

Task 6 C, D: Language Understanding

Introduction

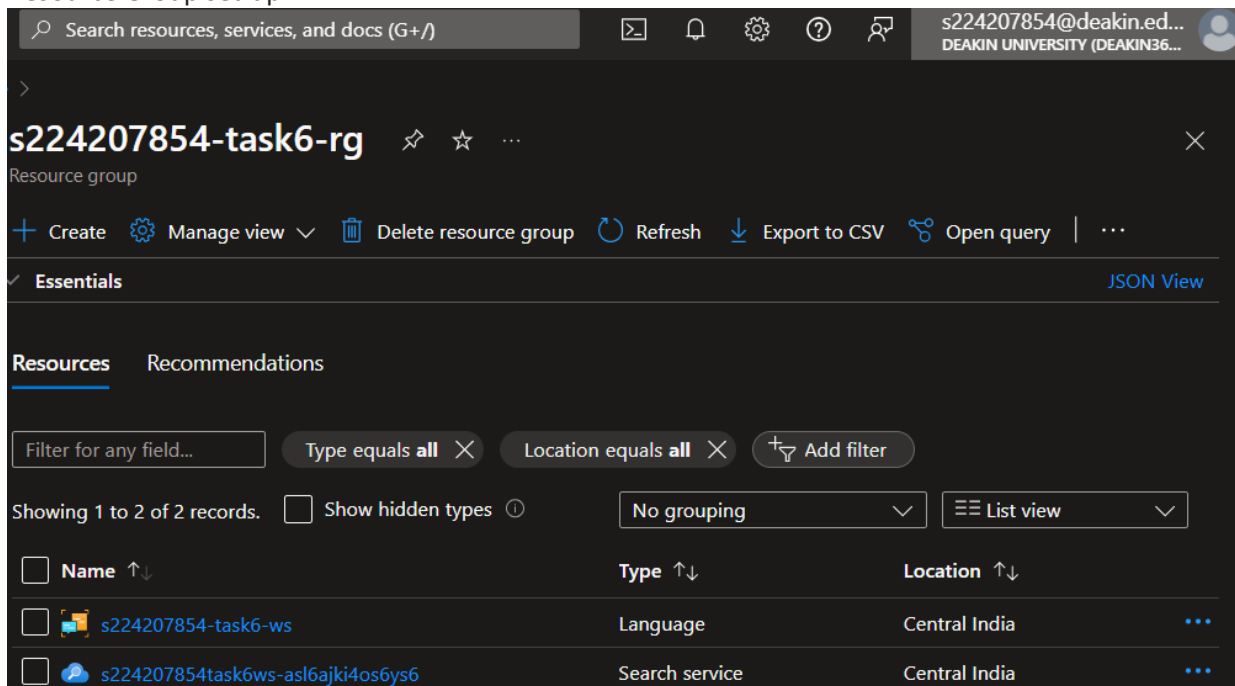
Azure Language Understanding (LUIS) is a cloud-based service that uses machine learning to interpret natural language text and predict the user's overall meaning. It can also extract relevant information from conversational phrases.

LUIS can be used to add natural language to apps, bots, and IoT devices.

Pre-requisites

In this case study we will be using Azure language understanding service using Azure Cognitive Services SDK. Before starting coding will be doing some set-ups:

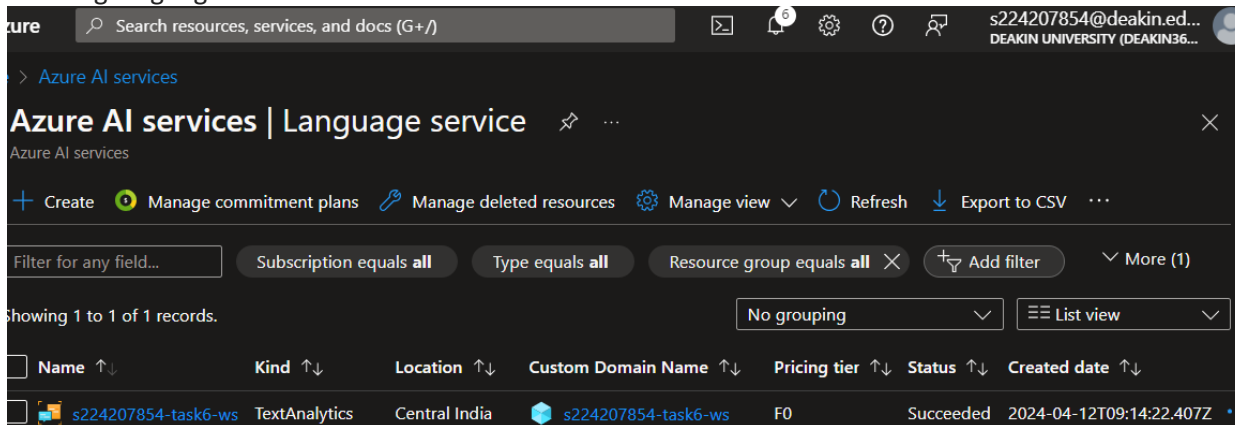
1. Resource Group set-up



The screenshot shows the Azure portal interface for the resource group 's224207854-task6-rg'. The 'Resources' tab is selected, showing a table of resources. The table has columns for Name, Type, and Location. Two resources are listed: 's224207854-task6-ws' (Language) and 's224207854task6ws-asl6ajki4os6ys6' (Search service), both located in Central India.

Name	Type	Location
s224207854-task6-ws	Language	Central India
s224207854task6ws-asl6ajki4os6ys6	Search service	Central India

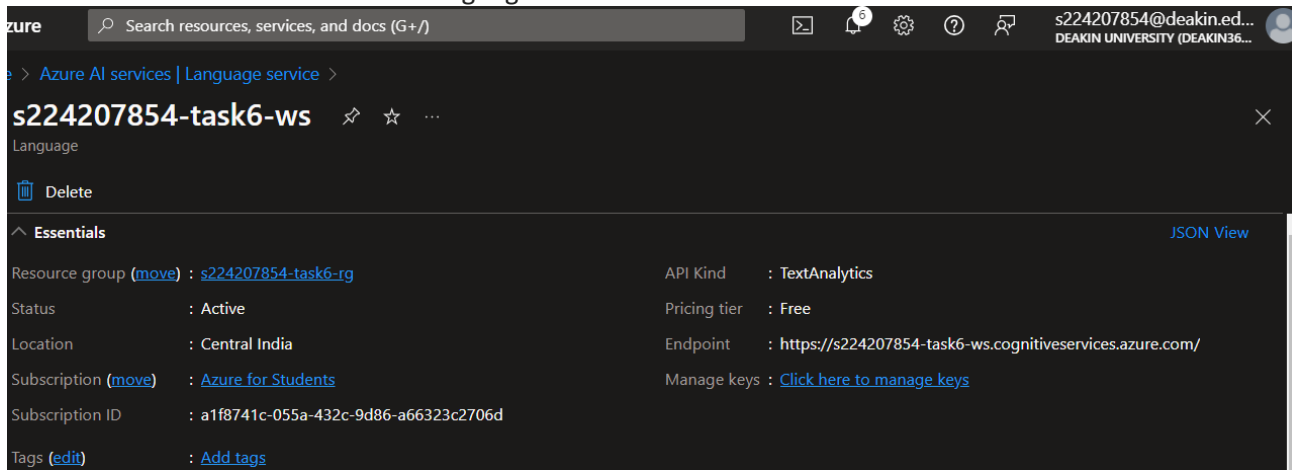
2. Creating language service in Azure AI services



The screenshot shows the Azure portal interface for the 'Azure AI services | Language service'. The table displays one record for the 's224207854-task6-ws' TextAnalytics service, which is in a 'Succeeded' state.

