



Sorbonne Université
Faculté de Science et d'ingénierie
Département Informatique

Rapport

*Spécialité :
Science et Technologie Logiciel*

DAAR

Rendu projet final

Tabellout Salim
Tabellout Yanis

le : 31/03/2025

1. Introduction

Ce rapport décrit le moteur de recherche de livres basé sur la bibliothèque Gutenberg. Il met en place des fonctionnalités avancées comme la recherche par mots-clés, la suggestion de livres basées sur la distance de Jaccard et des mesures de centralité (closeness, betweenness, cosine similarity). Un graphe est construit pour relier les livres en fonction de leur similarité, permettant d'améliorer la recherche et le classement des résultats.

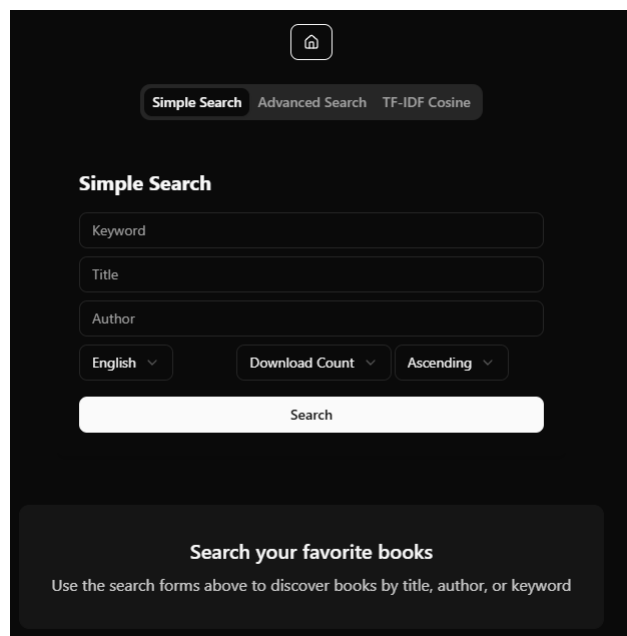
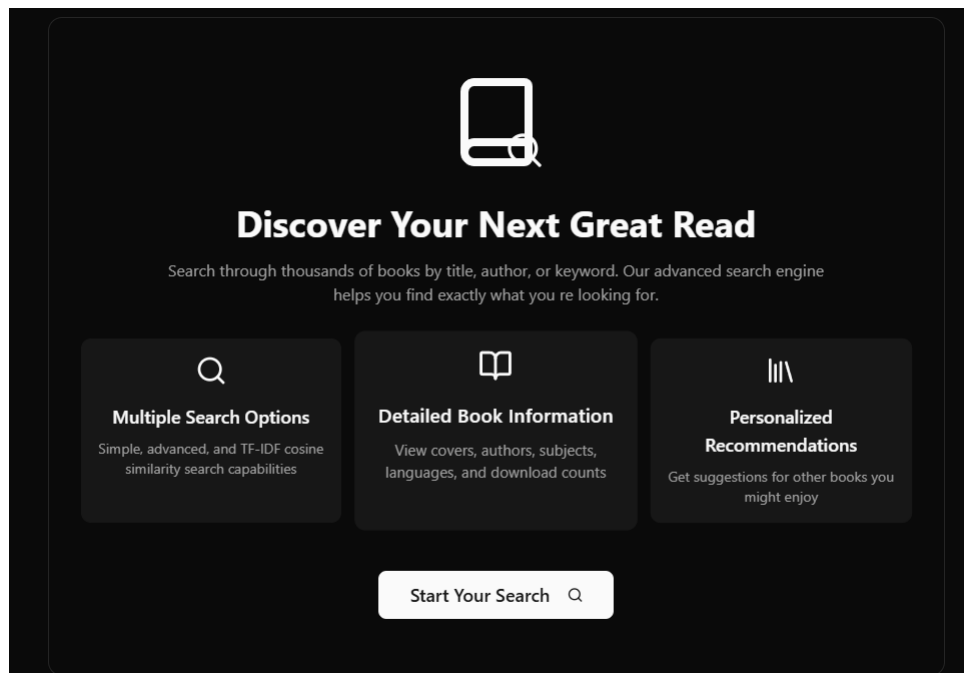
2. Architecture Technique


