Basic User Instructions and Special Instructions for Grading

Getting Started

1. Install Python 3.8.5.
    a. Navigate to this website: https://www.python.org/downloads/release/python-385/
    b. Download the installer file corresponding to your local machine. For instance, Mac users would click the "macOS 64-bit installer" button to download the installer file for Macs.
    c. Open the installer file and complete the installation steps.
    d. Verify the installation by running the commands `python3 --version` and `pip3 --version` in the command line (Terminal, PowerShell, etc.). The expected output for both commands is the relevant version information.
2. Install Node.js
    a. Navigate to this website: https://nodejs.org/en/download/
    b. Download the installer file corresponding to your local machine. For instance, Mac users would click the "macOS installer" button to download the installer file for Macs.
    c. Open the installer file and complete the installation steps.
    d. Verify the installation by running the commands `node --version` and `npm --version` in the command line (Terminal, PowerShell, etc.). The expected output for both commands is the relevant version information.
3. Install Android Studio and Android emulator
    a. Navigate to this website: https://developer.android.com/studio#downloads
    b. Download the installer file corresponding to your local machine. For instance, Mac users would click the "android-studio-2021.1.1.23-mac.dmg" or the "android-studio-2021.1.1.23-mac_arm.dmg" buttons to download the installer file for Macs.
    c. Open the installer file and complete the installation steps.
    d. Verify the installation by opening the Android Studio application.
    e. Download the Pixel 3XL API 26 Android 8.0 OS emulator.
4. Download/clone the project codebase
    a. Navigate to this GitHub repository: https://github.gatech.edu/gt-cs6440-hit-spring2022/Team-2-Mental-Illness-Chatbot
    b. Click on the green "Code" button and either clone the project using HTTP/SSH or download and open the ZIP file of the project in a location of your choosing.
5. Create Python virtual environment and install dependencies for back end (server) and machine learning models.
    a. Navigate to the parent directory of the project directory.
    b. Run the command `python3 -m venv ./cs6440_practicum_chatbot` in the command line to create a virtual environment called "cs6440_practicum_chatbot".
    c. Run the command source `cs6440_practicum_chatbot/bin/activate` to activate the virtual environment.
    d. Enter the project directory and run the command `pip3 install -r requirements.txt` to install the dependencies to the virtual environment.
6. Install dependencies for front end.

   a. Enter the project directory, then enter the directory called "react-native-app".
   b. Run the command `npm install` in the command line to install the dependencies for the front end.
   c. Run the command `sudo npm install --global expo-client` to install the Expo client for local testing.
7. Run back end.
   a. Enter the project directory (not a subdirectory).
   b. Run the command `python3 server.py` in the command line to run the backend.
8. Run front end.
   a. In a different command line window, enter the "react-native-app" directory.
   b. Run the command `npm start` in the command line to run the front end.
   c. Start interacting with the app, enjoy!

Training, Evaluating, and Testing the Machine Learning Models

1. Training and evaluation for the physical illness prediction (PIP) pipeline
   a. Enter the project directory (not a subdirectory).
   b. The PIP pipeline can be run by executing the "./models/physical_illness_prediction.py" script via command line and specifying the according arguments, which are described below.
      i. --proc_data: set to 0 (default) to process initial physical diseases dataset from scratch, set to any other integer (or omit this argument) to use existing processed datasets.
      ii. --fit_se_nn: set to 0 to fit and evaluate multi-output classifier of neural networks for symptom extraction from scratch, set to 1 to load existing model, set to any other integer (or omit this argument) to skip (default value is 2).
      iii. --fit_pip_rf: set to 0 to fit and evaluate random forest classifier for symptom extraction from scratch, set to 1 to load and evaluate existing model, set to any other integer (or omit this argument) to skip (default value is 2).
      iv. --fit_pip_nn: set to 0 to fit and evaluate neural network classifier for symptom extraction, from scratch, set to 1 to load and evaluate existing model, set to any other integer (or omit this argument) to skip (default value is 2).
   c. Processing/loading the dataset, and fitting/evaluating the PIP models can take time. To see the results of the already-processed datasets and already-fitted/evaluated models, see the "./models/physical_illness_prediction_logs.txt" file. To see the processed physical illness prediction dataset, see the "./models/processed_pip_dataset.csv" file. To see the produced symptom extraction dataset, unzip the "./models/pip_symptom_extraction_dataset.csv.zip" file and see the resulting CSV file. The symptom extraction neural network model is saved in the "./models/pip_se_neural_network.joblib" file. The physical illness prediction random forest model is saved in the "./models/pip_random_forest.joblib" file. The physical illness prediction neural network model is saved in the "./models/pip_neural_network.joblib" file.

   d. To process the dataset and fit the PIP models from scratch, run the command `python3 ./models/physical_illness_prediction.py --fit_se_nn 0 --fit_pip_rf 0 --fit_pip_nn 0 > ./models/physical_illness_prediction_logs.txt` in the command line, from the project directory. This takes hours, so please be patient. To see the results, see the "./models/physical_illness_prediction_logs.txt" file.
   e. To load the existing processed datasets and load/evaluate the saved/pre-trained PIP models, run the command `python3 ./models/physical_illness_prediction.py --proc_data 1 --fit_se_nn 1 --fit_pip_rf 0 --fit_pip_nn 0 > ./models/physical_illness_prediction_logs.txt` in the command line, from the project directory. This takes about half an hour, so please be patient. To see the results, see the see the "./models/physical_illness_prediction_logs.txt" file.
2. Training and evaluation for the mental illness prediction (MIP) pipeline.
   a. Enter the project directory (not a subdirectory).
   b. The MIP pipeline can be run by executing the "./models/mental_illness_prediction.py" script via command line. Run the command `python3 ./models/mental_illness_prediction.py` in the command line, from the project directory.
3. Testing PIP and MIP pipelines with sample messages
   a. Enter the project directory (not a subdirectory).
   b. The PIP pipeline and the MIP pipeline are tested by the "./test.py" script via command line. Run the command `python3 ./test.py > ./response_builder_test_logs.txt` in the command line, from the project directory.
   c. The results of the tests can be inspected in the "./response_builder_test_logs.txt" file.