Goodreads Books and Reviews

Diana Freitas, Diogo Samuel Fernandes,Juliane Marubayashi
up201806230@edu.fe.up.pt,up201806250@edu.fe.up.pt,up201800175@edu.fe.up.pt

ABSTRACT

This work focuses on the refinement, characterisation and querying of datasets regarding Goodreads[6] books and reviews. It has the ultimate goal of developing a search system that improves the quality of the search, specially by exploring Information Retrieval tools to extract new and relevant information from the books description and title. It also aims to improve the utility given to the user reviews regarding the books. To this end, the datasets, collected from UCSD book graph [7], were first refined and then explored in order to detect patters that could be relevant to later define the information needs. In particular, the exploration of the data, allowed to identify the most frequent words that relate to popular book themes and also to understand the aspects that were most discussed in reviews. The books and reviews documents were then defined, indexed and queried using Solr in order to satisfy some information needs that will be explained later in this article. Finally, it has been proven that the defined information needs were successfully satisfied, by evaluating the average precision, the recall and the precision at 5.

1 INTRODUCTION

It wasn't that long ago when people used to get together to decide the next book to read. By engaging in social interactions, either in personal or professional environments, readers could get book recommendations and avoid the difficult task of choosing the right book by chance.

Nowadays, with the influence of the internet in the globalization process, the access to information is at our fingertips. The ease of access to information, however, also comes with a great cost: the increasing amount of data has made it difficult to effectively make decisions. According to [9] "Information overload shuts our brains down", "The more choices we have, the more challenging the decision". Big Data [3], by providing high volume and diversity of information, brings up the increasing need to filter what is relevant to our needs.

Currently, the idea of data filtering is widely spread, and websites such as Goodreads [6], the world's largest website of book details and recommendations, are becoming more and more popular.

By taking into account this information overload and the need to filter the available data, in the scope of Book recommendations, the purpose of this project is to gather and process a subset of information about Goodreads books, with the ultimate goal of providing valuable book details and recommendations to readers, through the development of an information search system.

The article is divided in two parts. The first addresses the process of Data Preparation, focusing on the collection, refinement and characterisation of datasets regarding Goodreads books, reviews, genres and authors. The final version of the datasets was then represented in a conceptual model. The second part describes the Information Retrieval task. First, the information needs are defined and justified. The choice of retrieval tool is then justified, followed by a description of the documents and of the indexing process. Finally, the results are analysed and justified by comparing the expected results and the obtained ones. The conclusions were firmly supported by making use of quality measure methods such as precision and recall.

2 DATA PREPARATION

The Data Preparation stage will be introduced with a brief explanation of the pipeline's structure. In the following sections, each of the pipeline's stages (Data Gathering, Cleaning and Exploration) will be explained in more detail. The methodology used to assess the quality of the data and the issues that were uncovered will also be addressed in a separate section. Finally, a description of the conceptual model of the data domain is also included, followed by a brief conclusion about the obtained results.

2.1 Pipeline

A data pipeline consists of a chain of data processing elements, that represent the flow and transformation of the data. In the pipeline that resulted from this project, three stages can be clearly identified: Data Gathering, Data Cleaning and Data Exploration.

The **Data Gathering** stage simply shows the four original **json** datasets - books, authors, reviews and genres were downloaded from UCSD book graph[7]. This stage is followed by the **Data Cleaning** process, which consists of two consecutive tasks:

Firstly, it is responsible for modifying the data of each dataset individually, by removing irrelevant columns, converting data types or removing observations with missing values. After applying the desired transformations to the data, each dataset was stored. This output format was selected because it consumes less disc space and it is easier to work with. The format conversion could not be performed at the beginning of the pipeline, since there were nested objects.

The second task of the **Data Cleaning** stage combines the processed datasets by using the ids of the books and of the authors. When combining the datasets, some observations that had references to others that were dropped in the previous step were removed (Some reviews referred books that were removed for not having a description). Also, as a result of this processing step, some many-to-many relations, in particular, the relations between genres and books, and between authors and books were extracted. These relations were represented, in the original data, as columns that stored an array of foreign keys to the other relation.

Last, the **Data Exploration** process, unlike the other pipeline stages, does not modify the datasets. However, its performance and value are impacted by the previous stages of the pipeline. This last stage refers to the use of data visualization and statistical techniques to better characterise the datasets and identify valuable information about the nature of the data.

2.2 Data Gathering

Initially, when selecting the data that would be used for the project, the idea was to extract it from Goodreads.com using Web Scrapping, in order to get up to date book details and reviews. However, Goodreads' Terms of Use [11] clearly state that Data Gathering is no longer allowed, and that new developer keys for the public developer API are no longer being issued.

However, some datasets were available from the time when scrapping was allowed and when the Goodreads API was still active. From the available sources, we decided to use *Ucsd Book Graph* [7], which provides Goodreads datasets for academic use, collected in late 2017 and updated in 2019. These datasets include book metadata, user-book interactions and book reviews, and can be merged together by matching book/user/review ids.

2.3 Data Quality

When selecting the datasets that would be used for the project, it was important to assess the quality of the data and how it was to serve the project's purpose. Therefore, during this stage, a search was performed in order to find datasets that were rich in textual data, which, considering the selected theme, would probably be found in reviews or book descriptions. Also, the most common data quality problems were assessed, by identifying the quantity of missing or inconsistent values, by addressing the validity of the data, in particular, checking if the data conformed with type, unique and foreign-key constraints, and by searching for uniformity problems. Event though the four selected datasets had few data quality issues, some problems, that would need to be fixed in the Data Cleaning stage were detected, namely: the date format was not consistent in all the datasets, many integer ids were stored as strings and there were many books without any genre associated (missing values).

2.4 Data Cleaning

After exploring the datasets regarding Goodreads books, four of them were selected for the project - books, authors, genres and reviews. There was also a user dataset available in the same source, however, as most of the relevant user information was not available due to data protection, the available client details were discarded. Once the datasets were chosen, their volume was evaluated and the most relevant attributes were identified, which was essential to determine the size of the subsets to use, which attributes to drop, and which columns to aggregate or clean. The Data Cleaning process, which will be described and justified in detail was first performed on each dataset individually, using the json module [8]. The datasets were then stored in **csv** files in order to simplify integration.

Books

The original books dataset (**books.json**) consists of about 2.3M observations, each with 29 attributes. This dataset was initially reduced to a subset of 11 thousand observations, to allow a faster processing. All the selected books have a description, which will be used in for text search. Furthermore, the publication_year, publication_month and publication_day were merged into a single date column. Some columns were also dropped mainly due to one of the the following reasons:

- they were references to observations of datasets that were not used (series and work_id);
- they were related to commercial details about the book (asin and kindle_asin);
- they contained derived information that didn't make sense when using subsets of the original datasets (text_reviews_count and ratings_count);
- they contained information that was already represented in another dataset (instead of using the attribute popular_shelves, which represented the genres attributed by the users, a genres dataset, that was also available, was used);
- they did not add relevant information to the data (country_code, link, url, title_without_series).

