

Iteration 1

Project Setup

In Iteration 0, you set up a discord server and chatbot token. You will use the token and server for the rest of the project as well as the docker and mongo databases created during Lab 1.

Python Virtual Environment

After cloning the project, you will set up your python virtual environment using venv as follows for mac or linux:

```
python -m venv venv
source venv/bin/activate
```

or with the following for windows command (replace `activate.bat` with `activate.ps1` if using powershell)

```
python -m venv venv
.\venv\Scripts\activate.bat
```

requirements.txt

A number of different libraries that are used in the project must be loaded when first beginning development. The list of dependencies are contained in `requirements.txt`. NOTE: If you are running Mac OS, you will need to replace the `tensorflow` library in `requirements.txt` with `tensorflow-macos`. If you are using an M1 mac, please contact Dr. Gannod for assistance on getting your system set up.

When ready, you can initialize your virtual environment with the necessary libraries with the following command:

```
pip install -r requirements.txt
```

You will only need to do this once for each development environment. If you wish to set this up globally, execute the command outside of a `venv` environment.

MongoDB

You must begin by starting docker and the mongodb container that was installed during Lab 1. The easiest method is to use the Docker Desktop application. Within Docker Desktop you can start up the mongodb container.

You must create a database in mongo named `chatbot-database`. Afterwards, you will import the data used to train the learners as follows:

- import dictionary.json

```
mongoimport --db chatbot-database --collection dictionary --type json --jsonArray --uri mongodb://
```

- import intents.json

```
mongoimport --db chatbot-database --collection intents --type json --jsonArray --uri mongodb://loc
```

Configure environment variables

You must modify the `DISCORD_BOT_TOKEN` environment variable in the `.env` file to use the `discord` bot token created during Iteration 0.

Train the learner

Assuming you started the mongo database as described above, you must train the learner in both the base project directory (i.e., `advisor_bot`) and the `advisor_bot/test` directory.

```
python train.py
```

Coding Guidelines

Directories and files

The files and directories of the project contain the following elements:

- data - JSON files needed to train the system
- images - stores images used for documentation
- modules - all source files for the project minus bot.py
- tests - test files using unittest for the project
- bot.py - main driver class for the discord chatbot
- train.py - Python program needed to train the learner
- requirements.txt - listing of all the external modules needed for the program
- README.md - this documentation file

Extending the program

The code for the main conversation module is located in the `modules/states/conversation` directory. When writing code for the state classes, you will include an import statement similar to the following and make reference to classes with the `conversation.` prefix. For instance, if instantiating an instance of the `Business` state you would do the following:

```
import modules.states.conversation as conversation

...
business_state = conversation.Business("Business", self.context)
```

where the first parameter is a programmer defined string, and the second parameter (`self.context`) passes the context of the state model to the state.

Execution

To execute the program, start docker and execute the mongodb container. Then run the following:

```
python bot.py
```

You should see an output that looks similar to the following:

```
2023-03-25 20:45:50 INFO      discord.client logging in using static token
2023-03-25 20:45:50 INFO      discord.gateway Shard ID None has connected to Gateway (Session ID: 96a46
Users:
jgannod_advisor_bot
```