the University of Hong Kong

Faculty of Engineering

Department of Computer Science

Final Year Project

Year 2019 - 2020

An intelligent assistant for HKU students

Authors

Ma King Wai,

Lau Ngai Fung,

Chu Chi Hang

Supervisor

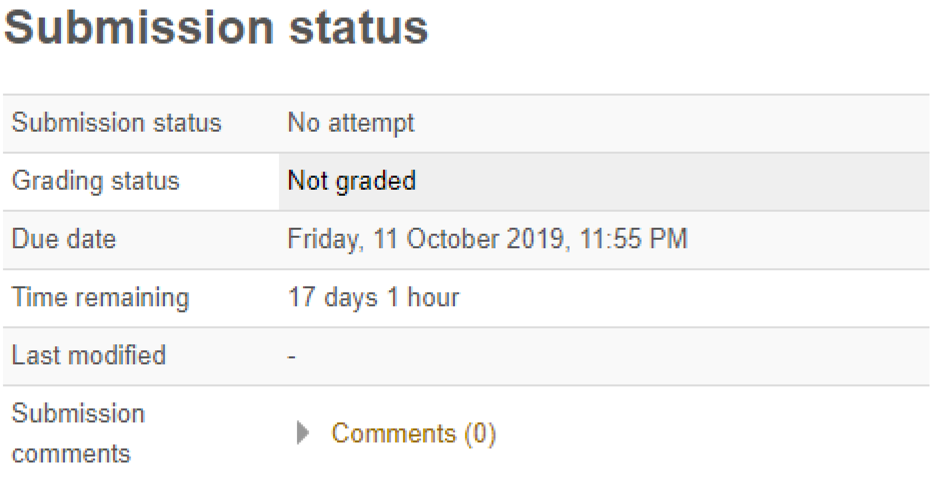
Dr. Chim

September 29, 2019

Contents

1 Motivation

HKU students use a number of Moodle and Portal platforms. However, user interfaces of these platforms are not well designed. For example, if someone wants to check the deadline of an assignment, he must log in to Moodle, enter his account name and password, find the course among 20 or more courses on the main page, go through half the course page to find the link of that assignment, and finally he can find the due date there. It usually takes 30 or more seconds to do that.



Submission status

Submission status	No attempt
Grading status	Not graded
Due date	Friday, 11 October 2019, 11:55 PM
Time remaining	17 days 1 hour
Last modified	-
Submission comments	▸ Comments (0)

Figure 1: Sample Submission Box

It is only a simple case. According to the data collected from our survey, it was found that Moodle is being used for downloading course materials and submitting assignments, and Portal is being used for course selection, checking the timetable and checking course grades in most cases. The time wasted in input and output is enormous, given that these tasks are being done repetitively during the 4-year university life.

Table 1: HKU moodle tasks

Rank	Task on HKU Moodle
1	Downloading course materials
2	Submitting assignments
3	Checking deadlines
4	Discussing on forums
5	Watching lesson recordings
6	Texting tutors and professors
7	Other

2 Objective

2.1 Functionalities

2.1.1 Cross Accounts / Websites

The Chatbot can search for information across all HKU related accounts. Student only require to login once and input once for searching HKU portal, HKU moodle and HKUL simultaneously. Huge amount of time can be saved.

2.1.2 Short Loading Time

The response of the Chatbot is faster than the HKU Portal in the same network condition. The program directly handles the HTTP request and parses the HTTP response, eventually answer the question raised by the user. Thus, users do not need to comprehend and extract the answer from a bunch of HTML pages, which is time-consuming.

2.1.3 Interactive response

The Chatbot can analyze user text query and reply automatically.

2.1.4 Responsive Design

Appearance of the Chatbot should be flexible. The Chatbot webpage renders well according to the screen size.

