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 [15]. 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.

Information Processing and Retrieval, FEUP,
2022.

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, 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.

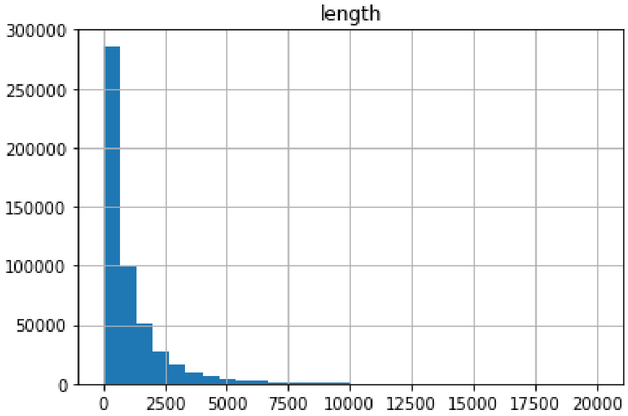


Figure 2: Distribution of the length of the reviews

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

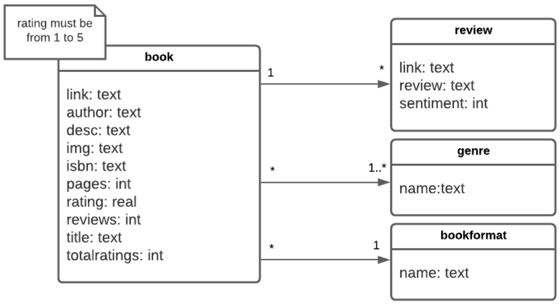


Figure 3: Domain Conceptual Diagram

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 needs 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. These weights were chosen because we wanted to focus the search on these three fields: genre, desc and positive reviews, however, the description should be the last resource to achieve this textual search and we wanted to overpower positive reviews in regards to gender but not with much distinction. After a few tries, these values were the ones that produced better results.

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 genre 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	0.95	1.0	1.0
P@10	0.80	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. The weights applied were chosen because the main focus of this information need was the genre and that would be interesting, but the genre was the most relevant part of the information need and this was the weight that produced more accurate results.

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 genre 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	0.33	0.98	0.91
P@10	0.10	0.60	0.70

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. As it asks for a family book, the fields chosen to add weight were the ones that could mention it. However, in this context, having family in the genre is more relevant than in the description and in the description more relevant than in the title. These were the weights that, after a few tries, resulted in the best results.

These weights were chosen because we wanted to focus the search on these three fields: genre, desc and positive reviews, however, the description should be the last resource to achieve this textual search and we wanted to overpower positive reviews in regards to genre but not with much distinction. After a few tries, these values were the ones that produced better results.

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 genre 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	0.40	0.70
P@10	0.10	0.50	0.60

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. The weights chosen represent the goal of this information need: the positive reviews are more important than the negatives, so the difference between these two fields should be notorious. Also, the genre is a major part of the information need, and so should be weighted, but in a less relevant way than the positive reviews.

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 genre 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	0.50	0.46	1.0
P@10	0.10	0.50	0.90

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. In this information need, both the genre and the Nazi part are major in the relevance theme, however as it mentions the "history lover dad" and "Nazi holocaust" and both can be gathered in history like genre. Therefore, the genre has slightly more weight than the description to show this relevance, and the description is slightly more weight than the title because is more likely to contain the full information need requisites.

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 genre 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.

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.

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	0.58	0.85	0.95
P@10	0.30	0.50	0.60

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

In addition, the corresponding Precision-Recall graphs of each information need 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 need 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 need, 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.

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,

Table 15: Mean Average Precision for each system

System	MAP
System 1	0.672
System 2	0.738
System 3	0.912

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 SEARCH SYSTEM IMPROVEMENT

In this stage of the project, the goal was to improve the search system described in Section 6. To do so, the following themes were explored:

(1) data improvement;
(2) information retrieval improvement;
(3) user interface.

7.1 Data Improvement

In this theme, was analysed the current state of the data given to the search system and were spotted the necessary improvements and the problems to solve.

The data was already prepared in previous stages but some additional details came up when performing the search through it, such as multiple languages in textual fields and wrong sentiment analysis evaluation.

When experimenting with the search system with a simple user interface, the necessity of knowing more about the authors was found.

7.1.1 Authors' Dataset. Understanding the need of identifying an author as an entity, it was decided to gather data about the authors. Web scraping was performed once again on the Goodreads website with the purpose of obtaining each author's photo and description.

This data enrichment involved, as well, a data cleaning stage removing all special characters and inserting an empty string in NaN entries. This dataset was added to the two previous datasets (books dataset and reviews dataset). The previous data inserted the reviews inside the book objects. This time, to that object, was added a type field with the value "book" and another type of object was added "author", the object that contains the author name, link to the image and its description and additionally the field "type", to distinguish these two types of objects.

These new fields were added with the types and indexation shown in Table 16.

Table 16: Author's additional fields, respective types and indexation

Field	Type	Indexed
author_name	text_general	Yes
author_image	string	No
author_description	text_general	Yes
type	string	No

7.1.2 Translations.
It was observed that the dataset suffered from a reasonable inconsistency in textual fields, having reviews, book descriptions and information about authors in several languages. To standardize the data and enhance the readability and accessibility of the system, all text was translated to English. Moreover, during the language identification and translation process, a few occurrences of "bad" values (wrongly encoded, incomprehensibly written, etc...) were identified and removed, further cleaning the data and therefore making it more valuable. As the last step, the sentiment analysis pipelines were repeated, taking advantage of this new improved data. Since the training data was fully in English, the instances that were written in other languages were almost always incorrectly classified. As a matter of course, more accurate results were obtained after the described translations.

7.2 Information Retrieval Improvement

After no further improvement is needed in the data, the next theme would be to directly improve our search system. The tool being used, Solr, provides some search components to improve the search system: synonyms, suggester, spell checker, learning to rank, more like this, result grouping, faceting, etc.

From these ones mentioned, documentation was analysed to better understand them:

- Synonyms: can be used to configure synonyms for use during both indexing and querying of textual data;
- Suggester: can be used to implement an auto-suggest feature in the application;
- Spell-checker: gives user inline query suggestion based on other similar terms;
- Learning to rank: allows running trained learning models on top of the results returned by the queries;
- More like this: enables the user to query for documents similar to a document in their result list;
- Result Grouping: groups documents with a common field value into groups and returns the top documents for each group;
- Faceting: arrangement of search results into categories based on indexed terms.