Also, the similar_books attribute was dropped, because the similarly between books can easily be evaluated when searching the dataset, by filtering by genre or author, for example.

Authors

For the authors dataset, essential information was kept, dropping some attributes that were derived from the original and larger dataset, in particular the text_reviews_count and the ratings_count.

Genres

In Goodreads, the genres of each book are defined by the users, and therefore, for each genre associated with a book_id, the original dataset included the number of users that categorized it with a certain genre. In this first Data Cleaning phase, the number of users that classified the book with each genre was discarded, only keeping the names of the genres associated with each book_id. All the observations that represented books without any genre classification were removed, and a subset of observations was selected to ensure an easier processing.

Reviews

The reviews dataset included more than 1.3M English book reviews of books. A subset of reviews was used, dropping the user_id and some details related with the management and statistics of each review, in particular the date_updated, started_at, read_at, n_comments and n_votes. As this dataset included reviews with spoilers, which are labeled using '(view spoiler)[' and '(hide spoiler)]', the spoilers were kept, but their delimiting tags were removed.

Combining the datasets

The datasets were then combined in the following order:

(1) The genres dataset was combined with the books dataset by book_id, dropping all books that did not have any genre observation and all genres observations that were not related to any book. Then, the **genres.csv** was adapted to map the name of each genre to an id. Finally, by combining this datasets, we extracted a many-to-many relation between books and genres, which was also saved as a **csv** file.
(2) The authors of each book, which were originally stored in a list in the authors column of **books.csv**, were combined with the books dataset by author_id, resulting a many-to-many relation which maps author_id with book_id. Also,

all authors that were not associated with any books of the chosen subset were discarded.

(3) The reviews were combined with the books, in order to discard reviews that were not related to any book of the chosen subset.

2.5 Data Exploration

In the last step of the data processing pipeline, the exploration of the processed datasets uncovered some patterns, and characteristics of the data, by means of visualization, with distribution and correlation plots and by generating tables that expose descriptive statistics - data types, number of missing values, maximum, minimum, mean, among others. This process does not aim to explore every bit of information of the Goodreads database, but rather to improve the understanding of the data properties and relations, which will be determinant for the success of the final search system.

Books

The books dataset contains a collection of 9483 books, with a representative amount of exemplars available in physical format.

To attest the variety of books in the dataset, the range and distribution of some of the attributes that refer to book meta-data, such as the number of pages, was analysed. If the dataset was only composed by books with a number of pages between 1 and 100, for example, it could mean that the sample lacked diversity. As we can see in plot [1], this scenario wasn't materialized.

In addition to verifying the diversity of a database, studying the composition of the data is also important. The books dataset contains missing values, namely the isbn, language_code, format, publisher, num_pages, isbn13 and edition_information. The missing information wasn't treated neither dropped, since it's common that some books don't have an isbn identification, for example, and this fact does not influence the quality of the book. Finally, a word-cloud was used in order to capture the main ideas of the meta-data.

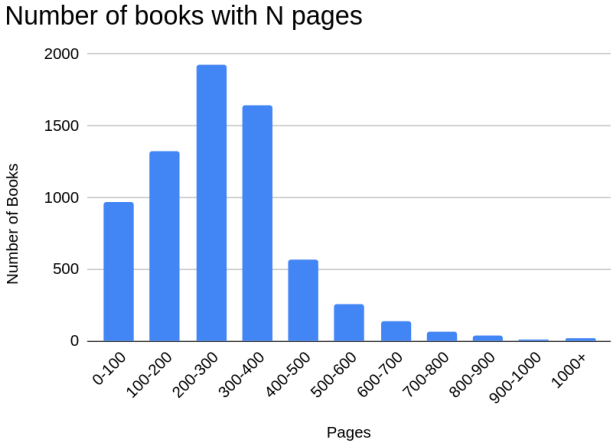


Figure 1: Distribution of the number of pages

Authors

The author database consists of 11508 authors, of which only 231 have an average rating, which implies that not all authors are popular.

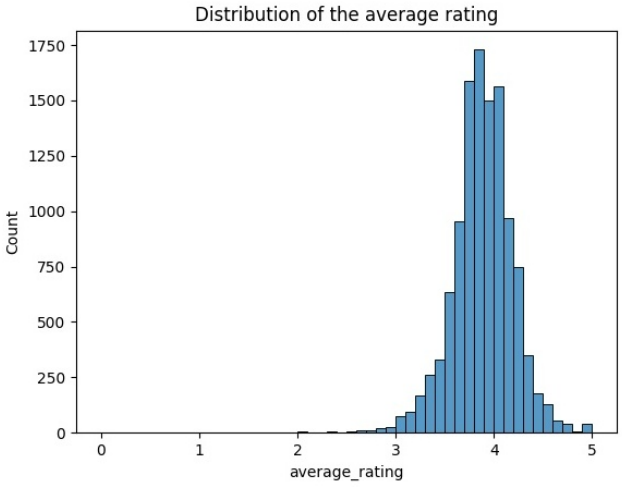


Figure 2: Average rating histogram

Moreover, each author has written at least one book, and, even the majority of authors has composed just one book, there are some outliers, particularly some books with 50 and 51 authors. Fortunately, the presence of outliers doesn't cause a negative impact in this research, thus these observations don't have to be dropped. In fact, they provide us valuable data.

Genres

The genres table contains 16 types of genres, from which the most common one is *fiction*, with over 6000 samples, and the least common is *poetry*.

An interesting fact is that the histogram for each of the genres follows a shifted normal distribution, where the center comes around the rating 4.

Reviews

By exploring the review dataset, which is composed of 2879 reviews, it was concluded that only 129 books contain a review. Following the same pattern of the genres and authors, the reviews' rating histogram follows a shifted normal distribution with mean around the number 4.

Finally, by analysing the word-cloud generated from the text of the reviews, the presence of some words that might influence the book rating, such as "love" and "enjoyed" was noticed. However, after analysing the correlation between those words and the ratings, no correlation was detected.

Figure 3: Reviews word-cloud

2.6 Conceptual Model

The datasets that were used in the project are directly related to the main entities of the Goodreads Conceptual Model - Book, Author, Genre and Book Review. Each book has a title, a description, and many other attributes that describe essential information about it, like the publisher, number of pages and isbn, among others. A book is either an ebook or a physical book, and can have many authors and many reviews. Each review has a date, a rating and a text field, which may be used for text-search. A book may belong to multiple literary genres, which are simply characterised by their name.

