

**School of Computing**

FACULTY OF ENGINEERING



**UNIVERSITY OF LEEDS**

---

**Intelligent Personal Assistants for School of Computing: A  
Chatbot for Student Service and Learning Assistance**

**Zihan Zhu**

**Submitted in accordance with the requirements for the degree of  
MSc Advanced Computer Science (Data Analytics)**

**Session 2020/2021**

The candidate confirms that the following have been submitted:

<b>Items</b>	<b>Format</b>	<b>Recipient(s) and Date</b>
<i>Deliverables 1</i>	<i>Report</i>	<i>SSO (09/12/21)</i>
<i>Deliverables 2</i>	<a href="https://github.com/RomaMarcus/ZihanZhu_MScproject_Q-A-Dataset">https://github.com/RomaMarcus/ZihanZhu_MScproject_Q-A-Dataset</a>	<i>Eric Atwell, Abdulrahman Altahhan (09/12/21)</i>

Type of Project: Empirical Investigation

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student): ZIHAN ZHU

## **Summary**

In the realm of artificial intelligence, chatbots are a hot topic. Chatbots are used in a variety of areas, including medical systems and corporate business services. The creation of a chatbot centred on student service and learning support for British education institutions is a viable research topic. This primarily entails providing students with pertinent university information or learning resources based on their requirements. This project intends to construct a chatbot assistant since the School of Computing at the University of Leeds cannot afford a personal assistant for each student and because the workload of tutors and administrators may delay the results of feedback to students. It is part of School of Computing. It informs students about the college and replies immediately to any inquiries they may have regarding the subject. At the same time, it can assist students with simple tasks such as library reservations.

This project is built using the Dialogflow platform from Google. The major goal is to integrate various forms of student consulting services into a logical framework for chatbot functioning. Different sorts of consultation are grouped into sub-frames. These sub-frames include: 1) simple school service processing (not main); 2) School of Computer Science service guidance (main); and import the question and answer set of computer course-related data into the knowledge base of this chatbot, as well as use different question and answer types from different students as training sentences to increase chatbots' learning abilities and, lastly, to create a service platform that can engage with students. In addition, the project was tested and assessed with the help of some students from School of Computing to assure the usefulness of the chatbot.

## **Acknowledgements**

First and foremost, I would like to express my gratitude to Dr Eric Atwell, my mentor. He was genuinely worried about my MSc Project and gave invaluable assistance and support throughout the last four months, allowing me to complete the project successfully.

Second, I would like to express my gratitude to my assessor Dr Abdulrahman Altahhan, who gave me extremely helpful advice on how to enhance my project and optimise the method.

Second, I would like to express my gratitude to Dr Mohammad Ammar Alsalka, my project management supervisor, who provided me with invaluable support and assistance during the project.

Finally, I want to express my gratitude to my family and friends for their support and encouragement throughout the writing of this report.

## Table of Contents

<b>Summary.....</b>	<b>iii</b>
<b>Acknowledgements.....</b>	<b>iv</b>
<b>Table of Contents .....</b>	<b>v</b>
<b>Chapter 1 Introduction.....</b>	<b>1</b>
1.1 Project Aim.....	1
1.2 Objectives .....	1
1.3 Deliverables .....	2
1.4 Ethical, Legal, and Social Issues.....	2
1.4.1 Ethical Issue.....	2
1.4.2 Legal Issue.....	2
1.4.3 Social Issue.....	2
<b>Chapter 2 Background Research.....</b>	<b>3</b>
2.1 Literature Survey .....	3
2.2 Methods and Techniques .....	6
2.3 Choice of Methods .....	10
<b>Chapter 3 Experiential Design .....</b>	<b>12</b>
3.1 Data Analysis .....	12
3.2 Data Collection .....	12
3.2.1 Google Scholar System .....	12
3.2.2 Wikipedia .....	12
3.3 Experiential Tools.....	13
3.3.1 NLU.....	14
3.3.2 User Input .....	15
3.3.3 Fulfillment.....	15
3.3.4 Dialogflow Dashboard.....	16
3.4 Chatbot Structure Framework .....	17
<b>Chapter 4 Experiential Implementation .....</b>	<b>19</b>
4.1 Dataset.....	19
4.1.1 Data Pre-Processing.....	19
4.1.2 Data Display.....	19
4.2 Implementation of Features.....	20
4.2.1 Library Reservation .....	23

4.2.2 QuestionAnswer.....	28
4.2.3 Integrations .....	31
<b>Chapter 5 Evaluation.....</b>	<b>34</b>
5.1 Usefulness Evaluation.....	34
5.1.1 Average Detection .....	34
5.1.2 F-Score .....	37
5.2 Satisfaction Survey.....	39
<b>Chapter 6 Conclusion and Future Work.....</b>	<b>41</b>
6.1 Conclusions.....	41
6.1.1 Challenges .....	41
6.1.2 Limitations.....	42
6.2 Future Work .....	42
<b>List of References .....</b>	<b>44</b>
<b>Appendix A External Materials.....</b>	<b>46</b>
<b>Appendix B Ethical Issues Addressed .....</b>	<b>47</b>
<b>Appendix C Questions and Answers Dataset.....</b>	<b>48</b>
<b>Appendix D Chatbot Function Display .....</b>	<b>50</b>

## **Chapter 1**

### **Introduction**

#### **1.1 Project Aim**

The main aim of this project is to employ artificial intelligence-based chatbot technology to create a virtual student assistant for the School of Computing at the University of Leeds, with the goal of facilitating communication between students and the college. The main goal of this project is to create a chatbot using Google's DialogFlow platform.

Higher education has progressed significantly in the last two decades due to advancements in communication technology and intelligence(Hien et al., 2018). The traditional educational model has gradually transitioned to intelligent and online education, which is inextricably linked to artificial intelligence support. The introduction of these new educational models has given the educated more convenience. Chatbot technology, as a study branch in the subject of artificial intelligence, allows for a variety of learning modes(Chang et al., 2022). Face-to-Face education is experiencing some challenges, such as when natural diseases such as Covid-19 are tough for humans to resist, and students need to know the most up-to-date information from schools and universities remotely. Furthermore, colleges and institutions in the United Kingdom cannot afford to optimise personal learning assistants for each student. Educators and college administrators, on the other hand, have a lot of work to do. Therefore they are unable to respond to every student's inquiry request in a timely manner. Hiring more employees will increase college and university costs, and most colleges and most colleges and universities lack a virtual student assistant. Therefore, attempting to design a chat robot that can answer students' queries in a timely manner and assist students in learning is beneficial in addressing the aforementioned issues.

The platform used for this project is Google's Dialogflow, a natural language understanding platform that allows for easy, non-programmatic development of conversational assistants that can be deployed across platforms.

#### **1.2 Objectives**

- Research on basic conceptual knowledge of chatbot, machine learning, NLP.
- Gather basic information on mainstream chatbot development platforms: Amazon, Google, Azure, Alice, Pandorabot, understand their charging models and the logic of their development methods, and select the appropriate platform for this project.
- Get familiar with the logic of Dialogflow.

- Get simple information about the School of Computing at the University of Leeds, ask students about their interest in computing, and collect a dataset of quiz questions related to this topic.
- Plan and design the conversation logic of the chatbot and import the dataset to select the appropriate training statements to optimise the chatbot.
- Research on how to deploy chatbot to Facebook and check related operations.
- Establish a comparator and find another successfully developed chatbot for comparative evaluation of response accuracy, usability.
- Invite at least three computing students to use the chatbot and give an evaluative opinion.

### **1.3 Deliverables**

- A chatbot that can achieve project goals.
- Dataset for this project.
- MSc Project Report.

### **1.4 Ethical, Legal, and Social Issues**

#### **1.4.1 Ethical Issue**

Since this chatbot is developed on an external platform, it may cause the information of students who use this chat system to be leaked. It is also possible to disclose student information in the process of seeking student evaluation or questionnaire survey.

#### **1.4.2 Legal Issue**

Since this chat system is object-oriented to students, and its function is based on campus life and learning guidance, this chatbot will not engage in illegal chats, nor will it use the chat system to carry out some illegal activities.

#### **1.4.3 Social Issue**

If there is a mistake, the chatbot may post some misleading information, such as the wrong school notice, which may mislead students in obtaining information and affect their decision-making. If there are some unknowable errors, it may be used by outsiders to release inducing information about the society.



## **Chapter 2**

### **Background Research**

#### **2.1 Literature Survey**

Research into chatbots can date back to the 1950s when Alan Turing envisaged the possibility of making the machines we use every day think like humans(Agarwal and Wadhwa, 2020). This idea can be seen as a prototype for the research and development of chatbots in the field of artificial intelligence. The idea of "making human-machine conversations possible" has been explored by researchers in the field of artificial intelligence, dating back to the 1970s(Lee et al., 2020). Later, as digital and computer technologies matured and grew, MIT developed a model based on human-computer dialogue(Weizenbaum, 1966), which can be considered the earliest model of the chatbot, but this programming system was developed manually by humans, requiring constant re-writing of logic and incorporation of dialogue cases, and the environment in which it could be used was very limited. Later, natural language-based human-computer dialogue programs became common(Valdivieso and Luzón, 2021). However, due to the limitations of the technology, the fault tolerance of the models used in the early construction of chatbots and their inherent lack of usability, the programs developed were limited and not well used. In the last decade, models in the computer field have been fully constructed and improved, and the availability of computer models that can handle complex vocabulary and complex transactions has provided a better platform for these programs, such as machine learning and the emergence of deep learning, which has made a large amount of available session data easier to process and analyse(Zhou et al., 2020), thus bringing convenience and satisfaction to the user. As a result, a new definition of chatbot technology has been developed. Some researchers stated that the chatbot industry is still in a developmental stage, but the growth in numbers and technology is very rapid, with many platforms enabling chatbots to be built and rapidly marketed at a low cost, leading to the development of better chatbot frameworks (Rahman et al., 2017). As the development mechanism continues to mature, the scope of use of chatbots has become more extensive(Ranoliya et al., 2017). They can be used in business, games, and school education(Abu Shawar and Atwell, 2007). This project mainly studies the development of chat robots in the direction of learning and education.

Chatbot, a product of the artificial intelligence field, still have the fundamental aim of serving humans with smarter ways to handle more tedious matters and increase efficiency. In a survey, a common phenomenon that exists today under the traditional education model is the increase in student dropout rates(Weizenbaum, 1966). The traditional model of teaching

is simply a question-and-answer session between the teacher and the students, but in this day and age, with the internet system in place, this single model of teaching does not focus well on the students, so new models of education need to be used to reduce the dropout rate(Valdivieso and Luzón, 2021), and also to reduce the workload of teachers to a certain extent. In addition, chatbots can have a positive impact on students. A research evaluated the functions of Hubert chatbot and found that chatbot can promote students' enthusiasm for learning(Abbas et al., 2021). In this sense, chatbot technology can be a powerful tool for higher education. Examples of the application of chatbot technology to the education system actually already exist, and there are already researchers working on programs in several schools. The following are the student-oriented chatbot projects mentioned above:

- FIT- EBot

Researchers at The Faculty of Information Technology of the Ho Chi Minh City University of Science found that their early deployments of the education system were struggling to handle student transactions, with students struggling to get the information they needed when conducting and retrieving information from the website. This has led to the development of an intelligent chatbot called FIT- EBot. The chatbot uses a development platform to generate an interface API to the university's internal academic database system, set up basic chat logic to migrate student transactions from the previous student system to the FIT-EBot program, and finally deploy the chatbot to a communication application, where students can ask questions about their questions such as scholarship registration and examination information through a dialogue(Hien et al., 2018). After evaluating the application, it was found that when students became familiar with the chatbot and used it, the efficiency of the student affairs process was greatly improved(Hien et al., 2018).

- Chatbot-facilitated Nursing Education

Students at Taipei Medical University have developed a chatbot deployed on smartphones to improve the quality of teaching in nursing medicine courses(Chang et al., 2022). The chatbot is based on a knowledge-based system, which provides a powerful database of medical expertise. The chatbot sets up a framework of three learning modes: anatomical knowledge, body anatomy and intelligent anatomical diagnosis. Students can interact with the chatbot through the interface provided, and the program will give a judgement based on the assessment of the anatomical diagnosis presented by the students, which will greatly stimulate their critical thinking. They found this chatbot can promote student interest in learning and improve student performance(Chang et al., 2022).

- AISS chatbot

Also applied to medicine, students at The Chinese University of Hong Kong found that in modern medical education, due to the limited number of teachers, students are not able to

fully engage in detailed problem solving with their tutors, and what is most needed in medical education is an in-depth conversation between the educator and the student on a particular issue(Li et al., 2021). So they used BERT as a framework, which is an open-source framework for machine learning in the field of natural language processing, using the UK Anatomical Society Anatomy Syllabus for Medical Graduates as a database. The highlight of this program is that the chatbot can actively interact with the students and will actively ask them questions as well as answer their questions. By putting it into use, the grades of students have improved, proving that intelligent educational learning systems can provide meaningful guidance to students in their academic lives.

- EDUKIDZZ

Research scholars at Satiabamar Institute of Science and Technology have developed a chatbot for young children to learn. It is based on the Google platform and allows users to make a login page for learning and eventually integrate the chatbot into a web page(Sophia and Jacob, 2021). The intention of opening this bot is mainly due to COVID-19, where students are not allowed to learn in the school. This chatbot can be deployed to any teaching platform, which facilitates online teaching.

- During COVID-19, various schools had to implement online education, which could lead to students not having immediate access to teacher support. Students and professors of Francis Xavier Engineering College developed a chat system based on this situation. This chatbot solves this problem to a certain extent (Kasthuri and Balaji, 2021).

