Information Processing and Retrieval Project

Beatriz Aguiar, João Marinho, Margarida Vieira*

Abstract

This report explores the creation of a search engine. It is divided into two main sections, each representing a project's milestone. Every procedure is carefully and minutely described. Data preparation is the initial and fundamental phase. Its focus is on collecting data, processing it, and conducting an exploratory analysis for further understanding of the convenient search tasks.

Keywords: datasets, data preparation, data analysis, information retrieval, search system

ACM Reference Format:
Beatriz Aguiar, João Marinho, Margarida Vieira. 2018. Information Processing and Retrieval Project. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Searching for information on the Web is, in the modern era, of utter importance. ICT (Information and Communications Technology) is a broad subject and the concepts are evolving. It covers any product that will store, retrieve, manipulate, transmit, or receive information electronically in a digital form (e.g., personal computers including smartphones, digital television, email, or robots) [2]. Therefore, improving and optimizing the process of finding information is one of the primary focuses of those who work in the field of Data Processing & Information Retrieval.

2 Data Preparation

The goal of the first task is to prepare and explore datasets, which may suffer from extraction actions such as crawling or scraping.

*Beatriz Aguiar, up201906230@up.pt; João Marinho: up201905952@up.pt; Margarida Vieira, up201907907@up.pt.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

2.1 Data Selection

The provided guidelines shepherd the pursuit of rich and ambiguous datasets that granted both information quantity (in the thousands range) and quality. Thus, movies came across as the ideal theme of choice.

Google Dataset Search was our first approach to the search for datasets, however, after a generic analysis, we considered the available data to be quite simple, containing only one table, while lacking textual features.

Given IMDb's high demand and popularity, it seemed to be the ideal dataset source. Its API did not only provide 7 TSV files, but it also granted reliable and updated content, with each table containing different media-related data up to millions of entries. Such information covered most of the common movie attributes (e.g. duration, genre), titles, directors, actors, and ratings.

Subsets of IMDb data are available for access to customers for personal and non-commercial use. You can hold local copies of this data while being subject to their terms and conditions. Information courtesy of IMDb [1]. Used with permission.

2.2 Data Processing

As demonstrated in figure 1, the original datasets underwent through a thorough processing, following an ETL (extract-transform-load) like pattern. To collect the data, the makefile starts by running a series of commands that download the zip files from the IMDb API and unzip the latter to the data folder.

An initial exploratory analysis was conducted on each dataset to understand its relevance and features importance. The *title_akas.tsv* dataset, for e.g., turned out to be uninteresting as the only feature we were interested in, the language of the movie, had 6 not-null values in a 33M entries dataset. On top of that, we concluded that the datasets regarding the principal crew members and episodes of the streaming content were also irrelevant to our search engine.

In terms of data preparation, we started by filtering the dataset *title_basics.tsv*, leaving only entries corresponding to movies released after 1990. We opted to do this not only for efficiency purposes, by reducing the total number of entries, but also due to the minimal value it added to the dataset - fewer individuals will likely search for older data.

However, the dataset still comprised hundreds of thousands of entries, which was far more data than necessary and would increase web-scraping latency. For this reason, we decided to discard movies with average ratings below 7, always aiming to keep the most relevant data possible.

