THE UNIVERSITY OF HONG KONG

DEPARTMENT OF

COMPUTER SCIENCE

Interim Report

[Industrial Based]

AI Bot to Make the Best Decision for the Customer

Course Information:

COMP4801 Final Year Project

Supervisor: Dr. Kenneth Wong

Team members:

Chen Xusheng, Michael 3035028520

Huang Kai, Kevin 3035086340

Tan Zhanwen, Francis 3035028518

Client: Microsoft Hong Kong Limited

Date of Submission: Jan 22nd, 2017

Abstract

As an industry giant in IT products, Microsoft produces a large number of digital products that are quite alike in many aspects, which makes it difficult for customers to distinguish among similar models and select the most suitable one. This project is an industrial-based project initiated by Microsoft Hong Kong Limited, which aims to solve the aforementioned problem by developing an AI Chat Bot that communicates with customers using online conversations in natural languages and recommend them with the particular product that fits their needs best.

Up to now, a bot prototype has been successfully built and hosted on Microsoft Azure platform, which can understand human languages and handle purchase or complaint requests from customers. Also, our preliminary presentation was delivered on Jan 10th, 2017 to for examiners to review progress and make comments. In addition, we have met the business representative from Microsoft on Jan 17th, 2017 to report our progress and get up-to-date requirements.

We plan to improve the user interaction with our bot, based on the interaction design to be provided by Microsoft, in the next stage. Also, we will deploy our bot to Facebook Messenger. After that, we plan to develop and train Cantonese language model. In addition, we will implement email alert whenever a complaint received.

Acknowledgements

The progress of the project *AI Bot to Make the Best Decisions for the Customer* could not have been made without the help of following people. We would like to express our most sincere gratitude to them.

Dr. Kenneth Wong, supervisor of this project, for giving directions and support throughout the designing and implementation of this project.

Dr. Anthony Tam, second examiner of this project, for facilitaing industrial connection and giving valuable comments during our preliminary presentation.

Mr. Samson Lee, contact person from Microsoft, for providing useful materials and helping us tackle technical problems.

Table of Contents

1 Introduction

Back in the late 20th century, information technology was not advanced and there were few choices for digital products provided by Microsoft, such as laptops, smart phones. Therefore, customers did not bother to consider which model to buy during that period as the choices were limited. However, due to the advancement of technology and recent expansion of its market, nowadays Microsoft provides considerable choices for a specific type of product. Even for a particular series, customers are prompted to choose from customized specifications with different CPU, RAM, hard disk space, etc, making it difficult for customers without much technical background to select the most suitable product.

Three main constraints exist when customers of Microsoft would like to query the differences among models in order to find the most suitable one. Firstly, the transformation from on-site to online shopping renders it difficult for customers to seek helps from shop assistants. Secondly, it costs Microsoft much money to hire technicians for providing online support to customers. Thirdly, the service hours of online support is restricted to a certain period of time within a day, as a result of which customers cannot get helps online during other time slots.

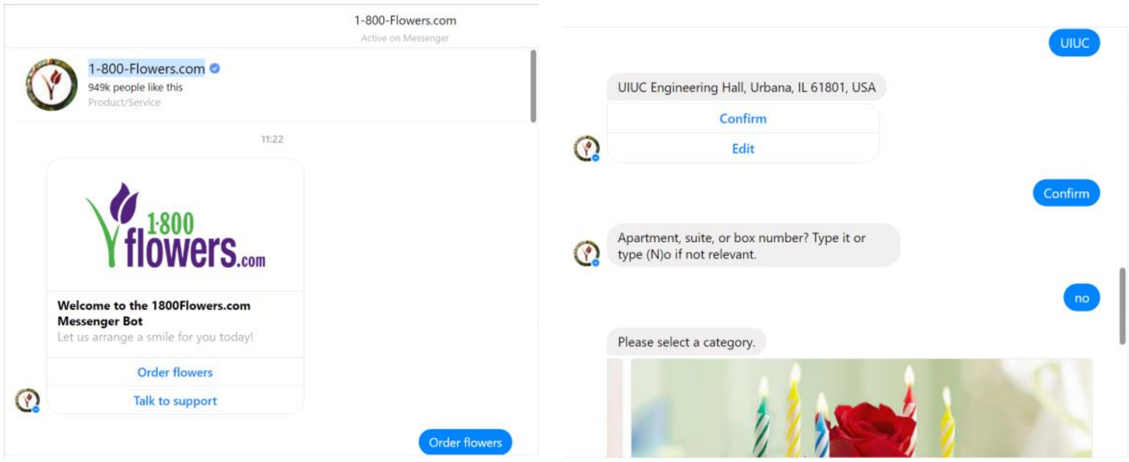
This project aims to provide a solution by developing an artificial intelligent chat box as an extensible standalone module to major social network websites or applications which can understand the customers' queries in natural languages and respond correspondingly, so that customers can ask for recommendation when purchasing Microsoft products anytime while human resources are saved. There are two main challenges for us to overcome in order to successfully develop our final product. One is that the language models in English, Cantonese and Mandarin need to be carefully designed to tailor to requirements from Microsoft. The other is that large amount of data relevant to Microsoft product queries needs to be collected and labelled manually, so that our bot is well trained to answers queries specifically for Microsoft products.