2.2 Metrics

2.2.1 Fast access

- Cross Account / websites
- Short Loading Time

2.2.2 User-friendly

- Interactive Response
- Responsive Design

2.2.3 High accuracy

- Speech Analysis
- Optimal Result

3 Methodology

3.1 Overview

The basic idea of the AI assistant is to go through the following process:

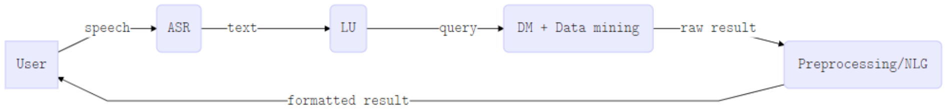


Figure 2: Workflow of the virtual assistant

1. Get user speech input

2. Automatic Speech Recognition: convert speech input to text input

3. Language Understanding: analyze the text input and add extra information about it, converting to query.

4. Dialogue Management + Data Mining: Extract raw result from website and determine the next reply.

5. Preprocess the raw result to give final reply

6. Output the findings

3.2 Microservice Architecture and Serverless Computing

Microservice architecture and serverless computing are the most advanced techniques in the industry. Microservice architecture breaks a large application into smaller, independent collaborating components. Unlike monolithic architecture, an independent component can be shut down without affecting or terminating the whole application. Modularity makes the applications highly maintainable and loosely coupled.

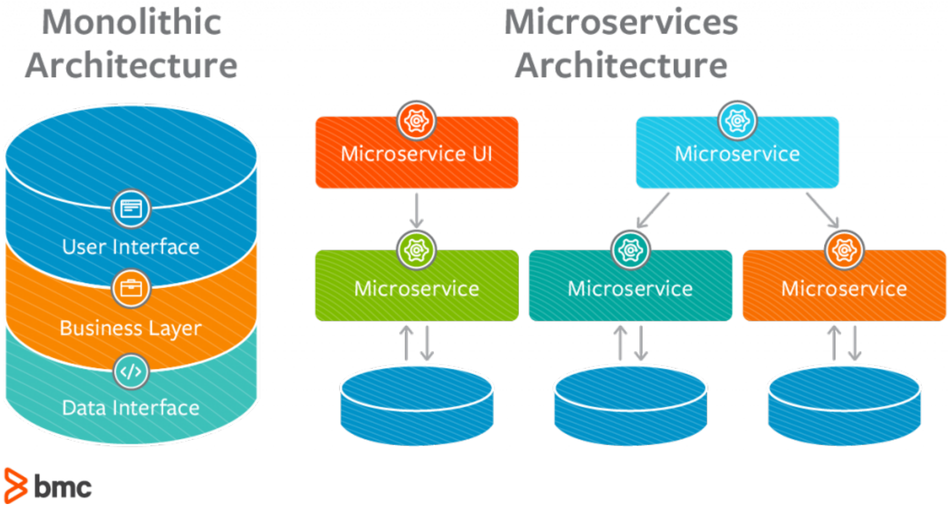
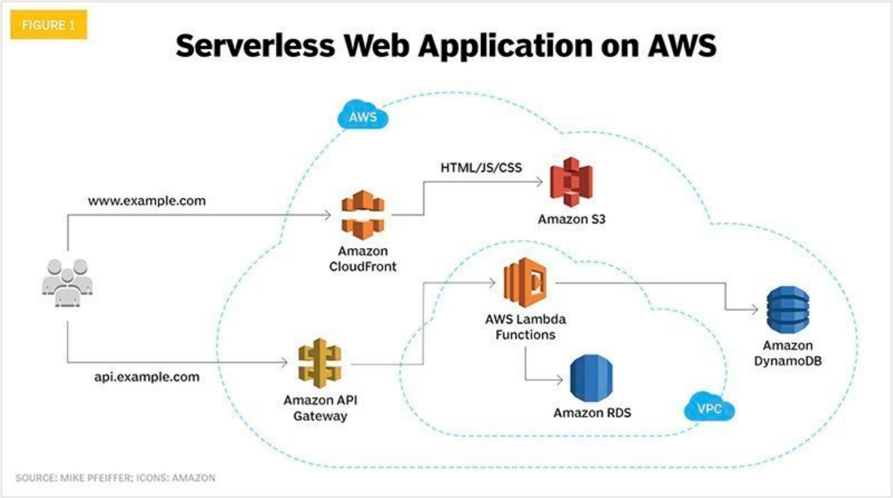