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

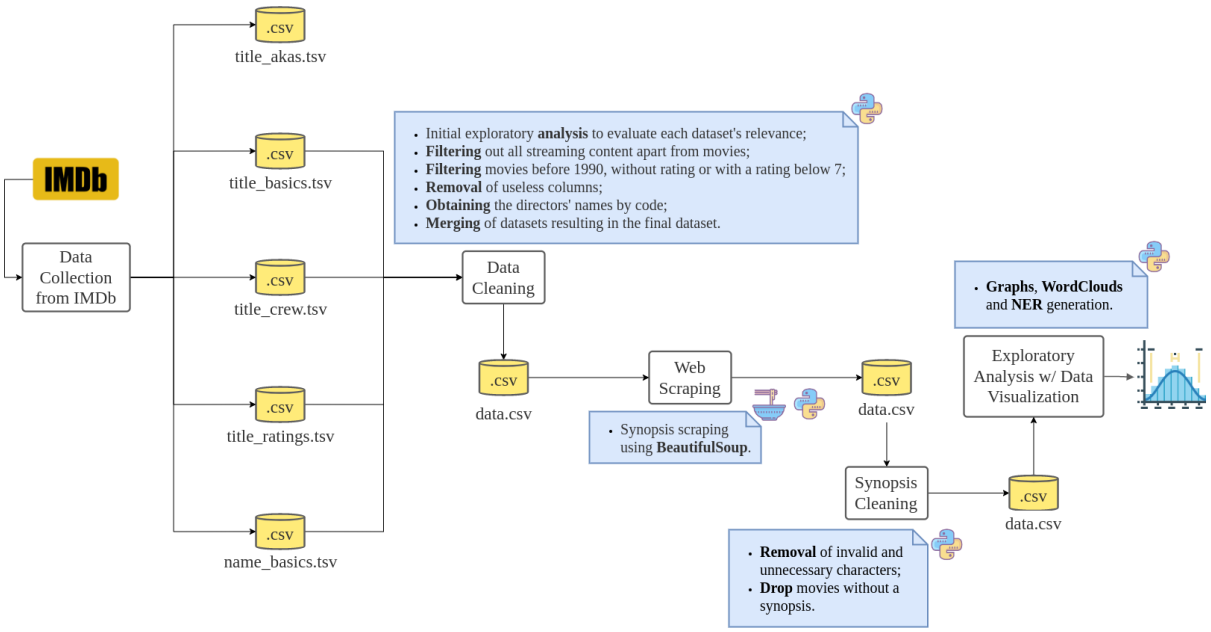

Figure 1. Pipeline

The final dataset now contained around seventy thousand entries, a considerably good dataset dimension.

Posteriorly, we moved on to merge the other datasets with the main movie dataset.

At this point, our dataset's features consisted in an id - *tconst*, the original movie name - *originalTitle*, a boolean indicating whether it is an adult movie or not - *isAdult*, the release year - *startYear*, the duration of the movie - *runtimeMinutes*, its genres, the average rating, the number of votes, and, finally, the directors, a string of ids referring to the crew's dataset.

The next step was to match the directors' string of ids to the corresponding real names and convert it to a string of names.

For the most important movie's piece of information, the synopsis, we performed web-scraping on both Wikipedia and IMDb's pages, using Python's *BeautifulSoup* package. This textual feature did not, however, suffer from any processing other than cleaning up irrelevant characters.

Finally, we chose to standardize the features' names to enhance the posterior visual analysis carried out - *startYear* changed to *year*, *runtimeMinutes* to *duration*, and *averageRating* to *avgRating*.

2.3 Conceptual Model

2.4 Characterization

To better understand the data in our hands, we developed a Python Notebook to plot general information about the dataset. The more information we have, the better the decisions will be made throughout the development of the project.

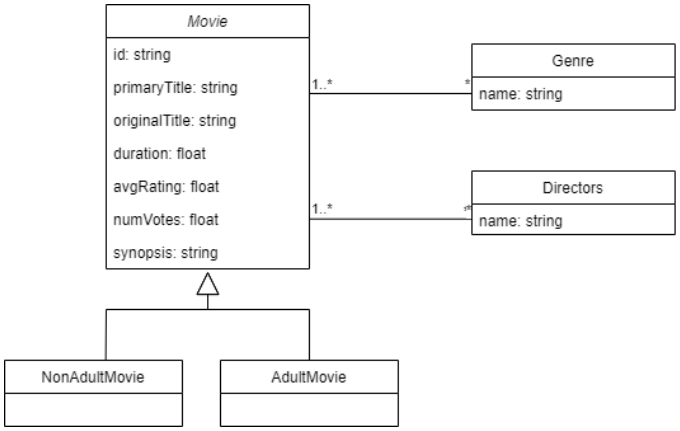

Figure 2. Conceptual Model

By taking a look at the obtained graphics, some conclusions can be easily drawn.

First, the number of movies released per year (figure 3) has increased in the last decade, meaning that more information can be drawn from more recent years. This increase is probably due to the latest technological advances.

The second plot (figure 4) shows that only a small minority of films were able to get ratings above 8, with most films scoring between 7 and 8. However, it is important to keep in mind that the final dataset did not include any films with an average rating of less than 7.

