

g

# Book Characterization using Project Gutenberg's Open Library with Goodreads Reviews

Information Processing and Retrieval



# Work done so far

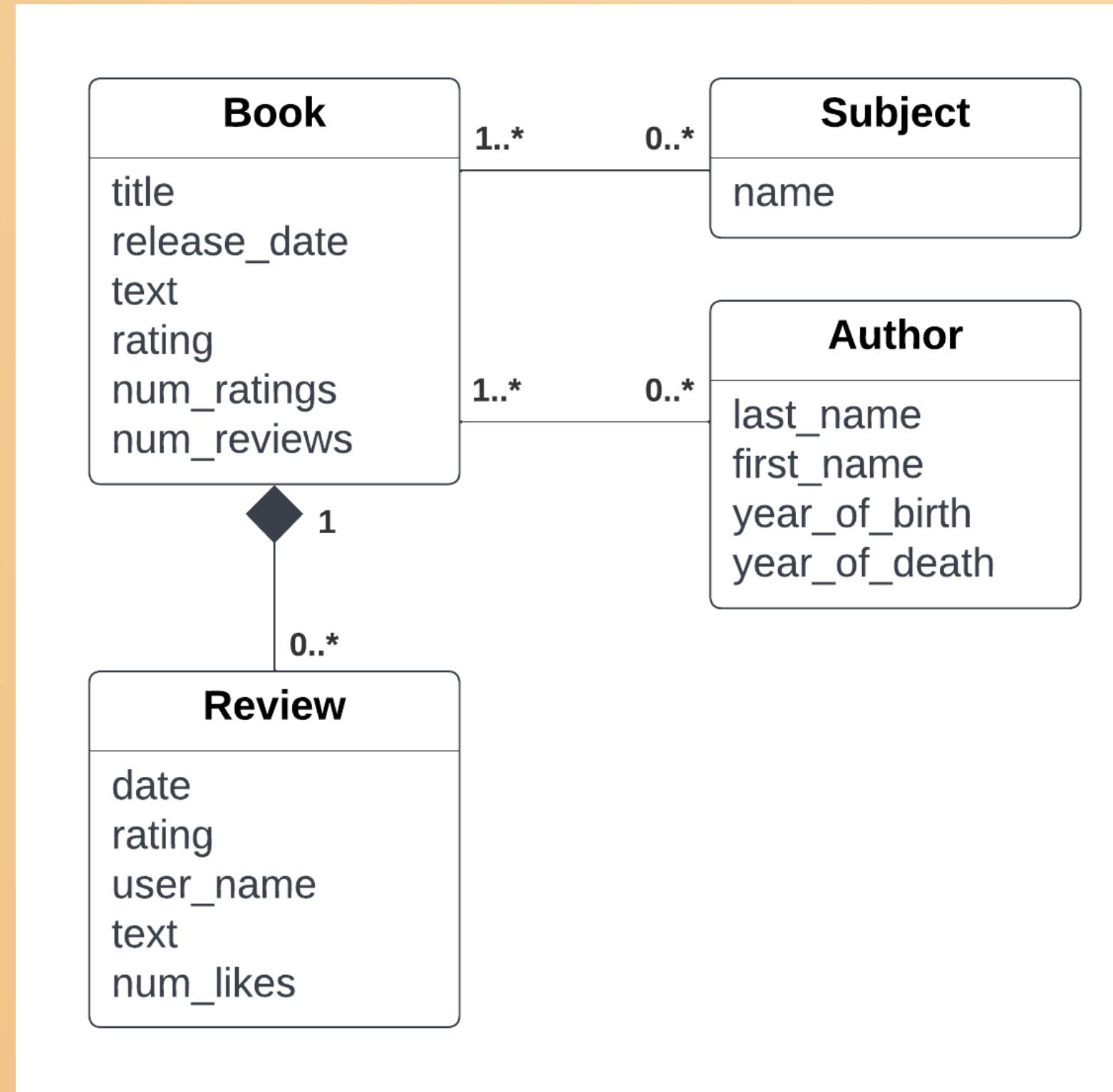


Fig. 1: Dataset Conceptual Model

- Three types of documents: Books, Authors, Reviews
- Two schemas: base schema and schema with synonyms
- Basic and improved field querying

## Search scenarios

- I want to find **excerpts** about a specific concept or event.
- I want to find context on a **book quote**.
- I want to read a book from a certain **time period**.
- I want to understand what other **people think** of a book.
- I want to **browse** an extensive library of copyright-free literature.

