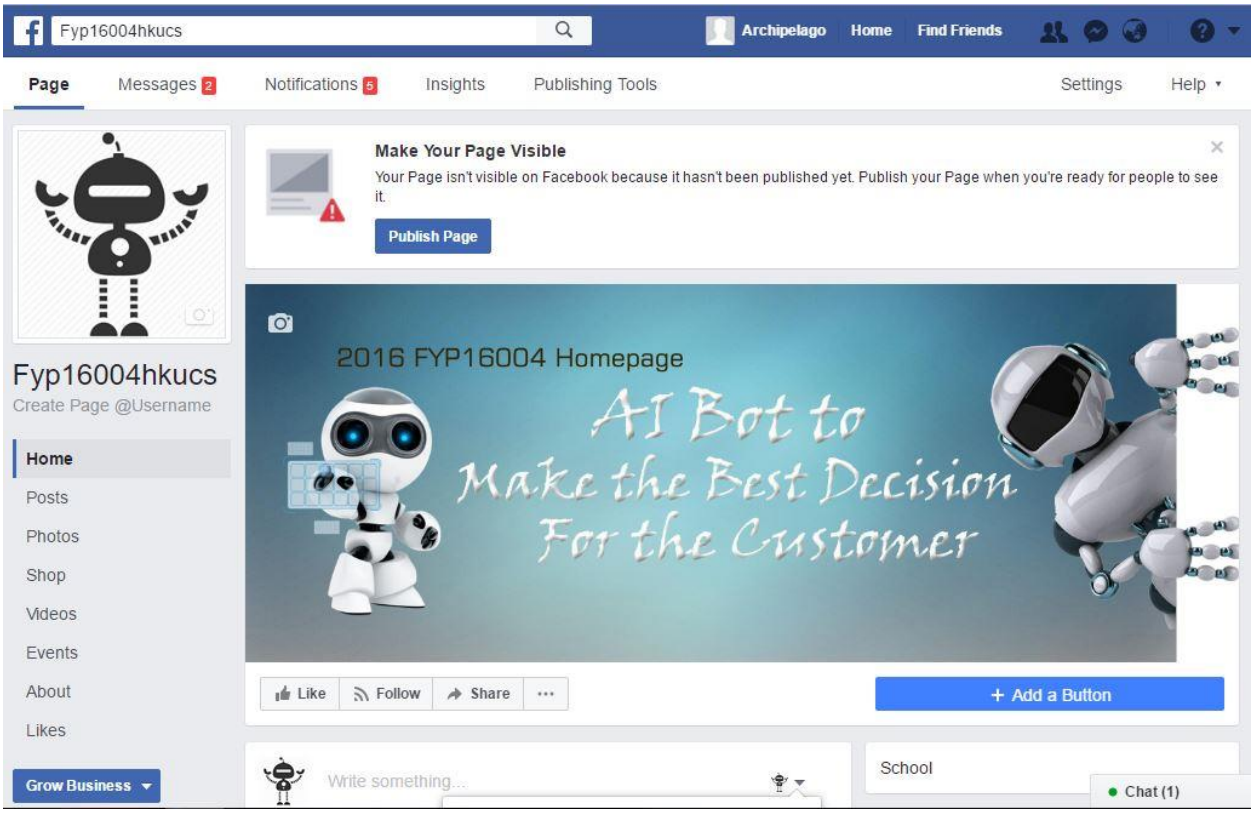


Figure 26: Facebook Page for the Bot

Figure 27: Bot on Facebook

9.3.6 Other Scenarios

When the user wants to chat with the Bot concerning issues other than product recommendation and complaint, the Bot will simply say that it cannot understand what the customer means since the intent returned by the LUIS model is 'None'. These scenarios are out of the scope of this project. At the same time, the Bot will directly ask the customer whether he or she can ask some questions about recommending digital products instead. Figure 28 below illustrates one of these scenarios.

Figure 28: Bot Demo (11)

10 Experiments and Analysis

This section discusses the experiments we conducted to review and evaluate the performance of the Bot and presents the results and our analysis.

10.1 Setup

We kindly invited twenty participants from different faculties and in different years of their study to talk to the Bot and we record their ratings as well as feedbacks. Table 1 on the next page shows the distribution of participants in terms of faculty and their current

year of study. Moreover, 45% of them are local students and the remaining 55% are from Mainland China but they have basic mastery of Cantonese. The experiments were conducted at HKU campus. Around half of the participants used the Bot via Skype and the others tried the Bot via Facebook Messenger.

