

Deakin University

SIG788- OnTrack Submission

Task 5.1 C

Submitted by

Neethu Sidhardhan
S223494027
Attempt # 2
6/5/2023

Target Grade: C

Task Details -

- 1) You need to select a domain and application and build a Custom Question Answering Bot using Azure Cognitive Services Language Studio, deploy it to Azure and attach it to a channel like Facebook.
- 2) You need to explain cell by cell of your code and provide us with the screenshot of your code running and the results.

Application & Domain

We have considered the Food & Services domain under Supply Chain Application for developing question and answering Bot. For this exercise we have used Pizza ordering Bot, where we get all the information regarding the Pizza's , their size, their ingredients, how to get reward points upon ordering Pizzas. For doing this, we have used question and answers related to Dominos Pizza, Pizza Hut, Papa John, California Pizza Kitchen. This is most common bot that we used in our daily life. We have similar types of Bot with all Food supply chain related for delivering, ordering as well as enquiring.

- ❖ Use the hyperlink: <https://language.cognitive.azure.com> for language studio.

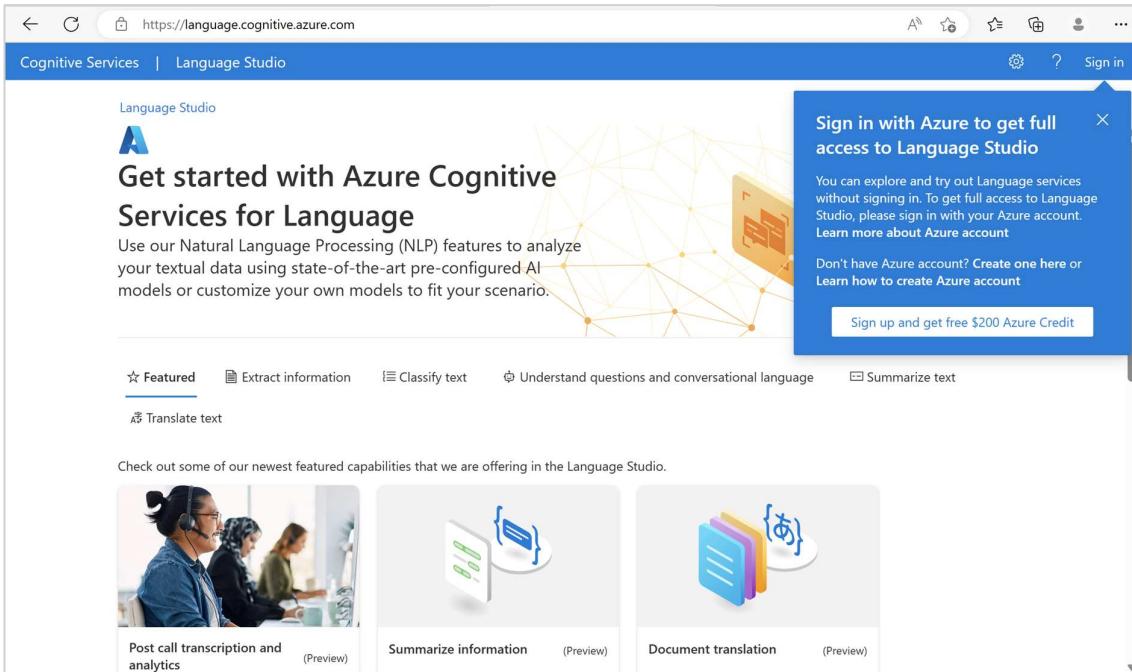


Fig 1 : Azure Cognitive Language studio home page

- ❖ Use Deakin login credentials to sign in to language studio
- ❖ Select Azure resource group or create a new resource group in Azure

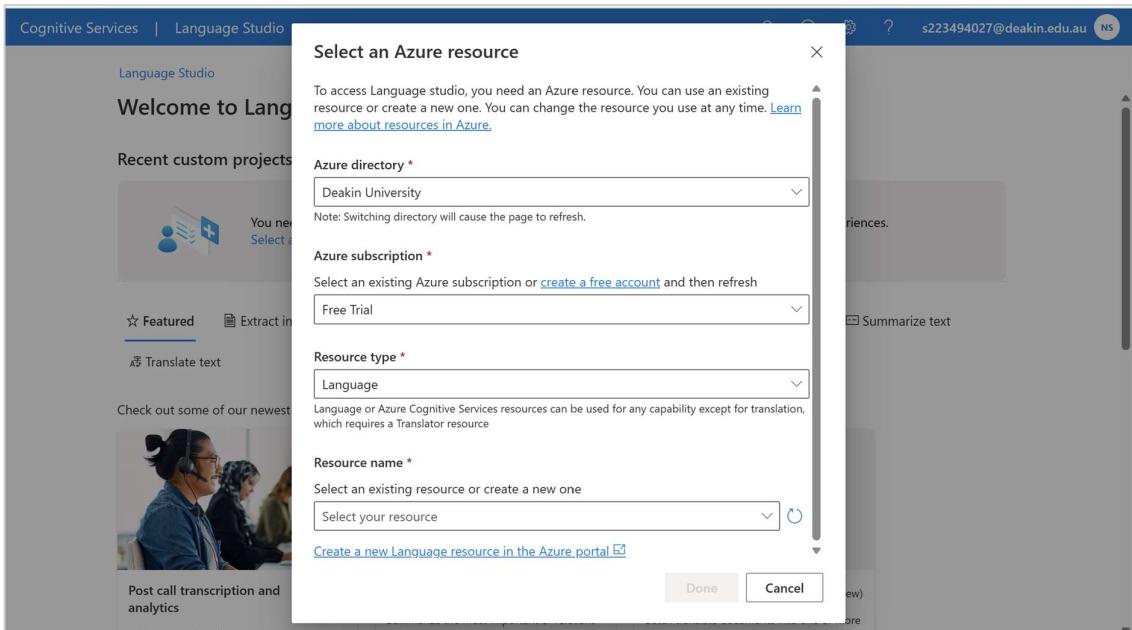


Fig 2: Select Resource group

- ❖ Once the resource group is created. Select the option “Language Service” for creating cognitive language services.

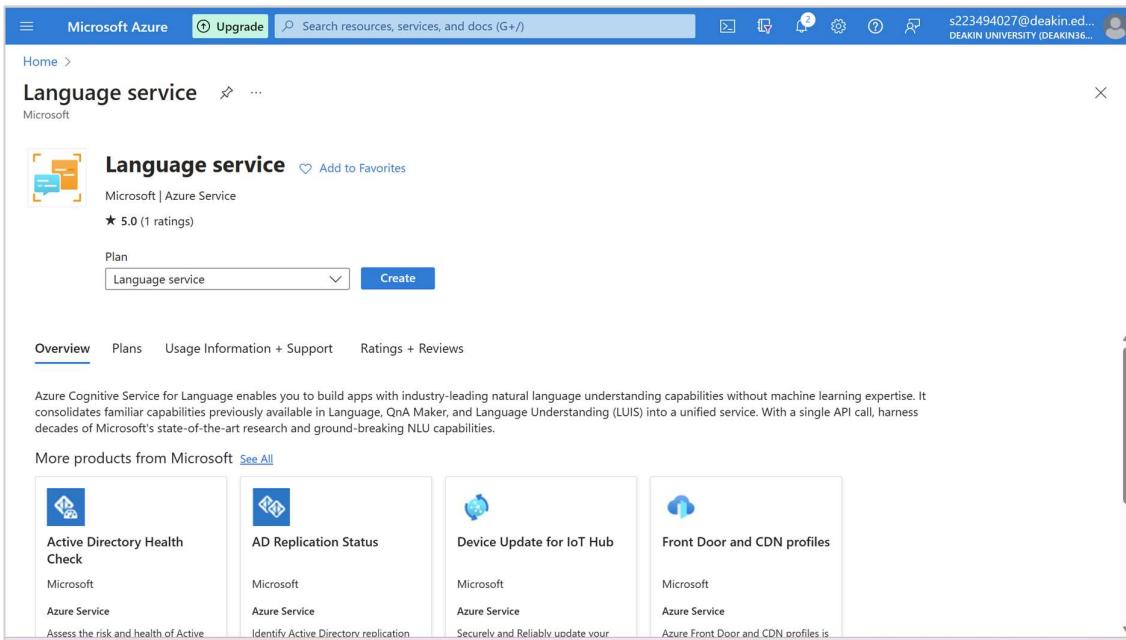


Fig 3: Language Services - Initial Screen

- ❖ Select the additional features and custom features in language services. We have the custom services like question answering and text classification. Before proceeding by creating the resource, we need to select the features.

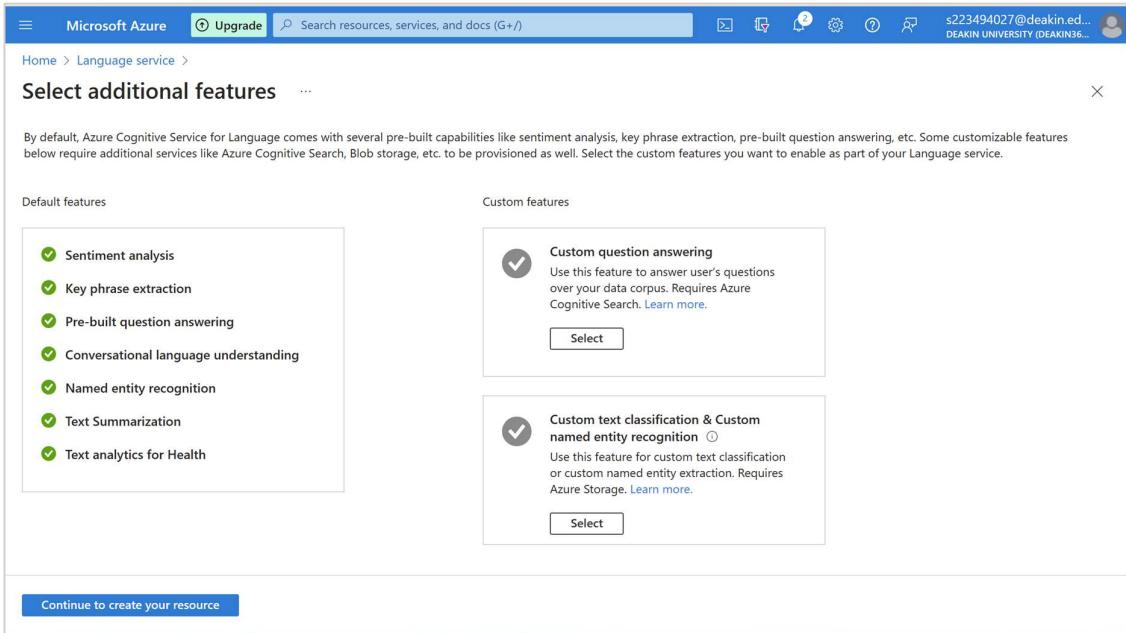


Fig 4: Select additional features before selection

- ❖ We have selected both the features and clicked the option “Continue to create your resource”

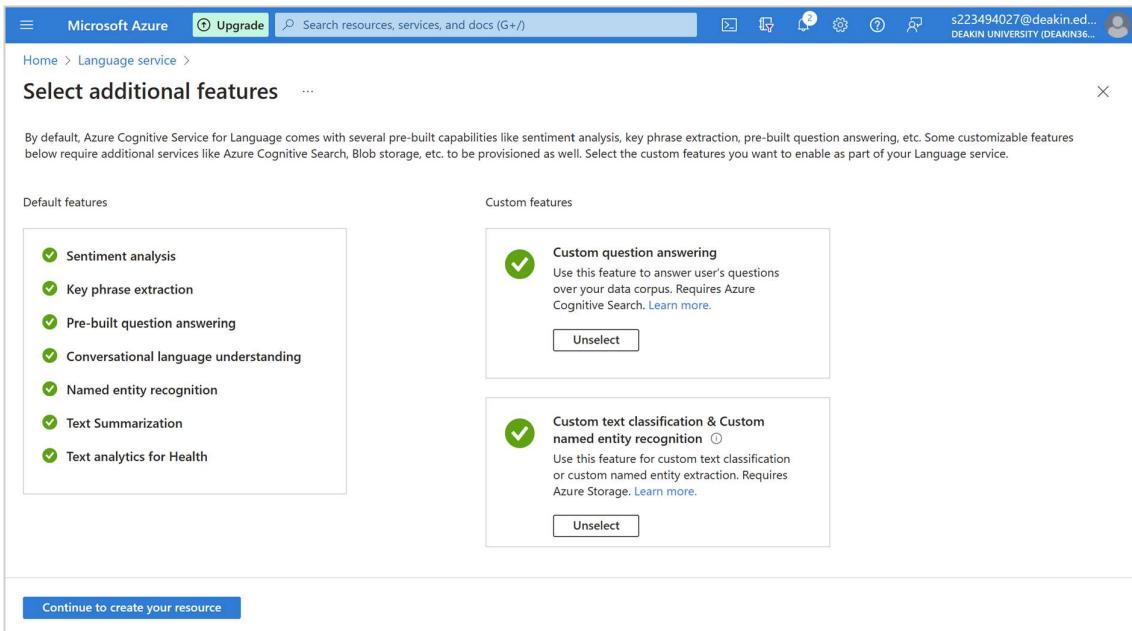


Fig 5: Selecting additional feature - Post selection

- ❖ We have to create the language. The form will be asked to mention the subscription details and resource group. If there are no resource groups available, we can create a new resource group here.

Fig 6: Create Language (Initial Screen)

- ❖ We have mentioned the subscription details as “Free Trail” along with Resource group. Here, the region is selected as “East US”. We have to provide a name for the language. Here, we have provided it as “Text Analysis” along with price tier. In this page we have the option to choose or create the storage account.

Create Language

Project Details

- Subscription *
- Resource group * [Create new](#)

Instance Details

- Region
- Name *
- Pricing tier *

Custom question answering

Custom question answering lets you answer user's questions over your data corpus. You can extract questions and answers from your data, customize them and create a knowledge base. The knowledge base is stored in an Azure

[View full pricing details](#)

[Review + create](#) [< Previous](#) [Next : Network >](#) [Give feedback](#)

Fig 7: Create Language - Form 1

- ❖ Here we have to accept the responsible AI Notice shared by Microsoft. There are three documentations that need to be read and accepted.

Create Language

New/Existing storage account *

- New storage account
- Existing storage account

Storage account name *

Storage account type *

[Learn more about storage account types](#)

Responsible AI Notice

Microsoft provides technical documentation regarding the appropriate operation applicable to this Cognitive Service that is made available by Microsoft. Customer acknowledges and agrees that they have reviewed this documentation and will use this service in accordance with it.

[Responsible Use of AI documentation for Text Analytics for Health](#)

[Responsible Use of AI documentation for PII](#)

[Responsible Use of AI documentation for Language](#)

By checking this box I certify that I have reviewed and acknowledge the terms in

[Review + create](#) [< Previous](#) [Next : Network >](#) [Give feedback](#)

Fig 8: Create Language - Form 2

- ❖ Once we click the option “Review + create”. Here, they system will process a validation process to check if there is any conflict in any of the information that we have opted.

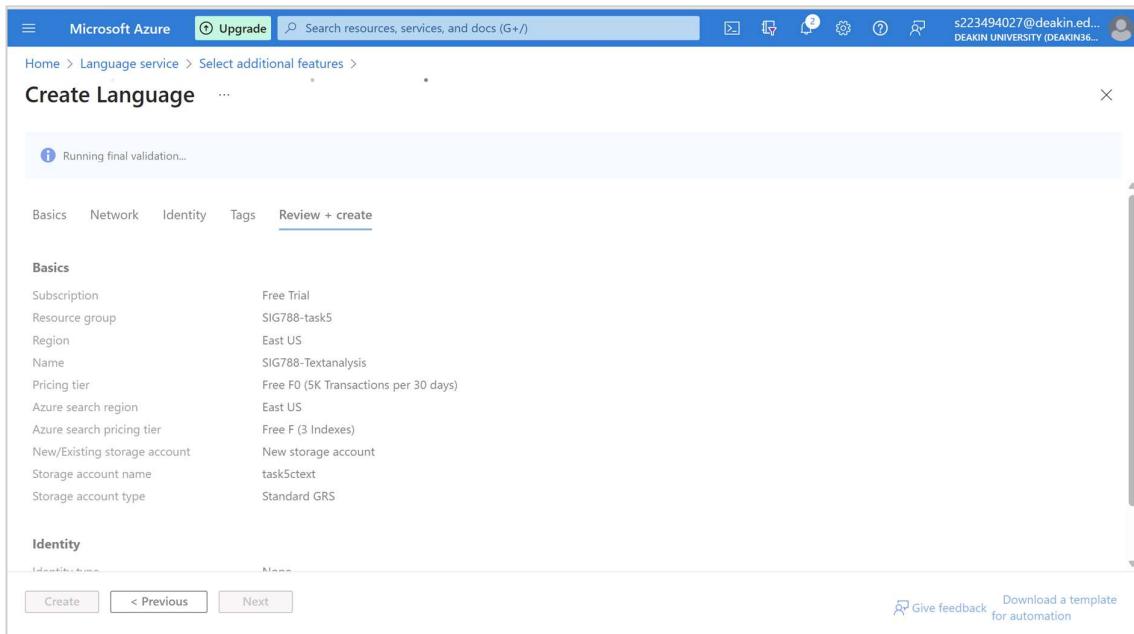


Fig 9: Create Language - Review

- ❖ Once the review validation process is completed, we will be getting message like "Validation Passed". Then we can successfully create the language group.

