Goodreads Books and Reviews

Information Processing and Retrieval

Inês Silva
FEUP, Porto, Portugal
up201806385@edu.fe.up.pt

Mariana Truta
FEUP, Porto, Portugal
up201806543@edu.fe.up.pt

Rita Peixoto
FEUP, Porto, Portugal
up201806257@edu.fe.up.pt

ABSTRACT

In the current days, we come across big amounts of data and so an increasing concern to index and search efficiently appears. In this paper, an approach to creating an information retrieval system for a book database is expounded. The proposed solution includes a description of the dataset preparation, enrichment, refinement, and exploration process, as well as the detailed exposition of the information retrieval stage, from the indexation of the documents to the evaluation of the resulting system. An overview of the used tools is also included. The final goal is to create a more complex alternative to Goodreads' search system, allowing users to efficiently find the most fitting book for their needs.

KEYWORDS

Goodreads, Books, Reviews, Dataset, Data Preparation, Data Analysis, Information, Retrieval, Processing, Refinement, Search Engine

1 INTRODUCTION

It's a well-known fact that reading is extremely important to maintain a healthy and sharp mind, as well as essential for anyone to continuously develop their literary skills throughout their life. But reading for pleasure has many more benefits, such as increasing empathy, improving relationships and reducing depression symptoms. Unfortunately, everyone has been through the frustrating situation of wanting to read something but spending an enormous amount of time aimlessly looking for the right book without any luck.

A common place for book lovers to connect and help each other find their next favourite book is Goodreads, a social cataloguing website that allows individuals to search its database of books, annotations, quotes, and reviews. Despite the website's popularity, a necessity for a richer search system of their books was identified and consists of the motivation for this work.

This paper starts by describing the exploration performed on the chosen dataset, as well as the steps taken to conduct data preparation and enrichment, using information regarding both book characteristics and user feedback. The database structure is then illustrated and the search tasks to perform in it are specified. Subsequently, the information retrieval stage is carefully discussed, clarifying each action taken towards achieving the designated goal. Lastly, the drawn conclusions are summarized and relevant future work possibilities are mentioned.

2 DATASETS

The project consists of two datasets, the books' dataset and the books reviews' dataset.

2.1 Books

The main dataset chosen contains the general information needed to describe a book, gathered from the Goodreads website. It was retrieved from **Goodreads 100k books**, where the author retrieved the data by scraping the Goodreads website.

It was stored in a CSV file, with 13 columns and 100 000 entries.

This dataset has both numerical data, such as the number of pages, and textual data, such as the book description, genres, etc.

2.1.1 Data Preparation. The initial dataset contained 100 000 books. After extensive analysis, however, it was noticed that the data was not as good as expected, as the original dataset had not been encoded correctly.

To initiate this process, it was first necessary to assess the consistency of the rows and the relevance of each column. With OpenRefine, it was observed that there were several missing values and, after analysing some of these books on Goodreads, it was found that the website did not contain all their information and, therefore, it did not make sense to include them in the dataset. In other words, all the lines with missing values were discarded.

After that, using python scripts with the pandas' library, the data consistency was evaluated. The first task was to explain why there were 3 003 with 0 pages, and since this information was available on Goodreads for some books, a web scraping was performed to replace the missing pages. Nevertheless, there were still 2943 books in this situation. The importance of these situations was discussed, and an agreement was reached that the absence of these values was not that problematic for the final goal, since these books may not have pages either because they are audiobooks or because there is no information on Goodreads.

Afterwards, the book format values were analysed and cleaned, having translated some of them into English and regrouped others to normalize this column. It was noticed that the isbn13 column had poorly standardized values, and therefore it was removed from the dataset since it was not relevant for the context and its preparation would become unnecessary. Posteriorly, it was also found that some rows had repeated genres, which made it necessary to remove the duplicates.

Following this initial preparation, the special characters of the textual fields (description, title and authors) were cleaned. The same process was used in these three columns, which consisted of web scraping the books containing special characters. The books that were still in this situation were eliminated after this processing,

Information Processing and Retrieval, FEUP,
2021.

since these books now included characters from non-European alphabets.

At last, all the extra white spaces that were in the dataset were trimmed so that the final dataset would be as clean and ready as it possibly could be for the next tasks.

Table 1: Number of missing values on each column of the original dataset

author	0	**isbn13**	11435
bookformat	3228	**pages**	7752
desc	6772	**rating**	1562
genre	10467	**reviews**	0
img	3045	**title**	1
isbn	14482	**totalratings**	0

2.1.2 Properties characterization. Throughout the analysis of the dataset was gathered information regarding the mean value, standard deviation, minimum and maximum values for each of its numerical properties (*pages*, *rating*, *reviews* and *total ratings*). These statistics are visible in Table 2.

Table 2: Statistics for the numerical values of the dataset

	pages	rating	reviews	total ratings
mean	276	3.89	182	2991
std	375.35	0.39	1449.45	36353.38
min	1	1.00	0	0
max	70000	5.00	158776	3819326

To characterize and better understand the categorical properties, it was found the most common values for each and analysed how these are related to the numerical properties.

(1) *genre*: there are 1182 different genres in this dataset. It was created a *"Word Cloud"*, shown in fig. 1, to visually represent the existing genres, giving greater prominence to words that appear more frequently. It was also found the 20 most common genres, showing the number of books each appears in, the average number of ratings and reviews, and an average rating of these books in fig. 4
(2) *author*: there are 68767 different authors in this dataset. It was found the 20 authors that have the most books in the dataset, showing the number of books each one wrote, the average number of ratings and reviews, and an average rating of these books in fig. 5
(3) *book format*: there are 203 different book formats in this dataset. It was found the 5 authors that have the most books in the dataset, showing the number of books each one wrote, the average number of ratings and reviews, and an average rating of these books in fig. 6
(4) *title*: since there are 100000 books in this dataset, it was found the top 20 books with the highest number of ratings, showing the number of reviews and the rating of these books in fig. 7

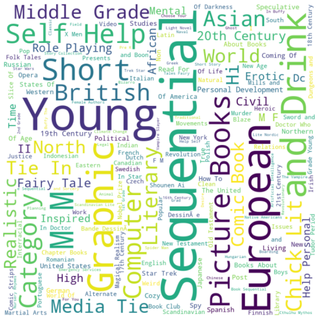


Figure 1: Word Cloud with the genres that occur in the dataset

2.2 Reviews

To enrich the main dataset was used web scraping to collect reviews that are written by Goodreads users to the books since the dataset only has the number of reviews and the numerical rating.

This dataset was stored in a CSV file with 2 columns (URL, review) and 510 000 entries.

2.2.1 Data Preparation. The original scraped dataset contained 510 709 entries, of which 130 were empty. These 130 entries were removed and 819 more were removed because they contained special characters.

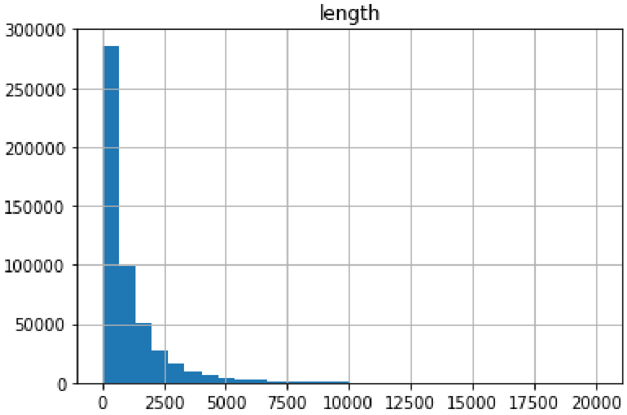
The remaining entries went through a process of data cleaning that removed unnecessary characters like '[]' and '"' that surrounded the reviews. It also removed the escape characters, the single quotes at the start and end of the review.

In the end, the dataset ready to use contained 509 760 entries.

2.2.2 Properties characterization. For the reviews' dataset, it was found relevant to know the average length of a review, and it is 1050 characters. In fig. 2, one can see the distribution of the length of the reviews.

As for most common reviews, it is the word "good" as there are 411 reviews with this value.

Figure 2: Distribution of the length of the reviews



2.3 Data Source

As for the authority of the data source, one can tell the author has already been working on datasets for a few years (has participated in a scholarly competition eight years ago), has at least five datasets and has published some of his work in the last year around this theme. There are also good comments on his discussion.

This dataset was a personal project of his to learn to scrape and was published in a very well-known dataset website, Kaggle and he also gave credits to the website from where it was retrieved and shared the code of the program.

Therefore, it is concluded that it is a good data source.

2.4 Data Quality

To perform the data quality assessment were used five metrics of data quality: completeness, correctness, timeliness, consistency and integrity.

As for completeness, the dataset is 96% complete.

The level of correctness is 98%, as only a few values from the pages are not correct, and the rest is just the format of description, book format and authors that do not comply with the expected format, as they have invalid characters or multiple languages.

In terms of timeliness, one can say that it is up-to-date for the intended use, since it was retrieved 5 months ago and the only differences noticed are the number of reviews and the rating, so it doesn't seem to be problematic for the final goal.

There are some issues regarding the consistency of some properties like the book format that, as said, contains data in different languages.

Finally, in terms of integrity, it is not possible to guarantee that all the counted reviews on the "review" column of the books' dataset are presented in extension on the reviews' dataset. The reviews' dataset was not made to extract all the reviews for each book, but the initial displayed reviews on the book's page that do not all correspond to all reviews of the book, if there is a lot.

3 DATA PROCESSING PIPELINE

In order to achieve a greater quality of the chosen data, various processing tasks were executed. These steps are represented in the data pipeline, shown in fig. 8.

3.1 Data Collection

Using the link to the webpage associated with each book, web scraping was performed. This was done with three purposes: to replace book descriptions that were inserted in the dataset with the wrong encoding and thus had special characters that made the text unreadable, to fill missing values in the *"pages"* column (books with 0 pages), and finally to get up to 10 reviews of each book in order to create a new dataset that allows the exploration of the reviews' textual data.

3.2 Data Cleaning

After the scraping phase, some cleaning tasks were needed to further improve the data quality.

Regarding the book dataset, the book formats were standardized (capitalized first letter, translated to English and aggregated similar values), the *"isbn13"* column was removed due to most values being missing and many inconsistencies in the data, and duplicated values in the *"genre"* column of each book were removed. Finally, special characters were removed from the *"title"* and *"author"* columns.

In the review dataset, the cleaning steps consisted of removing special characters, cleaning up the text and removing null reviews.

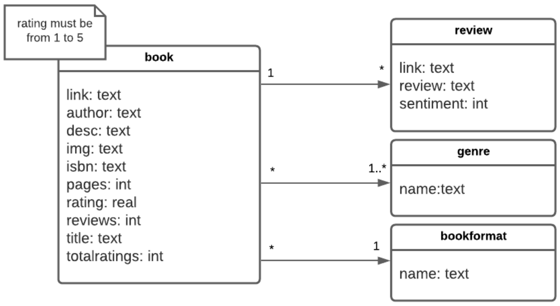
3.3 Data Enrichment

Following the cleanup of the textual data, the review data was enriched by adding a *"sentiment"* feature that indicates if it consists of a positive or negative review. To attain this, a natural language processing task was carried out to perform sentiment analysis in each review using spaCy, an open-source library for Natural Language Processing in Python.

The final step of the developed pipeline consisted of combining the datasets that resulted from the cleaning stage in a SQL database. The conceptual model is described in section 4.

4 CONCEPTUAL MODEL

Figure 3: Domain Conceptual Diagram



As shown in fig.3, the conceptual model consists of four entities:

- book: each book has a Goodreads link, a list of its authors, its description, a representing image, ISBN, the number of pages, rating, the number of reviews, its title, and the number of total ratings;
- review: each review consists of the link of the book it relates to and its text;
- genre: category of book characterized by a particular style, form or content;
- book format: format of the book.

5 SEARCH TASKS

The Goodreads website allows users to find desired books by searching for their title, author or genre. In order to develop a richer search system, the established goal was to make it possible to filter books by their book format, number of pages, rating and keywords that appear in their reviews. Therefore, our search tasks focus both on the book's features and on the user feedback collected from the Goodreads' website. Some of these are:

(1) search for book titles, authors, book formats and genres

(2) filter books by author, genre, book format and popularity
(3) filter reviews by keywords
(4) filter books and reviews by sentiment

6 INFORMATION RETRIEVAL

An information retrieval system deals with the organization, storage, retrieval and evaluation of information from documents. It can be used to retrieve documents that match a particular user's information needs.

In order to develop a complete and efficient information retrieval system, one must primarily define the tool to be used, followed by the indexation of the documents with some custom filters for improving the search, and, lastly, evaluation of the system.

6.1 Tool Selection

There were two main tools recommended for the information retrieval tasks, Solr and Elasticsearch. Both of them were considered:

Solr is an open-source enterprise search platform built on Apache Lucene. It is also a NoSQL datastore and includes features like full-text search, hit highlighting, faceted search, real-time indexing, dynamic clustering, database integration and rich document handling.

Elasticsearch is a distributed search and analytics engine built on Apache Lucene that is also open-source. It is the most popular search engine and provides features like log analytics, full-text search, security intelligence, business analytics, and operational intelligence use cases and is a NoSQL datastore.

Both tools have similar functionalities, but it was opted to use Solr. Even though the existing documentation isn't the best compared to Elasticsearch and has lower scalability, this tool meets better the project necessities. Solr is more text-oriented, while on the other hand, Elasticsearch is more used to parse queries, filter and group. This way, Solr is more adequate for the end goal.

6.2 Documents and Collection

At the end of the data preparation phase, there were two datasets:

(1) books dataset, with about 68700 entries,
(2) reviews dataset, with about 509900 entries

The first approach to creating the documents to be imported into Solr, and the one that seemed most logical, was for the reviews to be a nested document of the corresponding book. As the project progressed, it was realized that it did not make sense to have reviews without the corresponding books and Solr did not allow having weights on the reviews object attribute and still retrieve the parent book, so a different approach was pursued.

To import these into Solr, the reviews' dataset was inserted into the book objects and stored in a JSON file. This means that within each book object are its positive and negative reviews.

All the documents were indexed in a single collection where the information necessities will be queried upon.

6.3 Indexing Process

At the start of the indexing process, all fields were analysed to understand which ones should be indexable. It was then concluded that the link and image attributes should not be indexable since they are unique and are not relevant in an information need.

The schema fields are described according to their type and whether they are indexable in Table 3.

Table 3: Schema's fields, respective types and indexation

Field	Type	Indexed
author	commaText	Yes
bookformat	gramText	Yes
desc	text_general	Yes
genre	commaText	Yes
img	string	No
isbn	string	Yes
link	string	No
pages	pint	Yes
rating	pfloat	Yes
reviews	pint	No
title	gramText	Yes
totalratings	pint	Yes
sentiment	pint	Yes
positive_reviews	text_general	Yes
negative_reviews	text_general	Yes

The schema also indicates that all attributes should be stored and positive_reviews and negative_reviews are multivalued.

The indexed numerical values were defined using the default Solr field type pint for reviews and totalratings and Solr's pfloat for rating.

The textual values with a single instance of each value - img, isbn and link - were defined as a string.

The description, positive_reviews and negative_reviews fields were defined as text_general type, which includes the StandardTokenizer, LowerCaseFilter in both index and query time.

Lastly, even though Solr offers a set of default field types, some custom field types were created for text subjected to an analyser pipeline (as shown in Table 4): commaText for sequences of values separated by commas (genres and authors), gramText (book formats and titles):

commaText This field type applies the ASCIIFoldingFilter filter, which converts alphabetic, numeric and symbolic Unicode characters which are not in the basic Latin Unicode block to their ASCII equivalents. This is used, for example, for the cases of accents in a word, so when a user writes the word without the accent it can still retrieve as if he would write it correctly. It also applies the filter of LowerCaseFilter, which converts any uppercase letters in a token to the equivalent lowercase token, so if a user writes a word without the uppercase it can still be retrieved [9].

gramText This field type also applies the same filter as commaText with an additional filter, EdgeNGramFilter, that generates edge n-grams tokens of size in the range, in this schema, of 2 to 10 [9].

text_general This field type also applies LowerCaseFilter, already described, in combination with the StandardTokenizer, that splits the text field into tokens, treating whitespace and punctuation as delimiters[9] [10].

Table 4: Schema's custom field types

Field Type	Filter and Tokenizer	Index	Query
commaText	ASCIIFoldingFilter	Yes	Yes
	LowerCaseFilter	Yes	No
	PatternTokenizer	Yes	Yes
gramText	ASCIIFoldingFilter	Yes	Yes
	LowerCaseFilter	Yes	Yes
	EdgeNGramFilter	Yes	Yes

6.4 Retrieval

In order to evaluate the different systems' performance, 5 information needs were identified, considering the search tasks described in Section 5.

In this section, each information need is briefly described and presented the query associated with. With the query results given by each system, its performance is evaluated by analysing the top 10 books retrieved regarding their relevance and calculating the corresponding precision and recall.

From all the query parsers available in Solr, the ones explored were: the Standard query parser, the DisMax query parser and the Extended DisMax query parser.

After the exploration stage, the conclusion was that the Extended DisMax was the more suitable query parser, because it has improved proximity, includes advanced stop words handling, allows the specification of the fields the user is allowed to query, disallows the direct search on the fields and supports the specification of fields' weight.

From the available parameters from Extended DisMax, the ones used were:

(1) q - defines the main query that consists of the essence of the search [11].
(2) q.op - defines the default operator (AND, in this case) for query expressions [11].
(3) qf - list of fields, each of which is assigned a boost factor to increase or decrease that particular field's importance in the query [11].
(4) fq - defines a query that can be used to restrict the superset of documents that can be returned, without influencing score [11], it was only used in the first information retrieval to establish that the number of total reviews should be higher than 600.

To evaluate the effect of filters, tokenizers and weighted fields in the query output, three different systems were made:

(1) Schemaless (System 1),
(2) With the schema described in section 6 and default weights (System 2),
(3) With schema described in section 6 and with weighted fields (System 3).

System 1 is meant to represent a basic search system with no further exploration and analysis. System 2 was created to highlight the impact of applying Solr filters and tokenizers in the index and query time. The last system, System 3, aims to show the distinction of indicating some fields as more relevant than the others when querying and can also indicate if the chosen fields were correct or not.

In System 3, an *ad hoc* approach was followed, as the most important attributes for each queried were given a higher weight.

As there are many documents, for the evaluation task not to be too much time-consuming, for the first two information retrievals was used a subset of 200 random documents from the set of documents, to facilitate the manual evaluation.

For the evaluation were considered the first 20 results, being ruled out as relevant or non-relevant. In the following results tables is demonstrated this classification for the first 10 results, where 'Y' means it is relevant, 'N' means not relevant and '-' means there were no more results.

6.4.1 *Cooking book.*

One intends to find a great cooking book to offer their mom, who's vegan and doesn't have a lot of cooking skills

Query: easy and delicious vegan recipes

The weighted fields and the corresponding weight are shown in Table 5.

Table 5: Weights of the fields in cooking book query for the System 3

Field	Weight
genre	1.5
desc	default
positive_reviews	2

Relevance Judgement: The intention was to retrieve books where the reviews mentioned easy and delicious, with the mention of vegan in the gender and/or in the description.

Results: The relevance of each retrieved book, as well as the precision and recall for each system, are shown in Table 6.

Table 6: Cooking book information need's results

Rank	System 1	System 2	System 3
1	Y	Y	Y
2	Y	Y	Y
3	Y	Y	Y
4	Y	Y	Y
5	Y	Y	Y
6	N	Y	Y
7	Y	Y	Y
8	Y	Y	Y
9	Y	Y	Y
10	N	Y	Y
Avg Precision	0.920844	1.0	1.0
P@10	0.8	1.0	1.0

6.4.2 *Interesting books.*

One is looking for an interesting fiction book or a romance, with a good plot that will surely get them hooked on the story.

Query: interesting AND (fiction OR romance)

The weighted fields and the corresponding weight are shown in Table 7.

Table 7: Weights of the fields in the Interesting books' query for the System 3

Field	Weight
genre	1.8
positive_reviews	default

Relevance Judgement: The intention was to retrieve books whose positive reviews mentioned the fact that the book was interesting and the gender had either fiction or romance.

Results: The relevance of each retrieved book, as well as the precision and recall for each system, are shown in Table 8.

Table 8: Interesting fiction/romance book information need's results

Rank	System 1	System 2	System 3
1	N	Y	Y
2	N	Y	Y
3	Y	Y	Y
4	N	Y	Y
5	-	Y	N
6	-	N	Y
7	-	Y	N
8	-	N	Y
9	-	N	Y
10	-	N	N
Avg Precision	0.333333	0.807847	0.838675
P@10	0.1	0.6	0.7

6.4.3 *Family book.*

One is feeling lonely and nostalgic about their childhood household and is looking for a book that talks about family.

Query: family

The weighted fields and the corresponding weight are shown in Table 9.

Table 9: Weights of the fields in the Family book query for the System 3

Field	Weight
genre	2.5
title	default
desc	1.5

Relevance Judgement: The intention was to retrieve books about families, giving priority to the book who have in the gender and/or in the description family and that actually involves families.

Results: The relevance of each retrieved book, as well as the precision and recall for each system, are shown in Table 10.

Table 10: Family book information need results

Rank	System 1	System 2	System 3
1	Y	N	Y
2	N	N	Y
3	-	N	N
4	-	N	N
5	-	Y	N
6	-	Y	Y
7	-	Y	Y
8	-	Y	N
9	-	Y	Y
10	-	N	Y
Average Precision	1.0	0.464498	0.625451
P@10	0.1	0.5	0.5

6.4.4 *Clichés of adult romances.*

After a break-up, one feels the need to believe in love again, so books with good clichés of adult romance are always a comfort in the cold days of the holiday season.

Query: good cliche adult romance

The weighted fields and the corresponding weight are shown in Table 11.

Table 11: Weights of the fields in the Clichés of adult romances query for the System 3

Field	Weight
genre	0.8
negative_reviews	0.01
positive_reviews	1.2

Relevance Judgement: The intention was to retrieve books whose gender has romance, and that the positive reviews said it was a good cliché.

Results: The relevance of each retrieved book, as well as the precision and recall for each system, are shown in Table 12.

Table 12: Good clichés of adult romances information need results

Rank	System 1	System 2	System 3
1	N	N	Y
2	Y	N	Y
3	N	Y	Y
4	N	Y	Y
5	N	N	Y
6	N	N	Y
7	N	Y	Y
8	N	Y	Y
9	N	Y	Y
10	N	N	N
Avg Precision	0.252451	0.494054	0.934083
P@10	0.1	0.5	0.8

6.4.5 *Nazi Holocaust history.*

As a typical history lover dad, my dad is reading a book about the Nazi holocaust. I don't know the book's name, I only remember its cover, and I want to tell my sister about it.

Query: nazi holocaust history

The weighted fields and the corresponding weight are shown in Table 13.

Table 13: Weights of the fields in the Nazi Holocaust query for the System 3

Field	Weight
genre	1.8
title	default
desc	1.2

Relevance Judgement: The intention was to retrieve books whose gender contained history and holocaust and whose description and/or title refer to the word Nazi.

Results: The relevance of each retrieved book, as well as the precision and recall for each system, are shown in Table 14.

Table 14: Nazi holocaust history information need results

Rank	System 1	System 2	System 3
1	Y	Y	Y
2	N	Y	Y
3	N	N	Y
4	N	Y	Y
5	Y	Y	N
6	N	N	Y
7	N	Y	Y
8	N	N	N
9	Y	N	N
10	N	N	N
Avg Precision	0.491667	0.699391	0.813046
P@10	0.3	0.5	0.6

6.5 Evaluation

The metrics used to evaluate the results were:

- **Precision** expresses the fraction of relevant documents from the retrieved documents;
- **Recall** expresses the fraction of retrieved documents from the existing relevant documents.
- **Average Precision (AvP)** provides a measure of quality across recall levels for a single query.
- **P@10** expresses the precision in the first 10 results.
- **Mean Average Precision (MAP)** is the average of AvP and helps to better understand the quality of the system.

The precision, average precision and P@10 metrics can be seen in the tables of each information retrieval that can be found in Section 6.4.

In addition, the corresponding Precision-Recall graphs of each information retrieval are shown in figures 9, 10, 11, 12 and 13.

The figure 9 reflects the fact the weights have a small influence on this information retrieval as the field was already the main part of the query, so no difference is spotted between System 2 and System 3. As for System 1, since no filters or tokenizers are applied to it, it has the worst results because it can't reach as well the fields in the reviews.

The figure 10 shows that System 1 has poor performance compared to the other systems, as it only retrieves four results (one relevant). For the same reasons as the previous information retrieval, there is not a big gap between the System 2 and System 3 results.

In Figure 11, although the P@10 is the same for both Systems 2 and 3, it is visible in the precision-recall curve that the weights allow System 3 to give a more accurate relevance to the results, while System 2 attributes the correct ones a lower ranking. On the other hand, System 1 could only retrieve 2 books, which results in a line that is misleadingly similar to the other systems, despite its poor performance.

Figure 12 shows a similar situation to the one described before, where Systems 1 and 2 give the relevant books a lower ranking than System 3, illustrating once again how the weights are beneficial to the system. It is also worth mentioning the increase in precision when comparing Systems 1 and 2.

In Figure 13, all systems behave as desired, giving a higher rank to the correct results. In this case, the most notable conclusion to draw from the plot is the improvement in precision from System 1 to 2, and from System 2 to 3. This corroborates the previously discussed weight choices, as well as the benefits of a schema.

The results of the Mean Average Precision for each search system are shown in Table 15.

Table 15: Mean Average Precision for each system

System	MAP
System 1	0.599659
System 2	0.6773268
System 3	0.842251

As expected, the mean average precision is higher in System 3, which is the system that combines the schema with the field weights. The large difference in the MAP between the system without schema and the system with schema and field weights is clear. Regarding systems 2 and 3, there is a considerable improvement, thus proving the importance of the weights in the fields. Nonetheless, it is possible to, in the future, improves the weighting system and obtain better results, whilst the schema options were well analysed and are used the ones that better suit the needs of this search system.

7 CONCLUSION

In this paper, the developed information processing and retrieval process is meticulously explained, from the data gathering, cleaning and preparation phase to the assessment of the created retrieval system's quality.

Throughout this work, the datasets were well analysed and studied in order to conclude the appropriate data cleaning and preparation tasks to perform in order to prepare the information for the intended search tasks.

In a second stage of the process was developed an indexing process, sustained in the exploration and analysis of different Filters and Tokenizers provided by Solr. Several purposeful information needs were carefully conceived and used to evaluate and compare the developed retrieval systems.

The results obtained prove the initial belief that the weights and schema would have a big impact on the quality of the search system. It points out that the combination of schema with weighted fields brings better results, but still has room for improvement, as the mean average precision is still approximately 84%.

As a last note, it is pointed that throughout this experience with Solr was concluded that it is not an easy tool to work with, since it presents some cons as it has poor documentation, is not user-friendly and intuitive and has limitations regarding the attribution of weights to nested documents. This lead to a slightly slower learning curve than expected and a lot of back and forward during this phase.

8 FUTURE WORK

Having in consideration the entirety of this work, it is considered relevant to undertake the following steps to enhance the developed retrieval system:

(1) improve sentiment analysis, since it was found that the data used to train the model was not indicated for the dataset and therefore some of these analyses are incorrect. For example, if a review talks about how sad a book was but still is a good review, it is classified as negative;
(2) improve the quality of the search results, in order to provide an overall better search system to the end-users, as it is the main goal of the project;
(3) create the final version of the search system, that allows more complex queries in a user-friendly (easy queries and little time) mode.

REFERENCES

[1] GoodReads 100k books. (2021). Retrieved 14 November 2021, from https://www.kaggle.com/mdhamani/goodreads-books-100k

[2] Documentation · OpenRefine. (2021). Retrieved 14 November 2021, from https://openrefine.org/documentation.html

[3] Goodreads. (2021). Retrieved 14 November 2021, from https://www.goodreads.com/

[4] Goodreads Wikipedia - Wikipedia. En.wikipedia.org. (2021). Retrieved 13 December 2021, from https://en.wikipedia.org/wiki/Goodreads.

[5] pandas documentation — pandas 1.3.4 documentation. (2021). Retrieved 14 November 2021, from https://pandas.pydata.org/docs/

[6] matplotlib.pyplot — Matplotlib 3.5.0 documentation. (2021). Retrieved 14 November 2021, from https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html

[7] Apache Solr - Wikipedia. (2021). Retrieved 4 December 2021, from https://en.wikipedia.org/wiki/Apache_Solr

[8] Solr vs Elasticsearch: Performance Differences & More [2021] - Sematext. (2021). Retrieved 4 December 2021, from https://sematext.com/blog/solr-vs-elasticsearch-differences/

[9] Filter Descriptions | Apache Solr Reference Guide 8.11. (2021). Retrieved 4 December 2021, from https://solr.apache.org/guide/8_11/filter-descriptions.html

[10] Tokenizers | Apache Solr Reference Guide 8.11. Solr.apache.org. (2021). Retrieved 12 December 2021, from https://solr.apache.org/guide/8_11/tokenizers.html.

[11] The DisMax Query Parser | Apache Solr Reference Guide 8.11. Solr.apache.org. (2021). Retrieved 12 December 2021, from https://solr.apache.org/guide/8_11/the-dismax-query-parser.html.

[12] The Standard Query Parser | Apache Solr Reference Guide 8.11. Solr.apache.org. (2021). Retrieved 12 December 2021, from https://solr.apache.org/guide/8_11/the-standard-query-parser.html.

[13] The Extended DisMax (eDismax) Query Parser | Apache Solr Reference Guide 8.11. Solr.apache.org. (2021). Retrieved 12 December 2021, from https://solr.apache.org/guide/8_11/the-extended-dismax-query-parser.html.

[14] Sentiment Analysis with Spacy and Scikit-Learn. Engineering Education (EngEd) Program | Section. (2021). Retrieved 13 December 2021, from https://www.section.io/engineering-education/sentiment-analysis-with-spacy-and-scikit-learn/.

9 APPENDIX

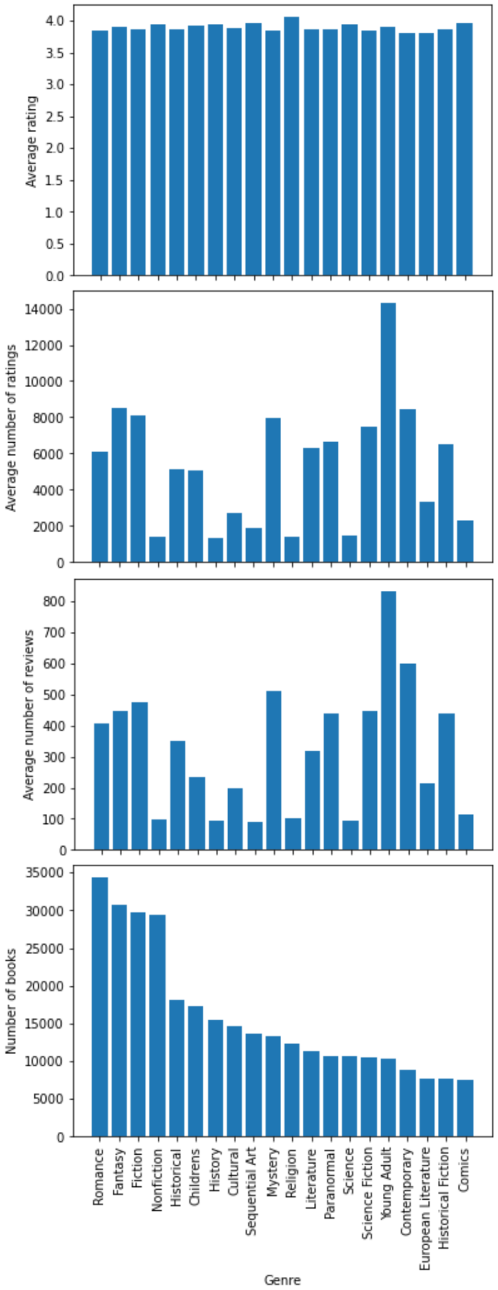


Figure 4: Average number of reviews and ratings, and average rating of the books that belong in the top 20 genres

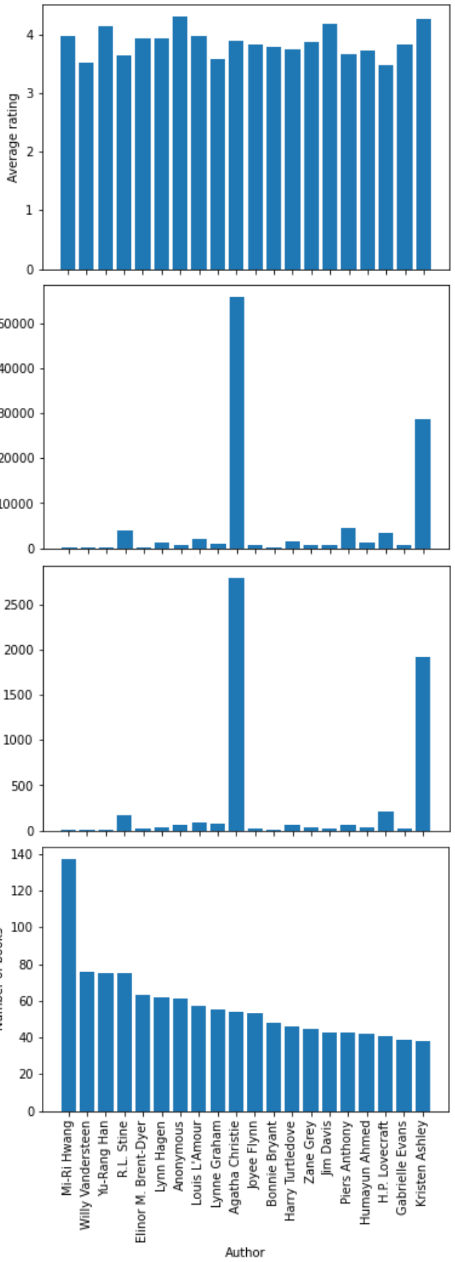


Figure 5: Average number of reviews and ratings, and average rating of the books written by the top 20 authors

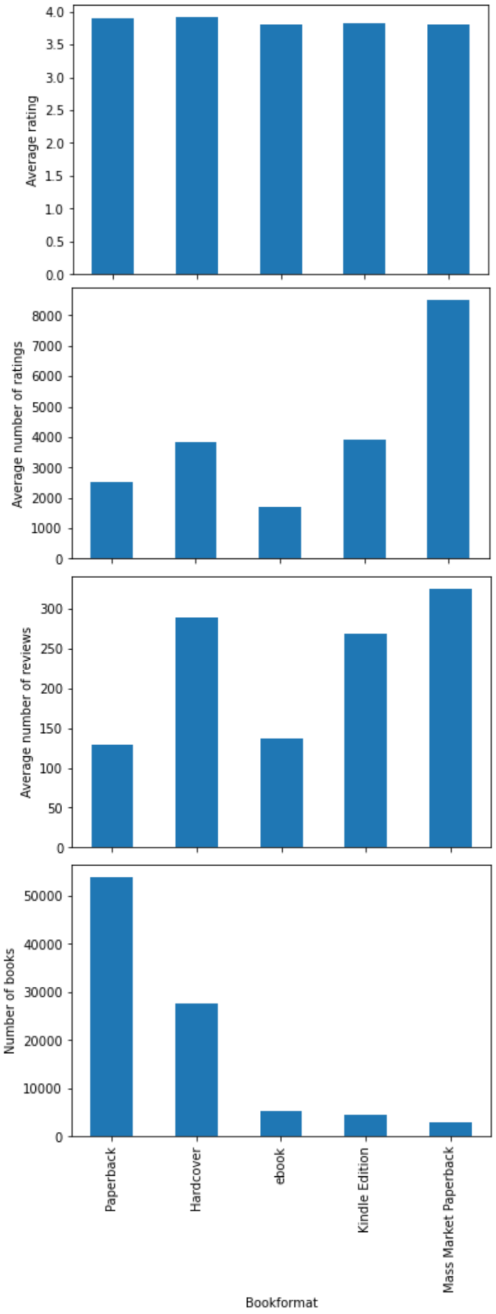


Figure 6: Average number of reviews and ratings, and average rating of the books that belong in the top 5 book formats

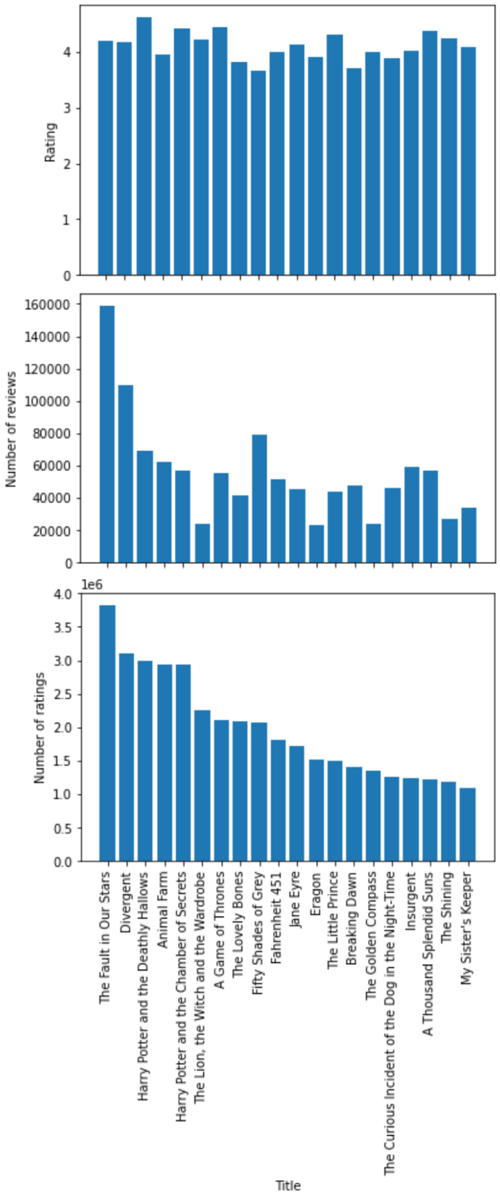


Figure 7: Average number of reviews and average rating of the top 20 books with most ratings

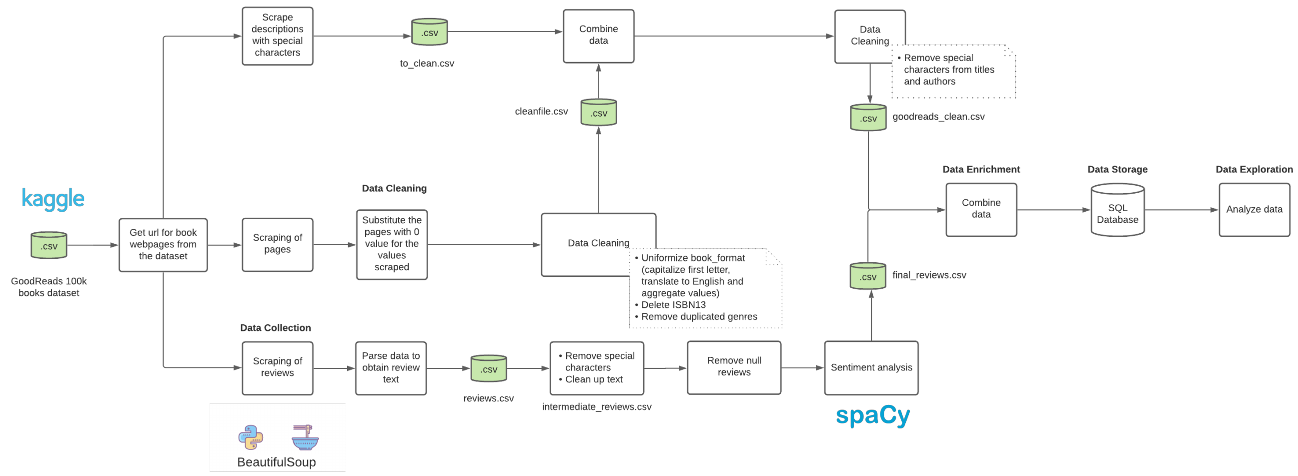
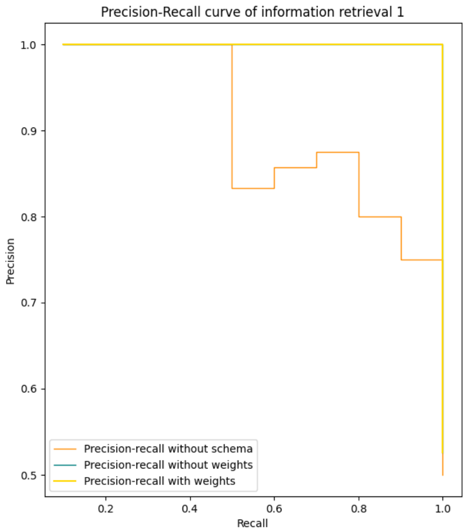

kaggle
.csv
GoodReads 100k books dataset
Get url for book webpages from the dataset
Scrape descriptions with special characters
.csv
to_clean.csv
Combine data
Data Cleaning
• Remove special characters from titles and authors
.csv
goodreads_clean.csv
cleanfile.csv
.csv
Data Cleaning
Scraping of pages
Substitute the pages with 0 value for the values scraped
Data Cleaning
• Uniformize book_format (capitalize first letter, translate to English and aggregate values)
• Delete ISBN13
• Remove duplicated genres
Data Enrichment
Combine data
Data Storage
SQL Database
Data Exploration
Analyze data
.csv
final_reviews.csv
Data Collection
Scraping of reviews
Parse data to obtain review text
.csv
reviews.csv
• Remove special characters
• Clean up text
intermediate_reviews.csv
Remove null reviews
Sentiment analysis
spaCy
BeautifulSoup

Figure 8: Data processing pipeline



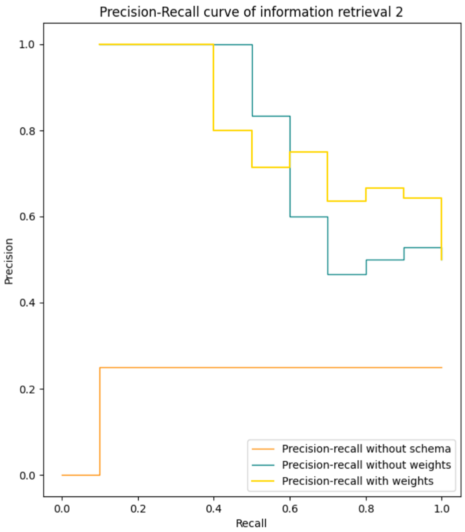


Figure 9: Precision-Recall graph for information retrieval 1

Figure 10: Precision-Recall graph for information retrieval 2

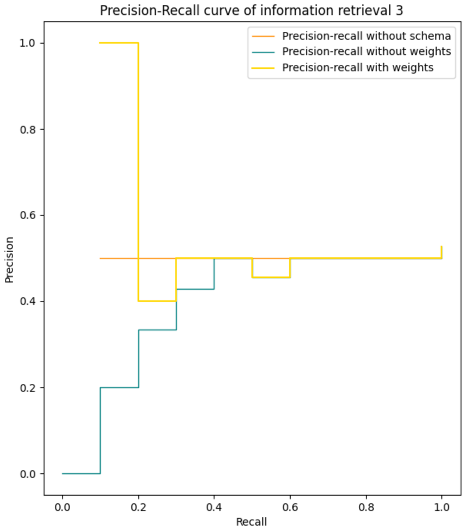


Figure 11: Precision-Recall graph for information retrieval 3

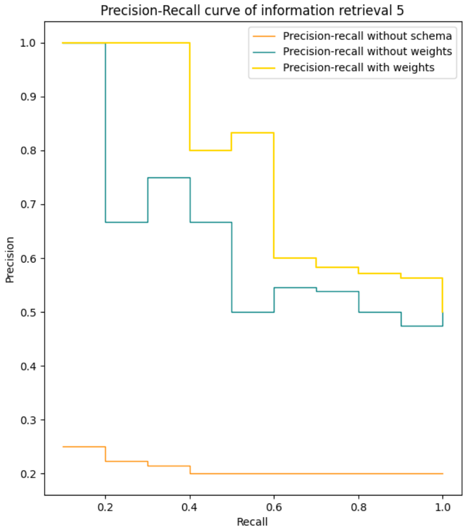


Figure 13: Precision-Recall graph for information retrieval 5

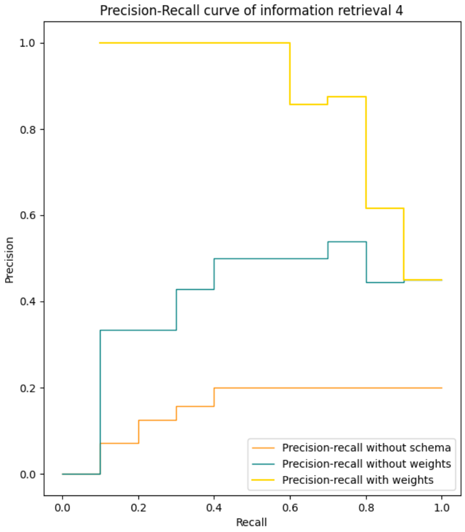


Figure 12: Precision-Recall graph for information retrieval 4