Figure 3: Comparison between architecture [1]

Serverless computing is a cloud server which can dynamically allocate hardware resources to applications on demand. In other words, if queries per second (QPS) is high, more hardware resources will be allocated to more scalable microservices which avoids suffering from performance issues. If queries per second (QPS) is low, less hardware resources will be allocated to save costs.

Since the microservices deploy and dynamically scale themselves on demand, it is cost-effective and performance-efficient to deploy a microservice application on serverless cloud server with advantages of high elasticity and scalability. After lying the concepts of microservice architecture and serverless computing out, here are the practical ways to achieve the goals.



SOURCE: MIKE PFEIFFER; ICONS: AMAZON

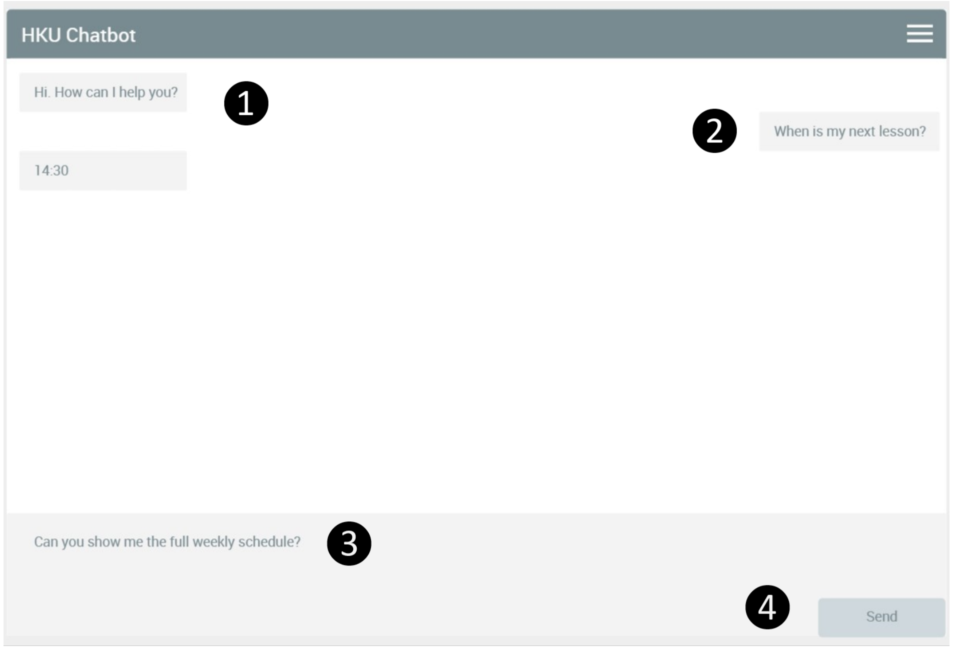
Figure 4: Structure of AWS Serverless Application [2]

Spring Cloud, which is a popular microservice framework based in Java, will be adopted as the backbone of the application. It will establish a link among the frontend, backend and the chatbot service written in Python, which is a language particularly suitable for machine learning and data analysis. To combine all the advantages of both languages, Sidecar, a tool provided by Spring Cloud, can integrate interfaces of third-party applications with the Spring Cloud application, such that those third-party applications can share modules like routing service, load balance service which are prepared by Spring Cloud application. With a microservice architecture, micro-applications written in different languages can communicate with each other easily.

A server is needed to hold the intelligent assistant, and Amazon Web Service (AWS) seems to be a great choice in terms of its high service integrity. Amazon Lambda, which is a platform providing serverless computing service, supports most major languages like Python and Java. It suits the need of combining Spring Cloud application and AI model trained in Python. Apart from this, there are many online resources like tutorial videos and articles for the team to learn how to deploy the intelligent assistant onto the server. They can give us inspirations for how to make the application better.