- Semi-Automated Chatbot Model

Students at Çankırı Karatekin University developed a semi-automated bot that responds to students' questions via email in order to improve the accuracy of the answers of the chatbot(Buyrukoğlu and Yılmaz, 2021). The accuracy and authority of the answered questions are ensured by evaluating the comparisons, which ensures that each student receives the same quality of answers.

Many well-developed robots for the education industry exist today that can help students. While the examples listed above have excellent features, after researching these projects, I found that they lack universality and life service. As a tool to serve students, the chatbot should not only have the ability to help students in their learning but should also be configured to address affairs of student life on campus. For example, some students want to take an easier and more convenient way to reserve a place in the library, and if chatbots could help them achieve this need, it would attract more people to use them. In addition, many students do not want to spend time searching for knowledge online, and this project could attempt to enter some of the topics into the chatbot so that students can simply open it and learn. The starting point of this project is to develop a student-friendly chatbot that

balances student learning with campus life while providing students with learning resources and assistance with campus affairs so as to truly help students deal with their problems.

Regarding the definition of chatbots, Some researchers defined them as software that uses natural language to engage in conversational services with users (AbuShawar and Atwell, 2015). Similarly, Some researchers states that a chatbot is essentially a program that allows a machine to chat in accordance with human thinking and to successfully chat with humans in a normal way, which can be used as a customer service system for businesses and customers (Shawar and Atwell, n.d.), as a marketing guide for shopping websites and in the most widely known area of real-time communication (Valdivieso and Luzón, 2021). Both of these definitions state that the nature of chatbots is that they need to be able to chat indistinguishably with humans under certain computer rule, and that one of the metrics used to assess the usability of chatbots is to test whether they can converse properly with humans. This leads to a research direction: the modelling mechanism of chatbots. (Agarwal and Wadhwa, 2020) argued that the task of chatbots is to deconstruct the natural language entered by the user to form the most logical response to the conversation (Agarwal and Wadhwa, 2020). Achieving machine-human dialogue, a set of rules needs to be developed and created using existing technologies, and these mechanisms are used to enable chatbot to respond to the user's messages. Over a long period of development and research, relatively mature methods and mechanisms have been developed in this area, and even today, these techniques are still being optimised and evolved.

## **2.2 Methods and Techniques**

As chatbot technology has evolved, different design approaches and mechanisms have emerged based on the habits and technical preferences of developers. Developers should choose a specific development platform and development logic based on specific needs. Developing a working chatbot application, it is necessary to implement the functionality of the application, such as the ability to engage in reasonable dialogue with users. The mechanisms for implementing chatbot functionality can be divided into two types based on different strategies: pre-defined rule-based approaches and machine-learning-based approaches (Agarwal and Wadhwa, 2020), as shown in Figure 2.1:

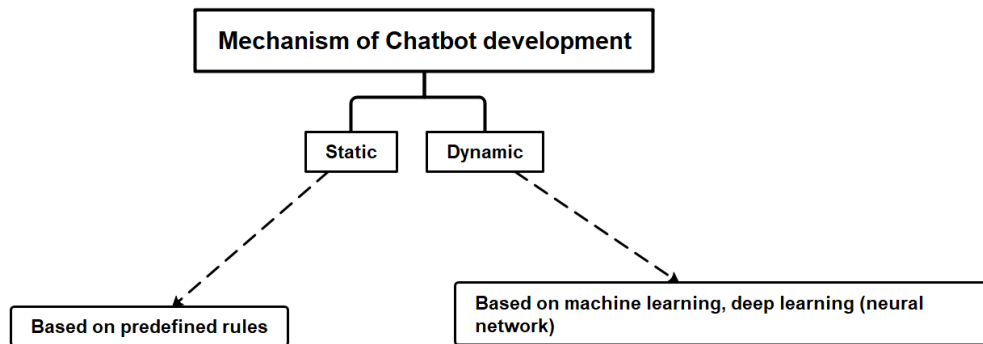


Figure 2.1: Two mechanisms for developing a chatbot

A pre-defined rules-based approach is a static approach to the development process that embeds a set of rules or a logical process that is pre-defined by the developer into the system of the chatbot, which could be a coded language, a pre-defined database, and when users ask a question to the chatbot, the program generates an answer to the user based on these rules which are pre-inputting question, answer the logic question and answer data. What the user asks is answered according to the answers that already exist. The machine learning-based approach is a dynamic approach that involves training a large dataset, and by continuously inputting large amounts of data, the chatbot learns on its own to generate responses as it answers the user, and the program gives the correct answer to the user's input, which is a form of autonomous learning that can be used over time to make the system smarter. The program will match patterns to different user questions to match the appropriate intent, verify the data collected, and then operate at the back end to select the most logical answer to generate a response. Valdivieso and Luzón (2021) also mentions similar development mechanisms that are generally used when performing chatbot development, one is to respond to the user through heuristic learning based on pre-coding, and the other is to talk to the user through a continuously generated machine learning-based approach trained through data(Valdivieso and Luzón, 2021). Both development mechanisms respond to the user through accurate pre-defined logic or leave it entirely to the chatbot itself to learn to generate responses. For example, developers can use data extracted through natural language processing of Dialogflow to generate dynamic responses, validate the data collected or trigger actions on the back-end. This study uses dynamic responses. Before developing a chatbot to meet needs, we need to establish which development mechanism to use so that we can establish the platform to be used next.

Developing a chatbot successfully, it is necessary to establish the development mechanism and the needs that the bot is developed to meet. Depending on the type of chatbot that needs to be developed, it can be categorised into goal-oriented and conversation-oriented chatbots(Rahman et al., 2017). Conversation-oriented chatbots, as the name suggests, only need to complete a simple conversation with the user. Figure 2.2 shows a simple dialog process of conversation-oriented chatbot:

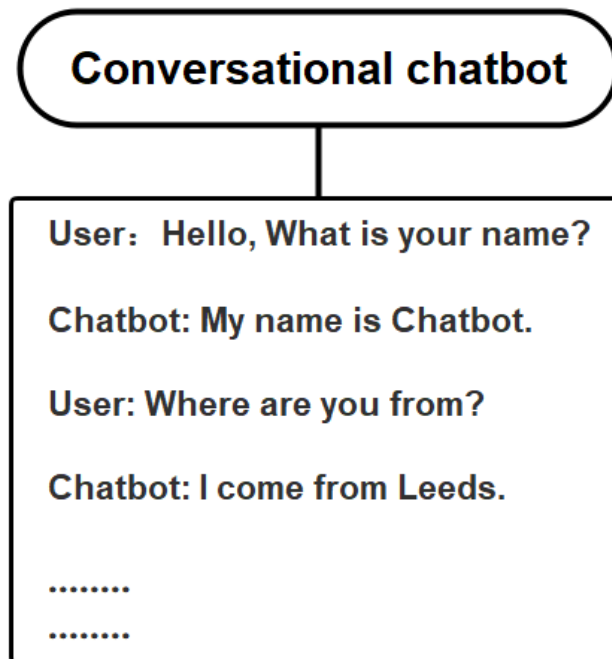


Figure 2.2: A conversation process of conversation-oriented chatbot

Chatbots only need to complete the user's questions and do not need to focus on the context of the conversation. This model of conversation is called non-linear conversation. This type of chatbot can simply give answers to the user's unconnected questions and are not capable of handling complex transactions.

A goal-oriented chatbot, which can be defined as a non-linear conversational chatbot. It can complete more complex conversations. This type of conversation is more in line with the logic of people's conversations. When users ask a question, the chatbot needs to make a contextual judgement and base its response on the information mentioned by the user. Figure 2.3 shows a simple example of a goal-oriented conversational chatbot. The system needs to extract important information from a non-linear conversation and generate an answer back to the user.



Figure 2.3: A conversation process of goal-oriented conversational chatbot

There are a variety of tools used to develop a chatbot. We can choose to build chatbots programmatically, using mainstream languages such as Python and Java to build a conversation engine. It is also possible to use platform tools to develop programs without programming. The choice of platform depends on the chatbot to be developed. Currently, there are three main types of chatbot development platforms available: platforms developed by leading technology companies in the industry, non-programmable chatbots and chatbots for conversational development.

Platforms developed by leading technology companies in the industry: Google's API ai Dialogflow platform is a natural language dialogue-based human-computer interaction technology development tool. The platform allows users to deploy conversational logic within a dashboard and train conversational data based on their own machine learning algorithms, allowing even non-experts to get started quickly. Dialogflow allows users to integrate a chatbot into popular applications such as Facebook, line and their own websites. Dialogflow does not disclose the source code. Amazon's Lex enables users to have conversations in voice or text form and can be easily deployed on all major platforms. When developing with Lex, users can make use of AWS cloud environment to call some handy functions such as the Lamanda function. The source code of Lex is fully public. Microsoft's Azure bot is created based on the Bot Framework SDK. When development is carried out, the interface is also developed to be simple and easy to understand, which is developer-friendly. This chatbot can be deployed on Microsoft applications such as Teams. Azure bot has a clear

dialogue logic framework when developed called visual designer, which allows the developer to clearly grasp the flow of the program they are developing, making it easy for the developer to model complex non-linear conversations. In complex conversations with complex logic and dynamic dialogue flows, the ability to see the dialogue path clearly is very beneficial for the chatbot to have a more realistic dialogue performance. Azurebot is developed for the enterprise, but to develop an Azure bot, the developer needs to get used to the intricacies of Microsoft cloud. IBM Watson Assistant is an artificial intelligence virtual assistant developed primarily for the enterprise. Unlike previous virtual assistants, Watson Assistant does not just passively complete questions and answers. Its highlight is that it knows when questions need to be answered. When employees need to understand a very large number of questions, perhaps thousands, Watson assistant can give them supports. It will help customers solve problems in terms of data. Different data can be stored on different clouds and can handle up to 1000 message requests at the same time. It does not just give a fixed and repetitive answer, and it focuses on the latest and most useful answer that the user wants. Chatbot development platforms geared towards conversational dialogue: ALICE bot is similar to Pandora bot in that they are part of a very old chatbot development platform that simplifies pattern matching algorithms. They do not require sophisticated natural language processing techniques but build simple logic to match input data to templates using some basic statements. Unlike the development platforms mentioned above, which are capable of handling very complex statements, Alice chatbot focuses on the scale of the data (AbuShawar and Atwell, 2015). AbuShawar and Atwell (2015) have developed versions with as many as 100w categories, which means that the chatbot can be quizzed on almost any everyday question rather than a specific domain, which would include the most basic conversations in normal life, which is convenient for developers who use Alice Chatbot (AbuShawar and Atwell, 2015).

### **2.3 Choice of Methods**

The previous sub-section discussed the research methodology, development strategy, development platform and solution for developing a chatbot. When thinking about how to develop a chatbot, we need to consider what purpose the chatbot is being developed for, then we need to establish the mechanics of the development, and once we have established both, we need to choose the right platform for the project.

The chatbot developed for this project needs to achieve simple campus transactions and student learning support. The type of this chatbot is a goal-oriented chatbot. When students ask chatbot questions about campus matters such as library bookings, it will inevitably result in a non-linear conversation. In addition, if a static development mechanism is chosen for the development of this chatbot, this module will not be able to be implemented. Therefore, I



chose a dynamic mechanism based on deep machine learning for the development mechanism. Through a selection process, I chose the Dialogflow platform for development. The rationale for choosing this platform was as follows:

- The mechanisms of Alice chatbot and Pandora Bot are not suitable for developing complex non-linear conversations, in addition to the amount of manual coding required when developers build conversation logic, and they need to keep doing the same work in code when building the basic chat logic, which would be time-consuming. In addition, the portability of the developed chatbot needs to be taken into account, as many university students now use Facebook to access information, so the Pandora bot would not be conducive to the development of this project.
- One of the biggest factors in choosing Dialogflow is that it comes in two versions. The standard version is completely free, but when external API deployment is required, there may be a cost, but this is optional. IBM will provide the developer with a fixed number of message entries per month, beyond which there will be charged. Lex is subject to AWS billing rules when the developer uses. AWS has a cost calculator, but its estimates are hardly accurate. Microsoft's complex software licensing agreements and the use of secret discounts can make the billing rules very complicated, which requires some knowledge of Microsoft billing rules. When students whose first language is not English type queries in languages other than English, Dialogflow supports up to 20 languages, whereas Lex only supports American English, which can be inconvenient. Support for many languages is also advantageous to the developer, as I can enter data in Chinese, for example, which will aid my development. Another most important point is that Azure and lex or even IBM are mainly for enterprises, and some of their internal settings will not be particularly friendly to students, Dialogflow can be a very good operation. Furthermore, I am a Google cloud user, so I am familiar with the Google series' rules and operations, which will drastically reduce my development time because I will not have to spend as much time learning the Google cloud's regulations. As a result of this study, I decided to develop with Dialogflow.

## **Chapter 3**

### **Experimental Design**

#### **3.1 Data Analysis**

This project is developed as a chatbot for students in the School of Computing at the University of Leeds. It will provide students with some basic help with school matters and will also provide students with quizzes on computer literacy. In designing the computer literacy module, I needed to collect knowledge about the subject of computing and enter it into the chatbot system in the form of quiz questions. When students ask the chatbot a question, the chatbot system will analyse and match the questions entered and provide an answer. The plan for this project is to collect 20 questions on knowledge of computing subjects and data analysis specialisms. Due to the limited development time, it was decided to use these questions and answers as the first version of the development pool. The selection of these questions requires a questionnaire survey of random students in the School of Computing to collect some of the most common questions, expand on these questions and then find answers. For the diversity of questions collected, such as how to understand CRISP and what is business understanding, data understanding and data preparation, the chatbot must have the ability to recognise these questions as the same and give the correct answers. Based on the variation in the questions asked of different students, the diversity of questions needs to be taken into account when collecting answers on a particular topic.

#### **3.2 Data Collection**

Having established the need for a question, I need to use an authoritative, reliable knowledge base to filter the question. Below is a list of popular websites and search systems that I have sought out regarding the field of computing.

##### **3.2.1 Google Scholar System**

This is the most well-known and popular search website. Users can find much well-known literatures covering most of the literary material and knowledge about computers.

##### **3.2.2 Wikipedia**

Wikipedia is an Internet encyclopedia with wide coverage, neutral content and accurate knowledge content (Voss, 2005). Searching for computer knowledge through Wikipedia is very rigorous. There will be many academic references in the

explanation of each knowledge, which is very authoritative. This also guarantees the scientific rigor of the data I collected.

Due to the need to collect data in a short time, I decided to use the above methods to collect datasets. However, I need to pay attention to the problem of data quality identification. The Internet has a lot of computer knowledge, but there are some wrong answers, including spelling mistakes. For example, spell "python" as "Pyton". These issues need attention when collecting data.

### 3.3 Experimental Tools

I select the Dialogflow platform for this project.

Dialogflow is a development platform based on Natural Language Understanding (NLU). Developers can design chat conversation modules on this platform, similar to the customer service software that is common in everyday life. A dialogue system developed based on this platform can generate a variety of applications, such as a virtual consumer assistant for a shopping website or a virtual employee assistant for an enterprise. Similarly, a chat system developed based on this platform can act as a student assistant at a university. Dialogflow allows developers to deploy chat interfaces to popular software such as Facebook and Twitter. Developers can use a very simple code call interface to connect to the database that chatbots need, or they can manually search for datasets for logical input. Since I do not have access to the university's internal database interface, the data for this project is manually sought and imported into the chatbot. Figure 3.1 shows how a Dialogflow agent works:

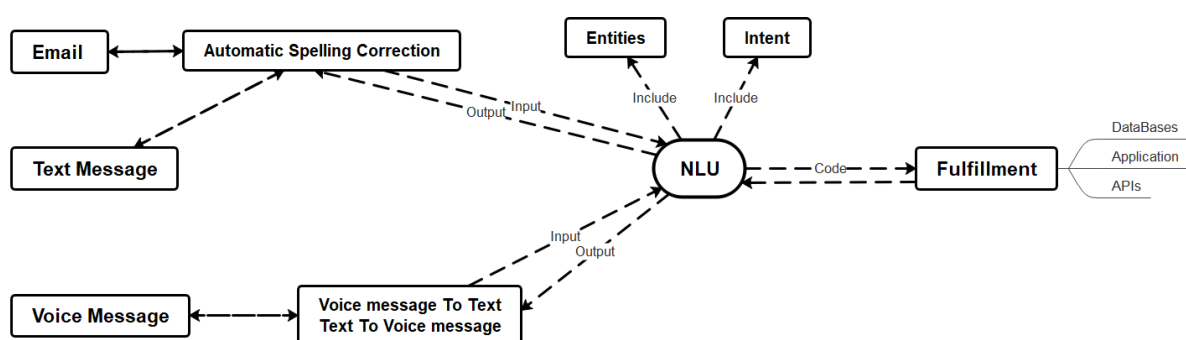


Figure 3.1: The internal operating structure of Dialogflow Agent

The Dialogflow agent works in three parts: Information processing entered by the user, NLU identification, and Integrated deployment of the chat interface. An explanation of how the Dialogflow agent works are explained below.

### 3.3.1 NLU

Natural language understanding is a branch of NLP(Natural Language Processing) that can be considered a subset of NLP. Its main function is that the computer processes natural language as its understandable data and then analyses the context and emotion of the data, thus giving the answer that best conforms to human emotion. At the core of Dialogflow is the NLU module, which consists mainly of intents and entities. Intents refer to a classification module that categorises the type of topic a user wants to speak. Users ask different questions when they talk to the chat system, each of which has a specific intent, and Figure 3.2 shows the role of intents and the entities:

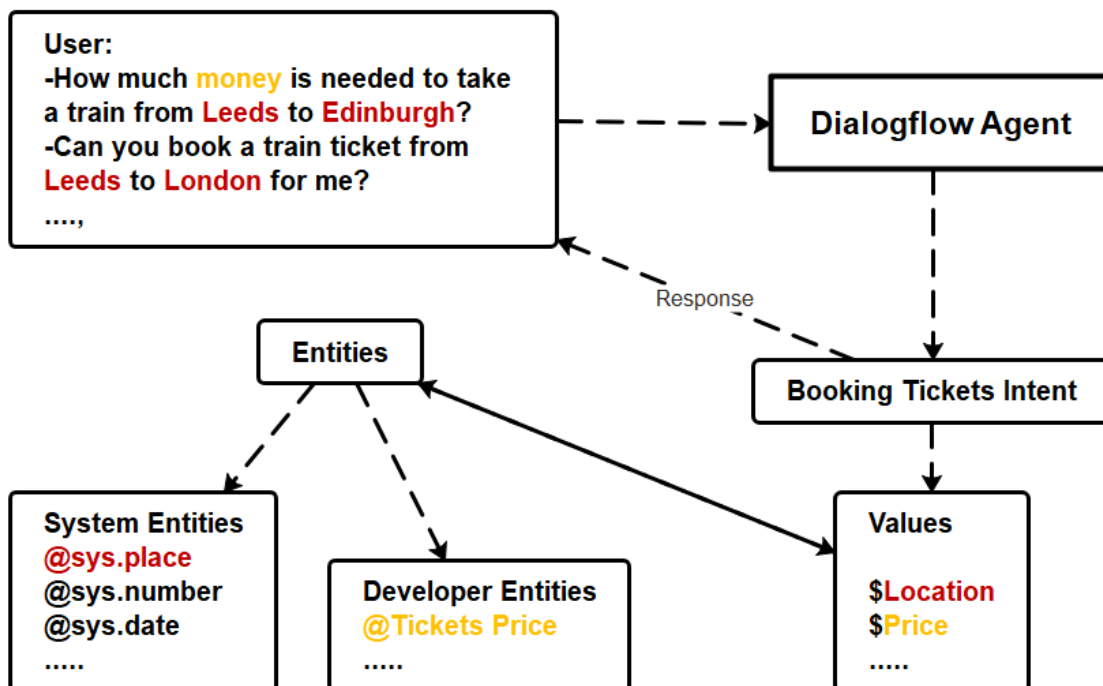


Figure 3.2: The role of intents and entities in a simple conversation

There are many intents in a Dialogflow agent that are intended to identify the categories of topics entered by the user, and when the user enters a problem, the agent matches the intent based on the specific information in the question entered by the user. As the example in the image above shows, when a user asks for a "train ticket from Leeds to London," the agent extracts the location in the question of the user and matches it to Booking Ticket intent. The intent is to extract the user input "Leeds" and "London" to match the Location value and finally return the answer to the user. Intent extracts the user questions as parameters. Each parameter has a developer-defined type, which is Entities. Entities are different defined attributes, such as information about locations in a user's question, which the agent matches to the system's place type so that the user's response can be given based on a specific entity. Entities are divided into two types: System Entity and Developer Entity. System

Entities are the entity types that the agent provides to developers, which have many basic attributes such as data, number, place and name. Developer Entities are created by developers according to their needs. For example, the developer needs the agent to identify the type of car in the user problem so that the @car type entity can be created. The role of entities is that the agent can perform logical-level operations after identifying the detailed data entered by the user. In other words, the intent is used to determine what services the user needs and entities are used to determine the details of the topics that the agent and the user are talking about, thus determining how to respond to the user.

### 3.3.2 User Input

As shown in the figure, Dialogflow allows users to send questions as an email, or they can type text or voice messages directly to ask questions to the agent. When a user asks questions in the form of a text message, the developer can use the Spelling Correction module to standardise the correctness of the text message entered because spelling errors inevitably occur during the input, but this module is optional depending on the decision of developer. The text information entered by the user is recognized by the agent to generate a response. When users ask a question to the agent in the form of a voice, Dialogflow activates the Speech Conversion function, which converts the voice messages from users into text messages that are analyzed by the agent and then output text messages to the user.

### 3.3.3 Fulfillment

Dialogflow provides Fulfillment function for developers. Developers can use it to implement more complex features, such as connecting external API, deploying chatbot to applications or connecting databases. Figure 3.3 shows how Fulfillment works:



Figure 3.3: The realization principle of Fulfillment function

After the developers have completed the creation of the chatbot, they may need to deploy it to some applications such as Facebook because this can allow more users to use it. When the developers need to connect to the database or access another system to improve the function of the agent internally, developers need to use API services. The API service is an on-demand billing service. Developers generate instructions by calling the API service

provided by the agent, using webhooks to interact with external services. Developers create instructions by writing simple code to match external applications.

Since I do not have access to the internal database of University of Leeds, I simply used the integration function to deploy the chatbot to Messenger.

### 3.3.4 Dialogflow Dashboard

Dialogflow Dashboard is shown in the following image. After creating an agent, the developer can select the corresponding module for development. Figure 3.4 is an explanation of the functions of the modules that need to be used:

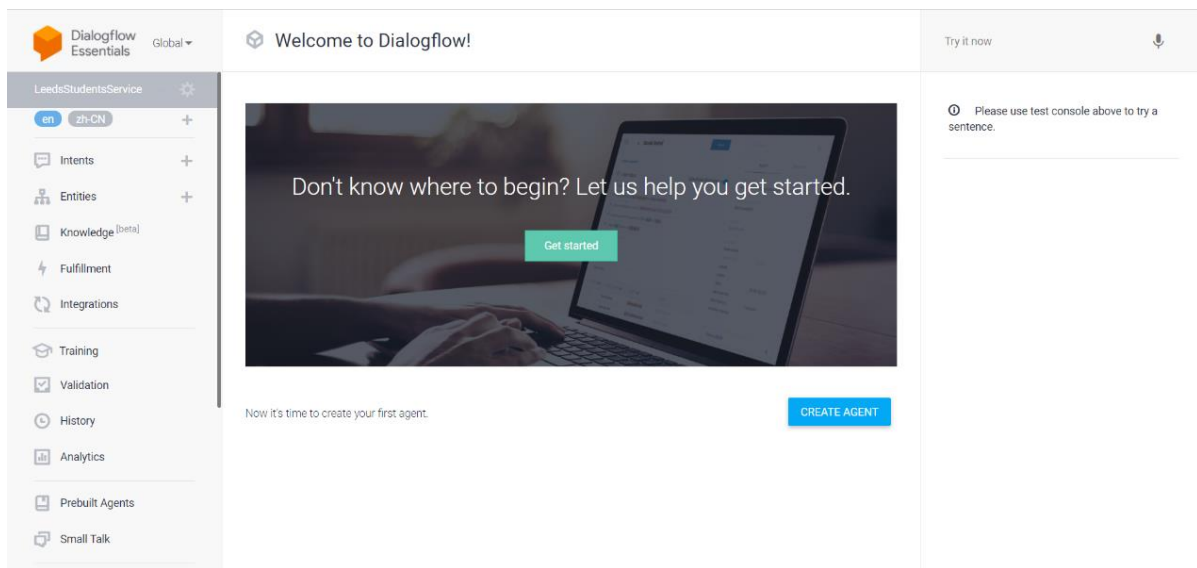
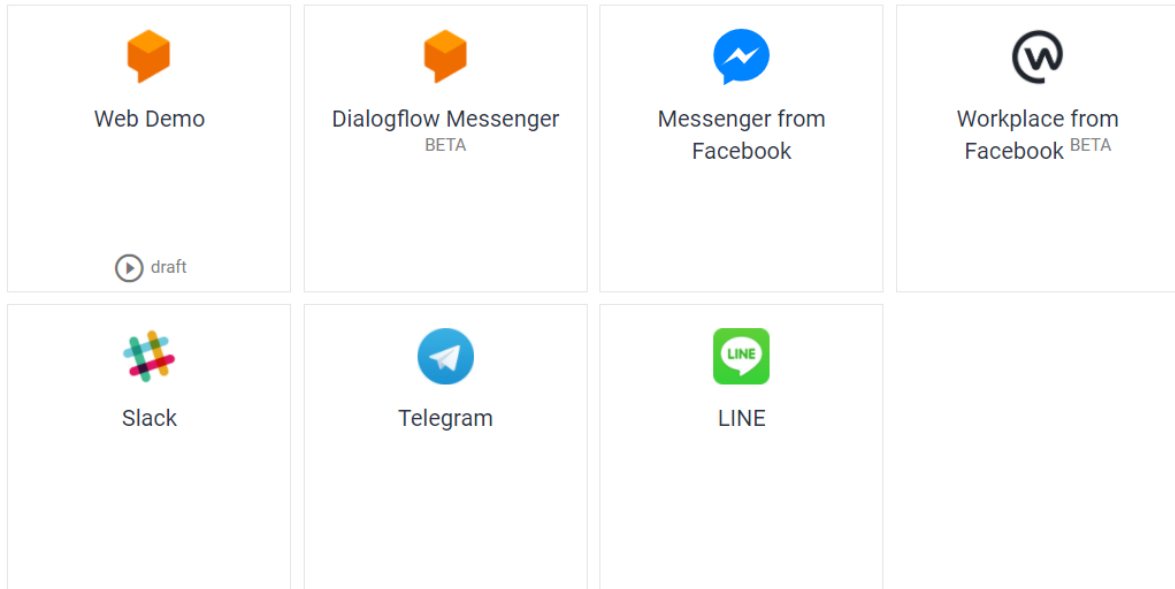


Figure 3.4: Dialogflow development interface dashboard

- **Intent.** Developers use this module to build different topic types. In this module, the developer needs to input training sentences and the corresponding entity types to make the agent learn independently and make it more intelligent.
- **Entities.** Developers need to enter some instance objects to create entities, which can be the brand name of cars or the name of different vegetables. The created entities are exploited by the agent intents, which automatically extend instances of the same types to the entities as the agent continues to talk to the user.
- **Integrations.** Developers use this module to deploy the agent to external applications. Figure 3.5 shows the external application platform supported by Dialogflow:
- **History.** Developers can view the details of the conversation completed by the agent in the history module, which includes any conversation records of users who successfully responded to and conversation records of users who did not respond successfully.

