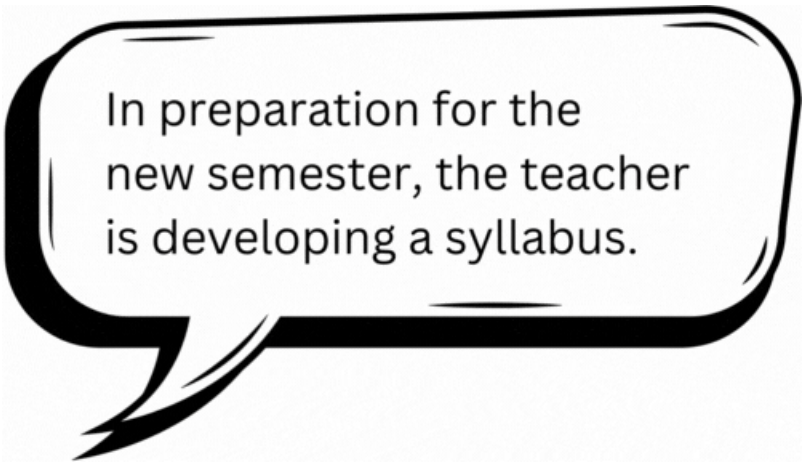
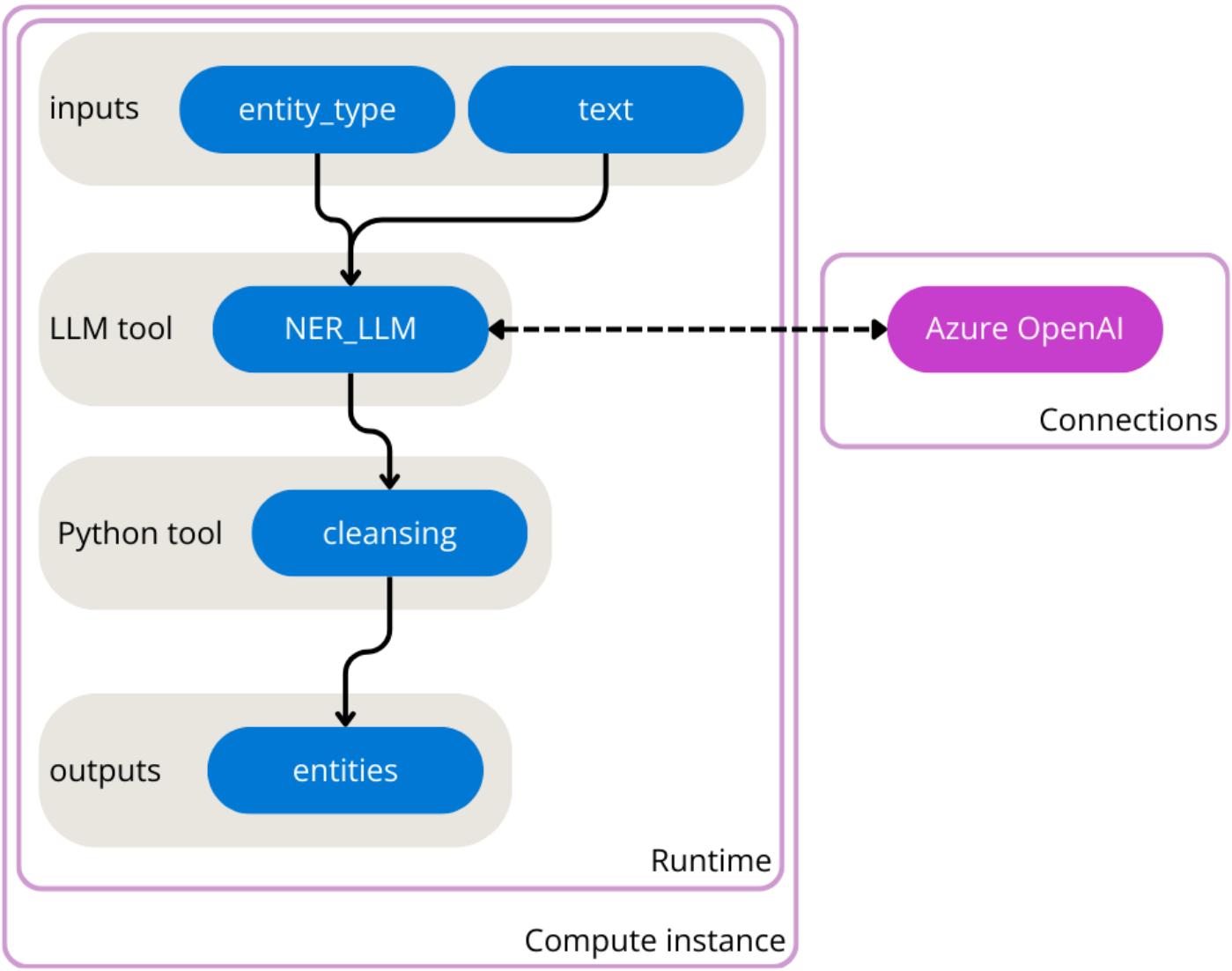
Exercise 6- AI Studio Build

Extracting valuable information from text is known as Named Entity Recognition (NER). Entities are key words that are of interest to you in a given text.



Large Language Models (LLMs) can be used to perform NER. To create an application that takes a text as input and outputs entities, you can create a flow that uses a LLM node with prompt flow.

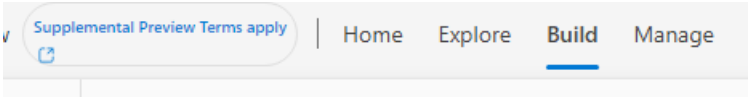
In this exercise, you'll use Azure AI Studio's prompt flow to create an LLM application that expects an entity type and text as input. It calls a GPT model from Azure OpenAI through a LLM node to extract the required entity from the given text, cleans the result and outputs the extracted entities.



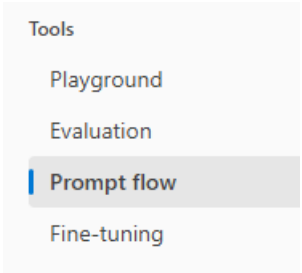
You first need to create a project in the Azure AI Studio to create the necessary Azure resources. Then, you can deploy a GPT model with the Azure OpenAI service. Once you have the necessary resources, you can create the flow. Finally you'll run the flow to test it and view the sample output.

Prompt flow

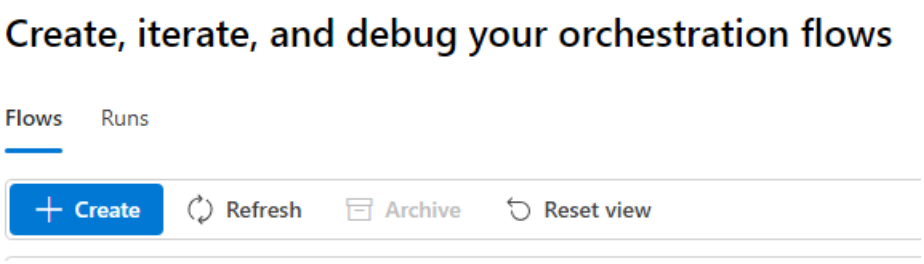
1. Click on the **Build** tab in the top navigation



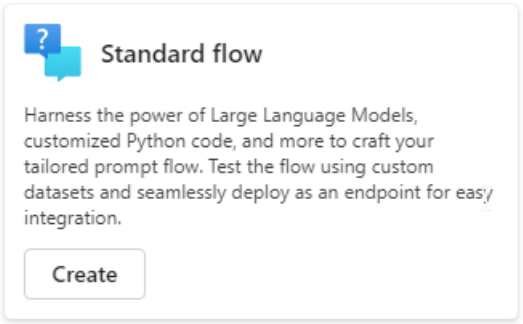
2. Then click on **Prompt flow** in the navigation