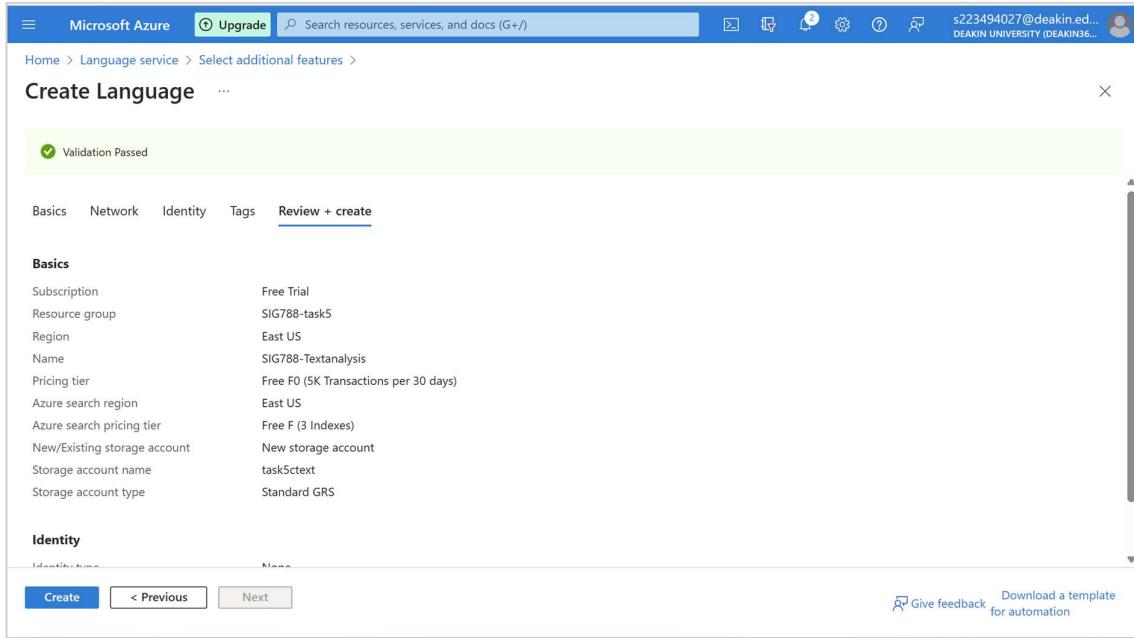


Fig 10: Create Language- Validation Successful Message

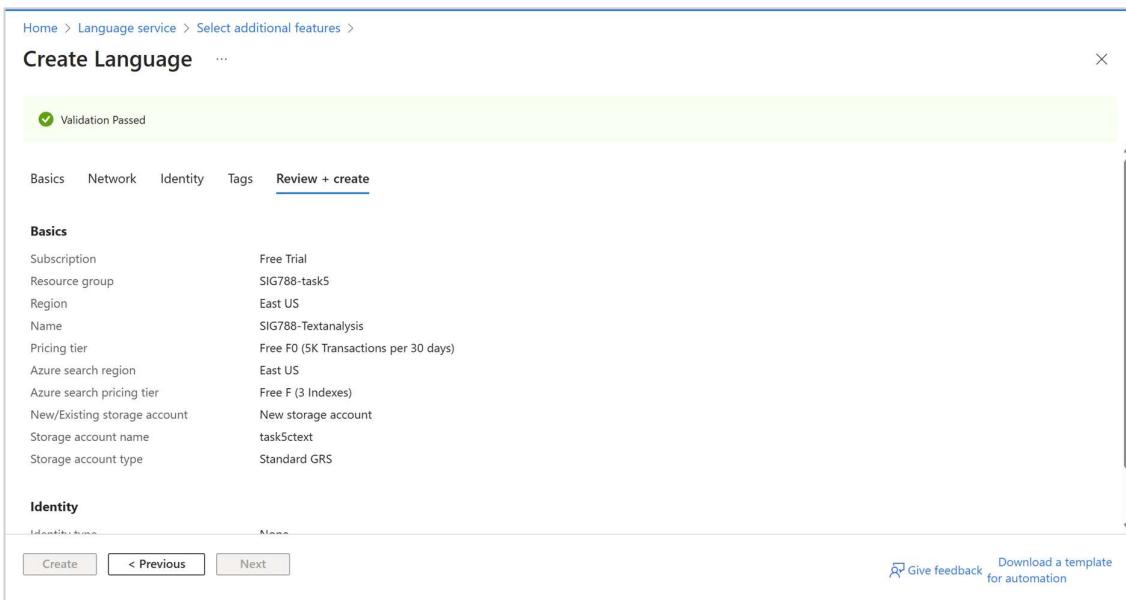


Fig 11: Create Language

- ❖ Once the create option is selected, we will be getting the message of deployment of the language group that we have created.

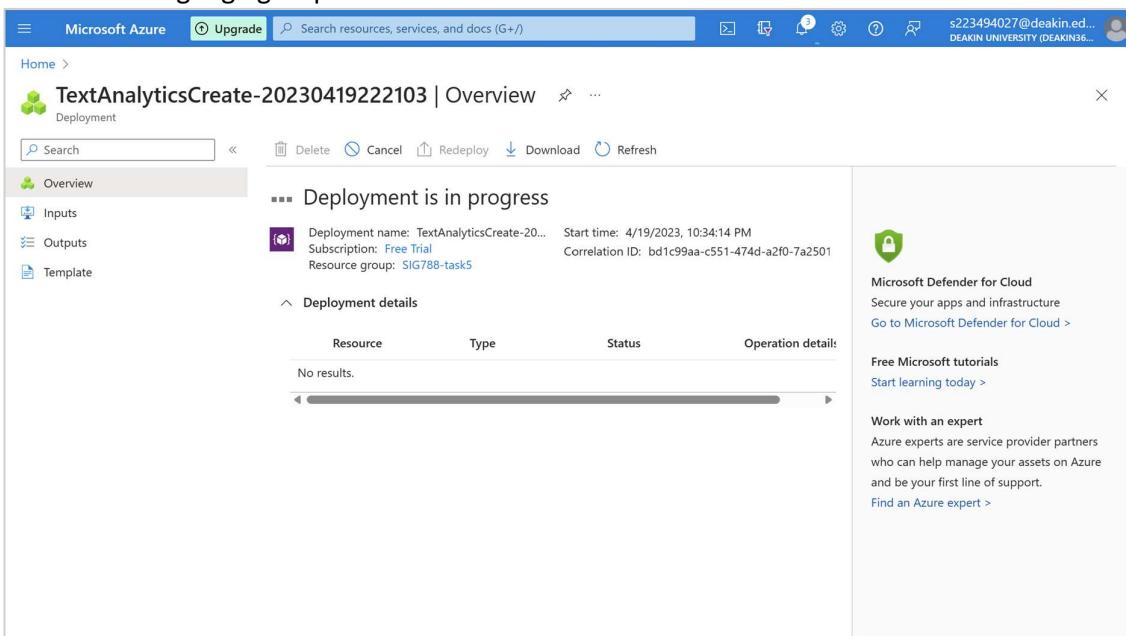


Fig 12: Deployment in Progress

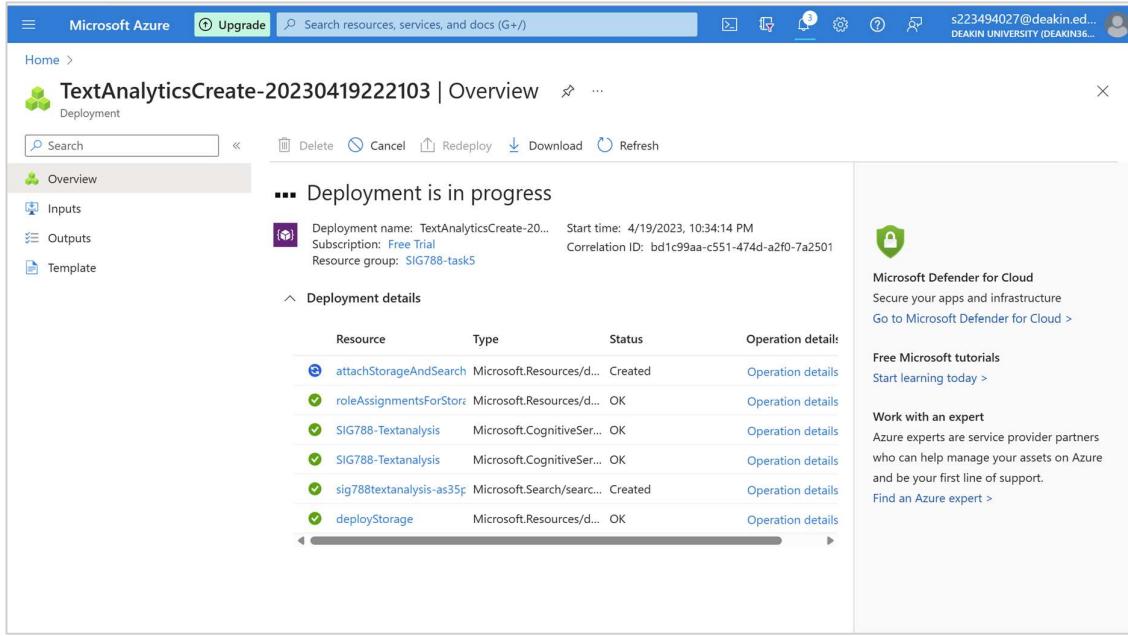


Fig 13: Deployment completed

- ❖ Once the deployment is successfully completed, we will be getting deployment details of the resource group that we have created. We will be getting a tick mark if all the groups are successfully created.
- ❖ In initial screen <https://language.cognitive.azure.com> we had started filling up the forms for mapping the resources to language studio. We will update the newly created resource group in the form.

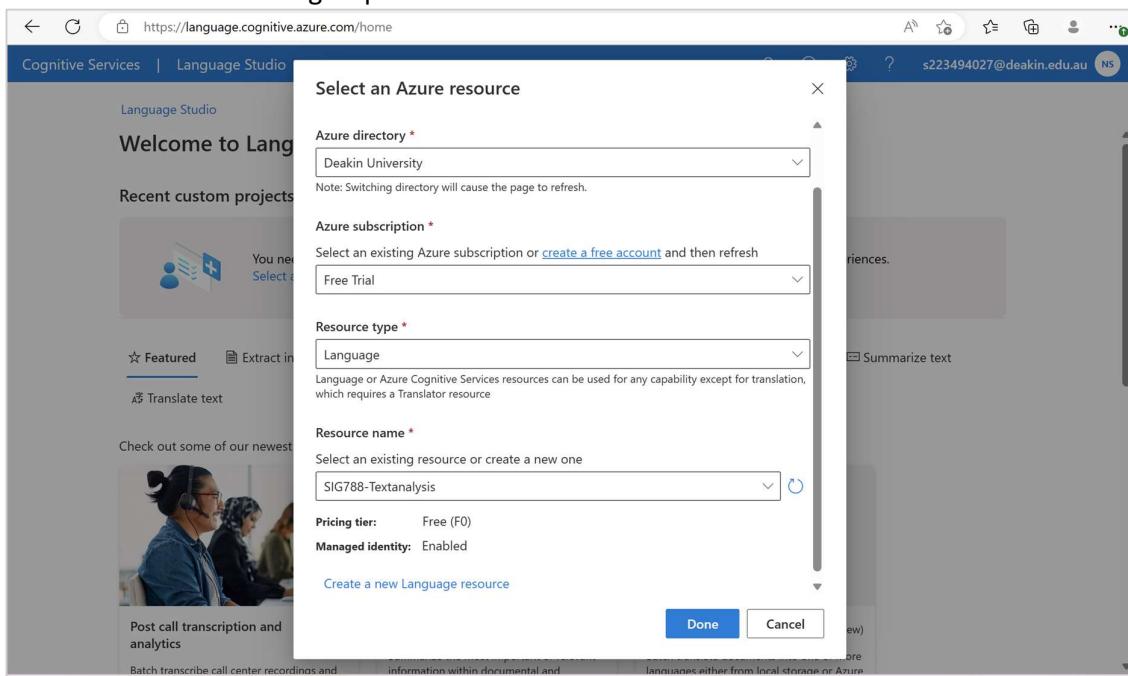


Fig 14: Mapping the resource group

- ❖ After the form is processed, we will be getting the below screen where we can select the option “Conversation Language understanding” as well as “Custom questioning answering “option.

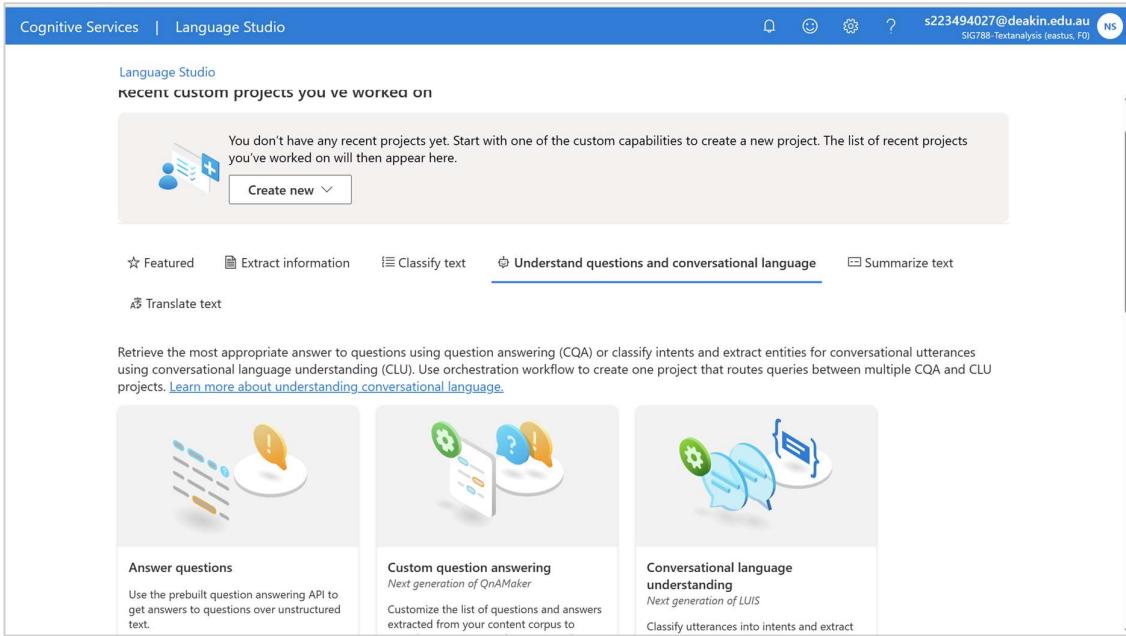


Fig 15: Language Studio initial screen

- ❖ Select the option “Conversational language understanding “from the tab “understand questions and conversational language”

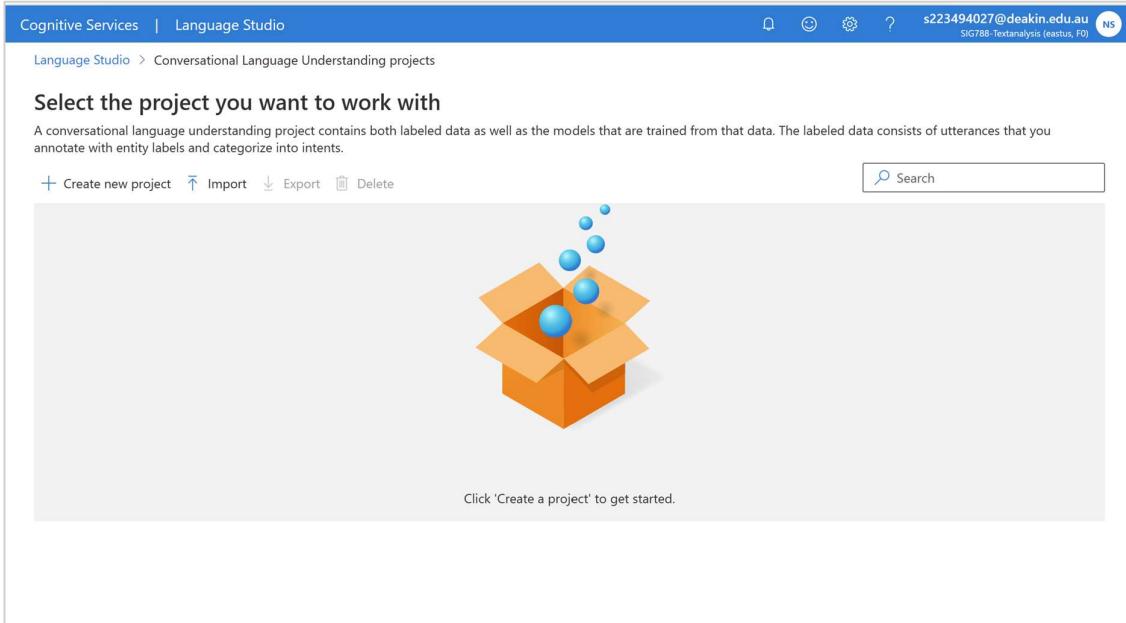


Fig 16: CLU initial screen

- ❖ Here we can either create a new project or we can import JSON file. Here we have used JSON file from the sample projects loaded in GitHub. The sample file used is related to Pizza online shopping.