Result grouping would produce the worst relevance results, for the fact that it would put less relevant documents first to aggregate them to other relevant documents with similar fields. More like this is not very interesting regarding this context, because it is hard to create the concept of more like this in the book concept; usually is not related to the fields expressed but the writing and history description itself. Lastly, learning to rank would imply implicit or pseudo-relevant feedback with subjective evaluation, which could most likely decrease the accuracy of the search system as it would work for some cases only.

Hence, the ones that better suited the needs of this search system, and the ones that were further explored in this stage, were synonyms, suggester, spell checker and faceting.

7.2.1 Synonyms. Throughout the realization of the search system, it became clear that there was a need to explore Solr's ability to handle synonyms. In this way, a file was created with several entries with synonyms that in the context of the book world makes sense, such as similar emotions and characteristics or even names that represent the same thing. Consider the following situation which will be presented in the next section: if the user wanted to find technical books about wine production and would therefore type something like "winery", the results would not be quite as expected. Thus, a synonym "winery, vineyard" was added to the list so that the results would be within the wine production theme and not disperse into novels, for example.

7.2.2 Suggester and spell checking. These two search components were applied together to assist the user search with data existing in the dataset. A spell-checker was implemented to give query suggestions based on similar terms.

The suggester implemented is quite powerful in the way that it provides suggestions at the beginning of the field content, taking advantage of a fuzzy search on top of the analysis chain provided with the field. The suggestions consider synonyms, stop words, stemming and any other token filter used in the analysis and support also misspelt terms by the user (spell checker)[16].

The query is analysed, the tokens produced are then expanded producing for each token all the variations accordingly to the max edit configured for the String distance function configured (default is Levestein Distance). The suggestions are identified starting at the beginning of the field content and include the entire content of the field[16].

It was also specified the parameter "exact math first" as true, to facilitate the recognition.

Note that this suggester is only applied to the title field and for this, it was necessary to consider whether the current type of this field was adequate. For a more enriched search, the title would have to continue with the type created and explained above, gramText. However, to have better suggestions, it would be necessary for the title to be of type text_general since the tokenizers used would facilitate the suggestions besides these being more precise and relevant. Therefore, a mechanism provided by Solr was used to make copies of fields so that several different field types can be applied to a single piece of incoming information, copyField.

7.2.3 Facets. These were used to retrieve two of the fields filters of the search system: genre and book format. The facet aggregates the fields in alphabetic order and limits the faceting to 1200 different facets.

7.3 User Experience Improvement

This final theme relies on improving the user experience in the developed search system.

For that, was developed a frontend (user interface) for the search system. Some of the implemented features were:

(1) free text search bar with implemented suggestions and spell checking, with a maximum of 15 suggestions and using the title as dictionary;
(2) filters for the fields: genres, book format, number of pages and rating. There is a filter box where the user can apply multiple filters to better find the results that the user is looking for;
(3) facets on the filters of book format and genres. On page reload a request to the Solr is made to retrieve these facets. This way, as they are grouped alphabetically it facilitates the user readability and also it always to choose multiple values knowing that all the values are valid, which did not happen before, since the user could write free text with the intention of being categorized as a genre or a book format but it did not exist in our dataset, so the information need would not be met;
(4) pagination, to improve user experience, the results are presented using pagination;
(5) applying weights to the fields the user wants to focus the search on. In the frontend, there are multiple checkboxes with the fields, that divide a total value of weight 10 equally for the checkbox selected, in order to prioritize results that match the query in those fields. If no checkbox is selected, all the fields have default weight;
(6) appealing design. We believe the design is a very important part of the user experience as motivates the user to use our search system and should facilitate its use. Also, the design must be thought for the user and its needs and to be the most straight-forward possible to reduce the entropy;
(7) information displayed in info box form, containing the main field of the document, to provide the user with the main information about the documents it is searching upon.

Although there was the possibility to directly make requests to Solr on the frontend, one would have to change the Solr configuration to allow Cross-Origin. Such a solution did not seem the most elegant, so it was created a node.js backend responsible for connecting the website to Solr. This approach turned out to be the right one since, by having this intermediary, the frontend does not have to address certain connection problems, only receiving clean and processed data.

Regarding the backend of the search system, axios was used which is a promise-based HTTP Client and makes it possible to write async/await code to perform XHR requests very easily. Initially, to simplify the Solr requests, it was created a new axios instance where the desired configuration was defined (Solr URL and a timeout). Using this instance, two types of requests were made to Solr, using two request handlers:

/select the default handler was used to receive the user's search results as well as make simple requests for a book or author using its id;

/suggest this request handler was created and added to *solrconfig.xml* in order to configure the default parameters of the suggester and incorporate the defined suggest search component. This request allows giving suggestions to the user according to the text input.

8 COMPARISON OF SYSTEMS

In the first system developed, the weighted fields and the weight value were costumed to each information need. Consequently, this system naturally resulted in more relevant documents being easily retrieved.

The improved search system works as global and because of that the weights can no longer be costumed and should be global. This way, it is expected to obtain the worst results.

In this chapter, one can find the comparison between:

(1) the initial search system (System 1);
(2) the improved search system using Solr interface (System 2);
(3) the improved search system using the developed user interface (System 3).

In addition to the metrics calculated in section 6.4, it was used a new metric, the Discounted Cumulative Gain (DCG), which measures the quality of the ranking and is often used to measure the effectiveness of the search system. Each document is classified on a scale from 0 to 3 regarding its relevance, where 0 means not relevant, 3 highly relevant and 1 and 2 relatively relevant. As DCG alone cannot be used to consistently compare search engines, the cumulative gain should be normalized across all queries, so the actual metric used in the results was normalized discounted cumulative gain.

8.1 Results

To demonstrate the improvements or its absence in the search system, it was decided to write three information needs that demonstrated the three problems addressed: the first one would show the value of the new dataset of authors; the second one would prove the improvement in the relevance of results due to the addition of synonyms; and the last one would compare the previous system with the current one by changing only the weights and their fields. To compare the search systems, similarly to the retrieval section of this report (section 6.4), the information needs were identified and will be briefly described along with the associated query, weights and result relevance analysis.

As mentioned before, system 2 must have global weights that remain the same for all information needs. As usually, the users are more interested in the books or authors themselves, the weights of each field are shown in Table 17.