Name	Kind	Location	Custom Domain Name	Pricing tier	Status	Created date
s224207854-task6-ws	TextAnalytics	Central India	s224207854-task6-ws	F0	Succeeded	2024-04-12T09:14:22.407Z

3. In Azure AI service we can visit our language service



4. Installing Azure question-answering library

```
# Install the azure question answering library
%pip install azure-ai-language-questionanswering
```

5. Importing different libraries and loading subscription key and endpoint from .env file for Azure client.

```
# To read the secret keys for Authentication
import os
from dotenv import load_dotenv
from azure.core.credentials import AzureKeyCredential

# To create new project
from azure.ai.language.questionanswering.authoring import AuthoringClient

# to create a question-answering client, and to ask questions using the knowledge base
from azure.ai.language.questionanswering import QuestionAnsweringClient
from azure.ai.language.questionanswering import models as qna

0.2s

# get service secrets
load_dotenv()
endpoint = os.environ.get("endpoint")
key = os.environ.get("subscription_key")
```

6. Once pre-requisites are done, we can use these info for our case study i.e. ICICI bank loan system FAQ for personal loan, home loan and car loan.

Azure Language Service Notebook

In our notebook we will be using azure authoring client to create new project, add knowledge base and then deployment. In the below code snippet, 3 steps have been followed, i.e.:

- Create new project in Language Studio and can be checked here: <https://language.cognitive.azure.com>
- Add knowledge base to your project. A minimum of one knowledge base is required.
- Once our knowledge base is ready, we can deploy our project for the usage as BOT or attach it to a different channel like Facebook, Telegram.

```
# Create new project, add knowledge base and deploy it.
with AuthoringClient(endpoint, AzureKeyCredential(key)) as authoring_client:

    # Step 1: create project
    print("\n***** Creating a new project *****")
    project_name = "ICICIBank-LoanFAQs"
    project = authoring_client.create_project(
        project_name=project_name,
        options={
            "description": "FAQs related to ICICI banking loan system for Personal, Home and Car Loan.",
            "language": "en",
            "multilingualResource": True,
            "settings": {"defaultAnswer": None},
        },
    )

    # Output 1: View the project details
    print("view created project info:")
    print("\tname: {}".format(project["projectName"]))
    print("\tlanguage: {}".format(project["language"]))
    print("\tdescription: {}".format(project["description"]))

    # Step 2: Add a knowledge base
    print("\n***** Adding a knowledge base *****")

    update_sources_poller = authoring_client.begin_update_sources(
        project_name=project_name,
        sources=[
            {
                "op": "add",
                "value": {
                    "displayName": "Personal Loan FAQ",
                    "sourceUri": "https://www.icicibank.com/personal-banking/loans/personal-loan/personal-loans-faqs",
                    "sourceKind": "url",
                },
            },
            {
                "op": "add",
                "value": {
                    "displayName": "Home Loan FAQ",
                    "sourceUri": "https://www.icicibank.com/personal-banking/loans/home-loan/home-loans-faqs",
                    "sourceKind": "url",
                },
            },
            {
                "op": "add",
                "value": {
                    "displayName": "Car Loan FAQ",
                    "sourceUri": "https://www.icicibank.com/personal-banking/loans/car-loan/car-loans-faqs",
                    "sourceKind": "url",
                },
            },
        ],
    )
    update_sources_poller.result()

    # Output 2: list sources
    print("\nlist project sources")
    sources = authoring_client.list_sources(project_name=project_name)
    for source in sources:
        print("knowledge base name: {}".format(source["displayName"]))
        print("\tsource: {}".format(source["source"]))
        print("\tsource Uri: {}".format(source["sourceUri"]))
        print("\tsource Kind: {}".format(source["sourceKind"]))

    # Step 3: deploy the project
    print("\n***** Deploying the project *****")

    deployment_poller = authoring_client.begin_deploy_project(
        project_name=project_name, deployment_name="production"
    )
    deployment_poller.result()

    # list all deployments
    deployments = authoring_client.list_deployments(project_name=project_name)

    print("view project deployments")
    for d in deployments:
        print(d)
```