3.3 User Interface Design

There are a number of pages in the virtual assistant website, such as login page, account settings and chat page. In this paper, we introduce the design chat page. Below is a simple UI prototype:



Hi. How can I help you?

When is my next lesson?

14:30

Can you show me the full weekly schedule?

Figure 5: User Interface of the Chatbot

① The Chatbot's message ② User's message ③ User text input ④ Send button

3.4 Data Collection

The data required in this project is a set of text queries and expected results. This is difficult to obtain since there are limited resources on the internet for website-specific chatbot training data. Therefore, we proposed to build a prototype for testing and training before collecting test data and make adjustments afterward.

3.4.1 Sample user

In order to have a diversity of sample users, we collected data from persons with different characteristics as below:

- Gender
- Age
- Major
- Ascent

These characteristics are used to analyze and train the virtual assistant in this project, such as inspecting error and area of frequent query.

3.4.2 Text request and expected result

The set of text queries and expected results are designed in pairs. Below is the detailed design of one question-answer pair refer to Table 2:

Table 2: Sample Query Structure

Query set feature	Possible options	Description
Input Type	keywords / sentence	the type of text input of the query, optional
Raw Input	–	the raw text input of the query
Domain	HKU Moodle / HKU Portal / HKU CS account / Search Engine	the website of searching, optional
Expected Result Type	Text / Image / List	the expected type of result
Expected Result Raw	–	the raw expected result

Some of the above input features are 'optional'. It means these input features are for training purposes. They will not be provided in the final product.

3.4.3 Other Training Dataset

Apart from the self-proposed training dataset, NLTK, a famous Natural Language Processing library provides a wide range of language datasets. Among these datasets, we choose the chat package as the base dataset. It contains more than 1,000 chatting example with tag of word and sentence structure, helping the virtual assistant to understand the meaning of user input.

3.5 Natural Language Processing

In general, virtual assistant is a chatbot that can handle user requests without predefined options (such as button) and carry out the corresponding action. A definition from Technopedia is cited:

> *A chatbot is an artificial intelligence (AI) program that simulates interactive human conversation by using key pre-calculated user phrases and auditory or text-based signals.* [3]

Chatbot is widely used in customer service. Suit for frequent trivial inquiry, Chatbot has a much lower cost than employing a human. Therefore, we choose Chatbot instead of a catalog or a list of functions to make searching more user-friendly.

3.5.1 Overview

Chatbot is a composition of various components. By using Natural Language Processing (NLP) technique, Chatbot can have response to different type of request. In general, there are two levels of operation.

