Book Characterization using Project Gutenberg's Open Library with Goodreads Reviews

João Baltazar
up201905616@up.pt
Faculty of Engineering of the
University of Porto
Porto, Portugal

Nuno Costa
up201906272@up.pt
Faculty of Engineering of the
University of Porto
Porto, Portugal

Pedro Gonçalo Correia
up201905348@up.pt
Faculty of Engineering of the
University of Porto
Porto, Portugal

Abstract

Catalogs of books and other texts are more available now than ever, as online libraries aggregate public domain literature. As such, rich unstructured data is plentiful but mostly unexplored. In this paper, we seek to find and showcase that underlying information by characterizing books based on their theme, word count, time period, and popularity. By further extracting book corpora and characteristics from Project Gutenberg's Open Library and joining them with Goodreads' reviews, we seek to fulfill prospective search tasks such as finding specific excerpts or understanding the context for a given book quote.

CCS Concepts: • **Information systems** → **Data mining**; **Information integration**.

Keywords: information theory, information retrieval, information processing, data characterization

1 Introduction

Data has become more and more accessible in recent years. Massive bodies of text, such as books, have become readily available in large online e-book libraries like Project Gutenberg's [3]. However, processing that same data and retrieving relevant information, such as patterns and trends, is neither automated nor easy. Finding context for a given book quote, for example, is a significant information need with a lackluster response, as misquotes are still frequently spread around. Open-source libraries do exist, but are not user-friendly, namely, with a dire lack of useful indexing and complementary information. This way, compiling and characterizing books with Project Gutenberg's Open Library and further joining them with Goodreads [2] Reviews appeals to both the archivists and casual readers alike: a big collection of plain text literature, processed and fully searchable, and a hub of recommendations and free books, tagged with genres, themes, time periods and authors.

2 Dataset

The data retrieval pipeline is shown in Appendix A.

The produced dataset has 8000 book entries, making up around 3.2 GB of memory space, and 8300 review entries (10 reviews each for 830 books), making up around 18 MB of memory space. Each book entry includes the book transcription, as well as its bibliographic information. Of the 8000 book entries, around 2700 of them have the overall rating.

2.1 Data Collection and Processing

The first step to collect data was to scrape the catalog of the most popular books in Project Gutenberg, retrieving the identifiers of each book, in fetch_book_ids.py. For each id retrieved, the book information was scraped from the respective webpage. This information underwent several filtering and processing stages in different scripts. Duplicate books that appear as different versions in Project Gutenberg were removed, keeping the version marked as "improved" if such information exists, or the first version processed otherwise. Books that did not have English as their language was removed. To ensure that the scraped content is free to use, books that did not have "Public domain in the USA" as copyright status was also removed.

After filtering and collecting each book's information into *books_info.jsonl*, the transcription of the books was also fetched from Project Gutenberg. The transcription file was processed, removing its header and footer, as well as single new lines, since they do not correspond to real paragraphs. Double new lines, which correspond to a single paragraph, were converted into single new lines. Each processed transcription was stored in a different *json* file, along with the respective book information.

For each book title fetched from Project Gutenberg's open library, a search query was made on Goodreads. To associate a book with the proper reviews, the python script *fetch_goodreads_ids.py* chose the top search result with a matching author name (when applicable).

The fetchers for reviews and ratings, however, proved to be a harder task. As reviews were loaded through *JavaScript*, pure HTML fetching was not possible. Even when using a time delay for the request's response, the list of reviews would still not be ready. As such, a different package, called **selenium** was used.

With **selenium** [4], an automated browser window would be created for each book. Whenever the previously mentioned *JavaScript* event was executed, the tool would recognize the changes on the HTML code and allow the script to properly scrape the review corpus.

This method, however, proved to be significantly slower when compared to the book scraper, which used simple HTTP requests. As such, the volume of review data that was possible to actually obtain was smaller. As such, ten reviews were fetched per book review page, ordered by relevance.

The overall rating information retrieved was stored inside the file with the respective book transcription, while the list of reviews of each book was stored in a different *json* file.

2.2 Data Source and Quality

The data sources were selected because of their large availability of free-to-use and reliable information. Project Gutenberg is an online library of over 60000 free eBooks that provide textual transcriptions for each book along with its bibliographic record. Goodreads is a website focused on book recommendations, from which it is possible to gather book ratings and reviews.

In terms of data completeness, the aforementioned data retrieval pipeline was carefully assembled in order to preserve as much information as possible from the sources. Uncertain dates of birth and/or death were accepted as they reflect real gaps in historical records. Furthermore, there are some null, *Anonymous* or *Various* writers, which also are broadly attributed to either faulty historical records, or works which, due to their nature, have no authors, like fables or religious/institutional texts; or multiple authors, like scientific journals. The data is timely, as both the books and reviews are up-to-date, and extracted directly from the source for this purpose. Regarding data correctness, there is occasionally some trace of Project Gutenberg's footer on the end of texts due to its non-standard nature, although these are rare occurrences. All data is consistent, in plain English, and adequately formatted to *json* files in a standard structure according to the conceptual model.

2.3 Conceptual Model

The produced dataset can be modelled as follows:

A book is defined by its title, the release date on which it was published on Project Gutenberg's platform, its Goodreads average rating, number of ratings, and number of reviews as well as the book corpus.

Each book can have zero to multiple subjects associated with it, which are defined by a name. They can also have zero to multiple authors (some books have no known author, such as the Bible), which are defined by their name and year of birth and death.

Books can also have Goodreads' user reviews associated with them. Each of these reviews is defined by the review date, the name of the reviewer, the number of likes the community gave to that review as well as the rating given to the book, and the review comment (text).

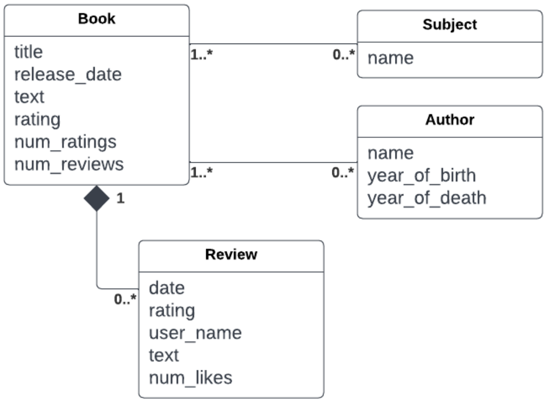

Figure 1. Dataset Conceptual Model

2.4 Data Characterization

There's a lot of data in this dataset, especially when it concerns the text fields such as the book text and the review text. The following tables and plots try to showcase some of its patterns.

Table 1. Top 10 most used words in the books

word	count	frequency
the	35999199	0.065195
of	19749394	0.035766
and	17721901	0.032095
to	14476953	0.026218
a	13785780	0.024966
in	10074152	0.018244
that	6193965	0.011217
was	5648257	0.010229
he	5636281	0.010207
I	5369173	0.009724

As Table 1 showcases, the books roughly follow Zipf's law [5], supporting the reliability of the textual data. Disregarding *stop words* [1], the most common words in our books dataset are:

Table 2. Top 10 most used words in the books - excluding *stop words*

word	count	frequency
time	859451	0.003968
man	859450	0.003968
great	686541	0.003170
good	591981	0.002733
day	575192	0.002656
men	558979	0.002581
life	492075	0.002272
long	488925	0.002258
place	396396	0.001830
people	386386	0.001784

As for the book categories, there's a total of 5182 and the top 10 are as follows:

Table 3. Top 10 most common book categories

category	count	frequency
short stories	485	0.024288
science fiction	407	0.020382
united states	348	0.017427
great britain	236	0.011818
england	233	0.011668
world war	232	0.011618
adventure stories	193	0.009665
fiction	185	0.009264
conduct of life	152	0.007612
detective and mystery stories	148	0.007411

The most frequent categories are consistent with major literature genres, such as fiction, science fiction, or adventure stories, the language of the content - United States, Great Britain, England - and the historic time period with the most representation on the dataset (World War).

As for the authors, the following plots showcase their distribution in terms of birth and death years:

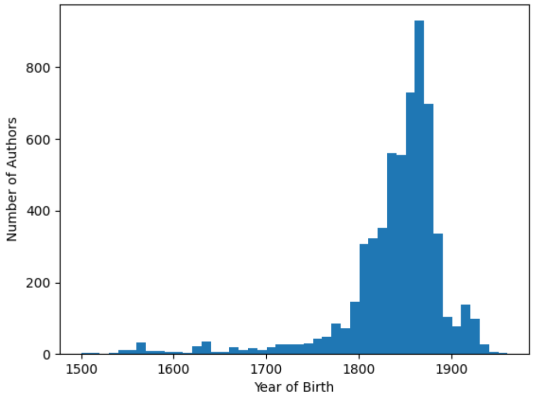

Figure 2. Authors' year of birth distribution

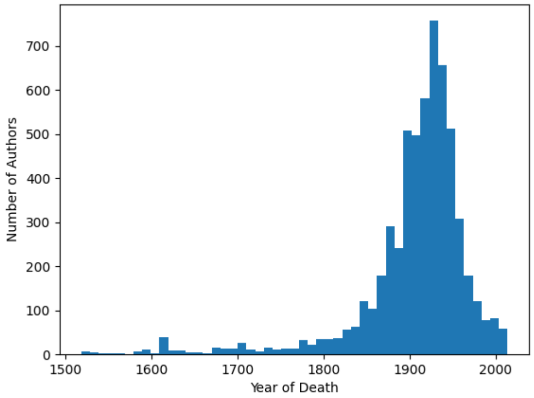

Figure 3. Authors' year of death distribution

As such, we can conclude that most books in the dataset were written between the early 19^{th} and mid 20^{th} century.

The authors with the most books in the dataset are the following:

Table 4. Top 10 most common authors

author	count
Mark Twain	36
Charles Dickens	27
Edward Bulwer-Lytton	25
Horatio Alger	23
William Henry Giles Kingston	23
William Shakespeare	21
Georg Ebers	21
Honor de Balzac	19
George Manville Fenn	18
Arthur Conan Doyle	18

As for book review ratings, their statistics are as follows:

Table 5. Rating value distribution

mean	3.707
standard deviation	0.581
minimum	1.000
25% percentile	3.420
50% percentile	3.770
75% percentile	4.010
maximum	5.000

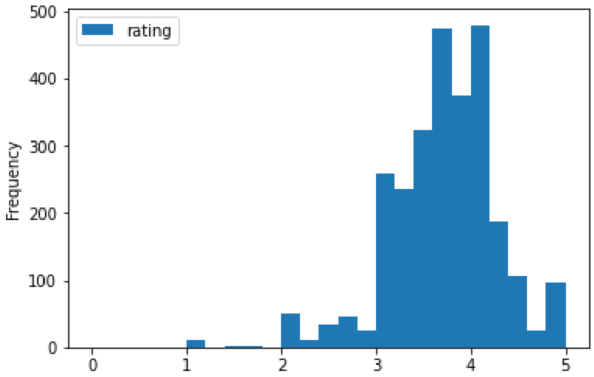

Figure 4. Distribution of the book rating values

Most of the books' average rating fall between 3 and 4 points (out of 5).

When plotting the mean number of words in a review according to the rating (whole number from 1 to 5) given in that same review, an interesting relationship can be noticed:

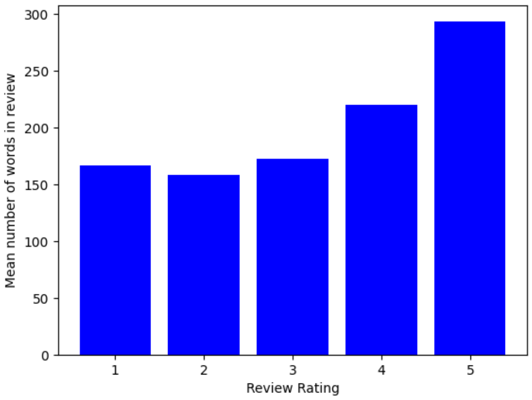

Figure 5. Mean number of words in reviews per rating

As such, we can observe that users tend to write significantly more when they give the book a positive rating, taking the time and effort to justify the score with more depth.