The conceptual model represents the Goodreads' data domain.

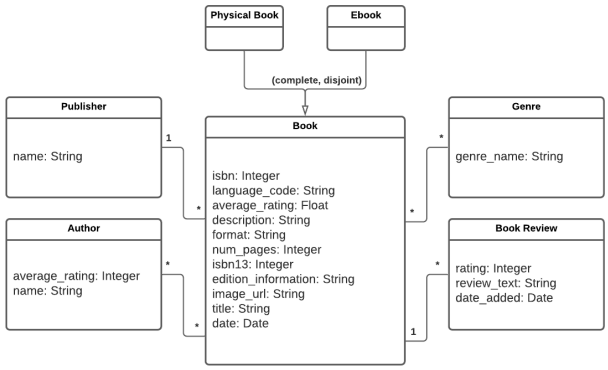


Figure 4: Conceptual Model

3 INFORMATION RETRIEVAL

Information Retrieval is the process of obtaining information about system resources that are relevant to an information need. The focus of Information Retrieval is on unstructured information, like document collections or the web [10]. Searches can be based on full-text or other content-based indexing. In pursuance of a good search system, many filter combinations were added to different fields in order to achieve better results and a more flexible search. After that, the search system was evaluated by using different metrics, such as average precision, recall and precision at 5 (P@5).

3.1 Information Needs

The selection of the information needs to be addressed in the Information Retrieval task was guided by the analysis of the data and by the exploration of the available tools. This lead to the conclusion that the quality of the search could be improved by providing the possibility to search books by thematic, specific events and named-entities[2], that may be of interest to the end user. Also, the user may want to search for books based on the opinion of other users, which suggests that the system would also benefit if the negative and positive reviews could be distinguished. All in all, the information needs to be approached are the following:

(1) **Search for books about a specific theme**: It aims to allow the end user to refine the search for books by specifying its thematic. For example, when a user wants to search for romantic tragedies, filtering by genres would not suffice. Therefore, it is necessary to maximize the relevance obtained when searching the description and title of the books for specific themes.
(2) **Search for positive and negative feedback on a book review**: To allow the user to search for books based on the opinion of others, by distinguishing negative and positive reviews.
(3) **Search for books about historical events/named-entities** : Given that historical events and some named-entities are a common topic in many books, it was important to understand the available search alternatives in order to boost the relevance of the results. This will be assessed by searching for a particular historical event, the "World War".
(4) **Search for scientific books**: Scientific books are usually included in the non-fiction genre. However, when searching for science related words without any search improvements, science-fiction books, which may not be based on facts, would also be retrieved. Thus, the system will be enhanced in order to improve the relevance of the results.

These information needs will be addressed in the following sections through the creation of queries and by evaluating the results obtained.

3.2 Retrieval Tool

Before diving deep into Information Retrieval we needed to choose one tool. We had many alternatives at our disposal like Apache Solr and ElasticSearch. Apache Solr [15] is an open-source search platform that uses Lucene Java search library at its core for full-text search and indexing. Solr also as a REST API and has a bigger community and more documentation in comparison to ElasticSearch. ElasticSearch [14] is developed alongside the log-parsing engine Logstash, the analytics, the visualization platform Kibana, and the collection of lightweight data shippers called Beats. These four products are referred to as the "Elastic Stack" [1] which was not necessary for this project. After weighting and analyzing both tools, it was decided that Solr engine would be a better fit for this information retrieval task.

3.3 Documents and Collections

Given that the main entities of this project are the books, and considering one of the information needs is to search through books by theme, it was necessary to have a books document, the source of information. This document stores, for each book, all the essential details about it, the most important being the title, description, genres and authors.

Since there will also be queries to retrieve information about reviews, like positive and negative feedback about a book based on the text of the reviews and its rating, a separate document was also used to aggregate all of them. This document stores the relevant information about each review, such as the text and rating, and also some details about the book they refer to, particularly its title, genres, authors and date.

Furthermore, to better evaluate the results of the queries regarding the defined information needs, different subsets were defined for both documents. The recall metric, for example, requires the identification of all relevant books, which can't be achieved in a large dataset.

In Solr, two main cores were used, one for the reviews, which contains 2819 documents, and the other for the books, which contains 9483 documents. Also, for each subset a new core was created. Considering the fact that there are multiple common attributes between cores, all cores share the same schema.

3.4 Indexing and Schema Definition

Since some fields will not be used for search, the first step was to identify these fields that only provide extra information, such as the isbn, the language, and the book format. These fields, among others that won't be used in queries, are stored using the appropriate field types but with the indexed property set to false.

The fields that are going to be used to retrieve data and to build the queries, which will satisfy the information needs, are the ones to be indexed. The indexed fields mostly represent text fields, such as the book description and the review text, which will be frequent search targets, either to extract new information regarding the theme of a book or to find feedback left by users. However, some numeric fields were used in the enhanced queries and, therefore, were also indexed. The indexing was quite simple for this numeric fields. The rating of a book, for example, is indexed and stored using Solr *TrieFloatField* type.

However, when it came to text fields, Solr's default type, *TextField*, was not enough, as each text field requires specific tokenizers and filters. Therefore, custom field types [4] were created, allowing the specification of different index and query analysers for each field. The field types that were created were the following: *comma-separated-list*, *description*, *title* and *review*.

Since book genres and authors are separated by semi-colons in the document, a field type was created in order to split them with a *Pattern Tokenizer*.

The *description*, *title* and *review* field types, which map to the fields with analogous names, use a *Standard Tokenizer* with a *Lower Case Filter* at index and query time, in order to convert uppercase letters to the equivalent lowercase and, this way, perform a case insensitive search.

Description texts and reviews also use the *English Minimal Stem Filter*, which converts words to singular. Since the reviews can be searched to look for spoilers, events or details about an author or about the plot, the *Porter Stem Filter* and the *Possessive Filter* were also used: the first to remove the endings of conjugated verbs and the second to remove trailing 's from the words. For reviews, stop words were also used, for example, to remove 'this' in the query 'hated this book', increasing the number of matches.