In order to compensate for the limitation that we have no way to obtain adequate training data for the Bot to learn from, every participant was first told how the Bot would generally behave as illustrated in §9.3 before they started the conversation. After the experiments, we further chatted with them for comments on the Bot.

Faculty	*Number*
Engineering	2
Business	4
Medicine	5
Social Science	6
Arts	2
Science	1
Total	20

Year	*Number*
1	6
2	5
3	1
4	8
Total	20

Table 2: Distribution of Participants

10.2 Results

Each participant is asked to rate on their degree of satisfaction with regard to the user interface, recommendation results and overall performance of the Bot respectively. The rating is represented by an integer within 1 to 10. Besides, we manually verify whether the intent returned from LUIS is correct or not.

It turns out that most participants are pleased with the user interface of the Bot, finding the conversational interface quite comfortable. The average rating for the user interface is 8.75 out of 10, with variance 1.0875. Six of them gave the highest rating 10 and three of them gave the lowest rating 7. We received positive feedbacks from participants appreciating the quick responses of the Bot.

As for the recommendation results, the average rating is 5.45, much lower than the rating for the user interface. The corresponding variance is 2.5475. The highest rating is 8 while the lowest is only 4. This reflects that what the Bot recommends doesn’t really meet the

needs of the customers due to many factors. Some participants commented that they it would be better if the Bot could give them more details about the recommended products such as user experiences by past customers together with some comparison among several similar models before they reached the final decision. We admit that our implementation does have pitfalls and more efforts could have been made for improvement in recommendation.

Participants gave an average overall rating of 6.6 with variance 1.64, highest rating 9 and lowest rating 5. The overall performance of the Bot is acceptable in general but still its performance varies a lot in different circumstances. Participants told us that they expected the Bot to have a longer conversation with them before making recommendations.

The accuracy of LUIS's intent detection is 0.6, which is not too bad. But the result comes under the situation where participants were told beforehand what types of questions they were advised to ask. Those misclassified inputs have been used as the training data for the LUIS model.

With the experiments results and feedbacks from those participants, we can propose future work in §11.

11 Future Work

This section discusses what can be done in the future to further improve the Bot.

(1) As discussed in §8.4, more training and testing data could be collected to improve the accuracy of machine learning models and the performance of the Bot. Before training the Bot, it will be good to make sure that data collected from different sources have compatible quality so that the Bot can have stable performance under difference circumstances.

(2) The interactions during conversations with the Bot can be more user-friendly. For example, the Bot can reply with some emoji to make the customer feel more like talking to a real sales assistant.

(3) Enable the Bot to support English and Simplified Chinese. Our final Bot is only capable of having conversations in Traditional Chinese, and thus the target users are only those Cantonese speakers. Since English, Simplified Chinese and Traditional Chinese are all commonly used in Hong Kong, in order to serve more potential customer, it will be good if the Bot can support multiple language.

Actually in the first semester, we focused on the English model for the Bot. However, as requested by the Microsoft business team, Cantonese is preferred and thus we proceeded with the Cantonese model.

Due to time restrictions, we are not able to combine multiple languages in one single Bot. But it is still feasible to add one more layer to first identify the input language of the customer and pass inputs in different languages to corresponding LUIS models to handle.

(4) Besides recommending computers, the Bot can also recommend some accessories such as earphone, wireless mouse, keyboard, etc.

(5) The flow and logic of the conversation with the Bot can be designed more sophisticated. It would be better if the Bot can detect the intent and make corresponding recommendations after having more interactions with the customers instead of interpreting the intent from a single sentence. Moreover, recommendation or scoring algorithms could be introduced for the sake of more accurate recommendation.

(6) Some advanced features such as speech and facial recognition can be added to the Bot to make it more appealing to customers.

(7) When doing the experiments to evaluate the Bot, participants with diverse personal characteristics can be recruited to make the results less biased. Because of many external factors, we can only invite students within the university to try our Bot. The results can only reflect the Bot’s performance on university students. To make the results more objective, it is advised to invite people among different age groups and with different occupations to participate if possible.

(8) In the future, if necessary, a direct link to the Bot on Microsoft online store can be added to provide another alternative for accessing the Bot.

12 Conclusion

In conclusion, for this project, our team has achieved the general requirements of Microsoft. We keep pace with the proposed schedule in the project and manage to produce a functional AI Chat Bot that aims at assisting Microsoft customers choose suitable digital devices and improving the efficiency of Microsoft staff in handling customer complaints. We have deployed the Bot on Skype and Facebook, which are the two most popular platforms used in Hong Kong.

