CS839 Project Stage 2: Crawling and Extracting Structured Data from Web Pages.

Date: 3/24/2018 Team name: Big HIT
Team members: **H**oai Nguyen, **I**saac Sung, **T**rang Vu

Summary: This report describes how we extracted structured data from two web sources. Table 1 below summarizes the minimal information required for the project report. We provide a more detailed description of the project in sections following this table.

Table 1: Project Summary

Entity	Movie	
Movie attributes (17 attributes)	title, cast, directors, writers, genres, keywords, content_rating, run_time, release_year, languages, rating, budget, revenue, opening_weekend_revenues, production_companies, production_countries, and alternative_titles	
Web sources	IMDb (Internet Movie Database) [1]	TMDb (The Movie Database) [2]
Extraction method	Manually extracted movie attributes by parsing each movie's HTML page using Beautiful Soup [3]	
Open-source tool	Beautiful Soup [3]: a Python package that allows us to navigate and search for specific elements in the HTML pages' structure	
Final CSV table	IMDb movies	TMDb movies
	3500 tuples	**5490** tuples

1) Entity and Web Data Sources

We extracted movie attributes from two movie databases, the Internet Movie Database (IMDb) [1] and The Movie Database (TMDb) [2]. While IMDb is a much older and larger database that contains lots of information related to movies, TV shows, video games, production crew, cast, and many more [3], TMDB is a younger, and leaner database that currently contains information mostly about movies, TV shows, cast and crew. Regardless, each database has sufficient number of movies and overlapping movie attributes that make up the structure data for us to extract. The movie attributes and their description are listed in Table 2 below. There are 13 overlapped attributes between IMDb and TMDb. Four attributes are unique to the IMDb.

Table 2: Movie Attributes' Description. The attributes shaded in grey are unique to IMDb movies
(*) Only directors listed in the movie's main page are extracted
(**) Only writers listed in the movie's main page are extracted.

Attributes	Description	Attributes	Description
title	Title of the movie	languages	The languages that appear in the movie
cast	The name of the top 5 billed actors/actresses	rating	Users' rating score
directors	Name of directors(*)	budget	Movie's budget
writers	Name of screenplay/novel writers(**)	revenue	Movies' revenue
genres	Movie genres (eg. action, adventure, thriller)	opening_weekend_revenues	The revenue of the movie in the opening weekend
keywords	Keywords from movie's plot	production_companies	The companies that produce the movie
content_rating	Movie's certification (eg. R, PG, PG-13)	production_countries	The countries that produce the movie
run_time	Duration of movie	alternative_titles	Other titles of the movie
release_year	The year in which the movie is released		

2) Extraction Method and Tools

2.1) Method:

We used a manual wrapper construction method. We manually examined HTML source codes for a few movies in both IMDb and TMDb to find the locations of the attributes that fit in our constructed schema above. For example, in IMDb movies, the name of a movie's director may appear in the HTML page's source code as shown in Figure 1 below.

```
1267
1268    <div class="credit_summary_item">
1269       <h4 class="inline">Director:</h4>
1270          <span itemprop="director" itemscope itemtype="http://schema.org/Person">
1271 <a href="/name/nm0000108?ref_=tt_ov_dr"
1272 itemprop='url'><span class="itemprop" itemprop="name">Luc Besson</span></a>          </span>
1273    </div>
1274    <div class="credit_summary_item">
1275       <h4 class="inline">Writer:</h4>
1276          <span itemprop="creator" itemscope itemtype="http://schema.org/Person">
1277 <a href="/name/nm0000108?ref_=tt_ov_wr"
1278 itemprop='url'><span class="itemprop" itemprop="name">Luc Besson</span></a>          </span>
1279    </div>
```

Figure 1: Example of how attribute is organized in a HTML page.

Using Beautiful Soup, we could extract the name ‘Luc Besson’ as the director of the movie. The Python code for this will be something like the following statement:

```
directors_list = bs.select('div.credit_summary_item span[itemprop="director"] a span')
```

in which bs is the Beautiful Soup object representing the nested data structure of the HTML page. Details of the extraction rules for other attributes can be found in the ‘imdb.crawler.py’ and ‘themoviedb_crawler.py’ in the code directory. To ensure that the final csv tables share some common movies, we chose to extract attributes for popular movies from each movie database, assuming that if a movie is popular, its rating will be high in both sites.

2.2) Tools:

BeautifulSoup[3] was used to parse the website HTML. BeautifulSoup allowed us to search and manipulate the DOM-tree of the websites to extract the necessary info for the wrappers.

3) Description of Output Data Tables

Both tables consist of the attributes listed in **Table 2** above**.** The IMDb table contains 3,500 tuples while TMDb table contains 5,490 tuples.

References:

[1] IMDb: http://www.imdb.com/
[2] TMDb: https://www.themoviedb.org/?language=en
[3] BeautifulSoup: https://www.crummy.com/software/BeautifulSoup/bs4/doc/