studocu

College Enquiry Chatbot Report v12

COMPUTER SCIENCE AND ENGINEERING (Andhra University)

Scan to open on Studocu

Studocu is not sponsored or endorsed by any college or university
Downloaded by HASSAN ALI (hxn112233@gmail.com)

IMPLEMENTING A COLLEGE ENQUIRY CHATBOT

A Project

Presented to the faculty of the Department of Computer Science

California State University, Sacramento

Submitted in partial satisfaction of
the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

by

Ujaliben Kalpesh Bavishi

SPRING
2019

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

© 2019

Ujaliben Kalpesh Bavishi

ALL RIGHTS RESERVED

Downloaded by HASSAN ALI (hxn112233@gmail.com)

IMPLEMENTING A COLLEGE ENQUIRY CHATBOT

A Project

by

Ujaliben Kalpesh Bavishi

Approved by:

_____, Committee Chair
Dr. Pinar Muyan-Ozcelik

_____, Second Reader
Dr. V. Scott Gordon

_____
Date

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

Student: Ujaliben Kalpesh Bavishi

I certify that this student has met the requirements for format contained the University format manual, and that this project is suitable for shelving in the Library and credit is to be awarded for the project.

_____, Graduate Coordinator _____
Dr. Jinsong Ouyang Date

Department of Computer Science

Downloaded by HASSAN ALI (hxn112233@gmail.com)

Abstract

of

IMPLEMENTING A COLLEGE ENQUIRY CHATBOT

by

Ujaliben Kalpesh Bavishi

This project is focusing on creating a chatbot to be used by students to get their queries responded easily from the college website. The College Enquiry Chatbot has the capacity to make friendly conversations; respond the course and faculty details; give the link for the academic calendar; answer the frequently asked questions; calculate the fees based on the student's input; and give the timings, address, contacts, and events information of the departments like Union, Library, IPGE, and AIRC. To build the chatbot, Microsoft Azure bot service as well as Microsoft cognitive services, namely, Text Analytics, LUIS, and QnA Maker are used.

Most of the existing chatbots lack empathy and fail to accommodate anything outside of the script. In order to address these problems, the College Enquiry Chatbot extends the implementation of the current chatbots by adding sentiment analysis and active learning. Although, sentimental analysis correctly recognizes the user's query as positive, negative and neutral, the system was partially successful in adding empathy to the chatbot. It is because the system requires more rigorous training data to handle all queries which are off-script. However, for such queries, active learning helps to improve the chatbot

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

performance since it correctly understands the user's questions, asks clarifying question, and then retrains the system to give the response what the user intends to get.

The future work include training the chatbot with more varied data; increasing the scope of the chatbot by adding a speech recognition feature so that users can speak to get responses; and including integration with multiple channels such as phone call, SMS, and various social media platforms.

_____ Committee Chair
Dr. Pinar Muyan-Ozcelik

_____
Date

Downloaded by HASSAN ALI (hxn112233@gmail.com)

ACKNOWLEDGEMENTS

I would like to express my gratitude towards Dr. Pinar Muyan-Ozcelik who has been a precious help in my voyage of finishing this undertaking. Without her assistance, direction, and support, making this project a reality would have been unimaginable. Her opportune criticism was tremendously essential for me and helped me greatly to reach the targets set for the project tasks. Likewise, I would also like to express gratitude toward Dr. V. Scott Gordon for his commitment and supporting this project as my second reader. In conclusion, I would like to thank the personnel and staff of the Department of Computer Science Engineering at California State University, Sacramento, for their ceaseless regulatory help.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

TABLE OF CONTENTS

Page

Downloaded by HASSAN ALI (hxn112233@gmail.com)

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

Downloaded by HASSAN ALI (hxn112233@gmail.com)

LIST OF FIGURES

Figures Page

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

Downloaded by HASSAN ALI (hxn112233@gmail.com)

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

1 INTRODUCTION AND MOTIVATION

This project is focusing on creating a chatbot to be used by students to get their queries responded easily from the college website. A chatbot is a program which can do real conversations with textual and/or auditory methods [1]. Using Artificial Intelligence (AI), chatbots can simulate human conversations. There are two categories of chatbots. One category is command based chatbots where chatbots rely on a databank of replies and heuristics. The user must be very specific while asking the questions so that the bot can answer. Hence, these bots can answer limited set of questions and cannot perform function outside of the code. The other category is chatbots based on AI or machine learning algorithms, these bots can answer ambiguous questions which means the user do not have to be specific while asking questions. Thus, these bots create replies for the user's queries using Natural Language Processing (NLP).

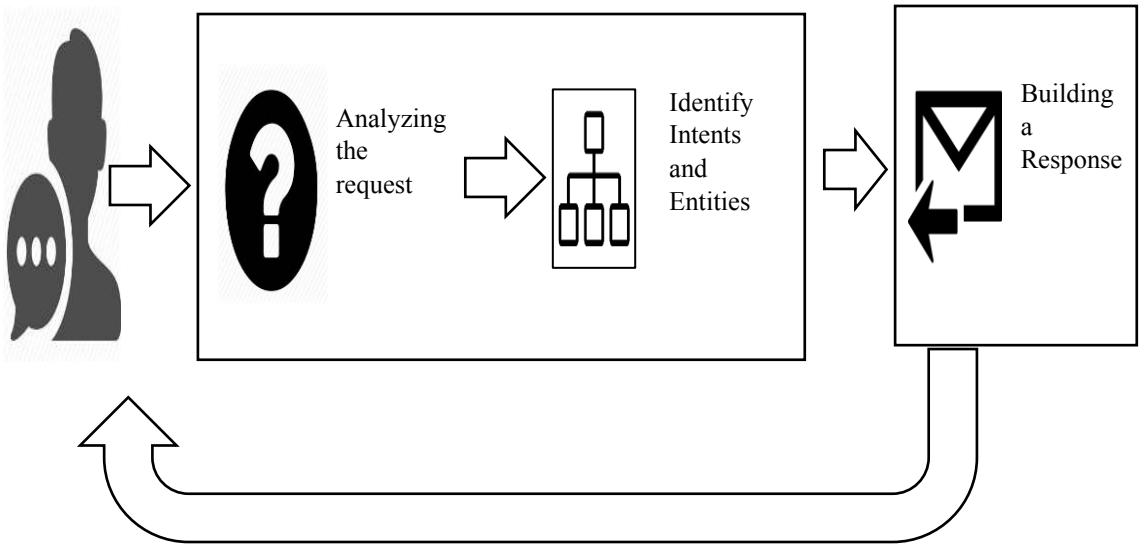


Figure 1: How a Chatbot Works [2]

Downloaded by HASSAN ALI (hxn112233@gmail.com)

Figure 1 shows how a chatbot works. Whenever a user asks any query, the bot will first analyze the request, then identifies intents and entities, builds a response and sends it back to the user. Now, intents mean intention of the query and entity means details of that query. For example, if a student wants to know the office hours of a faculty then the intent will be office hours and entity will be name of the faculty in this case.

AI-powered chatbots are motivated by the need of traditional websites to provide a chat facility where a bot is required to be able to chat with user and solve queries. When live agent can handle only two to three operations at a time, chatbots can operate without an upper limit which really scales up the operations. Also, if any school or business is receiving lots of queries, having a chatbot on a website takes off the load from support team. Having a chatbot clearly improves the response rate compared to human support team. In addition, since millennials prefer live chats over a phone call, they find a chatbot, which provide a highly interactive marketing platform, very attractive. Furthermore, a chatbot can automate the repetitive tasks. There can be some scenarios where a business or school receives same queries in a day for many times and support team must respond to each query repetitively. Lastly, the most important advantage of having a chatbot is that it is available 24/7. No matter what time it is, a user can get a query solved. All these advantages of a chatbot constitute the motivation to implement a College Enquiry Chatbot.

Before implementing College Enquiry Chatbot, various existing chatbots were reviewed such as Amazon Shopping App [3], Alexa [4], Bank of America (Erica bot) [5]

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

and CNN news bot [6]. In order to understand the requirement of a chatbot, consider an example of Amazon Shopping App. In this app, when a customer buys an item, he/she does not have any information about how to return the item. To get this information, the customer must call and wait to talk to customer representative for a long time. However, this whole process is tedious for a customer. Hence, Amazon created a chatbot to answer simple queries of customers.

Similarly, the College Enquiry Chatbot is designed to help students to get their queries solved on a fingertip. The most essential downside I found while utilizing the previously mentioned chatbots is absence of personality and conversational flow. As Storman [7] suggests, the CNN chatbot neglects to give sympathy and effortlessness. To be efficient, the chatbot must have the capacity to relate and associate itself with the user. For example, a discussion with the CNN chatbot is depicted in the article where at whatever point a user says anything aside from news or any current alternatives, it answers with the news comprising of those words and toward the end it says "Not sure I understand what you're looking for. Try again or pick one of the options below." and afterward it gives the choices as programmed. This implies, although it conveys the news proficiently, the CNN bot needs compassion.

Solution to this problem is described by Rahman et al. [8]. This study proposes that “there is a need to understand and consider the stability, scalability, and flexibility issues along with high level of intention of a human language”. Hence, for implementing a chatbot that handles complicated queries, the sentiment analysis is incorporated into College

Downloaded by HASSAN ALI (hxn112233@gmail.com)

Enquiry Chatbot. Sentiment analysis aims to obtain writer's feelings expressed in positive, negative or neutral comments. Based on sentiment analysis, the bot is trained to have empathy while answering to the user. For example, if a user says "I am sad today." then bot should reply to it with some empathy like "I'm sorry to hear that, how can I help you today?" and not just reply the standard message like "Sorry did not understand your question."

Another downside which was found during a research on chatbot is that bots are created in such way that they follow a specific route and mostly all of them fails to satisfy anything outside of the previously defined scripts. This means that if they are not part of a predefined scripts, a significant number of the bots will fail in understanding even the most fundamental kind of queries, which results in a repeating and horrendous experience. To resolve this issue, active learning can be introduced to the system to make probabilistic assessments and provide autonomous responses to the users [9]. Active learning is an algorithm which interactively queries user to obtain the desired output. Whenever a user asks anything which is outside of the script the chatbot will ask questions to the user by giving two to three options and based on the user's input, the bot returns the answer to that query. This whole learning process is called as active learning.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

2 SYSTEM DESIGN

System design of College Enquiry Chatbot consists of integration of multiple technologies. This system makes use of NodeJS for backend and ReactJS for frontend. Microsoft Azure provides services like Cognitive Service, Bot Service [10], and Web Apps. Cognitive services consist of Text Analytics, LUIS [11], and QnA Maker [12]. The bot service is used to create the bot application on Azure. In addition, Web Apps are used to host the application on Microsoft Azure. System uses Mongo DB to store the conversations history.

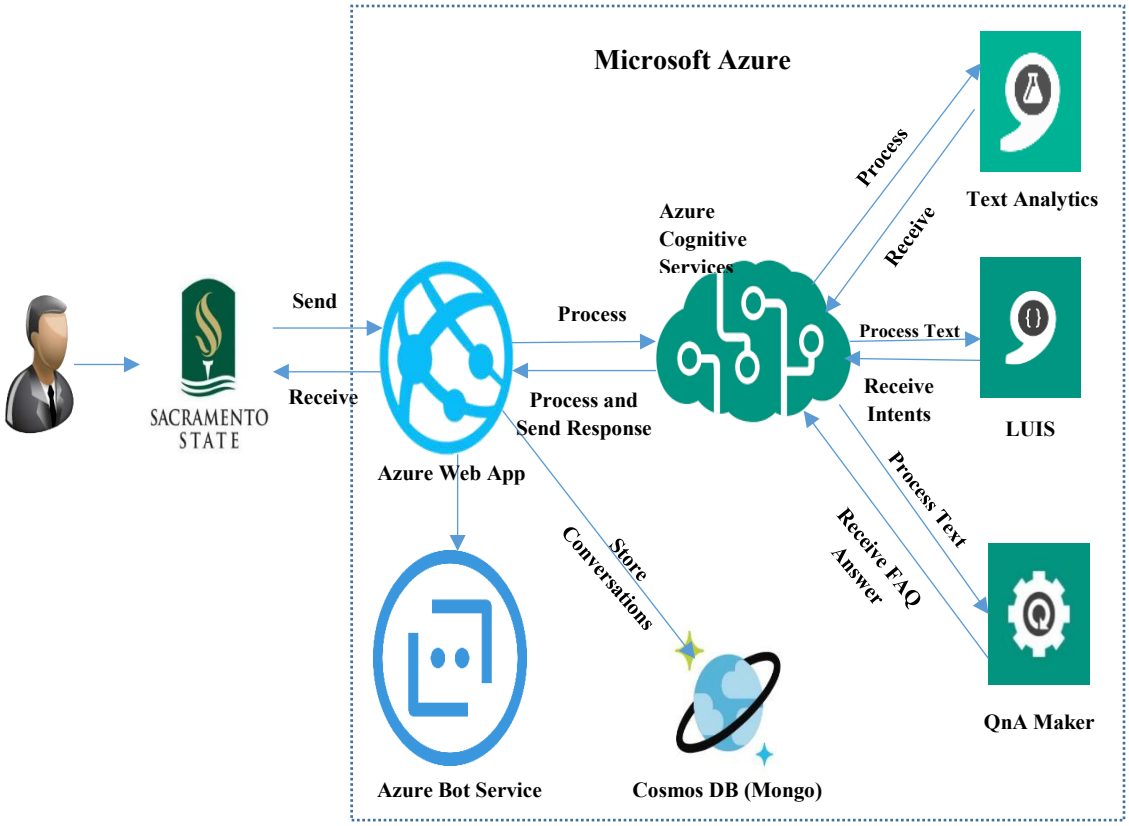


Figure 2: System Design

Downloaded by HASSAN ALI (hxn112233@gmail.com)

2.1 Microsoft Azure

Microsoft Azure provides cloud services to build, manage, and deploy applications on a cloud network which helps organization to meet their business needs using their favorite tools and frameworks. College Enquiry Chatbot uses all the services shown in Figure 2 which are provided by Microsoft Azure.

2.1.1 Azure Bot Service

Azure bot service is a service provided by Microsoft Azure which builds and artificial intelligence (AI) chatbot. It offers the ability to add intelligent agents that can do conversations with users without having to commit the resources to develop one's own AI [13].

2.1.2 Azure Web App

I have used Azure web app to host my bot application build by Azure bot service. Azure Web App are the web applications which are hosted on Microsoft Azure cloud without any programming language barrier. Since it is hosted on the cloud, infrastructure maintenance is not needed to host the applications. It also provides various features like auto scaling, automated deployment and also supports both Windows and Linux environments [14].

2.1.3 Azure Cognitive Services

Azure Cognitive services are used to add intelligence to the bots by adding features such as sentiment analysis and language understanding which help in analyzing user's queries [15]. These services can be added to an existing bot by adding service calls to the

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

Microsoft's SDK to get the desired results. College Enquiry Chatbot utilizes Text Analytics, LUIS, and QnA Maker services which are part of cognitive services and use natural language processing (NLP) for different purposes.

2.1.3.1 Text Analytics

Text Analytics is used to perform sentiment analysis on a user's queries. As shown in Figure 2, a text is sent to text analytics and it returns the sentiment of that text.

2.1.3.2 LUIS

"LUIS stands for Language Understanding Intelligent Services which aims at creating cloud-based machine learning language understanding (LU) models specific to an application and without machine learning expertise" [16]. To implement a College Enquiry Chatbot, I have first created all the possible intents and entities using LUIS tool. Based on these intents and entities, LUIS have built and train a LU deep learning model. Then, the created model needs to be pushed to the endpoint on the cloud. The queries from the user are passed to the endpoint to get the recognized intent and entities in JSON format. Based on various flows, response will be sent back to the user.

Active learning is also implemented using LUIS as shown in Figure 3. LUIS detects the user's query (utterances) in terms of intents and entities. It sends back the response to the service code where those utterances are examined based on the confidence score. Then, College Enquiry Chatbot labels these utterances, re-trains, and publishes the LU model [17]. Another method of active learning is also implemented which involves using user's response to re-train the model. The endpoint provided by LUIS responds with intents,

Downloaded by HASSAN ALI (hxn112233@gmail.com)

entities, and a corresponding confidence score. A threshold is maintained based on the confidence score and correspondingly prompts are provided to the user. When the users respond to it, the system keeps track of the responses and uses it to re-train the model.