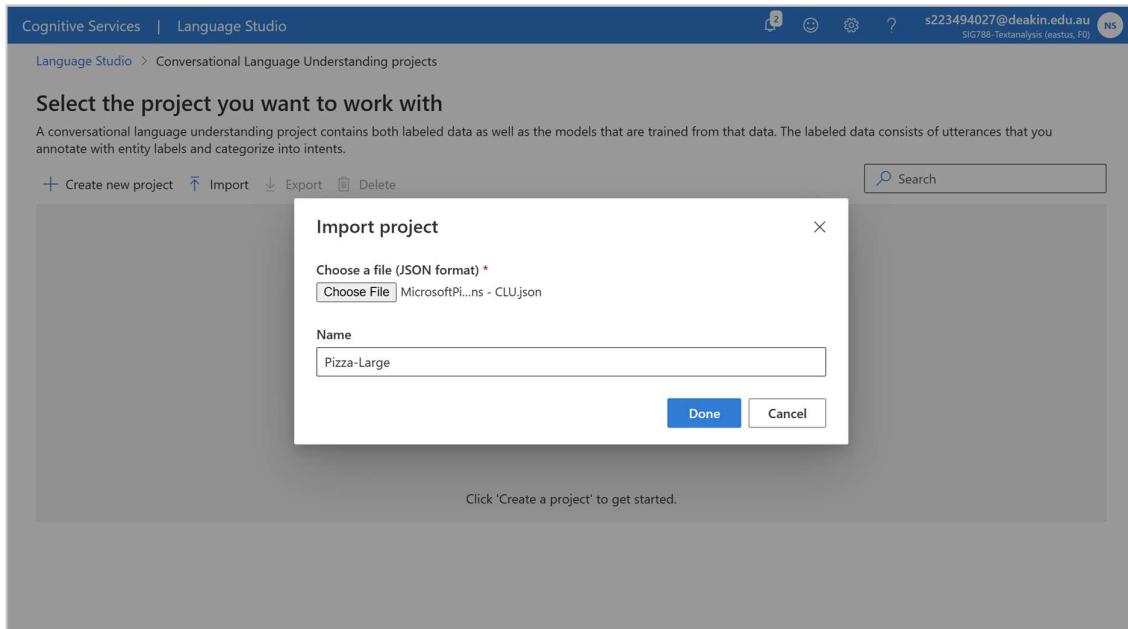


Fig 17: Import project

- ❖ Once the project is loaded successfully, we get the schema definition. Under the schema we can define the intents and entities

Intent	Labeled utterances	Entities used with this intent
ModifyOrder	90	FullPizzaOrder
None	33	

Fig 18: CLU Schema definition

- ❖ We can create an Intent and Utterances under data labelling. Here, we have selected the option “Modify” to create the intent and utterances. Under the activity pane, we have to add the entity. Here we have created the entity as “Full Pizza Order”

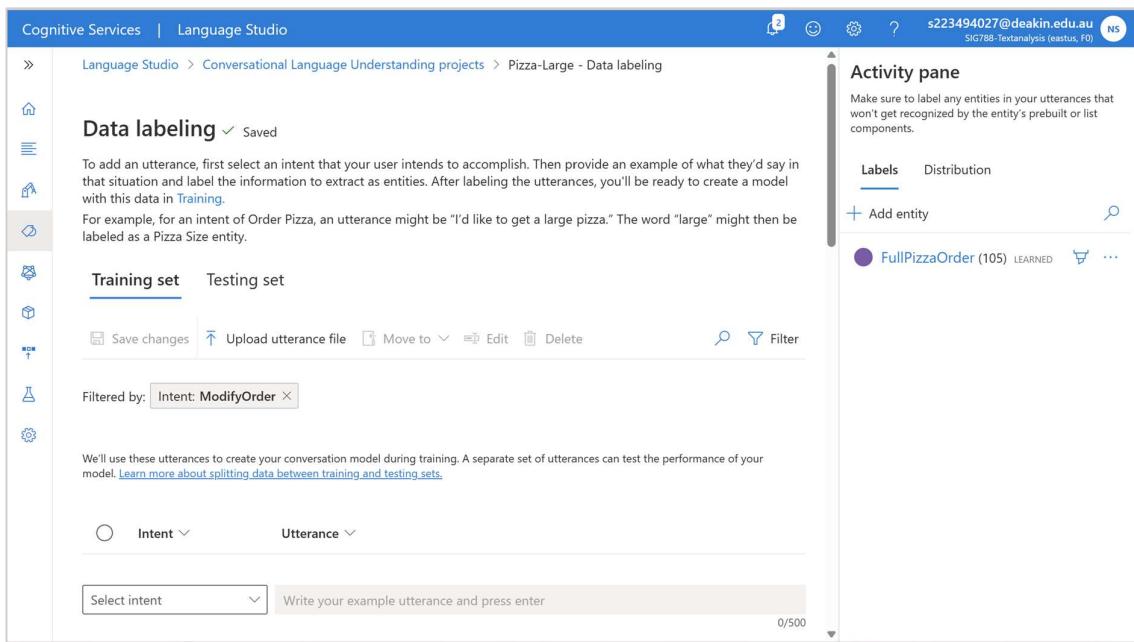


Fig 19: Data Labelling

Fig 20: Intent and Utterances along with Entity

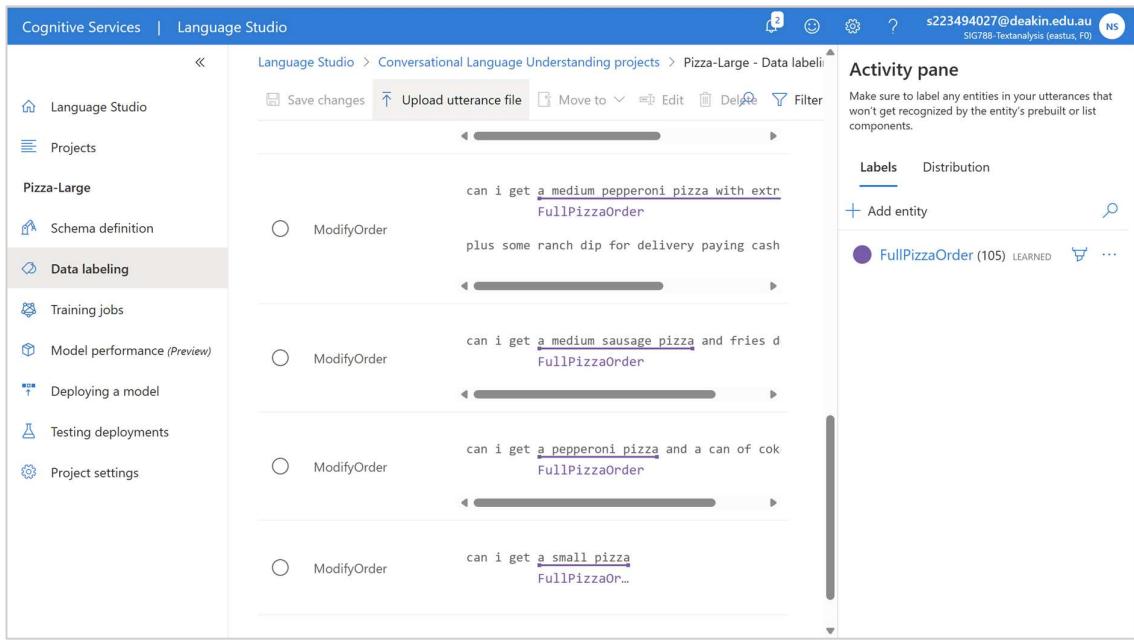


Fig 21: Data Labelling -2

- ❖ Once the intent and utterance are ready, we can train the model. From the left side pane in language studio, we have the option “Training Jobs”.

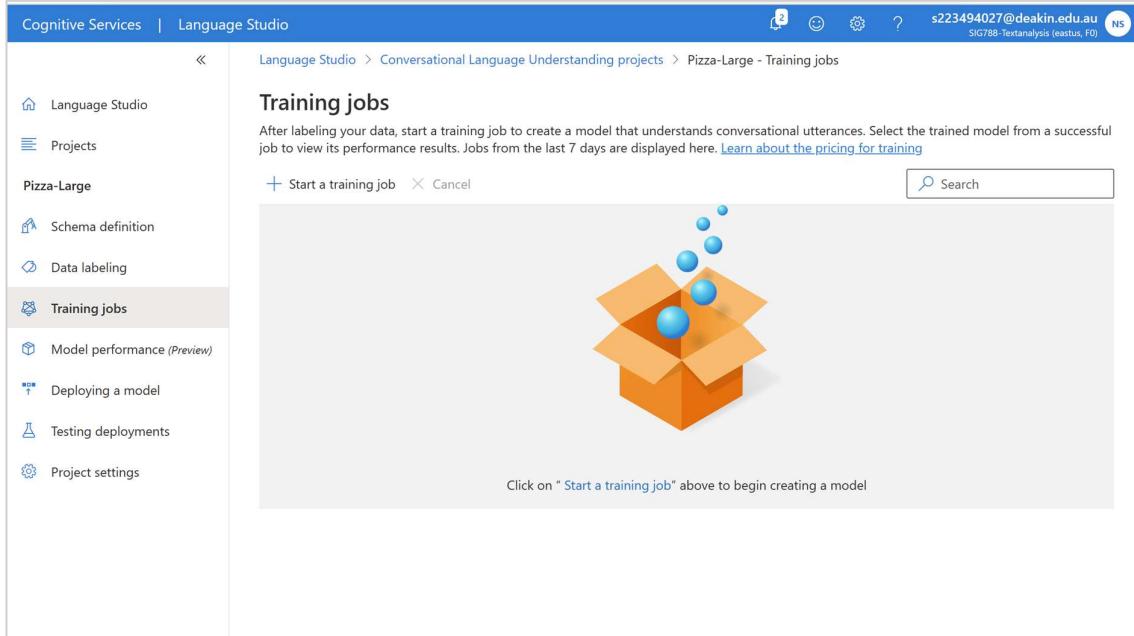


Fig 22: Training Job Initial Screen

- ❖ Click on “Start a training job”. We will be getting the below form. Here we have to mention the model’s name to training along with the training mode. We have used the standard training as mode of training. The standard data split is 80:20. If any custom changes is required that can be updated in the form.

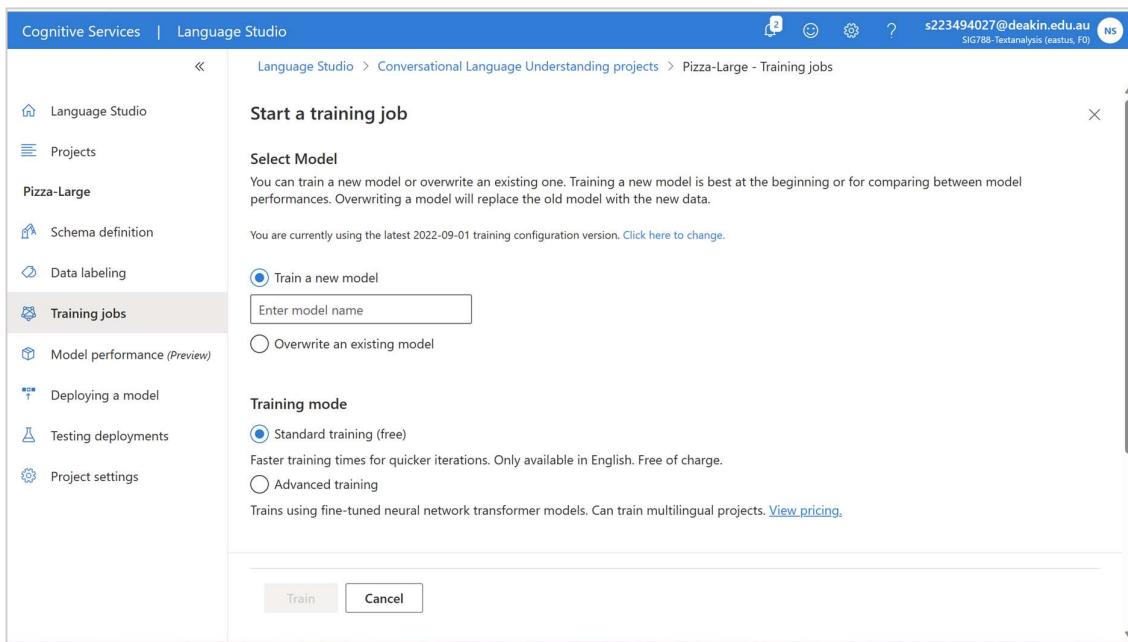


Fig 23: Training Job - Form

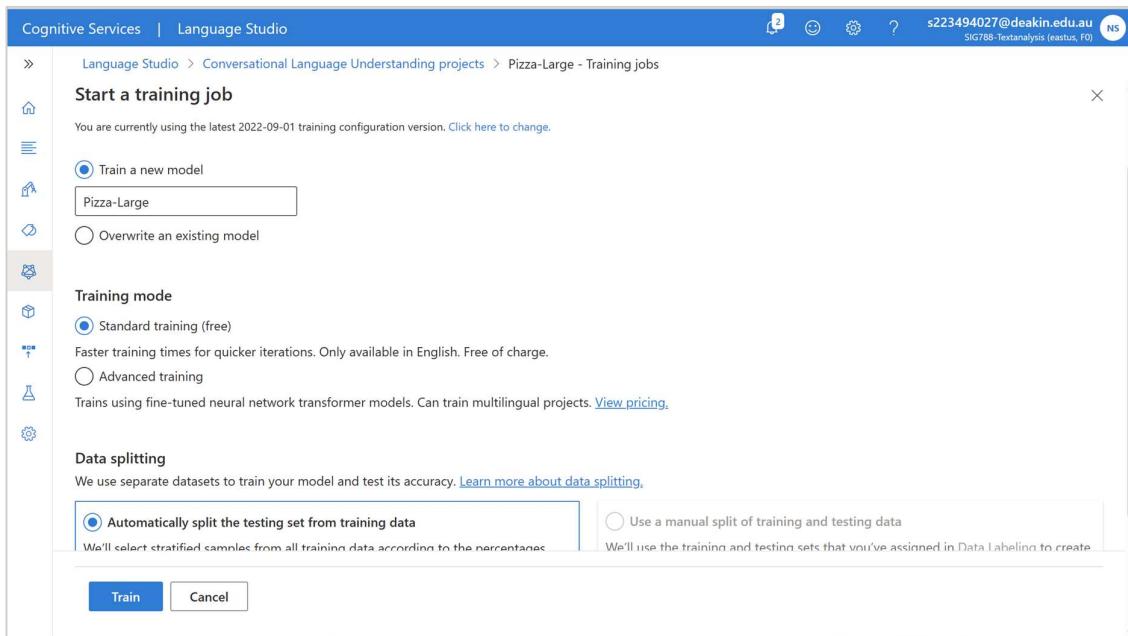


Fig 24: Training Job filled form part 1

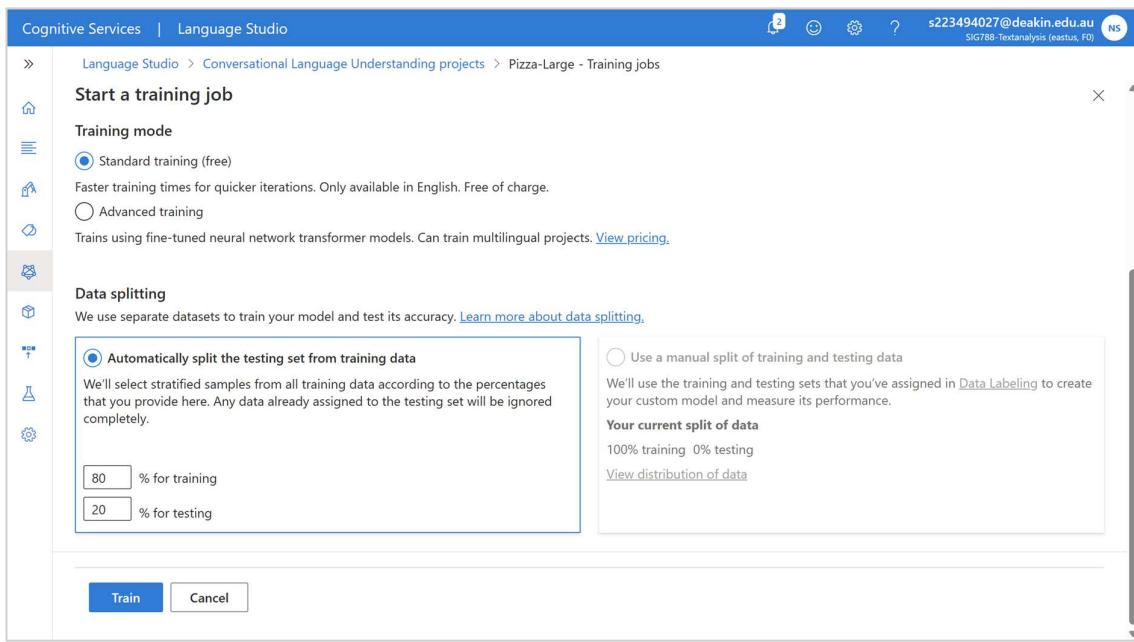


Fig 25: Training Job filled with data split

❖ Once the form is filled, click the option “Train”.