# Improvements

- **Expanded dataset** by collecting additional books and reviews
- **Named Entity Recognition** with OpenNLP
- Implementation of higher-level **REST API** and **frontend**
- **Faceting** for category suggestions
- **Highlighting** for quote search
- **Spelling correction**
- **Recomendation** of related books

# Expanded Dataset

## Before

- 8000 books
- 8300 reviews (10x830)
- 3.2 GB



## After

- 14581 books
- 19420 reviews (~5x3884)
- 5.4 GB

# Named Entity Recognition

```
{  
    "name": "NLPText",  
    "class": "solr.TextField",  
    "analyzer": {  
        "tokenizer": {  
            "class": "solr.OpenNLPTokenizerFactory",  
            "sentenceModel": "en-sent.bin",  
            "tokenizerModel": "en-token.bin"  
        },  
        "filters": [  
            {  
                "class": "solr.OpenNLPPoSFilterFactory",  
                "posTaggerModel": "en-pos-maxent.bin"  
            },  
            {  
                "class": "solr.OpenNLPChunkerFilterFactory",  
                "chunkerModel": "en-chunker.bin"  
            },  
            {  
                "class": "solr.TypeAsPayloadFilterFactory"  
            }  
        ]  
    },  
},  
}
```

Fig. 2: OpenNLP type for the text field

```
{  
    "name": "person_book",  
    "type": "string",  
    "indexed": true,  
    "multiValued": true,  
    "stored": true  
},  
{  
    "name": "location_book",  
    "type": "string",  
    "indexed": true,  
    "multiValued": true,  
    "stored": true  
},  
{  
    "name": "date_book",  
    "type": "string",  
    "indexed": true,  
    "multiValued": true,  
    "stored": true  
}
```

Fig. 4: Field definition for NER

```
<updateRequestProcessorChain name="add-unknown-fields-to-the-schema" default="${update.autoCreateFields:true}">  
    <processor class="solr.LogUpdateProcessorFactory"/>  
    <processor class="solr.DistributedUpdateProcessorFactory"/>  
    <processor class="solr.OpenNLPExtractNamedEntitiesUpdateProcessorFactory">  
        <str name="modelFile">conf/en-ner-person.bin</str>  
        <str name="analyzerFieldType">NLPText</str>  
        <str name="source">text</str>  
        <str name="dest">person_book</str>  
    </processor>  
    <processor class="solr.OpenNLPExtractNamedEntitiesUpdateProcessorFactory">  
        <str name="modelFile">conf/en-ner-location.bin</str>  
        <str name="analyzerFieldType">NLPText</str>  
        <str name="source">text</str>  
        <str name="dest">location_book</str>  
    </processor>  
    <processor class="solr.OpenNLPExtractNamedEntitiesUpdateProcessorFactory">  
        <str name="modelFile">conf/en-ner-date.bin</str>  
        <str name="analyzerFieldType">NLPText</str>  
        <str name="source">text</str>  
        <str name="dest">date_book</str>  
    </processor>  
    <processor class="solr.RunUpdateProcessorFactory"/>  
</updateRequestProcessorChain>
```

Fig. 3: Named Entity Extractor Configuration

```
{  
    "name": "text",  
    "type": "NLPText",  
    "indexed": true  
},
```