2.1. Structure du Projet

Le projet est structuré en plusieurs composants modulaires :

- **Backend** : Django
 - Gestion des modèles de données
 - Traitement des mots-clés
 - Construction du graphe de similarité
 - API REST pour la recherche et les suggestions
- **Frontend** : Next.js
 - Interface utilisateur réactive
 - Gestion des requêtes à l'API backend
 - Affichage des résultats de recherche
- **Base de Données** : SQLite
 - Stockage des livres
 - Table des mots-clés
 - Relations entre livres
 - Graphe de similarité

3. Résultat du frontend





Simple Search **Advanced Search** TF-IDF Cosine

Advanced Search

Keyword

Enter a keyword

Type

Classique

Title

Enter a title

Type

Classique

Regex

Author

Enter an author

Type

Classique

Language

English

Sort By

Download Count


Order

Ascending

Search

Search your favorite books

Use the search forms above to discover books by title, author, or keyword



Simple Search Advanced Search **TF-IDF Cosine**

TF-IDF Cosine Search

Keyword

Enter a keyword for cosine similarity search

Type

Classique

Search

This search finds books with content similar to your keyword using TF-IDF and cosine similarity.

Search your favorite books

Use the search forms above to discover books by title, author, or keyword



Discover Your Next Great Read

Search through thousands of books by title, author, or keyword. Our advanced search engine helps you find exactly what you're looking for.



Multiple Search Options

Simple, advanced, and TF-IDF cosine similarity search capabilities



Detailed Book Information

View covers, authors, subjects, languages, and download counts



Personalized Recommendations

Get suggestions for other books you might enjoy

Start Your Search



Results (6 books found)

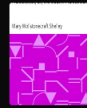
Related to title "Frank"



Frankenstein; Or, The Modern Prometheus
Shelley, Mary Wollstonecraft



Frankenstein; Or, The Modern Prometheus
Shelley, Mary Wollstonecraft



Frankenstein; Or, The Modern Prometheus
Shelley, Mary Wollstonecraft



Autobiography of Benjamin Franklin
Franklin, Benjamin



The Autobiography of Benjamin Franklin
Franklin, Benjamin



Frank and Andy at boarding school
Barnum, Vance

You Might Also Like (4 books)



Manon Lescaut
Prevost, abbe



Wieland; Or, The Transformation
Brown, Charles Brockden



Mathilda
Shelley, Mary Wollstonecraft



The Sorrows of Young Werther
Goethe, Johann Wolfgang von

Frankenstein; Or, The Modern Prometheus

Authors:

Shelley, Mary Wollstonecraft (1797 - 1851)

Subjects:

Gothic fiction, Horror tales, Science fiction, Frankenstein's monster (Fictitious character) -- Fiction, Frankenstein, Victor (Fictitious character) -- Fiction, Monsters -- Fiction, Scientists -- Fiction

Languages:

en

Downloads:

146616

Read the book

The Project Gutenberg eBook of Frankenstein; Or, The Modern Prometheus

This eBook is for the use of anyone anywhere in the United States and most other parts of the world at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg license included with this eBook or online at www.gutenberg.org. If you are not located in the United States, you will have to check the laws of the country where you are located before using this eBook.

Title: Frankenstein; Or, The Modern Prometheus

Author: Mary Wollstonecraft Shelley

Release date: October 1, 1991 [eBook #86]

Most recently updated: November 5, 2024

Language: English

Credits: Judith Boss, Christy Phillips, Lynn Hämäläinen and David Meltzer. HTML version by Al Haines. Further corrections by Menno de Leeuw.

*** START OF THE PROJECT GUTENBERG EBOOK FRANKENSTEIN; OR, THE MODERN PROMETHEUS ***

Frankenstein;

or, the Modern Prometheus

By Mary Wollstonecraft (Godwin) Shelley

CONTENTS

Letter 1

Letter 2

Letter 3

Letter 4

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

Chapter 10

Chapter 11

Chapter 12

4. Indexation et Extraction des Mots-clés

L'extraction des mots-clés suit un processus bien défini :

1. Récupération des livres depuis l'API Gutenberg.
2. Normalisation du texte (suppression des stopwords, lemmatisation).
3. Comptage des occurrences de chaque token pour chaque livre.
4. Stockage des token dans une base de données avec le nombre d'occurrence, et le livre où il apparaît.
5. Calcul des scores TF-IDF pour chaque mot-clé, permettant une meilleure pondération des termes.