Training job id	Status	Output model	Submitted
88975c67-b552-45fd-b1f2-09a1f108fa7...	Job in queue. Starting soon ...	pizza-large	4/19/2023

Training jobs

After labeling your data, start a training job to create a model that understands conversational utterances. Select the trained model from a successful job to view its performance results. Jobs from the last 7 days are displayed here. [Learn about the pricing for training](#)

Start a training job **Cancel** **Search**

Fig 26: Train the model

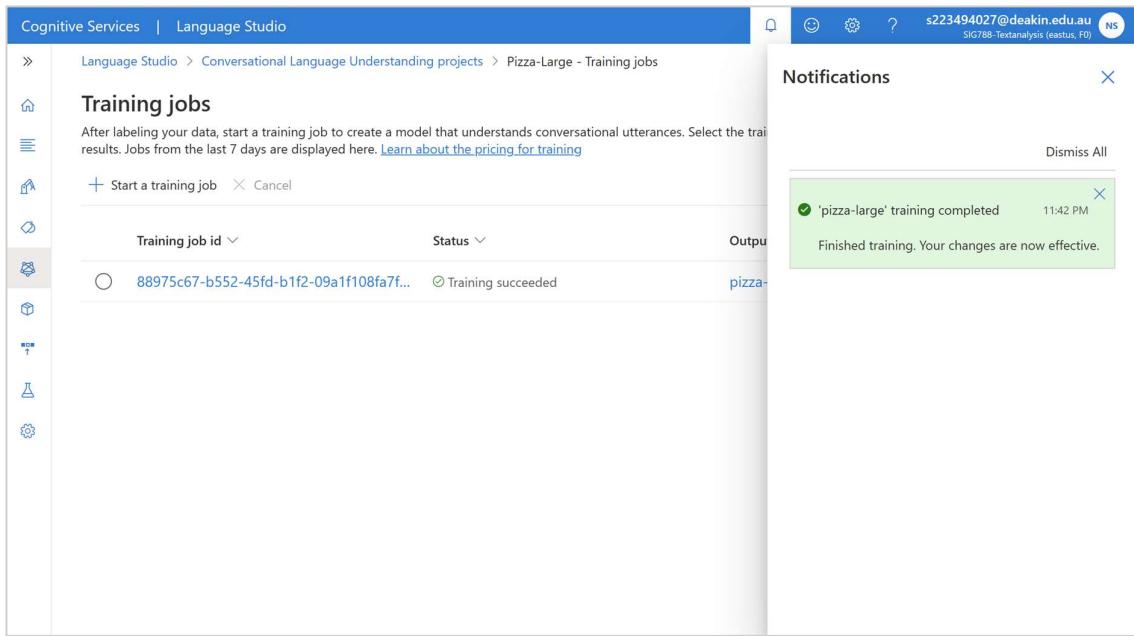


Fig 27: Training - Successfully Completed

- ❖ If the training job ID, is selected, we can get all the training job details.

This screenshot is similar to Fig 27 but includes a detailed modal window on the right labeled 'Training job details'. The modal lists several parameters for the training job:

- Training job ID:** 88975c67-b552-45fd-b1f2-09a1f108fa7f_638174592000000000
- Submitted:** 4/19/2023, 11:42:25 PM
- Total time:** 0 hours, 0 minutes, 36 seconds
- Job status:** Succeeded
- Training mode:** Standard training (free)
- Training progress:** Progress: 100% complete
Time elapsed: 0 hours, 0 minutes, 5 seconds
- Evaluation progress:** Progress: 100% complete
Time elapsed: 0 hours, 0 minutes, 27 seconds

 A 'Close' button is at the bottom right of the modal.

Fig 28: Details of Train data

- ❖ Once the model is trained, we have to deploy the model. To deploy the model, we use the option “Deploy a Model” from the left pane.

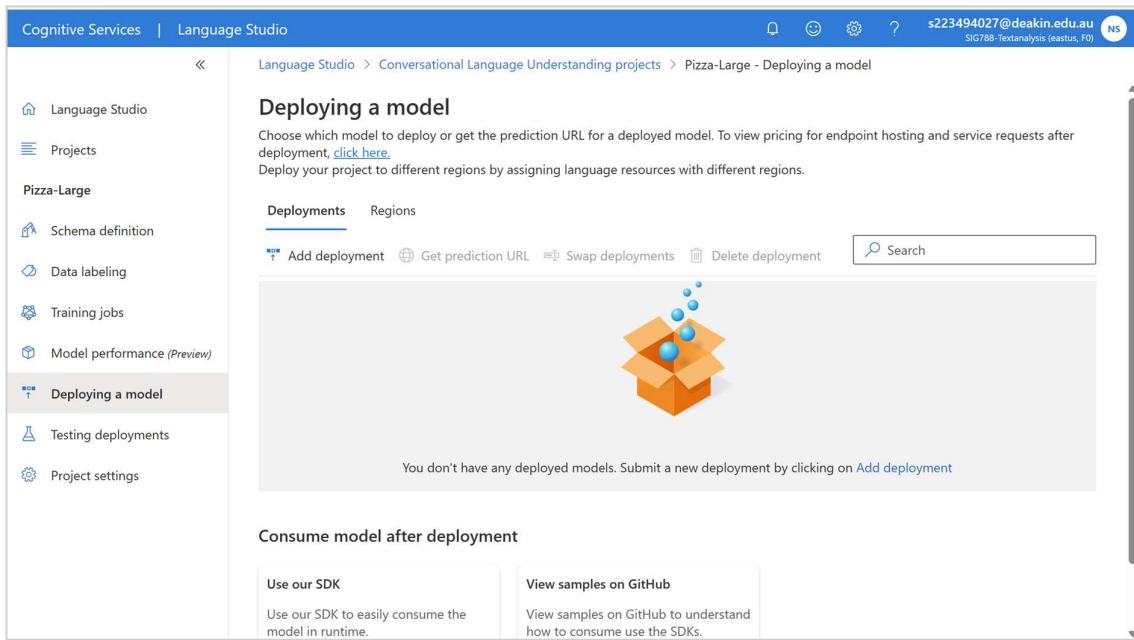


Fig 29: Deploy a model (Initial Screen)

❖ Select the option “Add deployment”.

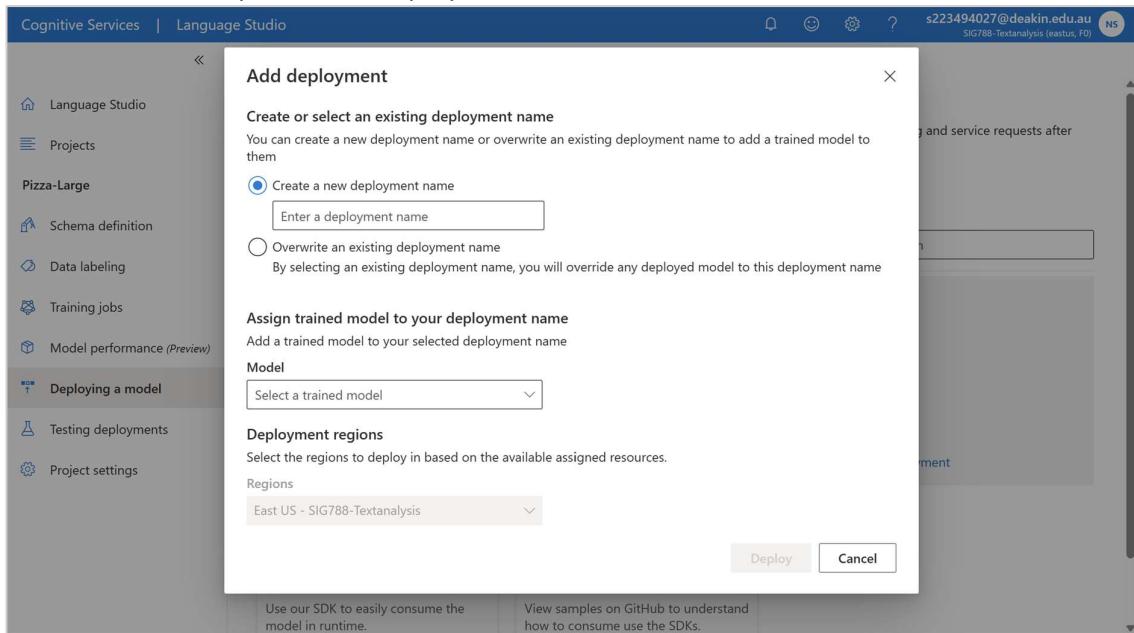


Fig 30: Add deployment – Initial page

- ❖ In this deployment form, we need to create a new deployment name. Here we have used the name “pizza-large pan” as the name of model that we are going to deploy. Assign the model that we have created while training the model.

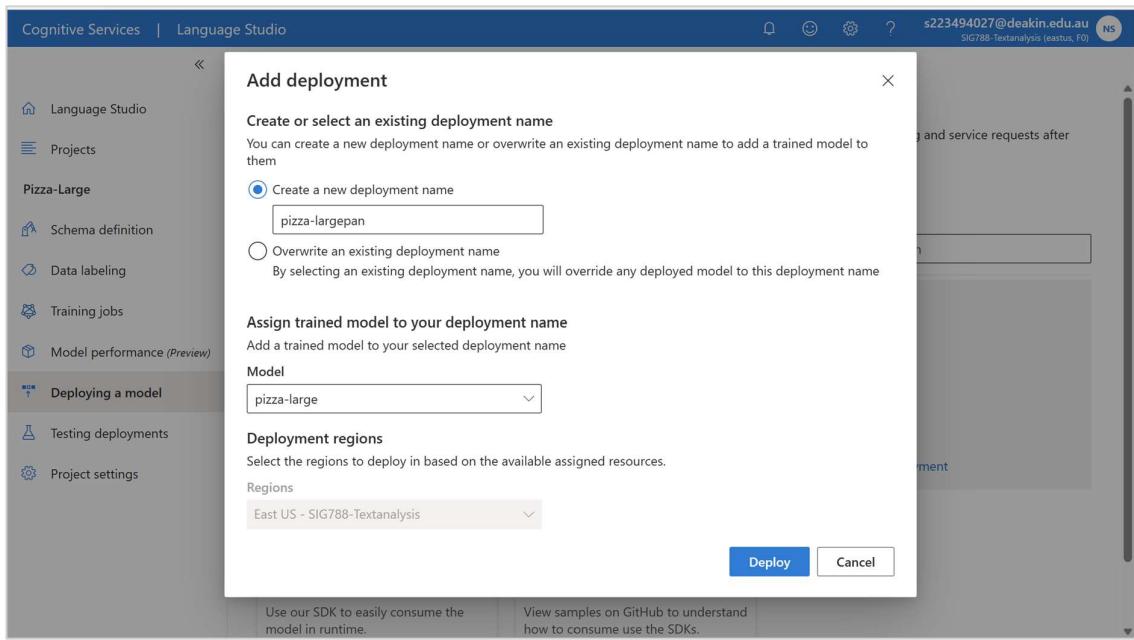


Fig 31: updated deployment form

❖ Select the option “Deploy”.

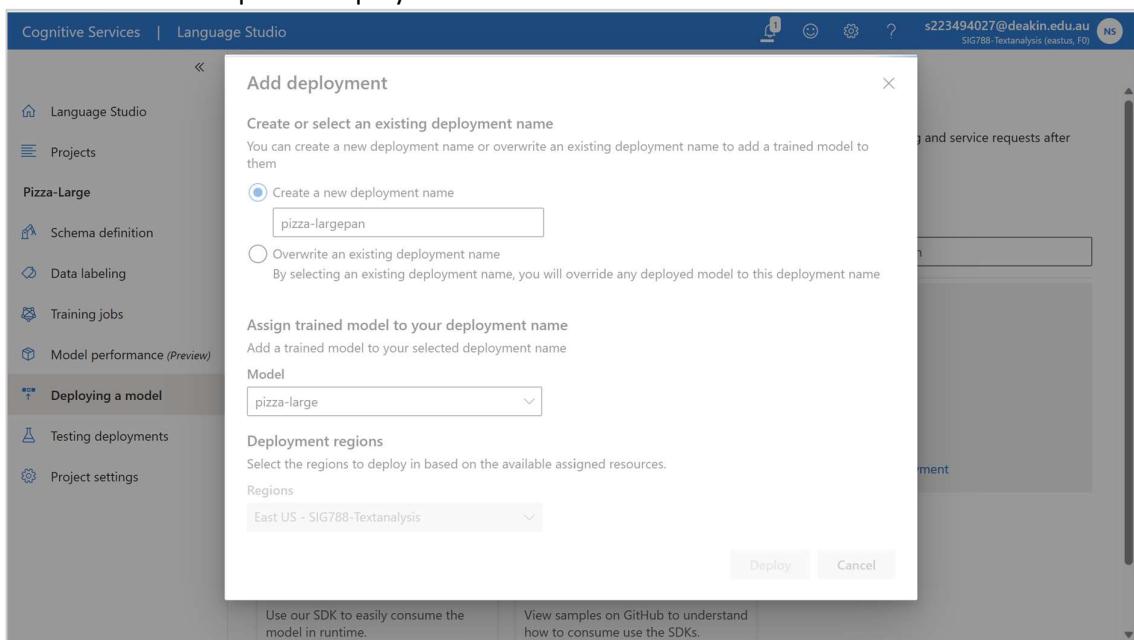


Fig 32: Select Deploy

The screenshot shows the Azure Cognitive Services Language Studio interface. The left sidebar has a tree view with 'Language Studio', 'Projects', 'Pizza-Large' (selected), 'Schema definition', 'Data labeling', 'Training jobs', 'Model performance (Preview)', 'Deploying a model' (selected), 'Testing deployments', and 'Project settings'. The main content area shows the 'Deploying a model' section for the 'Pizza-Large' project. It includes a navigation bar with 'Language Studio > Conversational Language Understanding projects > Pizza-Large - Deploying a model'. Below this is a heading 'Deploying a model' with a sub-instruction: 'Choose which model to deploy or get the prediction URL for a deployed model. To view pricing for endpoint hosting and service requests after deployment, [click here](#)'. There's also a note: 'Deploy your project to different regions by assigning language resources with different regions.' A 'Deployments' tab is selected, showing a table with one row:

Deployment name	Model name	Deployed on	Deployment expires	Region
pizza-largepan	pizza-large	4/19/2023	8/31/2024	East

Below the table is a section titled 'Consume model after deployment' with two options: 'Use our SDK' (with a 'Get SDK' button) and 'View samples on GitHub' (with a 'SDK samples' dropdown).

Fig 33: Deploying the model

- ❖ Once the model is successfully deployed, we can test the deployment. To test the deployment, we use the option “Testing deployment” from the left pane.

The screenshot shows the Azure Cognitive Services Language Studio interface. The left sidebar has a tree view with 'Language Studio', 'Projects', 'Pizza-Large' (selected), 'Schema definition', 'Data labeling', 'Training jobs', 'Model performance (Preview)', 'Deploying a model', 'Testing deployments' (selected), and 'Project settings'. The main content area shows the 'Testing deployments' section for the 'Pizza-Large' project. It includes a navigation bar with 'Language Studio > Conversational Language Understanding projects > Pizza-Large - Testing deployments'. Below this is a heading 'Testing deployments' with a sub-instruction: 'Test a deployment by providing a sample utterance to find out what intent and entities get recognized. These are the same predictions as when making API calls against your model in code. [Learn about the info in these API requests and responses](#)'.

The 'Run the test' button is visible. Below it is a 'Deployment name' dropdown set to 'pizza-largepan'. An 'Enter your own text, or upload a text document' input field contains the text 'order pepperoni pizza and coke'. To the right of the input field is a 'Clear text box' button. At the bottom, there are tabs for 'Result' and 'JSON', with 'Result' currently selected. A large empty text area follows.

Fig 34: Test deployment

The screenshot shows the Microsoft Cognitive Services Language Studio interface. On the left, there's a sidebar with options like Language Studio, Projects, Pizza-Large, Schema definition, Data labeling, Training jobs, Model performance (Preview), Deploying a model, Testing deployments (which is selected), and Project settings. The main area has tabs for Language Studio, Conversational Language Understanding projects, and Pizza-Large - Testing deployments. A search bar at the top says "pizza-largepan". Below it is a text input field containing "order pepperoni pizza and coke". The result section is titled "Result JSON" and shows an "Intent" card for "Top intent" which is "ModifyOrder" with a confidence of 97.00%. It also shows an "Entities" section with no predicted entities and the original text "order pepperoni pizza and coke".

Fig 35: Test Results -1

- ❖ To do the test, we can enter own text or we can upload a text document. Here we have used own text like “Can I get 3 peperoni pizzas and four cheese pizza with large house salad and large fires”.
- ❖ We will be getting the result as Intent and Entity. We will be getting the confidence % for the test message.

This screenshot shows the same interface as Fig 35, but with a different test message: "can i get 3 pepperoni pizzas and a four cheese pizza with a large house salad and a large fries". The "Intent" card now shows "Top intent" as "ModifyOrder" with a confidence of 100.00%. The "Entities" section displays three cards: "FullPizzaOrder" with confidence 100.00% (3 pepperoni pizzas and a four cheese pizza), "FullPizzaOrder" with confidence 100.00% (a large house salad), and "FullPizzaOrder" with confidence 100.00% (a large fries).