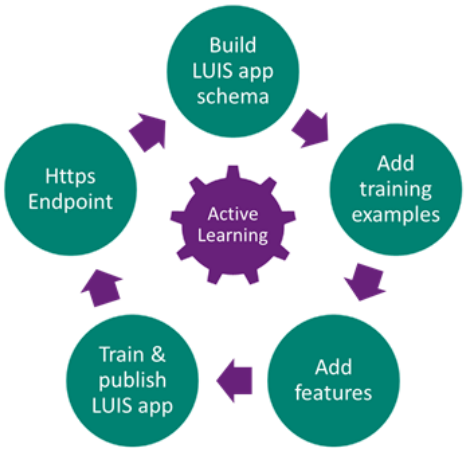


Figure 3: Active Learning Process [16]

2.1.3.3 QnA Maker

For answering simpler queries, instead of detecting intents and entities, College Enquiry Chatbot uses QnA Maker for structuring answers from a semi-structured document like FAQs. This web-based service is useful in incorporating multiple FAQs from an URL, structured documents, product manuals, or editorial contents; and building the LU model for natural language processing.

2.1.4 Cosmos DB (Mongo)

Cosmos DB is a database service provided by Microsoft Azure. College Enquiry Chatbot uses this database service to store my conversational history.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

3 IMPLEMENTATION DETAILS

In this section of the report, step by step demonstration of how to setup the bot, LUIS (for NLP), and QnA Maker as well as explanation of service code which includes the implementation of the Active Learning are provided. In addition, setting up databases to store the conversation history is explained. All these steps serve as a template to get a basic bot deployed and published on Azure cloud.

The followings are couple of the prerequisites that are needed in order to get started:

- Microsoft Azure Student Account
- Installation of Git (https://git-scm.com/download/win) and NodeJS (https://nodejs.org/en/download/) on Windows.

3.1 Azure Bot Setup

Open https://portal.azure.com as shown in Figure 4 and follow the steps explained in the following subsections.

Downloaded by HASSAN ALI (hxn112233@gmail.com)

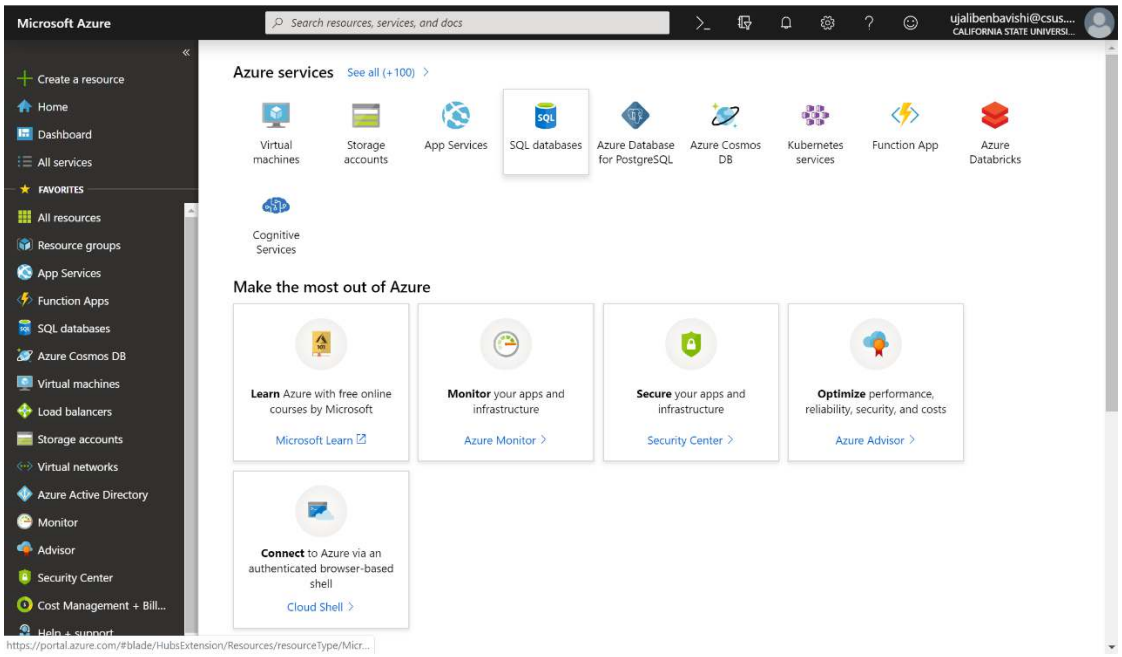


Figure 4: Azure Portal Home Page

3.1.1 Bot Creation

In the left side pane, click on “Create a Resource” and search for “Web App Bot”. As a next step click on “Create” to start creating a bot. Fill up all the required fields as shown in Figure 5.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

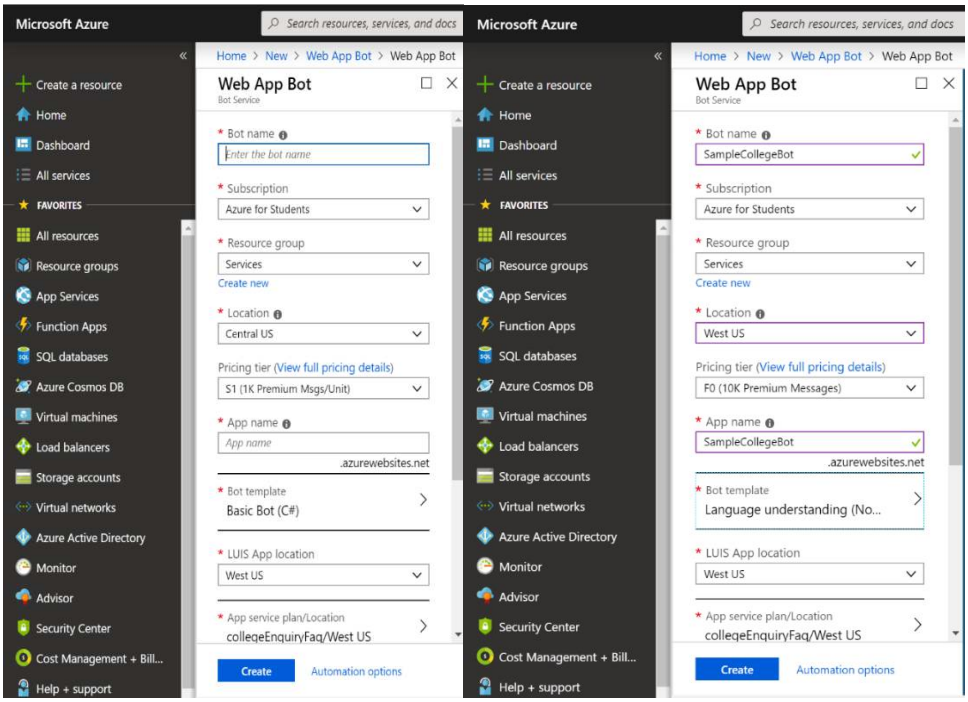


Figure 5: Web App Bot Creation

3.1.2 Adding the DirectLine Channel

After the bot has been created, several channels can be configured to access the bot such as Skype, Webchat, Slack, Facebook, etc. In this project, DirectLine channel is used to communicate with the bot. Figures 6 and 7 show how to add a channel to a bot. After the channel is added, copy the secret key to be used in the service and click on “Done”.

Downloaded by HASSAN ALI (hxn112233@gmail.com)

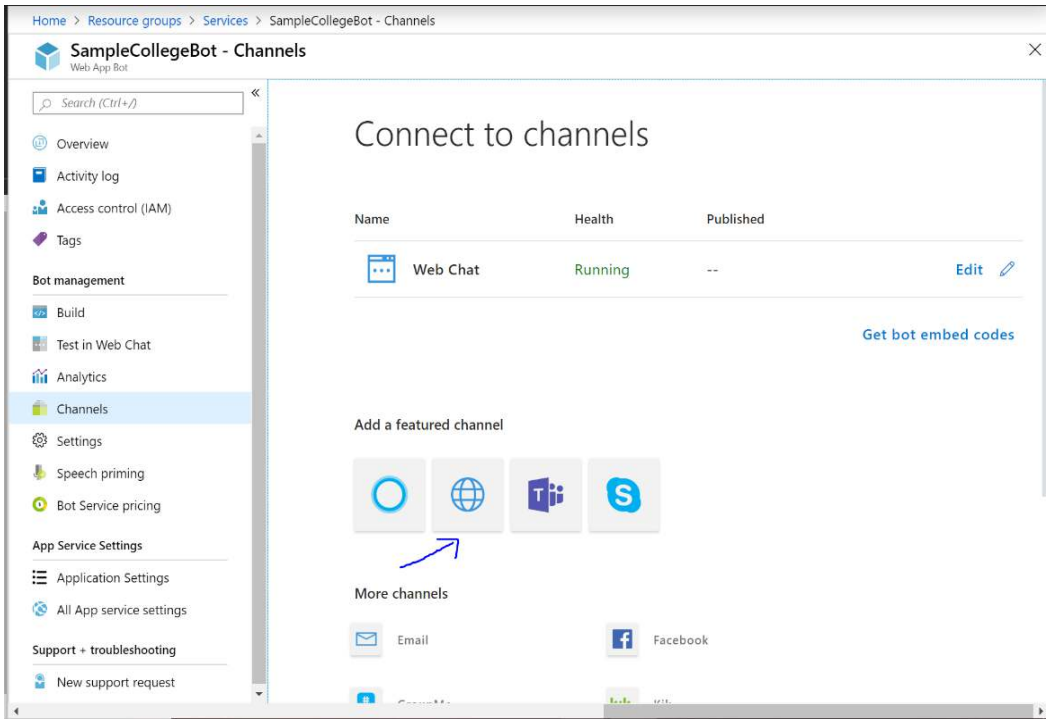


Figure 6: Channel Configuration Page of the Bot

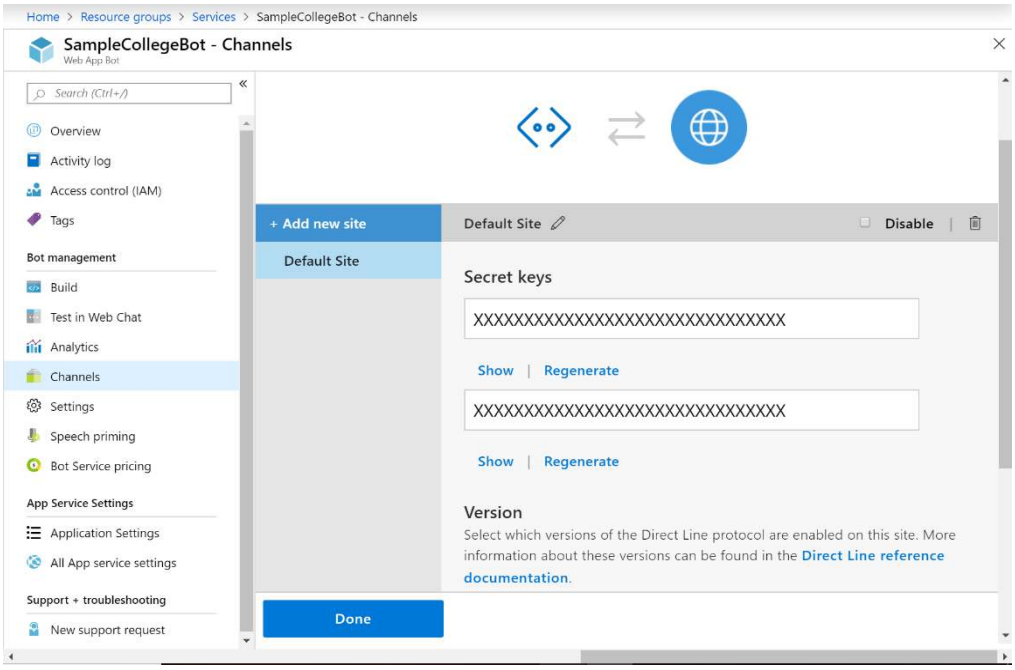


Figure 7: Configure DirectLine Channel for the Bot

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

3.1.3 Testing the Echo Bot

After the channel is configured, we can test the new bot with some preconfigured bot template where we can echo back whatever the user queries as shown in Figure 8.

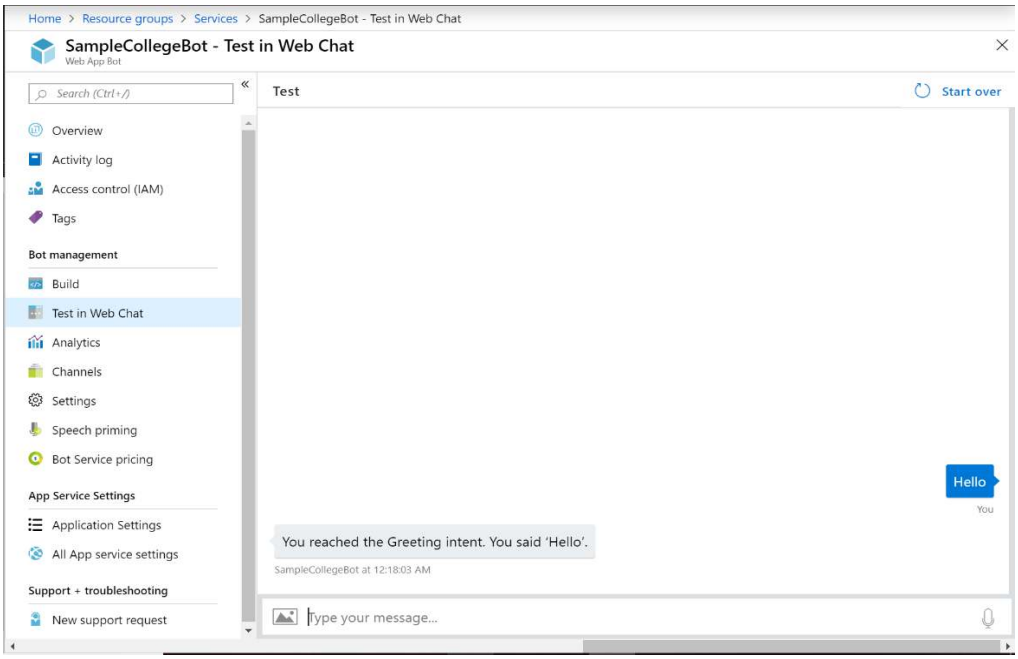


Figure 8: Web Chat Testing of the Created Bot

3.2 LUIS Setup

Go to https://www.luis.ai to setup the intents and entities to create a model and train the natural language processor.

3.2.1 LUIS App Creation

After visiting the website, sign in with the same Azure credentials, scroll down and click on the button “Create a LUIS app now” as shown in Figure 9. Please note that if a Web

Downloaded by HASSAN ALI (hxn112233@gmail.com)

App Bot is created with LUIS bot template, then a LUIS App will be created automatically with preconfigured intents and entities to work with.