The remaining of this interim report proceeds as follows. In the first place, the previous work in the field and overall objectives of our product are provided, followed by meeting contents with Microsoft and project scope. Then we present our technical details, implementations, together with current progress. Next, future plans and limitations and difficulties encountered are clearly stated. Finally, we summarize this report with a conclusion.

2 Previous Work

2.1 AI Bots

There are some other companies using AI recommendation bots to assist customers in buying specific products. *Figure 1* below is an AI bot for an online flower delivery store[1]:



1-800-Flowers.com
Active on Messenger

1-800-Flowers.com
949k people like this
Product/Service

11:22

Welcome to the 1800Flowers.com Messenger Bot
Let us arrange a smile for you today!

UIUC

UIUC Engineering Hall, Urbana, IL 61801, USA

Apartment, suite, or box number? Type it or type (N)o if not relevant.

no

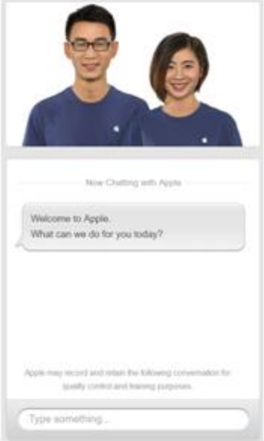
Please select a category.

Figure 1: Conversation with another AI bot

The aforementioned AI bot is hosted on Facebook Messenger and waiting for queries from customers. However, as shown in *Figure 1*, basically it just requests information by asking customers to select a predefined answer from pop-up boxes, which is unnatural and inflexible. Our goal is to develop a similar bot specifically for Microsoft products, while customers can query in their own languages, instead of selecting from predefined answers.

2.2 Online Support

There are some existent chat boxes providing online support to customers for companies like Microsoft and Apple as shown in *Figure 2* [2] and *Figure 3* [3] below:



Now Chatting with Apple

Welcome to Apple.
What can we do for you today?

Apple may record and retain the following conversation for quality control and training purposes.

Figure 2: Apple online support

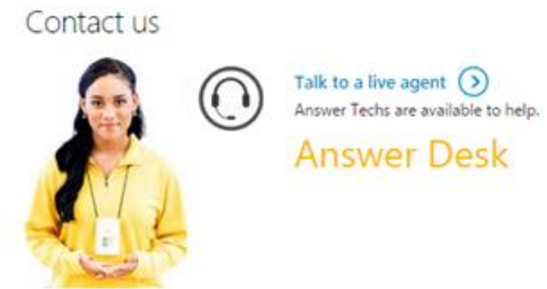


Figure 3: Microsoft online support

For both the chat boxes, the one talking to customers is a real person instead of an AI program. After we typed in some questions, it took around 10 to 20 seconds to get responses. In addition, the service hours are 9am-5pm and 9am-9pm for Apple and Microsoft respectively, which is inflexible for customers who purchase late at night.

Different from the two chat boxes above, our bot will be an AI program that is always online and can respond immediately, so that customers can query anytime and get responses in a timely manner.

3 Objectives

This project aims at creating an artificial intelligent chat box to be used by Microsoft Hong Kong Limited that can understand customers’ queries in natural languages through online conversations, and then recommend the most suitable products to the customers. The chat box can be deployed on different channels. After successful implementation of the project, basic queries in both Chinese and English can be well understood by our chat box with satisfactory responses returned.

Our bot has four distinctive features. Firstly, instead of prompting customers to select from predefined multiple choices, as many other AI bots in the market do, our bot allows customers to query in their own languages. Secondly, relevant information of customers will be stored in the database during conversations, so that the information can be retrieved in the future to facilitate the process of purchase. In addition, machine learning techniques will be used so that our bot becomes more intelligent as it talks to more customers. Last but not least, our bot can be deployed to various platforms, including Skype, Facebook Messenger, so that it can be easily accessible to customers.

4 Meetings with Microsoft

We have had three meetings with Microsoft up to now. During our last meeting, we agreed to split the development into the following 4 modules with details as follows:

4.1 Purchase handling

Things to improve on the current bot:

- Improve user experience by categorizing users at the beginning, for example, into computer science experts/students and those without much knowledge in computers. Then we can deal with users of different categories in a different way. For example, for those without much technical background, they may be more concerned about the color, screen size or the weight of the product. Others may be more interested in technical specifications like processor, RAM, graphic cards, etc. Anna Chow, the business representative of Microsoft, will contact their promoter, who is experienced in dealing with customers' inquiries, to give us a more reasonable interaction flow.

