# Information Processing and Retrieval

## GOODREADS' BOOKS AND REVIEWS

---

Diogo Almeida (up201806630)
Pedro Queirós (up201806329)

# Milestone 2 Overview

- 4 information needs were defined with their respective queries;
- 3 systems were developed:
  - Without schema;
  - With schema;
  - With schema and weights.
- The performance of each system was measured using P@10 and AvP metrics;
- The system with the best performance was the third system, that used both schema and weights.

# Improvements introduced

- More **complex synonyms** for query expansion (a basic implementation of the synonyms was used previously for one of the information needs);
- **OpenNLP toolkit**:
  - Tokenizers
    - OpenNLPTokenizerFactory
  - Filters
    - OpenNLPPOSFilterFactory
    - OpenNLPChunkerFilterFactory
  - Named entity recognition
    - People
    - Locations
    - Time

# Improvements introduced

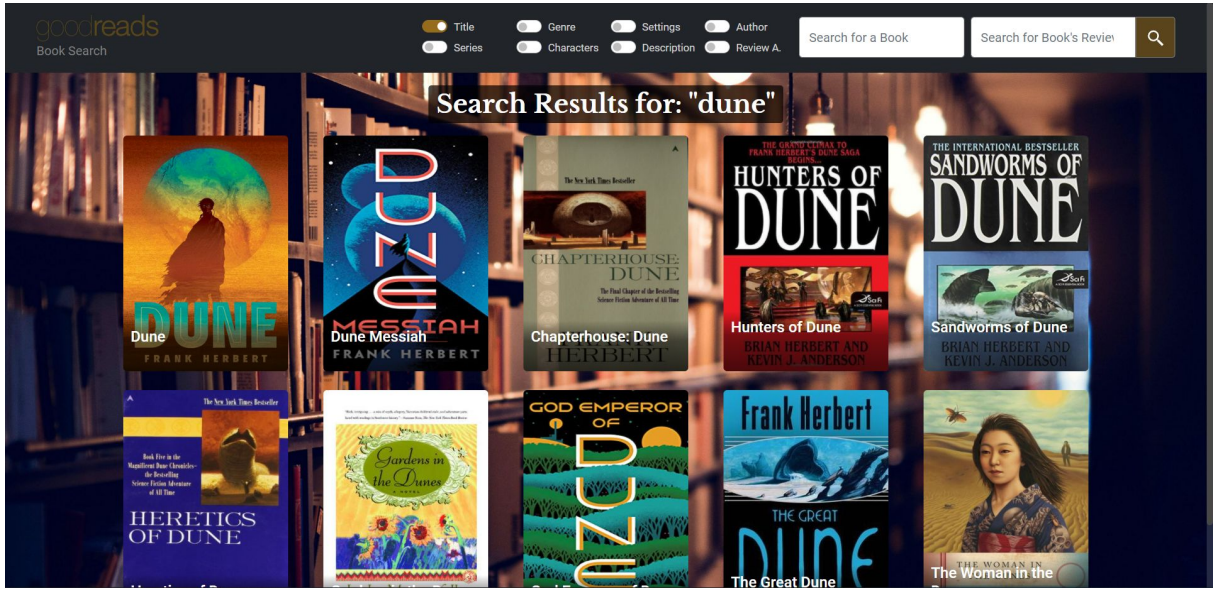
- **Learning to Rank (LTR)**:
  - Model selection: **Linear Model** - computes scores by using the dot product between the selected features and their respective weights.
  - Feature selection:
    - Original score;
    - Average rating;
    - Number of ratings;
    - Description;
    - Number of times recommended, by adding a new attribute to the books - Basic Page Rank implementation.

A SVM model (*SVMRank*) was used, in Python, to predict the best weights for the Linear Model. The training data consisted of the previously used 4 Information Needs.