Fig 36: Test results -2

- ❖ Once the test deployment is successfully completed. We can check the performance of the model. The left pane has the option “Model Performance (preview)”. Here we get the F1 Score, Confusion Matrix, recall and precision.

The screenshot shows the Microsoft Cognitive Services Language Studio interface. The left sidebar shows a navigation tree with 'Language Studio', 'Projects', 'Pizza-Large' (selected), and 'Model performance (Preview)' (also selected). Under 'Model performance', there are options for 'Deploying a model', 'Testing deployments', and 'Project settings'. The main content area is titled 'Model performance' and displays a table of model performance metrics. The table has columns for 'Model name', 'Intent F1 score', 'Entity F1 score', 'Training Ut...', 'Testing Ut...', and 'Model'. A single row is shown for 'pizza-large' with values: Intent F1 score 1, Entity F1 score 0.700, Training Ut... 99, Testing Ut... 24, and Model 9/1.

Fig 37: Model Performance.

The screenshot shows the Microsoft Cognitive Services Language Studio interface, specifically the 'Performance Overview' page for the 'pizza-large' model. The left sidebar is identical to Fig 37. The main content area is titled 'pizza-large' and includes tabs for 'Overview' (selected), 'Model Performance', 'Test set details', 'Dataset distribution', and 'Confusion matrix'. Under 'Model Type', it says 'Intents'. The 'Overview' section features a large blue circle with 'F1 score: 100%' and a status message 'Status: Trained successfully'. To the right, there are three circular performance indicators: a teal circle for 'Precision: 100%', a dark purple circle for 'Recall: 100%', and a smaller grey circle for 'F1 score: 100%'. Below these are detailed training statistics: 'Finished training on: 4/19/2023, 11:43:35 PM', 'Total training time: 0 hour(s), 1 minute(s), 13 second(s)', and 'Training Percentage data splitting type: Number of 99 training utterances'.

Fig 38: Performance Overview

The screenshot shows the Microsoft Language Studio interface. The left sidebar has a tree view with 'Language Studio' at the top, followed by 'Projects', 'Pizza-Large' (which is expanded), and 'Model performance (Preview)' (which is also expanded). Under 'Model performance (Preview)', there are links for 'Deploying a model', 'Testing deployments', and 'Project settings'. The main content area is titled 'pizza-large' and shows the 'Model Performance' tab selected. It includes a note: 'This is a snapshot of how your model performed during testing. The metrics here are static and tied to your model, so they won't update until you train again.' Below this is a 'Model Type' dropdown set to 'Intents'. A table follows, showing intent names, precision, recall, F1 score, training instances, and testing instances. The table data is:

Intent name ↑	Precision %	Recall %	F1 score	Training la...	Testing lab...
ModifyOrder	100.00	100.00	1	72	18
None	100.00	100.00	1	27	6

Fig 39: Model Performance

The screenshot shows the Microsoft Language Studio interface. The left sidebar has a tree view with 'Language Studio' at the top, followed by 'Projects', 'Pizza-Large' (which is expanded), and 'Model performance (Preview)' (which is also expanded). Under 'Model performance (Preview)', there are links for 'Deploying a model', 'Testing deployments', and 'Project settings'. The main content area is titled 'pizza-large' and shows the 'Test set details' tab selected. It includes a note: 'Select an utterance to review its labels and predictions for intents and entities. The information here is static and tied to your model, so it won't update until you train again. Use the filter to select which types of predictions and errors you want to view.' There are two toggle switches: 'Show errors only' (disabled) and 'Showing entity labels' (enabled). Below this is a table showing test set data. The table data is:

Model Type ↑	Text	Labeled as	Predicted as
Entity	5 party size marinera pizzas	FullPizzaOrder	No Prediction
Intent	-	ModifyOrder	ModifyOrder
> (with a circled question mark icon)	can i get a medium sausage pizza and fries delivered	FullPizzaOrder	ModifyOrder

Fig 40: Test set data

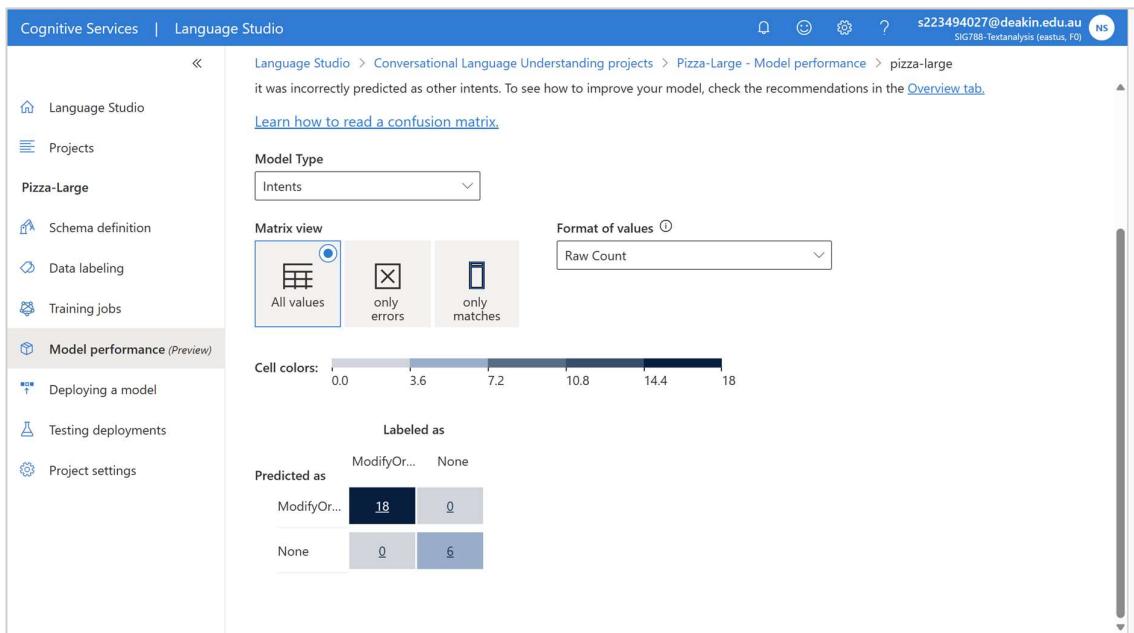


Fig 41: Confusion Matrix

❖ Here we can have different views of confusion matrix.

❖ From the language studio, we will select the option “Custom Question Answering” for creating a chat bot with Azure.

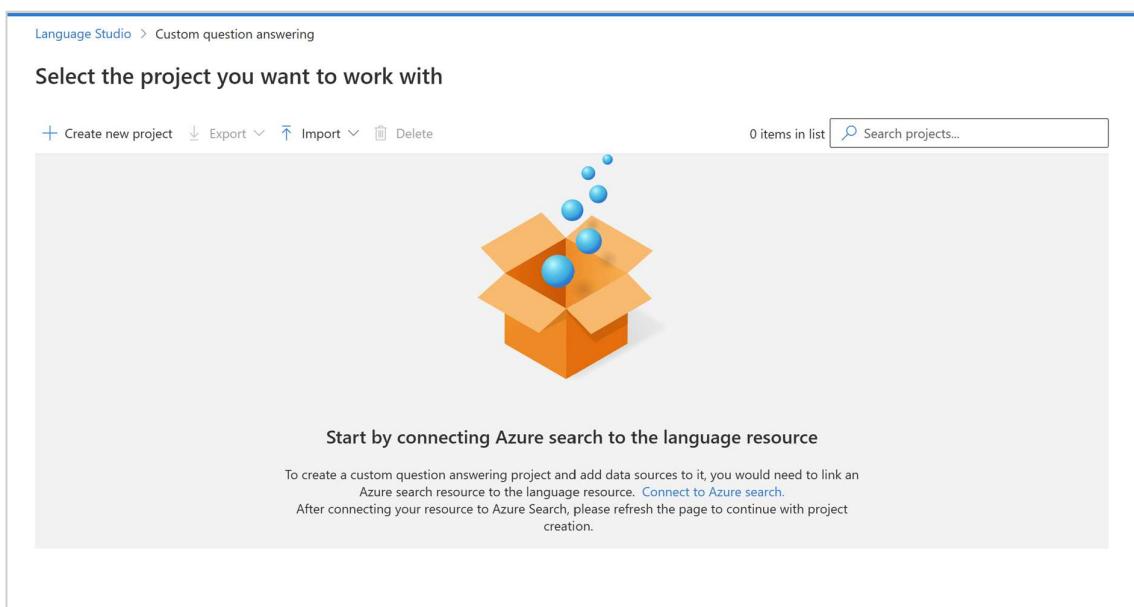


Fig 42: Initial Screen Custom question answering

- ❖ We need to connect to a resource that we have created in Fig 7

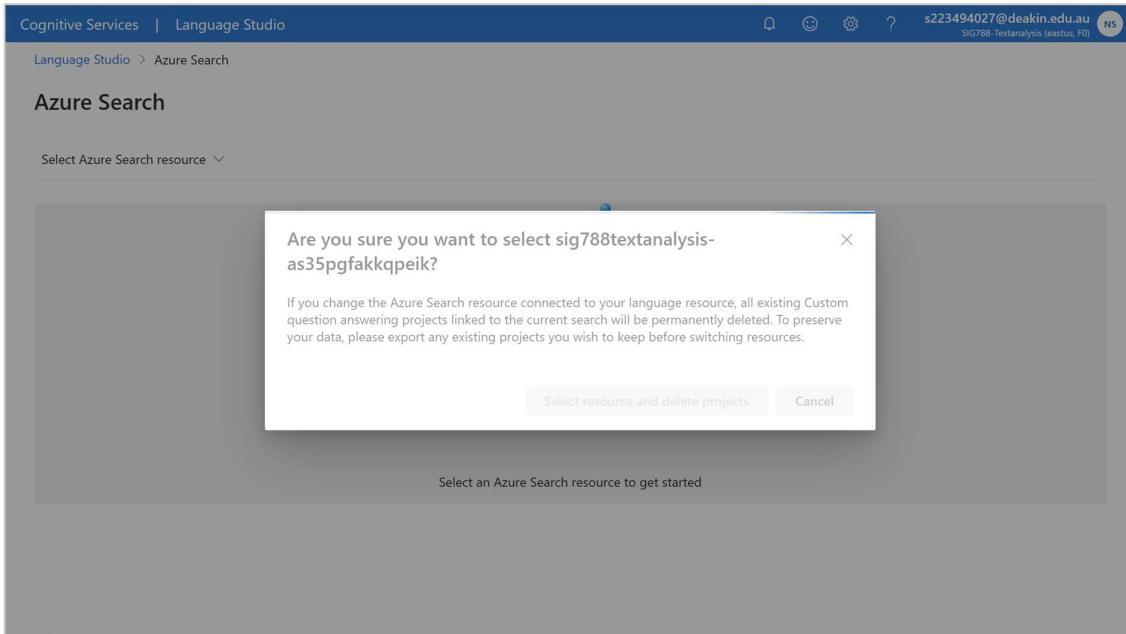


Fig 43: selecting the resources

- ❖ Earlier in Fig 5, we had selected the custom question and answering while creating the language studio. Here we have generated the same description while aligning to the resource group.

The screenshot shows the Azure Search resource details page. At the top, a green banner says 'Your current resource, sig788textanalysis-as35pgfakkqpeik, can be used to create a custom question answering project'. Below this, there are two sections: 'Language resource' and 'Azure Search'. Under 'Language resource', it shows 'Pricing tier: Free (F0)', 'Managed identity: Enabled', and 'Location: eastus'. Under 'Azure Search', it shows 'Azure Search resource: sig788textanalysis-as35pgfakkqpeik', 'Subscription: Free Trial', and 'Location: eastus'.

Fig 44: Generating the details of resources

- ❖ Creating a new project under the select the project. Here we have to define the language that we are going to use in this project. There are many languages available under the region. Here we have opted “English” as the language for all projects.

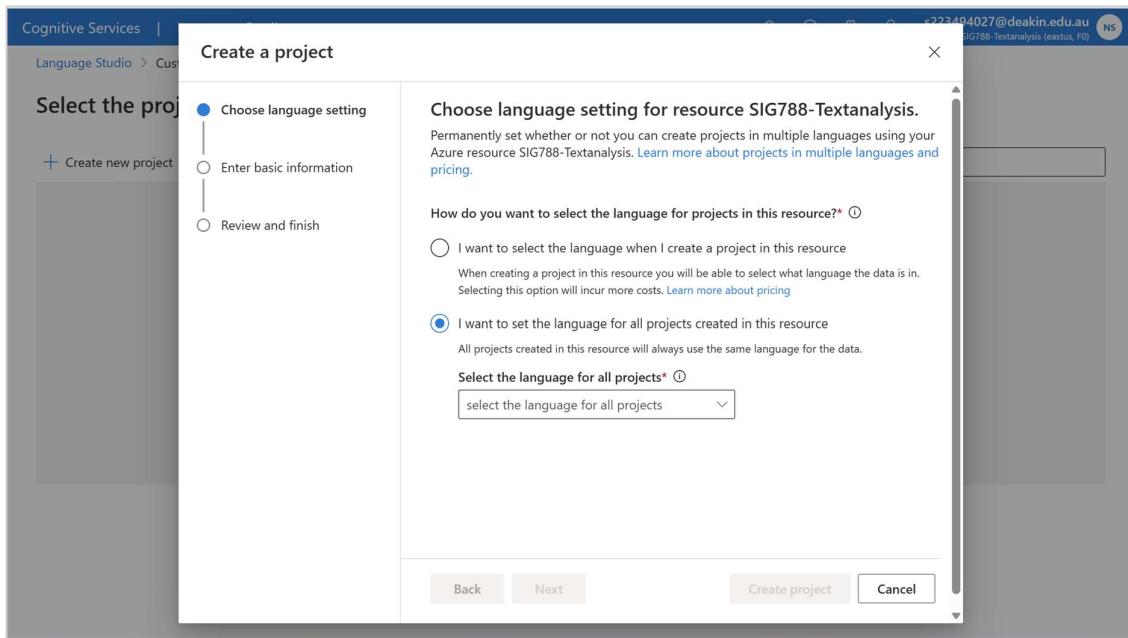


Fig 45: Create a project - language setting

- ❖ Under basic information, we have to update the name of the project, description of the project, language of the project. Here we can update a default answer if no resolution is found.

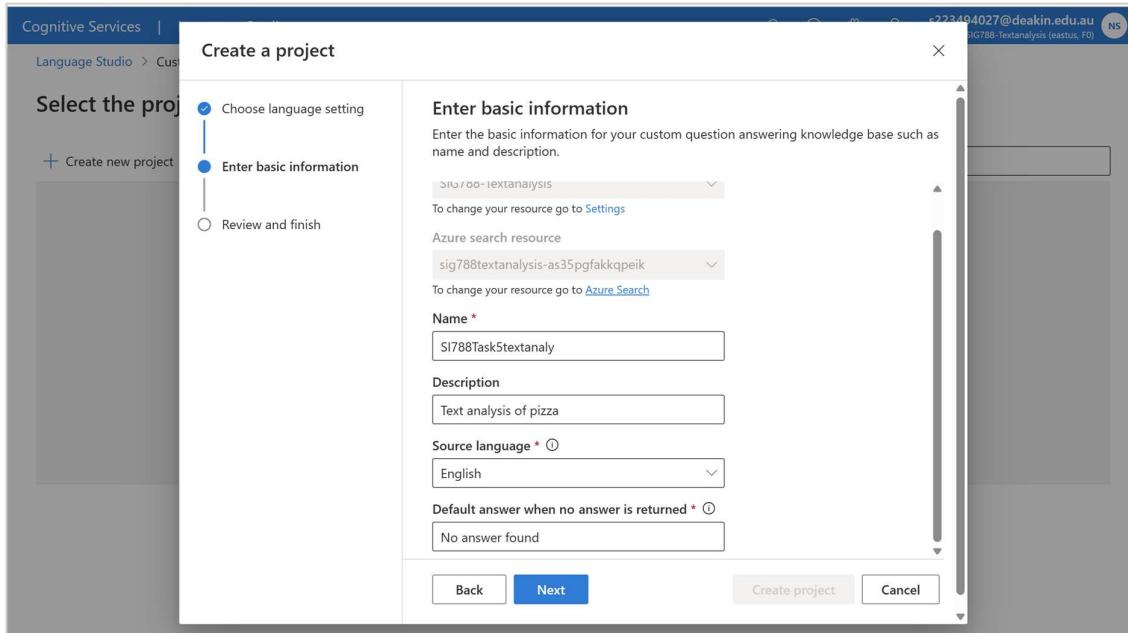


Fig 46: Create a project - Basic information

- ❖ Review the forms and then click “create project”

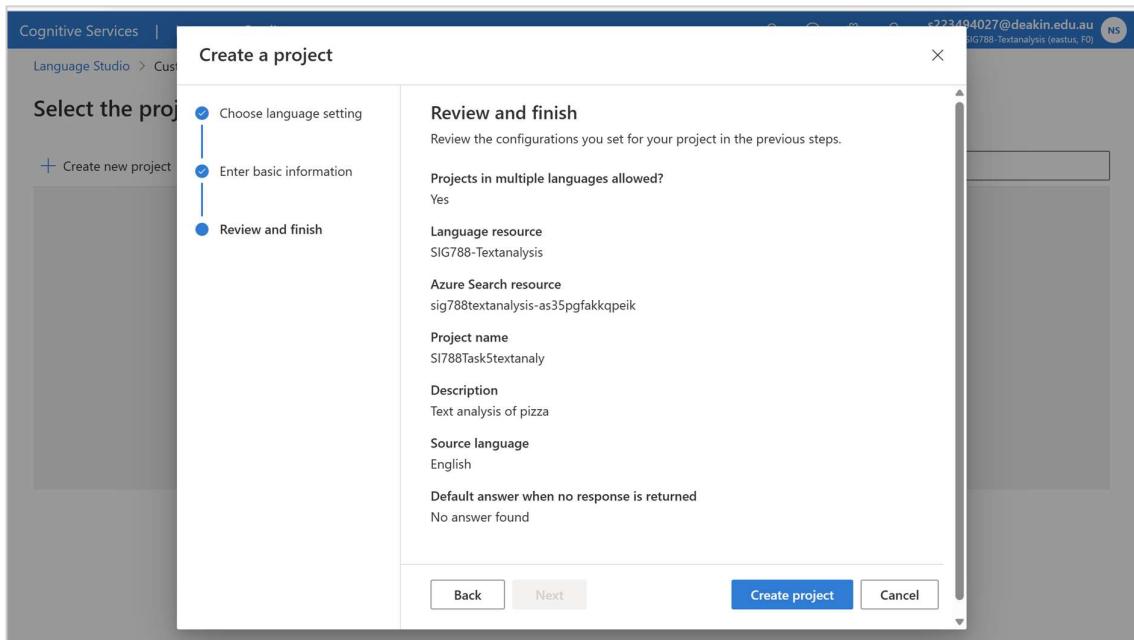


Fig 47:Create a project - Review and Finish

- ❖ Once the project is created, we can upload the question and answer as URL, Chitchats or files. Here we have taken the URL option. We have taken up frequently asked questions related pizzas from three pizza dealers. We used Pizza hut, Papa john, California Pizza kitchen and Dominos F&A.

Fig 48: Manage sources

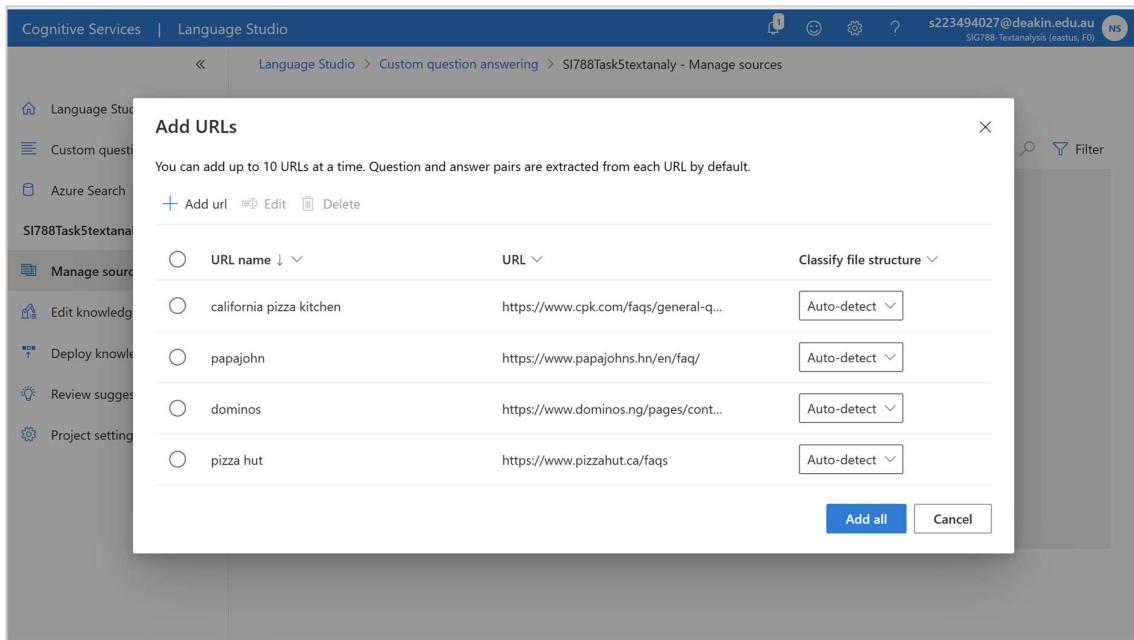


Fig 49: Add URLs

- ❖ URL name is given as vendor name and their website address are updated as URL. We have given the option to classify the Q&A as structured or unstructured as auto detect from the format.

Fig 50: Manage source

Fig 51: Manage Sources

- ❖ Select the option “Edit knowledge base” from the left pane. Here we get all the question and answer that we have uploaded from various URLs.

Fig 52: Edit Knowledge base

- ❖ Select all question and answer then select the option “Test” from the pane.

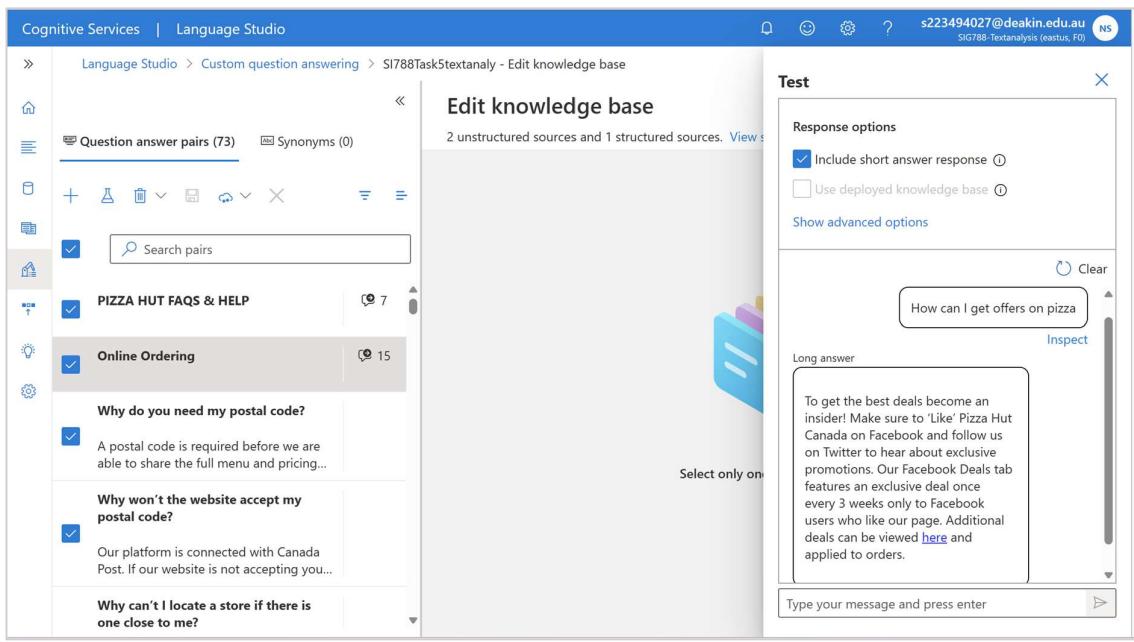


Fig 53: Test the Edit knowledge base

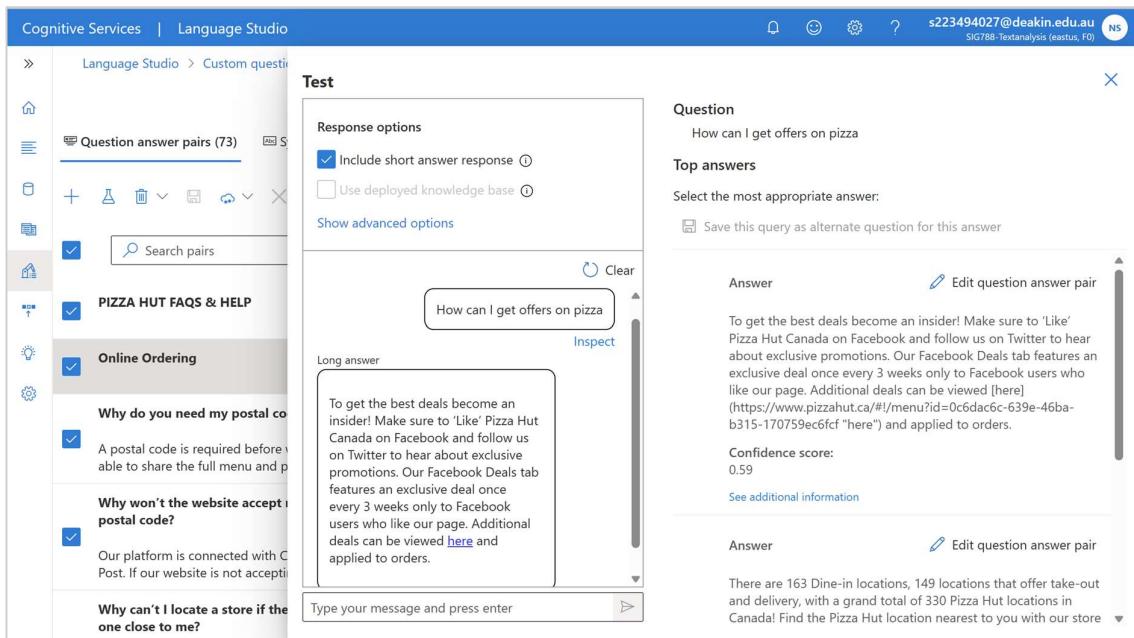


Fig 54: Test - 2

- ❖ Once the test is successful, we have to deploy the model to create a Chat bot.

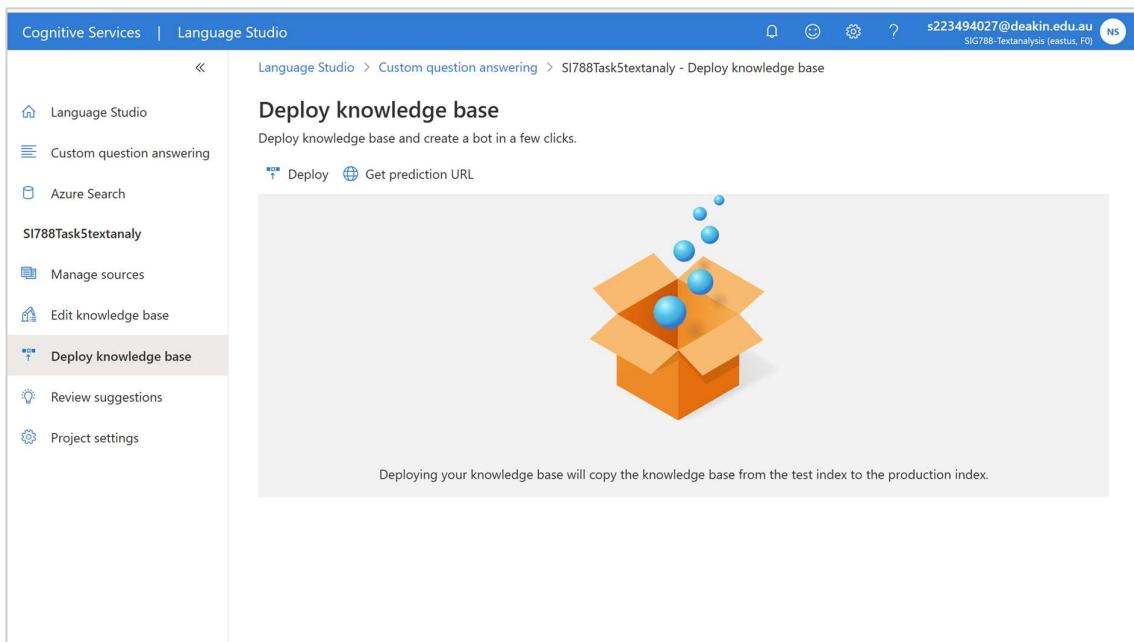


Fig 55: Deploy knowledge base

❖ Select the option “Deploy” from the above Figure.

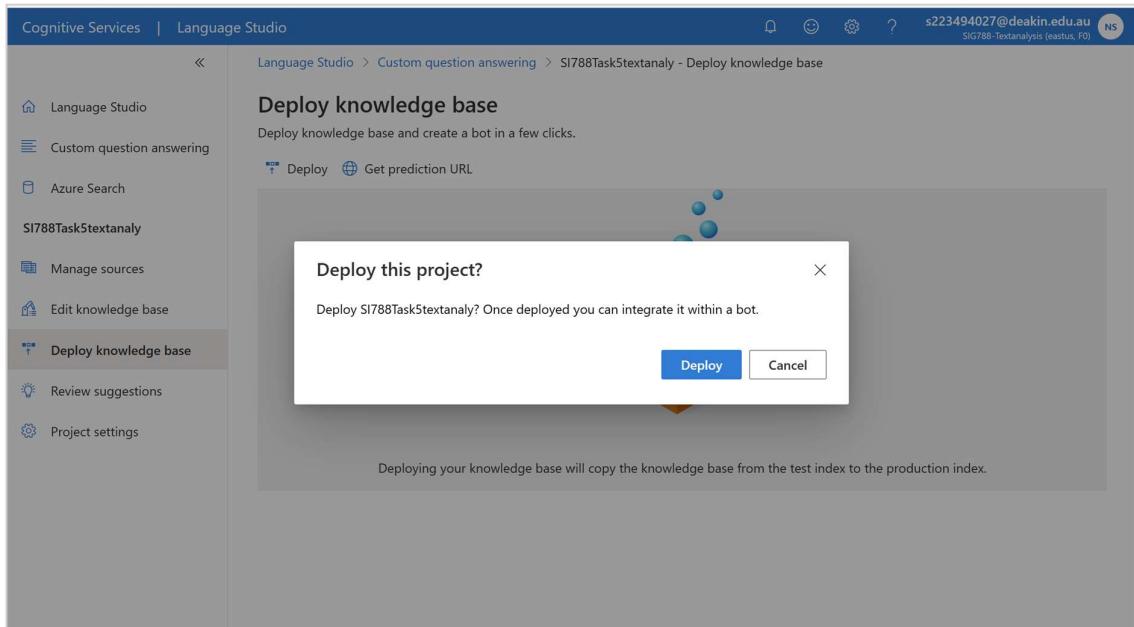


Fig 56: Select the project to deploy

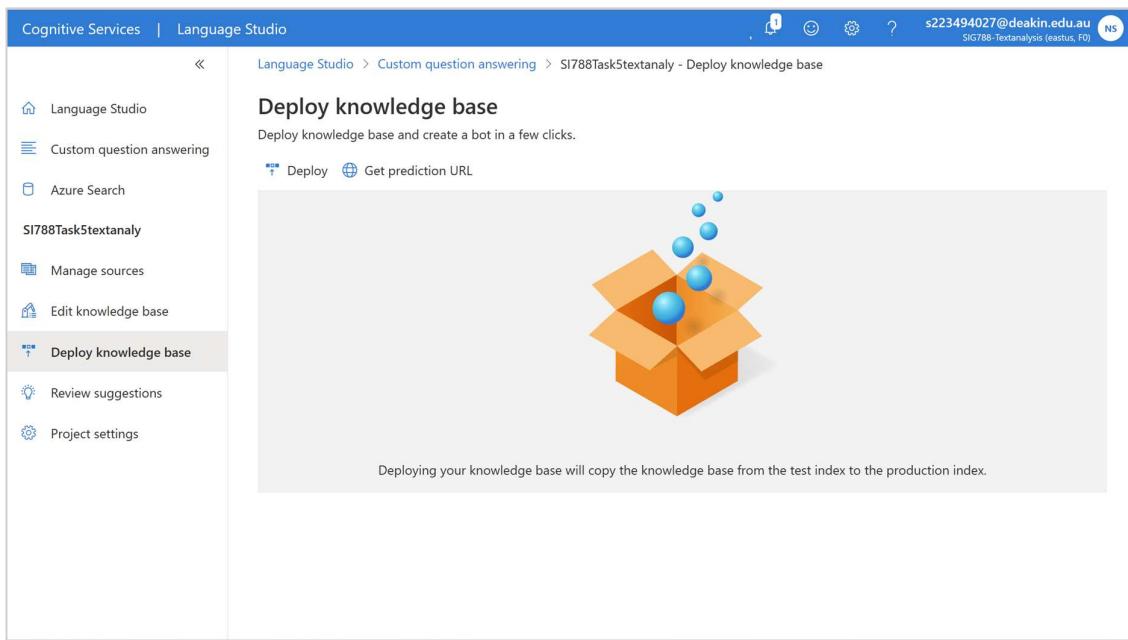


Fig 57: Deploying the model

- ❖ Once the model is successfully deployed, we get the option “Create a Bot”. While clicking on the option “create a bot” the page navigates to Azure portal from Language studio.