When it comes to genres (figure 5), Documentary and Drama seem to be the most popular ones by an obvious

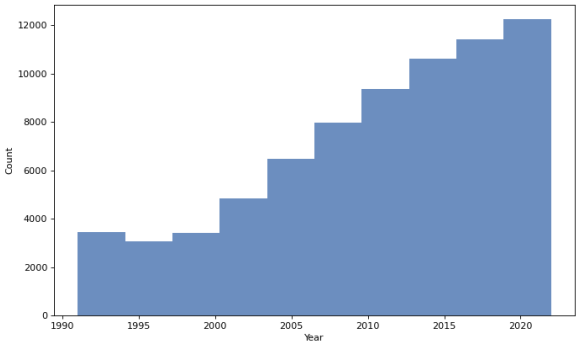

Figure 3. Number of movies per year

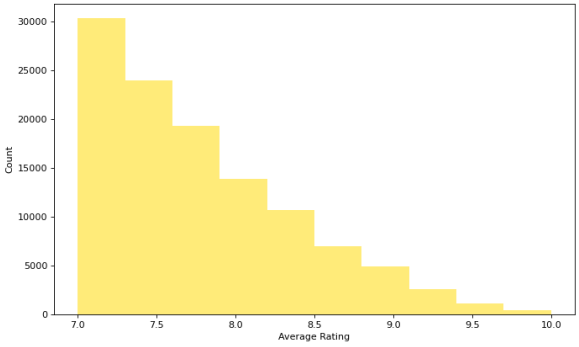

Figure 4. Number of movies per rating

margin, with Comedy, Biography, and Romance completing the top 5.

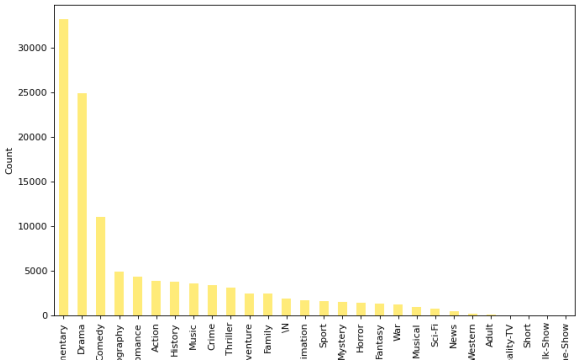

Figure 5. Number of movies per genre

Since the average rating for practically every genre falls within the same range, no meaningful conclusions can be drawn about the average rating per genre (figure 6).

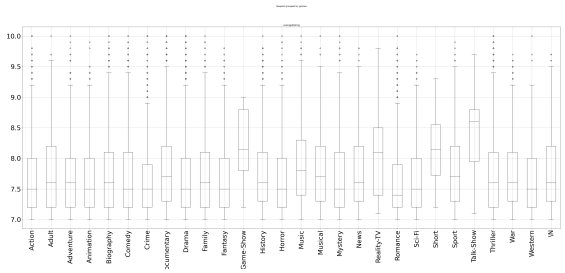

Figure 6. Average rating per genre

The following plot, *Rating per number of votes* (figure 7), demonstrates that just a minor number of individuals give a rating above 8, leading us to believe that ratings above this threshold are simply the consequence of fewer votes when compared to ratings between 7 and 8, which usually receive the majority of votes.

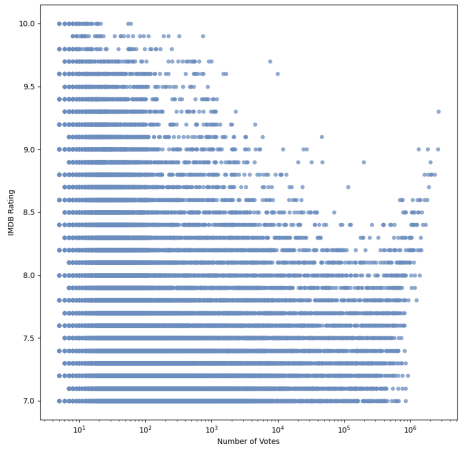

Figure 7. Rating per number of votes

We also decided to plot the number of adult vs. non-adult movies (figure 8).

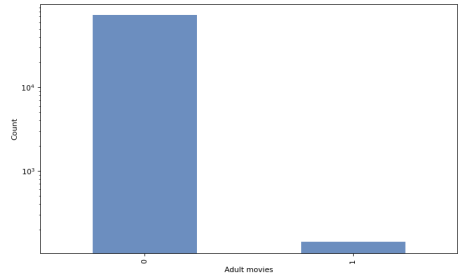

Figure 8. Number of movies per adultness

Finally, we plotted a WordCloud for 3 of the top 5 genres - Documentary, figure 9, Biography, figure 10, and Romance, figure 11.

Figure 9. Documentary movies' synopsis wordcloud

Figure 10. Biography movies' synopsis wordcloud

A further step towards information extraction was conducted, Named Entity Recognition (NER), in the newest feature attained, the synopsis. This process mainly tries to locate and classify entities such as characters' names and/or specific places. Figure 12 is an example of this, applied to a biography synopsis.

In addition to NER, it was also performed a Keyword extraction analysis. For the previous NER example, the resulting keyword extraction list was:

Figure 11. Romance movies' synopsis wordcloud

Figure 12. Example of a NER biography synopsis.

1. 'drama follows manoel oliveira'
2. 'follows manoel oliveira life'
3. 'oliveira life times dictatorship'
4. 'life times dictatorship portugal'
5. 'manoel oliveira life'
6. 'dictatorship portugal'

2.5 Prospective Search Tasks

- Family Movies
- Movies about the space
- Romance teen movies
- Movies to watch in your twenties
- Life-changing movies

3 Information Retrieval

Information Retrieval refers to the activities of obtaining information resources (usually in the form of textual documents) from a much larger collection, which are relevant to a user's information need, usually expressed as a query. It is the science of searching for information in a document, searching for documents themselves, and also searching for the metadata that describes data, and for databases of texts, images or sounds.

The goal of the second task is to implement and use an information retrieval tool - Solr - on the final dataset, assembled in the previous Section 2, and explore it with free-text queries.

3.1 Collection and Indexing

In the system, a movie defines a document. As there was no other type of documents, the queries were made upon a single collection. The recommended *Solr* [2] tool was utilized to achieve this.

Formerly, we described how the data sources were merged into a single database-like structure. However, some refinements could be made to improve our experience with the said tool. These included:

- Splitting the genres and directors fields into a list of strings.
- Converting the end-resulting CSV file to JSON format, allowing for easier use of multivalued fields.

Regarding the indexing process, two schemas were created, a basic one and an improved version. Both contain the same fields, and field types, differentiating the tokens and filters applied.

The first step was to determine which features would be relevant for the user to query. These features compromised the movie's primary and original titles, release year, genres, average rating, directors, and synopsis. Although not being indexed, all the other features are stored since they provide some other interesting aspects about movies (Table 1).

Field	Type	Indexed
tconst	string	No
primaryTitle	textualField	Yes
originalTitle	textualField	Yes
isAdult	bool	No
startYear	int	Yes
runtimeMinutes	float	No
genres	textualField	Yes
averageRating	float	Yes
numVotes	int	No
directors	nameField	Yes
synopsis	textualField	Yes

Table 1. Schema fields.

Secondly, we had to specify which tokenizer and filters would be applied at index and query time to the more rich textual fields (*textualField*). For the basic schema definition (Table 2), this more complex pre-processing employed the ***StandardTokenizerFactory*** class, and two filters:

- ***ASCIIFoldingFilterFactory*** which converts alphabetic, numeric, and symbolic Unicode characters which are not in the Basic Latin Unicode block to their ASCII equivalents, if one exists.
- ***LowerCaseFilterFactory*** which converts any uppercase letters in a token to the equivalent lowercase token.

[2](https://solr.apache.org/guide/8_1/)

Name	Tokenizer	Filters
textualField	StandardTokenizerFactory	ASCIIFoldingFilterFactory LowerCaseFilterFactory

Table 2. Basic schema custom field types.

While the enhanced schema relied on the ***LowerCaseTokenizerFactory***, which behaves much like the combination of the Standard Tokenizer and the Lowercase Filter, with the following filters:

- ***StopFilterFactory*** which stops (by discarding) the analysis of, tokens that are on the default stop words list.
- ***EnglishMinimalStemFilterFactory*** which stems plural English words to their singular form.
- ***EnglishPossessiveFilterFactory*** which removes singular possessives (trailing 's) from words.
- ***PorterStemFilterFactory*** which applies the Porter Stemming Algorithm for English.

