Gilmore Girls Dialogue Generator Documentation

Author: Yang Yang

Date: 12/9/2018

1. An overview of the function of the code (i.e., what it does and what it can be used for).

The project contains four main parts:

1. Scraped training data (all seasons TV scripts from web)
2. Trained a character-level LSTM neural net model using python library Keras and tensorflow GPU backend on Gilmore Girls TV scripts by actors (Rory vs Lorelai)
3. Saved the model object and weights to make prediction from user input using 40-character sequence
4. Developed a Web-UI to display prediction results and allows interactions through button react component.

This Web-UI can be used to generate texts (length between 100 to 400 characters) given user input sequence. It can be used to compare results from varying diversity parameter, as well as compare results from the two actors using the same seeding sequence.

1. Documentation of how the software is implemented with sufficient detail so that others can have a basic understanding of your code for future extension or any further improvement.

I will describe in detail each of the four parts mentioned above here.

1. The code for web scraping of the Gilmore Girls script is included in the \scrape folder. The output file is .jl and can be read in as a json file.
2. The LSTM model training and prediction code is save in the \source\_code folder.
   1. I subset the dataset by the two actors (Rory and Lorelai) and created character level corpus using all the lines by each of them. The two corpuses are very similar but not 100% the same. For simplicity I separated the modelling part by the two actors into two code files (lstmmodel\_lorelai.py and lstmmodel\_rory.py)
   2. The model training employed tensorflow GPU and saved the two model weights objects as weights\_lorelai.hdf5 and weights\_rory.hdf5.
   3. The predict.py code will take user input (40 characters in length all lower cases), converts this string into the required shape for LSTM model and output the next character based on a probabilistic distribution.
   4. In the \ui folder, execute **gunicorn server:app** and open local browser and type **localhost:8000** to interact with the web UI.
3. Documentation of the usage of the software including either documentation of usages of APIs or detailed instructions on how to install and run a software, whichever is applicable.

Step 1: Create a new environment

After you have cloned the project, I recommend you create a new environment before installing the required libraries.

Go to your terminal and run

**conda create -n myenv**

Note: replace **myenv** with the environment name you prefer

Activate the new environment:

Windows: **activate myenv**

macOS and Linux: **source activate myenv**

NOTE: Replace **myenv** with the name of the environment.

Step 2: Install the following using pip

1. tensorflow
2. keras
3. falcon
4. gunicorn
5. numpy
6. pandas

Step 3: Navigate to the ui subdirectory and execute in command line

**gunicorn server:app**

Step 4: Open any browser, and go to ‘**localhost:8000**’ to access the UI

Step 5: