CS 839 : Stage 2 Report

Abbinaya Kalyanaraman Bob Effinger Rajan Dalal

Team 9

Web Sources

For Stage 2, our group decided to gather information on movies. For this, we gathered the information from the following two web sources:

Metacritic

Metacritic aggregates reviews of movies from leading critics. They have a sizeable collection of movies on their website that we decided to extract based on scores.

IMDB

IMDB has a page that lists movies based on popularity, with the release year, along with some information about each movie. By extracting the information based on popularity, we can gather a sizeable amount of movie data.

We decided on these web data sources after browsing through various other websites which provided movie information. We looked for two web sources that would provide a large number of common attributes and then we selected the set of attributes that we wanted to extract from the two sources.

Data Extraction

Metacritic

The data in Metacritic was initially sorted based on scores.

The first step we took towards extraction was removing the appropriate head and tail from the HTML data. This is so we would have a more focused view on the HTML file segments containing our table, and to make it easier to find the data that we wished to extract. After eliminating much of the extra information from our HTML scrapped data, we then extracted each of the attributes that we were looking for. For Metacritic we found the attributes:

```
movie_title
release_date
movie_meta_rating
movie_user_rating
movie_summary
```

We extracted these by regular expression matching using the default regex package in python.

`movie_meta_rating` here refers to the official Metacritic rating, rated out of a 100, while `movie_user_rating` refers to the aggregate rating by Metacritic users, rated out of 10. The identifying patterns we used for our regex matching were HTML tags. The various Metacritic rating tags were converted to a single tag before extraction.

IMDB

The data in IMDB was sorted based on popularity. As for Metacritic, we first eliminated large parts of the HTML by segmenting on the head and tail of the required data. From the IMDB raw files, we extracted:

```
movie_title
release_year
movie_rating
runtime
movie_genres
movie_summary
num_imdb_votes
movie_gross_collection
```

`movie_rating` here refers to the IMDB users' score, rated out of 10.

Entity and Schema Description

As mentioned above, we decided to extract the entity movies.

From the two tables, we selected a common schema of:

column name	data type	data description
`title`	`string`	the movie title
`release_year`	`string`	the release year of the movie
`movie_rating`	`floating point`	the user rating the movie was given in the specific source
`summary`	`text`	the summary from the specific source

`release_year` is a `string` as it could have a value of 'TBA.' We use user rating from both tables for the reason that since IMDB only provides a user score, a comparison with a Metacritic user score would be more applicable than one with an official score.

From our Metacritic data, we extracted 3100 rows, and from our IMDB data, we extracted 3250 rows. We estimate there are over 500 common entities between the two tables.

Open Source Tools

The tools we used for this stage were python and the default regex package (`re`) within python for the data extraction. The raw files were obtained manually from the websites.