Since the review text and the book description include a wide range of words, if the results were limited to books and reviews that contained an exact match, then there would be some relevant results regarding the searched topic that wouldn't be retrieved. For example, by adding synonyms, if the user searches for the word 'love', books containing the word 'romance' should be also considered relevant. For this reason, the *Synonym Graph Filter* is used. The synonym analysis could take place at two distinct phases, at index time or query time [13]. In the first one, when a field is being created, the token that results from the analysis is added to an index. At query time, the values being searched for are analyzed and the terms that result are matched against those that are stored in the field's index. Using the synonyms filter at query or index time has different advantages. Index time synonym expansion requires re-indexing every time the synonym file is changed. On the other hand, if the filter is applied at query time, the index keeps its size and there's no need to re-index. Query time synonyms make it difficult to support multi-word synonyms, however, since we're only using single words, this problem does not affect us [5]. All things considered, given that most of the queries will be quite short, the synonym filter will be used at query time.

Listing 1: Synonyms File

```
# Simple synonyms
love, romance, romantic, heartbreak, heart, heartache,
     passion, passionate
tragedy, drama, tragic, disastrous, disaster

# Sentiment synonyms
amazing, good, quality, nice, great, impressive, love,
     not bad, fascinating, really liked
bad, worst, horrible, hate, awful, boring, terrible,
    disappointed

# Scientific synonyms
science, mathematics, scientific, study, numeric
```

3.5 Retrieval and Evaluation

The evaluation is the heart of the information retrieval process. By studying the effectiveness of each query, the system can be improved by applying gradual updates that maximize the relevance of the retrieved results.

The standard evaluation approach for information retrieval systems revolves around the binary classification of results (relevant or not relevant). This project follows this method of classification and creates a controlled environment (a data subset) to make the accounting of relevant and irrelevant documents possible.

For each information need, at least two queries were created and classified as distinct systems. One of these systems attempts to satisfy the information need by performing a simple query. The other system tries to improve the first query by making use of new approaches and parameters offered by the Solr engine. In some cases, this systems are also tested with two different schemas, in order to study the impact of the selected filters on the relevance of the results.

For each query analysis an hypothesis is made: the simple system is worst than the second one. By making experiments, studying and analysing the results of each system, the verification of the prediction is analyzed and discussed.

The precision recall plots that portray the results for each system can be visualized in the section "Precision recall plots" of the annexes .

Books about romantic tragedy

This search seeks to find books based on a specified thematic, in this case books with romance and tragedy would be retrieved. The query chosen to achieve the search goals is the [romantic tragedy]. For this search two systems were created: the first one is a simple query that searches for the words "romantic" and "tragedy" in the *description* field with a limit of 8 results. The system 2 explores the proximity of the two words and applies a boost to the description. Both systems use the final schema, which uses the *Synonym Graph Filter Factory* to also match the defined synonyms of the words "romantic" and "tragedy".

Table 1: Thematic search

	System 1	System 2
Parameters	description:romantic tragedy q.op = OR rows=8	q= {!q.op=OR df=edismax}romantic tragedy qf=description^2 ident=true ps=4 rows=8

Table 2: Thematic search systems comparison

	System 1		System 2	
Rank	**Title**	**R**	**Title**	**R**
1	Three Sides of a Heart: Stories about Love Triangles	Y	Wuthering Heights	Y
2	A Boy in Winter	Y	Three Sides of a Heart: Stories about Love Triangles	Y
3	The Medicine Man: Book 2	Y	A Boy in Winter	Y
4	Ride a Dark Horse	Y	The Medicine Man: Book 2	Y
5	Devil You Know (Lost Boys #1)	Y	Ride a Dark Horse	Y
6	Crown of Midnight (Throne of Glass, #2)	N	Devil You Know (Lost Boys #1)	Y
7	Wickett's Remedy	N	Crown of Midnight (Throne of Glass, #2)	N
8	Titus Andronicus	N	Wickett's Remedy	N
AP	0.8965		0.950893	
AP@5	1.0		1.0	
Recall	0.83		1.0	

As expected, system 2 retrieved better results than the first one: the second system presents betters results by retrieving 6 out of the 7 relevant books, while the first system retrieves 5 out of the 7. By analysing the results manually, we can say that the difference between both queries relies on the terms distance: while the first query prioritizes the number of occurrences of each word, the second relies on the distance. Since the first book in system 2 raking contains the following phrase: "love in the wake of tragedy" (the distance between 'love' and 'tragedy' is 3) it appears as the most relevant book in the raking. This same title, though, contains low frequency of the words related to 'tragedy' and 'love' and for this motive is not among the most 8 relevant books in the first system.

World War

Still might be the case where the user searches for a thematic related to historical moments or named entities[2]. It's not uncommon that specific moments in history and named-entities are referenced with more than one word, such as: "World War II" and "Barack Obama". Although words grouped might have a different meaning than when separated, the frequency with which these terms appear in sequence can be high, and thus it's possible to search for these words separately and limit the distance between them.

The three systems to be presented in this section tries expose the logic explained above and to compare both approaches.

Table 3: Search for World War

	System 1	System 2	System 3
Parameters	description: "world war" rows=10	description: "world war" rows=10	description:"world war"~5^5 defType:edismax qf=description^1 rows=10

Table 4: Search for world war theme results

	System 1		System 2		System 3	
Rank	**Title**	**R**	**Title**	**R**	**Title**	**R**
1	The Omega Covenant	Y	The Winds of War (The Henry Family, #1)	Y	The Winds of War (The Henry Family, #1)	Y
2	The Pearl Harbor Murders (Disaster, #3)	Y	Ancestors in Arms	Y	Ancestors in Arms	Y
3	The Mandibles: A Family, 2029-2047	Y	Gated Grief: The Daughter of a GI Concentration Camp Liberator Discovers a Legacy of Trauma	Y	Gated Grief: The Daughter of a GI Concentration Camp Liberator Discovers a Legacy of Trauma	Y
4			Women Heroes of World War II: 26 Stories of Espionage, Sabotage, Resistance, and Rescue	Y	Women Heroes of World War II: 26 Stories of Espionage, Sabotage, Resistance, and Rescue	Y
5			Season In Purgatory	Y	Season In Purgatory	Y
6			History of Modern World	Y	History of Modern World	Y
7			Mein Kampf - My Struggle: Unabridged edition of Hitlers original book - Four and a Half Years of Struggle against Lies, Stupidity, and Cowardice	Y	Mein Kampf - My Struggle: Unabridged edition of Hitlers original book - Four and a Half Years of Struggle against Lies, Stupidity, and Cowardice	Y
8					Star Wars: The Ultimate Visual Guide: Updated and Expanded	N
9					Other People's Houses	Y
AP	1.0		1.0		0.973765	
AP@5	0.6		1.0		1.0	
Recall	0.3		0.7		0.8	

As expected, even tough the first system contains a high average precision, it doesn't return many results from the relevant titles contained in the sub dataset: the lack of filtering prevents the engine from retrieving data with upper case or plurals. By activating the filtering, the second query improves the results significantly by returning books that contain other variations of "world war", such as "World War", "World Wars", etc. Despite the fact that system 3 searches for "world war" separately, it still managed to return more relevant results than the system 2. However, to achieve this results the average precision slightly decreases, which was also expected.

In addition to that, the book *Other's People Houses* was not retrieved in the first system, because it does not contain an exact match to "World War", even though the contents of the description indirectly refer to this historical event.

It is worth mentioning that it's still possible to improve this search, by adding a boost in the *genres* field, filtering it by "history". However, as mentioned before, the query is not restricted to only history books, but rather terms that references events, named-entities, historic moments, etc.

Negative Reviews

Table 5: Search for negative reviews

	System 1	System 2
Parameters	q = {!q.op=OR df=review_text}hated this books rows=10	q= {!q.op=OR df=review_text}hated this books AND rating:[0 TO 3.5] sort = field(rating, min) asc rows=10

This search attempts to find all bad reviews. Normally, a user could try to find bad reviews, for example, by adding words such as "hate" and "terrible" to the search field. To simulate this approach the system 1 performs a simple query and proves that this query might return some relevant results, but the majority is not: the words referred in the search field do not necessarily describe the book, but rather characters or part of the history. The system 2, however, tries to improve the relevance by also limiting the reviews rating and sorting them from the worst to the best. Still, this query does not return all the relevant results, since not necessarily a bad review has a low rating.

Table 6: Search for negative reviews results

	System 1		System 2	
Rank	**Title**	**R**	**Title**	**R**
1	On Stranger Tides	N	The Collector (Dante Walker, #1)	Y
2	Taunting Destiny (The Fae Chronicles, #2)	Y	I Am the Messenger	Y
3	I Am the Messenger	Y	On Stranger Tides	Y
4	By Any Other Name (Forbidden, #1)	Y	The Six Rules of Maybe	Y
5	Taunting Destiny (The Fae Chronicles, #2)	Y		
6	Taunting Destiny (The Fae Chronicles, #2)	Y		
7	The Name of the Wind (The Kingkiller Chronicle, #1)	Y		
8	A Love Letter to Whiskey	N		
9	Solaris: The Definitive Edition	N		
10	Born a Crime: Stories From a South Afican Childhood	Y		
AP	0.435794		1.0	
AP@5	0.4		0.8	
Recall	0.285714		0.571429	

Likewise, a the search for good reviews was performed, documented[11] and studied. The conclusions for this second analysis will not be provided in this report, due to the similarity of the first one.

Scientific books

When the initial system was queried for Scientific books, we noticed that Science Fiction books were also retrieved. Therefore, the goal of this query was to improve the relevance of the results when searching for words related to "science" by boosting books that have the non-fiction genre and by excluding the ones that are categorized as fiction or historical fiction. For the evaluation, two systems were tested: one that simply searches for the word "science" in the book description and title, and another that uses the edismax query parser to exclude the books with fiction genres and to boost the non-fiction books.

Table 7: Search for scientific books

	System 1	System 2
Parameters	description:science title:science q.op:OR rows=10	q:science -genres:fiction-genres:"historical-fiction" defType:edismax qf:description title q.op:AND bq:genres:"non-fiction"^4 rows=10

The results were first evaluated by using a simple schema, which only uses the Lower Case filter:

Table 8: Search for Scientific books - results for simple schema

	System 1		System 2	
Rank	**Title**	**R**	**Title**	**R**
1	Science-Fiction 2007	N	Science Matters: Achieving Scientific Literacy	Y
2	Science Matters: Achieving Scientific Literacy	Y	The Science of Getting Rich	Y
3	A Reformed Approach to Science and Scripture	Y	A Reformed Approach to Science and Scripture	Y
4	The Science of Getting Rich	Y	Cure: A Journey Into the Science of Mind over Body	Y
5	Cure: A Journey Into the Science of Mind over Body	Y	Probability and Computing: Randomized Algorithms and Probabilistic Analysis	Y
6	Lightspeed Magazine, October 2012	N		
7	All Judgment Fled	N		
8	L. Ron Hubbard Presents Writers of the Future 27	N		
9	The Affair	N		
10	Probability and Computing: Randomized Algorithms and Probabilistic Analysis	Y		
AP	0.543		1.0	
AP@5	0.8		1.0	
Recall	0.4545		0.4545	

By analysing the results, we can verify that the average precision and the precision improved significantly as the retrieved results in the enhanced system will not be considered science fiction books.

Then, the systems were evaluated when using the *Synonym Graph* and *English Minimal Stem* filters:

Table 9: Search for Scientific books - results for the enhanced schema

	System 3		System 4	
Rank	**Title**	**R**	**Title**	**R**
1	Science-Fiction 2007	N	Science Matters: Achieving Scientific Literacy	Y
2	Science Matters: Achieving Scientific Literacy	Y	The Science of Getting Rich	Y
3	The Science of Getting Rich	Y	A Reformed Approach to Science and Scripture	Y
4	A Reformed Approach to Science and Scripture	Y	Cure: A Journey Into the Science of Mind over Body	Y
5	Cure: A Journey Into the Science of Mind over Body	Y	Encounters: Two Studies in the Sociology of Interaction	Y
6	Encounters: Two Studies in the Sociology of Interaction	Y	Probability and Computing: Randomized Algorithms and Probabilistic Analysis	Y
7	Probability and Computing: Randomized Algorithms and Probabilistic Analysis	Y	Real Change: Conversion	Y
8	The Affair	N	Literary Theory: The Basics	Y
9	Lightspeed Magazine, October 2012	N	Topoi: The Categorial Analysis of Logic	Y
10	Math for Smarty Pants	Y	Math for Smarty Pants	Y
AP	0.652381		1.0	
AP@5	0.8		1.0	
Recall	0.636364		0.91	

The results translate the capability of the filters to improve the relevance of the results in both systems by matching books that include other words rather "science", such as "mathematics" or "mathematic".