Table 17: Weights of the fields of the System 2

Field	Weight
title, desc, author, author_name	1.7
author_description	1.3
positive_reviews, negatives_reviews	0.8
remaining fields	default

8.1.1 *New York Times bestseller.*

One wishes to find New York Times bestsellers since it is a well-known newspaper for its critiques.

Query: New York Times bestsellers

In System 1, since the description is most likely the only place that can contain this mention, it was given the weight of 1.2. In System 3, to obtain these results, it was chosen the authors option, since having bestseller authors will most likely lead to their best seller books, as can be seen in Figure 17.

Relevance Judgement: It is considered relevant if it is a best-seller book or bestseller author and it is possible to verify it according to the retrieved document.

Results: The relevance of each retrieved book, as well as the precision and recall for each system, are shown in Table 18.

Table 18: New York Times bestsellers information need's results

Rank	System 1	System 2	System 3
1	2	0	3
2	0	3	2
3	1	3	3
4	3	3	3
5	2	2	2
6	0	0	2
7	0	1	2
8	1	0	3
9	0	2	2
10	3	2	3
Avg Precision	0.74	0.69	1.00
P@10	0.60	0.70	1.00
nDCG	0.77	0.77	0.97

In this information need it is noticeable that there is a decrease in the average precision from System 1 to System 2, which can be justified by the use of global weights instead of custom and was already expected. However, from System 1 and 2 to System 3 there is a remarking improvement because the interface has the option to choose only the authors and it is easier for the system to find bestselling authors because it is a usual reference in their description.

8.1.2 *Wine.*

One recently got a sudden interest in wines, therefore it looking for books that will teach him more about wines and wineries.

Query: winery

In System 1, since the description is the best place to find the wanted query it was given a weight of 1.2 and the title is the only other place where it can be mentioned so it has default weight (1). In System 3, to obtain these results, it has chosen the fields title and description to prioritize, so they both got a weight of 5 and selected only books and not books and authors as can be seen in Figure 18.

Relevance Judgement: It is considered relevant if it is a technical book about wineries or vineyards.

Results: The relevance of each retrieved book, as well as the precision and recall for each system, are shown in Table 19.

Table 19: Winery information need's results

Rank	System 1	System 2	System 3
1	3	3	3
2	3	2	2
3	0	3	3
4	0	0	0
5	0	1	1
6	0	2	2
7	2	0	3
8	3	3	2
9	0	0	0
10	0	0	0
Avg Precision	0.73	0.90	0.91
P@10	0.40	0.60	0.70
nDCG	0.90	0.93	0.94

In this information need, it is understandable that the improved system has better performance, as synonyms were added for the query word and better results can be retrieved. The minor difference between the performance of System 2 and System 3 can be justified with the prioritize feature that gives more weight to the fields it specifies which are more adequate to the search.

8.1.3 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

In System 1, the genre was given a weight of 1.8, the description 1.2 and the title default, as was explained in section 6.4.5. In System 3, was selected the title and description for being prioritized, so to them, both were attributed a weight of 5 and chosen in the filter genre the "holocaust" as can be seen in Figure 19.

Relevance Judgement: The intention was to retrieve books whose genre 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 20.

Table 20: Nazi holocaust information need's results

Rank	System 1	System 2	System 3
1	3	3	3
2	3	3	3
3	2	0	1
4	3	1	2
5	0	3	3
6	2	0	2
7	1	2	3
8	0	0	2
9	0	0	1
10	0	0	0
Avg Precision	0.95	0.85	1.00
P@10	0.60	0.50	0.90
nDCG	0.98	0.94	0.96

The results mirror the initial perception of using global weights would decrease the performance of the search system (difference between System 1 and System 2) and the aids applied in the user interface compensate for the previous loss.

8.2 Comparison Conclusions

When the improvement of the search started, one could easily understand that hardly there would be improvements when comparing with the previously developed search system, except in very specific cases where the improvement was easily demonstrated. This happens because of a few different reasons:

- The first system design already had an optimal approach was the focus was to produce as relevant results as possible, where costumed weights were used according to each query context, so it is easy to understand that if now, in the improved search system, global weights are used, the fields might not have the same priority in the results and therefore may no longer aid as much as in the initial system, producing worst results overall;
- In the context of the designed search system, there were not many improvements possible regarding the way the system retrieved documents, besides adding synonyms, what was possible was to improve the type of data in the request so it better suited the user needs. This way, more work was focused on reducing the user mistakes when specifying a query for the system. As mentioned before, a suggester and a spell-checker help the user to understand that they might be a misspelling, also reducing the chances of specifying certain fields the wrong way using filters that use facets from genre and book format and ranges for the numerical values allowing wider and more specific ranges. Even though these improvements were applied, they do not positively impact directly the results obtained, they improve the query given to the system and not the system itself;
- Implementations like ranking using algorithms or more like this are not adequate to the context as when a user searches for a book it most likely is looking for something specific rather than what is more popular or more queried by other users. There isn't an evaluation function that can easily be described to suit the requirements of this search system and it would probably tailor the search system too much to a usual user having poor performance on less active users and less common queries;
- Lastly, in the system itself there was an improvement of data quality, that is not measured by the mentioned metrics and has significant importance in the value/performance of a search system. With an authors dataset, it is easier to group information about an author a find books of the same author. With the translations, as the universal language (English) is better understood by the majority of the population, the data is more usable and it also affected the wrongfully classified reviews because of the language difference.

The results of the Mean Average Precision for each search system are shown in Table 21.

The Mean Average Precision cannot be taken as a global evaluation since it is highly dependent on the choice of information needs

Table 21: Mean Average Precision for each system

System	MAP
System 1	0.807
System 2	0.813
System 3	0.970

that as mentioned before were chosen very carefully to demonstrate the improvements. In a larger concept, with more global information needs, the results would not be the same.

Having calculated the normalized discounted cumulative gain for each system in each information need results, these values can be averaged to obtain the mean normalized discounted cumulative gain performance of each search system. These results are shown in Table 22.

Table 22: Mean Normalized Discount Cumulative Gain for each system

System	MNDCG
System 1	0.882
System 2	0.880
System 3	0.957

Unlike pure classification use cases where the system is right or wrong, in a ranking problem, the system is more or less right or wrong [18]. Observing the results, there is a small decrease in the average of normalized discounted cumulative gain from system 1 to system 2, which was expected since the fields and their weights in the first system were customized. Moreover, a clear improvement can be noticed from the first two systems to the third one which is explained by the fact that relevance is subjective and since system 3 depends on user input and choice of filters and search fields, this search will have more relevant results to the user.