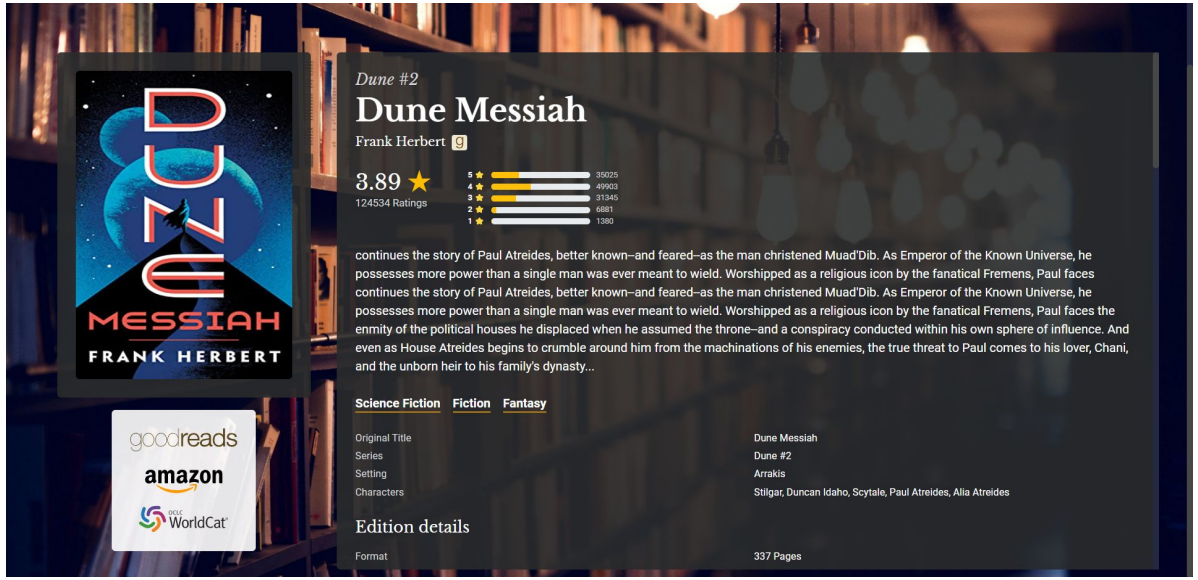
# Improvements introduced

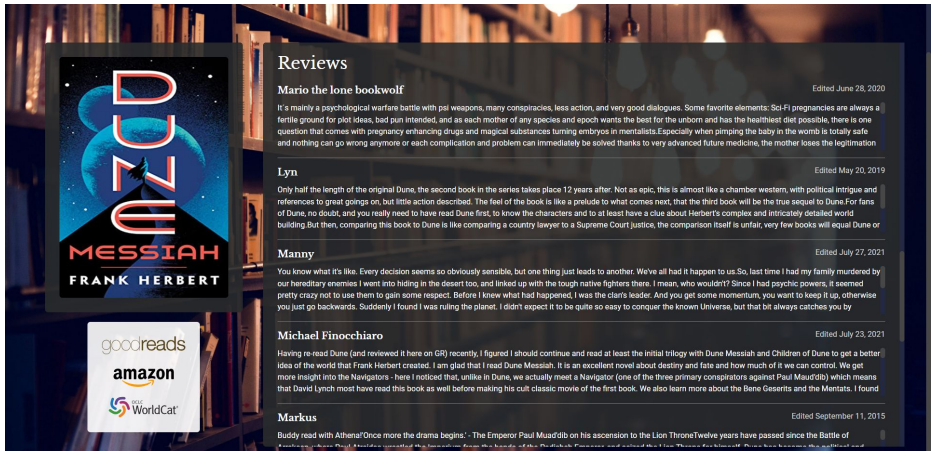
- **Multiple Core Search**
  - Added another core for reviews;
  - Now allows users to search by more review parameters - author and date.
- **Graphical User Interface**
  - The graphical interface allows the user to search in several different fields in books, e.g. , title or description, and also in the books' reviews and its authors;
  - Grants a simpler and better experience for users that don't have experience in query syntax.

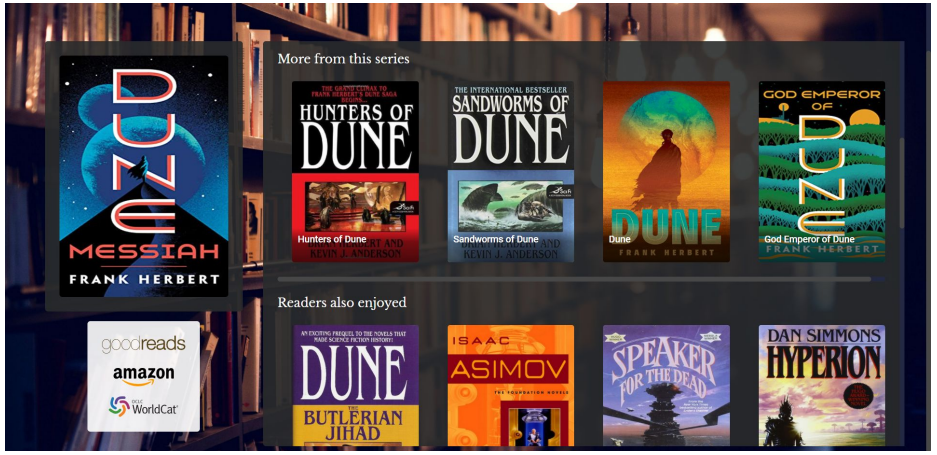
# Graphical Interface

# Graphical Interface



# Improved System Evaluation

After the implementation of the previously mentioned improvements, performance was measured using *P@10* (Precision at 10), *AvP* (Average Precision) and *MAP* (Mean Average Precision) metrics. These results were then compared with System 3's results, the configuration that used schema and weights and the one with the best performance in the previous milestone. Only 3 of the 4 previous information needs were used since one of them was very specific, not allowing a good comparison or extraction of conclusions.