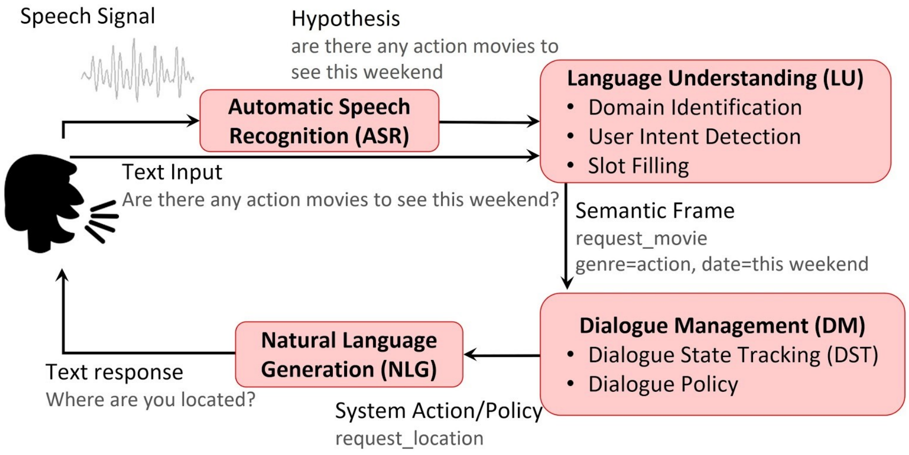

Speech Signal
Hypothesis
are there any action movies to see this weekend
Automatic Speech Recognition (ASR)
Language Understanding (LU)
• Domain Identification
• User Intent Detection
• Slot Filling
Text Input
Are there any action movies to see this weekend?
Semantic Frame
request_movie
genre=action, date=this weekend
Dialogue Management (DM)
• Dialogue State Tracking (DST)
• Dialogue Policy
Natural Language Generation (NLG)
Text response
Where are you located?
System Action/Policy
request_location

Figure 6: Workflow of chatbot [4]

As illustrated in Figure 6, Automatic Speech Recognition (ASR) is responsible for the conversion between speech and text; Language Understanding Unit (LU) interpret the input by adding more information to the text; Dialogue Management (DM) is determining the answer direction, and finally the Natural Language Generation (NLG) formats the response in human language.

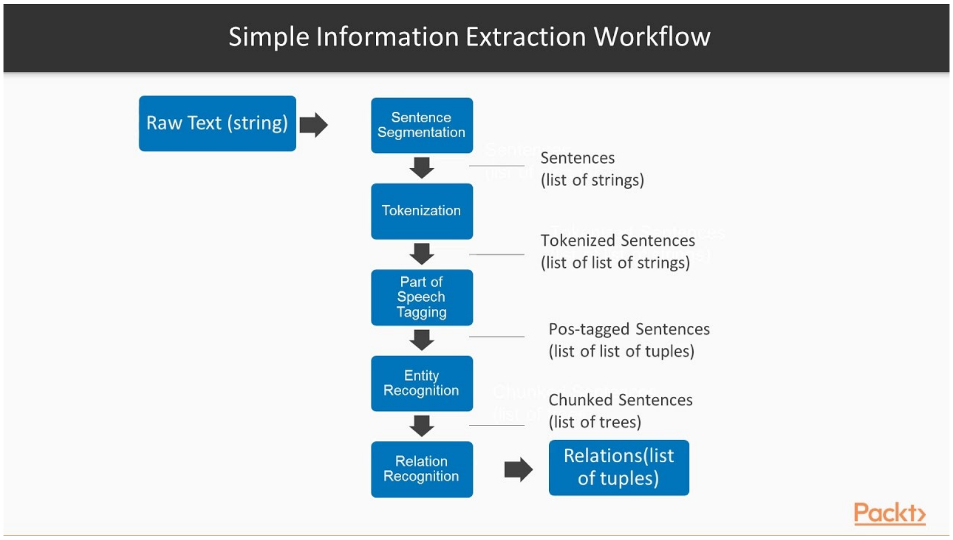
In the chatbot structure, LU and NLG are using the technique in Natural Language Processing. Natural Language Processing is an application of machine learning in text analysis. Since most individuals do not communicate in programming language, NLP takes the role of conversion. The coming below explains the detail of each component.

3.5.2 Automatic Speech Recognition

Most people agree that speaking is faster than typing in presenting ideas. However, turning voice commands into text for computer to process is complicated. It is not cost-effective to develop our own Speech to Text function, coupled with that this is not even the main objective of the project. Therefore, we planned to adopt Google Cloud Speech API to finish this job, given that Google provides highly accurate Speech to Text service and some special features like noise cancellation for users using intelligent assistant in public places. In addition, the price of Google Cloud Speech API is highly transparent. It allows the cost to be manageable and predictable.

3.5.3 Language Understanding

For this project, the primary goal of NLP is to extraction meaning and identify the meaning of the text input. This is the task refer to the Language Understand Unit in Chatbot structure (see Chatbot in Literal Review). We use a NLP pipeline to work with the text input, and then pass the query to the Data Mining Module (see Data Mining in Methodology).