Concluding, although the results themselves have not improved much, the fact that the user has control over the search fields as well as being able to select the filters that best suit their case, it is much easier to customize their search which consequently leads to more relevant results.

9 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 note of this stage, 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.

Improving the search system was interpreted more like aiding the user to produce better queries as it was the most meaningful action to take in the context of the search system goals. The results represent that it was a good path to follow since the measure of effectiveness (mean normalized discount cumulative gain and mean average precision) have significantly improved when compared to the previous system and to the one using Solr interface.

Creating a ranking algorithm that would have an optimal performance in the overall use of the search system would be a very complex task with a considerably big trial and error approach. Therefore, it was not considered in this project as it would fall out of its scope.

As a final note, the team is proud of the work developed in this project, having retained a lot of knowledge about search systems and how they are optimized which is not recognizable when used as the end-user.

10 FUTURE WORK

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

(1) improve the spell checker accuracy since it was found that Solr's default spell checker is quite limited and does not perform ideally in many cases;
(2) integrates artificial intelligence algorithms to analyze user relevance feedback, taking careful advantage of information like clicks or frequent searches to evaluate query results and continuously improve the system's quality;
(3) understands and utilizes each user's personal preferences to retrieve more relevant information and generate pertinent suggestions, tailoring a personalized experience for each user.

11 REVISIONS INTRODUCED

In this final version of the report, a few revisions were made and thus the changes that were made are described in this section. In section 6, the calculations for each System's *Avg Precision* and *P@10* regarding each query were corrected as well as the values for the *MAP*. A short explanation of the way the weights of fields were chosen for System 3 in each query was also included.

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/.

[15] Booktrust.org.uk. n.d. [online] Available at: <https://www.booktrust.org.uk/globalassets/resources/bookbuzz/benefits-of-reading-for-pleasure.pdf> [Accessed 11 January 2022].

[16] Solr: You complete me! - The Apache Solr Autocomplete - Sease. Sease. (2022). Retrieved 13 January 2022, from https://sease.io/2015/07/solr-autocomplete-you-complete-me.html.

[17] Discounted Cumulative gain - Wikipedia. En.wikipedia.org. (2022). Retrieved 17 January 2022, from https://en.wikipedia.org/wiki/Discounted_cumulative_gain

[18] Discounted Cumulative Gain the ranking metrics you should know about. Medium. (2022). Retrieved 17 January 2022, from https://medium.com/@maeliza.seymour/discounted-cumulative-gain-the-ranking-metrics-you-should-know-about-e1d1623f8cd9

12 APPENDIX

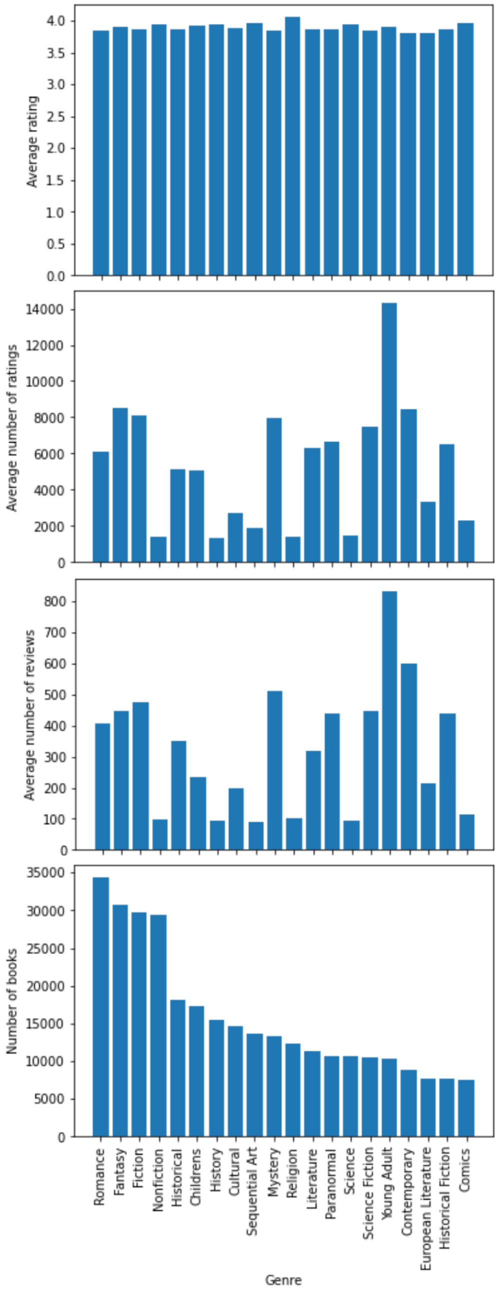


Figure 4: Average number of reviews and ratings, and average rating of the books that belong in the top 20 genres

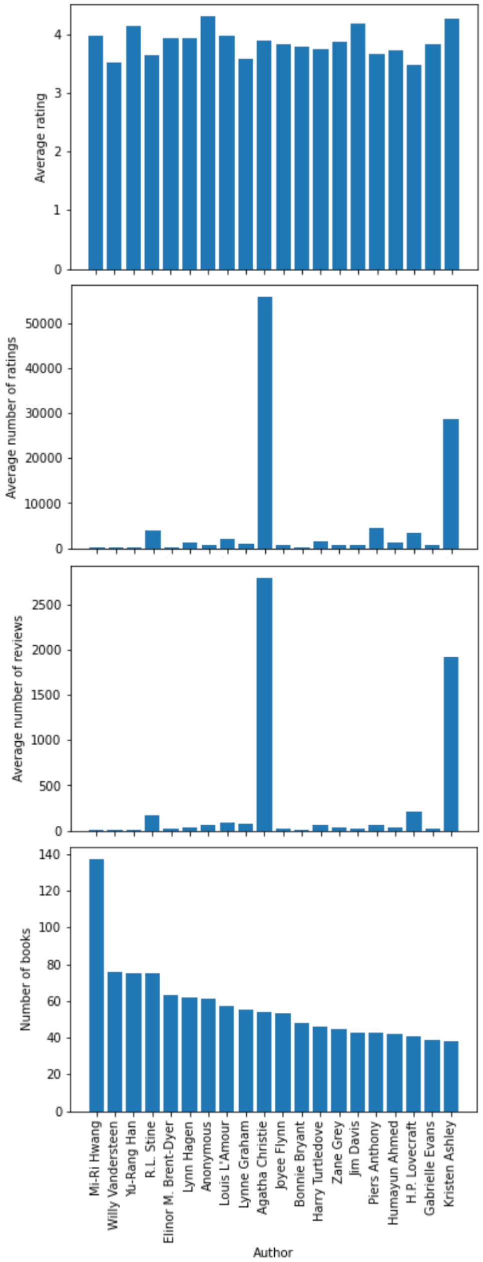


Figure 5: Average number of reviews and ratings, and average rating of the books written by the top 20 authors