Throughout the whole academic year, we had regular meetings with our supervisor every week to report our progress and get feedbacks. Besides, we also kept the Microsoft side updated with our progress and asked them for comments on a regular basis.

However, to be frank, what we have actually accomplished does really fall below our original expectations and there is still much space of improvement. But we truthfully value this project as a good opportunity for us to implement an industry-based product, gain some practical experience and get prepared for future career. It is always our privilege to cooperate with Microsoft Hong Kong Limited. It is also our pleasure to have Dr. Kenneth Wong as our supervisor. Hope that the Bot we have developed can have a chance to be further refined to really be the best decision maker for Microsoft customers!

References

[1] Microsoft [Internet]. USA: Microsoft Bot Framework. [cited 2017 Apr 15]. Available from: https://dev.botframework.com/

[2] 1-800-Flowers [Internet]. USA: 1800flowers.com Messenger Bot. [cited 2017 Apr 15] Available from: https://www.messenger.com/t/1800flowers/

[3] Clever Bot [Internet]. UK: Clever Bot. [cited 2017 Apr 15] Available from: http://www.cleverbot.com

[4] Apple Inc [Internet]. USA: Apple online chat support. [cited 2017 Apr 15]. Available from: https://aoschat.apple.com/Chat/getCustomerDetails.do

[5] Microsoft HK Ltd [Internet]. USA: Microsoft online chat support. [cited 2017 Apr 15]. Available from: https://www.microsoft.com/surface/en-hk/support/contact-us

[6] https://github.com/Microsoft/BotBuilder/tree/master/CSharp/Samples

[7] Microsoft [Internet]. USA: Microsoft Visual Studio. [cited 2017 Apr 15]. Available from: https://www.visualstudio.com/

[8] Microsoft [Internet]. USA: Visual C – C#. [cited 2017 Apr 15]. Available from: https://msdn.microsoft.com/en-us/library/kx37x362.aspx

[9] Atlassian Inc [Internet]. USA: Bitbucket. [cited 2017 Apr 15]. Available from: https://bitbucket.org/

[10] Microsoft [Internet]. USA: Bot Framework Emulator. [cited 2017 Apr 15]. Available from: https://docs.botframework.com/en-us/tools/bot-framework-emulator/

[11] Microsoft [Internet]. USA: Microsoft Azure. [cited 2017 Apr 15]. Available from: https://azure.microsoft.com/en-us/

[12] Microsoft [Internet]. USA: Language Understanding Intelligent Service. [cited 2017 Apr 15]. Available from: https://www.microsoft.com/cognitive-services/en-us/language-understanding-intelligent-service-luis.aspx

[13] Microsoft [Internet]. USA: What is Azure Machine Learning Studio? [cited 2017 Apr 15]. Available from: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-what-is-ml-studio

[14] Microsoft [Internet]. USA: Microsoft Azure SQL Database [cited 2017 Apr 15]. Available from: https://azure.microsoft.com/en-us/services/sql-database/?b=16.50

[15] Microsoft [Internet]. USA: SQL Server Management Studio (SSMS) [cited 2017 Apr 15]. Available from: https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms

[16] USMAN_MALINK [Internet]. Udemy Blog. [update 2014 May 29, cited 2017 Apr 15]. Available from: https://blog.udemy.com/ado-net-entity-data-model/
[17] Microsoft [Internet]. USA: LINQ: .NET Language-Integrated Query [cited 2017 Apr 15]. Available from: https://msdn.microsoft.com/en-us/library/bb308959.aspx
[18] Microsoft [Internet]. USA: Cognitive Services [cited 2017 Apr 15]. Available from: https://www.microsoft.com/cognitive-services/en-us/apis
[19] Microsoft [Internet]. USA: Microsoft Azure Pass. [cited 2017 Apr 15]. Available from: https://azure.microsoft.com/en-us/offers/azure-pass/?wt.mc_id=AID607388_SEM_ti4HGycK
[20] Microsoft [Internet]. USA: Microsoft Azure Join Visual Studio Dev Essentials. [cited 2017 Apr 15]. Available from: https://azure.microsoft.com/en-us/pricing/member-offers/vs-dev-essentials/?wt.mc_id=AID_SEM_ti4HGycK
[21] Microsoft [Internet]. USA: Microsoft Imagine. [cited 2017 Apr 15]. Available from: https://imagine.microsoft.com/en-us
[22] Microsoft [Internet]. USA: Microsoft Azure Pay-As-You-Go. [cited 2017 Apr 15]. Available from: https://azure.microsoft.com/en-us/offers/ms-azr-0003p/
[23] https://i.cs.hku.hk/~fyp16004/