The above-mentioned field types apply the same provisions at both the index and query time.

We have also used a custom type for the list of director names since it does not make sense to apply most of the filtering as in rich text features (Table 3).

Name	Tokenizer	Filters
textualField	LowerCaseTokenizerFactory	ASCIIFoldingFilterFactory StopFilterFactory EnglishMinimalStemFilterFactory EnglishPossessiveFilterFactory PorterStemFilterFactory
nameField	LowerCaseTokenizerFactory	ASCIIFoldingFilterFactory

Table 3. Enhanced schema custom field types.

3.2 Retrieval

After the indexing process is complete, the next step was to decide how to retrieve the information from the indexed documents.

Three different information needs were defined, based on the prospective search tasks previously defined in Subsection 2.5. For each information need, the motivation for the topic, a brief description, and the basic and boosted queries and correspondent parameters are provided.

3.2.1 Information Need 1 - Movies about the space.

With this information need, we intended to retrieve all movies about the space.

Considering the ambiguity of the word "space", the query was not as straightforward as it seemed when we first thought of it. There were other words that directly related more to the topic, so while we had to include the word "space" in the query, we couldn't give it too much relevance.

Basic Query

- **query**: space, astronaut, galaxy, planets
- **query fields**: originalTitle, primaryTitle, synopsis

Boosted Query

- **query**: space, astronaut, galaxy, planets
- **query fields**: originalTitle^1.5, primaryTitle^1.5, synopsis^2.0
- **boost query**: genres:sci-fi^3.0, synopsis:stars

3.2.2 Information Need 2 - Romance teen movies.

Since the Earth started spinning, teenagers have been falling in love. That sweet puppy love that's nectarous as divine mana and absurdly scary at the same time. And it's completely irrelevant if one's succumbing to the syrupy dreams of love right now or has experienced it decades earlier, it is always nice to either relate to the experience or reminisce on the infatuation from years ago. The easiest way to do that is by watching one of those teen romance movies, which was exactly what we intended to find with this information need.

After some research on how movies of this type are often described and referred to on the internet, we came up with a set of keywords to compose the query. As for genres, despite many of these movies being characterized as *romcoms* (romantic comedies), we chose to give more importance to romances and dramas, in the boosted query.

Basic Query

- **query**: romance, teen, crush, heart-break, in love, high-school, college, friends, friendship, campus, gossip, passion, attraction
- **query fields**: originalTitle, primaryTitle, synopsis

Boosted Query

- **query**: romance, teen^3.0, crush, heart-break^3.0, in love^2.0, high-school^2.0, college, friends, friendship, campus, gossip, passion, attraction
- **query fields**: originalTitle^1.5, primaryTitle^1.5, synopsis^4.0
- **boost query**: genres:drama, genres:romance, synopsis: crush^2.0

3.2.3 Information Need 3 - Movies to watch during Christmas Eve.

The best memories from our childhood Christmases (aside from the toys under the tree) are probably wrapped up in the family coming together to watch some classic Christmas movies. There's just something about Christmas movies that bring out the best in people. So while preparing for another holiday filled with joy, with this information need, we intended to find just the perfect movies to watch during that season.

The strategy used to put together this query, especially its keywords, was fairly similar to the one used for the previous query. However, in this case, and for what we think to be obvious reasons, we chose to boost documents where the term "christmas" was featured in the movie's title or synopsis.

Basic Query

- **query**: christmas^2.0, santa^2.0, snow, elf, rodolf, festive, claus, merry, holiday, candy cane, christmas tree, eve, advent
- **query fields**: originalTitle, primaryTitle, synopsis

Boosted Query

- **query**: christmas^2.0, santa^2.0, snow, elf, rodolf, festive, claus, merry, holiday, candy cane, christmas tree, eve, advent
- **query fields**: originalTitle^1.5, primaryTitle^1.5, synopsis^2.0
- **boost query**: primaryTitle: christmas^5.0, synopsis:christmas^3.0

3.3 Evaluation

It is of utter importance to evaluate the results obtained systematically, to obtain a more objective picture of the performance of our system, as well as compare different versions of the same base system. That said, we will be comparing basic and boosted query results with a simpler and enhanced schema version.