# Information Need 1

**Information Need:** Biographies about Adolf Hitler

**Relevance Judgement: “**Biography” needs to be in the book’s genre, title or description. “Hitler” is also important to appear in the book’s title or author and can also appear in the description, although with less relevance.

**Query(q):** hitler **AND** biography

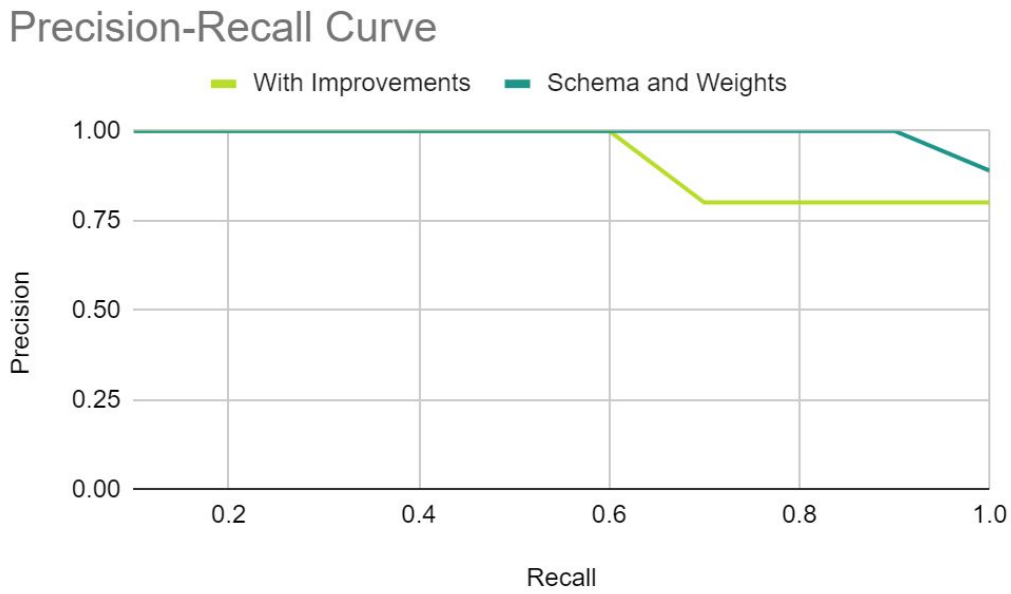
**Query fields(qf):** title genre_and_votes description author

**Boost Query(bq):** title^3 genre_and_votes^2 description^1 author^3

**Rerank Query(rq):** {!ltr model=my_efi_model efi.text="hitler biography"}

# Information Need 1 - Evaluation

| Rank | System 3 | Improved System |
|---|---|---|
| AVP | 0.986111 | 0.836376 |
| P@10 | 0.8 | 0.880258 |



# Information Need 3

**Information Need:** Non-fictional books about the life's work of Albert Einstein

**Relevance Judgement:** The scientist must be either the book's author or have his name in the title. He can also be part of the book's characters. His name can also appear in the description although it is less relevant. Science or Philosophy must be in the book's genre and can also appear in the description or the title.

**Query(q):** einstein **AND -**fiction **AND** (philosophy **OR** science)

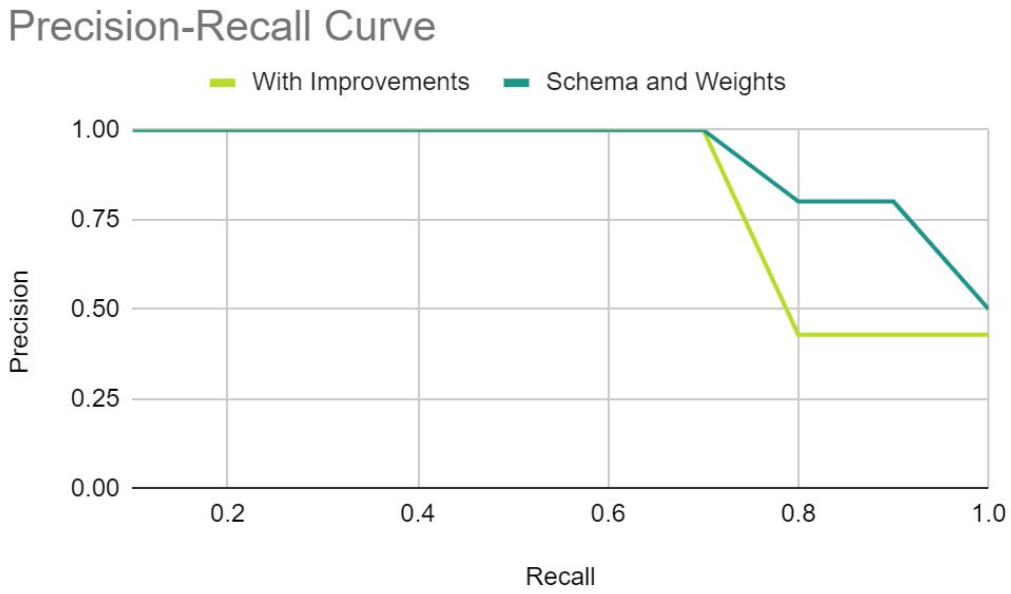
**Query fields(qf):** title genre_and_votes description characters