## Integrations

### Text based



### Open source



Figure 3.5: Accessible external applications in the Integrations module

### 3.4 Chatbot Structure Framework

Figure 3.6 shows the framework for the chatbot project I designed:

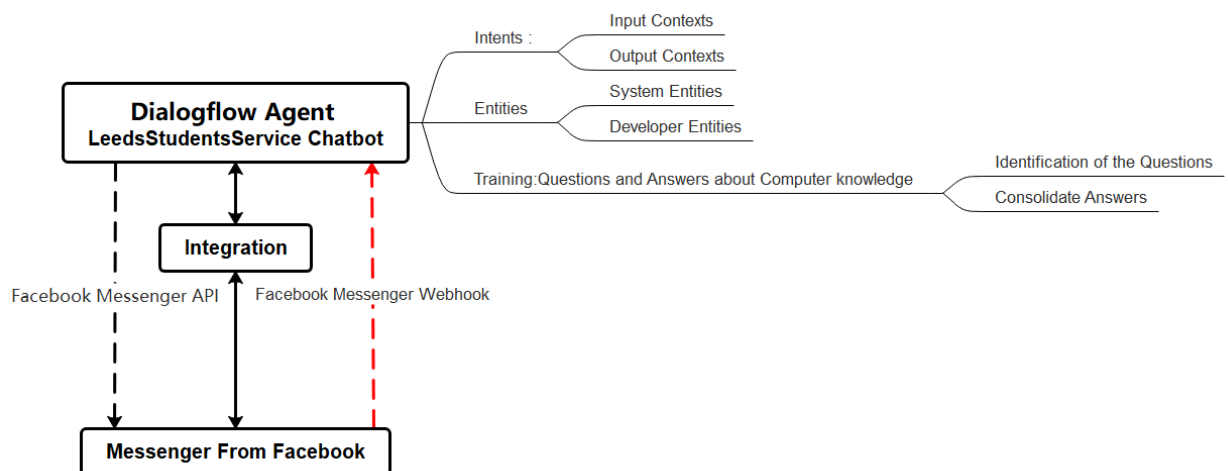


Figure 3.6: The overall design architecture of the chatbot of this project

Specifically, this chatbot completes the basic framework of chat by creating an agent in the Dialogflow dashboard and logically setting up Intents and Entities. The questions I collected are trained in Train module that comes with the dashboard. This step makes the chatbot smarter. In terms of interaction with external applications, I choose the Intergrations function to inherit the chatbot on Messenger through webhook to achieve interaction with users. Users send messages to the agent, and the agent receives messages through Messenger Webhook, and then uses the API to give the user a reply.



## **Chapter 4**

### **Experimental Implementation**

#### **4.1 Dataset**

Processing the collected data is necessary because the accuracy of the questions and answers are not carefully reviewed during the data collection phase, such as if there is a misspelling in some answers or the answer does not match the question. Since only twenty questions and answers are required for this project, so all the data I have is collected manually. I sorted out the computer knowledge questions raised by my classmates and then searched for answers to these questions through Google Scholar System and Wikipedia. This work is very time-consuming because I need to repeat the same operation and need to typeset carefully. All questions and answers will be shown in the attachment. The process of collecting data must ensure that the source of each question is indicated, and the person viewing the dataset can find the source of each data. In addition, the data I collected is legal and valid.

##### **4.1.1 Data Pre-Processing**

Once the answers to the questions are collected, one of the most important steps is to process the data. I need to check the answer for extra punctuation and spelling mistakes. After careful review, I ensured the correctness and credibility of the data and collated them.

##### **4.1.2 Data Display**

Since the answers to each question are too long, I put the answers section in the Appendix Chapter, which shows the twenty questions I have collected. Different students have different ways of asking questions. The chatbot needs to analyze and identify different ways of asking questions to accurately locate them to the corresponding answers. What needs to be stated here is that these questions are collected from the Internet, and the detailed content is shown in the Chapter Appedix C. Since students need standard and authoritative answers, I do not need to rewrite the collected data, but I have added the source of these questions. I have made a statement which this annotates the source of each piece of data in the README file in the dataset I uploaded using GitHub. All data collection processes are rigorous. I add references from the source websites, and every question can be found Its source. Table 4.1 shows the questions I collected, and the questions are sorted randomly:

Table 4.1: Twenty questions about computer knowledge required for this project

1. What is CRISP-DM?
2. How to understand “Decision Tree”?
3. What are the steps to establish a decision tree?
4. What is data mining?
5. What are the common data mining techniques?
6. What are the common data mining tools?
7. What are the methods of data analysis?
8. The case of Applied data mining
9. What is web mining?
10. What can web mining dig?
11. What are the types of web mining?
12. What are the core technologies for big data?
13. What the base Apache Hadoop framework includes?
14. What is Hadoop?
15. What is Hadoop distributed file system?
16. What are the services provided by HDFS?
17. What are the features of Cloud Computing?
18. What courses are offered for Master Degree?
19. Can you provide the tutors' information to me?
20. How do I view basic information about my majors?

## 4.2 Implementation of Features

The chatbot developed in this project has two module functions. One module is to answer questions from students about school admissions opening hours and library seat reservation services. The other module is to provide students with answers to questions. The questions of students come from the items shown in the table above. I put the detailed chatbot function display in the GitHub link.

For the first module, I created two conversation storylines:

- Students consult the opening hours of their postgraduate degree application at the University of Leeds.

- Chatbot helps students reserve library seats. For this conversation scenario, since I do not have access to the interface to the Library Management System at the University of Leeds, I am unable to match student appointment information to the library calendar, but the Library Reservations module can record student information, which will be shown later.

Figure 4.1 shows a logical example of a storyline in which students reserve a library seat in the first module:

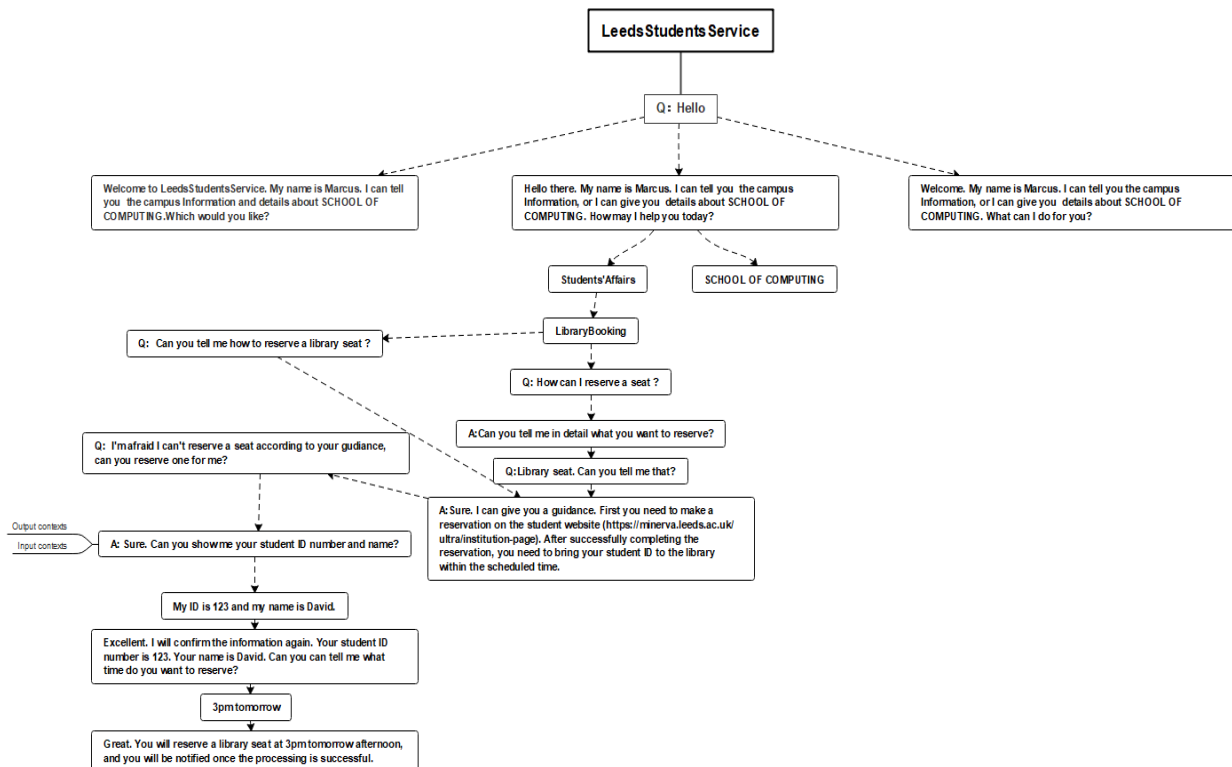


Figure 4.1: Dialogue logic structure of library seat reservation module

As shown, students send a reservation request to the agent after a brief hello. If the agent can identify the library reservation from the conversation, the student is given the reservation method directly, but due to individual differences, some students are unable to accurately represent the information, if the student proposes “ I want to reserve a seat ” information because it is not possible to accurately understand the needs of students, the agent will ask students what kind of seat need to be reserved. when the student gives details of the reservation, the agent completes the guidance step. In another case, the student is unable to reserve a seat. The student needs the agent to help reserve a seat. Similarly, after determining the need of student, If the student continues to make the “you can help me reserve a seat ” request, agent will recognize the new needs of student and match the dialogue intent to the reserved seat. Finally, the agent asked the students for basic information and responded to the students with a message about the success of the seat reservation.

The following sections will show the implementation of the Library Reservation module and the computer knowledge question and answer module. Since Library Reservation is the most complex module, but also includes the implementation of the remaining module functions of the operating steps. In order to avoid redundancy, I will mainly state the realization of the function of this module. The difference between the modules is the logical structure. In the QuestionAnswer section, I will show the logical flow I constructed. After describing the implementation of these functions, I will show the steps to deploy the agent on Messenger. Table 4.2 lists all intents for this chat system:

Table 4.2: The intents of the chatbot developed by this project

<b>Intents</b>	<b>An example of user input information</b>
Welcome	Hello, what's your name?
Masterdegreetime	Can you tell me when the University of Leeds opens the graduate application channel?
Librarybooking	Can you tell me how to reserve a library seat?
librarybooking-custom	I can't complete my reservation by myself, can you book it for me?
QuestionAnswer	What is a decision tree? (20 questions depend on the student, here is just one example).

Table 4.3 is the definition of all Developer Entities:

Table 4.3: Entities of the chatbot developed by this project

<b>Entities</b>	<b>Definition</b>
@Library-booking	Students need a chat system to help with requests for library seats
@QuestionAnswer	Questions that students want to ask
@StudentInformation	Basic information about students

### 4.2.1 Library Reservation

As a specific conversation scenario, I first need to create an intent called Librarybooking, which means that when a student raises the topic of library seats, the agent matches the conversation topic. At the same time, I need to create an entity for this topic called Library-Booking. Values are stored in this entity to determine whether the information entered by the student contains keywords related to library reservations, and the corresponding synonyms can be entered after each value, which enriches the values of this entity. Dialogflow provides an automatic expansion function. When this function is selected, the agent will expand similar words according to the value I input. When I define the value in entities as “library” or “library seat”, if the information entered by the student contains such words and their synonyms, the agent will match this entity type, and then successfully match the intent, and finally, give the student response. Figure 4.1 shows the operation to achieve this logic:

The screenshot shows the 'Librarybooking' intent configuration page in Dialogflow. At the top, there's a header with a blue 'Librarybooking' title and a 'SAVE' button. Below the header, the 'Training phrases' section is visible, featuring a search bar and a list of phrases. The phrases are: 'Add user expression', 'i want to learn in the library', 'how can i come into the library', 'what way i can do to apply for a library seat', 'i want to study in somewhere of campus', 'I want to learn in the library, how i can come in?', 'book', 'library booking', and 'I want to book a seat of library'. The words 'library' and 'library seat' are highlighted in yellow in the original image.

Training phrases
Add user expression
i want to learn in the library
how can i come into the library
what way i can do to apply for a library seat
i want to study in somewhere of campus
I want to learn in the library, how i can come in?
book
library booking
I want to book a seat of library

Figure 4.1: Training phrases module in library seat reservation intent

Training phrases need to be entered when an Intent is created. Training Phrases refer to problems that different users may enter and can be defined as different expressions from students. Different students have different ways of asking questions due to individual differences, but to ensure that the agent can match the correct intent from different questions on the same topic, I need to extend training phrases. Dialogflow is designed to collect and train different student representations by creating this feature, a process that improves the

ability of agent to identify different expressions. The more conversations the agent has, the more it recognizes the conversation.

The created entities match the parameter module to the intents automatically. As shown in Figure 4.2, @librarybooking is an established entity where various parameters are stored to match the information of students. Add a reply to the Response module, where the content is replied to the student when and only if there is a word such as “Library seat” in the statement of student that accurately represents the library reservation. When a student expresses a problem unclearly, such as a student entering "I want to reserve a seat", @librarybooking does not match the keyword “library”, the agent is unable to trigger a response but rather needs to ask the student for further questions. This needs to be set up in the prompt module in the parameter list. Considering that users may have problems that are not clearly stated but are not expected to waste resources due to terminating the conversation, Dialogflow sets up the logic of questioning.

Librarybooking

SAVE

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	library-booking	@Library-Booking	\$library-booking	<input checked="" type="checkbox"/>	Can you tell me...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

+ New parameter

Responses

DEFAULT +

Text Response

1

Sure. I can give you a guidance. First you need to make a reservation on the student website (<https://minerva.leeds.ac.uk/ultra/institution-page>). After successfully completing the reservation, you need to bring your student ID to the library within the scheduled time.

2

Enter a text response variant

ADD RESPONSES

Figure 4.2: Response and Library-booking entity action

Prompts for "library-booking"		
NAME	ENTITY	VALUE
library-booking	@Library-Booking	\$library-booking
PROMPTS		
1	Can you tell me in detail what you want to book	

Figure 4.3: Prompts for Library-booking information

As shown in Figure 4.3, in the prompt module, the developer can set up questions about more accurate information that the agent needs to obtain. This allows agent to determine whether students need a library seat reservation. This is entirely in line with the logic of the real dialogue.

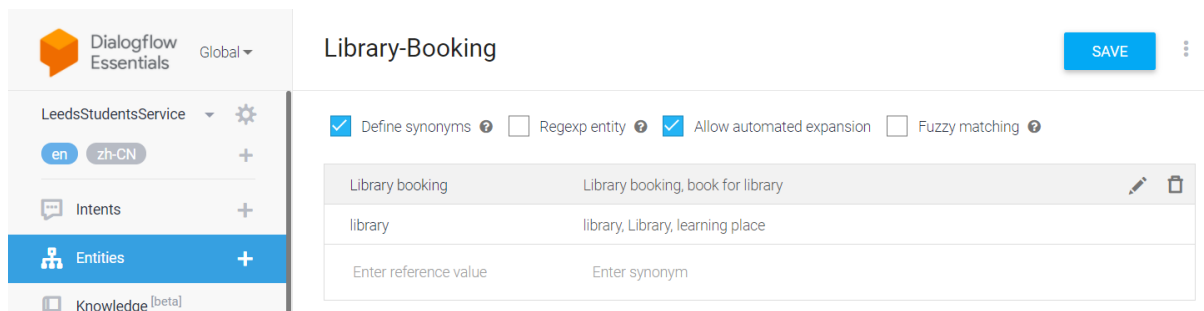


Figure 4.4: Library-Booking entity(Does not contain the complete reference value)

Figure 4.4 is the creation of an entity.

When a student needs an agent to help book a library seat, the agent needs to get basic information about the student. This is a typical non-linear conversation pattern because chatbots need to judge context based on context and give reasonable answers.

Dialogflow has the functionality of context. The context is divided into the input context and the output context, both of which are more like a trigger button. For example, when the agent asks "Please give me your student ID" and only after the student enters student ID, the context mechanism is triggered, the agent identifies to conduct the next round of conversations. In the next round of conversation, the agent asks the student name, and only after the student enters the name information, the agent will ask when this student has scheduled it. The purpose of setting up the context is to ensure that the agent has all the information it needs to complete the user's needs. In this library booking function, student information and scheduled dates are required to successfully book the library. When a student enters all of this information into an agent in a one-on-one conversation, the agent responds directly to a successful appointment. When a student enters only Student ID, the

agent cannot complete a library reservation by using Student ID alone, so the agent needs to continue to ask student for information based on context. Context is primarily used in non-linear conversations to determine consistency up and down a topic of user. When developer sets a follow-up intent for an intent, the next round of conversations can occur when and only if the input context in the follow-up intent is the output context of its parent intent, and the agent completes the conversation with the user in the conversation content that matches the sub-intent. Dialogflow has a fallback intent option, which creates the original sub-intent and generates a context that the developer needs to set the context based on conversation logic. Since the operation was the same when creating the intent, as shown in the above process, I use the logical framework to state the process of reserving a library seat, and Figure 4.5 shows the logical framework for booking a library seat:

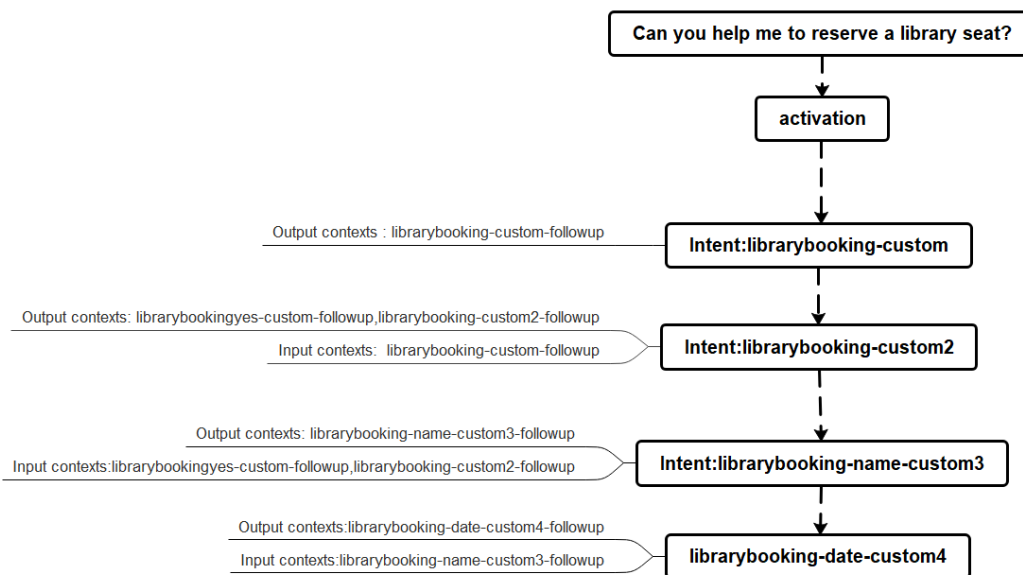


Figure 4.5: The role of input context and output context in dialogue

Firstly, I set up four intents. When a student needs agent to help reserve a seat in the library, the agent matches the "librarybooking-custom" intent according to the context, at which point the "librarybooking-custom" intent is activated, and its output context is activated. The agent then asks for the information of Student ID. If the student enters student ID, the agent recognizes the number. The number can be recognized by calling the system's number entity in "librarybooking-custom2" intent. This triggers a new round of conversation where the agent matches the "librarybooking-custom2" intent, at which point the input context in "librarybooking-custom2" intent is activated, and its output context is activated. The conversation moves on to the "librarybooking-name-custom3" intent only if the input and output contexts are fully activated. This sequence of dialogue execution continues until the end of the dialogue. Figure 4.6 and Figure 4.7 show the specific implementation interface.



This is a conversation between a student and an agent about library seat reservation. Figure 4.6 is implemented in Messenger:

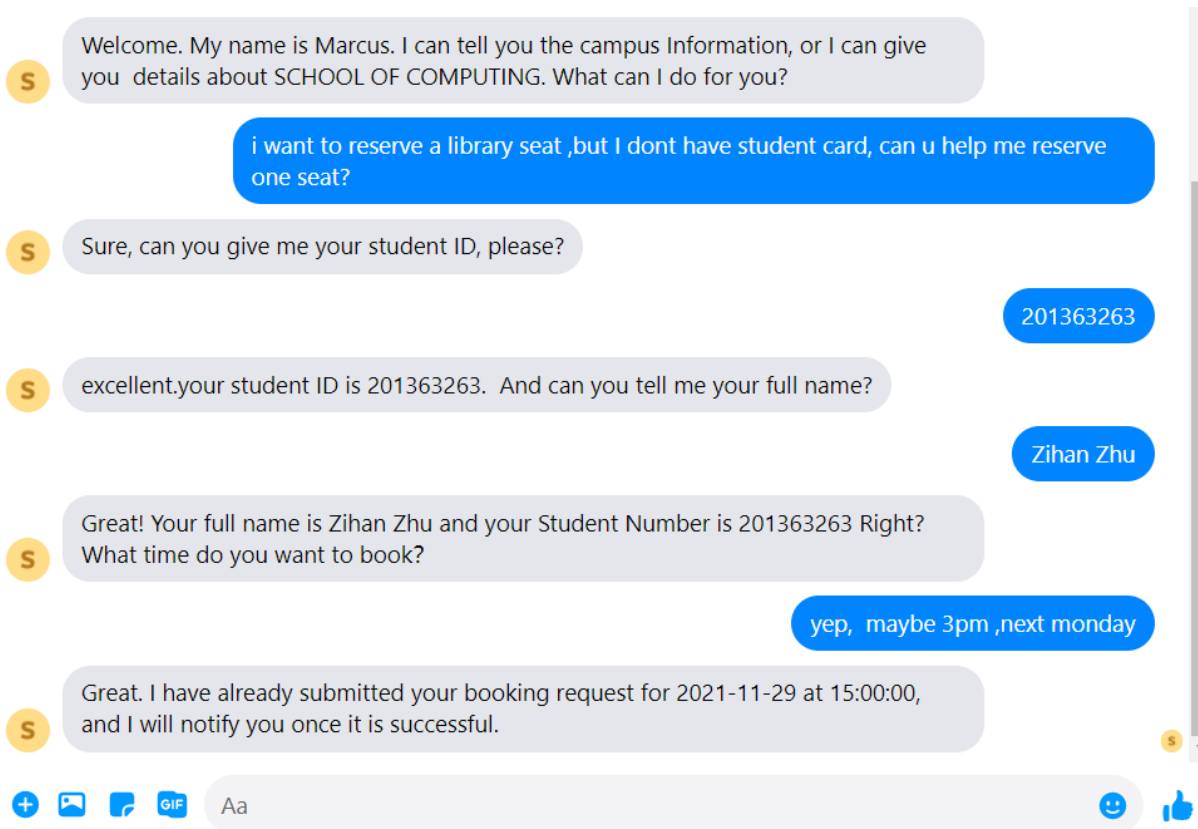


Figure 4.6: Dialogue display in Messenger

The agent can extract the required information according to the information entered by the student. In the above dialogue, the agent matches the student name I entered and reply to me as a confirmation message. I need to add a function to retrieve user information in the reply statement to achieve this reply. It can be shown in the following image. If the agent needs to include the student name in the response, I need to add \$person to the text response. The code “#librarybooking-custom2-followup.number” appears in the Response module in Figure 4.7. The function of this code is to retrieve the student ID information I entered in the previous intent because after a round of conversation, there is no @sys.number entity in in “librarybooking-name -customer3” intent, so it needs to be called using the code specified by Dialogflow.

• librarybooking-name-custom3

SAVE

Action and parameters

librarybooking.librarybooking-custom.librarybooking-custom-custom.librarybooking-custom2-custom

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?
<input type="checkbox"/>	person	@sys.person	\$person	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Responses ?

DEFAULT +

Text Response		
1	Great! Your full name is \$person and your Student Number is #librarybooking-custom2-followup.number Right? What time do you want to book?	
2	Enter a text response variant	

Figure 4.7: Set to extract the content of the previous round of conversations

## 4.2.2 QuestionAnswer

The above section is the process of implementing a module function. Similarly, in the QuestionAnswer module for student computer knowledge, I created The QuestionAnswer intent and entered the questions collected in the previous section as a Training Phrases input so that the agent can identify the different questions and answers of students. Figure 4.8 shows students asking questions about data mining tools:

The description of this question is "What are the common data mining tools? ". But the student asked the agent "Can you recommend some data mining tools?". This is allowed because the agent can identify different ways of asking the same question. I entered the amount of training phrases in the process of developing the agent, and the agent learns autonomously when there are "datamining" and "tools" in the student input information, or synonyms such as "application", the agent matches to give the correct answer. In addition, when the agent recommends the data mining tools commonly used by the student, the

student will further request the agent to provide the appropriate website link. Figure 4.9 shows this process. I realize this function by setting parallel intent. Parallel intent is a sub-intent that is set up under QuestionAnswer intent, and when the student triggers any of the common data mining tools given by the agent, the agent identifies the match to the corresponding intent.

Can you recommend me some several tools for data mining?

1. RapidMiner : RapidMiner is a data science software platform developed by the company of the same name that provides an integrated environment for data preparation, machine learning, deep learning, text mining, and predictive analytics. It is used for business and commercial applications as well as for research, education, training, rapid prototyping, and application development and supports all steps of the machine learning process including data preparation, results visualization, model validation and optimization. RapidMiner is developed on an open core model.

2. Google Cloud Platform (GCP), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, Google Drive, and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. Registration requires a credit card or bank account details.

3. R: R is a programming language and free software environment for statistical computing and graphics. It is supported by the R Core Team and the R Foundation for Statistical Computing. It is widely used among statisticians and data miners for developing statistical software and data analysis.

4. NLTK: The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. NLTK includes graphical demonstrations and sample data. It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit.

Figure 4.8: QuestionAnswer module dialogue display

4. NLTK: The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. NLTK includes graphical demonstrations and sample data. It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit, plus a cookbook..

5. Weka: A suite of machine learning software applications written in the Java programming language. Advantages of Weka include:Free availability under the GNU General Public License.Portability, since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform.A comprehensive collection of data preprocessing and modeling techniques.Ease of use due to its graphical user interfaces.

wow its great. i think it is good to use Rapid Miner, can u give me the website?

Of course. You can go to this website to learn about Rapid Miner:  
<https://rapidminer.com/>

Emm.....I also want to use Weka, can you give me a link to it

That is a greate choose. This is the link to WEKA:  
<https://www.cs.waikato.ac.nz/ml/weka/>

Thank u very much

Anytime. That's what I'm here for.