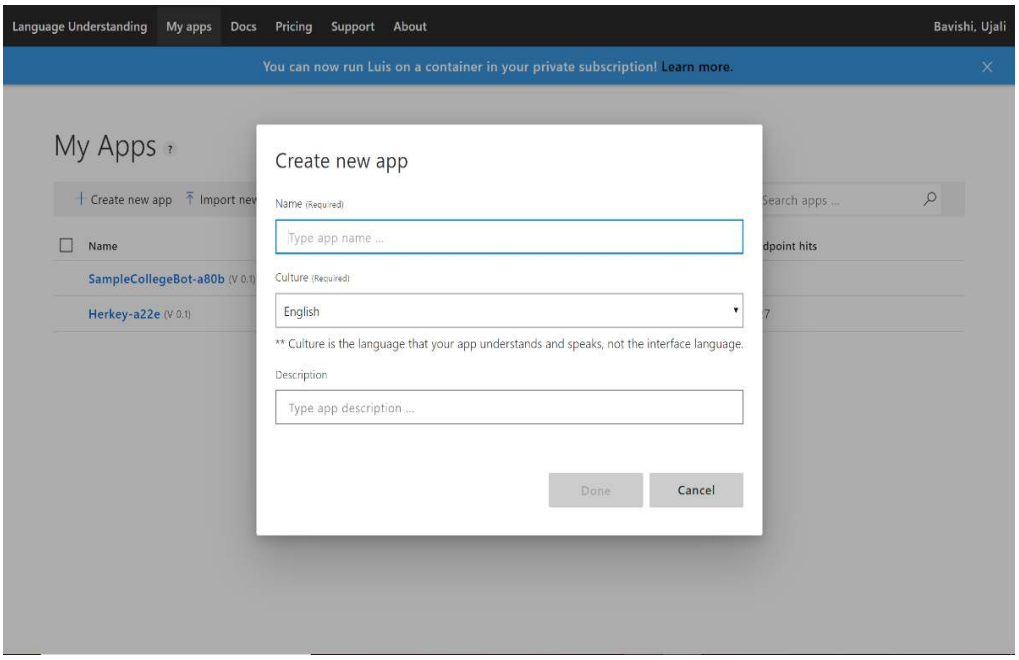


Figure 9: LUIS App Creation

3.2.2 Intents Creation

Go to created LUIS app and click on “Create new intent” and provide a specific intent name as shown in Figure 10. The next step would be to add training data to train the intent with all different types of utterances as shown in Figure 11.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

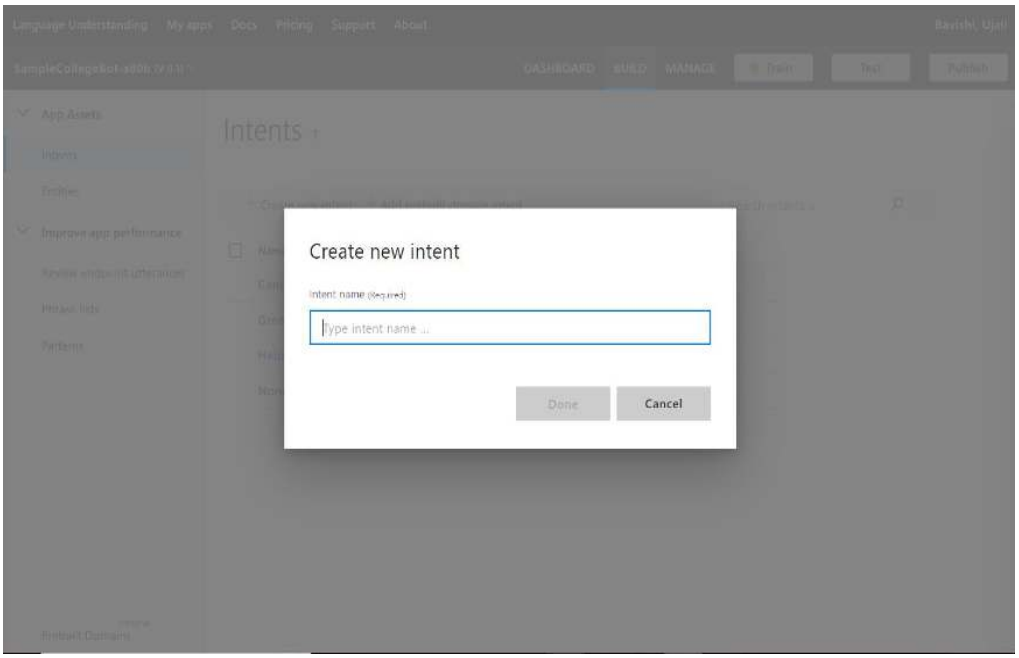


Figure 10: Intent Creation in LUIS

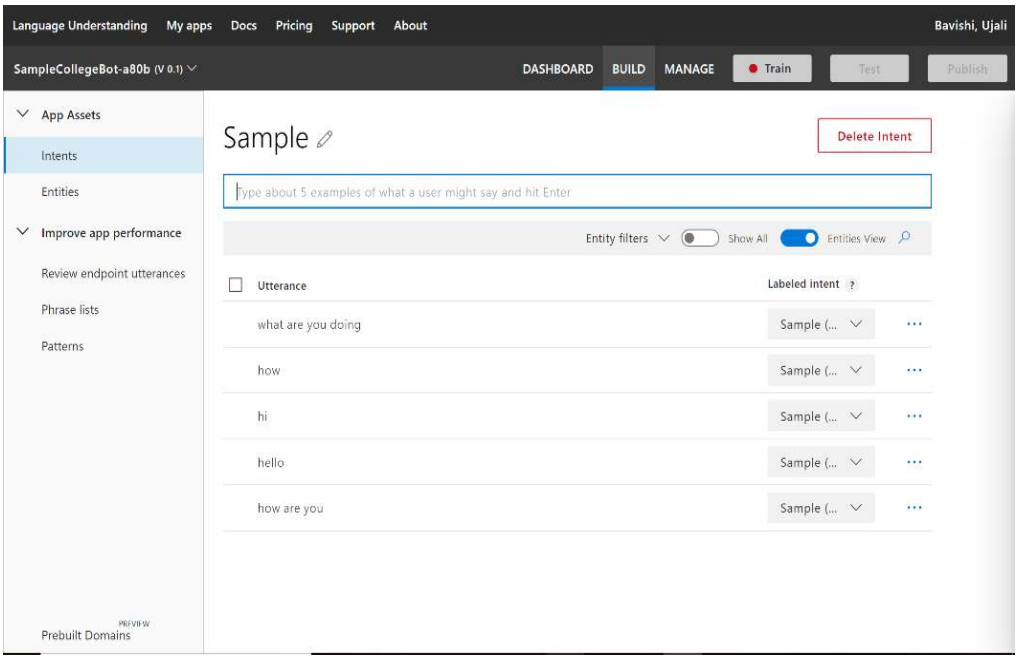


Figure 11: Adding Training Data to Created Intent

Downloaded by HASSAN ALI (hxn112233@gmail.com)

3.2.3 Entity Creation

In the left side pain, click on “entities” and then select “Create an Entity” and provide a name for the entity. In this project, a list type of entity is used since the professor and course values are stored in the entity sub list as shown in figures 12 and 13.

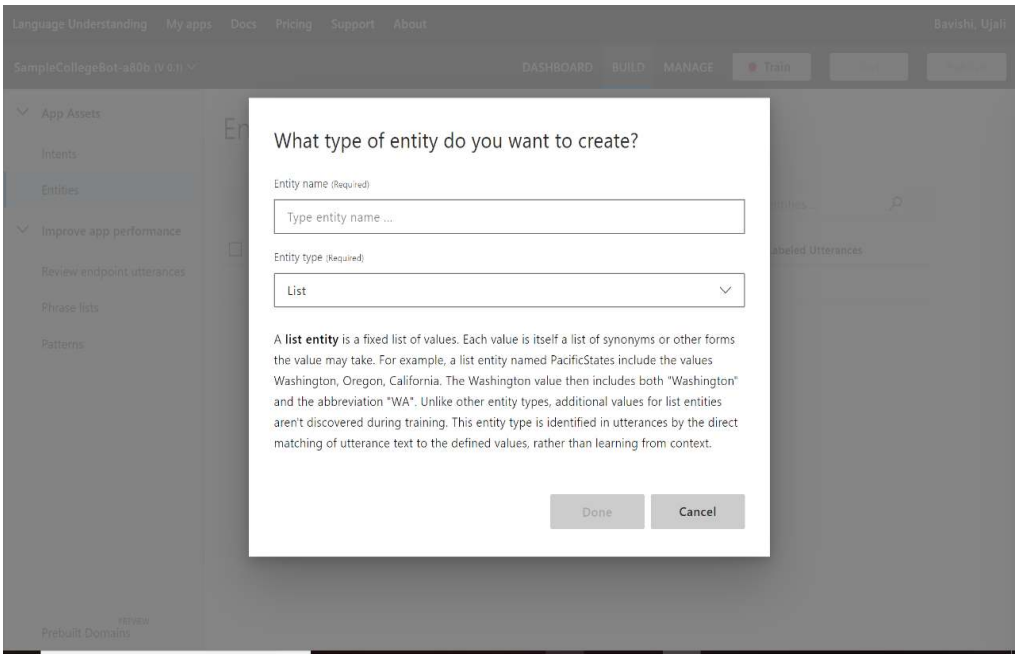


Figure 12: Entity Creation in LUIS - 1

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

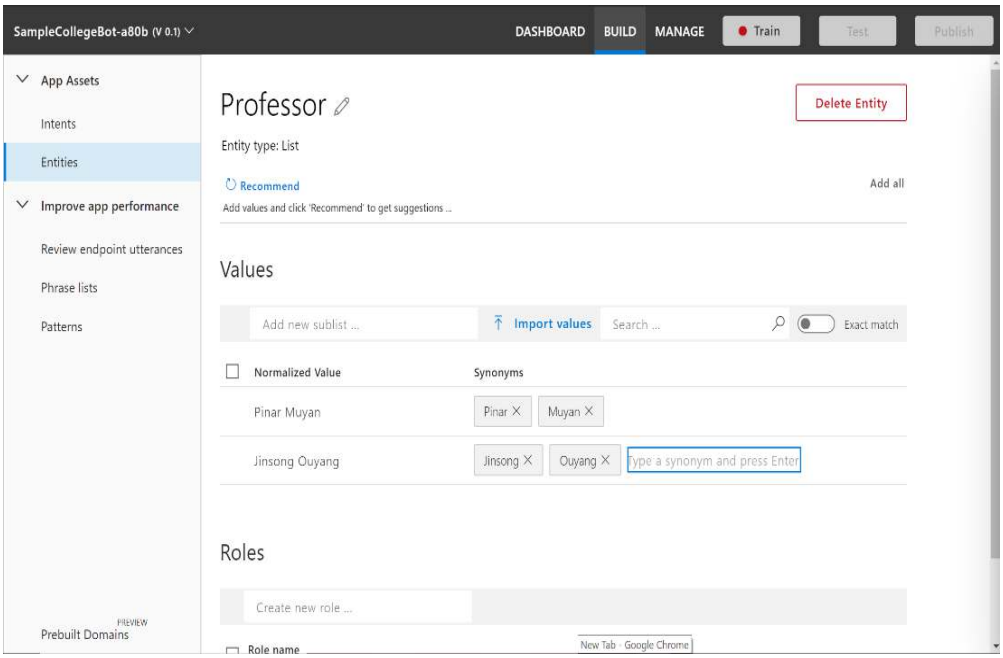


Figure 13: Entity Creation in LUIS - 2

3.2.4 Train and Publish the Model

After adding the intents and entities, the model is trained and published. After the model is published, REST API can be accessed from the service to access the LUIS endpoint to get the intent and entity from the user's query.

3.3 QnA Maker Setup

Go to https://www.qnamaker.ai to setup the questions and answers from the FAQs listed in any website and also to include some of the responses to be displayed depending on user's emotions based on some metadata.

Downloaded by HASSAN ALI (hxn112233@gmail.com)

3.3.1 QnA Maker App Creation

After visiting the website, sign in with the same Azure credentials, click on “Create a knowledge base” from the top navigation bar as shown in Figure 14. Then, follow the steps as shown in figures 15 and 16.

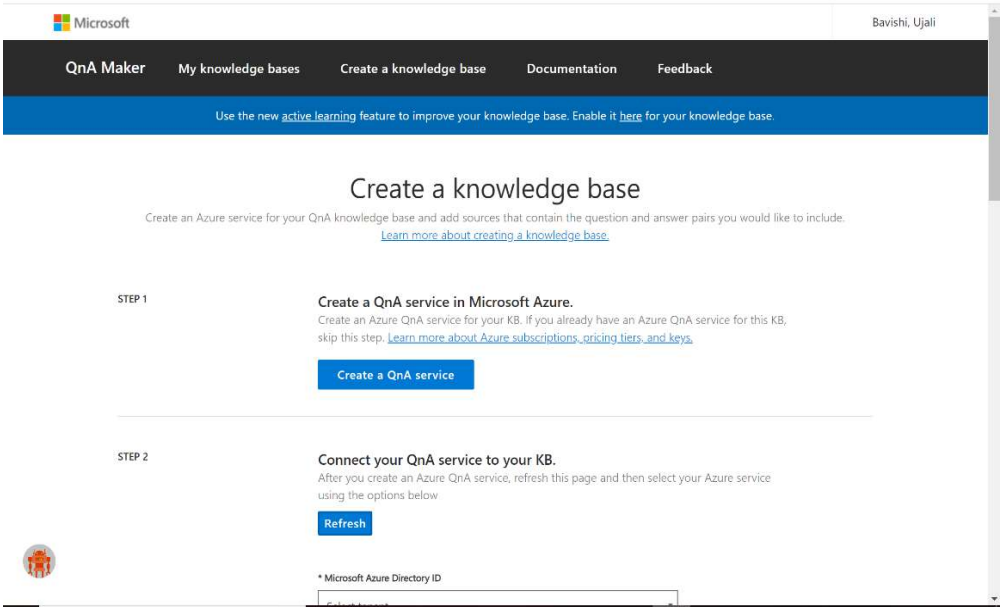


Figure 14: QnA Maker App Creation - 1

This document is available free of charge on studocu
Downloaded by HASSAN ALI (hxn112233@gmail.com)

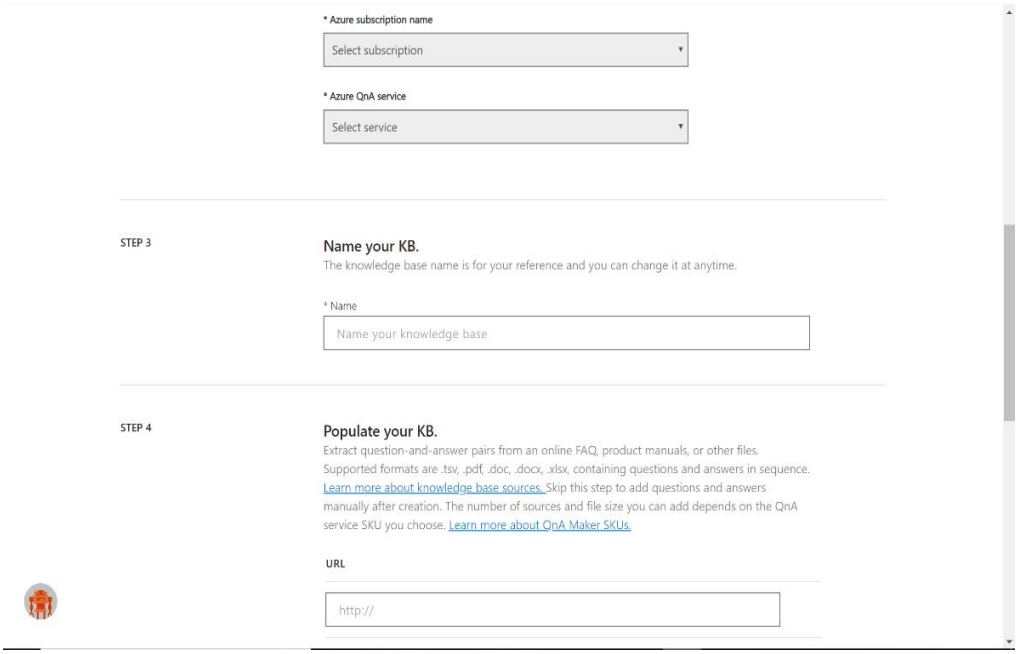


Figure 15: QnA Maker App Creation - 2

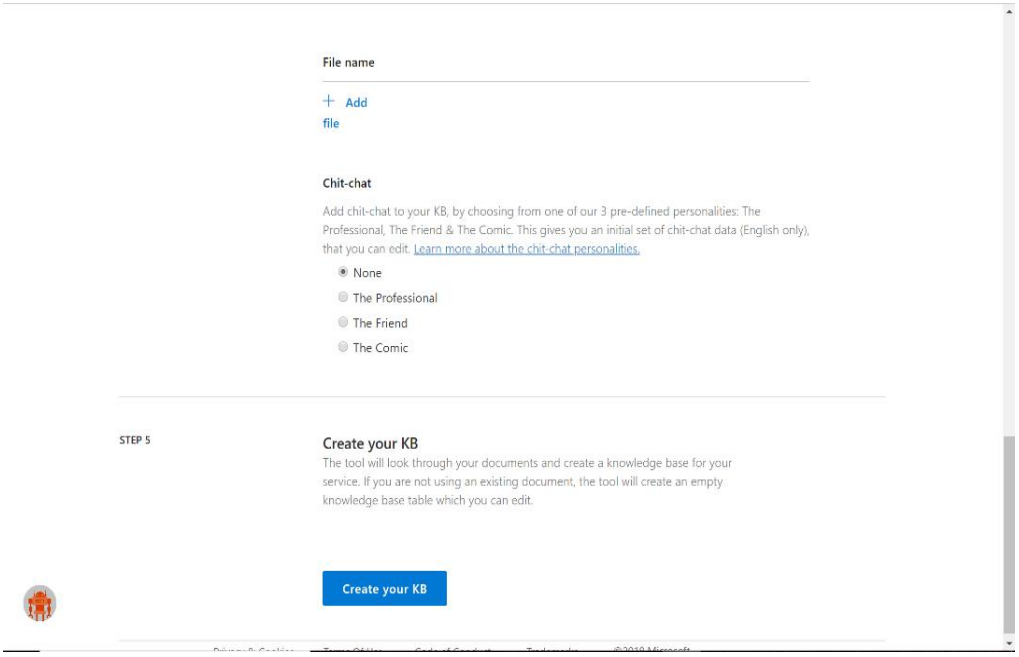


Figure 16: QnA Maker App Creation – 3

Downloaded by HASSAN ALI (hxn112233@gmail.com)