5m 27.7s

- Output of above snippet

```
***** Creating a new project *****
view created project info:
  name: ICICIBank-LoanFAQs
  language: en
  description: FAQs related to ICICI banking loan system for Personal, Home and Car Loan.

***** Adding a knowledge base *****

list project sources
knowledge base name: Personal Loan FAQ
  source: https://www.icicibank.com/personal-banking/loans/personal-loan/personal-loans-faqs
  source Uri: https://www.icicibank.com/personal-banking/loans/personal-loan/personal-loans-faqs
  source kind: url
knowledge base name: Home Loan FAQ
  source: https://www.icicibank.com/personal-banking/loans/home-loan/home-loans-faqs
  source Uri: https://www.icicibank.com/personal-banking/loans/home-loan/home-loans-faqs
  source kind: url
knowledge base name: Car Loan FAQ
  source: https://www.icicibank.com/personal-banking/loans/car-loan/car-loans-faqs
  source Uri: https://www.icicibank.com/personal-banking/loans/car-loan/car-loans-faqs
  source kind: url

***** Deploying the project *****
view project deployments
{'deploymentName': 'production', 'lastDeployedDateTime': '2024-04-12T09:46:20Z'}
```

- We can check our project from language studio

Azure AI | Language Studio

Language Studio > Custom question answering > ICICIBank-LoanFAQs - Manage sources

Manage sources

+ Add source Edit name Refresh URL Delete 3 items in list Filter

Source	Source name	Unstructured	Source type
https://www.icicibank.com/personal-banking/loans/personal-loan/personal-loans-faqs	Personal Loan FAQ	No	url
https://www.icicibank.com/personal-banking/loans/home-loan/home-loans-faqs	Home Loan FAQ	No	url
https://www.icicibank.com/personal-banking/loans/car-loan/car-loans-faqs	Car Loan FAQ	No	url

- Deployment of knowledge base can also be done from UI and after that we can use it as BOT for our service.

Azure AI | Language Studio

Language Studio > Custom question answering > ICICIBank-LoanFAQs - Deploy knowledge base

Deploy knowledge base

Deploy knowledge base and create a bot in a few clicks.

Deploy Get prediction URL

✓ Your knowledge base is now deployed. You can get your prediction URL or create a bot.

Knowledge base status			
State: ①	Deployed	✓	Resource: s224207854-task6-ws
Deployment Date: ①	4/12/2024	✓	Location: centralindia
Deployment Time: ①	3:16:20 PM	✓	Tier: Free (F0)

Next steps: Create a bot

Step 1: [Read the documentation](#) to learn more about creating bots.

Step 2: Go to Azure to create a bot.

Create a bot

Testing

For testing our Azure language service, we will be using question-answer client.

In the below code snippet, we are using deployed project and with confidence threshold more than 50%, we are taking that answer as our output to the service.

```
def ask_question(question):
    with QuestionAnsweringClient(endpoint, AzureKeyCredential(key)) as qna_client:
        output = qna_client.get_answers(
            question=question,
            top=3,
            confidence_threshold=0.5,
            include_unstructured_sources=True,
            short_answer_options=qna.ShortAnswerOptions(
                confidence_threshold=0.5, top=1
            ),
            project_name=project_name,
            deployment_name="production",
        )
        answers = (
            [a for a in output.answers if a.confidence and a.confidence > 0.5]
            if output.answers
            else []
        )
        if len(answers) > 0:
            best_candidate = answers[0]
            print(f"Q: {question}")
            print(f"A: {best_candidate.answer}")
        else:
            print(f"No answers returned from question '{question}'")
```

Question Test-1

- Input question: *personal loan interest rate?*

Q: personal loan interest rate?

A: Personal Loan interest rates are applied basis the defined pricing matrix followed and standardised by the bank as per customer's profile. Personal Loan rates vary from 10.80% to 16.15% per annum.