Fig. 5: Target Field for NE Extraction



# Named Entity Recognition

The screenshot shows the Solr admin interface with the 'books' core selected. The left sidebar has links for Dashboard, Logging, Security, Core Admin, Java Properties, Thread Dump, books (selected), Overview, Analysis, Dataimport, Documents, Files, Ping, Plugins / Stats, Query (selected), Replication, Schema, and Segments info. The main area is titled 'Request-Handler (qt)' and contains fields for 'q' (set to '{!parent which="content\_type:BOOK"} id:"1929"'), 'q.op' (set to 'OR'), 'fq' (empty), 'sort' (empty), 'start, rows' (0, 10), 'fl' (empty), 'df' (empty), 'wt' (empty), and checkboxes for 'indent on' (checked) and 'debugQuery' (unchecked). Below these are sections for 'defType' (set to 'lucene'), 'hl' (unchecked), 'facet' (unchecked), 'spatial' (unchecked), and 'spellcheck' (unchecked). A 'Raw Query Parameters' field contains 'key1=val1&key2=val2'. At the bottom is a blue 'Execute Query' button. To the right of the interface is a large text box displaying the JSON response from the Solr query. The URL shown is `http://localhost:8983/solr/books/select?indent=true&q.op=OR&q=%7B!parent%20which%3D%22content_type%3ABOOK%22%7D%0Aid%3A%221929%22`. The response includes the query parameters, the document ID (1934), title ('Songs of Innocence and of Experience'), release date ('1999-10-01T00:00:00Z'), subjects ('Pastoral poetry', 'English poetry -- 18th century'), content type ('BOOK'), person book ('Shepherd', 'Ah', 'John', 'O', 'Sees', 'Tom Dacre', 'Tom', 'As', 'Tom', 'Dick', 'Joe', 'Jack', 'Tom', 'Tom', 'Tom', 'Mary', 'Susan', 'Peace', 'Peace', 'Peace', 'Merry', 'Eternal', 'Babe', 'Ruby'), and a timestamp ('\_': '1670762286620').

Fig. 6: Output of NER on *Songs of Innocence and of Experience*

# Frontend

The screenshot shows a search interface integrated with a bookshelf background. At the top left is a logo with a magnifying glass icon and the text 'Booksearch'. Below it is a search bar with the placeholder 'Search...'. To the right of the search bar is a checkbox labeled 'Quote Search' and an 'Advanced' dropdown menu. A magnifying glass icon is positioned next to the search bar. The background features a wooden bookshelf filled with numerous books, with titles like 'The Magna Carta', 'The Great Charter', 'The English Experience', 'Essays and Studies Presented to India', 'The English Royal', 'Pax Britannica', 'The Trench', and 'The Russian Empire' visible.

## The Magna Carta

Authored by Anonymous (?)

Magna Carta, Constitutional history -- England -- Sources

Released on 01 Mar 2006

★★★★★ • 220 ratings

MORE LIKE THIS

Fig. 7: Front page of the search system

# Frontend

The screenshot shows a web-based book search interface titled "Booksearch". The background features a photograph of a wooden bookshelf filled with numerous books. In the foreground, there is a search form with various input fields and filters.

**Search Bar:** A large input field labeled "Search..." with a magnifying glass icon to its right. Below it is a checkbox labeled "Quote Search".

**Advanced Search Buttons:** Two buttons: "Advanced ^" and "Advanced ▾".

**Search Filters:** Several input fields and dropdown menus for filtering results:

- Title:** Input field.
- Category:** Input field.
- Author first name:** Input field containing "Fernando".
- Author last name:** Input field containing "Pessoa".
- Released after:** Input field with a calendar icon.
- Released before:** Input field with a calendar icon.
- Min rating:** Input field.
- Max rating:** Input field.
- Min # ratings:** Input field.
- Max # ratings:** Input field.
- Alive after:** Input field with a calendar icon.
- Alive before:** Input field with a calendar icon.

**SEARCH Button:** A white button with the word "SEARCH" in blue capital letters.

**Result Preview:** Below the search form, a preview of a book entry is shown:  
Title: English Poems, Volume 02 (of 2)  
Authored by Fernando Pessoa (1888-1935)  
Poetry  
Released on 11 Aug 2021  
★ ★ ★ ★ ★ • ratings