3.3.2 Add/Manage Knowledge base

Click on “Settings” from the top navigation and add FAQs URL from the CSUS website which are needed to be included in the knowledge base as shown in Figure 17

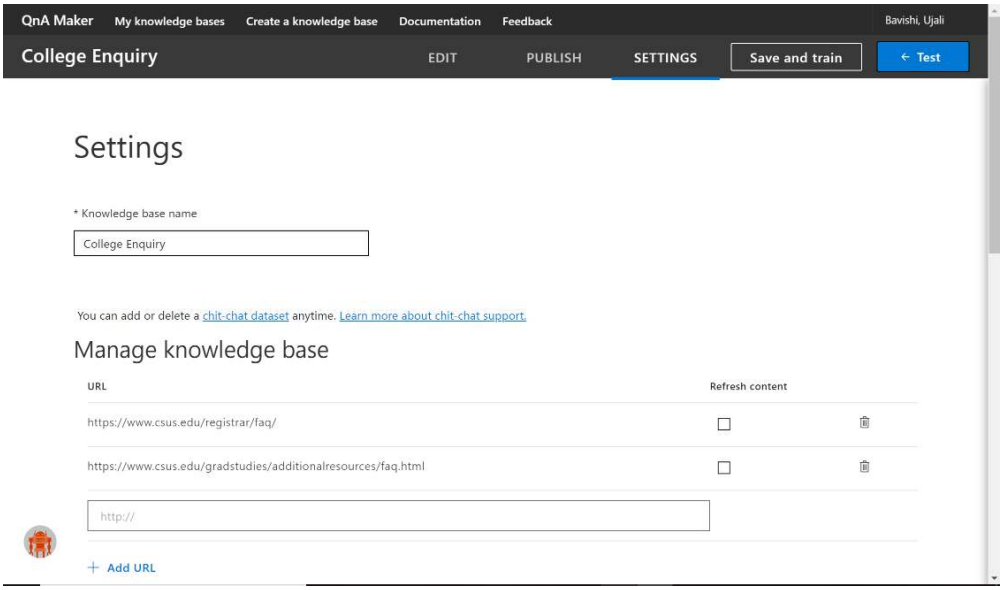


Figure 17: Knowledge Base Configuration Page

3.3.3 Train and Publish the Knowledge Base

After the knowledge bases are added, the next step is to train the app and publish to get a published endpoint to use the QnA Maker service.

3.4 Service Code Setup

I am using Azure DevOps to setup continuous deployment with a source control repository where the code will reside.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

3.4.1 Create Project, Repository, and Branches

Sign in to Azure DevOps using the Azure account, create a new project and setup a repository based on the needs as shown in figures 18-20.

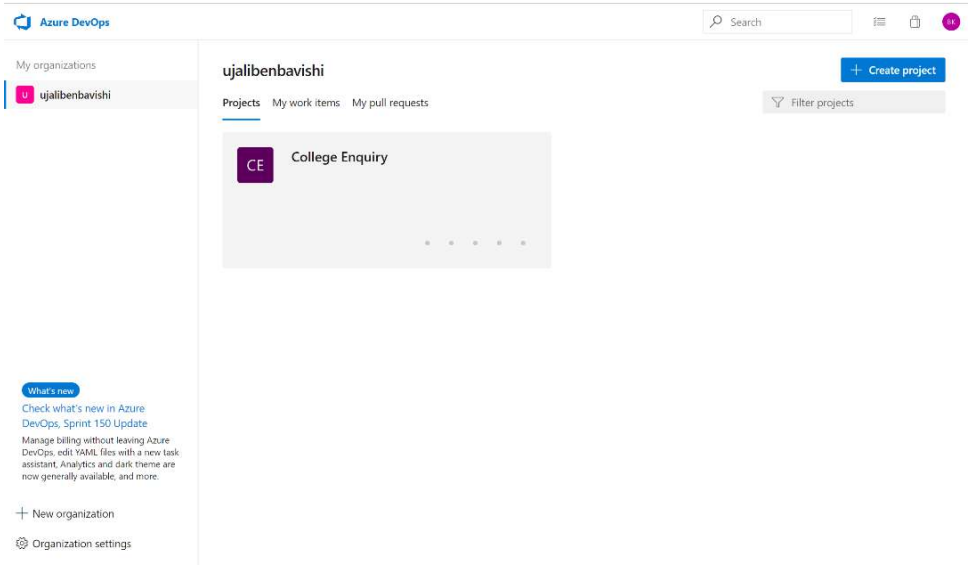


Figure 18: Azure DevOps Source Control Repo

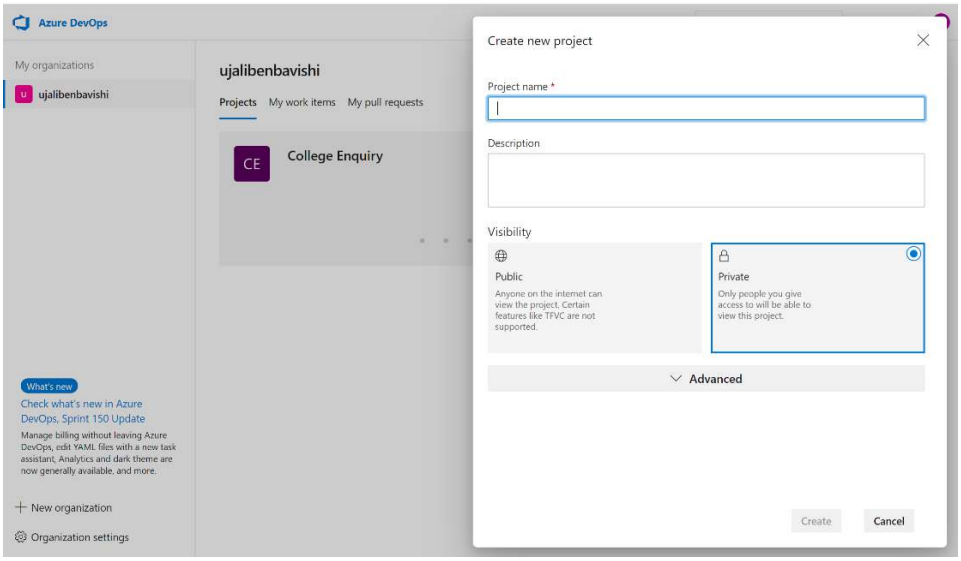


Figure 19: Project Creation on Azure DevOps

Downloaded by HASSAN ALI (hxn112233@gmail.com)

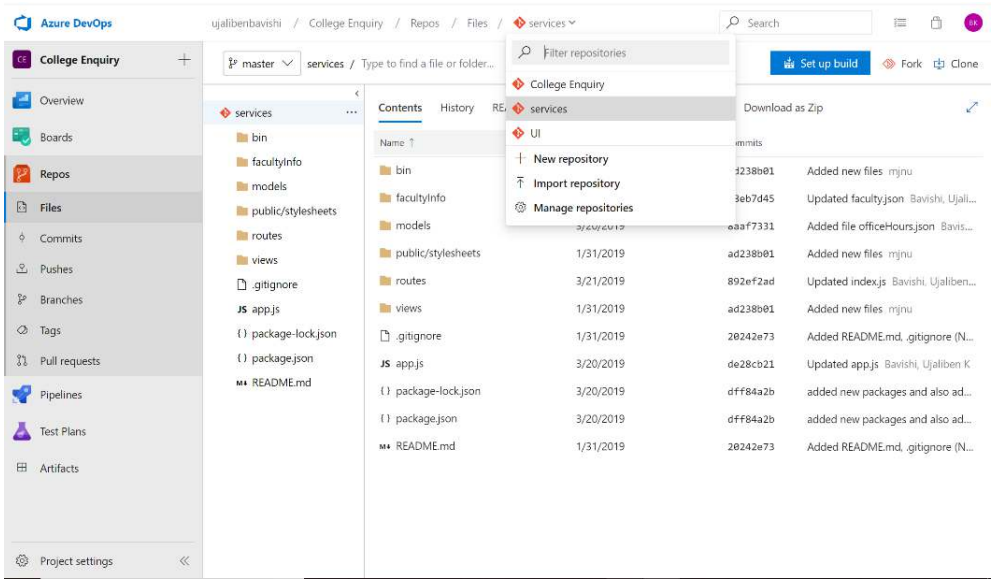


Figure 20: Repository and branch configuration in Azure DevOps Project

3.4.2 Sync Service Code with Azure DevOps

After the repository is setup, use Git clone functionality to clone the repo onto the local system and add all the files related to the service into the project. Finally, when a Git push is made from the same folder, all the files will be copied over to Azure DevOps.

3.5 Setup Continuous Deployment

As a last step, we need to configure the project created on Azure DevOps with the application created on Azure portal as shown in figures 21-24.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

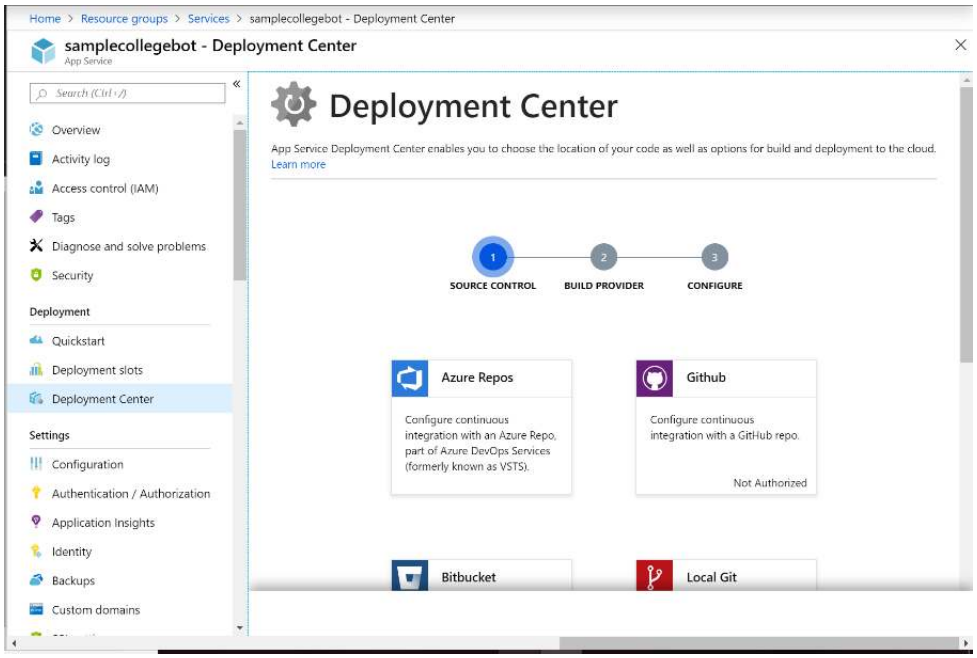


Figure 21: Continuous Deployment Configuration - 1

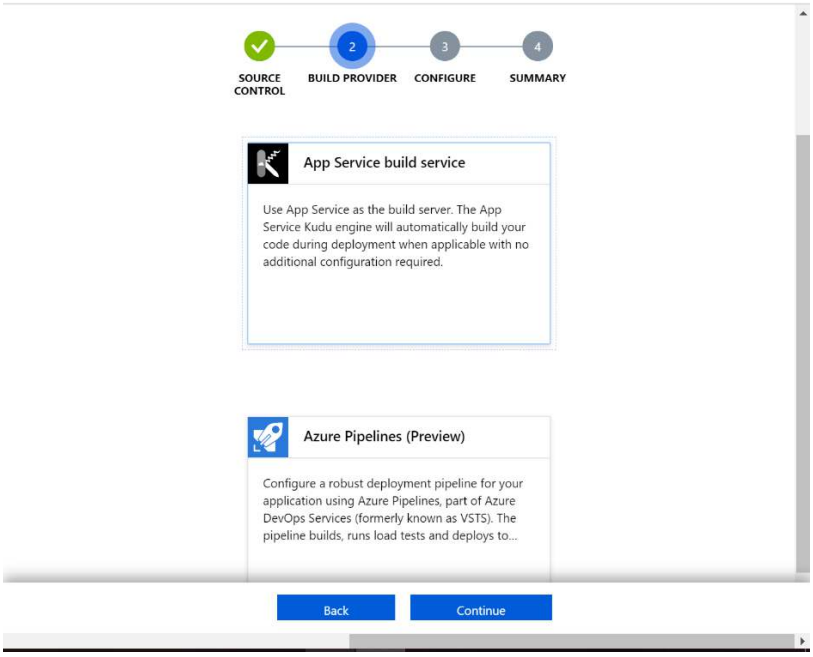


Figure 22: Continuous Deployment Configuration - 2

Downloaded by HASSAN ALI (hxn112233@gmail.com)