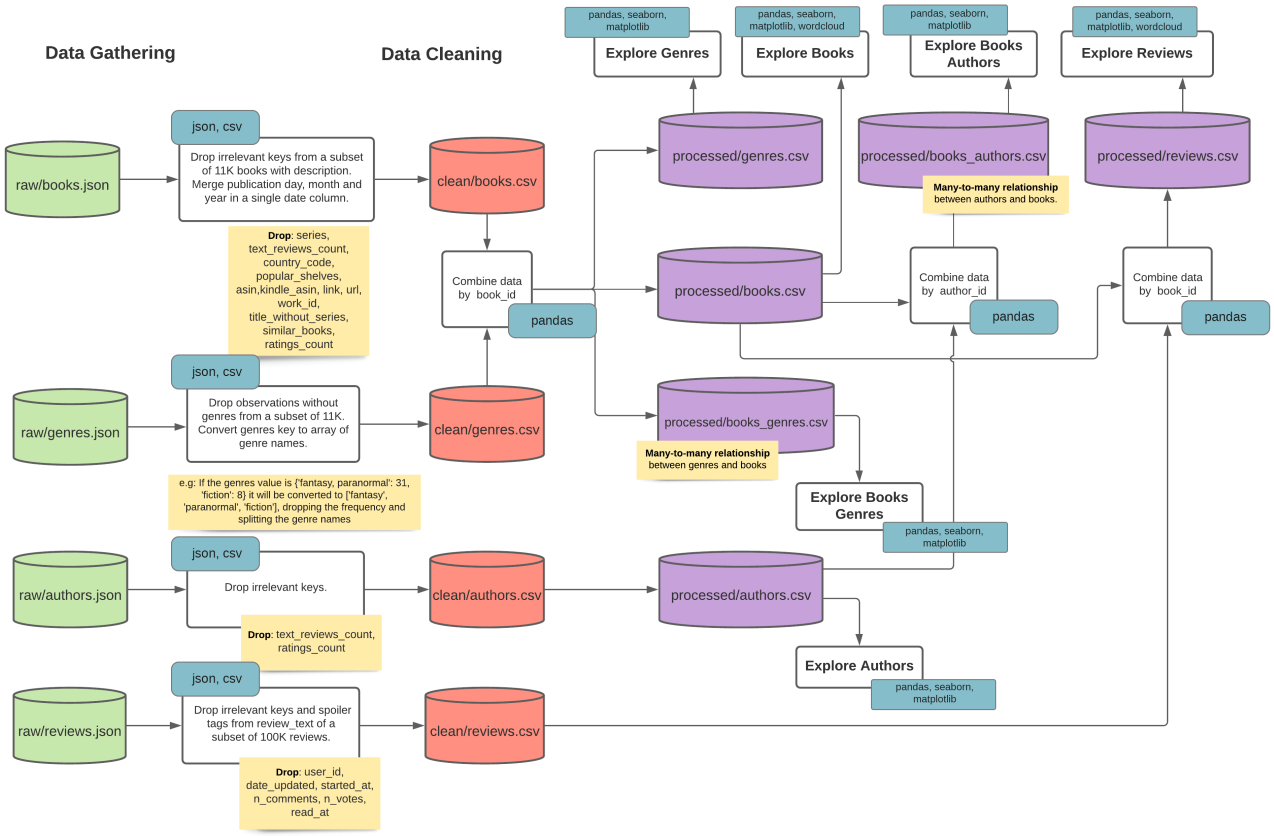
4 CONCLUSION

This paper covered the development of an initial prototype for the books and reviews search system. It first addressed the work developed in the first milestone, Data Preparation, which included the collection, refinement and exploration of the selected datasets and also the definition of the conceptual domain. The result of this milestone was a subset of the original data, free of irrelevant attributes and that clearly represented the main entities of the Goodreads domain - Books, Authors, Reviews and Genres; as well as the relations between them. It also uncovered possible information needs, which were then answered in the following milestone. In the second milestone, Information Retrieval, two documents were created and indexed: one for the books and another for the reviews. Then, Solr search platform was used to query the documents regarding the defined information needs: search for books about a specific thematic, named entities and historical moments, filter positive and negative reviews, and the ability to correctly discriminate scientific books. The results obtained, which were then evaluated using adequate metrics, prove that the choice of index and query filters was appropriate and also that the use of boosts and other improvements in the queries lead to an improvement of the search quality. Even though the results were as expected, some limitations that were already anticipated were now confirmed regarding the use of synonyms for Sentiment Analyses in the reviews: using synonyms is not enough to correctly distinguish all positive and negative reviews. For example, a positive review that also uses negative words to express an opinion about something else that is not related to the book would be mistakenly classified as negative. Therefore, for the next milestone, we plan to use *NLTK* for natural language processing. Also, the synonyms will be generated automatically with a script, and we will experiment with multi-word synonyms. New queries that use the release date of the books and the publication date of the reviews will also be tested. Furthermore, we are considering the possibility of completing the current data with recent books and reviews. Finally, we plan to develop a user-friendly graphic interface for the search system.

REFERENCES

[1] Acessed at December 2021. Elastic Stack. https://www.elastic.co/elastic-stack/
[2] Acessed at December 2021. Named Entity. https://en.wikipedia.org/wiki/Named_entity
[3] Big Data. Acessed at November 2021. https://pt.wikipedia.org/wiki/Big_data
[4] Field Type Definitions and Properties. Acessed at December 2021. https://solr.apache.org/guide/8_11/field-type-definitions-and-properties.html
[5] Synonym Filter Factory. Acessed at December 2021. https://cwiki.apache.org/confluence/display/solr/AnalyzersTokenizersTokenFilters#AnalyzersTokenizersTokenFilters-solr.SynonymFilterFactory
[6] Goodreads. Acessed at December 2021. https://www.goodreads.com/
[7] UCSD Book Graph. Acessed at October 2021. https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/books?authuser=
[8] json JSON encoder and decoder Python documentation. Acessed at November 2021. https://docs.python.org/3/library/json.html/
[9] Joe McCormack. 2016. Information overload science behind bad decisions. https://thebrieflab.com/blog/information-overload-science-behind-bad-decisions/
[10] Sérgio Nunes. 2006. State of the Art in Web Information Retrieval. (June 2006). https://web.fe.up.pt/~ssn/2005/prodei/soa-webir.pdf
[11] Goodreads Terms of User. Acessed at November 2021. https://www.goodreads.com/about/terms
[12] Information overload. Acessed at November 2021. https://en.wikipedia.org/wiki/Information_overload
[13] Analysis Phases. Acessed at December 2021. https://solr.apache.org/guide/8_11/analyzers.html#analysis-phases
[14] Elastic Search. Acessed at December 2021. https://www.elastic.co/
[15] Apache Solr. Acessed at December 2021. https://solr.apache.org/
[16] Mengting Wan and Julian McAuley. 2018. Item Recommendation on Monotonic Behavior Chains. (2018). https://cseweb.ucsd.edu/~jmcauley/pdfs/recsys18b.pdf
[17] Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. 2019. Fine-Grained Spoiler Detection from Large-Scale Review Corpora. (July 2019), 2605–2610. https://aclanthology.org/P19-1248

A PIPELINE FLOW



B INFORMATION RETRIEVAL ANALYSIS

Table 10: Search for positive reviews

	System 1	System 2
Parameters	q= {!q.op=OR df=review_text}in love with this book rows=10	q= {!q.op=OR df=review_text}in love with this book AND rating:[3 TO *] sort=field(rating, min) desc rows=10