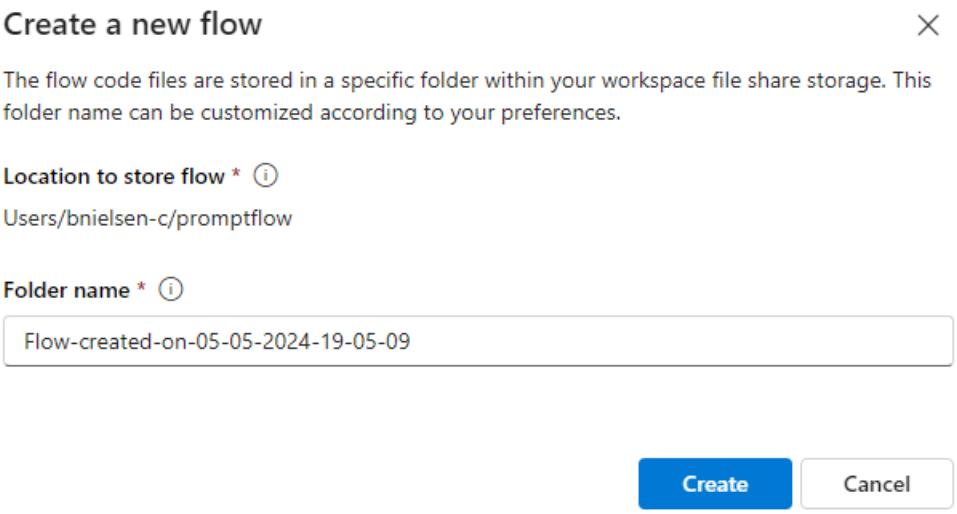
3. Then click on the **Create** button



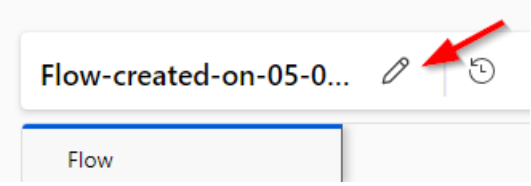
4. In the popup click on **Create** for the **Standard flow**



5. In the popup leave the defaults and click **Create**



6. Rename the flow, by clicking the pencil icon, to `entity-recognition`



Start the automatic runtime

To test your flow, you need compute. The necessary compute is made available to you through the runtime.

1. After creating the new flow that you named `entity-recognition`, the flow should open in the studio.
2. Select the **Select runtime** field from the top bar.
3. In the **Automatic runtime** list, select **Start** to start the automatic runtime.
4. Wait for the runtime to start.

Configure the inputs

The flow you'll create will take two inputs: a text and the type of entity you want to extract from the text.

1. Under **Inputs**, one input is configured named `topic` of type `string`. Change the existing input and update with the following settings:
    - **Name**: `entity_type`
    - **Type**: `string`
    - **Value**: `job title`
2. Select **Add input**.
3. Configure the second input to have the following settings:
    - **Name**: `text`
    - **Type**: `string`
    - **Value**: `The software engineer is working on a new update for the application.`

Configure the LLM node

The standard flow already includes a node that uses the LLM tool. You can find the node in your flow overview. The default prompt asks for a joke. You'll update the LLM node to extract entities based on the two inputs specified in the previous section.

1. Navigate to the **LLM node** named `joke`.
2. Replace the name with `NER_LLM`
3. For **Connection**, select the OpenAI connection previously created.
4. For **deployment_name**, select the `gpt-35-turbo` model you deployed.
5. Replace the prompt field with the following code:

```
# system:

 Your task is to find entities of a certain type from the given text content.
 If there're multiple entities, please return them all with comma separated, e.g. "entity1,
entity2, entity3".
 You should only return the entity list, nothing else.
 If there's no such entity, please return "None".

 # user:
 Entity type: {{entity_type}}
 Text content: {{text}}
```

6. Select **Validate and parse input**.
7. Within the LLM node, in the **Inputs** section, configure the following:
    - For `entity_type`, select the value `${inputs.entity_type}`.
    - For `text`, select the value `${inputs.text}`.

Your LLM node will now take the entity type and text as inputs, include it in the prompt you specified and send the request to your deployed model.

Configure the Python node

To extract only the key information from the result of the model, you can use the Python tool to clean up the output of the LLM node.

1. Navigate to the Python node named `echo`.
2. Replace the name with `cleansing`.
3. Replace the code with the following:

```
from typing import List
from promptflow import tool


@tool
def cleansing(entities_str: str) -> List[str]:
    # Split, remove leading and trailing spaces/tabs/dots
    parts = entities_str.split(",")
    cleaned_parts = [part.strip(" \t.\"") for part in parts]
    entities = [part for part in cleaned_parts if len(part) > 0]
    return entities
```

4. Select **Validate and parse input**.
5. Within the Python node, in the **Inputs** section, set the value of `entities_str` to `${NER_LLM.output}`.

Configure the output

Finally, you can configure the output of the whole flow. You only want one output to your flow, which should be

the extracted entities.

1. Navigate to the flow's **Outputs**.
2. For **Name**, enter `entities`.
3. For **Value**, select `${cleansing.output}`.
4. Click **Save**

Run the flow

Now that you've developed the flow, you can run it to test it. Since you've added default values to the inputs, you can easily test the flow in the studio.

1. Select **Run** to test the flow.
2. Wait until the run is completed.
3. Select **View outputs** in the **NEW_LLM**. A pop-up should appear showing you the output for the default inputs. Optionally, you can also inspect the logs.