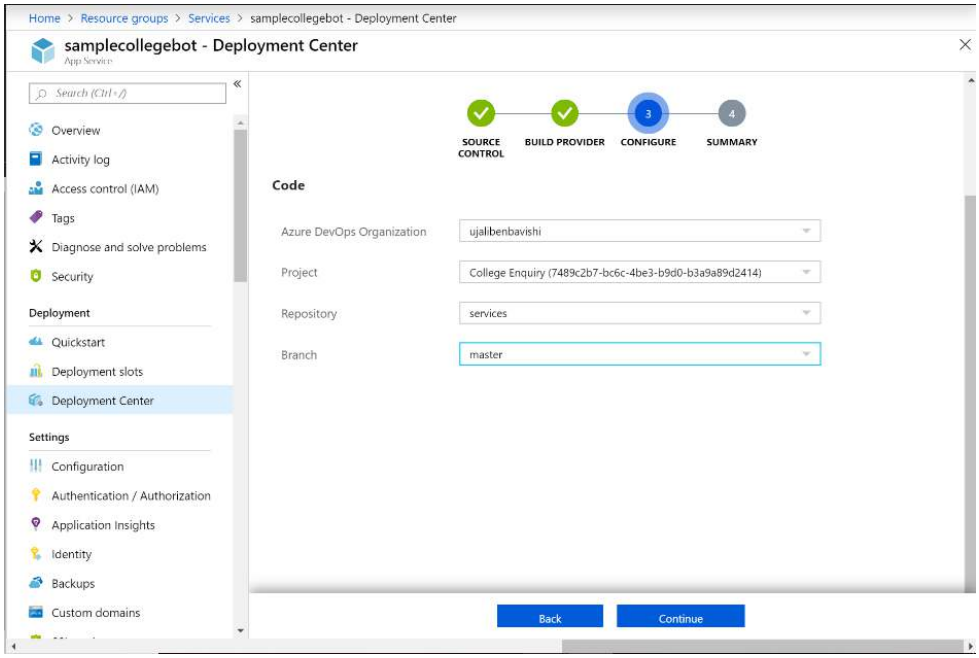


Figure 23: Continuous Deployment Configuration - 3

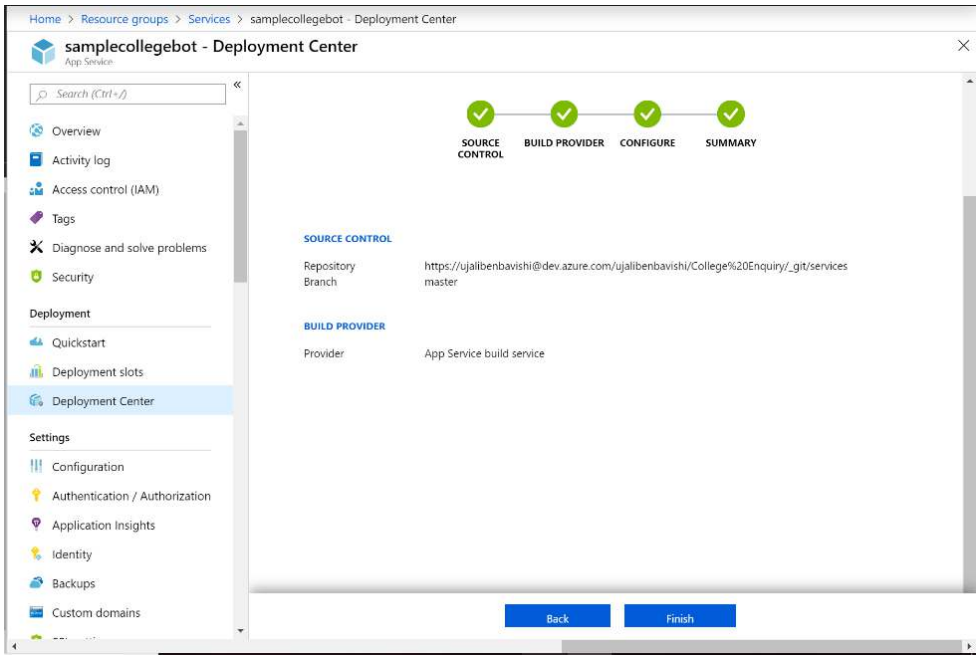


Figure 24: Continuous Deployment Configuration – 4

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

3.6 Service Code Snippets

In order to provide overview of the service code and some of the utility functions that are used by the bot, this section provides some screenshots. Figure 25 shows the overall structure of the code which utilizes NodeJS and ExpressJS frameworks.

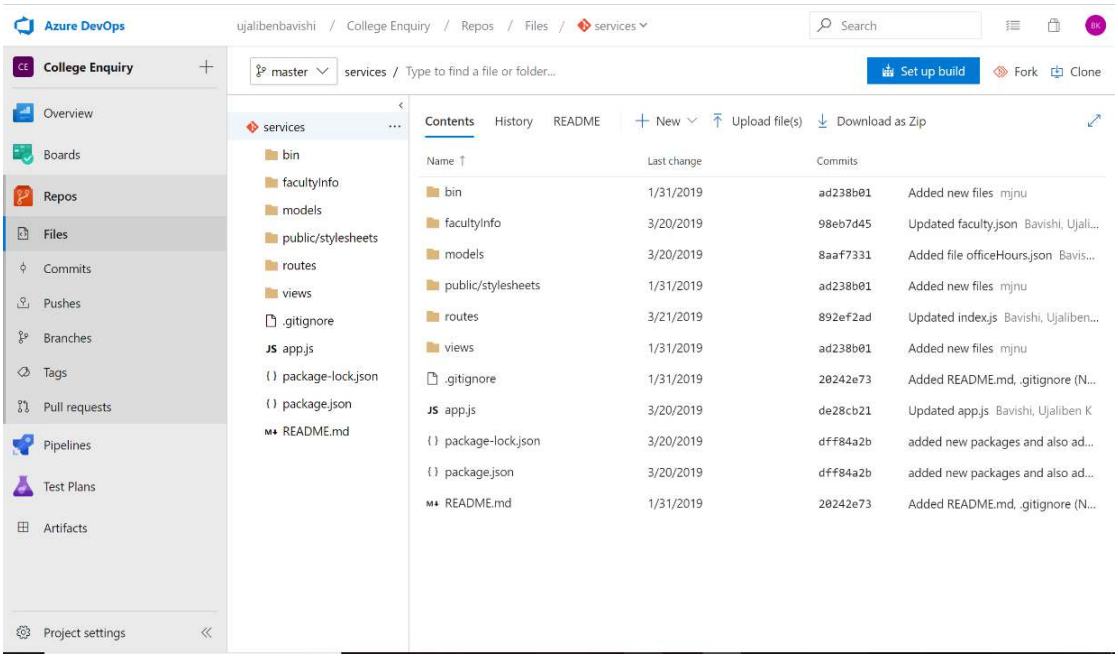


Figure 25: Code Structure Overview

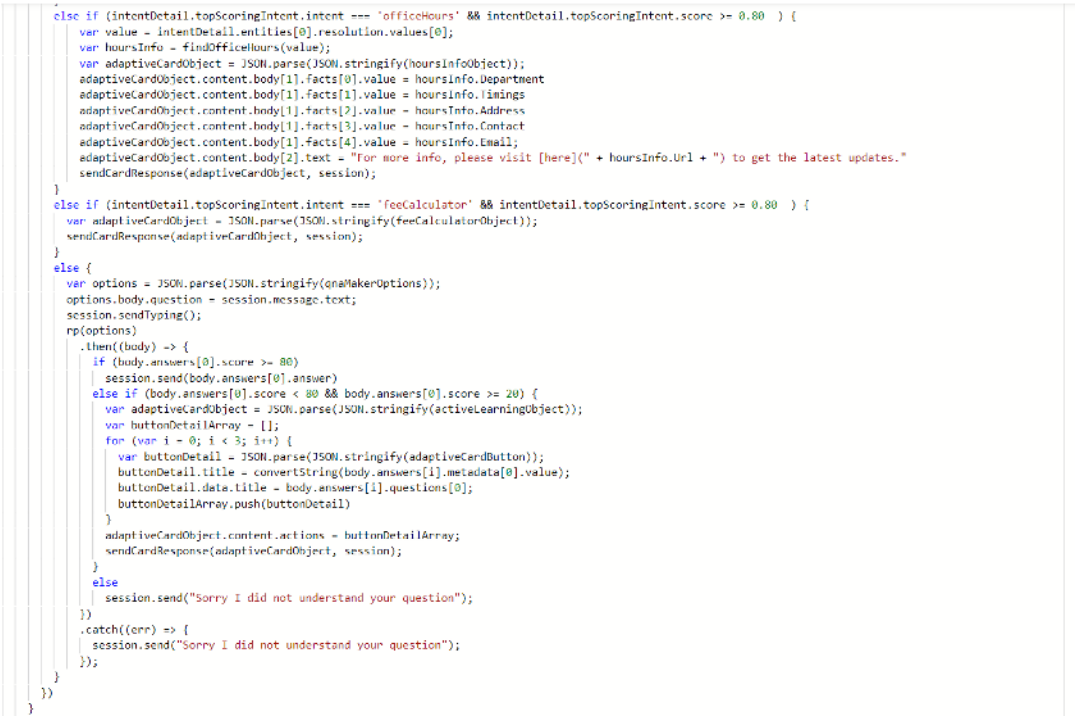
Figures 26 and 27 show various functionalities including calling LUIS REST API, intercepting bot messages to save the chat history to the database and presenting the logic to show feedback card used in active learning.

Downloaded by HASSAN ALI (hxn112233@gmail.com)

```
bot.use({
  botbuilder: function (session, next) {
    if (session.message && session.message.value) {
      var studentType = session.message.value.studentType;
      var credits = session.message.value.credits;
      var registrationFee = 0;
      var totalFee = 0;
      if (studentType === "UnderGraduate") {
        if (parseInt(credits) > 6)
          totalFee = 3655 + parseInt(credits) * 396;
        else
          totalFee = 2449 + parseInt(credits) * 396;
      }
      else if (studentType === "Graduate") {
        if (parseInt(credits) > 6)
          totalFee = 4372 + parseInt(credits) * 396;
        else
          totalFee = 2866 + parseInt(credits) * 396;
      }
      var resp = "Based on your input, the total fees would be approximately $" + totalFee + ".00";
      session.send(resp);
    }
    else if (session.message && session.message.text !== "") {
      getIntentOptions.url = LuisModelUrl.concat(session.message.text);
      rp(getIntentOptions).then((result) => {
        var intentDetail = JSON.parse(result);
        console.log(result);
        var timeStamp = moment(new Date()).tz("America/Los_Angeles").format("MM/DD/YYYY hh:mm:ss a");
        var updates = {};
        if (intentDetail.sentimentAnalysis) {
          if (Math.round(intentDetail.sentimentAnalysis.score) === 1) {
            updates = { $setOnInsert: { startTimeStamp: timeStamp }, $set: { endTimeStamp: timeStamp }, $inc: { positive: 1 }, $push: { conversations: { $each: [{ "use
          }
          else {
            updates = { $setOnInsert: { startTimeStamp: timeStamp }, $set: { endTimeStamp: timeStamp }, $inc: { negative: 1 }, $push: { conversations: { $each: [{ "use
          }
        }
        else {
          updates = { $setOnInsert: { startTimeStamp: timeStamp }, $set: { endTimeStamp: timeStamp }, $push: { conversations: { $each: [{ "user": session.message.text,
        }
        insertIntoDb(session.message.address.conversation.id, updates)

        if (intentDetail.topScoringIntent.intent === 'courseInfo' && intentDetail.topScoringIntent.score >= 0.80  ) {
```

Figure 26: LUIS and Active Learning Logic - 1



```
    else if (intentDetail.topScoringIntent.intent === 'officeHours' && intentDetail.topScoringIntent.score >= 0.80  ) {
      var value = intentDetail.entities[0].resolution.values[0];
      var hoursInfo = findOfficeHours(value);
      var adaptiveCardObject = JSON.parse(JSON.stringify(hoursInfoObject));
      adaptiveCardObject.content.body[1].facts[0].value = hoursInfo.Department
      adaptiveCardObject.content.body[1].facts[1].value = hoursInfo.Timings
      adaptiveCardObject.content.body[1].facts[2].value = hoursInfo.Address
      adaptiveCardObject.content.body[1].facts[3].value = hoursInfo.Contact
      adaptiveCardObject.content.body[1].facts[4].value = hoursInfo.Email;
      adaptiveCardObject.content.body[2].text = "For more info, please visit [here](" + hoursInfo.Url + ") to get the latest updates."
      sendCardResponse(adaptiveCardObject, session);
    }
    else if (intentDetail.topScoringIntent.intent === 'feeCalculator' && intentDetail.topScoringIntent.score >= 0.80  ) {
      var adaptiveCardObject = JSON.parse(JSON.stringify(feeCalculatorObject));
      sendCardResponse(adaptiveCardObject, session);
    }
    else {
      var options = JSON.parse(JSON.stringify(qnaMakerOptions));
      options.body.question = session.message.text;
      session.sendTyping();
      rp(options)
        .then((body) => {
          if (body.answers[0].score >= 80)
            session.send(body.answers[0].answer)
          else if (body.answers[0].score < 80 && body.answers[0].score >= 20) {
            var adaptiveCardObject = JSON.parse(JSON.stringify(activeLearningObject));
            var buttonDetailArray = [];
            for (var i = 0; i < 3; i++) {
              var buttonDetail = JSON.parse(JSON.stringify(adaptiveCardButton));
              buttonDetail.title = convertString(body.answers[i].metadata[0].value);
              buttonDetail.data.title = body.answers[i].questions[0];
              buttonDetailArray.push(buttonDetail)
            }
            adaptiveCardObject.content.actions = buttonDetailArray;
            sendCardResponse(adaptiveCardObject, session);
          }
          else
            session.send("Sorry I did not understand your question");
        })
        .catch((err) => {
          session.send("Sorry I did not understand your question");
        });
    }
  })
}
```

Figure 27: LUIS and Active Learning Logic - 2

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

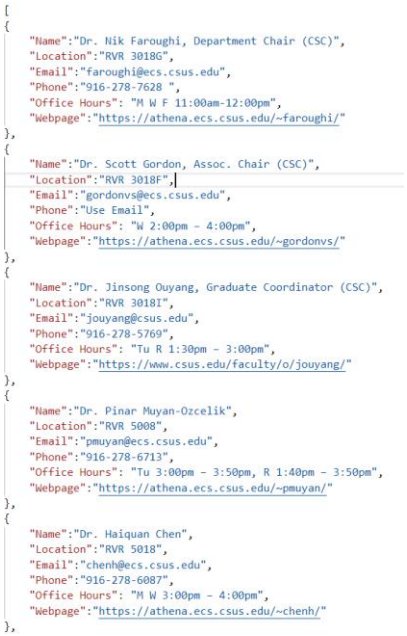
Figure 28 shows a sample file which contains basic information related to the course like course id, name, information, prerequisites, professor's name and URL of the course.

```
[
    {
        "Course Id": "CSC201",
        "Course Name": "Programming Language Principles",
        "Course Information": "Notations for the specification of programming language syntax and semantics; attribute, translational, operational, axiomatic, algebra
        "Prerequisites of Course": "Fully classified graduate status in Computer Science or Software Engineering",
        "Units": "3",
        "Professor for the course": "Dr. Cui Zhang",
        "Course Url": "https://www.ecs.csus.edu/wcm/csc/pdfs/advising/csc201.pdf"
    },
    {
        "Course Id": "CSC204",
        "Course Name": "Data Models for Database Management Systems",
        "Course Information": "Database management system (DBMS) architecture. Database file organizations and access methods. The relational model and relational alg
        "Prerequisites of Course": "Fully classified graduate status in Computer Science or Software Engineering",
        "Units": "3",
        "Professor for the course": "Bill Mitchell",
        "Course Url": "https://www.ecs.csus.edu/wcm/csc/pdfs/advising/csc204.pdf"
    },
    {
        "Course Id": "CSC205",
        "Course Name": "Computer Systems Structure",
        "Course Information": "Overview of computer systems structure, covering hierarchical structure from software and hardware points of view. Concepts of relocati
        "Prerequisites of Course": "Fully classified graduate status in Computer Science or Software Engineering",
        "Units": "3",
        "Professor for the course": "Dr. Weide Chang",
        "Course Url": "https://www.ecs.csus.edu/wcm/csc/pdfs/advising/csc205.pdf"
    },
    {
        "Course Id": "CSC206",
        "Course Name": "Algorithms and Paradigms",
        "Course Information": "Design and analysis of algorithms. Classical design paradigms including greedy, divide-and-conquer, dynamic programming, and backtracki
        "Prerequisites of Course": "Fully classified graduate status in Computer Science or Software Engineering",
        "Units": "3",
        "Professor for the course": "Dr. Chung-E Wang",
        "Course Url": "https://www.ecs.csus.edu/wcm/csc/pdfs/advising/csc206.pdf"
    },
```


Figure 28: Sample Course Details

Figure 29 shows a sample file which contains basic information related to the professors like name, office hours, location, and email.

Downloaded by HASSAN ALI (hxn112233@gmail.com)



```
[
{
    "Name":"Dr. Nik Faroughi, Department Chair (CSC)",
    "Location":"RVR 3018G",
    "Email":"faroughi@ecs.csus.edu",
    "Phone":"916-278-7628 ",
    "Office Hours": "M W F 11:00am-12:00pm",
    "Webpage":"https://athena.ecs.csus.edu/~faroughi/"
},
{
    "Name":"Dr. Scott Gordon, Assoc. Chair (CSC)",
    "Location":"RVR 3018F",
    "Email":"gordonvs@ecs.csus.edu",
    "Phone":"Use Email",
    "Office Hours": "W 2:00pm - 4:00pm",
    "Webpage":"https://athena.ecs.csus.edu/~gordonvs/"
},
{
    "Name":"Dr. Jinsong Ouyang, Graduate Coordinator (CSC)",
    "Location":"RVR 3018I",
    "Email":"jouyang@csus.edu",
    "Phone":"916-278-5769",
    "Office Hours": "Tu R 1:30pm - 3:00pm",
    "Webpage":"https://www.csus.edu/faculty/o/jouyang/"
},
{
    "Name":"Dr. Pinar Muyan-Ozcelik",
    "Location":"RVR 5008",
    "Email":"pmuyan@ecs.csus.edu",
    "Phone":"916-278-6713",
    "Office Hours": "Tu 3:00pm - 3:50pm, R 1:40pm - 3:50pm",
    "Webpage":"https://athena.ecs.csus.edu/~pmuyan/"
},
{
    "Name":"Dr. Haiquan Chen",
    "Location":"RVR 5018",
    "Email":"chenh@ecs.csus.edu",
    "Phone":"916-278-6087",
    "Office Hours": "M W 3:00pm - 4:00pm",
    "Webpage":"https://athena.ecs.csus.edu/~chenh/"
},
```


Figure 29: Sample Faculty Details

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

4 EXECUTION RESULTS

In this section, execution results of College Enquiry Chatbots are provided by listing several functionalities included in this project and explaining these functionalities with the help of the screenshots.

4.1 How to chat

To start chat with the chatbot, a student will have to go to the university website www.csus.edu and click on the question mark icon as shown in Figure 30.

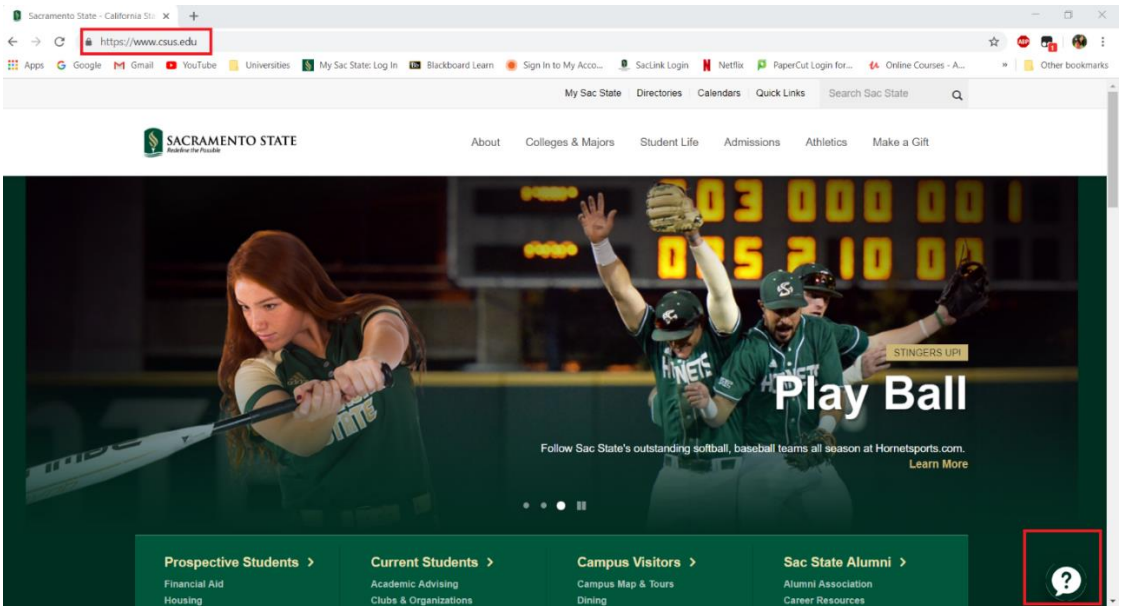


Figure 30: How to Chat

To have the chatbot on a live website, access to the codebase of a website is required. However, for now a software called Tampermonkey is used to inject the Java script into the browser.

Downloaded by HASSAN ALI (hxn112233@gmail.com)

4.2 Sentiment Analysis

Figures 31 and 32 show how a bot is answering the irrelevant questions of a user for which normal bots are not showing any kind of empathy. To make the bot to be able to respond these kinds of responses, the bot is trained with the chitchat integration which is provided by QnA Maker in which there are lots of questions and answers pairs are available.

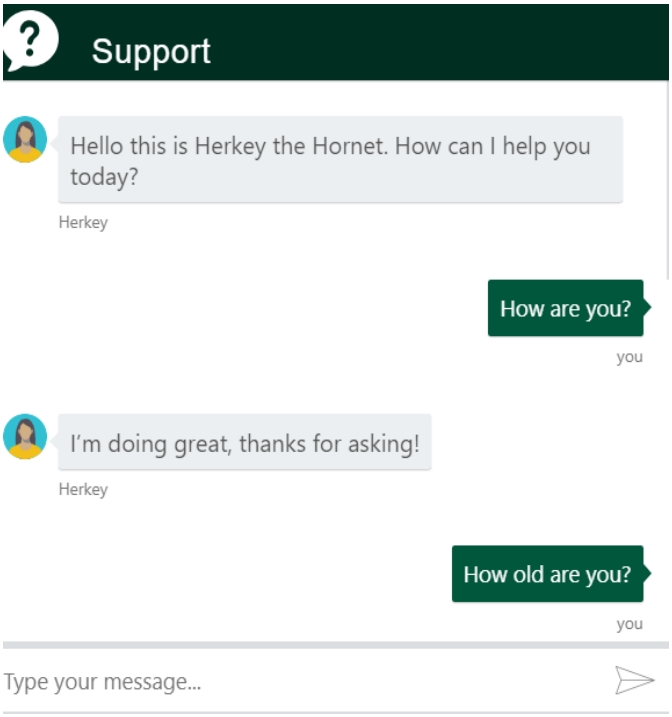


Figure 31: Friendly Conversations - 1

This document is available free of charge on studocu
Downloaded by HASSAN ALI (hxn112233@gmail.com)

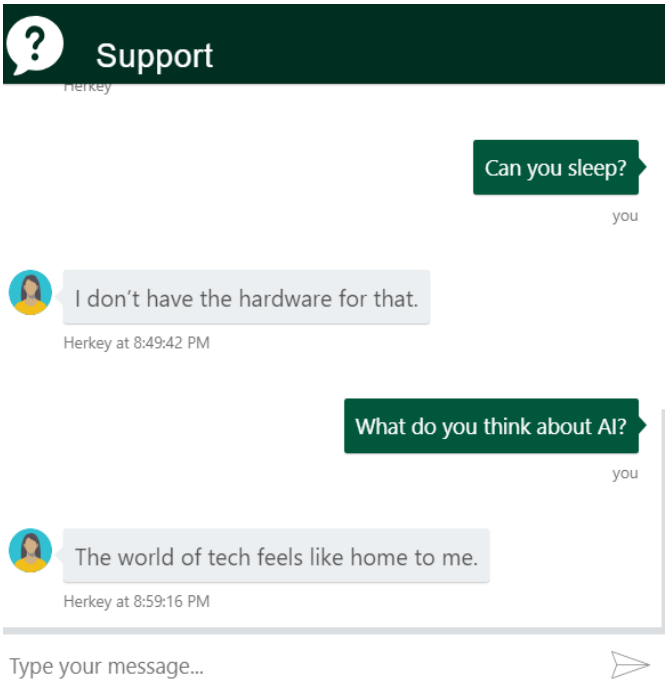
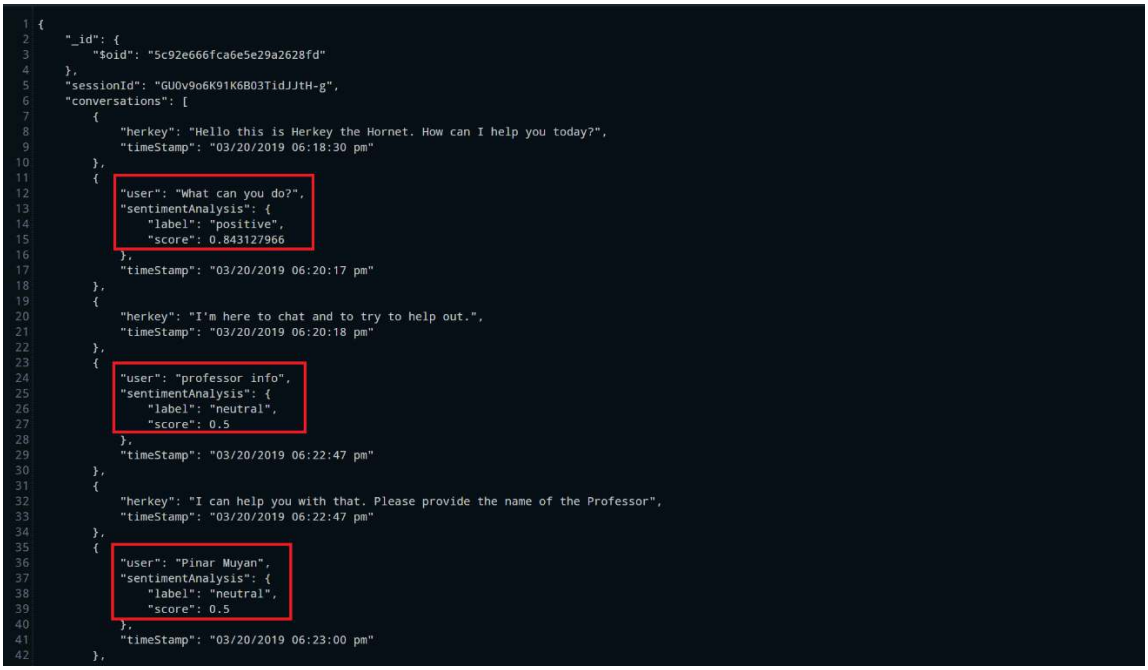


Figure 32: Friendly Conversations – 2

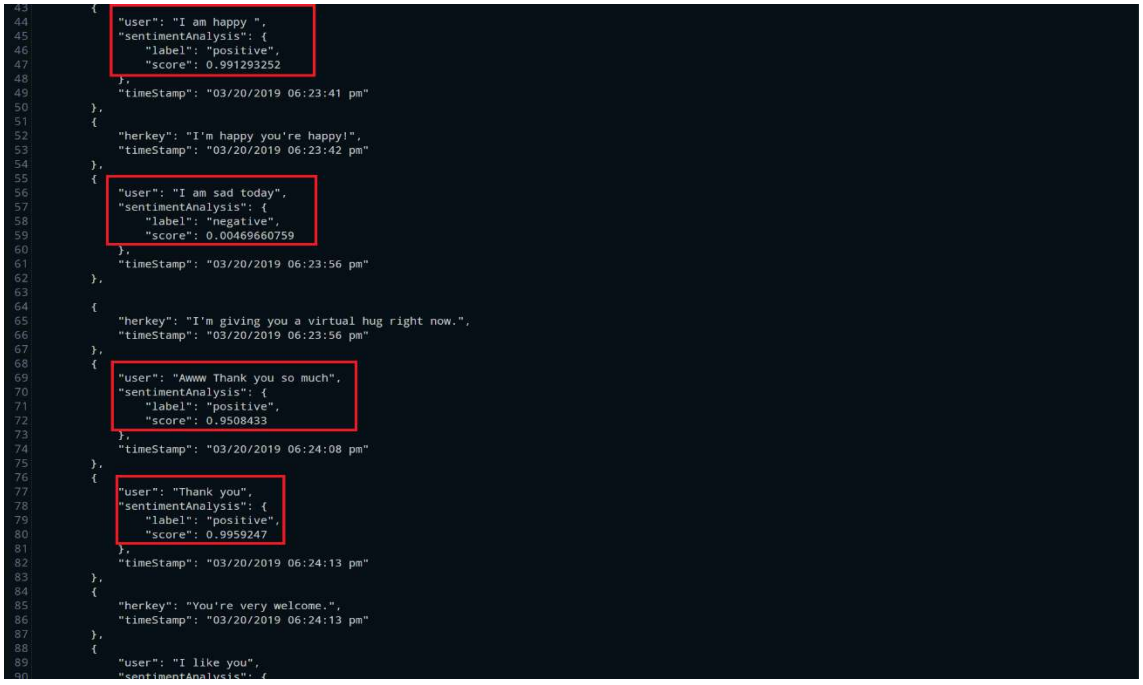
Figures 33 and 34 shows how conversations are being stored in the database and how LUIS performs sentiment analysis on user queries based on which it gives the response back to the user. A bot interceptor is used in the service where conversations are tracked and user's query is passed onto the LUIS endpoint which has an in-built sentiment analyzer to track whether the query is positive, negative or neutral. These results are then stored into the database for any further analysis.

Downloaded by HASSAN ALI (hxn112233@gmail.com)



```
{
    "_id": {
        "$oid": "5c92e666fca6e5e29a2628fd"
    },
    "sessionId": "GUOv9o6K91K6B03TidJJtH-g",
    "conversations": [
        {
            "herkey": "Hello this is Herkey the Hornet. How can I help you today?",
            "timeStamp": "03/20/2019 06:18:30 pm"
        },
        {
            "user": "What can you do?",
            "sentimentAnalysis": {
                "label": "positive",
                "score": 0.843127966
            },
            "timeStamp": "03/20/2019 06:20:17 pm"
        },
        {
            "herkey": "I'm here to chat and to try to help out.",
            "timeStamp": "03/20/2019 06:20:18 pm"
        },
        {
            "user": "professor info",
            "sentimentAnalysis": {
                "label": "neutral",
                "score": 0.5
            },
            "timeStamp": "03/20/2019 06:22:47 pm"
        },
        {
            "herkey": "I can help you with that. Please provide the name of the Professor",
            "timeStamp": "03/20/2019 06:22:47 pm"
        },
        {
            "user": "Pinar Muyan",
            "sentimentAnalysis": {
                "label": "neutral",
                "score": 0.5
            },
            "timeStamp": "03/20/2019 06:23:00 pm"
        },
```


Figure 33: Sentiment Analysis on User Queries - 1



```
        {
            "user": "I am happy ",
            "sentimentAnalysis": {
                "label": "positive",
                "score": 0.991293252
            },
            "timeStamp": "03/20/2019 06:23:41 pm"
        },
        {
            "herkey": "I'm happy you're happy!",
            "timeStamp": "03/20/2019 06:23:42 pm"
        },
        {
            "user": "I am sad today",
            "sentimentAnalysis": {
                "label": "negative",
                "score": 0.00469660759
            },
            "timeStamp": "03/20/2019 06:23:56 pm"
        },

        {
            "herkey": "I'm giving you a virtual hug right now.",
            "timeStamp": "03/20/2019 06:23:56 pm"
        },
        {
            "user": "Awww Thank you so much",
            "sentimentAnalysis": {
                "label": "positive",
                "score": 0.9508433
            },
            "timeStamp": "03/20/2019 06:24:08 pm"
        },
        {
            "user": "Thank you",
            "sentimentAnalysis": {
                "label": "positive",
                "score": 0.9959247
            },
            "timeStamp": "03/20/2019 06:24:13 pm"
        },
        {
            "herkey": "You're very welcome.",
            "timeStamp": "03/20/2019 06:24:13 pm"
        },
        {
            "user": "I like you",
```


Figure 34: Sentiment Analysis on User Queries – 2

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

4.3 Active Learning

Whenever a user asks anything which is outside of the script, the bot asks questions back to the user to understand what exactly user wants to ask and gives the responses accordingly. Figures 35 and 36 shows how a student has asked “on campus” randomly and then bot has asked question in return with three different options to understand what the user means. Once user has selected one of the options it did frame a question based on that option and gave the response to the user.