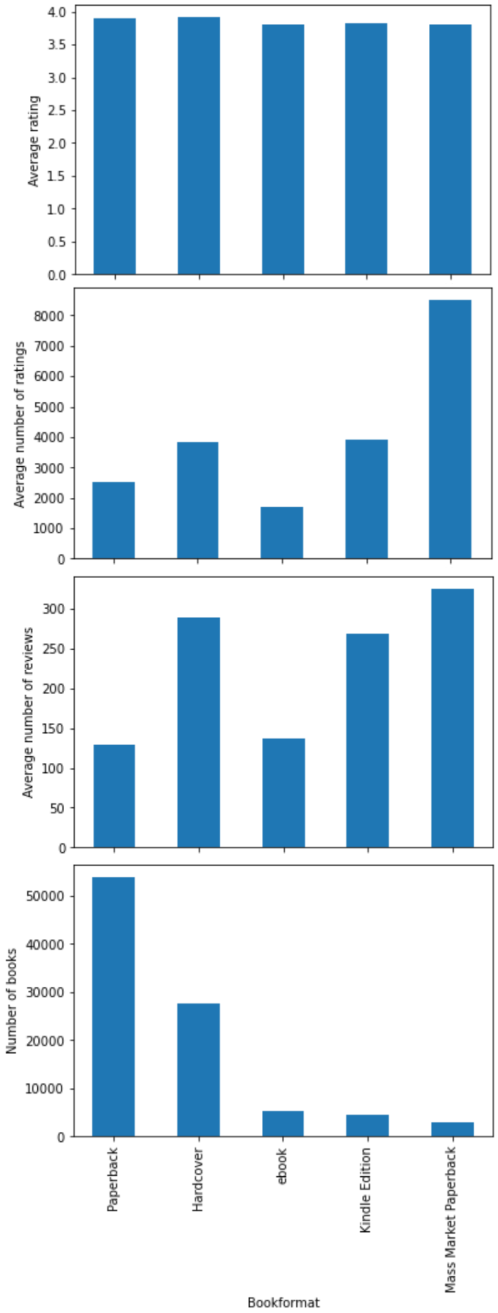


Figure 6: Average number of reviews and ratings, and average rating of the books that belong in the top 5 book formats

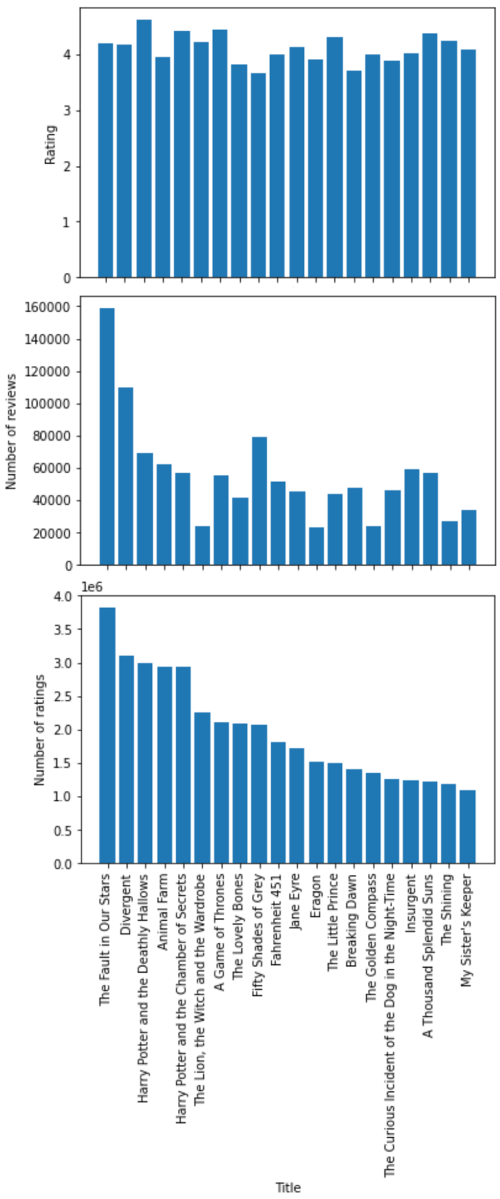


Figure 7: Average number of reviews and average rating of the top 20 books with most ratings

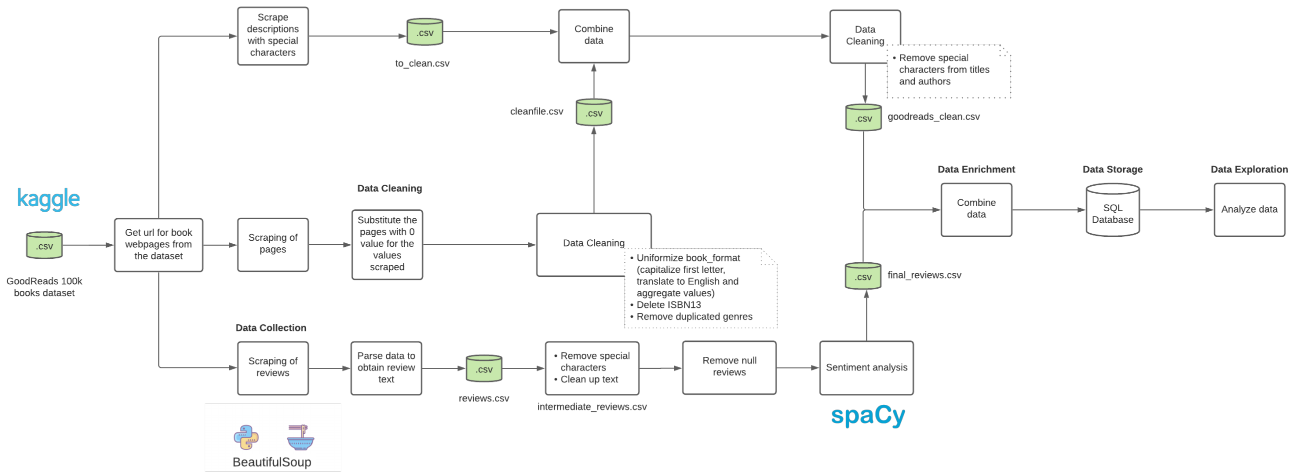

Scrape descriptions with special characters
.csv
to_clean.csv
Combine data
Data Cleaning
• Remove special characters from titles and authors
cleanfile.csv
.csv
goodreads_clean.csv
Data Enrichment
Data Storage
Data Exploration
Combine data
SQL Database
Analyze data
kaggle
.csv
GoodReads 100k books dataset
Get url for book webpages from the dataset
Scraping of pages
Data Cleaning
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
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