**More Like This:** A button labeled "MORE LIKE THIS" in a dark blue box.

Fig. 8: Advanced search with match

# Frontend

A screenshot of a book details page. The background is a dark image of several open books. The title "Aesthetic Poetry" is displayed in large white font. Below it, the author "Walter Pater (1839-1894)" and the genre "Aesthetics, English poetry – 19th century" are shown. A rating of four stars out of five is indicated with the text "• 38 ratings". The release date "01 Jul 2003" is also mentioned. At the bottom of the page, there are two buttons: "TRANSCRIPTION" (highlighted in pink) and "REVIEWS". Below these buttons, the text "Produced by Alfred J. Drake. HTML version by Al Haines." is displayed. Further down, the text "AESTHETIC POETRY+", "WALTER HORATIO PATER", and a quote from the book: "[213] THE "aesthetic" poetry is neither a mere reproduction of Greek or medieval poetry, nor only an idealisation of modern life and sentiment. The atmosphere on which its effect depends belongs to no simple form of poetry, no actual form of life. Greek poetry, medieval or modern poetry, projects, above the realities of its time, a world in which the forms of things are

Fig. 9 Book details with transcription

# Frontend

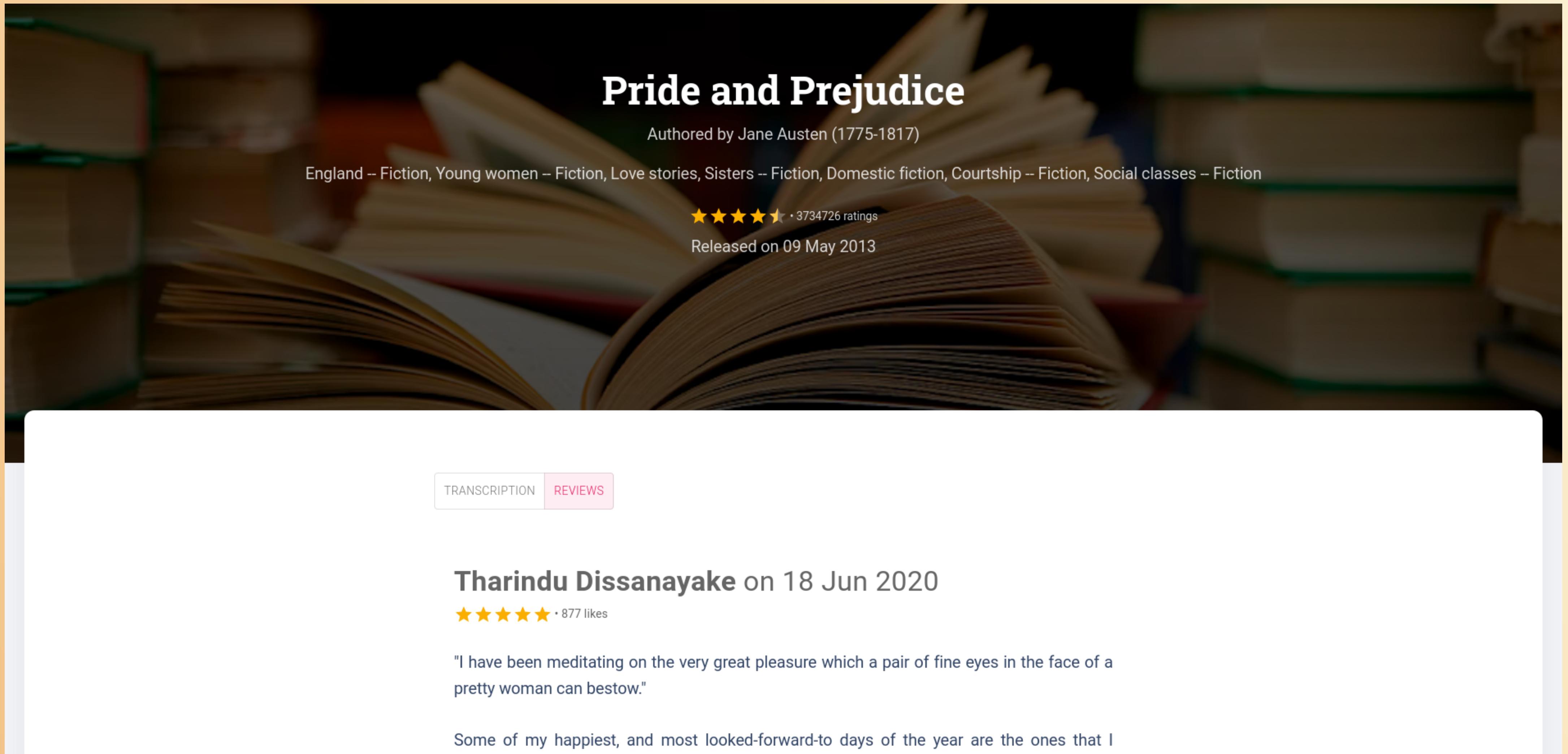