Simple Information Extraction Workflow

Raw Text (string)

Sentence Segmentation

Sentences (list of strings)

Tokenization

Tokenized Sentences (list of list of strings)

Part of Speech Tagging

Pos-tagged Sentences (list of list of tuples)

Entity Recognition

Chunked Sentences (list of trees)

Relation Recognition

Relations(list of tuples)

Figure 7: NLP Pipeline [5]

Figure 7 shows how the NLP pipeline works. It first breaks text into sentences, and then further break down into individual words. After recognizing the part of speech of the word, it reconstruct the sentence into a tree structure, and finally extract the relationship between words. After these processes, the input has more information for machine to understand the meaning of sentence. In fact, the process is like how you read a sentence: find the full stop and comma, look at the verb and noun one-by-one, and then get the relationship between words.

3.5.4 Natural Language Generation

After receiving the raw result from Data Mining (refer to Figure 2) , the Natural Language Generation unit should decide how to format the final output. With the understanding of the query in above part, the chatbot should be able to determine the format. For example, if the query include the word (image, noun) follow after (What, adverb), then it is likely to have an image output; if the query does not state clearly, such as "timetable of this week's lecture", then it may refer to image or table format. This can be trained by sample user input (see Data Collection in Methodology). Simple Machine Learning algorithm such as Logistic Regression is expected to solve the problem.

3.5.5 Technical details

Python is suggested for this project due to its extensive open source libraries in the field of NLP and simple coding style. There are a famous libraries such as NLTK and Textblob for NLP. In this project, we choose NLTK as the starting library.

While training the Language Understanding Unit, we will use NLTK dataset and sample dataset from HKU students (see Data Collection in Methodology), and borrow predefined NTLK identifier for elementary tasks such as tag-of-speech, sentence analysis and lemmatization. After the language understanding unit is mature, we will switch to TextBlob, an industry-standard for NLP, having faster access and simpler coding than NLTK.

For Dialogue Management in chatbot, we first proposed to build an intent-based chatbot, and then progressively add more flow-based content to it. During this process, several adjustments such as training dataset and workload will be tested. In the end, we expect that chatbot can handle most of the natural human requests in a reasonable scope.

3.6 Data Mining

3.6.1 Definitions

Before introducing the draft of data mining module, some terms are defined as follow:

- query: a piece of analyzed text with part of speech, sentence structure and meaning after processed by NLP unit
- action: a sequence of procedures to find the result

3.6.2 Overview

Below is a workflow for data mining:

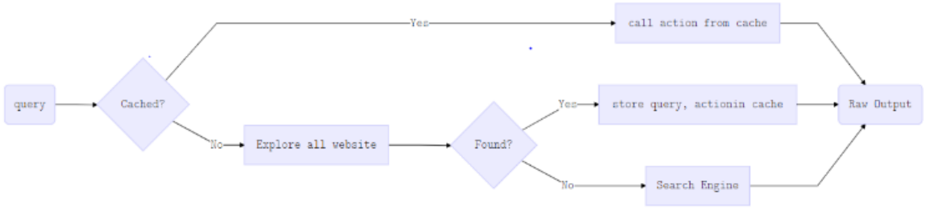


Figure 8: Caching Mechanism of data mining module

The data mining module first recognize the domain of searching. This can bound the search time under a upper limit and speed up searching. However, it is difficult for computer to recognize the domain of search. Even for some student, they spent time on searching for suitable information in the portal. Facing this situation, we propose a caching mechanism to tackle the above difficulties.

As shown in Figure 8, the first check is whether the query is cached. If similar query are repeated, then the virtual assistant will not explore the website; instead, it follows the past query and extract the result. If the query is not encountered before, the virtual assistant then searches on all websites. The ordering of search is determined by past record: if some websites appears more in some past queries having the same keywords, they are considered as highly possible target website. This can optimize the first priorities. In case there is no relevant result from all HKU accounts, the virtual assistant will turn to the search engine. It will inspect top results to find possible matches to the query.