Figure 4.9: QuestionAnswer module dialogue display

Figure 4.10 is a comparison of the question and answers parallel intent with the follow-up intent of the library reservation module:

●	Librarybooking - custom	^
●	↳ librarybooking-custom2	^
●	↳ librarybooking-name-custom3	^
●	↳ librarybooking-data-custom4	
<input type="checkbox"/>	QuestionAnswer	^
●	↳ QuestionAnswer1	
●	↳ QuestionAnswer2	

Figure 4.10: Comparison of parallel intent and follow-up intent

The order of precedence between intents can be seen in the figure. The former is to execute the previous intent before moving on to the next intent, while the latter is to execute with the same priority, depending on the input of user.

### 4.2.3 Integrations

To deploy the agent to an external application, I need to use Integrations function. The first external application platform I chose was Messenger from Facebook. The agent is deployed to Messenger in a text-based interaction. Messenger is an instant messaging software where users can chat with others on the platform. To complete the deployment, I first need to have permission to use the Facebook API. As shown in Figure 4.11, I need to enter the Facebook application development website and follow the process to register as a developer so that I can create my own application on Facebook, which also includes using webhook to call external applications.

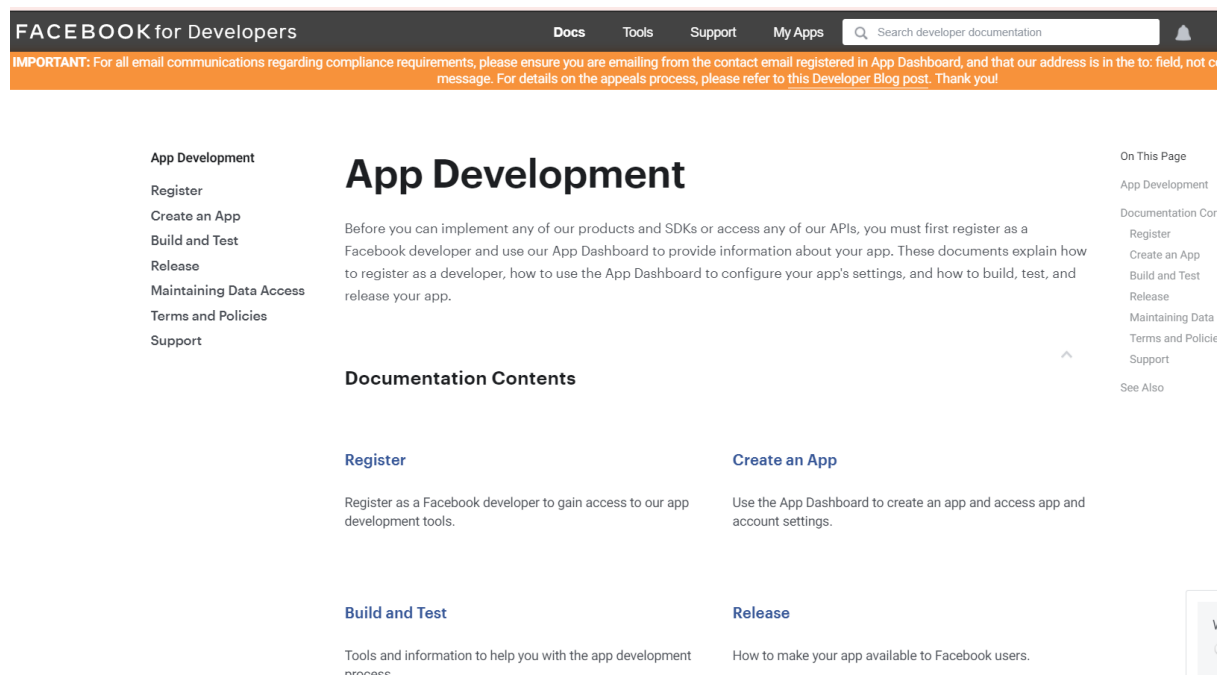


Figure 4.11: Facebook for developers guidance

After completing the registration, following the steps, I created an application called "StudentsService". Figure 4.12 shows the dashboard of the Facebook Developer interface:

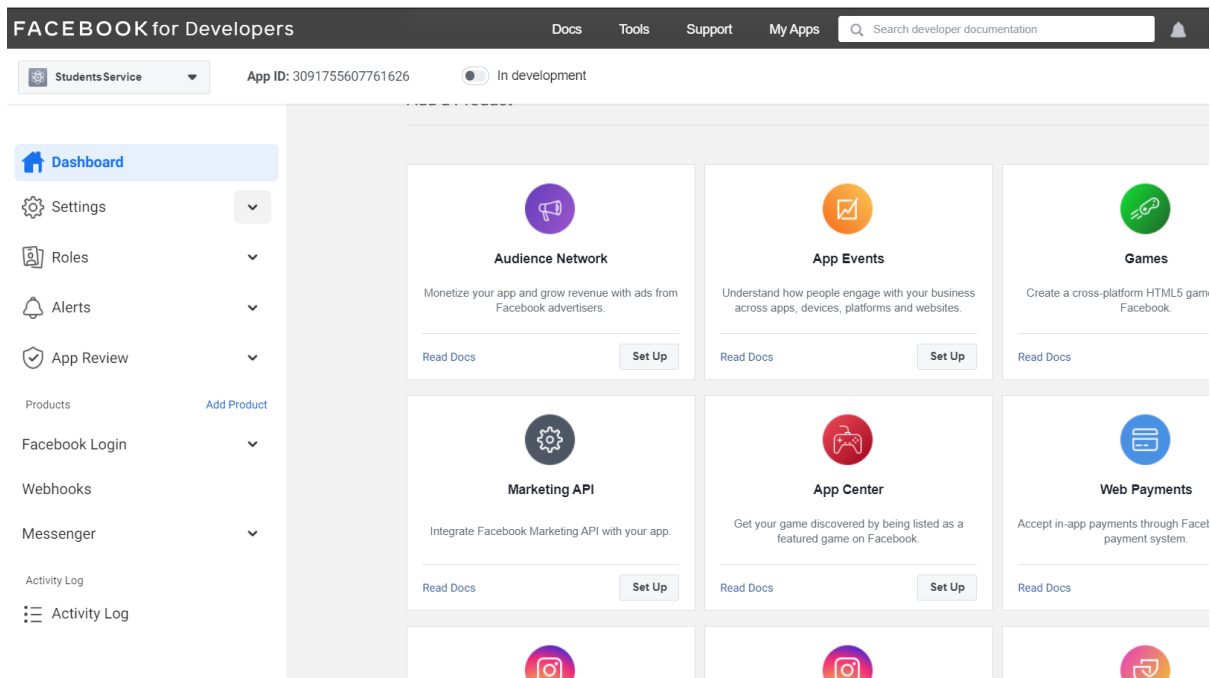


Figure 4.12: Facebook for developers interface

I need to add the Messenger platform to my Facebook application. After successfully completing this step, I can deploy the Dialogflow agent to Messenger. Specific steps are as follows:

- Get a Facebook access token, as shown in Figure 4.13, and clicking Generate Token in the Messenger interface in the dashboard generates a token.
- Copy the token and go back to the Dialogflow dashboard, selecting the Messenger app to paste the Token in the Integration interface.

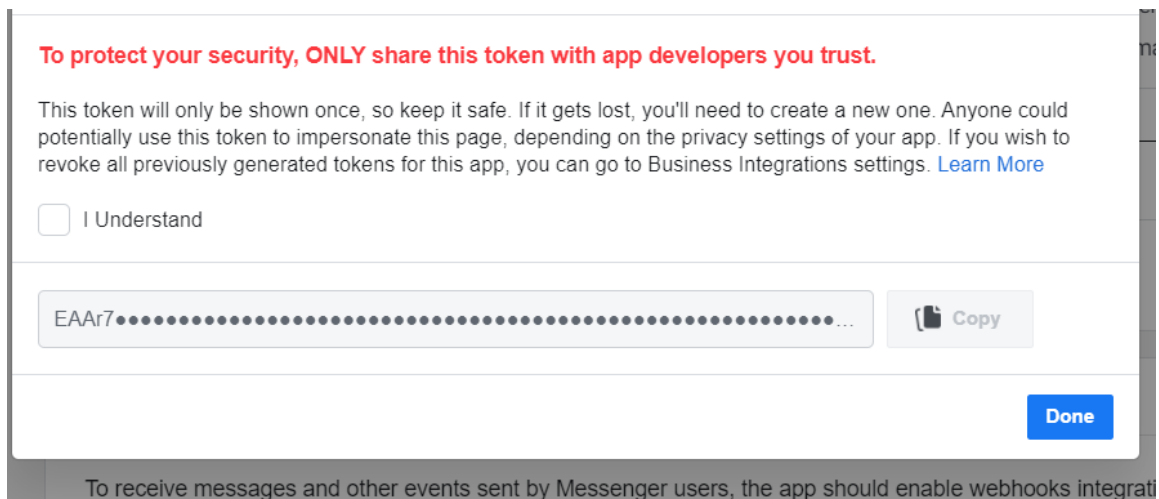


Figure 4.13: Token generation

After completing the above operations, I use webhook to integrate the agent with external applications. Dialogflow provided me with a Callback URL, and I also need to create a Verify

Token myself. Use the Callback URL and Verify Token to create an event in the Facebook Messenger Webhook Setup. Figure 4.14 shows the specific operation process:

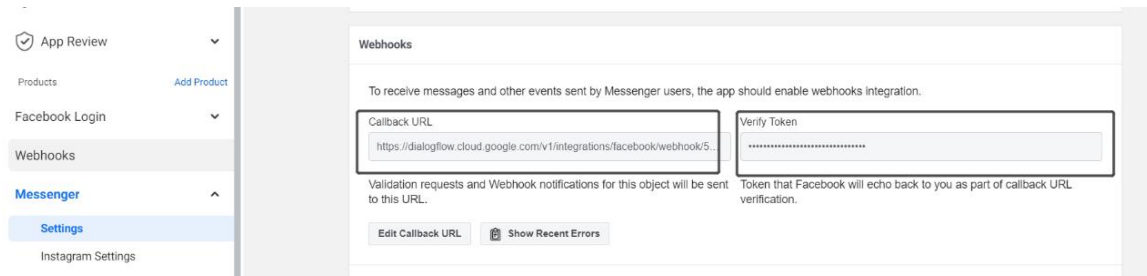


Figure 4.14: Webhook operation

The above is the deployment process of the agent. Figure 4.15 is the Messenger dialogue box of the web version. This is also allowed when the user uses the mobile version to chat with the agent.

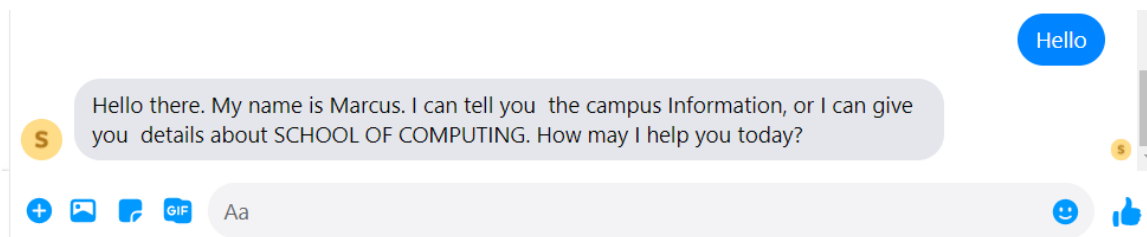


Figure 4.15: Web interface of chatbot

This chapter describes in detail the entire process of the function realization of the chatbot developed in this project. The content shown in the above section is the dialogue conducted on the Messenger web page and the operation in Dialogflow. I show the functions of this chatbot in Chapter Appendix D. The terminal is a mobile phone. The next chapter will evaluate the chatbot.

## **Chapter 5**

### **Evaluation**

The usefulness of a natural language chat system can be used as an indicator for evaluating an NLP natural language system (AbuShawar and Atwell, 2016). This dialogue system has different topic modules and different dialogue types. This chapter will conduct an overall evaluation of the module with usefulness and user satisfaction as indicators.

#### **5.1 Usefulness Evaluation**

Walker et al. (1997) gave four different methods for evaluating natural language dialogue systems, which are subjective or objective, quantitative or qualitative. The subjective evaluation mainly relies on the user's use and evaluation of the dialogue system. Through the use of chatbots, it can be determined that the system is able to give the user the correct answer. This can be used as a criterion for evaluating the usefulness of this chatbot. I will use the F-score method to evaluate the usefulness of this system. Since this chatbot performs intent matching based on the information entered by the user and gives answers in combination with context, it is feasible to use precision, recall, and F-score methods to evaluate the accuracy and usability of the agent.

I selected ten postgraduate students from the School of Computer at University of Leeds to use this chat system.

##### **5.1.1 Average Detection**

Since this chat system has multiple functional modules, the largest one belongs to the computer knowledge question-and-answer module because this module contains 20 questions for students to answer. As a more concerning function, I specially set up the evaluation of the computer knowledge question and answer module. The specific ideas are as follows: I used the average method for analysis (Abu Shawar, 2005).

- 10 students input computer questions in the dialogue interface. These questions come from 20 questions I have collected. After entering the question, the agent will give the student the answer to the relevant question. Students can ask chatbot questions according to their chat habits, which can be just a phrase. They will not be asked to enter the complete question.
- Count the number of students who got the correct reply from the agent. From this, judge the accuracy of the response of agent to the question. Table 5.1 lists the collected experimental results:



Table 5.1: The number of students who got the correct answers

Questions	Students who got the correct answer	Overall
1. What is CRISP-DM?	10	10
2. How to understand "Decision Tree"?	8	
3. What are the steps to establish a decision tree?	9	
4. What is data mining?	7	
5. What are the common data mining techniques?	4	
6. What are the common data mining tools?	5	
7. What are the methods of data analysis?	8	
8. The case of Applied data mining	7	
9. What is web mining?	10	
10. What can web mining dig?	10	
11. What are the types of web mining?	9	
12. What are the core technologies for big data?	10	
13. What the base Apache Hadoop framework includes?	6	
14. What is Hadoop?	9	
15. What is Hadoop distributed file system?	10	

16. What are the services provided by HDFS?	10	
17. What are the features of Cloud Computing?	7	
18. What courses are offered for Master Degree?	10	
19. Can you provide the tutors' information to me	10	
20. How do I view basic information about my majors?	10	

According to the data in the table, the average number of students who get the correct answer when asking the agent can be counted.

Use the average calculation formula (Abu Shawar, 2005):

$$Average = \frac{\sum_1^q Ni}{q} \quad (\text{formula 1})$$

Where:  $N_i$  represents the number of students who get the correct answer for each question.  $Q$  represents the total number of questions, and its value is 20.

The proportion of students who get the correct answer can be calculated by the following formula:

$$Proportion = \frac{\sum_1^q Ni}{(q * Y)} \quad (\text{formula 2})$$

Where:  $N_i$  represents the number of students who get the correct answer to each question.  $Q$  represents the total number of questions, its value is 20, and  $Y$  is the total number of students 10.

After calculation, it can be obtained that 84.5% of the students can get the answer. By communicating with the students participating in the assessment again, ask them about the form of input questions. They said that when they input questions about datamining tools, the answer provided by the chatbot is datamining technology. I think this may be due to the improper logical design, which caused the chatbot to make an error in identifying the problem and give an inconsistent answer. Another possible reason for this phenomenon is the insufficient amount of agent training. In addition, some students said that when they entered "decision tree Concept", they expected the agent to give an answer of "what is

decision tree", but the agent gave an answer of "how to build a decision tree", which did not match the actual answer. This may be due to the fact that when the user enters two independent words, the agent has ambiguities in the recognition keywords and fails to give a reasonable answer.

If the student enters a question that basically matches the question stem in the collected question bank, the agent will give the correct answer. However, due to individual differences, everyone will have their own habits. When inputting text information, it is impossible for everyone to be accurate in the information presented. Therefore, when testing different methods of the same question, the agent may make mistakes. However, there are only a few cases of errors, and the overall accuracy of the chat system is considerable. After the user corrects some grammatical expressions, the agent can still give the correct answer.

### 5.1.2 F-Score

Because this dialogue system also has other modules. For the overall evaluation of the agent, I asked these ten students to use all the modules of the chat system and select the History and Analysis modules on the Dialogflow dashboard interface to view the record statistics of ten sample students' conversations with the chatbot. It can count the number of times that students use different modules. When students input information that matches a certain module, whether the agent can successfully match the corresponding intent. For example, a student needs agent to help reserve a library seat. When asking the agent, whether the agent can successfully match the "librarybooking" intent and help the student solve the problem. When the agent does not respond to the student with the correct answer, it will be prompted with an exclamation mark in the History module of the dashboard. Figure 5.1 shows this feature:

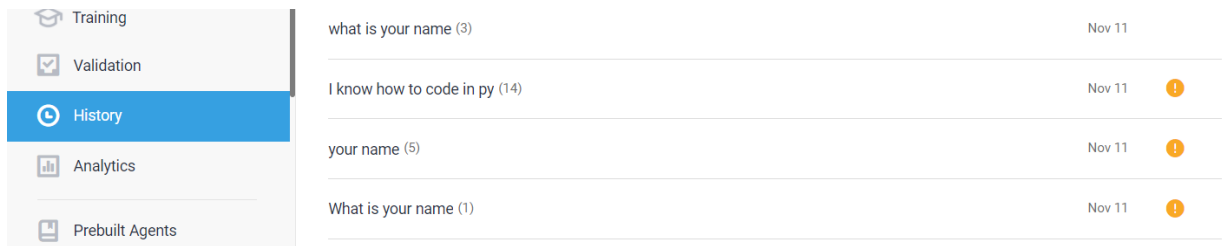


Figure 5.1: History module

The method used in the above evaluation is F-score method. Here are the three basic formulas:

$$P_i = \frac{TP_i}{TP_i + FP_i} \quad (\text{formula 3})$$

$$Ri = \frac{TPi}{TPi + FNi} \quad (\text{formula 4})$$

$$Fi = (1 + \beta^2) * \frac{Pi * Ri}{\beta^2 * (Pi + Ri)} \quad (\text{formula 5})$$

Where: P, R, F are precision, recall and F-Score respectively. Pi Ri Fi are calculations for different categories. i refers to different intents. TPi refers to the number of correct predictions or correct matches to i. FPi refers to the number of times that the i intent is not correctly matched. FNi refers to the number of incorrect matches or other intents predicted.

The specific implementation methods of this assessment are as follows:

- Ten students randomly use all modules of this chat system, and each module is prescribed twenty times.
- Evaluate the usage of each module when everyone has finished using it.

Regarding the number of intentions of this chat agent, the table has been listed in the previous chapter. There are six intents in total. After ten students completed the test, I used the statistics of the Dialogflow dashboard to sort out the number of matches for each intent.

Table 5.2 is the statistical data:

Table 5.2: TP, FP, FN values of all intents

Intents	TP	FP	FN
Welcome	20	0	0
Masterdegreetime	19	1	0
Librarybooking	16	0	8
librarybooking-custom	19	7	0
QuestionAnswer	18	2	0

According to the content in the above table, calculate the F-Score value of different intents. The  $\beta$  in the F-Score calculation formula is generally taken as 1. Table 5.3 shows the Pi, Ri and Fi of different intents:

Table 5.3: Pi, Ri, Fi values of all intents

Intents	Pi	Ri	Fi
Welcome	100%	100%	100%
Masterdegreetime	95%	100%	97.44%

Librarybooking	100%	66.66%	80.00%
librarybooking-custom	73.08%	100%	84.45%
QuestionAnswer	90%	100%	94.74%

From the F-Score test results of different intents in the above table, Welcome intention has the highest score, which is 100%. The F value of Masterdegreetime and QuestionAnswer intents also exceeded 90%, respectively 97.44% and 94.74%. The F-values of Librarybooking and librarybooking-custom are both above 80%, 80% and 84.45%, respectively.

Through the above evaluation, it can be determined this chatbot can respond to user questions based on correct answers, but there are still many problems, such as matching wrong intents and giving users wrong answers. Due to the intents of this chatbot and the number of questions and answers collected are not very large, and the number of samples set for this evaluation is only ten times. In the case of insufficient sample size and insufficient evaluation times, the calculated data cannot be used as an evaluation of the chat system in a strict sense, but according to the data provided by students after use, the usefulness of this project meets the expected standard.

## 5.2 Satisfaction Survey

After the students used the dialogue system in accordance with the instructions, they distributed condolences to ten students and asked them to do the last task: filling out the satisfaction questionnaire. The questionnaire is filled out anonymously, and the questions in the questionnaire are shown in Table 5.4:

Table 5.4: Questionnaire

Questions	Options				
	Great	Good	Medium	Not very good	Very bad
1. What do you think of the function of this chat system?					
2. Does this chat system help you solve the problems in the testing process?	Yes				No

3. Do you think this chat system has the ability to help students learn?	Yes				No
4. Are you willing to promote this chatbot to others?	Yes				No
5. Write down your suggestions for the chat system and questions.					

After collecting all the student evaluations, I sorted the results. From these results, it can be found that 6 students think the function of the chat system is good, and four students think the function is medium. During the test, eight students were helped by the chatbot to solve the test problems, and two students' problems were not successfully solved. Ten students think that this chat system is helpful to help students learn, and they are willing to help promote this chat system, regarding the suggestions of students on the chat system. The most-mentioned suggestion is "When operating the library reservation module, I input need to reserve the library seat". In fact, I only need him to give me a guide, but it asks for my student information." After I input the student information, it needs my name information, my name is very long, I typed in the chatbox, but when I sent my name information, he did not successfully receive ". Regarding this question, the reason may be that I did not set the life cycle of follow-up intents. The Dialogflow platform has time to set the cycle of follow-up intents. After a certain period of time, the follow-up intents will die. They also mentioned some suggestions, such as "add more functions such as class change function", "online postponement function", "when asking questions about computer knowledge, I hope to add some pictures, which can help understand the answer to the question rather than read a lot of text, or you can pay attention to the layout of the text. Otherwise the dense text will affect the reading. "The question bank should be increased, and the assessment function can be added if possible." These are very helpful suggestions. Although the sample size is small, there are only ten students, but through their serious evaluation, they also wrote valuable satisfaction feedback. Generally speaking, as far as users are concerned, they are very satisfied with this chatbot.

## **Chapter 6**

### **Conclusion and Future Work**

#### **6.1 Conclusions**

This project was intended to develop a chat robot for students of School of Computing at University of Leeds that can help them deal with school affairs and provide computer knowledge questions and answers to help them improve their learning efficiency and help them learn. This chatbot is developed using a platform based on google Dialogflow. The research process of the project is as follows: First, compare different natural language chat system development platforms and choose Dialogflow for development. At the same time, I manually collected twenty questions and answers about computer science from Google Scholar and Internet, standardized the answers to these questions, and then expanded the different ways of asking a question. Secondly, create an agent in the Dialogflow platform, and develop the logic diagram of the module by studying the principle of the chat system on this platform. The main module functions are: helping students reserve seats in the library, providing students with the latest enrolment information and computer knowledge questions and answers. The basic framework of the chat system is formed by establishing intents and entities in the platform and inputting training sentences to train the sentence recognition ability of the agent and expand it. After completing the development, deploy the agent to Facebook Messenger. Finally, ten students from the School of Computing were selected to evaluate the usability and satisfaction of the system. The evaluation results show that the chat system is highly usable and has also been supported and affirmed by users.

##### **6.1.1 Challenges**

1. After creating a new agent model and perfecting the functions of all modules, I need to spend time training, and I need to manually input various training sentences in the early stage.
2. Since twenty essay questions are collected manually, I need to carefully check the accuracy of the answers and the correctness of the spelling of the words. After completing the data check, I need to manually import the questions and answers into the chat system, which is prone to errors.
3. Since the Dialogflow platform does not disclose the internal logic algorithm code of its chat system, it is necessary to constantly check the information to be familiar with this knowledge.
4. When I checked the data set, I found that although it was carefully sorted out, there were still duplicate questions. For example, two identical answers appeared to answer "data

mining techniques". I also uploaded the revised data set to GitHub. To deal with duplicate data requires carefulness and patience.

### **6.1.2 Limitations**

1. The question and answer module of this project contains only twenty questions, which is very small in the amount of data and cannot answer all the questions that students want to solve;
2. The library seat reservation module of this chatbot has realized that the chatbot records student reservation information, confirms the student reservation schedule, and uploads the student reservation information to the schedule of the chat system. The basic functions have been fully implemented, but because I cannot obtain the internal system permissions of the University of Leeds library, I cannot update the student request to the internal calendar of library.
3. The number of students participating in the evaluation of the chat system is too small, and due to time constraints, it is impossible to get a larger number of students to evaluate. To a certain extent, this will affect the accuracy of the evaluation.
5. In the library reservation module, identification errors are prone to occur, which may be caused by the inaccurate matching logic of chatbot for the user's input information.
4. Due to time constraints, this project should have been better optimized in terms of module function logic.

### **6.2 Future Work**

This project has achieved all functions and achieved the expected goals after evaluation. But in order to make the chatbot function more perfect, there are still areas for improvement in the future:

1. Improve functional modules. On the basis of the original functional modules, functional modules such as student registration, student course selection, and student make-up exams can be added, which can make the chatbot more widely applicable. At the same time, I also need to optimize the existing module functions. For example, in the library seat reservation module, input more training sentences, and rebuild the logical structure of this module to ensure that the chat system can respond accurately when the user needs to reserve a seat.
2. Use the database. In the computer question and answer module, since there are only 20 questions, it is impossible to show the full capabilities of the chatbot. It can be connected to an external learning material database to expand the knowledge storage of the system. Once connected to the database, as the database is constantly being updated, the chatbot is constantly adapting to the update iteration.



3. Try to communicate with the school to obtain the campus library system API, so that students can also make reservations for the library through this chat assistant. According to the function displayed in the previous chapter, it is very convenient to use this chat assistant to make library reservations. If I can have the authority, this function can be made more comprehensive.
4. Get more students to use. The purpose of the chatbot is to serve a specific target group. If I spend time looking for more classmates to promote and use this chatbot, it will help me collect more and more accurate evaluation data, and it can be based on many feedback from students to get more inspiration to optimize module functions and increase module functions.