Fig. 10: Book details with reviews

# REST API

Django REST framework

Api Root

## Api Root

The default basic root view for DefaultRouter

OPTIONS    GET ▾

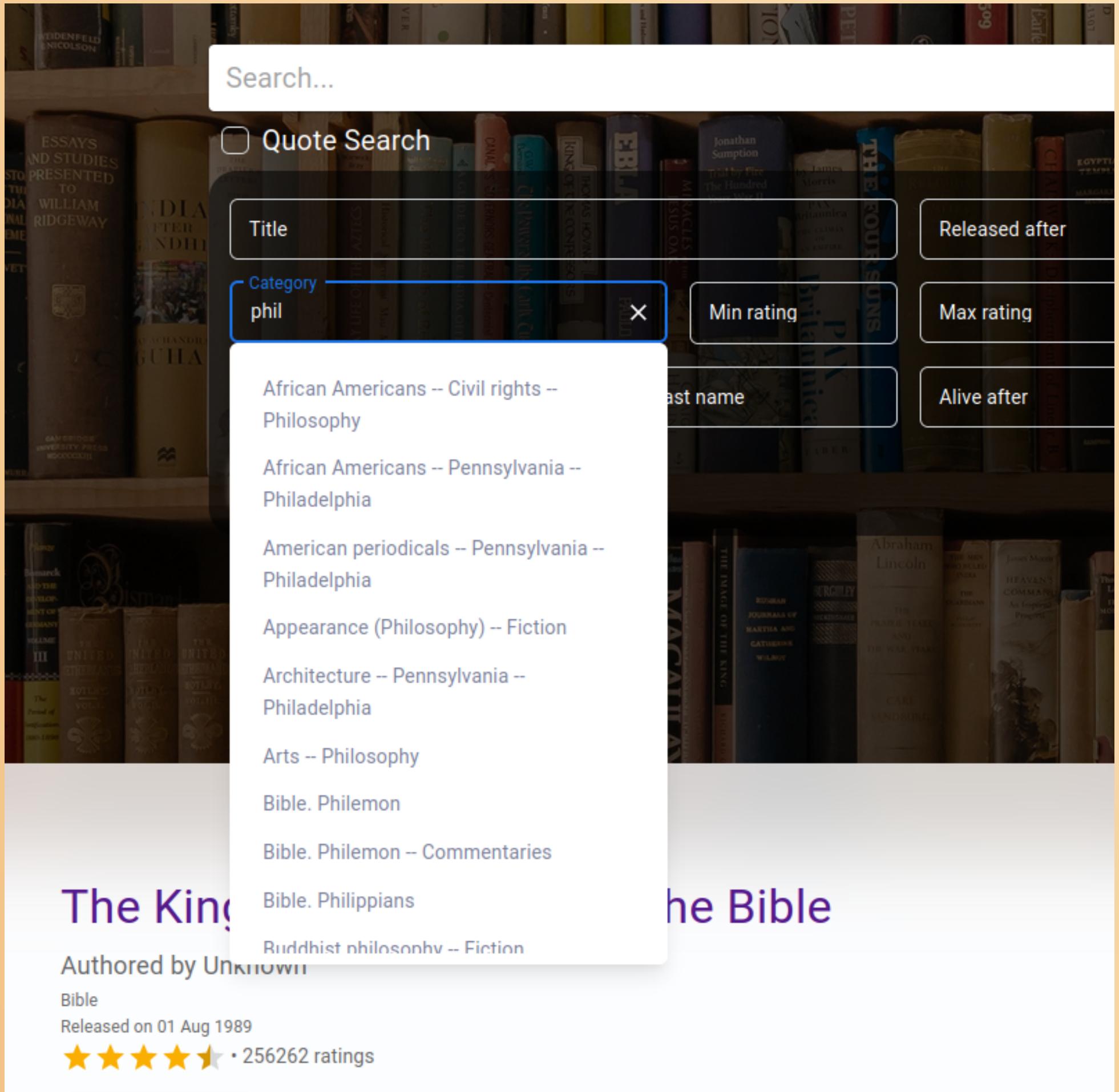
GET /api/v1.0/

HTTP 200 OK  
Allow: GET, HEAD, OPTIONS  
Content-Type: application/json  
Vary: Accept

```
{  
    "search": "http://localhost:8000/api/v1.0/search/",  
    "advancedSearch": "http://localhost:8000/api/v1.0/exactSearch/",  
    "exactSearch": "http://localhost:8000/api/v1.0/exactSearch/",  
    "browse": "http://localhost:8000/api/v1.0/browse/",  
    "categories": "http://localhost:8000/api/v1.0/categories/",  
    "moreLikeThis": "http://localhost:8000/api/v1.0/moreLikeThis/"  
}
```

Fig. 11: REST API

# Category Suggestions



- Suggestions based on what the user has typed
- Solr's faceting system to find categories
- Non-stored string field `subjects_facet`

# Quote Search

- Applies proximity search of 5 words
- Uses Solr highlighting feature (*hl* flag)



Fig. 13: Book card with quote highlighting and relevant information

# Spelling Correction

- Robustness to mistypes
- TextBlob python library (processes textual data)
- User may have intended to type like that → option to search without TextBlob processing

