MovieLike
CS410 - Project Documentation

Shubham Jain (shubham9), Sameksha Reddy(screddy2)

Code link : https://github.com/shubham1310/cs410project

Overview of the code and Documentation:

The repository consists of two main components: imdb crawler and the search engine. The imdb crawler scrapes the reviews and other movie information such as the plot and poster of the movies listed in the movienames.txt. The reviews and titles are stored in dat files which are used by the search engine component which implements an opinion based search. Each of these components is described in detail below.

Crawler

There are two parts of crawling that we did. The first part was to get the movie names, their rating, genres, and their imdb id. The other part is to get the movie review, poster etc. The following is the description of the files for crawling:

- crawler/script/crawl_movie_list.sh: uses the scrapyIMDB folder's code to get the movie names, id etc and saves it inside the data folder
- crawler/scrapyIMDB/spiders/movies_spider.py: is the spider that uses scrapy library to crawl and get the movie details
- crawler.ipynb/.py: crawls the imdb website using the movie names and ids above to get their reviews, poster etc. This stores these things inside the SearchEngine folder in their respective places to be used for the search part.

Search Engine

This section of the project deals with the implementation of the opinion based search engine on top of the data collected from the imdbcrawler (described above). The following is a short description of files contained in this section of the project

reviews: folder containing the data set gathered from imdbcrawler. Contains reviews.dat, titles.dat, moviesnames.txt. reviews.dat contains the reviews scraped from imdb of movies corresponding to the ones listed in moviesnames.txt. moviesnames.txt is a list of movie names for which the reviews have been scraped. titles.dat contain a brief summary displayed on the web page for each movie listed in the moviesnames.txt

static: this folder contains the coffee script files, css files, images and other static files required for the web page. Following are the editable files present within the static folder

- css/custom.css: Add custom styling components to the web page
- image/* : Contains images corresponding to the movies listed in the movienames.txt to be displayed on the web page
- coffeescript/index.coffee: coffee script file for modifying the javascript files used by the web page. (Description of each function is mentioned as comments in the code)
- index.html: Main html page rendered

searcher_server.py: The main python file implementing the search API used by index.html by using flask. Each API is described in detail in the code as comments. Refer to the code for more details regarding individual functions

searcher.py: This python file implements the major search engine. searcher_server.py calls the main search function in this file which is then routed to index.js file. Each function is explained in more detail in the code.

config.toml: configuration file used by the search engine. Analyzers and default search engine parameters are mentioned here

3) Documentation of the usage of the software

Crawler: Detailed description on how to run the search engine is specified in README.md file. The following are the steps that can be done to get the dataset. Also, the data is already uploaded to the repo and since it can take several hours to get the reviews of all the movies, it isn't advice to run this part again:

- To run the crawler to get the movie names and other details, run this (stores in a file in ScrapyIMDB/data/ called movie_list.csv)
  cd ScrapyIMDB/script
  sh ./crawl_movie_list.sh 2017, 2016, 2015, 2014, 2013, 2012, 2011, 2010, 2009, 2008, 2007, 2006, 2005, 2004, 2003, 2002, 2001, 2000
- Then to get the movie reviews:
  we have to the run the crawler.ipynb file. This essentially gets all the reviews and some additional movie details as well like movie plot on main page, poster image etc.
  Or we can run crawler.py directly

Search Engine: Detailed description of how to run the search engine is specified in SearchEngine/README.md file. The below are the steps to set up and run the search engine:

- Install the prerequisite packages by running the commands. All the requirements for this component are mentioned in the requirements.txt file

  ```
  pip install --upgrade pip
  virtualenv meta-pyenv
  source meta-pyenv/bin/activate
  pip install -r requirements.txt

  source meta-pyenv/bin/activate
  import nltk
  nltk.download('punkt')
  ```
- To run the search engine server,

  ```
  python search_server.py config.toml
  ```

- To leave the virtual environment, simply close your terminal or type 'deactivate'

4) Brief description of contribution of each team member:

Shubham Jain: Did the crawler part. Modified code at https://github.com/vitid/ScrapyIMDB to get the movie names and was modified to get more movies as compared to the original numbers. Then wrote the crawler to get the movie reviews, poster, plots etc. Also, did the website layout and added the code to add movie details like poster, plot, genre, rating etc on the webpage. Wrote a small snippet to get the similar words dictionary used in the data expansion

Sameksha Reddy: Worked to setup the search engine, and modified it for the usage for the dataset. Was responsible for implementing multiple preferences opinion based search. This feature allows users to specify multiple preferences in the form of comma separated values. Also implemented query expansion in order to handle similar opinions expressed in multiple ways by the user.