## List of References

- Abbas, N., Pickard, T., Atwell, E. and Walker, A. 2021. University Student Surveys Using Chatbots: Artificial Intelligence Conversational Agents *In: P. Zaphiris and A. Ioannou, eds. Learning and Collaboration Technologies: Games and Virtual Environments for Learning*. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp.155–169.
- Abu Shawar, B. and Atwell, E. 2007. Different measurement metrics to evaluate a chatbot system *In: Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies* [Online]. Rochester, NY: Association for Computational Linguistics, pp.89–96. [Accessed 9 December 2021]. Available from: <https://aclanthology.org/W07-0313>.
- Abu Shawar, B.A. 2005. *A corpus based approach to generalise a chatbot system*. [Online] phd, University of Leeds. [Accessed 7 December 2021]. Available from: <https://etheses.whiterose.ac.uk/1323/>.
- AbuShawar, B. and Atwell, E. 2015. ALICE chatbot: Trials and outputs. *Computacion y Sistemas*. **19**(4), pp.625–632.
- AbuShawar, B. and Atwell, E. 2016. Usefulness, localizability, humanness, and language-benefit: additional evaluation criteria for natural language dialogue systems. *International Journal of Speech Technology*. **19**(2), pp.373–383.
- Agarwal, R. and Wadhwa, M. 2020. Review of State-of-the-Art Design Techniques for Chatbots. *SN Computer Science*. **1**(5), p.246.
- Buyrukoğlu, S. and Yılmaz, Y. 2021. A Novel Semi-Automated Chatbot Model: Providing Consistent Response of Students' Email in Higher Education based on Case-Based Reasoning and Latent Semantic Analysis. , p.7.
- Chang, C.-Y., Kuo, S. and Hwang, G.-H. 2022. Chatbot-facilitated nursing education: incorporating a knowledge-based chatbot system into a nursing training program. *Educational Technology & Society*.
- Hien, H.T., Cuong, P.-N., Nam, L.N.H., Nhung, H.L.T.K. and Thang, L.D. 2018. Intelligent Assistants in Higher-Education Environments: The FIT-EBot, a Chatbot for Administrative and Learning Support *In: Proceedings of the Ninth International Symposium on Information and Communication Technology - SoICT 2018* [Online]. Danang City, Viet Nam: ACM Press, pp.69–76. [Accessed 6 December 2021]. Available from: <http://dl.acm.org/citation.cfm?doid=3287921.3287937>.
- Kasthuri, E. and Balaji, S. 2021. A Chatbot for Changing Lifestyle in Education *In: 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*., pp.1317–1322.
- Lee, J.H., Yang, H., Shin, D. and Kim, H. 2020. Chatbots. *ELT Journal*. **74**(3), pp.338–344.
- Li, Y.S., Lam, C.S.N. and See, C. 2021. Using a Machine Learning Architecture to Create an AI-Powered Chatbot for Anatomy Education. *Medical Science Educator*.

- Rahman, A.M., Mamun, A.A. and Islam, A. 2017. Programming challenges of chatbot: Current and future prospective *In: 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*., pp.75–78.
- Ranoliya, B.R., Raghuwanshi, N. and Singh, S. 2017. Chatbot for university related FAQs *In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* [Online]. Udipi: IEEE, pp.1525–1530. [Accessed 9 December 2021]. Available from: <http://ieeexplore.ieee.org/document/8126057/>.
- Shawar, B.A. and Atwell, E. n.d. Chatbots: Are they Really Useful? , p.22.
- Sophia, J.J. and Jacob, T.P. 2021. EDUBOT-A Chatbot For Education in Covid-19 Pandemic and VQAbot Comparison *In: 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)* [Online]. Coimbatore, India: IEEE, pp.1707–1714. [Accessed 6 December 2021]. Available from: <https://ieeexplore.ieee.org/document/9532611/>.
- Valdivieso, P.Á.C. and Luzón, M. del C.A. 2021. The use of chatbot as an element of tutorial action in university teaching. *ReiDoCrea*. **10**(24), pp.1–14.
- Voss, J. 2005. Measuring Wikipedia [Accessed 8 December 2021]. Available from: <http://eprints.rclis.org/6207/>.
- Weizenbaum, J. 1966. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*. **9**(1), pp.36–45.
- Zhou, L., Gao, J., Li, D. and Shum, H.-Y. 2020. The Design and Implementation of Xiaolce, an Empathetic Social Chatbot. *Computational Linguistics*. **46**(1), pp.53–93.
- Walker, M., Litman, A., Kamm, D., and Abella, A. 1997. Evaluating interactive Dialogue systems: Extending component evaluation to integrated system evaluation. *In Proceedings of the ACL/EACL workshop on spoken dialogue systems*, pp.1–8.

## **Appendix A**

### **External Materials**

The development of this project requires external tools, including applications and development platforms. These tools help me develop chatbot more conveniently:

- Google Dialogflow: <https://dialogflow.cloud.google.com/>
- Facebook for Developer: <https://developers.facebook.com/>
- Messenger: <https://www.messenger.com/>

## **Appendix B**

### **Ethical Issues Addressed**

1. This chatbot is provided to students for free, it will not become a profit tool.
2. All questionnaires are destroyed immediately after collection and statistics. At the same time, all evaluation methods are anonymous, which ensures the privacy of students.
3. The information of students will not be leaked because only people with permission can participate in the management of this chatbot, and only a verification token can be managed on the Facebook Developer side. When students perform library seat reservation operations, their student information will not be leaked.
4. The management of student data is in the Facebook Developer, so it is very safe. Here are the relevant links:  
  
Facebook for Developers:  
[https://developers.facebook.com/search/?referrer=dev\\_header&q=messages%20](https://developers.facebook.com/search/?referrer=dev_header&q=messages%20)  
  
Messenger:  
<https://www.facebook.com/messenger/>
5. This chatbot is suitable for students. Any chat content does not involve speech that is not conducive to society, and will not conduct illegal operations through this chatbot.

## Appendix C

### Questions and Answers Dataset

I list below all the references of every answers in the QuestionAnswer module of this chatbot:

**Declaration:** The answers to these questions are from Wikipedia, and I add a source of reference after each answer so there is no plagiarism.

The detailed dataset can be viewed by viewing the files I uploaded in GitHub:

[https://github.com/RomaMarcus/ZihanZhu\\_MScproject\\_Q-A-Dataset](https://github.com/RomaMarcus/ZihanZhu_MScproject_Q-A-Dataset)

For the answers in the above, the references are listed below (Since the above answer comes from many encyclopaedia entries, there is no author name, and it is Anon):

Anon 2021a. Apache Hadoop. *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Apache\\_Hadoop&oldid=1053186958](https://en.wikipedia.org/w/index.php?title=Apache_Hadoop&oldid=1053186958).

Anon 2021b. Big data. *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Big\\_data&oldid=1059261866](https://en.wikipedia.org/w/index.php?title=Big_data&oldid=1059261866).

Anon 2021c. Cloud computing. *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Cloud\\_computing&oldid=1059266571](https://en.wikipedia.org/w/index.php?title=Cloud_computing&oldid=1059266571).

Anon 2021d. Cross-industry standard process for data mining. *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Cross-industry\\_standard\\_process\\_for\\_data\\_mining&oldid=1055377187](https://en.wikipedia.org/w/index.php?title=Cross-industry_standard_process_for_data_mining&oldid=1055377187).

Anon 2021e. Data analysis. *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Data\\_analysis&oldid=1055861488](https://en.wikipedia.org/w/index.php?title=Data_analysis&oldid=1055861488).

Anon 2021f. Data mining. *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Data\\_mining&oldid=1059125769](https://en.wikipedia.org/w/index.php?title=Data_mining&oldid=1059125769).

Anon 2021g. Decision tree. *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Decision\\_tree&oldid=1058119806](https://en.wikipedia.org/w/index.php?title=Decision_tree&oldid=1058119806).

Anon 2021h. Google Cloud Platform. *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Google\\_Cloud\\_Platform&oldid=1059080772](https://en.wikipedia.org/w/index.php?title=Google_Cloud_Platform&oldid=1059080772).

Anon 2021i. Natural Language Toolkit. *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Natural\\_Language\\_Toolkit&oldid=1026979174](https://en.wikipedia.org/w/index.php?title=Natural_Language_Toolkit&oldid=1026979174).

Anon n.d. People | School of Computing | University of Leeds. [Accessed 8 December 2021a]. Available from: <https://eps.leeds.ac.uk/computing/stafflist>.

Anon 2021j. R (programming language). *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=R\\_\(programming\\_language\)&oldid=1058424566](https://en.wikipedia.org/w/index.php?title=R_(programming_language)&oldid=1058424566).

Anon 2021k. RapidMiner. *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: <https://en.wikipedia.org/w/index.php?title=RapidMiner&oldid=1048649117>.

Anon n.d. Research and innovation | School of Computing | University of Leeds. [Accessed 8 December 2021b]. Available from: <https://eps.leeds.ac.uk/computing-research-innovation>.

Anon 2021l. Web mining. *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Web\\_mining&oldid=1045158979](https://en.wikipedia.org/w/index.php?title=Web_mining&oldid=1045158979).

Anon 2021m. Weka (machine learning). *Wikipedia*. [Online]. [Accessed 8 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Weka\\_\(machine\\_learning\)&oldid=1051745380](https://en.wikipedia.org/w/index.php?title=Weka_(machine_learning)&oldid=1051745380).

Anon 2011n. Category:Applied data mining. *Wikipedia*. [Online]. [Accessed 9 December 2021]. Available from: [https://en.wikipedia.org/w/index.php?title=Category:Applied\\_data\\_mining&oldid=455699764](https://en.wikipedia.org/w/index.php?title=Category:Applied_data_mining&oldid=455699764).

## Appendix D

### Chatbot Function Display

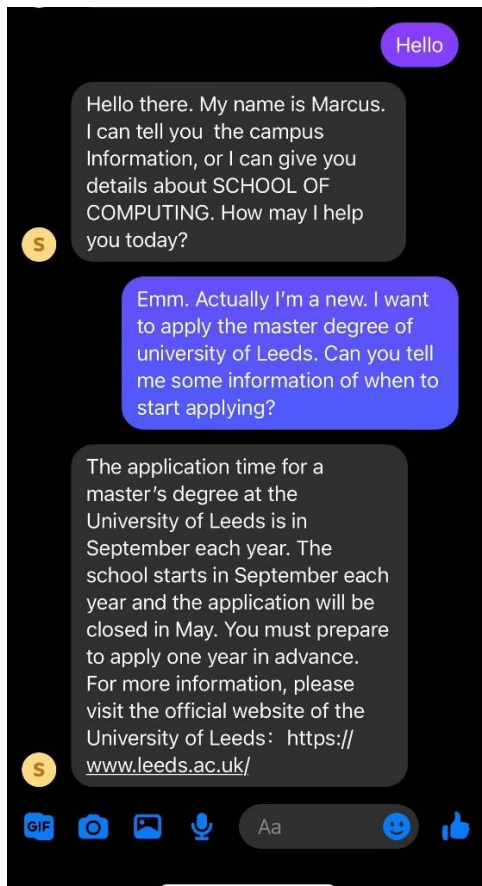
The figures below show the function of this chatbot: (In the question answer module in the picture below, I only list three questions. Since the picture takes up too many pages, I put the rest of the display in Github)

#### Library Reservation:

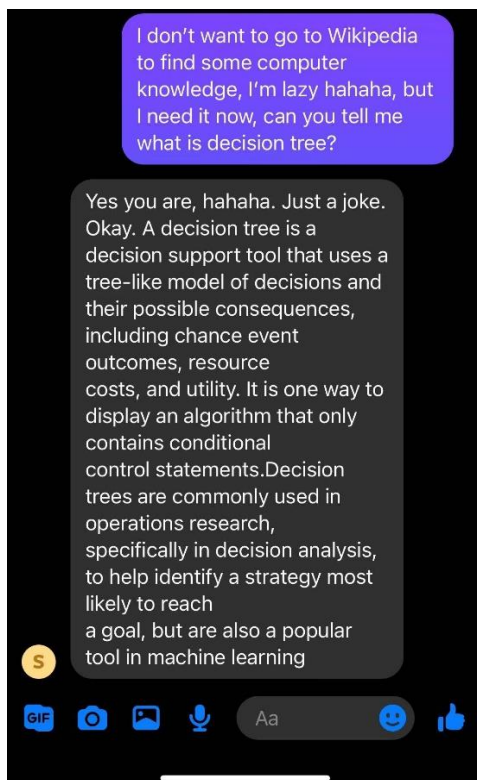


#### University Information Consultation:





### Question and Answer module:



I don't want to go to Wikipedia to find some computer knowledge, I'm lazy hahaha, but I need it now, can you tell me what is decision tree?

Yes you are, hahaha. Just a joke. Okay. A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning

And emm, I also want to know some tools for data mining, can you give me some information?

1. RapidMiner : RapidMiner is a

language. Advantages of R include: Free availability under the GNU General Public License. Portability, since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform. A comprehensive collection of data preprocessing and modeling techniques. Ease of use due to its graphical user interfaces.

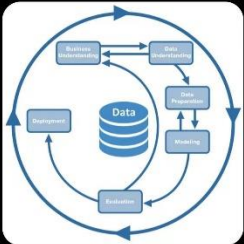
ah, I think R programming is nice because I have this course this term, maybe you can give me a website that I can download something? could you?

This is pretty good. If you want to learn about R-programming, I can share its official website with you. This is the website link: <https://www.r-project.org/>. You can also go to Wikipedia to see more about R technology, here is another link: [https://en.wikipedia.org/wiki/R\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/R_(programming_language))

ebbsite that I can download something? could you?

This is pretty good. If you want to learn about R-programming, I can share its official website with you. This is the website link: <https://www.r-project.org/>. You can also go to Wikipedia to see more about R technology, here is another link: [https://en.wikipedia.org/wiki/R\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/R_(programming_language))

Emm. I don't want words, so dizzy right? Anyway, I need a CRISP-DM flow chart, can you tell me what is Crisp-Dm



The diagram illustrates the CRISP-DM (Cross-Industry Standard Process for Data Mining) flow chart. It is a circular process with five main stages: Business Understanding, Data Understanding, Data Preparation, Modeling, and Evaluation. These stages are connected by arrows in a clockwise cycle. In the center of the cycle is a database icon labeled 'Data'. There are also feedback loops from the later stages back to the earlier ones. The entire process is enclosed in a circular border with a share icon on the right.