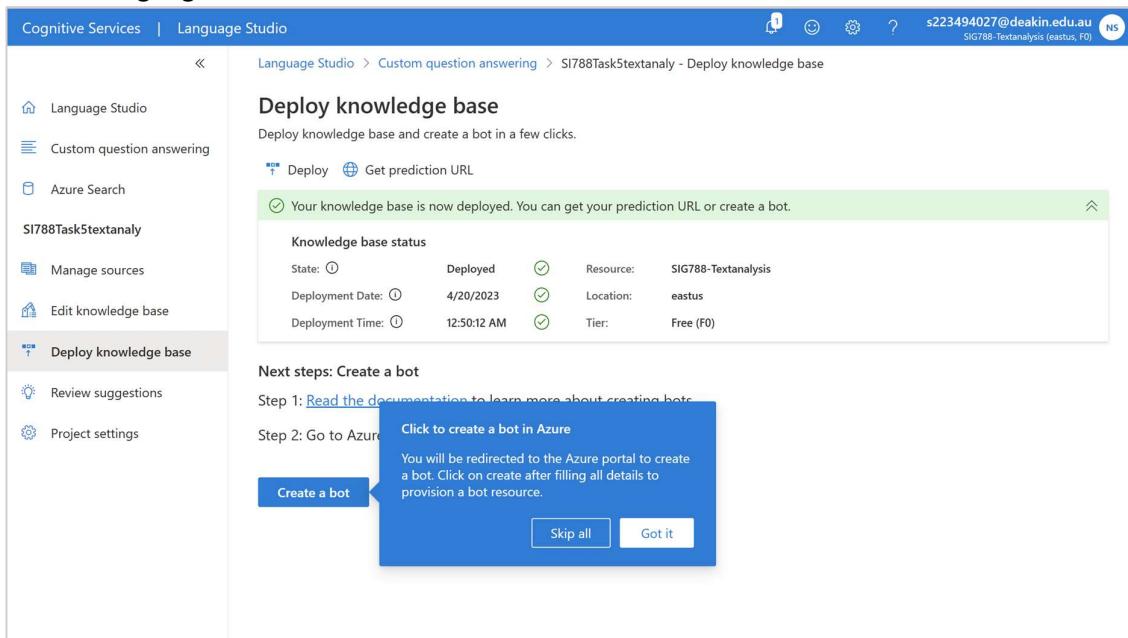


Fig 58: Model successfully deployed

- ❖ The page redirects to custom deployment where we have to mention the subscription details, price tier and bot details.

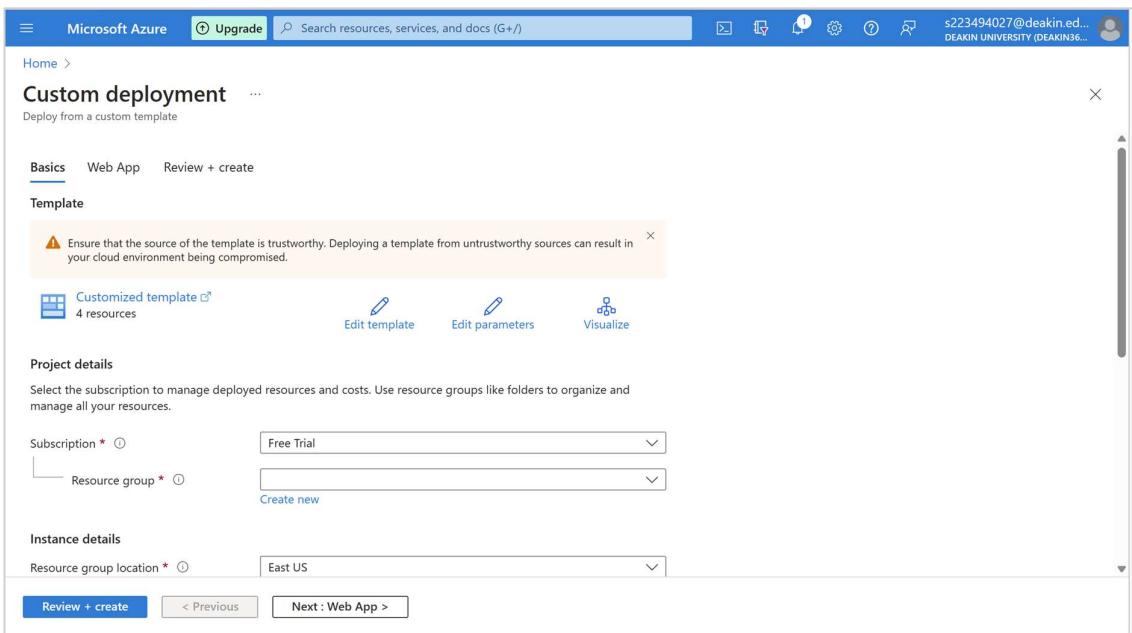


Fig 59: Custom deployment in Azure

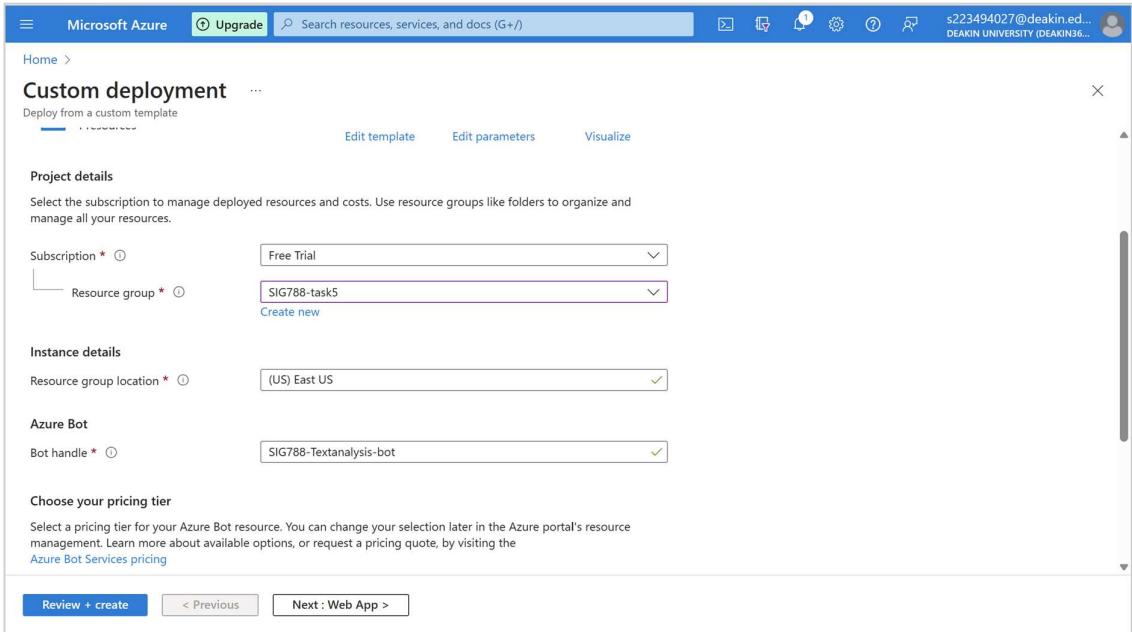


Fig 60: Updated custom deployment form -1

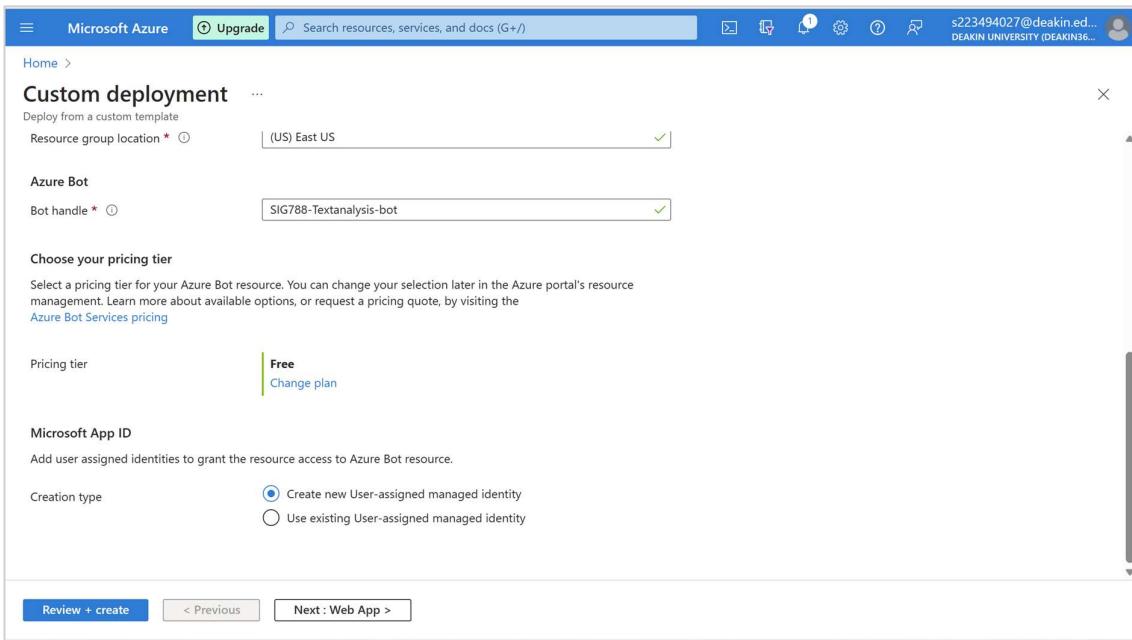


Fig 61: Updated custom deployment form - 2

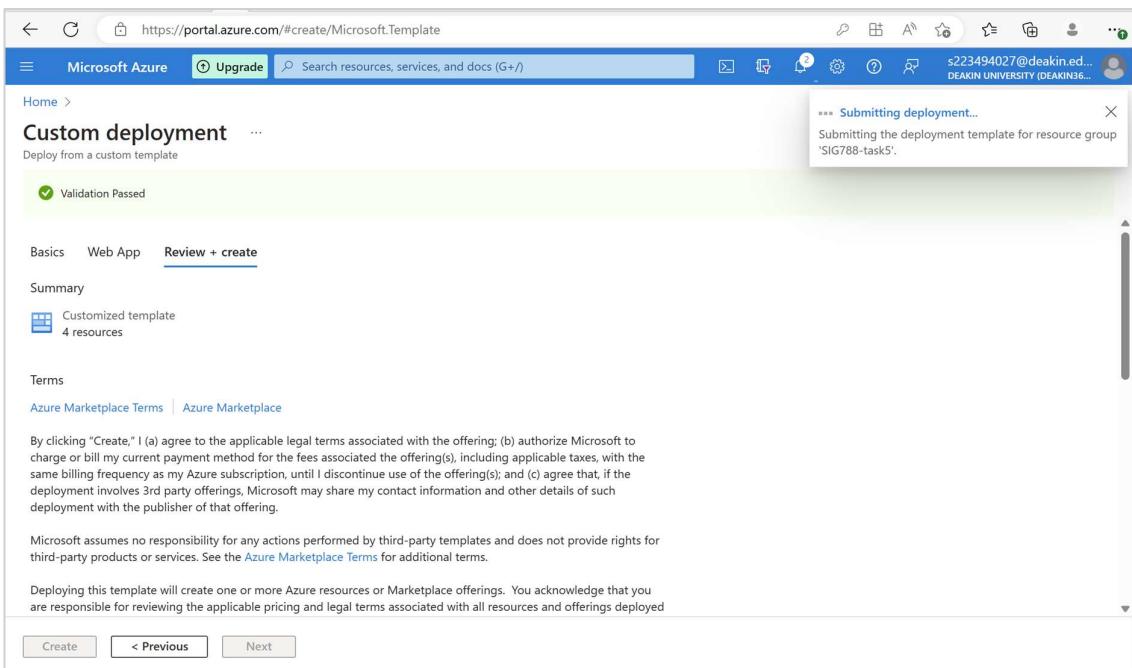


Fig 62: Custom deployment - Review + Create

- ❖ Once selected the option create, the deployment will be in progress.

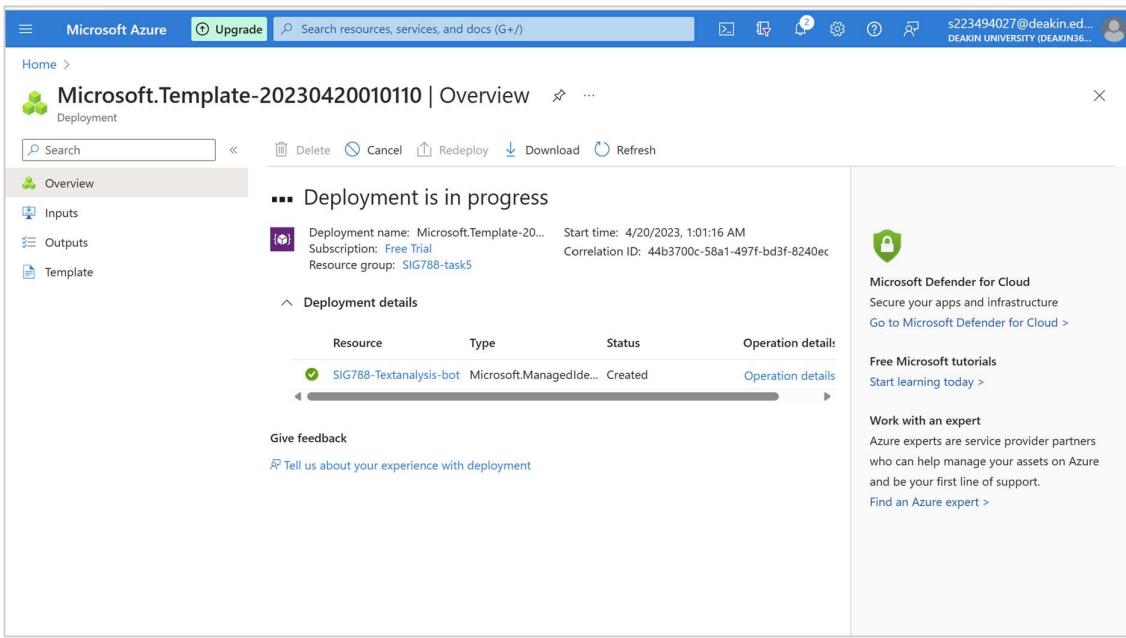


Fig 63: Deployment in progress

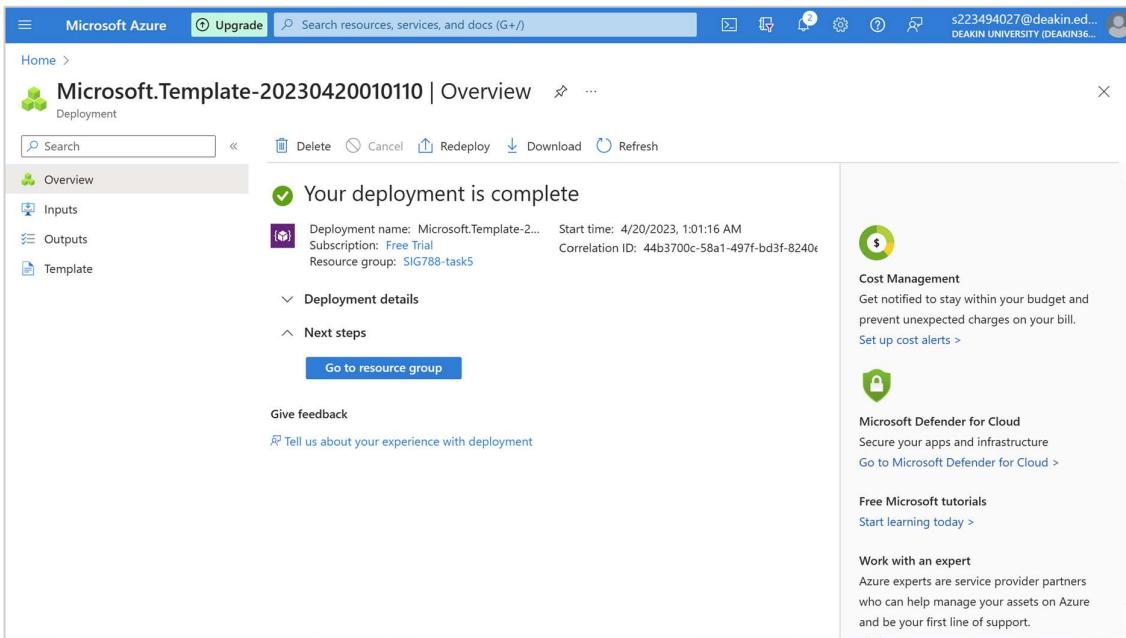


Fig 64: Deployment is completed

- ❖ After the successful deployment of bot, we can generate the overview of the bot as well as different setting. Even we can attach the bot to different channels.

The screenshot shows the Microsoft Azure portal interface for a bot service. The top navigation bar includes 'Microsoft Azure', 'Upgrade', 'Search resources, services, and docs (G+)', and user information 's223494027@deakin.edu... DEAKIN UNIVERSITY (DEAKIN36...)'. The main title is 'SIG788-Textanalysis-bot' under the 'Azure Bot' category. The left sidebar lists 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Settings' (with sub-options like 'Bot profile', 'Configuration', 'Channels', 'Pricing', 'Test in Web Chat', 'Encryption', 'Networking', 'Properties', 'Locks', and 'Monitoring'), and a survey prompt 'Help us improve Bot Service. Take our survey!'. The right panel displays 'Essentials' details: Resource group (SIG788-task5), Subscription (Free Trial), Subscription ID (c094e16e-df8d-4a15-bbc5-0dd2487e788c), Tags (WebAppLinked : true), Bot Service pricing tier (F0), Messaging endpoint (https://SIG788-Textanalysis-bot-4cb2.azurewebsites.net/api/mess...), and Location (Global). A 'JSON View' button is also present. Below this, a section titled 'Build enterprise-grade conversational AI' provides instructions for developing AI experiences.