- Deploy the bot on Facebook Messenger, as it is more popular than Skype in Hong Kong.
- Provide the photos, or link of the products in our recommendations to make the conversation more vivid and user-friendly.
- Cantonese version is more important than the English version.
- Anna Chow will also get their dynamic database storing product information for us, so that we can point to their real database, instead of using the static database now with product information manually input by us.
- Some emojis might be needed to make the responses from bot more user-friendly.
- Probably further promotion or recommendation for other items such as Microsoft Office should be added after the purchase.
- Recommend at least three products so that the user can select their favorite one.

4.2 Complaints handling

- In addition to store the complaints into DB, Microsoft would like to alert the contact person by email. For example, if there is a complaint about model X, we should store this complaint into the database, and alert the person in charge of model X by emails.
- Microsoft will provide us some real-life examples, possibly in text file, from their call center about handling complaints, so that we can use them to train our language model.

The above two module are of high priority. Other modules that are important but not necessarily in the scope are:

4.3 Information collection

- Questions should be prompted up after users purchase some products using our Bot, e.g., in a Q&A mode to collect information from users and store in the db. Possibly there should be some checkings on input from users.
- For this module, the requirements are not very clear. And we will develop it only after the first two modules are satisfactory enough.

4.4 Support

To handle some questions that users frequently ask. For example, what to do if I can't log in Microsoft account? What to do if something else goes wrong. For the standard questions, there should be a template, which will be pretty much similar to the training of a new employee for dealing with such kind of services. If the user has some other questions that can't be handled directly by the bot, the bot will make an appointment with the user so that a real person will phone back to deal with the issue.

5 Scope

The project scope is defined in five aspects, namely the language models, bot interaction, models for recommendation, host platforms, and handling complaints.

5.1 Language Models

Language models in both Cantonese and English need to be developed and trained using the LUIS interface for this project, with Cantonese one in the first priority. Language models in other human languages are not taken into account for this project.

5.2 Bot interaction

Certain types of queries should be able to be handled for all language models. For this project, query types to be handled are "Greeting", "Purchase" and "Complaint". For example, when customers type "I want to buy a laptop for playing games", or "I want a PC", both of these two queries should be categorized as "Purchase". Queries like "I want to complain about the model I just bought", or "There is something wrong with your product XXX" should be categorized as "Complaint". Based on the query type, certain responses should be made. The way our bot responses to the users, namely the interaction design, will be based on the interaction flow to be provided by Microsoft. Anything that's not specified in the interaction flow is not in our scope. In addition, other query types are not in the scope either and simply "I don't understand" is returned by our bot.

5.3 Models for Recommendation

For "purchase" query type, models will be recommended at the end of conversation. Models for recommendation will be confined to the models in the database provided by Microsoft. Also, top three most relevant models should be recommended to users for their selection, with corresponding photos or links attached.

5.4 Host Platforms

The bot should be deployed on Facebook Messenger and Skype through Bot Framework portal from Microsoft. The bot should also be accessible from Microsoft online store. Other platforms, such as Slack are not specified by Microsoft and thus not in our scope.

5.5 Complaints handling

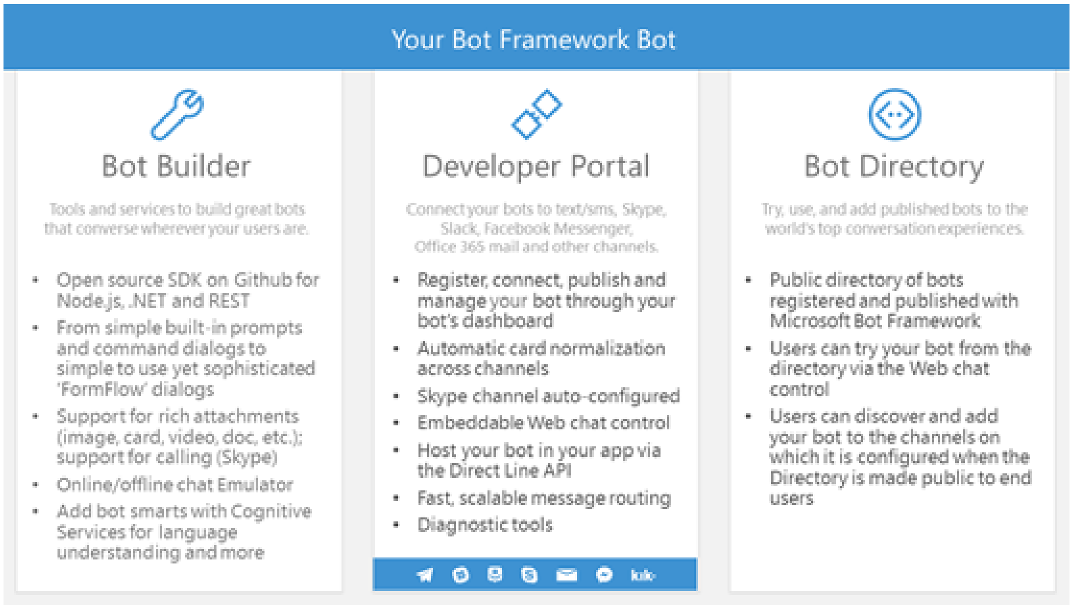
For "Complaint" query type, the complaints from users should be stored in the database. In addition, an alert email should be sent to the person in charge for a specific type of complaints.