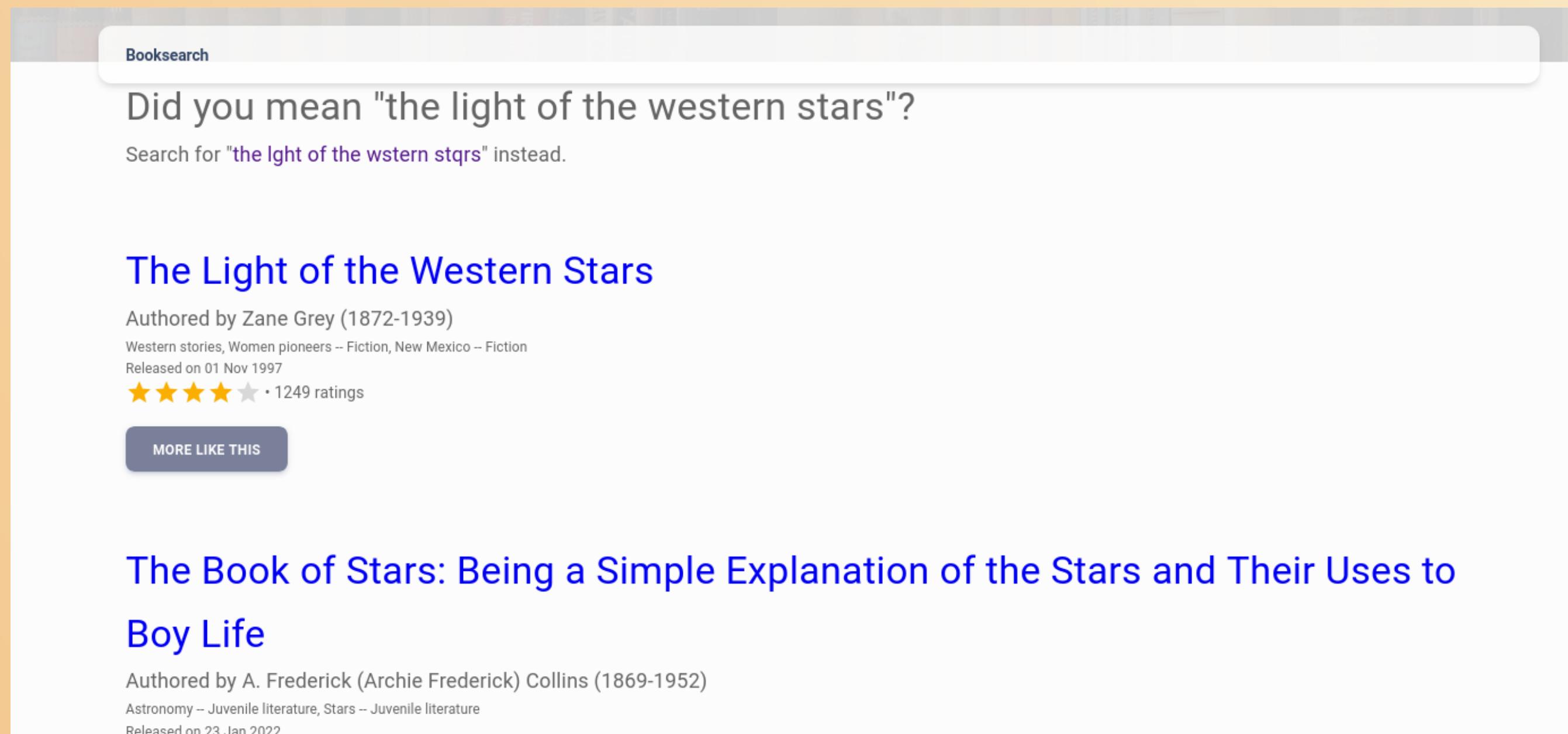


Fig. 14: Search results of corrected text

# Related Book Recommendation

- Click on “More like this” to search for related books
- Uses “More like this” feature of Solr
- Loaded “More like this” component to make */mlt* route available
- Target field: text