Fig 65: Overview of Bot created

- ❖ Under the settings, we have the option to test the chat bot under web chats. Here we can test by asking several questions to the chat bot. We will be getting answers if the bot is aware of else, we will be getting the default answer.

The screenshot shows the 'Test in Web Chat' interface for the SIG788-Textanalysis-bot. The top navigation bar and sidebar are identical to Fig 65. The main area is titled 'SIG788-Textanalysis-bot | Test in Web Chat'. It features a 'Test' button and a 'Start over' button. At the bottom, there is a text input field with a microphone icon labeled 'Type your message' and a send arrow icon.

Fig 66: Test in Web chat

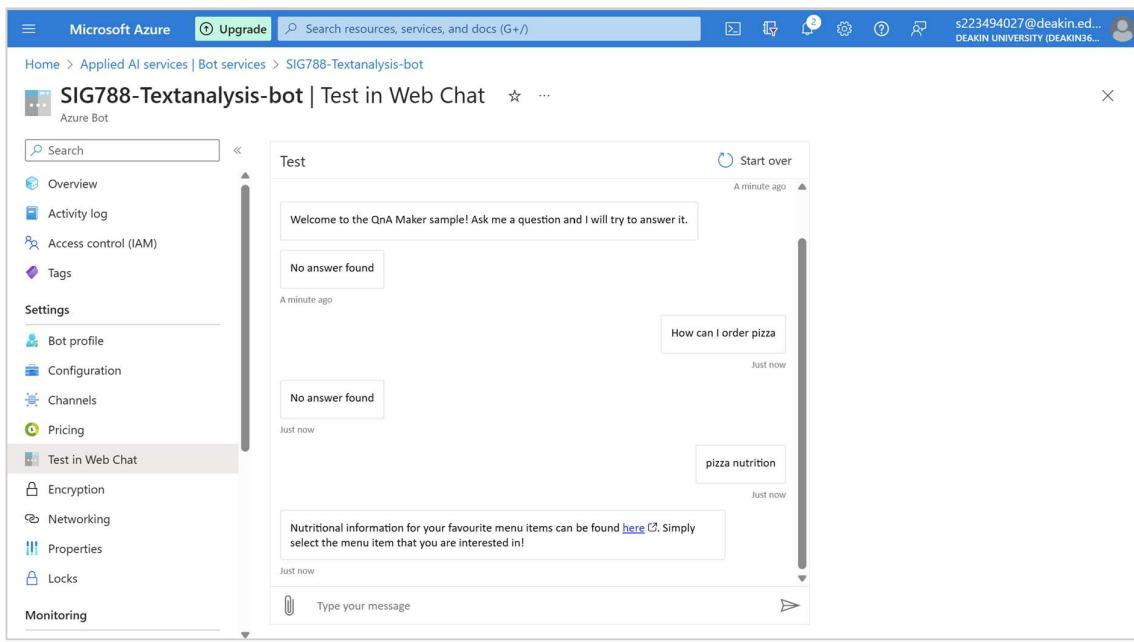


Fig 67: Test the bot -1

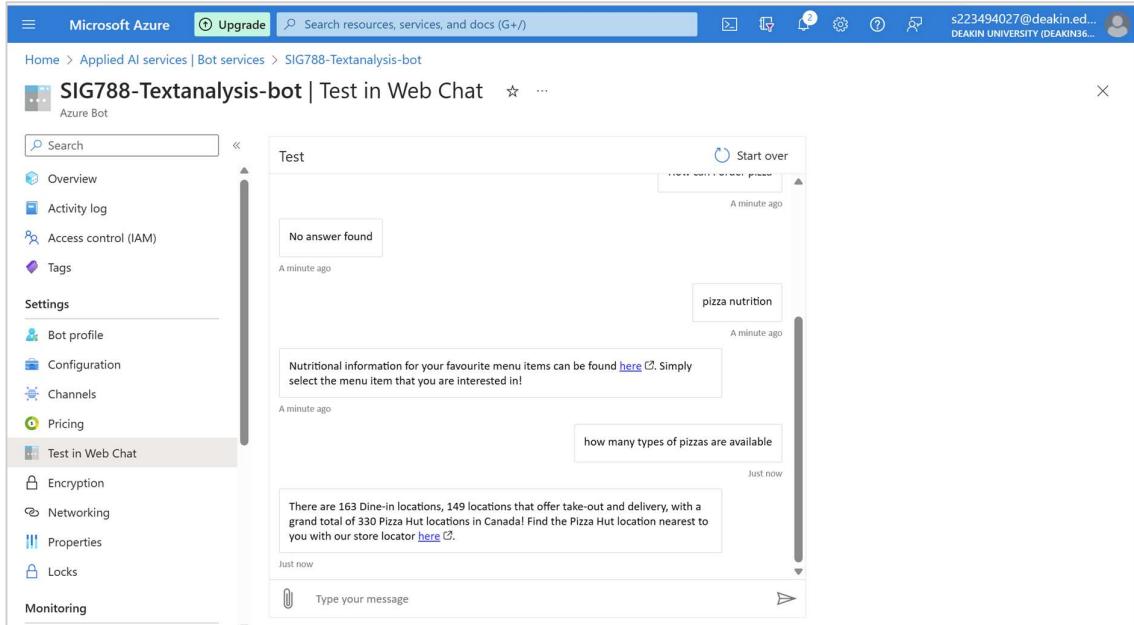


Fig 68: Test the bot -2

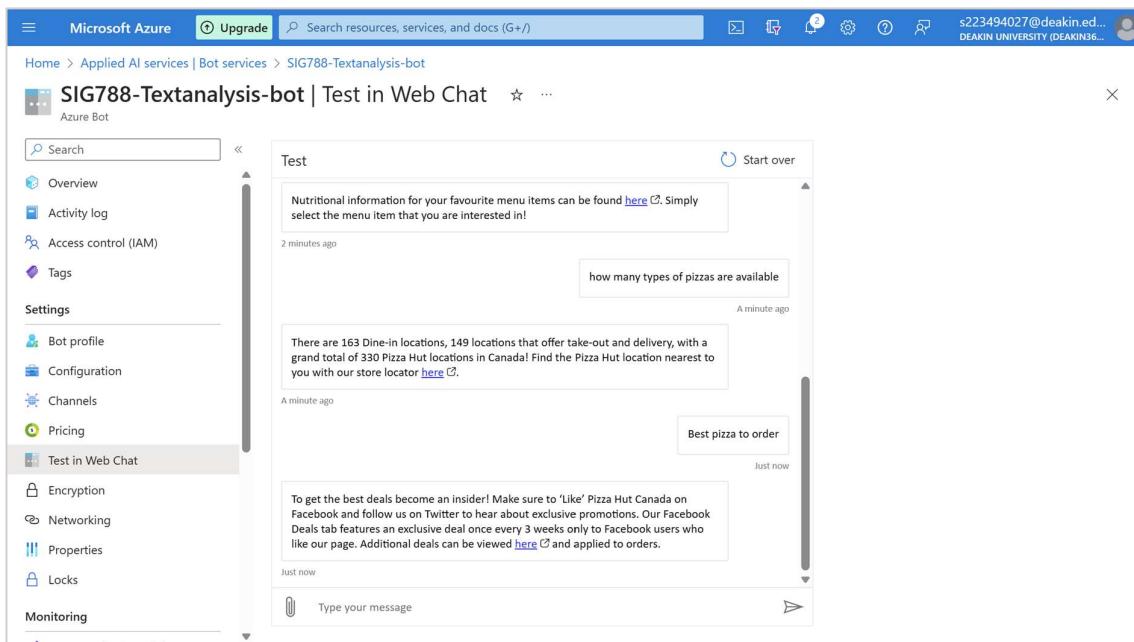


Fig 69: Test the bot -3

Attaching Bot to a Channel

- ❖ To attach the bot to the channel, we have to use the option “Channels” from settings. Here there are many channels like Slack, Teams, LinkedIn, Facebook, etc.

Fig 70: Attaching to a channels

- ❖ Here, we have used the channel as Teams. To attach the chat bot as a channel in teams we need to agree the consent form given in Terms of service.

Fig 71: Teams consent form

The screenshot shows the Microsoft Azure portal interface for a bot service named 'SIG788-Textanalysis-bot'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Settings (with Bot profile, Configuration, Channels selected), Pricing, Test in Web Chat, Encryption, Networking, Properties, Locks, and Monitoring. The main content area displays the 'Messaging' tab under the 'Bot services' section. It includes a message stating 'Messaging is available by default for your bot.' with a 'Learn more' link. Below this, there are two radio buttons: 'Microsoft Teams Commercial (most common)' and 'Microsoft Teams Government', each with a 'Learn more' link. A note at the bottom says 'To change the Teams Messaging selection, you will need to delete the channel and recreate it with a different messaging endpoint.' On the right side, a 'Notifications' panel is open, showing two recent events: 'Microsoft Teams channel enabled' (a few seconds ago) and 'Deployment succeeded' (18 minutes ago). There are buttons to 'Pin to dashboard' and 'Go to resource group'.

Fig 72: Channel added to Teams

- ❖ We need to publish our project in order to get the project attached into Teams.

The screenshot shows the Microsoft Azure portal interface for the same bot service. The top navigation bar has tabs for Upgrade, Search resources, services, and docs, and a user profile. The left sidebar is identical to Fig 72. The main content area now shows the 'Publish' tab selected in the top navigation bar. It includes sections for 'Test your bot in Teams' (with a link to 'Create a Teams app package') and 'Publish your bot for Teams' (with links to 'Publish your bot as a Line of Business app' and 'Publish your bot as an app in the Teams App store').

Fig 73: Publish the channel

- ❖ Here, we have got the option that our chat bot is successfully added in teams and we can open teams from the hyperlink given under actions to validate if we can view the channels that we have created.

The screenshot shows the Microsoft Azure portal interface for managing a bot. The left sidebar has a 'Channels' section selected under 'Bot profile'. The main content area displays the 'SIG788-Textanalysis-bot | Channels' page. It shows the bot is connected to three channels: Direct Line (Healthy, REST API), Microsoft Teams (Healthy, Microsoft Teams Channel), and Web Chat (Issues (2), Embeddable Web Chat control). There is also an 'Available Channels' section for Alexa.

Fig 74: Chat Bot linked with Microsoft Teams

- ❖ We will login to Microsoft teams and check if we have the chat bot visible under channels.

The screenshot shows the Microsoft Teams application window. The left sidebar has a 'Chat' section selected. The main area shows the 'SIG788-Textanalysis-bot Chat' conversation. It lists 'Pinned' messages from 'NEETHU SIDHARDHAN (You)' and a 'Recent' message from 'SIG788-Textanalysis-bot Draft'. Below the messages is a draft message with three recipients and a smiley face icon. A message box at the bottom says 'You're starting a new conversation' and 'Type your first message below.' The bottom navigation bar includes icons for file attachments, mentions, and other message options.

Fig 75: Microsoft Teams

- ❖ We can see the chat bot created by us under chat.

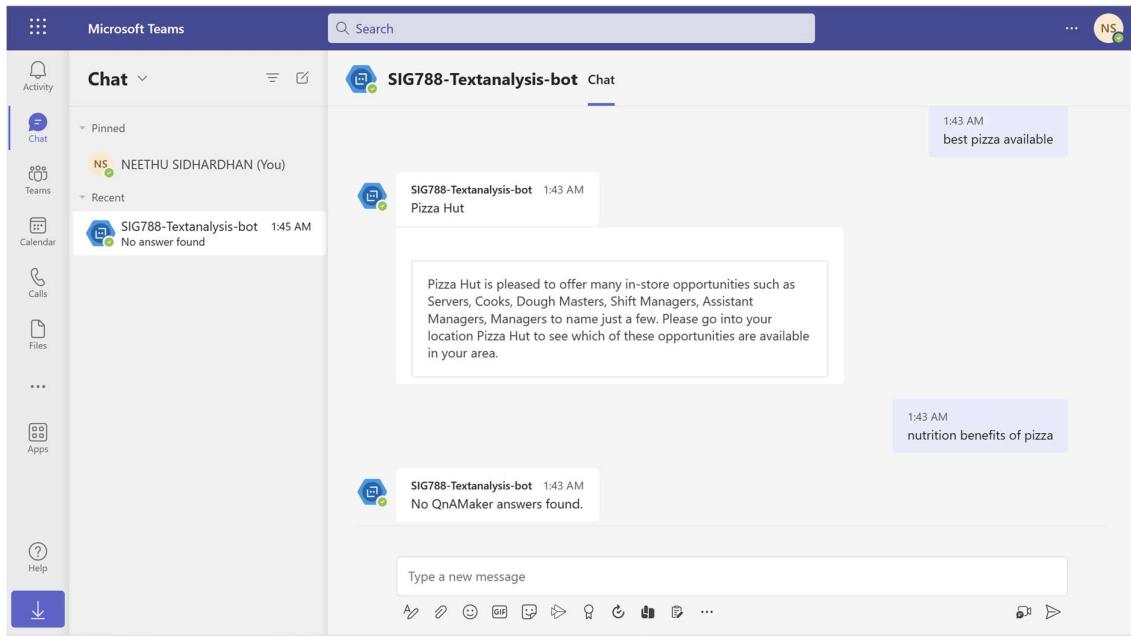


Fig 76: Test with the chat bot created

- ❖ Under Bot profile, we can upload the icon of the Bot profile. The same icon will be appearing in Microsoft teams.

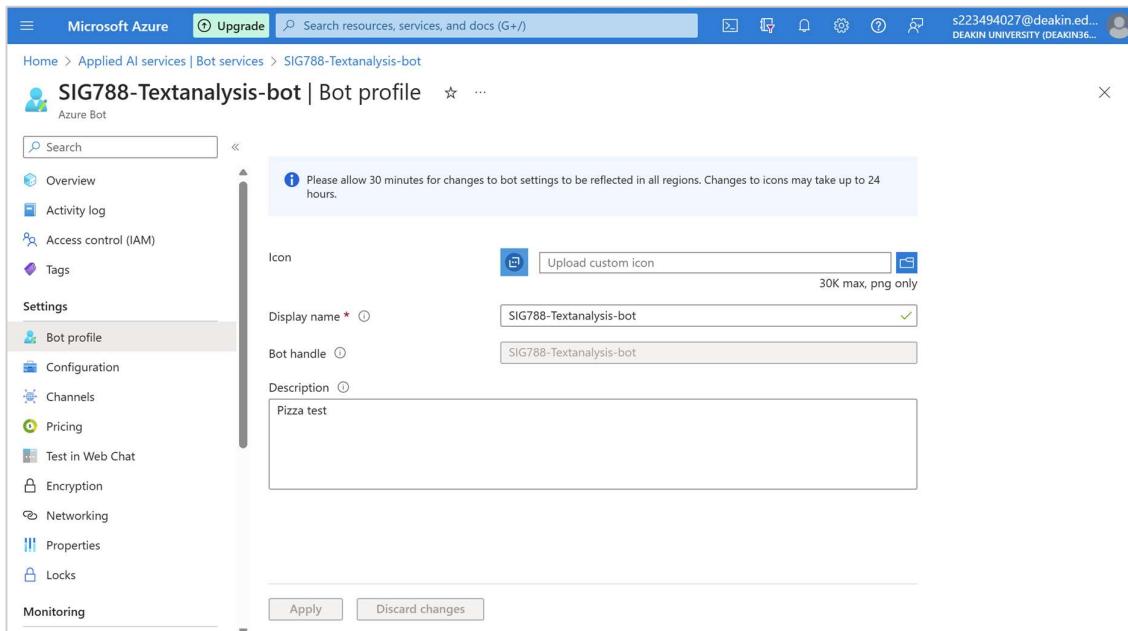


Fig 77: Bot Profile

Works Cited

1. Anon., n.d. *California Pizza Kitchen*. [Online] Available at: <https://www.cpk.com/faqs/general-questions> [Accessed 19 April 2023].

2. Anon., n.d. *Domino's*. [Online]
Available at: <https://www.dominos.ng/pages/content/customer-service/faq#:~:text=Frequently%20Asked%20Questions%201%20How%20to%20order%20Domino%E2%80%99s,purchase%2C%20what%20can%20I%20do%3F%20...%20More%20items>
[Accessed 19 April 2023].
3. Anon., n.d. *papa john*. [Online]
Available at: <https://www.papajohns.hn/en/faq/>
[Accessed 19 April 2023].
4. Anon., n.d. *Pizza Hut*. [Online]
Available at: <https://www.pizzahut.ca/faqs>
[Accessed 19 April 2023].
5. Berry, D., 2018. *GitHub*. [Online]
Available at: <https://github.com/Azure-Samples/cognitive-services-language-understanding/blob/master/documentation-samples/quickstarts/change-model/TravelAgent.json>
[Accessed 16 April 2023].
6. Hazemelh, 2022. <https://github.com/Azure-Samples/cognitive-service-language-samples/tree/main/CLU-Pizza-Samples>. [Online]
Available at: <https://github.com/Azure-Samples/cognitive-service-language-samples/tree/main/CLU-Pizza-Samples>
[Accessed 20 Apr 2023].
7. Sandford, M. M., 2023. *Great Learning*. [Online]
Available at:
https://olympus.mygreatlearning.com/courses/91262/files/8084264?module_item_id=4062513
[Accessed 19 April 2023].