Azure OpenAI

This notebook goes over how to use Langchain with Azure OpenAI.

The Azure OpenAI API is compatible with OpenAI's API. The `openai` Python package makes it easy to use both OpenAI and Azure OpenAI. You can call Azure OpenAI the same way you call OpenAI with the exceptions noted below.

API configuration

You can configure the `openai` package to use Azure OpenAI using environment variables. The following is for `bash`:

```
# Set this to `azure`
export OPENAI_API_TYPE=azure
# The API version you want to use: set this to `2023-03-15-preview` for the released version.
export OPENAI_API_VERSION=2023-03-15-preview
# The base URL for your Azure OpenAI resource.  You can find this in the Azure portal under your Azure OpenAI resource.
export OPENAI_API_BASE=https://your-resource-name.openai.azure.com
# The API key for your Azure OpenAI resource.  You can find this in the Azure portal under your Azure OpenAI resource.
export OPENAI_API_KEY=<your Azure OpenAI API key>
```

Alternatively, you can configure the API right within your running Python environment:

```
import os
os.environ["OPENAI_API_TYPE"] = "azure"
...
```

Deployments

With Azure OpenAI, you set up your own deployments of the common GPT-3 and Codex models. When calling the API, you need to specify the deployment you want to use.

Let's say your deployment name is `text-davinci-002-prod`. In the `openai` Python API, you can specify this deployment with the `engine` parameter. For example:

```
import openai

response = openai.Completion.create(
    engine="text-davinci-002-prod",
    prompt="This is a test",
    max_tokens=5
)
```

```
pip install openai
```

```
import os

os.environ["OPENAI_API_TYPE"] = "azure"
```

```
os.environ["OPENAI_API_VERSION"] = "2023-03-15-preview"
os.environ["OPENAI_API_BASE"] = "..."
os.environ["OPENAI_API_KEY"] = "..."
```

```
# Import Azure OpenAI
from langchain.llms import AzureOpenAI
```

```
# Create an instance of Azure OpenAI
# Replace the deployment name with your own
llm = AzureOpenAI(
    deployment_name="td2",
    model_name="text-davinci-002",
)
```

```
# Run the LLM
llm("Tell me a joke")
```

```
    "\n\nWhy couldn't the bicycle stand up by itself? Because it was...two tired!"
```

We can also print the LLM and see its custom print.

```
print(llm)
```

```
    AzureOpenAI
    Params: {'deployment_name': 'text-davinci-002', 'model_name': 'text-davinci-002', 'temperature': 0.7,
```

```
'max_tokens': 256, 'top_p': 1, 'frequency_penalty': 0, 'presence_penalty': 0, 'n': 1, 'best_of': 1}
```