In order to prevent infinity search, a timer is enforced to the virtual assistant if it expired the timer limit. In this scenario, the virtual assistant will reply 'no positive result'. On the other hand, if any of the above new query success, the virtual assistant will further merge this piece of result into database for future use, making it more powerful. Before outputting to the user, a preprocessing is necessary to narrow the range and predict the format of result, which is mentioned in Natural Language Processing in Methodology.

3.6.3 Technical details

Python is a good candidate for web scraping and it will be adopted for the web scraping part of the project. Python supports HTML parsing by its default library. Also, there are several well-known web scraping framework based on Python. For instance, Scrapy and Beautiful Soup can help us retrieve suitable data from HKU moodle and portal to generate an appropriate answer to users. For example, if the user asked for the time and venus of the next lesson. The program will generate an HTTP GET request about the “My Weekly Schedule” to the HKU portal server, and then the server will respond with HTML which contains the timetable. The timetable HTML can be parse by python and ordered it according to time. We then can compare the timetable with the current time to check out what is the next lesson. And now the bot can answer the user easily

3.7 Database Design

We choose MongoDB for the database management system. It has two advantages: (i) highly compatible with RESTful API, and (ii) flexible storage structure for further modification.

The database consists of two parts: query, keywords-query pairs and account details. First, the keywords - query pairs stores the keywords that match the query. Notice that the query and individual keyword may repeat. Second, the account details part stores the information about the user. Primary information such as user ID, account usernames and passwords are stored. In the future, it may extend to include user behavior and preferences.

4 Schedule

Table 3: Proposed Schedule

Due Date	Task
Late October	Finish training Chatbot module • Accurate part-of-speech tagging above 0.95 • Accurate sentence structure analysis above 0.85
Late November	Collect Sample input and answer • Sample size above 5 persons • Sample input and answer greater than 100 pairs
Late December	Finish data extraction module • Enable search in HKU Moodle • Enable search in HKU Portal • Enable search in HKU Library
Late January	Integration (I): integration chatbot module and data extraction module on pure python script
Late February	Integration (II): move chatbot module and data extraction on website
Late March	Finalize design

5 Future Work

5.1 Smart User Behavior Prediction

Through recording the user activity, the virtual assistant analyzes the pattern and predict his behaviour. For instance, a student open a course page on every Monday 0930. This suggests the student is having a weekly lecture on Monday 0930. To help the student, the virtual assistant may suggest opening that course page for the user at every Monday 0925. However, user pattern may change randomly given irregular schedule, and multiple experiments are required to adjust the remainder function.

5.2 User Feedback

Apart from developers or administrator adjusting the virtual assistant settings, the virtual assistant may learn itself by absorbing user feedback. The virtual assistant randomly draws some requests or chats and ask whether the user how it is performing and which criteria is concerned. By surveying the performance, the virtual assistant can adjust itself. However, it is hard to determine the validity of user feedback. Different users has different preference, and it is difficult for the virtual assistant to apply its learning to all the users.

References

[1] Watts, S., & Shiff, L. (2018, October 9). An Overview of Monolithic vs Microservices Architecture (MSA). Retrieved from https://www.bmc.com/blogs/microservices-architecture/.

[2] Knuth: Pfeiffer, M. (2017) Serverless computing architecture, microservices boost cloud outlook.Techtarget Network. Retrieve from https://searchaws.techtarget.com/feature/\Serverless-computing-architecture-microservices-boost-cloud-outlook

[3] Technopedia *Chatbot* Retrieve from https://www.techopedia.com/definition/16366/chatterbot

[4] Mu (2018, Nov 23) [NLP] Intent-based and flow-based) Chatbot. Retrieved from https://medium.com/botbonnie/nlp-%E9%97%9C%E6%96%BC%E6%84%8F%E5%9C%96%E5%BC%8F-intent-based-%E8%B7%9F%E6%B5%81%E7%A8%8B%E5%BC%8F-flow-based-chatbot-1555fbfc322c

[5] Ghoash, S. and Gunning, D. (2019, March 30) *Natural Language Processing Fundamentals*