Comp Sci 839 - Stage 2

Somya Arora, Mohammed Danish Shaikh, Swati Mishra (Group 16)

Crawling and Extracting Structured Data from Web Pages

I. Web data sources

We selected two Web sources in the domain of *Movies*:

A. IMDb (Internet Movie Database)

Website Link: *https://www.imdb.com/*
Web seed page used: *https://www.imdb.com/sci-fi*

Originally a fan-operated website, the database is owned and operated by IMDb.com, Inc., a subsidiary of Amazon.
The web page crawled and used for extraction is the list of sci-fi movies sorted according to popularity of the movies. The popularity measure is estimated through multiple factors including average user rating and number of votes. This page contains 150,619 movie titles out of which top 5000 have been extracted.
We selected this Web source because it is the largest easily accessible database.

B. TMDb (The Movie Database)

Website: *https://www.themoviedb.org/*
Web seed page used: *https://www.themoviedb.org/movie*

The Movie Database (TMDb) is a crowd-sourced, community built movie and TV database and is not owned by any for-profit corporation.
This web page contains a list of 13846 movie titles sorted on popularity. These movies span across all genres and are not limited to sci-fi. The formula of popularity is different than the one in IMDb, which means the two datasets may differ widely and yet have some intersections due to the popular sci-fi movies. Here again, top 5000 movies have been extracted.

II. Extraction process

The extraction process followed was as follows:

A. Select two web sources containing template based data about overlapping entities. We chose IMDb and TMDb to extract *Movies*.
B. Select an Open Source extractor capable of extracting information from the selected web sources. We chose *Scrapy*.
C. Decide attributes to be extracted from the selected web sources. We chose movie *name, duration, release year, genres and actors* as the attributes to be extracted.
D. Analyze the selected web sources to come up with rules to extract the attributes mentioned in step C. We used a combination of *CSS* and *XPath* based rules for extracting the attributes from the web sources.
E. Analyze the selected web sources and come up with validation rules for eliminating extracted data that is not clean. The motivation behind performing this process was to ensure that the data we obtain is as clean as possible, so that we can proceed with the next stage of the project without any difficulties. We eliminated the extracted tuple from web sources which fulfilled any one of the following conditions:
    1. Any of the attributes, i.e. (name, year, duration, genre, actors) in the extracted tuple is *empty*.
    2. The extracted tuple belongs to a TV series. These tuples were eliminated because TV series usually run for several years, hence can't really be mapped to a single year.
F. Write spiders for extracting data from the selected web sources, using the extracting rules. Initialize each spider with a seed (initial) web page.
G. Start the extraction process using the spider. Continue iterating through several pages of each web source until the spider has extracted a given number of clean (refer step E) tuples (for example, 5000) from the corresponding web source. Save the tuples in a CSV file.
H. Open the CSV files, eyeball through the data to check if there are any data cleaning issues. If so, add a rule for eliminating such tuples from getting extracted and repeat the extraction process (Steps F and G).
I. Eyeball through both CSV files to ensure they have some tuples in common.

III. Data Description

A. Type of entity:

Entity extracted is *Movie*. The tables from both the databases have same schema with the following five attributes:

1. *Name* of the movie.
2. *Year* of release.
3. *Duration/Runtime* of the movie.
4. *Genre* of the movie (e.g. Action, Adventure, Sci-Fi, Mystery, etc).
5. The cast or *Actors* involved in the movie.

Since genre and cast may be multiple, it is a comma separated concatenation of various strings as can be seen in the examples below.

B. Table schemas:

The schema of the imdb.csv table:

ID	Name	Year	Duration	Genre	Actors

Below are few examples of the tuples in the imdb.csv table:

3	Aquaman	2018	143 min	Action, Adventure, Fantasy	Jason Momoa,Amber Heard,Willem Dafoe,Patrick Wilson
7	The Meg	2018	113 min	Action, Horror, Sci-Fi	Jason Statham,Bingbing Li,Rainn Wilson,Cliff Curtis
25	Interstellar	2014	169 min	Adventure, Drama, Sci-Fi	Matthew McConaughey,Anne Hathaway,Jessica Chastain,Mackenzie Foy

The schema of the tmdb.csv table:

ID	Name	Year	Duration	Genre	Actors

Below are few examples of the tuples in the tmdb.csv table:

3	Aquaman	2018	2h 24m	Action,Adventure, Fantasy,Science Fiction	Jason Momoa,Amber Heard,Willem Dafoe, Patrick Wilson,Nicole Kidman
27	Escape Room	2019	1h 39m	Horror,Thriller, Action,Science Fiction,Adventure, Drama,Mystery	Taylor Russell,Logan Miller,Deborah Ann Woll,Tyler Labine,Jay Ellis
4995	Boogeyman 3	2008	1h 34m	Horror	Erin Cahill,Chuck Hittinger,Mimi Michaels,Matt Rippy,Nikki Sanderson

The schema of both the tables are same and thus there will not be any need to carry out any kind of schema alteration at the start of the entity matching process in the next project stage (Stage 3).

The duration of the movie is represented in integral number of minutes in the IMDb source eg. 181 min. On the other hand, the duration of the movies from TMDb source is of the format <*x*>h <*y*>m representing x hours and y minutes long movie, e.g. 3h 1m.

The total number of tuples in both imdb.csv and tmdb.csv is **5000**.

IV. Open-source tools used

Scrapy, an open source and collaborative framework for extracting the data you need from websites in a fast, simple, yet extensible way.
Scrapy is an application framework for crawling web sites and extracting structured data which can be used for a wide range of useful applications, like data mining, information processing or historical archival.
Even though Scrapy was originally designed for web scraping, it can also be used to extract data using APIs (such as Amazon Associates Web Services) or as a general purpose web crawler. Further documentation about the tool can be found here.