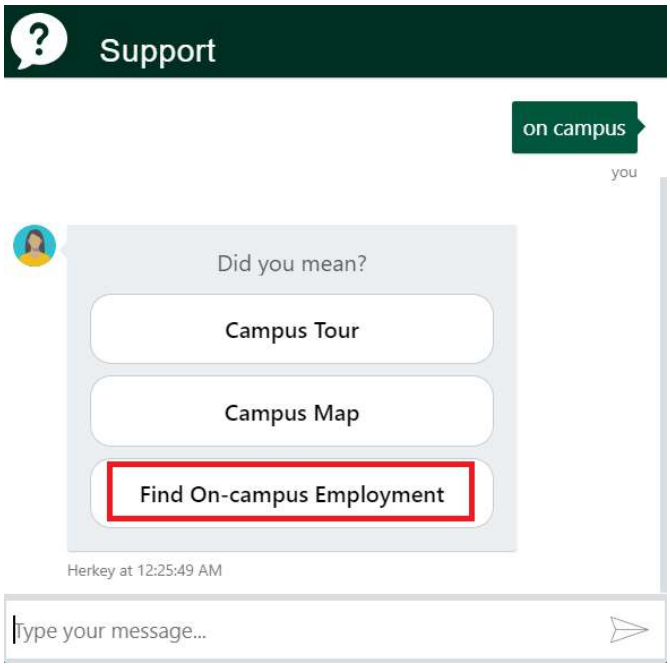


Figure 35: Active Learning - 1

Downloaded by HASSAN ALI (hxn112233@gmail.com)

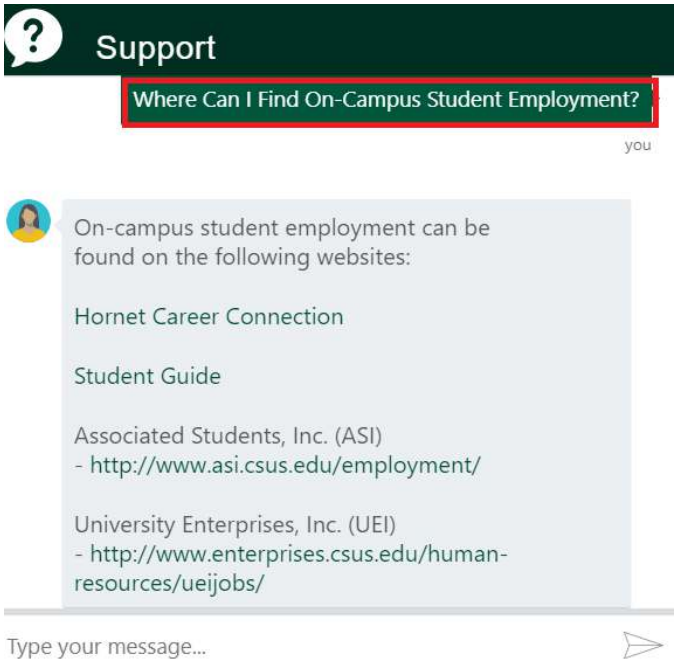


Figure 36: Active Learning – 2

4.4 Frequently Asked Questions

The system inserts FAQs related to academics, registration, records, evaluations/graduation, applications, current students, and campus information. Figures 37 and 38 show two different highlighted questions and answers from two different FAQ URLs which are added to QnA Maker.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

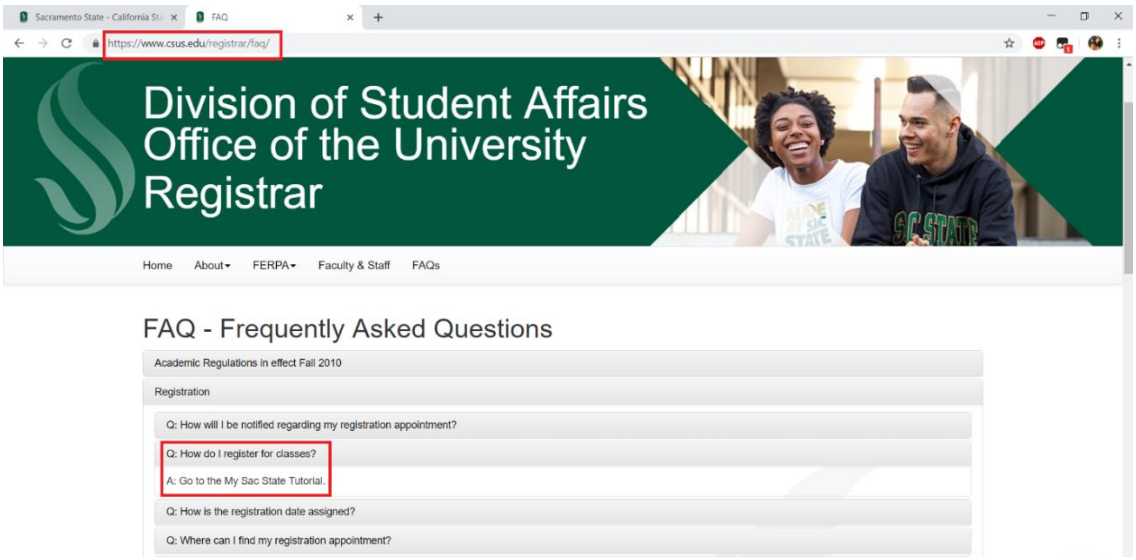


Figure 37: FAQs from Registrar

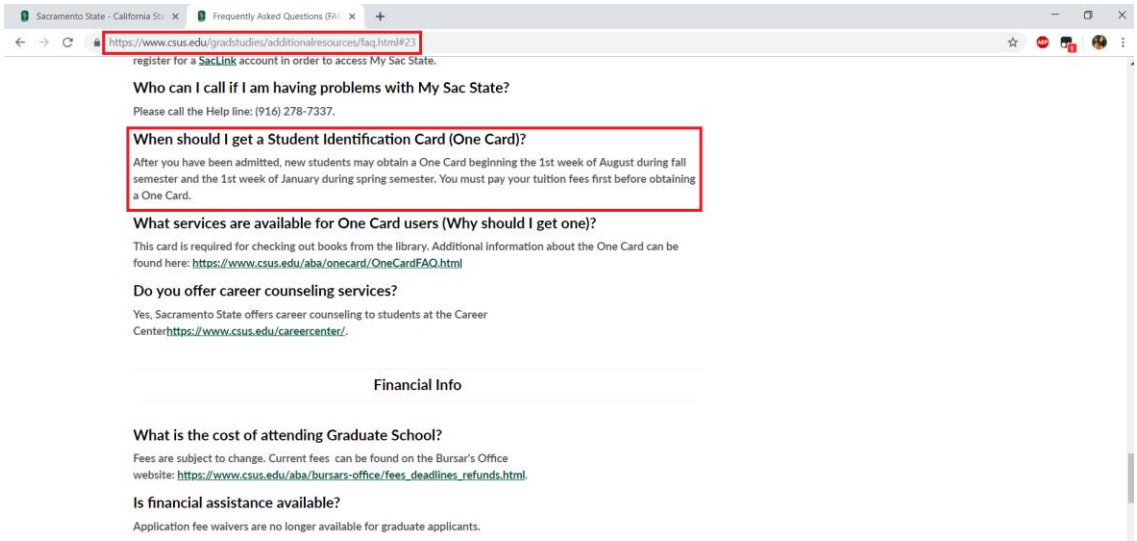


Figure 38: FAQ from Graduate Studies

Downloaded by HASSAN ALI (hxn112233@gmail.com)

Figure 39 shows how questions from FAQs are asked and responded in an exact same way. In this case, notice that a student does not have to ask an exact same question included in FAQs. This is the main advantage of an AI-powered chatbot.

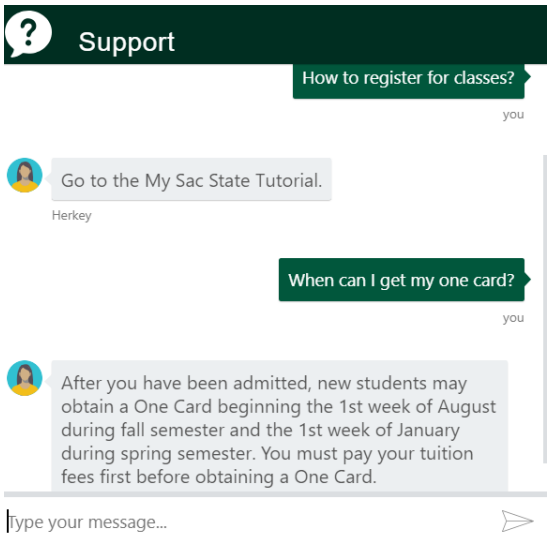


Figure 39: FAQs Asked to Chatbot

4.5 Course and Faculty Details

Figure 40 shows how chatbot is giving the information about course when asked. It shows name, overview, prerequisites, number of units, and professor of the subject. It also gives a course link to the student for more information.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

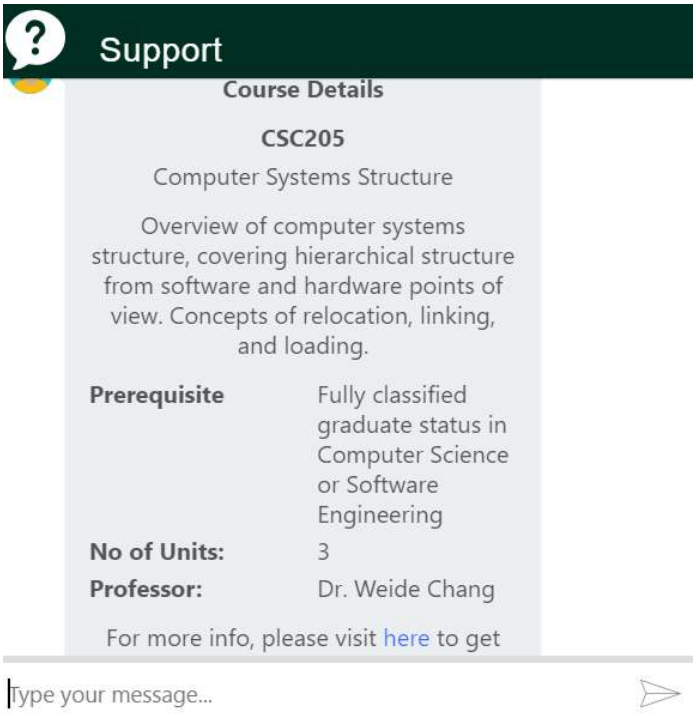


Figure 40: Course Information

Figure 41 shows the chatbot's response about office hours of a faculty. It gives all the necessary details about the faculty such as name, location of the office, email, phone, and office hours of the faculty. It also gives a link to the faculty's website for more information.

Downloaded by HASSAN ALI (hxn112233@gmail.com)

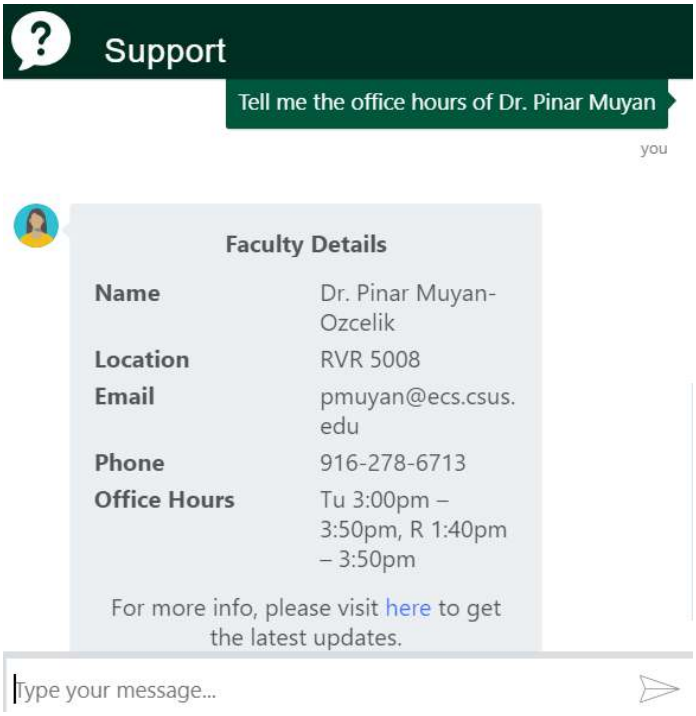


Figure 41: Faculty Information

4.6 Academic Calendar

Figure 42 shows how the bot gives a link to an academic calendar if a student is asking about all the holidays of the semester.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

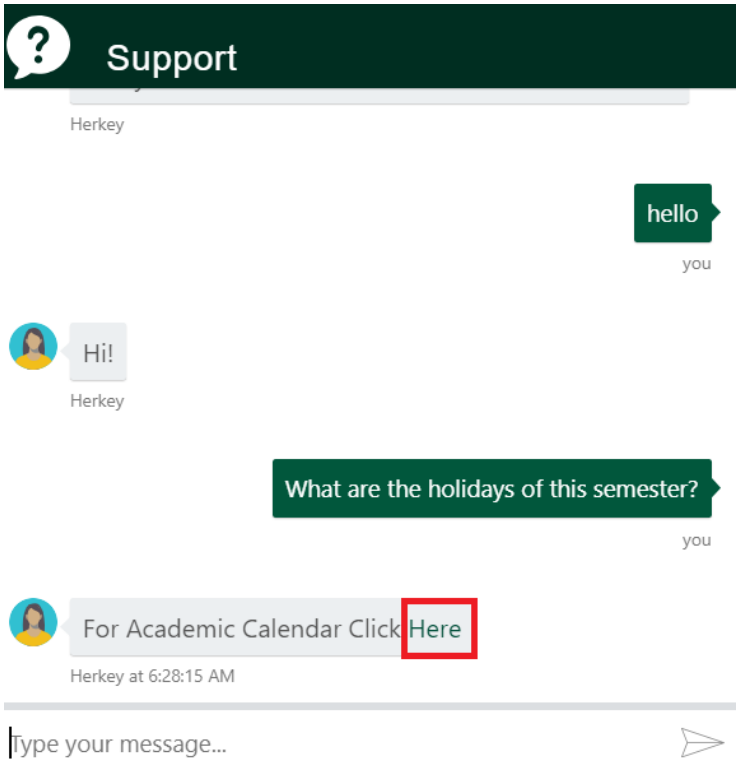


Figure 42: Academic Calendar

4.7 Fee Calculator

Figures 43 and 44 shows about the chatbot providing facility of calculating the fees by just selecting undergraduate or graduate and inserting number of units.

Downloaded by HASSAN ALI (hxn112233@gmail.com)

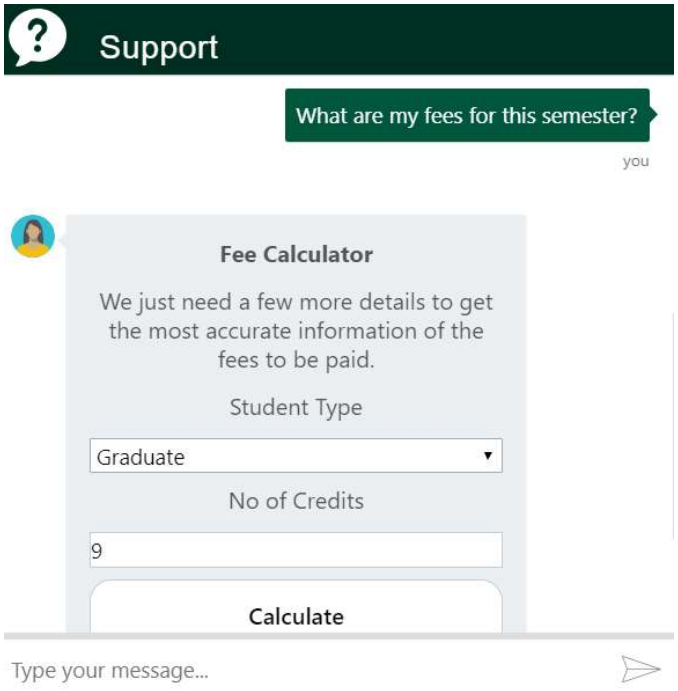


Figure 43: Fee Calculator – 1

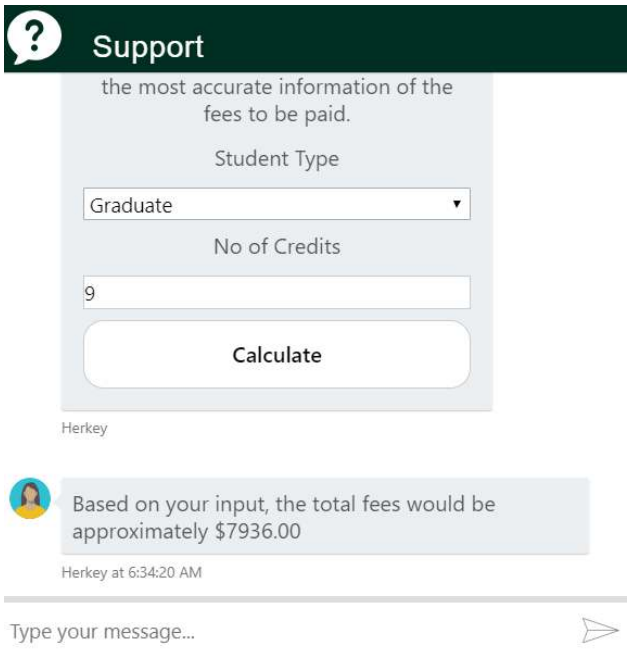


Figure 44: Fee Calculator - 2

This document is available free of charge on studocu
Downloaded by HASSAN ALI (hxn112233@gmail.com)

4.8 Other Departmental Information

Figures 45 and 46 shows how the chatbot gives information about IPGE and Union office hours, address, contact info, and email. It also gives a link to the respected department for more information.