**Boost Query(bq):** title^3 genre_and_votes^2 description^1 characters^3 ner_person_field^1

**Rerank Query():** {!ltr model=my_efi_model efi.text="einstein science philosophy"}

# Information Need 3 - Evaluation

| Rank | System 3 | Improved System |
|---|---|---|
| AVP | 0.86 | 0.809524 |
| P@10 | 0.5 | 0.3 |



# Information Need 4

**Information Need:** Fantasy children's books set in the medieval era that are easy to read

**Relevance Judgement:** We are looking for books that are easy to read, something that is not in the books' informations. So it is important to search in the reviews of each book. The genres of the books must include fantasy and children's and the story should be set in a medieval era (should mention "kingdom")

**Query(q):** easy read

**Filter Query(fq):** genre_and_votes:fantasy

**Filter Query(fq):** genre_and_votes:childrens

**Filter Query(fq):** description:kingdom

**Query fields(qf):** reviews

**Phrase fields(pf):** reviews^5

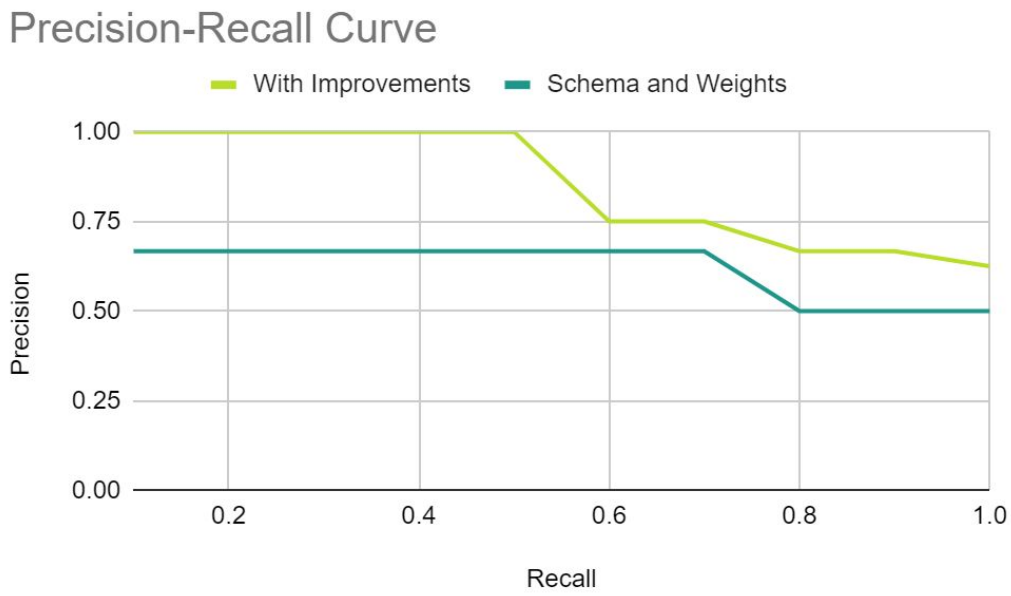
**Phrase slop(ps):** 5

**Boost Query(bq):** reviews^2

**Rerank Query():** {!ltr model=my_efi_model efi.text="kingdom"}

# Information Need 4 - Evaluation

| Rank | System 3 | Improved System |
|---|---|---|
| AVP | 0.555556 | 0.808333 |
| P@10 | 0.3 | 0.5 |



# Mean Average Precision

| System 3 | Improved System |
| --- | --- |
| 0.800556 | 0.832705 |

# Conclusions and Future Work

The search system improvements are working as intended, providing a better search experience and query matching.

Regarding the performance of the improved system:

- Better overall performance;
- Better performance for more general information needs with vague searched terms (“kingdom”);
- Worst performance for more specific and ambiguous information needs.

Regarding future work:

- Improve Learning to Rank model implementation in order to improve its performance in more specific information needs;
- Improve the Page Rank implementation to be more complex - grant priority to books that are recommended by popular books (higher number of books).