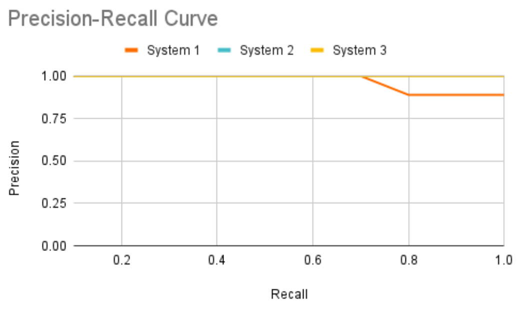


Figure 9: Precision-Recall graph for information retrieval 1

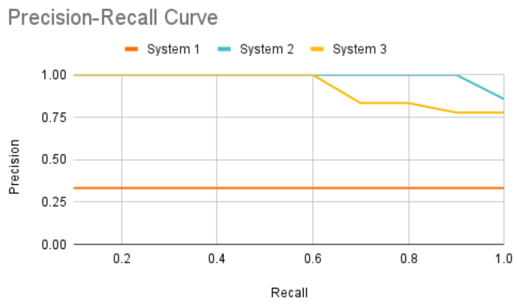


Figure 10: Precision-Recall graph for information retrieval 2

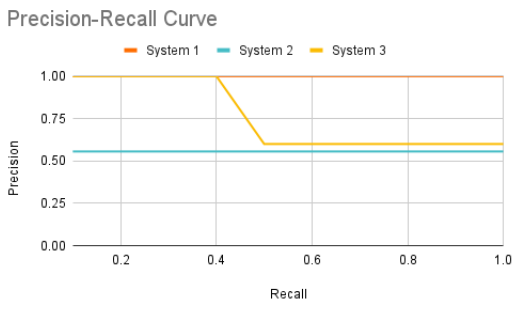


Figure 11: Precision-Recall graph for information retrieval 3

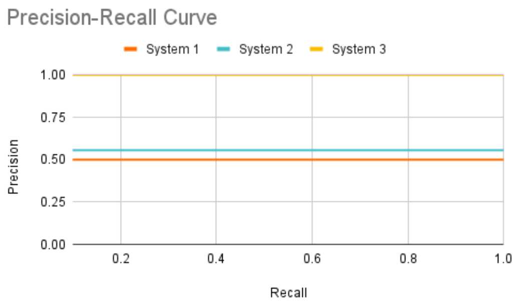


Figure 12: Precision-Recall graph for information retrieval 4

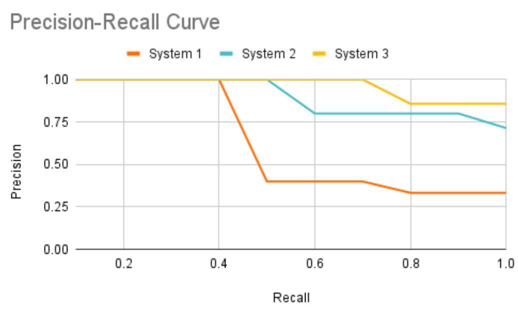


Figure 13: Precision-Recall graph for information retrieval 5

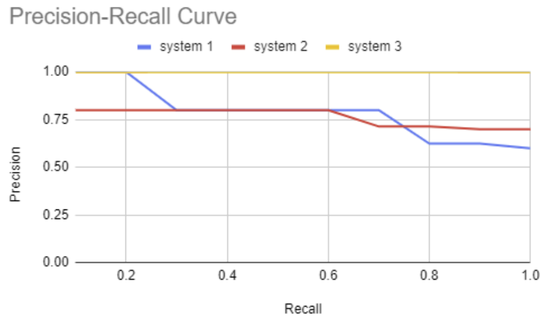


Figure 14: Precision-Recall graph for information retrieval 1 of the improved system

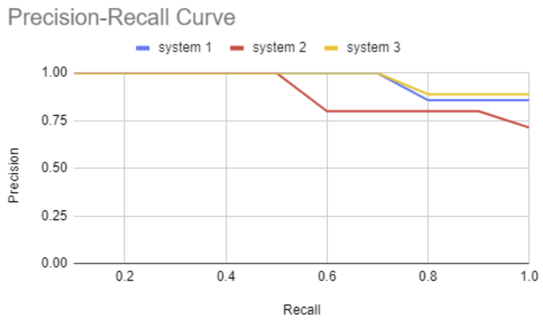


Figure 15: Precision-Recall graph for information retrieval 2 of the improved system

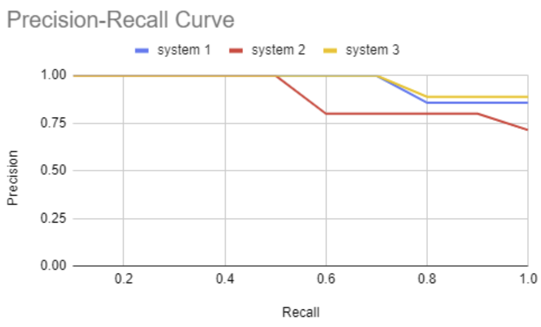
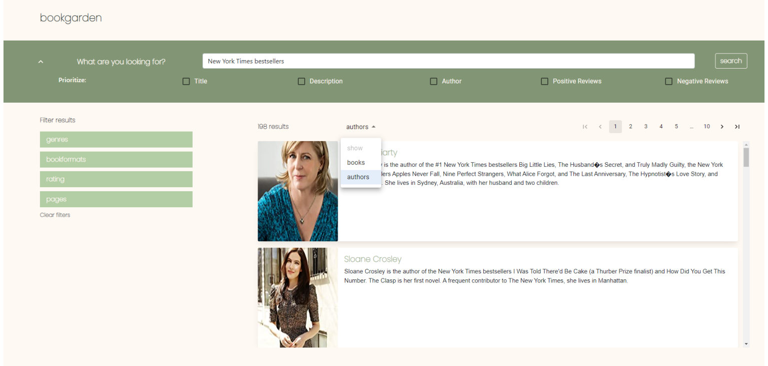