6 Technical Details

6.1 Bot Framework

Microsoft Bot Framework [4] is primarily employed as the basis of our bot, which facilitates to build and deploy high quality bots. The framework provides tools to easily develop intelligent bots including basic I/O, language and dialogue skills as well as user connection interfaces.

Bot Framework consists of three major components: Bot Builder SDK, Developer Portal and Bot Directory. *Figure 4* gives a brief overview of these components. The framework also contains an emulator so that we can test our bot locally.



Your Bot Framework Bot

Bot Builder

Tools and services to build great bots that converse wherever your users are.

- Open source SDK on Github for Node.js, .NET and REST
- From simple built-in prompts and command dialogs to simple to use yet sophisticated 'FormFlow' dialogs
- Support for rich attachments (image, card, video, doc, etc.); support for calling (Skype)
- Online/offline chat Emulator
- Add bot smarts with Cognitive Services for language understanding and more

Developer Portal

Connect your bots to text/sms, Skype, Slack, Facebook Messenger, Office 365 mail and other channels.

- Register, connect, publish and manage your bot through your bot's dashboard
- Automatic card normalization across channels
- Skype channel auto-configured
- Embeddable Web chat control
- Host your bot in your app via the Direct Line API
- Fast, scalable message routing
- Diagnostic tools

Bot Directory

Try, use, and add published bots to the world's top conversation experiences.

- Public directory of bots registered and published with Microsoft Bot Framework
- Users can try your bot from the directory via the Web chat control
- Users can discover and add your bot to the channels on which it is configured when the Directory is made public to end users

Figure 4: Bot Framework Overview

Some sample bots with different functionalities developed by Microsoft using Bot Framework can be found on GitHub: [5].

6.2 Language Understanding Intelligent Service (LUIS)

In our project, we highly rely on the Microsoft's Language Understanding Intelligent Service (LUIS) [6] to analyze natural languages. In this section, we will firstly give a brief introduction about LUIS, and then describe the challenges we have met while using LUIS and elaborate our solutions.

6.2.1 LUIS Overview

LUIS is a service built by Microsoft based on machine learning algorithms. Users can create language models on the platform, train them with labeled data and deploy the models as web services. There are two important concepts in LUIS, namely *intent* and *entity*. *Intent* describes what intention the speaker want to convey while saying the sentence. Each input sentence will be categorized into exactly one *intent*, which is similar to the concept of *class* in supervised learning. *Entities* are the useful information in input sentences. Each *entity* will be represented as a pair of *type* and *value*, with *value* being a specific substring from the input and *type* showing what kind of useful information the substring shows. For example, if a customer says "*I want to buy a laptop for text editing*". The *intent* can be categorized as "*purchase*" and the *entities* will be two type-value pairs as *{product: "laptop", usage: "text editing"}.* In LUIS, the machine learning algorithm and training logic is handled by the platform, while users need to define *intent* and *entities* clearly and provide sufficient labeled training data.

Besides, LUIS has two special types of entity: *hierarchical entity* and *composite entity*. *Hierarchical entity* enable an entity have more specific children entities. *Children entities* are of different types but sharing some same characteristics. For example, in our project, we have a parent entity called "*feature*" and several children entities like "cpu type", "memory size" and "disk size". *Composite entity*, on the other hand, shows the associative pattern among entities, which can be thought of as "*hasA*" relationships. For example, the sentence "I want 3 medium pizza" has three simple entities: "number", "size", "item", while we can define a composite entity as the type of "order" to associate them together.

6.2.2 Challenges and solutions

However, even with the help of LUIS, we still have met several technical challenges while trying to analyze customer inputs. First, LUIS is closed sourced which means we can only view the training logic as a black box. When the training performance is not optimal, what we can do is try to provide more training data or to change our design of *intent* and *entity*. Moreover, LUIS has a very strict restriction on the number of entities. We can define only 10 entities of each type, and the *hierarchical entity* can only have two levels, parent and child. While we are making designs on the *intent* and *entities* we must keep this restriction into consideration.