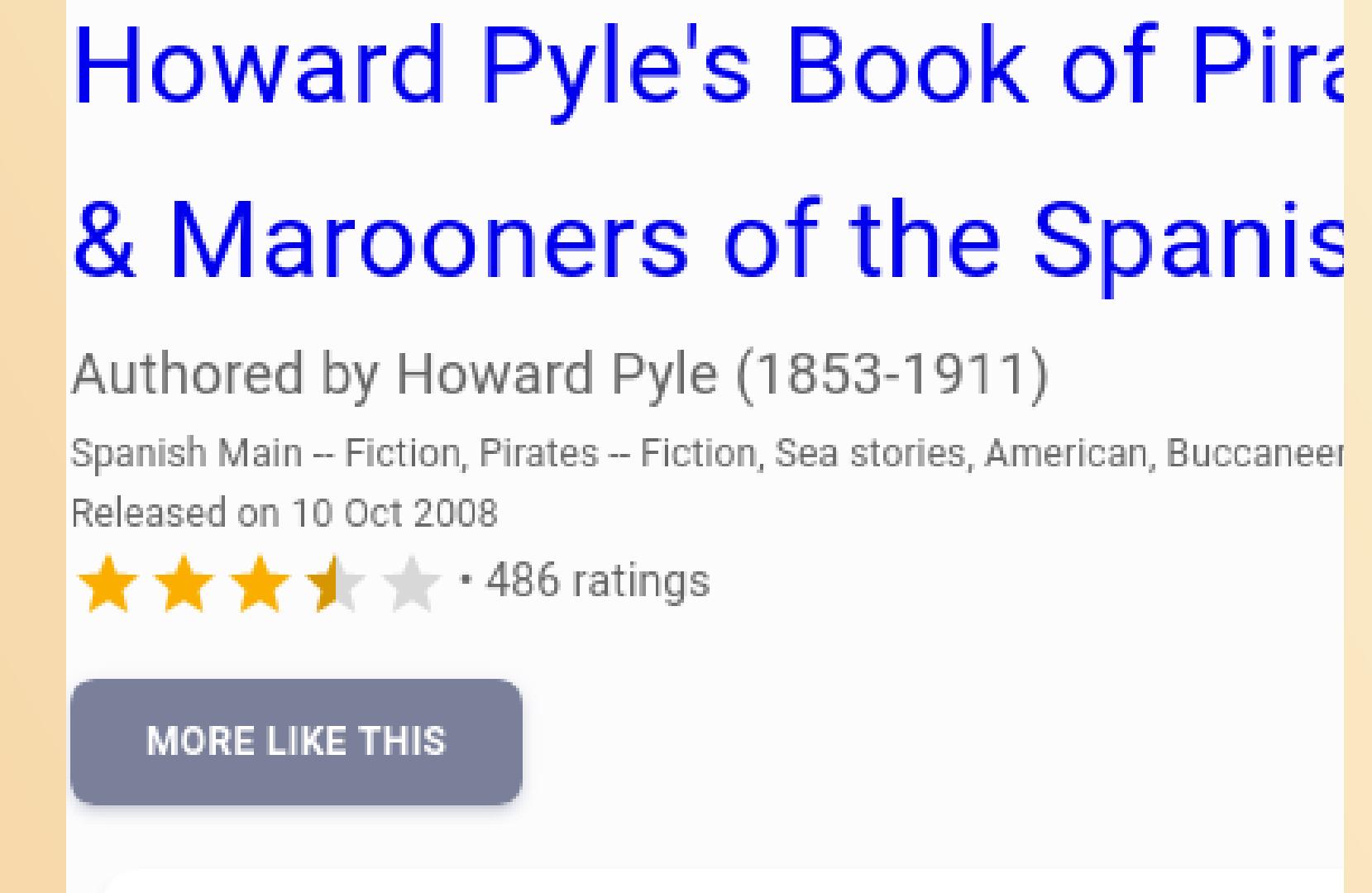


Fig. 15: More like this button

# Evaluation

## Information Need 1

Books explaining home economics

## Information Need 2

Recent romantic novels released preferably around 2018

## Information Need 3

Books containing the quote "waste not, want not", even if the user doesn't quite know the quote (inputting "want not, waste not")

## Information Need 4

Books containing the quote "waste not, want not", even if the user doesn't quite know the quote and has typos (inputting "want nto, waste not")

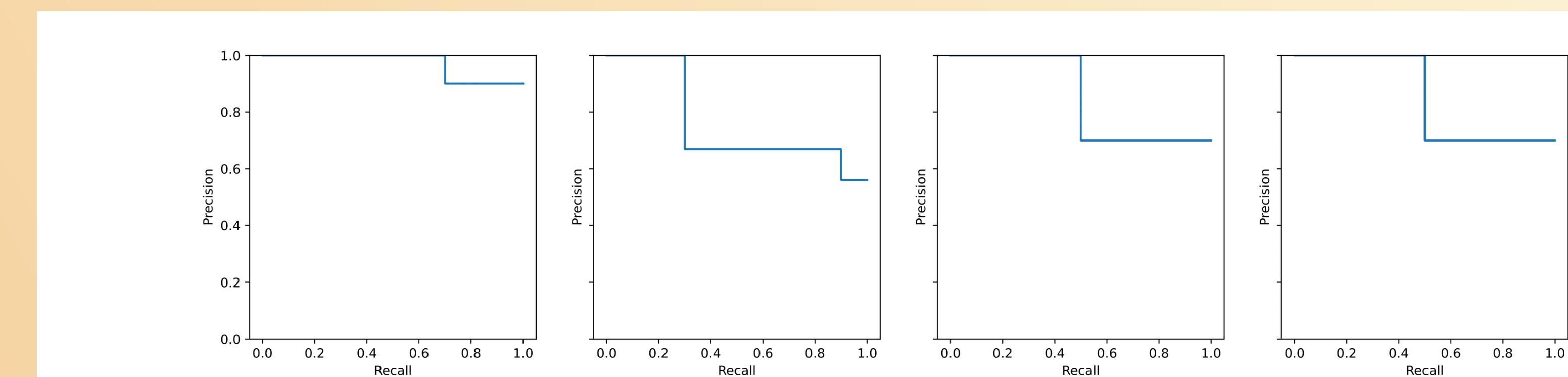


Fig. 16: P-R Curve

Table 1: Information Need 1 Metrics

Rank	IN1	IN2	IN3	IN4
1	R	R	R	R
2	R	I	R	R
3	R	R	R	R
4	R	I	I	I
5	R	R	I	I
6	R	R	R	R
7	I	I	I	I
8	R	I	R	R
9	R	R	R	R
10	R	I	R	R
<b>P@10</b>	0.9	0.8	0.7	0.7
<b>AP</b>	0.952103	0.606032	0.757976	0.757976

Mean average precision:  
0.768522

# Conclusions and Future Work

## Conclusions:

- Information retrieval system covered **all** search scenarios
- Good schema definition and query optimization tools are essential for good performance

## Future work:

- Named Entity-based queries
- OpenNLP on query – allowing for natural language queries
- Explore suggestion while writing query