IMDb Scraper

I. Background

This project serves to provide various functions on movies, using the actors of the movies to make connections. For example, we can find the actors with the farthest distance from each other, or we can find the farthest movies from actors. The limitations of the Bacon Number only taking in actors as inputs are not here, as movies and actors can be used in either order when finding the farthest or any arbitrary distances between two movies, actors, or one of each. Additionally, our recommendation algorithms use both triadic closure and neighborhood overlap to recommend the top 10 movies to a user based on either a list of up to 3 input movies or an input of the user's favorite actor. These two recommendation algorithms use the features of the neighborhood in the Graph to provide movies which would likely be similar to the user's taste and recommend them.

The program runs in a basic user shell, where the user can type 'help' to see the commands accepted by the program. After seeing them, the user can simply type a number 1-9 to access the prompt for a command. At this point, the user can provide their input and the graph based on the IMDb database will have the algorithms performed on it to return the output to the user. The dataset was cleaned in a Jupyter notebook, which is provided in the code. Originally, we had wanted to web-scrape the IMDb website ourselves and output the data, but unfortunately IMDb does not allow web-scraping, but using their data from datasets for personal, non-commercial projects is allowed, so we do not violate their Terms of Service this way.

II. Set-up

The IMDb Scraper can be set up by installing the file directory and running the main method in Main class. This uses the Graph package which is implemented in the package/src directory to perform the graph operations on the IMDb dataset, which is found in the data directory. The features of the dataset are not hard-coded, so if an updated dataset is found it can replace the current one in the data directory and the program would continue to work, provided that the columns/features remain the same.

III. Finding Distances

The commands numbered 1-7 involve calculating distances in the Graph. The Breadth-First Search algorithm is instrumental to these calculations. Simply inputting the number of choice

and then the actors/movies desired allows the commands to be run. However, it should be noted that the inputs are case-sensitive. The commands are:

1. What are the farthest movies from (input movie)?
2. Who is the farthest actor from (input actor)?
3. What is the farthest movie from (input actor)?
4. Who is the farthest actor from (input movie)?
5. What is the shortest path between (input movie) and (input movie)?
6. What is the shortest path between (input actor) and (input actor)?
7. What is the shortest path between (input movie) and (input actor)?

However, these need not be memorized, as typing ‘help’ into the user shell will output the list of all user commands.

IV. Receiving Recommendations

The commands number 8 and 9 involve recommending movies to the user. The first of these commands, command 8, recommends movies to a user based on a specific input actor. This works by calculating the neighborhood similarity between the input actor and other actors. The actors who have movies in common with this actor will then have their movies tallied. The movies most common between these actors have the most neighborhood similarity to the actor input and the top 10 of them will be output. The second command, command 9, takes in a list of up to 3 movies as inputs. If the user wants to only input one or two, they can simply hit enter to skip inputting more than one movie. After this, the triadic closure is calculated between the movies and the actors in them to find the top 10 movies which appear most frequently in these triadic closure calculations. Then, these top 10 movies are output to the user.

V. Examples

Figure 1. Using command 9 to find the recommended movies based on “The Dark Knight”, “The Shawshank Redemption”, and “Fight Club”

```
Enter a command or type 'help' for a list of commands:
$ 9
What are some recommended movies based on (input movie(s))?
Input movie 1: $ The Dark Knight
Input movie 2: $ The Shawshank Redemption
Input movie 3: $ Fight Club
The recommended movies for The Dark Knight, The Shawshank Redemption, and Fight Club are:
    Fight Club
    The Dark Knight
    The Shawshank Redemption
    Rescue Dawn
    London Has Fallen
    I'm Not There
    Se7en
    Batman Begins
    Olympus Has Fallen
    The Big Short
```

Figure 1

Figure 2. Using command 2 to find the farthest actors from Leonardo DiCaprio

```
Enter a command or type 'help' for a list of commands:
$ 2
What are the farthest actors from (input actor)?
Input Actor: $ Leonardo DiCaprio
The farthest actors from Leonardo DiCaprio are:
    Val Bettin
    Barrie Ingham
    Susanne Pollatschek
```

Figure 2

Figure 3. Using command 8 to find the recommended movies for Johnny Depp

```
Enter a command or type 'help' for a list of commands:
$ 8
What are some recommended movies for (input actor)?
Input actor: $ Johnny Depp
The recommended movies for Johnny Depp are:
    Alice in Wonderland
    Pirates of the Caribbean: At World's End
    Pirates of the Caribbean: The Curse of the Black Pearl
    The King's Speech
    Alice Through the Looking Glass
    Pirates of the Caribbean: Dead Men Tell No Tales
    Sweeney Todd: The Demon Barber of Fleet Street
    Charlie and the Chocolate Factory
    Corpse Bride
    Ned Kelly
```

Figure 3

Figure 4. Using command 6 to find the shortest path between “The Godfather” and “Scarface”

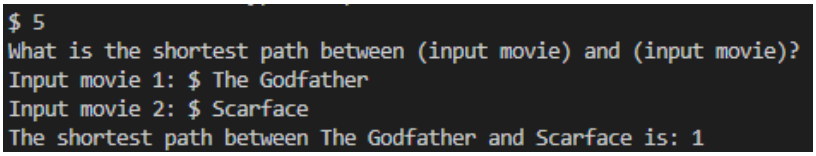

Figure 4

V. Exiting the Program

Exiting the program is extremely simple and can simply be achieved by typing ‘exit’ into the user shell.

VI. References

Program code and documentation written by Andrew Lukashchuk and Hassan Rizwan.

Regex help provided by Bart Kiers on StackOverflow in the following pos from 2009t: https://stackoverflow.com/questions/1757065/java-splitting-a-comma-separated-string-but-ignoring-commas-in-quotes