Second, while the type-value pair seems very reasonable for human reading but it is not easy for programmatic handling. The entity value is a substring of any format but it is still anonymous for our bot. In cope with this problem, we divide the entities into two categories. The first is those entities whose information is in the substring itself, such as the *size of memory* or the *price range*. For these entities, we use the returned substring directly in our bot. In order to facilitate data handling, when we train the model, we label these entities to contain only the values to be taken out. For example, rather than label “8GB of RAM”, we will only label “8GB”. The other group of entities are those whose “type” rather than the substring itself matters. We will summarize the possible “subtypes” and do the justification on the subtypes. For example, the “usage” entity representing what the customer want to buy computer for belongs to this type. The customer may have multiple ways of saying he want to use the computer to play games, like “game playing”, “play computer games”, “gaming”. The bot can never understand the substring, but we will put all of them in the subtype “usage::game”, and do a switch on the subtype.

Third, when customers explain their requirement, they usually tends to give a range rather than specific value. No one will say he want to buy a computer for exactly 8000 dollars. To solve this problem, our design has a special entity called “*relation*”, which belongs to the second group of entities mentioned in last paragraph, with subtypes “greater”, “smaller”, “approx”, “without” etc. We also add one composite entity to show the association of “relation” and other feature entities. For example, for the input “*I want to buy a laptop with at least 8GB memory*”, “at least” is an entity of type “relation::greater”, “8GB” is of type “RAM” and “at least 8GB” is a composite entity showing that “at least” is used to describe the ram size.

Although our current design serves well for our prototype, we know that this design may face scalability problem, mainly because of the limitation of LUIS. LUIS restricts the number of entities to only 10. If we have more features to realize in future and have more complicated logic, we may exceed that limit. One alternative approach is to have multiple hierarchical LUIS model, with the children model performing further analysis on the entity substring returned by parent model. However, that will increase the training complexity and reduce the accuracy of the whole language model. Moreover, it will also increase the response time which is critical for chat bot. Currently, we only keep this approach as a backup and will use it only if necessary.

6.3 SQL Database

SQL Database is created on Azure [7] and used for the bot to store and access data. The bot should be configured to connect to the database when it is published to Azure. Each table in SQL Database is converted as an ADO.NET Entity Data Model [8] in C# to allow programmatic access to the database. There are two tables in the SQL Database. One is the sample product catalogue with fields: *Item Name, Processor, RAM, Graphic Card, Hard Drive, Price, and Brand*. The other is to store complaints with fields: *userID, Channel, Date, and Complaint Details*.

7 Implementations

When a user inputs a sentence to our bot, the input is processed in the following flow:

7.1 Processed by the LUIS language model

As a user inputs something to our bot, possibly through Messenger or Skype, the input will be firstly sent to our trained language model for its labelling. Our language model will classify the input to one of the intents, and label the entities within the input sentence. Then the labelled result will be returned to our bot hosted on Azure in a JSON format.

7.2 Processed by Bot Framework

The JSON string from LUIS will be received by our bot developed using Bot Framework. The JSON string is parsed by our bot, and the intent and entities are extracted for further processing. Corresponding response will be made to users based on the intent. For example, if the intent is “greeting”, simply a predefined sentence “Hi, what can I do for you?” will be returned. If the intent is “purchase”, our bot will guide the user through the purchase by pop-up questions, recommendations, etc.

7.3 Database Query and Insertion

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

8 Current Progress

This section presents our work in the first semester as well as the current deliverables.

8.1 Project Webpage

We have established the project webpage [9], which provides a brief introduction of our project together with team members and contact details. One can also review the project progress from this webpage.

8.2 Detailed Project Plan and Interim Report

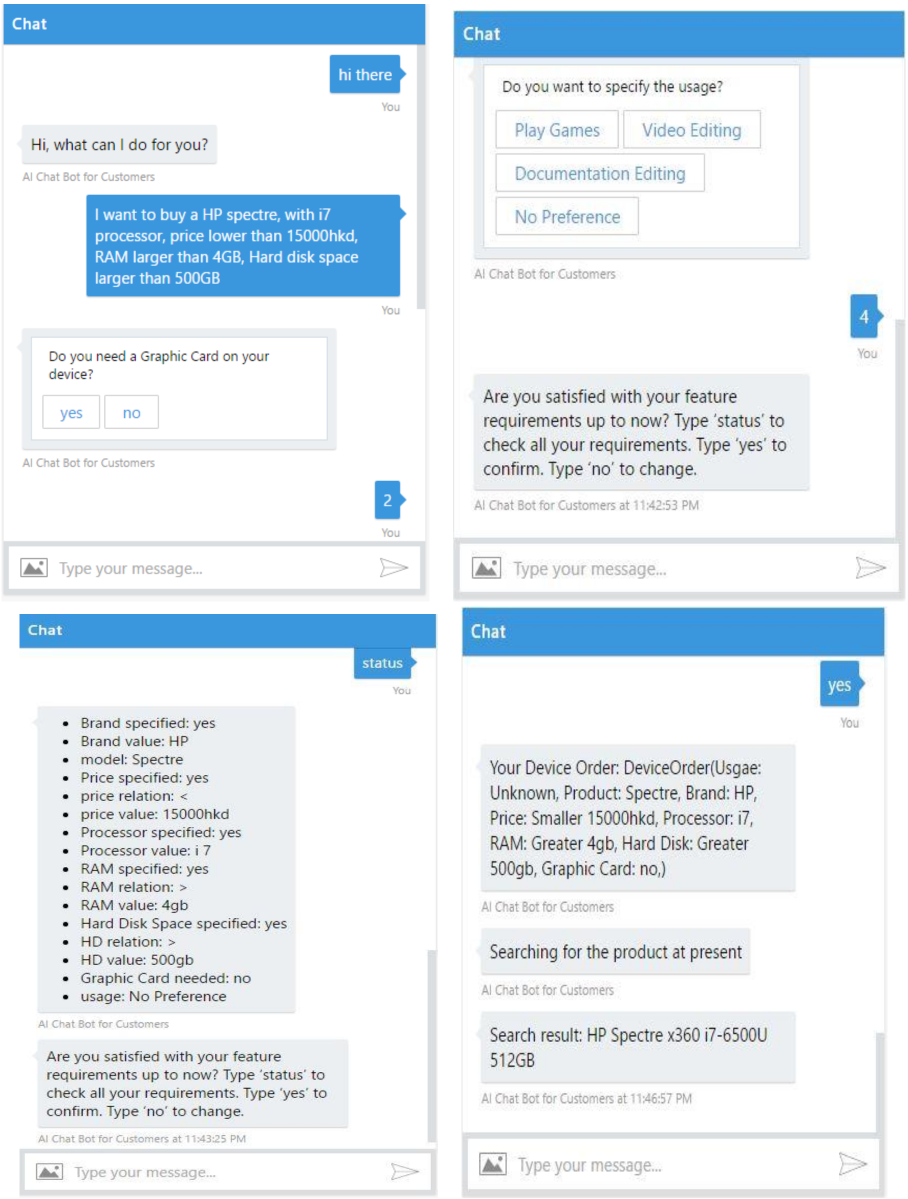
The project plan mainly contains the motivation, objectives, scope, methodologies and tentative schedule of this project. The interim report, namely this document, serves to discuss the technical details of this project, present our current progress, review the initial results, describe the difficulties encountered and propose future plans.

These two documents are both available on the project webpage.

8.3 Bot Prototype

We have implemented a functional bot prototype which is able to understand the intent and entities in a user input by using LUIS (details discussed in §6.2), ask follow-up questions, search the SQL database based on user demands and finally recommend the product that meets these requirements to the customer.

The bot demo can be accessed from the link on the project webpage as well. One can try talking to the bot in similar manners illustrated by the following screenshots.


Chat
You
AI Chat Bot for Customers
yes
no
Type your message...
Chat
Play Games
Video Editing
Documentation Editing
No Preference
You
AI Chat Bot for Customers
AI Chat Bot for Customers at 11:42:53 PM
Type your message...
Chat
You
AI Chat Bot for Customers
AI Chat Bot for Customers at 11:43:25 PM
Type your message...
Chat
You
AI Chat Bot for Customers
AI Chat Bot for Customers at 11:46:57 PM
Type your message...

hi there

Hi, what can I do for you?

I want to buy a HP spectre, with i7 processor, price lower than 15000hkd, RAM larger than 4GB, Hard disk space larger than 500GB

Do you need a Graphic Card on your device?

2

Do you want to specify the usage?

4

Are you satisfied with your feature requirements up to now? Type 'status' to check all your requirements. Type 'yes' to confirm. Type 'no' to change.

status

- Brand specified: yes
- Brand value: HP
- model: Spectre
- Price specified: yes
- price relation: <
- price value: 15000hkd
- Processor specified: yes
- Processor value: i 7
- RAM specified: yes
- RAM relation: >
- RAM value: 4gb
- Hard Disk Space specified: yes
- HD relation: >
- HD value: 500gb
- Graphic Card needed: no
- usage: No Preference

Are you satisfied with your feature requirements up to now? Type 'status' to check all your requirements. Type 'yes' to confirm. Type 'no' to change.