Figure 16: Precision-Recall graph for information retrieval 3 of the improved system



New York Times bestsellers

198 results

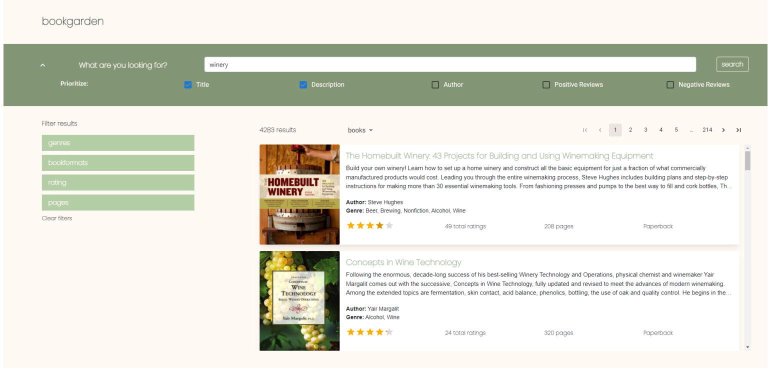
iarty

is the author of the #1 New York Times bestsellers Big Little Lies, The Husband�s Secret, and Truly Madly Guilty, the New York lers Apples Never Fall, Nine Perfect Strangers, What Alice Forgot, and The Last Anniversary, The Hypnotist�s Love Story, and . She lives in Sydney, Australia, with her husband and two children.

Sloane Crosley

Sloane Crosley is the author of the New York Times bestsellers I Was Told There'd Be Cake (a Thurber Prize finalist) and How Did You Get This Number. The Clasp is her first novel. A frequent contributor to The New York Times, she lives in Manhattan.

Figure 17: User interface for the New York Times bestsellers information retrieval



winery

4283 results

The Homebuilt Winery: 43 Projects for Building and Using Winemaking Equipment

Build your own winery! Learn how to set up a home winery and construct all the basic equipment for just a fraction of what commercially manufactured products would cost. Leading you through the entire winemaking process, Steve Hughes includes building plans and step-by-step instructions for making more than 30 essential winemaking tools. From fashioning presses and pumps to the best way to fill and cork bottles, Th...

Author: Steve Hughes
Genre: Beer, Brewing, Nonfiction, Alcohol, Wine

★★★★☆ 49 total ratings 208 pages Paperback

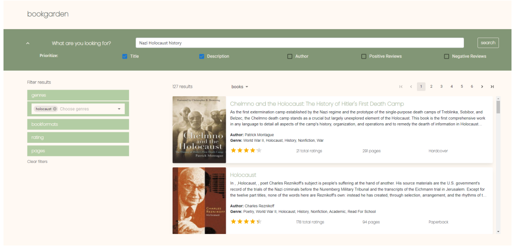
Concepts in Wine Technology

Following the enormous, decade-long success of his best-selling Winery Technology and Operations, physical chemist and winemaker Yair Margalit comes out with the successive, Concepts in Wine Technology, fully updated and revised to meet the advances of modern winemaking. Among the extended topics are fermentation, skin contact, acid balance, phenolics, bottling, the use of oak and quality control. He begins in the...

Author: Yair Margalit
Genre: Alcohol, Wine

★★★★☆ 24 total ratings 320 pages Paperback

Figure 18: User interface for the Wine information retrieval



bookgarden

What are you looking for? Nazi Holocaust history search

Prioritize: ☑ Title ☑ Description □ Author □ Positive Reviews □ Negative Reviews

Filter results

genres

holocaust Choose genres

bookformats

rating

pages

Clear filters

127 results books

1 2 3 4 5 6

Chelmno and the Holocaust: The History of Hitler's First Death Camp

As the first extermination camp established by the Nazi regime and the prototype of the single-purpose death camps of Treblinka, Sobibor, and Belzec, the Chelmno death camp stands as a crucial but largely unexplored element of the Holocaust. This book is the first comprehensive work in any language to detail all aspects of the camp's history, organization, and operations and to remedy the dearth of information in Holocaust…

Author: Patrick Montague
Genre: World War II, Holocaust, History, Nonfiction, War

21 total ratings 291 pages Hardcover

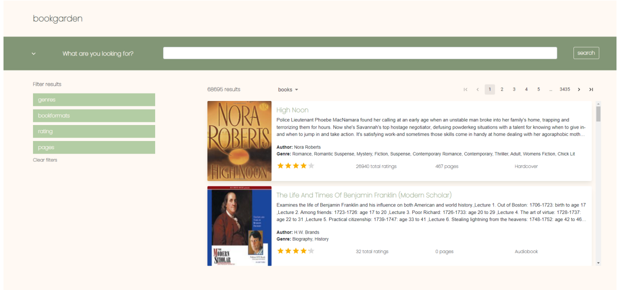
Holocaust

In , ,Holocaust, , poet Charles Reznikoff's subject is people's suffering at the hand of another. His source materials are the U.S. government's record of the trials of the Nazi criminals before the Nuremberg Military Tribunal and the transcripts of the Eichmann trial in Jerusalem. Except for the twelve part titles, none of the words here are Reznikoff's own: instead he has created, through selection, arrangement, and the rhythms of t…

Author: Charles Reznikoff
Genre: Poetry, World War II, Holocaust, History, Nonfiction, Academic, Read For School

178 total ratings 94 pages Paperback

Figure 19: User interface for the Nazi Holocaust information retrieval



bookgarden

What are you looking for? search

Filter results

genres

bookformats

rating

pages

Clear filters

68695 results books

1 2 3 4 5 … 3435

High Noon

Police Lieutenant Phoebe MacNamara found her calling at an early age when an unstable man broke into her family's home, trapping and terrorizing them for hours. Now she's Savannah's top hostage negotiator, defusing powderkeg situations with a talent for knowing when to give in-and when to jump in and take action. It's satisfying work-and sometimes those skills come in handy at home dealing with her agoraphobic moth…

Author: Nora Roberts
Genre: Romance, Romantic Suspense, Mystery, Fiction, Suspense, Contemporary Romance, Contemporary, Thriller, Adult, Womens Fiction, Chick Lit