Question Test-2

- Input question: *offer me tea in bank?*

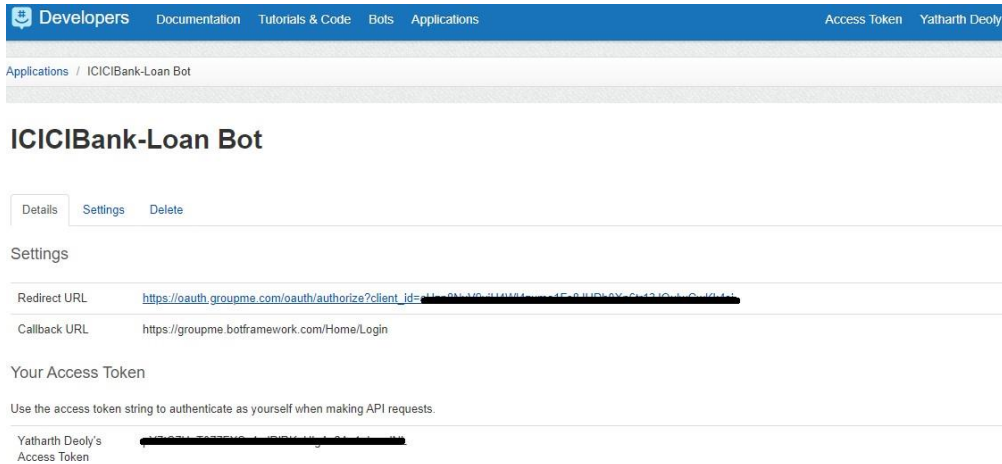
No answers returned from question 'offer me tea in bank?'

Bot Creation and plugging to channel

In this case study we are using GroupMe application for BOT testing.

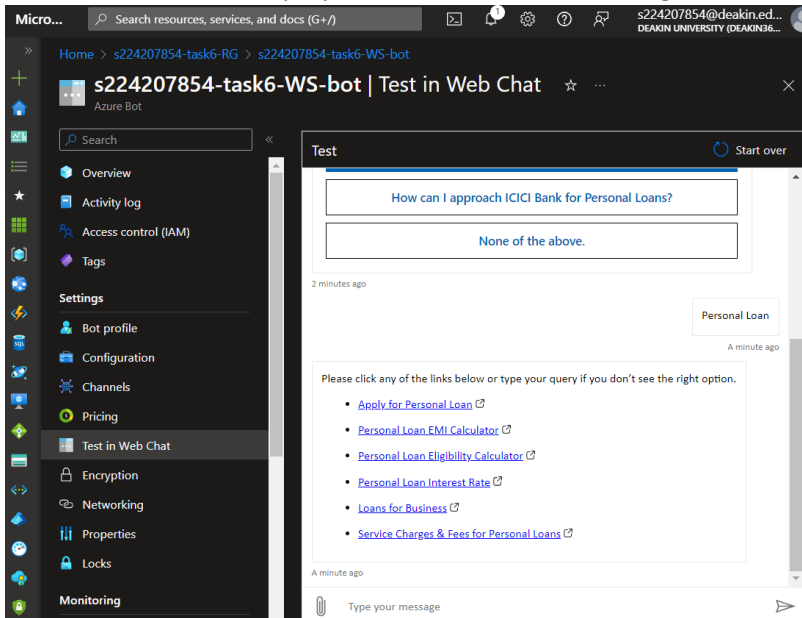
Create a GroupMe application

- 1) Go to the GroupMe developers' site and sign in to your account.
- 2) [Create a GroupMe application](#) for your bot.
 - a) Enter a name for your application.
 - b) For the Callback URL, enter <https://groupme.botframework.com/Home/Login>.
 - c) Enter the rest of the information requested.
 - d) Agree to GroupMe's terms of use and branding standards.
 - e) Select **Save** to complete creation of the app.
- 3) Get your app credentials

