PRI - Information Processing and Retrieval, 2022/2023

Carlos Veríssimo
up201907716@up.pt

Chloe Vidal
up202202860@up.pt

Joaquim Monteiro
up201905257@up.pt

Faculdade de Engenharia da Universidade do Porto

This report was developed within the scope of the curricular unit of "PRI" - Information Processing and Retrieval practical project, which is divided into three incremental milestones.

Keywords include: data, processing, information, collection, datasets, characterization

1 Introduction

As part of the curricular unit, students need to develop a project whose final goal is to implement a search system using a dataset of their choice. The project is divided into several milestones/deliveries. Each milestone focuses on a specific part of the process of building the information search system, which includes identifying, collecting and preparing data, to optimize querying and information retrieval.

For the theme of their project, students were required to choose a dataset that included a significant amount of textual information, with a good sample size. We looked into multiple possible datasets. They had different subjects: countries, governments and even music lyrics. In the end, we decided to settle for a dataset related to movies and TV shows. This decision was mainly due to the fact that the dataset was provided by a legitimate source: Internet Movie Database – *IMDb*[1]. Another reason that comforted our choice was the great amount of textual data from forums and other sources, that we can gather using the movies titles. Gathering data from forums mostly involves data scraping from Reddit.

2 Data Preparation

The first deliverable consists in the collection/preparation and characterization of the chosen dataset. Our project is divided in two main data entities: Reddit data and IMDb data. As previously mentioned, the platform is built around the IMDb dataset, which contains records of millions of movies, shows, articles about those medias and about the actors starring in them. Every information wasn't always relevant to our project. What we were interested in were the textual data, such as reviews and comments, which is obtained through scraping/crawling of the Reddit forum r/movies[2].

2.1 IMDb Data

IMDb provides access[3] to a variety of datasets, which are available to the public for personal and non-commercial use. The datasets are updated daily and follow the TSV[4] format. We downloaded the datasets that seemed most relevant to our project : *title.basics.tsv* and *title.ratings.tsv*

2.1.1 title.basics table

Contains information about titles.

- 9 267 897 records
- 927 MB in size
- Titles are identified by the **tconst** column
- 9 columns

2.1.2 title.ratings table

Contains the IMDb rating and votes information for titles

- 1 262 870 records[5]
- 21 MB in size
- Titles are identified by the **tconst** column
- 3 columns

After looking under the hood, we assessed the quality and validity of the data. Data is well-structured. There are no duplicate rows, and missing values are highlighted by a particular pattern : "\N". While we were looking for the missing data, we were able to determine that missing cells would not impact the normal flow of the project. It can be explained by the fact that they appeared to be in columns that were either:

[1] https://www.imdb.com/

[2] https://www.reddit.com/r/movies

[3] https://www.imdb.com/interfaces/

[4] Tab-separated values

[5] The discrepancy of records between the datasets is due to the fact that some titles do not have any ratings/reviews

not used to make queries, or were removed at a later stage.

2.1.3 Pipeline

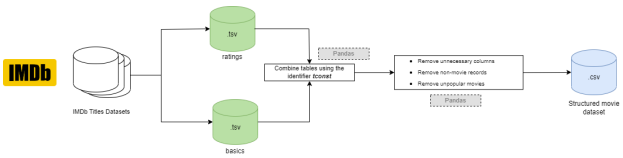

Figure 1: IMDb Data Pipeline

As you can observe in the figure 1, representing the IMDb Data Pipeline, we chose to :

- Combine the two datasets (Basics and Ratings) into one, using the **tconst** column, since the attribute was present in both tables.

For each movie title, we can now also have access to the number of votes given by viewers/IMDb users, as well as the average rating.

- Remove all records such that we are left with only movies. For our project, we needed to have a limited data subset, that we could easily manipulate. So we decided to focus on popular movie and discard some items such as TV Shows related information, movies with worse rating, etc.
- Clear away unnecessary columns:
  - **isAdult**: Whether a certain title is R-rated
  - **endYear**: TV Series end year. As we're dealing with only movies, we removed it.
  - **titleType**: The type/format of the title. As we're left with only movies, we removed it.
  - **originalTitle**: original title, in the original language. We're only interested in the **primaryTitle**, so we remove this column
- Only keep movies that are popular

Our project is set and developed on a decade-based approach, i.e., 2000s movies need to have more votes to be included in the final dataset, while 2010s and 2020s movies can be included even if they have a significantly less amount of votes than older ones.

Note: The term "popular" is widely used in this report and should be perceived as the movies that currently have a lot of community discussion. They might be older movies which people still talk about, or newer releases that make headlines.

- Save the dataset.

2.2 Reddit Data

Reddit is a website that offers an up-to-date stream of news, stories, pictures, and videos on a great number of topics. Communities of people are gathered in this network to share on the subject of their interest. The /r/movies subreddit is a forum for people to discuss and share, post and comment about movies. On this subreddit, users can have access to discussions, movie trailers, movie posters, news and articles on movies, and so on.

We decided to focus on the former, the discussions and comments, which are more relevant to our project.

2.2.1 Collection Process

pushshift.io provides a RESTful API to access and search through its archived Reddit posts and comments. We used it to collect all the posts (and comments) from /r/movies by requesting the most recent posts, then repeatedly asking for the posts older than the last received post, until there are no more posts left.

2.2.2 Pipeline

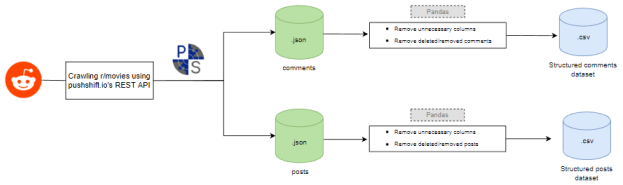

Figure 2: Reddit Data Pipeline

- Dispose of unnecessary data (this is done while the posts are being saved)
  - For each post, we keep its ID, title, flair text, author, score, timestamp and content.
  - For each comment, we keep its ID, the ID of the post, the ID of the parent comment, its text, author, score and timestamp.
- Eliminate deleted and removed posts/comments.
- Filter by posts with 20 or more upvotes, to get rid of irrelevant/low quality content.

2.3 Data Characterization

Data for the project in divided into three datasets: *movies.csv*, *posts.json* and *comments.json*

Before filtering out unpopular movies, this is what our *movies.csv* dataset looked like in terms of yearly movie releases.

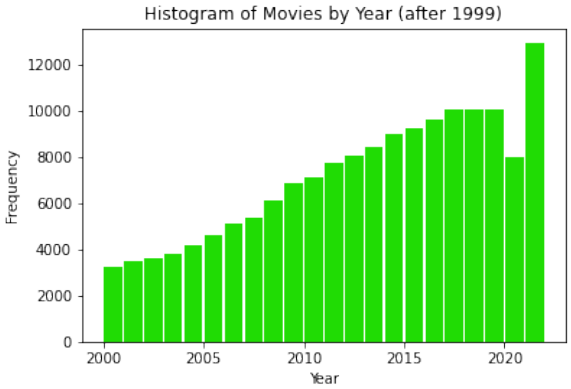

Figure 3: Movies Histogram

The figure above portrays the dataset after its modification to contain only the movies records. As expected, it is a right skewed histogram. The same distribution for the popular movie dataset can be found below:

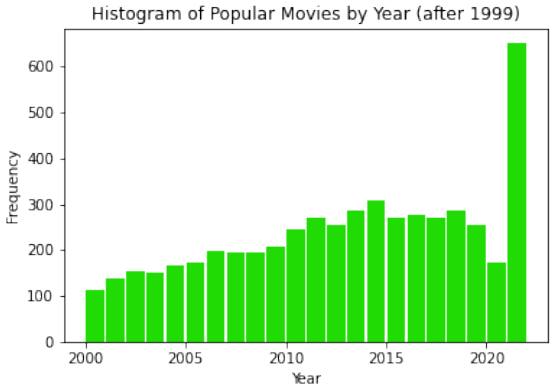

Figure 4: Popular Movies Histogram

There are around 6,800 entries of movies in the final movie dataset. The oldest one, *The Cabinet of Dr. Caligari*, was released in 1920. By looking at figure 2, we can clearly see the impact of the *COVID-19 pandemic* on the movie industry. Figure 3 presents the distribution of popular movies throughout the 21st century. The year with the most popular movies (besides 2022) is 2014, with 307 movies.

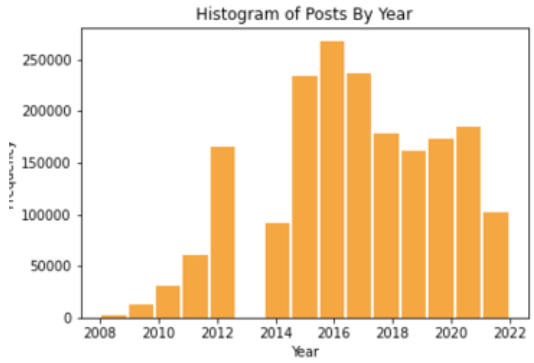

Figure 5: Posts Histogram

In the 4th figure, we can notice that the year 2013 stands out. There were no scraped posts from 2013. The reason for that was apparently a server shutdown on the pushshift.io API, caused by a hard drive failure[6].

Besides that particular year, we can witness that the top 3 most active years for the subreddit were 2015, 2016 and 2017, all three surpassing 250 000 posts.

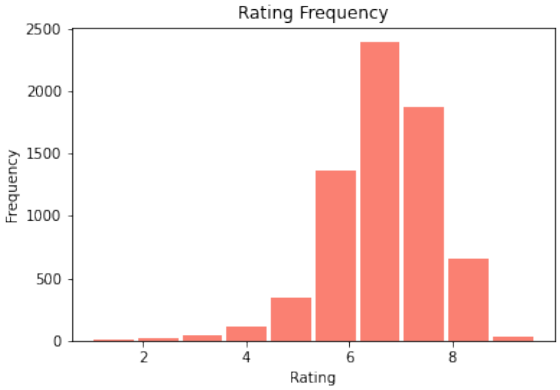

Figure 6: Movie Rating Distribution

The average rating sits at 6.64/10, and the best rated movie is *The Silence of Swastika*, at 9.6/10.

[6]No posts from 2013?

Figure 7: Word cloud for Post Title

We built a word cloud based on the titles of the posts provided by the r/movies subreddit. After a careful observation, we can conclude that some of the most used words are: **great, scene, favorite, Box Office, character, Official Discussion, actor**. These are some of the words that might be searched by the users, in the platform.

2.4 Data Domain Conceptual Model

After collecting and preparing the data, we ended up with the following model:

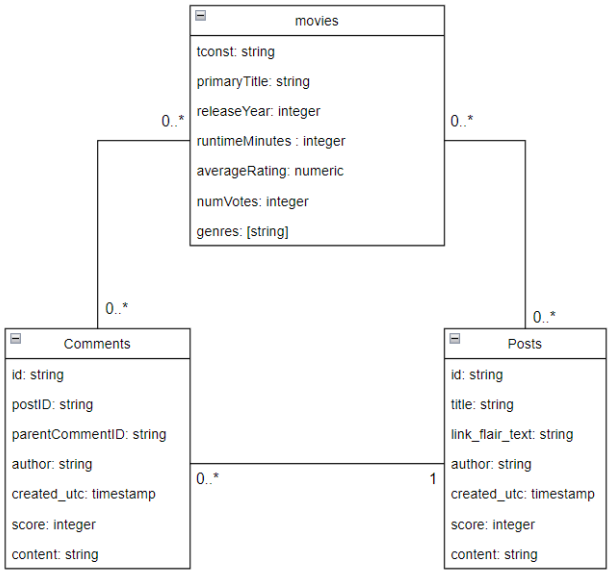

Figure 8: Data Conceptual Model

Our main dataset is **movies** and is composed by the following columns/attributes:

- **tconst**: alphanumeric unique identifier of the title
- **primaryTitle**: the more popular title / the title used by the filmmakers on promotional materials at the point of release
- **releaseYear**: represents the release year of a title
- **runtimeMinutes**: primary runtime of the title, in minutes
- **averageRating**: weighted average of all the individual user ratings
- **numVotes**: number of votes the title has received
- **genres**: includes up to three genres associated with the title

Each one of the movies can be linked to many comments and many posts. The **comments'** table has the following attributes:

- **id**: unique identifier of the comment
- **postID**: post to which the comment is responding to
- **parentCommentID**: if responding to a comment, this is the id of that comment. Otherwise, it's the postID.
- **author** of the comment
- **created_utc**: date of creation of the post
- **score**: Number of up/down votes the comment has received
- **content** of the comment

While the **posts'** table, has the following attributes:

- **id**: unique identifier of the post
- **title** of the post
- **link_flair _text**: post's flair/tag
- **author** of the post
- **created_utc**: date of creation of the post
- **score**: Number of up/down votes the post has received
- **content** of the post

2.5 Prospective Search Tasks

We intend on having two main types of search on the platform:

- **Search by Movie**: In this type of search, users should input a movie name. After that, they will be shown posts and comments in which that movie is mentioned.
- **Search by comment/post title**: In this type of search, users will write a sentence or even just some words. As a result, they will get the movie title or posts/comments that contain those particular words/phrases.
  - E.g.: If users search for "DiCaprio was great in this movie", they will probably be shown comments for movies in which Leonardo DiCaprio is starring, such as "The Wolf of Wall Street", "Titanic", "Shutter Island"...

  In both kinds of searches, the user will be able to filter the search results by the type of the results (comment or post title), release year, genre, rating...

References

[1] Awesome Data. Awesome public datasets. https://github.com/awesomedata/awesome-public-datasets, 2014. Last access: 26-09-2022.

[2] IMBD. Imdb datasets. https://www.imdb.com/interfaces, 2022. Last access: 09-10-2022.

[3] Sérgio Sobral Nunes. Pri tutorials. https://git.fe.up.pt/pri/tutorials/, 2022. Last access: 07-10-2022.

[4] Pushshift. pushshift.io api. https://redditsearch.io/, 2022. Last access: 08-10-2022.

[5] Reddit. r/movies. https://www.reddit.com/r/movies/, 2022. Last access: 09-10-2022.

[1] [2] [3] [4] [5]