26940 total ratings 467 pages Hardcover

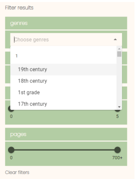
The Life And Times Of Benjamin Franklin (Modern Scholar)

Examines the life of Benjamin Franklin and his influence on both American and world history.,Lecture 1. Out of Boston: 1706-1723: birth to age 17 ,Lecture 2. Among friends: 1723-1726: age 17 to 20 ,Lecture 3. Poor Richard: 1726-1733: age 20 to 29 ,Lecture 4. The art of virtue: 1728-1737: age 22 to 31 ,Lecture 5. Practical citizenship: 1739-1747: age 33 to 41 ,Lecture 6. Stealing lightning from the heavens: 1748-1752: age 42 to 46…

Author: H.W. Brands
Genre: Biography, History

32 total ratings 0 pages Audiobook

Figure 20: User interface for the main view of the search system interface



Filter results

genres

Choose genres

1

19th century

18th century

1st grade

17th century

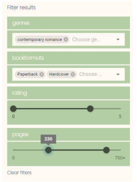
0 5

pages

0 700+

Clear filters

Figure 21: User interface for filter box, where it is possible to see the facets implemented



Filter results

genres

contemporary romance Choose ge…

bookformats

Paperback Hardcover Choose …

rating

0 5

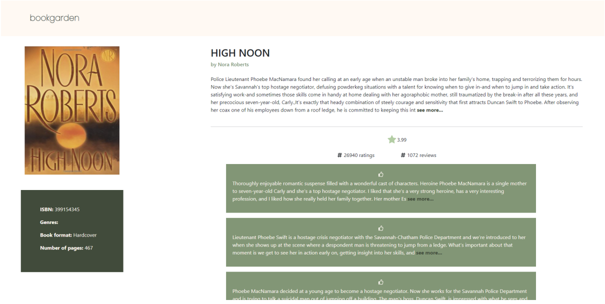
pages

230

0 700+

Clear filters

Figure 22: User interface for the filter box, containing the different fields a user can use to filter the query



bookgarden

HIGH NOON

by Nora Roberts

Police Lieutenant Phoebe MacNamara found her calling at an early age when an unstable man broke into her family's home, trapping and terrorizing them for hours. Now she's Savannah's top hostage negotiator, defusing powderkeg situations with a talent for knowing when to give in-and when to jump in and take action. It's satisfying work-and sometimes those skills come in handy at home dealing with her agoraphobic mother, still traumatized by the break-in after all these years, and her precocious seven-year-old, Carly.,It's exactly that heady combination of steely courage and sensitivity that first attracts Duncan Swift to Phoebe. After observing her coax one of his employees down from a roof ledge, he is committed to keeping this int **see more...**

3.99

26940 ratings # 1072 reviews

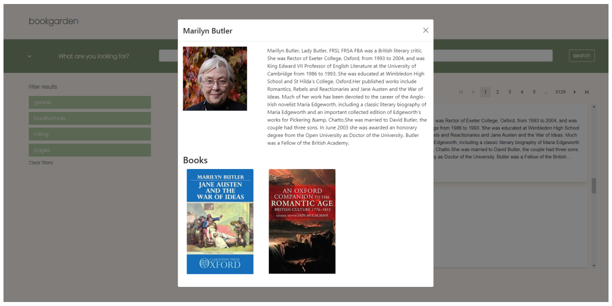
Thoroughly enjoyable romantic suspense filled with a wonderful cast of characters. Heroine Phoebe MacNamara is a single mother to seven-year-old Carly and she's a top hostage negotiator. I liked that she's a very strong heroine, has a very interesting profession, and I liked how she really held her family together. Her mother Es **see more...**

Lieutenant Phoebe Swift is a hostage crisis negotiator with the Savannah-Chatham Police Department and we're introduced to her when she shows up at the scene where a despondent man is threatening to jump from a ledge. What's important about that moment is we get to see her in action early on, getting insight into her skills, and **see more...**

Phoebe MacNamara decided at a young age to become a hostage negotiator. Now she works for the Savannah Police Department

ISBN: 399154345
Genres:
Book format: Hardcover
Number of pages: 467

Figure 23: User interface for book's page



bookgarden

What are you looking for? search

Filter results

genres

bookformats

rating

pages

Clear filters

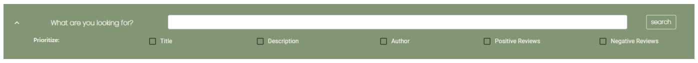
1 2 3 4 5 … 3129

Marilyn Butler

Marilyn Butler, Lady Butler, FRSL FRSA FBA was a British literary critic. She was Rector of Exeter College, Oxford, from 1993 to 2004, and was King Edward VII Professor of English Literature at the University of Cambridge from 1986 to 1993. She was educated at Wimbledon High School and St Hilda's College, Oxford.Her published works include Romantics, Rebels and Reactionaries and Jane Austen and the War of Ideas. Much of her work has been devoted to the career of the Anglo-Irish novelist Maria Edgeworth, including a classic literary biography of Maria Edgeworth and an important collected edition of Edgeworth's works for Pickering &amp, Chatto.She was married to David Butler, the couple had three sons. In June 2003 she was awarded an honorary degree from the Open University as Doctor of the University. Butler was a Fellow of the British Academy.

Books

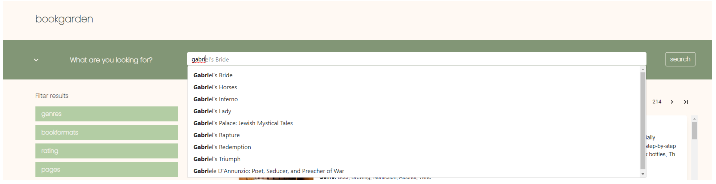
Figure 24: User interface for author's page



What are you looking for? search

Prioritize: □ Title □ Description □ Author □ Positive Reviews □ Negative Reviews

Figure 25: User interface check boxes to choose which fields to prioritize



bookgarden

What are you looking for? gabriel's Bride search

Gabriel's Bride
Gabriel's Horses
Gabriel's Inferno
Gabriel's Lady
Gabriel's Palace: Jewish Mystical Tales
Gabriel's Rapture
Gabriel's Redemption
Gabriel's Triumph
Gabriele D'Annunzio: Poet, Seducer, and Preacher of War

Filter results

genres

bookformats

rating

pages

Figure 26: User interface with suggestions on the search bar