Developing and Deploying Intelligent Chat Bots

Microsoft

LUIS Application

Objectives

The aim of this lab is to create a LUIS app through a real-life example and understand the various components of LUIS. In this lab, we will create a LUIS app for grocery support. The idea here is to define intents and entities related to buying grocery products.

Create App

1. Go to https://www.luis.ai and sign in if you have an account. If you do not have an account, create an account. LUIS homepage at https://www.luis.ai looks like the below image:
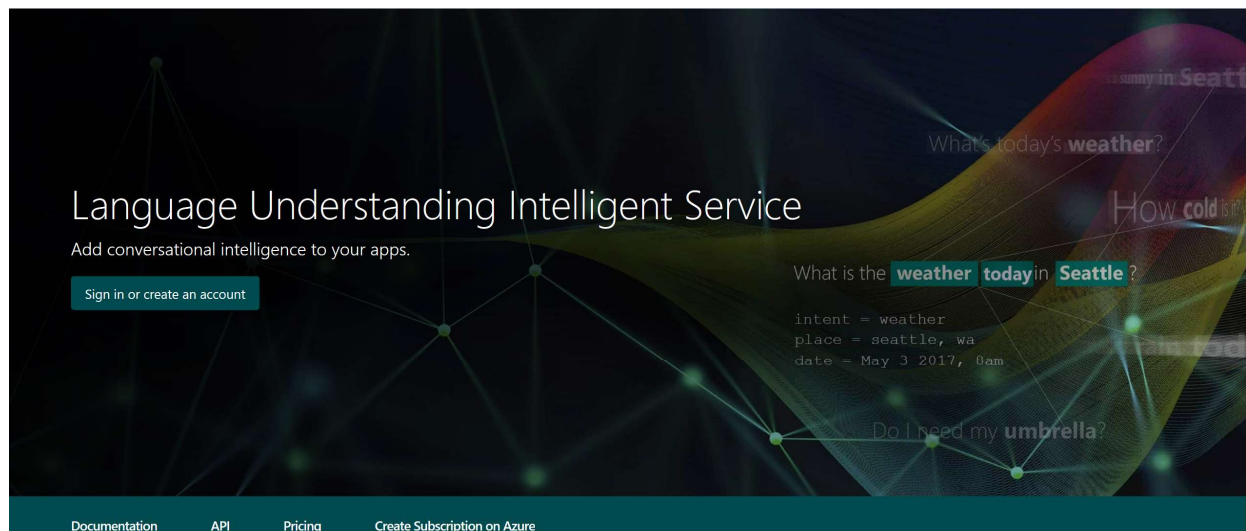
2. On successful login, you will be taken to *My Apps*. The existing apps created are displayed here. To create a new app, select *New App*.

3. Fill out the fields in *Create a new app* window. Ensure the key used is BootstrapKey. BootstrapKey is used for testing purposes.

4. After creating the app, you will be able to see your newly created app in *My apps*. In this example, notice GroceryApp listed in *My apps*.

My Apps

Create and manage your LUIS applications ...Learn more

New App   **Import App**

Name ↓	Culture	Created date	Endpoint hits	Controls
GroceryApp	en-us	Jun 30, 2017 4:38:20 PM	0	✎ ☁ {} 🗑
HydLab3	en-us	Apr 26, 2017 8:01:19 PM	0	✎ ☁ {} 🗑
VMLApp	en-us	Jun 1, 2017 8:06:41 AM	0	✎ ☁ {} 🗑
testApp	en-us	Jan 16, 2017 11:05:45 PM	0	✎ ☁ {} 🗑

Intents

1. The core part of any LUIS app is intents. Let us begin by adding prebuilt intents by selecting *Add Prebuilt intents*. We can add Places.GetPhoneNumber and Places.GetPriceRange for our prebuilt intents.