Evaluation measures provide a way of quantifying retrieval effectiveness. Individual metrics are prone to bias and give a tunnelled vision of the system. Therefore, it is important to always evaluate over a set of distinct metrics.

Precision and Recall are two well-known measurements and, when calculated globally, ignore the ranking of the relevant documents returned. Precision measures the fraction of relevant retrieved documents, i.e., it is the ratio between the number of relevant documents and the total number of documents retrieved. Recall, on the other hand, measures the fraction of retrieved relevant documents, i.e., what proportion of all the relevant documents has been retrieved.

Precision and Recall tend to vary inversely: higher precision is often coupled with lower recall. Therefore, they are often presented together, since it's also dependent on the system's goal.

As the evaluation method, we opted for the precision-recall curve, a plot that traces the evolution of the system's precision as recall increases. This allows us to observe the trade-off between the metrics at certain thresholds. The area under the curve is considered a measurement of the system's quality: a bigger area means that the system was able to keep a higher overall precision.

Other metrics used are the Precision at 10 (P@10) and Average Precision (AP). The former calculates Precision at a specific cut point in the result-ranked list. On the other hand, the Mean Average Precision is the result of the average of another metric applied at query level, Average Precision (AP), a metric regarded as very stable.

The evaluation approach used relies on test collections, i.e., a set of resources that includes the corpora indexed by

the system, and a matrix of relevance judgements between both. Given the extensive dataset and the need to extract relevant movies for each query, resulting in an extremely time-consuming process, a sample dataset of 300 documents was carefully selected alongside a list of contained relevant movies, solely for evaluation purposes. All information needs were obtained and evaluated using this sample subset.

By incorporating the previously mentioned evaluation metrics over the results obtained, we were able to compare the different retrieval strategies and algorithms systematically.

3.3.1 Information Need 1 - Movies about the space.

The evaluation results can be seen in figures 13 and 14 and tables 4 and 5.

The results were very satisfactory, having an Average Precision always above 0.95 and a Recall and Precision at 10 of 1.0. Our top results were relevant and all the relevant documents were retrieved.

The boosted query did improve the results when compared to the basic one. Despite the latter having a very high Average Precision, the applied boosts proved to be effective but not in a very noticeable way.

When comparing the results of the basic schema (figure 13 and table 4) to the enhanced schema (figure 14 and table 5), we found it surprising that the enhanced schema showed a worse Average Precision.

After manually analysing the results, we concluded that this deterioration was due to the application of the *EnglishMinimalStemFilterFactory* in the enhanced schema. For example, an irrelevant movie was retrieved because the 'spaces' word, contained in the synopsis ("...Santa can't move quickly or through tight spaces like he does, so the kids are forced to help him..."), was converted to 'space' and, therefore, increased the document's score.

Nevertheless, we observed that the order of the results was subjectively better when using the enhanced schema and the irrelevant documents retrieved had the lowest score. This may allow for future filtering according to the score given by the used tool, with the imposition of a lower bound for the said attribute.

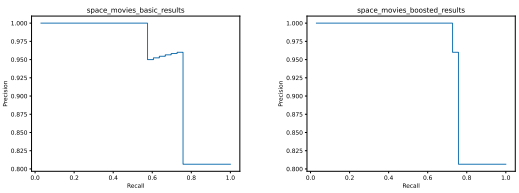

Figure 13. Space Movies

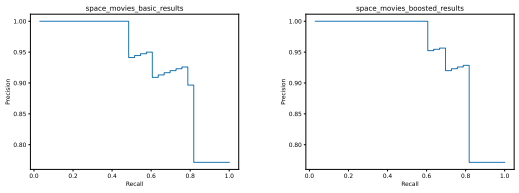

Figure 14. Space Movies Enhanced Schema

Metric	Basic	Boosted
Average Precision	0.973584	0.981191
Precision at 10 (P@10)	1.0	1.0

Table 4. AP and P10 metrics for basic schema.

Metric	Basic	Boosted
Average Precision	0.949862	0.9611
Precision at 10 (P@10)	1.0	1.0

Table 5. AP and P10 metrics for enhanced schema.

3.3.2 Information Need 2 - Romance teen movies.

The evaluation results can be seen in figures 15 and 16 and tables 6 and 7.