yes

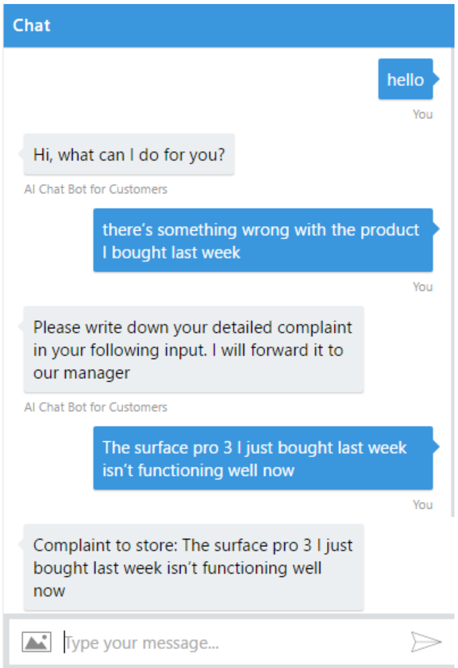
Your Device Order: DeviceOrder(Usgae: Unknown, Product: Spectre, Brand: HP, Price: Smaller 15000hkd, Processor: i7, RAM: Greater 4gb, Hard Disk: Greater 500gb, Graphic Card: no,)

Searching for the product at present

Search result: HP Spectre x360 i7-6500U 512GB

Figure 5: Conversation with the bot for purchase

As shown in *Figure 5* above, when you initiate a conversation by saying ‘hi there’, the bot will interpret the intent as ‘greeting’ and respond by saying hi and asking what it can do to help you. If you tell the bot that you want to buy a certain product with some particular requirements on technical specifications (such as ‘I want to buy a HP spectre, with i7 processor, price lower than 15000hkd, RAM larger than 4GB, Hard disk space larger than 500GB’ in this case), the bot will know your ‘purchase’ intent and parse the entities corresponding to the features, including brand (‘HP’), model (‘Spectre’), RAM size (4GB) with relationship (larger than >), etc. Then the bot will proactively ask you whether you have requirements on the other features that are not specified in your previous input and you can make decisions by clicking buttons. Afterwards, you can type ‘status’ to view your requirements for the product that you intend to purchase. You can also type ‘yes’ to proceed your order or ‘no’ to make modifications on previous specified requirements. Eventually after confirmation, the bot will search the database of product category and return the desired product back to you (‘HP Spectre x360 i7-6500U 512GB’ in this case).



hello

Hi, what can I do for you?

there’s 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

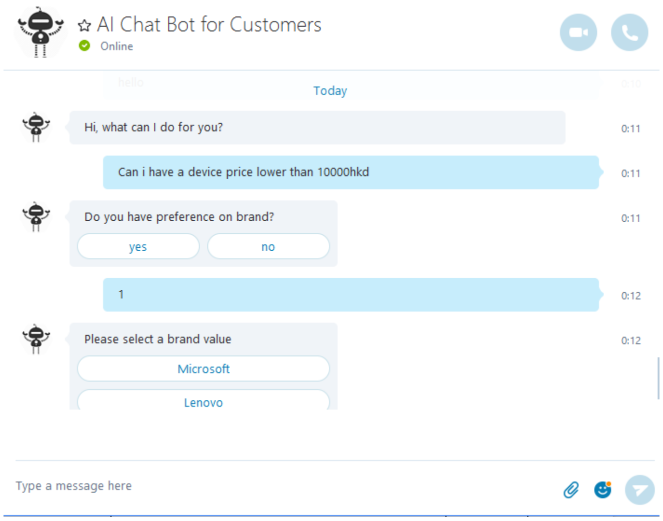
The surface pro 3 I just bought last week isn’t functioning well now

Complaint to store: The surface pro 3 I just bought last week isn’t functioning well now

Figure 6: Conversation with the bot for complaint

Besides purchase, you can also make a complaint to the bot about a particular service or model as demonstrated in *Figure 6* above. In this case, the bot will store your detailed complaint in the database for those in charge of customer services to handle.

Furthermore, we have published our bot to Microsoft Azure platform and deployed it on Skype so that users can directly access it using Skype. Users can add the bot to the contact list and it will always be online waiting for users to have a chat with it. *Figure 7* presents a screenshot of the conversation interface on Skype, just the same as you chat with your friends.



Hi, what can I do for you? 0:11

Can i have a device price lower than 10000hkd 0:11

Do you have preference on brand? 0:11

1 0:12

Please select a brand value 0:12

Figure 7: Bot Demo on Skype

9 Future Plans

We will adopt the main logic flow of the bot prototype we have implemented so far and further develop the bot according to the business requirements proposed by Microsoft. This section discusses the main tasks we are going to accomplish in the second semester.

9.1 Language Model Development

Currently we have developed the LUIS model for English. Nevertheless, the Chinese language models, especially traditional Chinese, are of greater importance for Microsoft. We will follow similar design of the English model to develop the language models for simplified Chinese and traditional Chinese. The design of LUIS intents and entities may change slightly to better meet the business requirements.