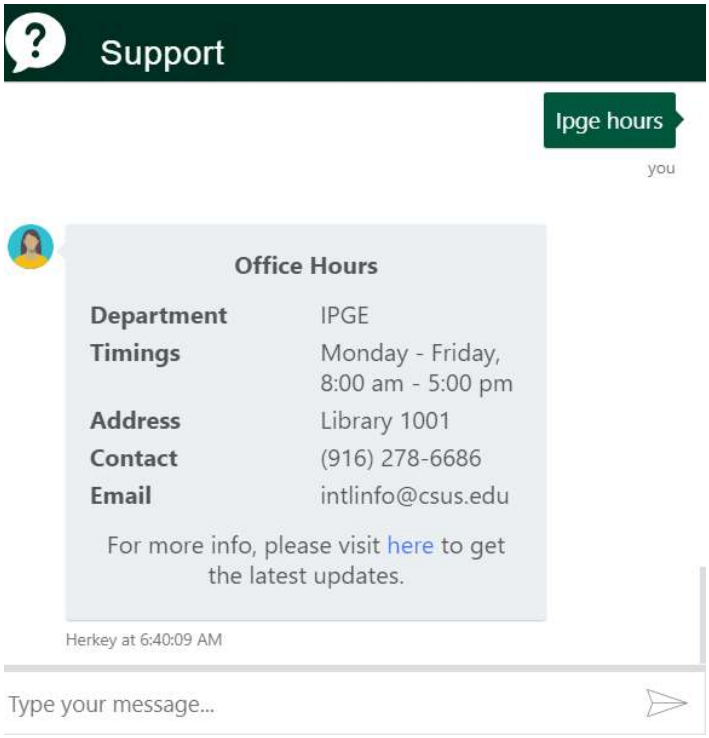


Figure 45: IPGE Hours Information

Downloaded by HASSAN ALI (hxn112233@gmail.com)

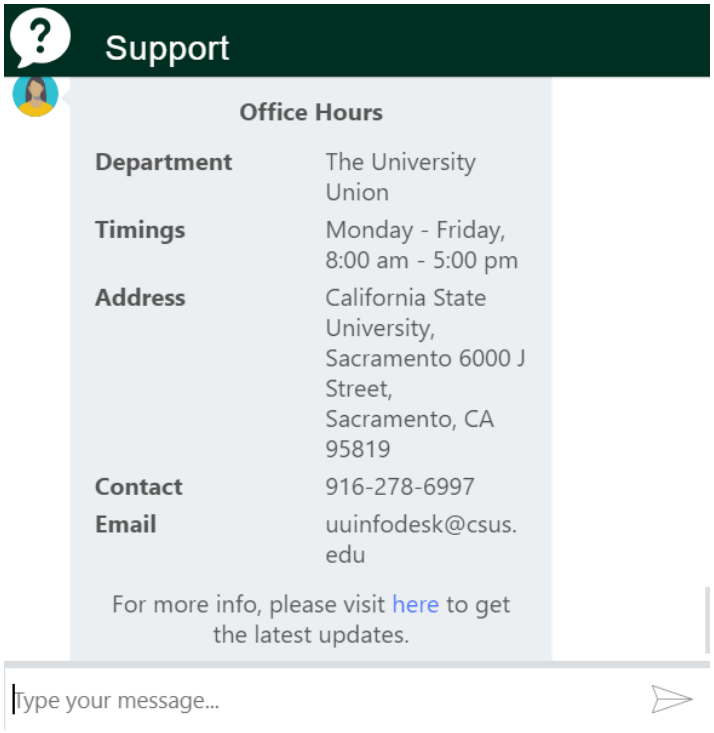


Figure 46: Union Hours Information

4.9 Results

The results of this project are measured in whether sentiment analysis and active learning is correctly implemented or not. Sentiment analysis correctly recognizes the user's query such as positive, negative, and neutral by storing all the conversations in the database (as shown in Figures 33 and 34). These results are used to add empathy to the bot. However, the system was partially successful in adding empathy to the bot. It is because, although large amount of data was added to include some common answers to the queries which are off scripts and to add empathy to the bot (so that it understands what is the current mood of the user and responds accordingly), since scope of these queries is vast, the system

This document is available free of charge on studocu
Downloaded by HASSAN ALI (hxn112233@gmail.com)

requires more rigorous data to handle all the questions which are out of script. This can be illustrated in the Figure 47 where the bot understood that I am in a negative sentiment, so it answered with a proper response. But, it did not understand the meaning of 'I am excited' since bot did not have any training data related to the typed query as shown in Figure 48.

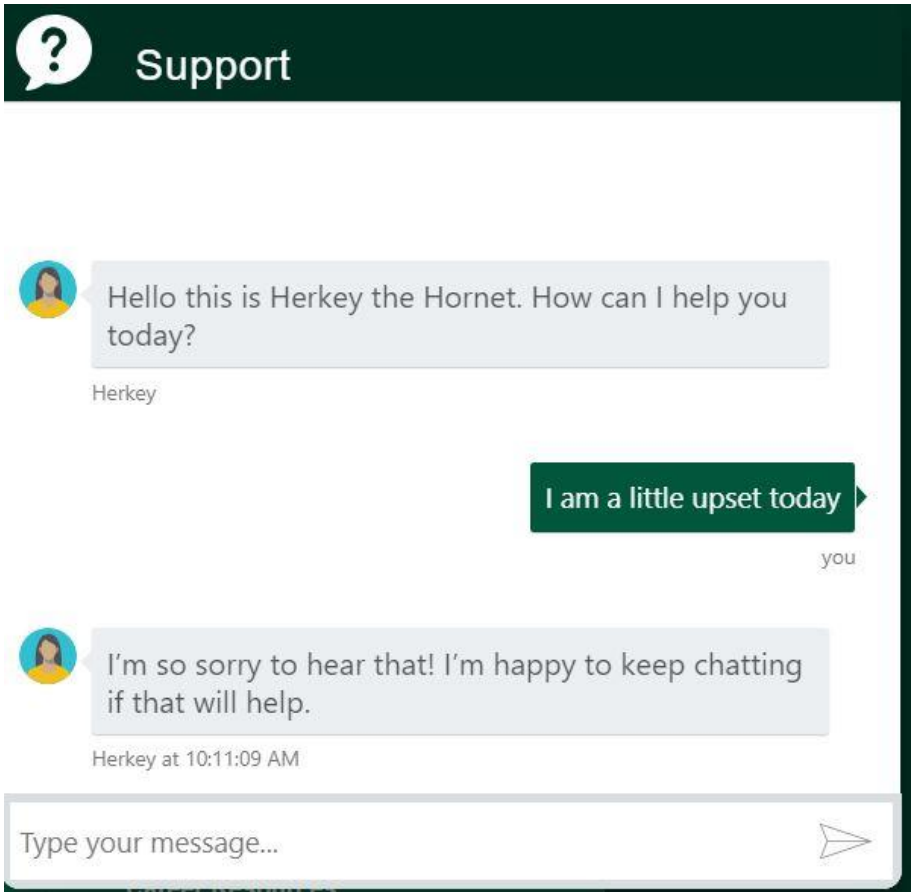


Figure 47: Bot successfully detecting the sentiment of the query

Downloaded by HASSAN ALI (hxn112233@gmail.com)

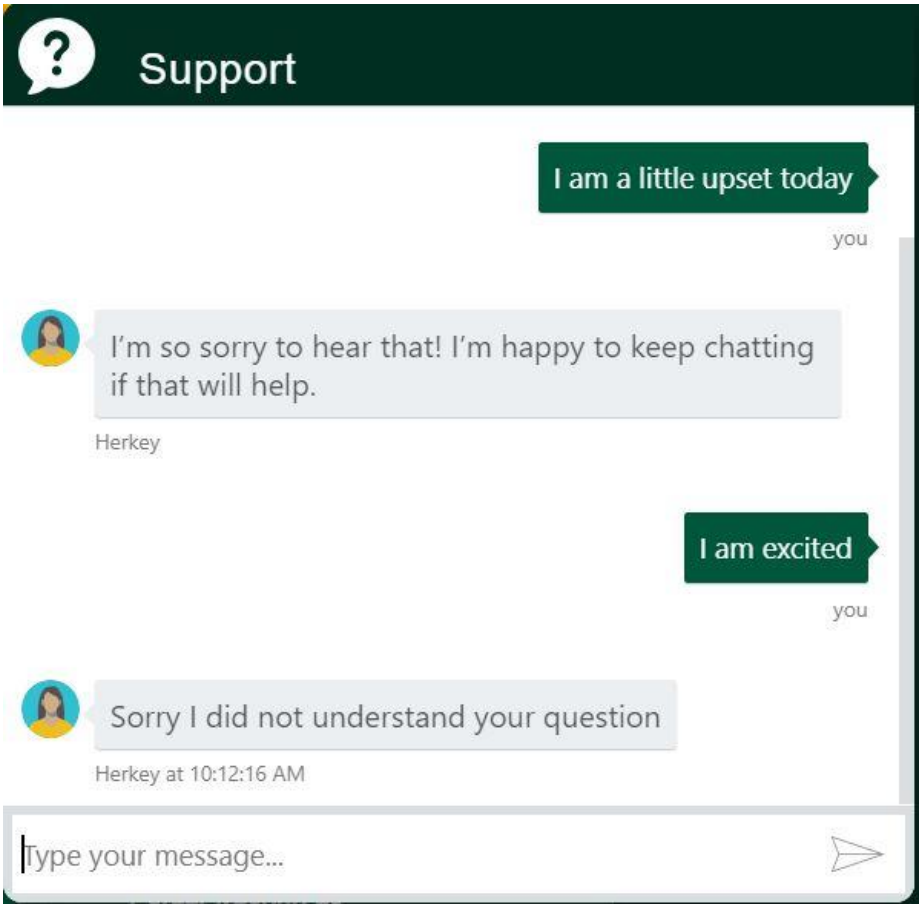


Figure 48: Bot failed to detect the sentiment of the query

On the other hand, active learning helps to improve the bot performance for handling off-script queries. It correctly understands the user's questions, asks clarifying question, and then re-trains the NLP to give response what the user is intended to get.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

5 CONCLUSION AND FUTURE WORK

To conclude, College Enquiry Chatbot is helpful in guiding students with correct and most up to date sources of information. It is advantageous for international applicants for queries such as fee payment and academic matters. Students can get the information at their fingertips rather than visiting college office. It improves efficiency by taking over tasks for which humans are not essential.

Sentiment analysis implemented in College Enquiry Chatbot correctly recognizes the user's query such as positive, negative, and neutral by storing all the conversations in the database. However, the system was partially successful in adding empathy since scope of these queries is vast and the system requires more rigorous data to handle all the questions which are out of script. Nevertheless, active learning helps to improve the bot performance for handling off-script queries.

To improve the current functionalities of College Enquiry Chatbot, in the future, the scope of the chatbot can be increased by inserting data for all the departments, training the bot with varied data, testing it on live website, and based on that feedback inserting more training data to the bot. Some of the new features which can be added to the bot are 1) speech recognition feature through which students can ask their queries verbally and get the answers from the bot, 2) integration with multiple channels such as phone call, SMS, and various social media platforms like Skype, Facebook and Twitter, 3) handling context aware and interactive queries in which bot will be aware of the context of an ongoing conversation with a student, 4) integration with services such as password reset and course

Downloaded by HASSAN ALI (hxn112233@gmail.com)

enrollment, and 5) adding a capability for the bot to perform analytics based on user's sentiment based on which the bot can be re-trained on human emotions so that more empathy can be added to the bot.

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

References

[1] "Chatbot", en.wikipedia.org, 2010. [Online]. Available: https://en.wikipedia.org/wiki/Chatbot. [Accessed Jul. 10, 2018].

[2] "Chatbot: What is Chatbot? Why are Chatbots Important? -", Expertsystem.com, 2018. [Online]. Available: https://www.expertsystem.com/chatbot/. [Accessed Sep. 11, 2018].

[3] "Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more", Amazon.com, 2017. [Online]. Available: https://www.amazon.com/. [Accessed Sep. 12, 2018].

[4] "Amazon Alexa - Build for Amazon Echo Devices", Developer.amazon.com, 2017. [Online]. Available: https://developer.amazon.com/alexa. [Accessed Sep. 12, 2018].

[5] "Meet Erica, Your Financial Digital Assistant From Bank of America", Bank of America, 2018. [Online]. Available: https://promo.bankofamerica.com/erica/. [Accessed Sep. 12, 2018].

[6] C. Doug Gross, "Facebook: Get Messenger app or else - CNN", CNN, 2014. [Online]. Available: https://www.cnn.com/2014/07/29/tech/social-media/facebook-messenger/index.html. [Accessed Sep. 12, 2018].

[7] A. Storman, "5 Chatbot Challenges and How to Overcome Them", Chatbots Magazine, 2016. [Online]. Available: https://chatbotsmagazine.com/5-chatbot-challenges-and-how-to-overcome-them-caccc3a26d7c. [Accessed Sep. 11, 2018].

Downloaded by HASSAN ALI (hxn112233@gmail.com)

[8] A. M. Rahman, A. A. Mamun and A. Islam, "Programming challenges of chatbot: Current and future prospective," *in 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, 2017*.

[9] "Problems with Chatbots Today - Attunix", Attunix, 2018. [Online]. Available: https://attunix.com/problems-chatbots-today/. [Accessed Sep. 11, 2018].

[10] "Azure Bot Service - chatbot | Microsoft Azure", Azure.microsoft.com, 2017. [Online]. Available: https://azure.microsoft.com/en-us/services/bot-service/. [Accessed Jul. 22, 2019].

[11] "LUIS (Language Understanding) – Cognitive Services – Microsoft Azure", Luis.ai, 2017. [Online]. Available: https://www.luis.ai/home. [Accessed Jul. 15, 2018].

[12] "QnA Maker", Qnamaker.ai, 2016. [Online]. Available: https://www.qnamaker.ai/. [Accessed Jul. 17, 2018].

[13] "What is Azure Bot Service? - Definition from WhatIs.com", WhatIs.com, 2018. [Online]. Available: https://whatis.techtarget.com/definition/Azure-Bot-Service. [Accessed Jan. 17, 2019].

[14] "Azure App Service Documentation - Tutorials, API Reference", Docs.microsoft.com, 2015. [Online]. Available: https://docs.microsoft.com/en-us/azure/app-service/. [Accessed Dec. 17, 2018].

[15] "Cognitive Services | Microsoft Azure", Azure.microsoft.com, 2017. [Online]. Available: https://azure.microsoft.com/en-us/services/cognitive-services/. [Accessed Dec. 22, 2018].

This document is available free of charge on studocu

Downloaded by HASSAN ALI (hxn112233@gmail.com)

[16] "LUIS.AI: Automated Machine Learning for Custom Language Understanding", azure.microsoft.com, 2017. [Online]. Available: https://azure.microsoft.com/en-us/blog/luis-ai-automated-machine-learning-for-custom-language-understanding/. [Accessed Jul. 22, 2019].

[17] "Review user utterances - Language Understanding - Azure Cognitive Services", Docs.microsoft.com, 2019. [Online]. Available: https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-how-to-review-endoint-utt. [Accessed Sep. 14, 2018].

Downloaded by HASSAN ALI (hxn112233@gmail.com)