3 Search Scenarios

The resulting information search system will set out to fulfill search scenarios with added value, with a focus on responding to information needs that are not well attended to already, namely:

- I want to find excerpts about a specific concept or event.
- I want to find context on a book quote.
- I want to read a book from a certain time period.
- I want to understand what other people think of a book.
- I want to browse an extensive library of copyright-free literature.

4 Conclusions and Future Work

The data pipeline produced a sizeable and representative sample of texts and reviews in the English language that will serve as a basis for an information search system. We plan to implement a system capable of satisfying the search scenarios proposed, using Elasticsearch and the produced dataset.

References

[1] 2008. Dropping common terms: stop words. https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html

[2] 2022. Meet your next favorite book. https://www.goodreads.com/

[3] 2022. Project gutenberg. https://www.gutenberg.org/

[4] 2022. The selenium browser automation project. https://www.selenium.dev/documentation/

[5] Zipf G. 1936. The Psychobiology of Language.

A Pipeline

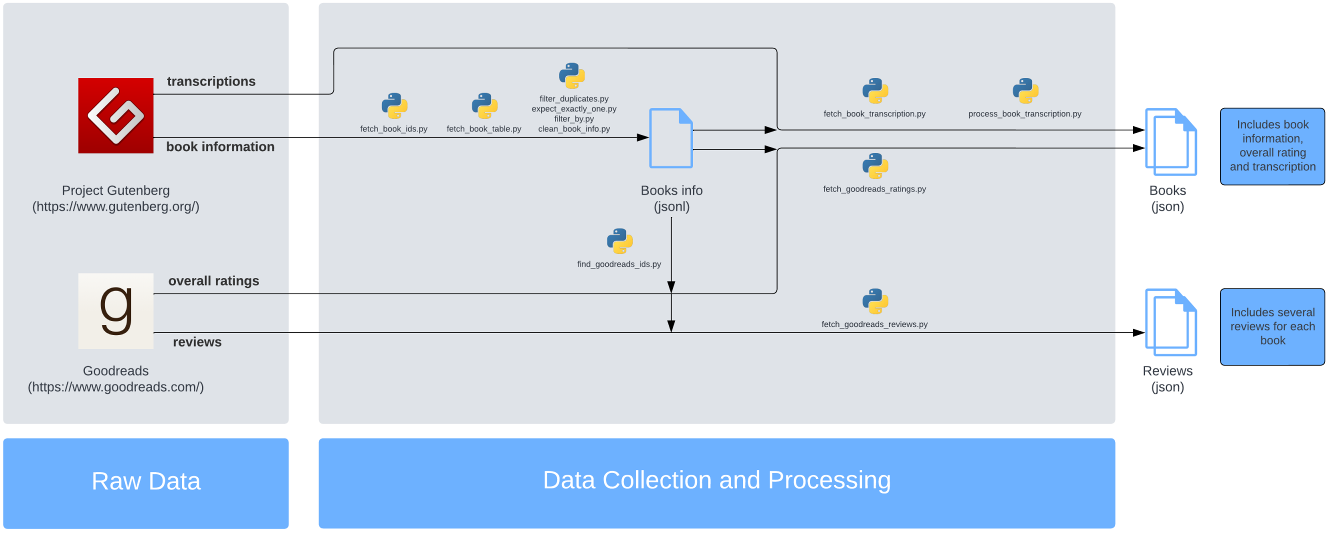
transcriptions
book information
Project Gutenberg
(https://www.gutenberg.org/)
overall ratings
reviews
Goodreads
(https://www.goodreads.com/)
fetch_book_ids.py
fetch_book_table.py
filter_duplicates.py
expect_exactly_one.py
filter_by.py
clean_book_info.py
Books info
(jsonl)
find_goodreads_ids.py
fetch_book_transcription.py
process_book_transcription.py
fetch_goodreads_ratings.py
fetch_goodreads_reviews.py
Books
(json)
Includes book information, overall rating and transcription
Reviews
(json)
Includes several reviews for each book
Raw Data
Data Collection and Processing

Figure 6. Data Scraping, Collection and Processing pipeline