Table 11: Positive reviews search results

	System 1		System 2	
Rank	**Title**	**R**	**Title**	**R**
1	The Boyfriend Mandate (The Boyfriend Chronicles, #2)	Y	By Any Other Name (Forbidden, #1)	Y
2	Taunting Destiny (The Fae Chronicles, #2)	Y	Taunting Destiny (The Fae Chronicles, #2)	Y
3	A Love Letter to Whiskey	Y	The Name of the Wind (The Kingkiller Chronicle, #1)	Y
4	On Stranger Tides	N	A Love Letter to Whiskey	Y
5	The Collector (Dante Walker, #1)	N	The Boyfriend Mandate (The Boyfriend Chronicles, #2)	Y
6	The Paris Wife	Y	Taunting Destiny (The Fae Chronicles, #2)	Y
7	The Name of the Wind (The Kingkiller Chronicle, #1)	Y	Solaris: The Definitive Edition	Y
8	I Am the Messenger	N	Born a Crime: Stories From a South African Childhood	Y
9	Taunting Destiny (The Fae Chronicles, #2)	Y	The Paris Wife	Y
10	Solaris: The Definitive Edition	Y	The Six Rules of Maybe	N
AP	0.7722		0.99	
AP@5	0.6		1.0	
Recall	0.7		0.9	

C PRECISION RECALL PLOTS

C.1 Books about romantic tragedy

Figure 5: System 1
Precision recall without boost

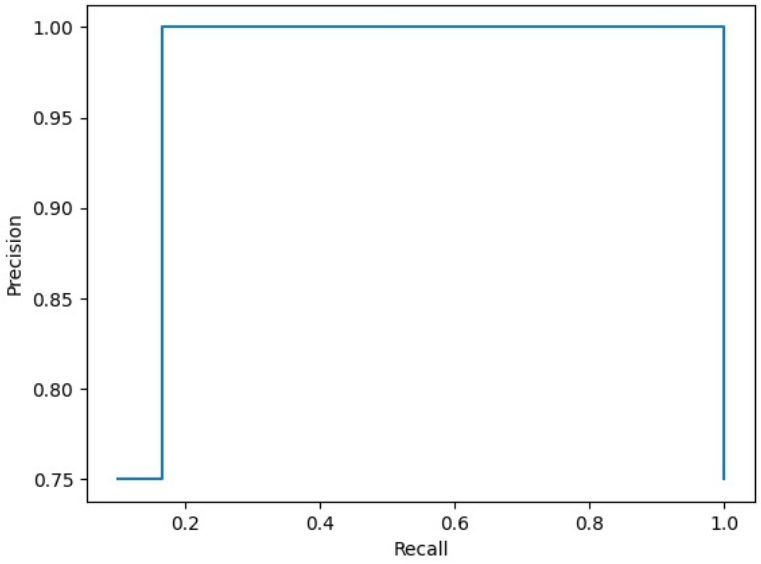
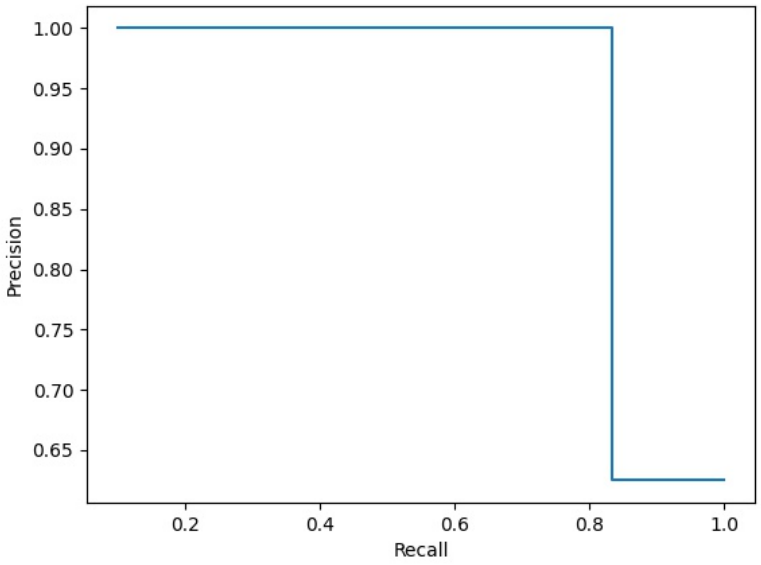


Figure 6: System 2
Precision recall with boost



C.2 World War books

Figure 7: System 1
Precision recall without boost

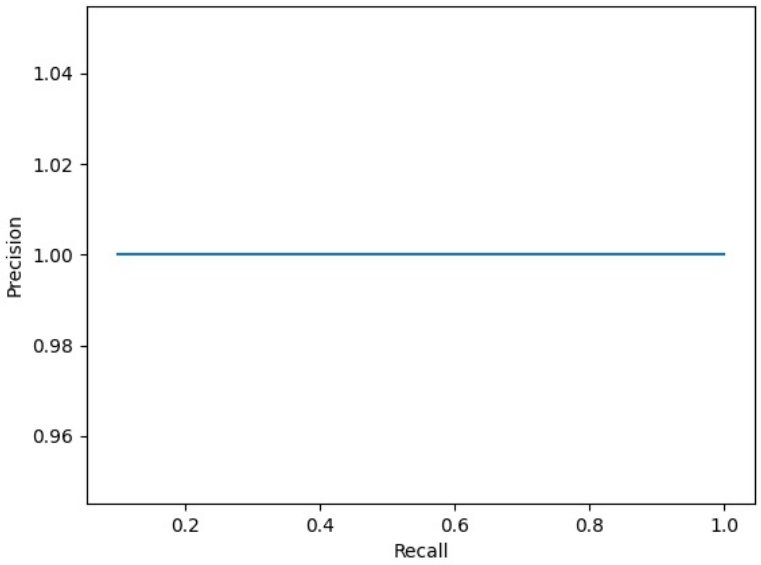
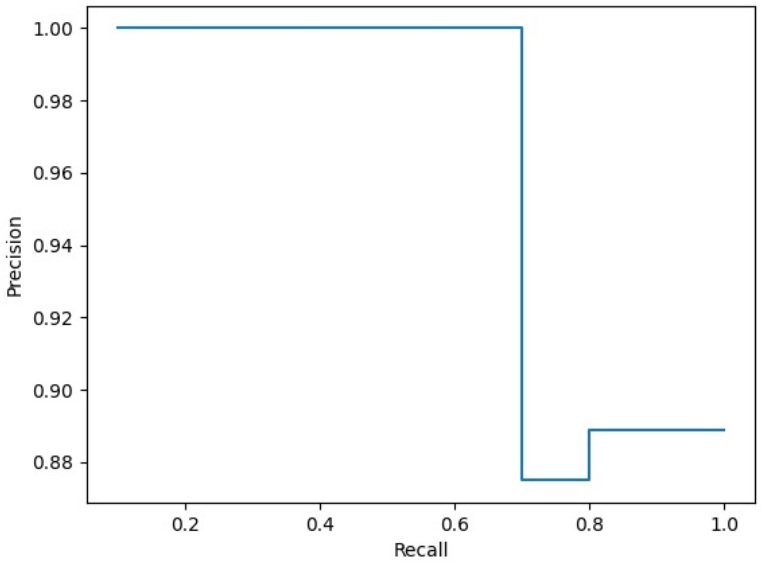