   Places.GetPhoneNumber can be useful for a grocery support system where a user may call and ask for a local branch address and phone number. Similarly, Places.GetPriceRange would be useful when users query for grocery product prices.
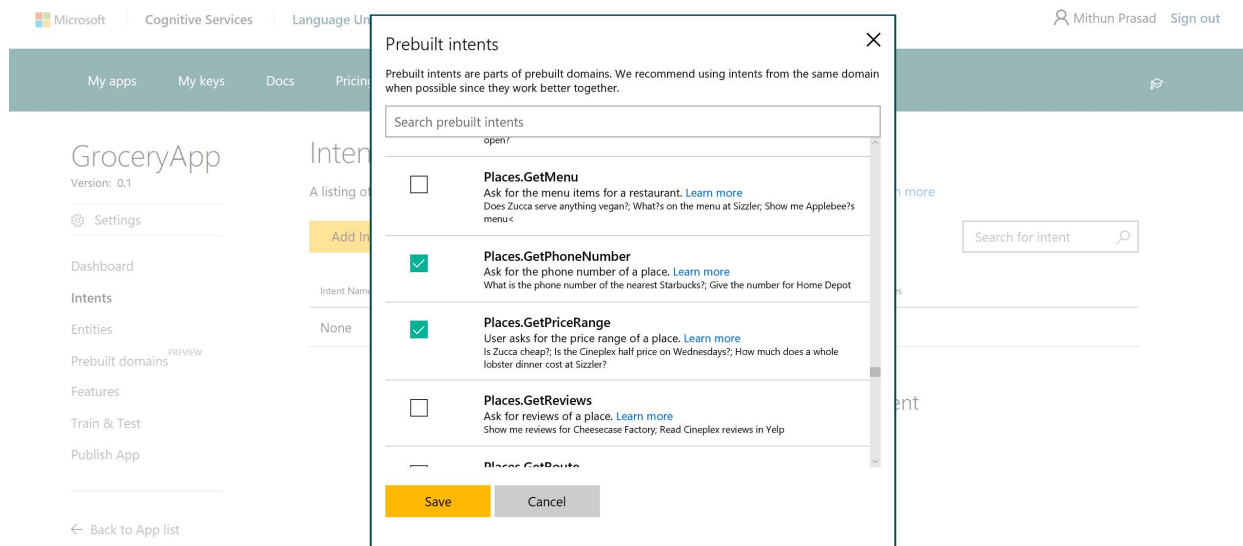
2. Next, lets add our own intent. In this case, we can call it *BuyGroceries*.

3. For the intent *BuyGroceries*, add utterances related to buying groceries.

4. For the utterance text, you can select token(s) and assign entities by right-clicking and creating an entity. For example, in the utterance text "I want to buy milk", milk can be assigned to an entity GroceryProducts.

Features

Features are a great way to add similar tokens. Features help LUIS recognize both intents and entities, by providing hints to LUIS that certain words and phrases are part of a category or follow a pattern. When the app has difficulty identifying an entity, adding a feature and retraining the LUIS app can often help improve the detection of related intents and entities.

1. Begin by adding phrase list for grocery store names. We can call this GroceryStores. It is worth noting that when you add a value for a grocery store such as "Whole Foods", LUIS makes suggestions to similar values. Not all of them are related, however. Drag the ones that you think are similar to the phrase list.


- Features are "exchangeable" in the sense that an utterance where one word/phrase appears would be labeled similarly if the word/phrase were *exchanged* with another. For example, for exchangeable phrases "January" and "February", "show the calendar for January" has the same intent as "show the calendar for February".

- Is active makes the phrase list active for the app.

Train & Test

Development of a robust natural language understanding system would take quite a bit of refinement. When you think you have a rich amount of entities, intents or utterances in your current model, train your app before testing and publishing. With training a model, LUIS generalizes from the examples you have labeled, and builds model to recognize the relevant intents and entities in the future, thus improving its classification accuracy.

There are two ways of testing your app – interactive and batch.