In this information need, the improvements of the schema are more visible and the boosting raises the Precision significantly, reaching an Average Precision of 1.0 with the enhanced schema. These results are extremely good, since, for our best environment, solr retrieves only relevant documents, which is ideal in a search engine.

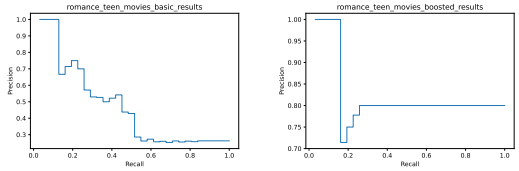

Figure 15. Romance Teen Movies

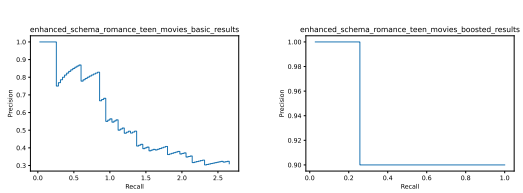

Figure 16. Romance Teen Movies Enhanced Schema

Metric	Basic	Boosted
Average Precision	0.411838	0.897266
Precision at 10 (P@10)	0.7	0.8

Table 6. AP and P10 metrics for basic schema.

Metric	Basic	Boosted
Average Precision	0.46774	1.0
Precision at 10 (P@10)	0.9	0.9

Table 7. AP and P10 metrics for enhanced schema.

3.3.3 Information Need 3 - Movies to watch during Christmas Eve.

The evaluation results can be seen in figures 17 and 18 and tables 8 and 9.

In this information need, neither the boosting nor the schema have a major impact on the Precision. However, there is an evident improvement when evaluating the metric *Precision at 10*.

This means our boosted query, besides from retrieving more relevant documents, scores them higher than non-relevant ones. In the basic query, for example, 60% of irrelevant documents are in the top 10 results. We found this improvement to be very useful due to the importance of top results in a search engine.

One of the other top search engines after Google, Bing, reports that websites on the top get 42% of the traffic; the second gets 11% and the third only gets 8%. So, it clearly indicates why your website needs to be on top. [1]

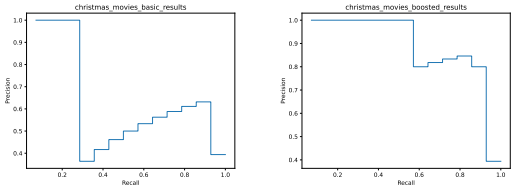

Figure 17. Christmas Movies

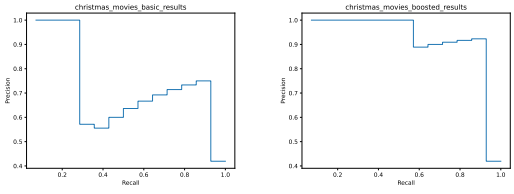

Figure 18. Christmas Movies Enhanced Schema

Metric	Basic	Boosted
Average Precision	0.58253	0.733349
Precision at 10 (P@10)	0.4	0.8

Table 8. AP and P10 metrics for basic schema.

Metric	Basic	Boosted
Average Precision	0.665005	0.769875
Precision at 10 (P@10)	0.8	1.0

Table 9. AP and P10 metrics for enhanced schema.

Overall, we can conclude that both the boosted queries and the enhanced schema perform as desired, by pushing the most relevant documents to the top and by retrieving fewer irrelevant documents.

4 Conclusion

This paper describes, in detail, the process of creating an IMDb movies search engine. The first part consisted of the data selection and processing, the whole procedure is represented as a pipeline in figure 1. Its importance resides in creating a final dataset, cleaned and properly analysed, to be used as the main data source of the final search engine. In the second part, we implemented and used the information retrieval tool, Solr, with the generated dataset. Regarding the retrieval process, two schemas for indexing the dataset were created and two queries, for each information need, were constructed. The implemented system was evaluated by comparing the use of each schema in both queries. In future work, making use of features and techniques to improve the quality of the search results will be developed.

References

[1] Ramona Sukhraj. 2017. How Important Is a Top Listing In Google? (And Why?). https://www.impactplus.com/blog/important-top-listing-google [Online; accessed 16-November-2022].

[2] Wikipedia. 2022. Information and communications technology — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Information%20and%20communications%20technology&oldid=1113616939. [Online; accessed 06-October-2022].