Figure 8: System 2
Precision recall with boost



C.3 Negative and Positive Reviews

Figure 9: System 1
Precision recall rating on negative reviews

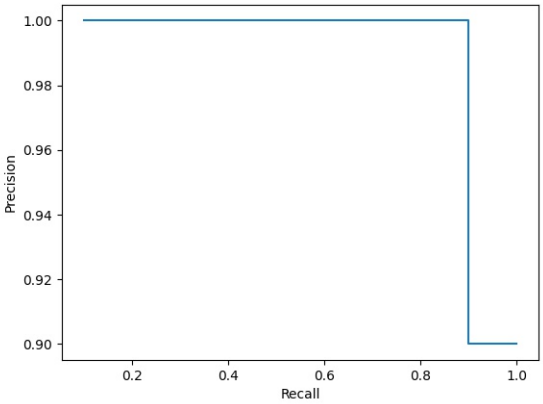


Figure 10: System 2
Precision recall rating with synonyms on negative reviews

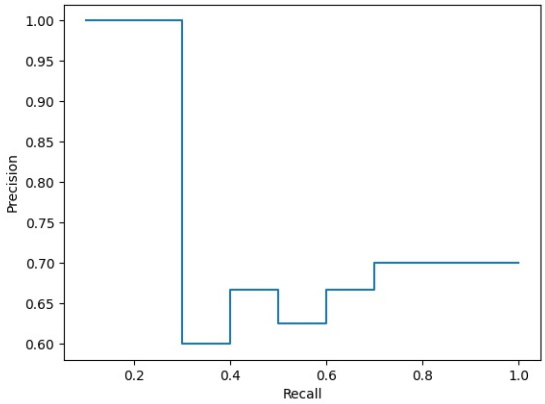


Figure 11: Precision recall on positive reviews

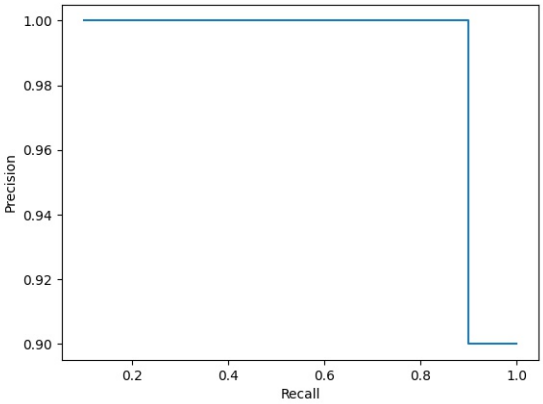
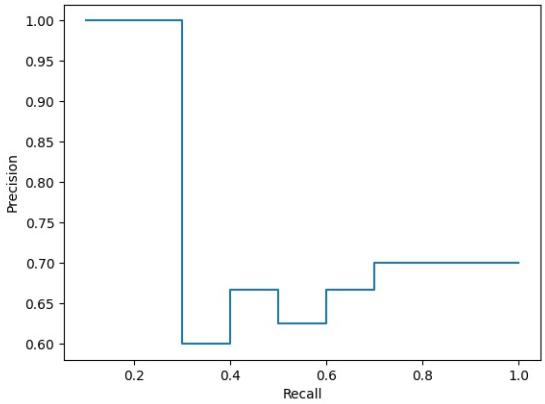


Figure 12: Precision recall with synonyms on positive reviews



C.4 Scientific books

Figure 13: System 1
Precision recall without boost [no filters]

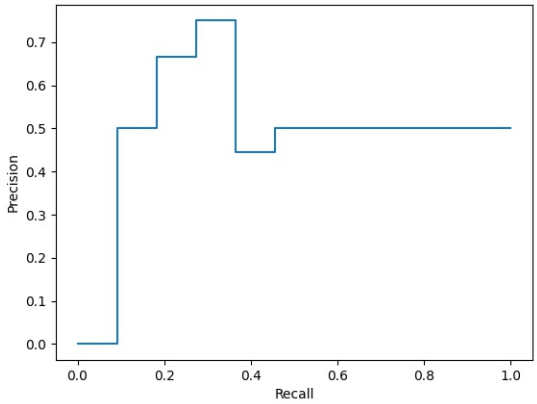


Figure 14: System 2
Precision recall with boost [no filters]

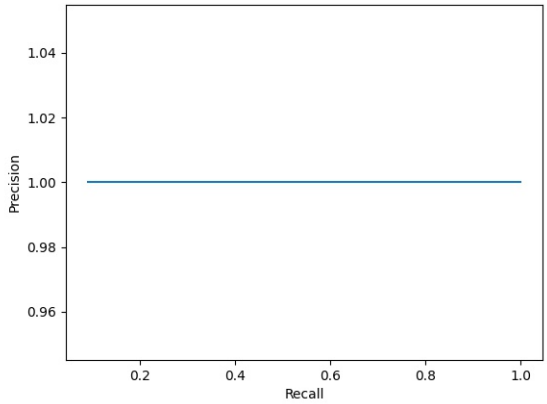


Figure 15: System 3
Precision recall without boost [with filters]

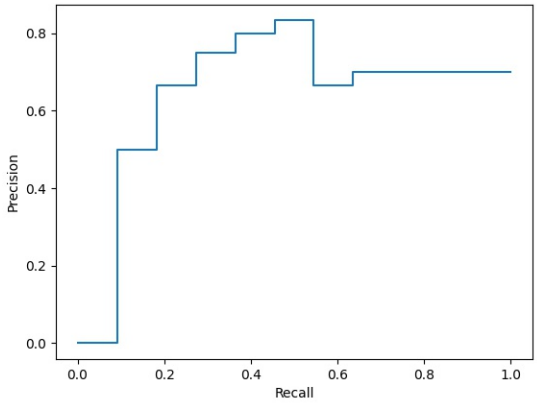


Figure 16: System 4
Precision recall with boost [with filters]