9.2 Data Collection

Data is quite critical in determining the performance and accuracy of our bot. However, the data we have so far is really limited, only the utterances generated and labelled by ourselves as well as the sample category from Microsoft online store containing 14 products, which is far from necessary. Thus, we will make efforts to collect more data and ensure the quality of data. On one hand, we will ask Microsoft side to provide us with some real-life data, for example, what questions that customer usually ask; on the other hand, if necessary, we will try to grab data from Microsoft's Facebook and Twitter pages using the web crawling techniques and at the same time generate more realistic data by ourselves. Finally, all the data will be manually labelled and converted into text format for training and testing.

9.3 Training and Testing

Majority of the collected data will be used to train the language models. After labelling the training data, the LUIS will do the training for us while the machine learning algorithms behind is unknown to us.

Testing contains two parts. First of all, we will split a portion of the collected data to test the language models and evaluate their accuracy in identifying intents and labelling entities. After the completion of the bot, we will test the overall performance of the bot by talking to the bot directly by ourselves and inviting students from different faculties to try the bot. The bot will be evaluated by the average degree of satisfaction (from 1 to 10) of all the participants.

9.4 User Experience Improvement

As the bot is potentially an industrial product of Microsoft aimed at serving its customers, Microsoft lays much emphasis on the bot's interaction with users. In order to improve user experience, instead of keep asking the user whether he or she has requirements on each unspecified feature as the bot prototype does for the time being, the bot needs to keep the conversation as concise as possible but at the same time it is still able to capture the most useful user demands. Additionally, we will make the bot add some emojis and images of products to make the conversation more interesting for the customer.

10 Difficulties and Limitations Encountered

This section discusses the two major difficulties we encountered and the possible solutions.

10.1 Design of LUIS Model

As mentioned in §6.2**,** due to some restrictions of LUIS itself as well as the complexity of human languages, we changed the design of the LUIS model from time to time. At the very beginning, we proposed to use multiple LUIS models for one language; each model only responsible for one layer of meaning and all these models form a hierarchy. However, using multiple models for one language greatly increases the complexity of implementation. Moreover, there is no guarantee that these models are trained to exactly the same standard and have the same performance. As suggested by Microsoft, to make things simpler at this stage, we now stick to using a single model for one language to capture the most important information in a sentence.

10.2 Lack of Business Input

Even though we have met up with business representatives for several times, what they provided to us are relatively high-level ideas instead of actual business data. Therefore, due to lack of business input, it is difficult for us to proceed and we don't know exactly the behavior of Microsoft's potential customers. Currently, the bot prototype just asks the customer's requirements on product features and search the database to make the recommendation. However, we are not sure whether customers will ask for recommendation in other ways. Microsoft has promised us to get us connected to its promoters for the exact flow of a conversation soon so that we will know what common questions that a customer tend to ask when making a purchase.

11 Conclusion

In conclusion, we are keeping pace with the proposed schedule in the project plan. We report our progress to our supervisor and the representative from Microsoft on a regular basis and receive timely feedbacks from them. The logic flow of bot prototype we have implemented so far will be the basis of our future work. With a good understanding of the project objectives and scope as well as the techniques we are going to use, we will further develop and refine the prototype to make it a real product that well fits Microsoft's requirements.

However, to be frank, our original schedule seems to be quite loose and we need to set higher goals for ourselves. In the second semester, we will devote more time to the project and try to make our bot as interactive and intelligent as possible. We believe our final product can really help Microsoft customers to make wise decisions and help Microsoft to handle complaints efficiently.

References

[1] 1-800-Flowers [Internet]. USA: 1800flowers.com Messenger Bot. [cited 2016 Dec 1] Available from: https://www.messenger.com/t/1800flowers/

[2] Apple Inc [Internet]. USA: Apple online chat support. [cited 2016 Dec 1]. Available from: https://aoschat.apple.com/Chat/getCustomerDetails.do

[3] Microsoft HK Ltd [Internet]. USA: Microsoft online chat support. [cited 2016 Dec 1]. Available from: https://www.microsoft.com/surface/en-hk/support/contact-us.

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

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

[6] Microsoft [Internet]. USA: Language Understanding Intelligent Service. [cited 2016 Oct 20]. Available from: https://www.microsoft.com/cognitive-services/en-us/language-understanding-intelligent-service-luis.aspx

[7] Microsoft [Internet]. USA: Microsoft Azure. [cited 2016 Oct 20]. Available from: https://azure.microsoft.com/en-us/

[8] USMAN_MALINK [Internet]. Udemy Blog. [update 2014 May 29, cited 2017 Jan 22]. Available from: https://blog.udemy.com/ado-net-entity-data-model/

[9] https://i.cs.hku.hk/~fyp16004/