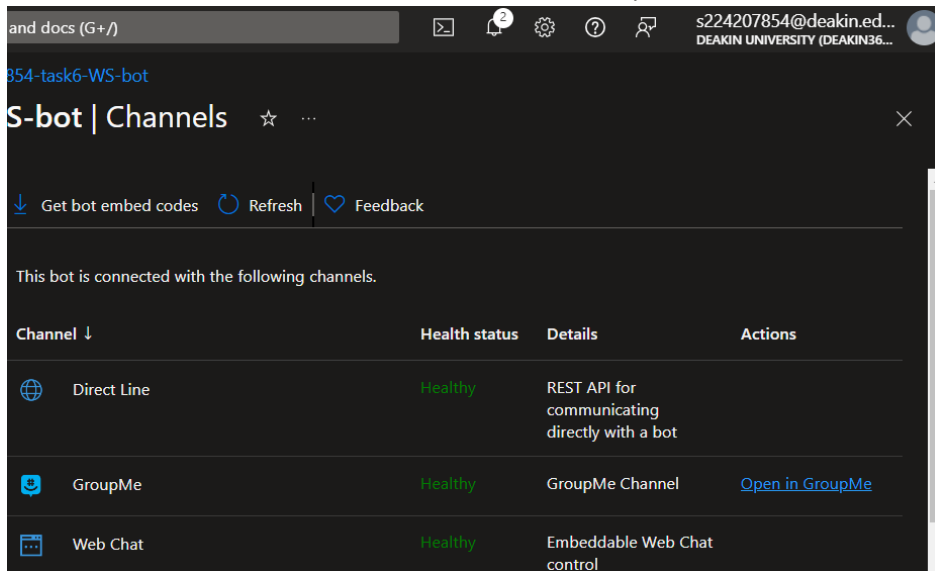


Configure bot in Azure

- 1) Create bot from 'Create a bot' button present in Deploy knowledge base page.
- 2) In custom deployment full-up the details as per required. In Language Resource Key, please provide subscription key that has been used by our SDK.
- 3) Once deployment is completed, got-to the BOT.
 - a) BOT which has been deployed can be tested via UI using 'Test in Web Chat' feature present in Azure bots.



- b) Open Channels and select GroupMe. Enter the required information.
- c) Once done, we can check the added channels in our portal