1. Begin with interactive testing by submitting a test utterance in the console. For a test utterance, you will see intents with different scores. LUIS also lists out the intent with the highest score.
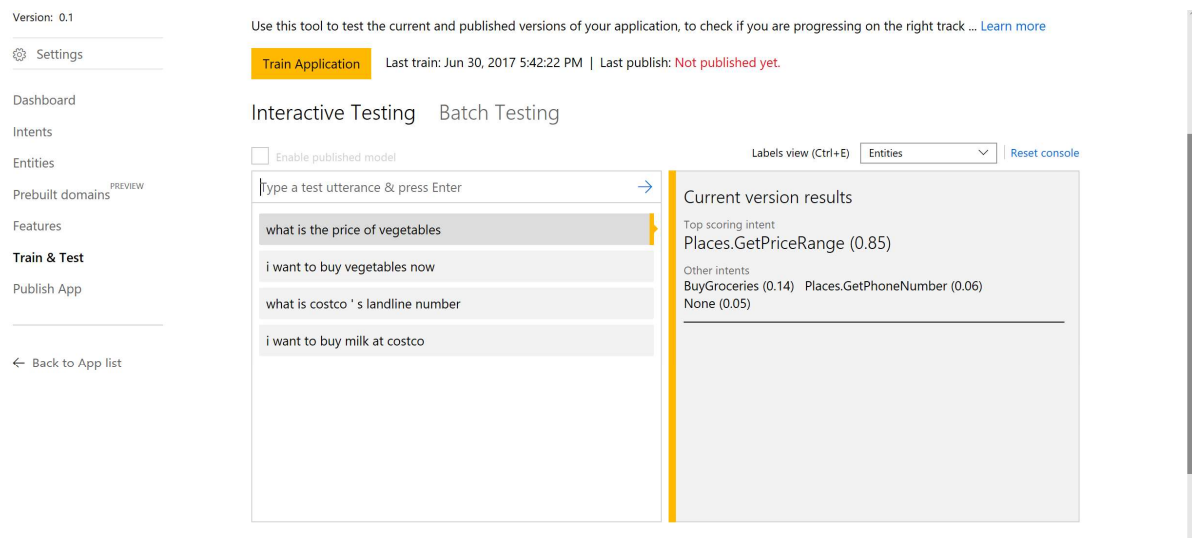
2. Batch testing allows for testing of a large number of utterances. Since a dataset is submitted for testing, LUIS also generates a confusion matrix from validation which can help users identify areas to improve.

The dataset has to have a .json extension and must follow the below json format:
```
[
    {
        "text": "I want to buy milk",
        "intent": "None",
        "entities":
        [
        {
            "entity": "GroceryProducts",
            "startPos": 14,
            "endPos": 17
        }
        ]
```

```
        },
        {
                "text": "I want to buy chocolates at costco",
                "intent": "None",
                "entities":
                [
                {
                        "entity": "GroceryProducts",
                        "startPos": 14,
                        "endPos": 23
                }
                ]
        }
    ]
```

GroceryApp
Version: 0.1

⚙ Settings

Dashboard

Intents

Entities

Prebuilt domains PREVIEW

Features

Train & Test

Publish App

Test

Use this to...   ...sing on the right track ... Learn more

Train Ap

Interac

Import c

Import dataset   ✕

Dataset name (REQUIRED)

GroceryDataset

Dataset JSON file (REQUIRED)

C:\Users\miprasad\Downloads\groceryDataset.json   Browse...

Learn about the supported dataset syntax ⌄

Save   Cancel

Status   t Test Success   Controls

▷ Test   GroceryDataTest   2   Not tested yet   Not tested yet   ↓ ✎ 🗑

3. On successful import, you will be able to view the results (confusion matrix) by selecting *See results*.

You will be able to select different confusion matrix views by selecting entities/intents

Publish App

On successful testing, you are ready to publish your app. Select *Publish App* from the left panel. The BootstrapKey we initially setup for the app was for testing purposes only. Click *Add a new key to your account* to obtain a new key and ensure the *Endpoint slot* is set to Production.

You will get an *Endpoint url* after publishing to use in your app as shown below:

Exercise

1. For the grocery app, can you add more entities and intents related to grocery shopping?
2. Can you include more features to capture similar milk products? For example, almond milk, soy milk, lactose free milk, etc.
3. Perform batch setting with a larger dataset (about 500 utterances) and investigate the results (confusion matrix) of the app