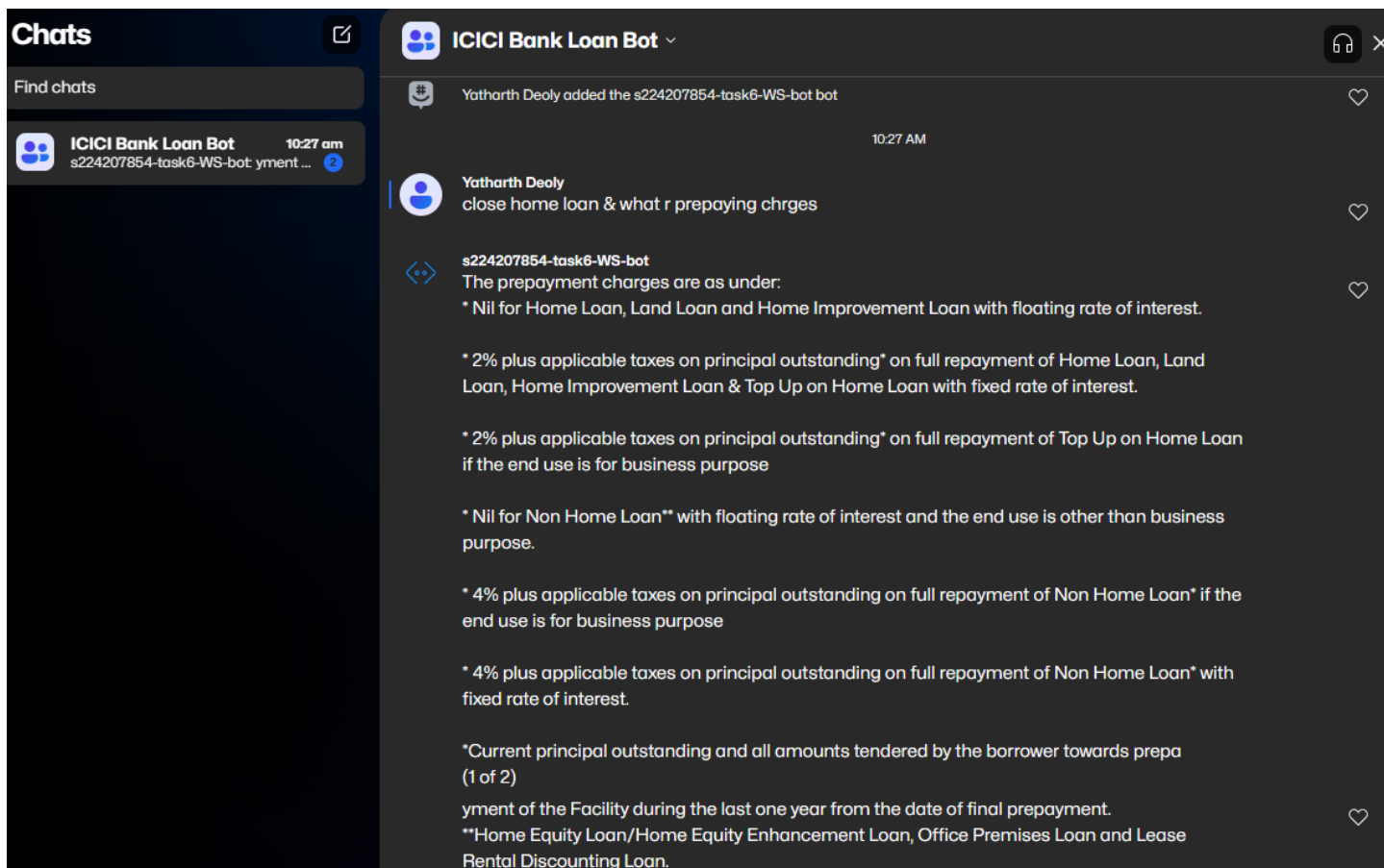


BOT Testing

Once all configuration is done, please go-to your [GroupMe](#) application, and select your created BOT.

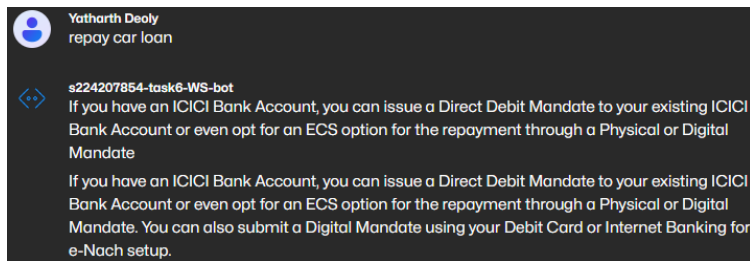
Test-1

Positive case scenario tested



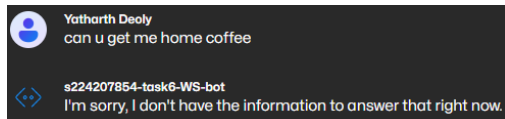
Test-2

Positive case scenario tested



Test-3

Negative case scenario tested



Resource Creation Clean-up

Once task has been completed, will start resource clean-up activity. We can directly clean the resource from our “Resource groups” section, which will remove all resources created under this group, like bot, language service, storage, workspace and more.

Summary

In this task we have learnt about natural language processing using BOT service and the usage of Azure with language service. Using pre-defined libraries to predict answers for different questions.

Bibliography

Learning, Great, 2024. *SIG788 - Engineering AI solutions Content*. [Online]
Available at: https://olympus.mygreatlearning.com/courses/109578?module_id=747613
[Accessed 18 April 2024].

Microsoft, 2022. *Configure a bot to run on one or more channels*. [Online]
Available at: <https://learn.microsoft.com/en-us/azure/bot-service/bot-service-manage-channels?view=azure-bot-service-4.0>
[Accessed 18 April 2024].

Microsoft, 2024. *Connect a bot to GroupMe*. [Online]
Available at: <https://learn.microsoft.com/en-us/azure/bot-service/bot-service-channel-connect-groupme?view=azure-bot-service-4.0>
[Accessed 18 April 2024].