5. Processus de Traitement des Données

5.1. Extraction et Normalisation des Mots-clés

Le processus de traitement des données comprend plusieurs étapes cruciales :

1. Récupération des Données

- Utilisation de l'API Gutendex
- Extraction de livres en français et anglais
- Multithreading avec 5000 threads pour optimiser la récupération

2. Prétraitement Textuel

- Utilisation de la bibliothèque spaCy
- Normalisation du texte :
 - Conversion en minuscules
 - Suppression des mots-vides (stopwords)
 - Lemmatisation
- Modèles linguistiques spécifiques (anglais et français)

3. Extraction des Mots-clés

- Comptage des occurrences avec **Counter**
- Création de fichiers JSON par livre
- Stockage des mots-clés avec leurs fréquences

6. Stratégies d'Optimisation

6.1. Réduction des Tokens

La gestion des tokens a nécessité une stratégie d'optimisation rigoureuse :

— Problématique Initiale

- 10 millions de tokens pour l'anglais

- 400k tokens pour le français
- Temps de traitement : 8 heures pour 250 000 tokens
- Table d'index de 1 million de lignes
- **Mécanisme de Seuillage**
 - Seuil de 5 : réduction de 74-77%
 - Seuil retenu : 25 pour l'anglais (580k tokens, à 95% réduction)
 - Seuil retenu : 10 pour le français (40k tokens à 88% de réduction)
- **Filtrage des Tokens à Basse Fréquence**
 - Exclusion des tokens apparaissant une seule fois
 - Réduction des tokens :
 - Anglais : de 10 millions à 36 000
 - Français : de 400k à 44k

6.1.1. Graphiques et Visualisations

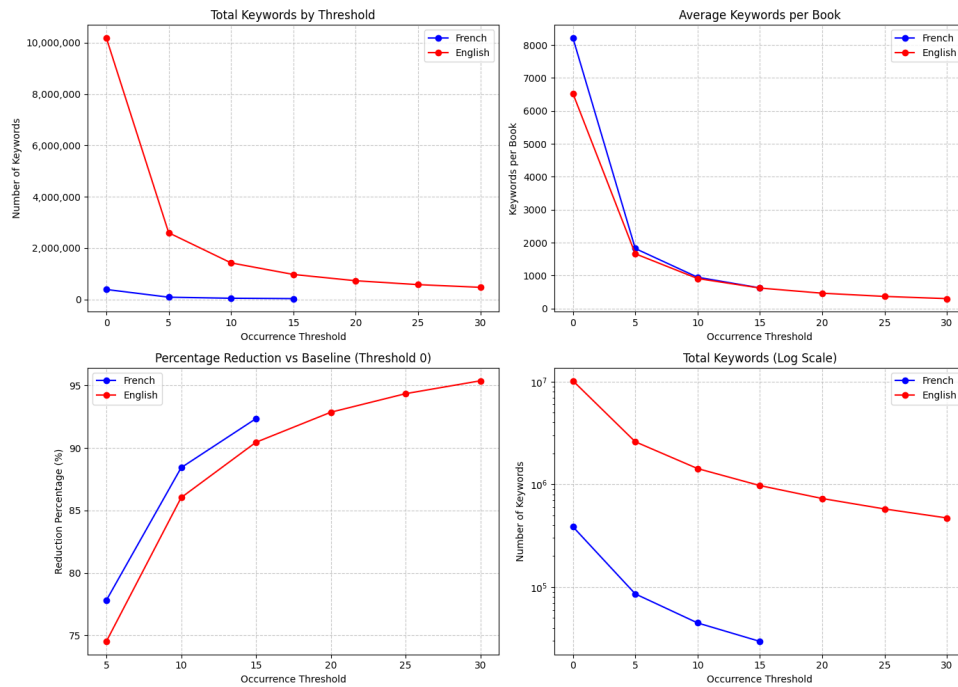


FIGURE 1 – Analyse combinée des mots-clés

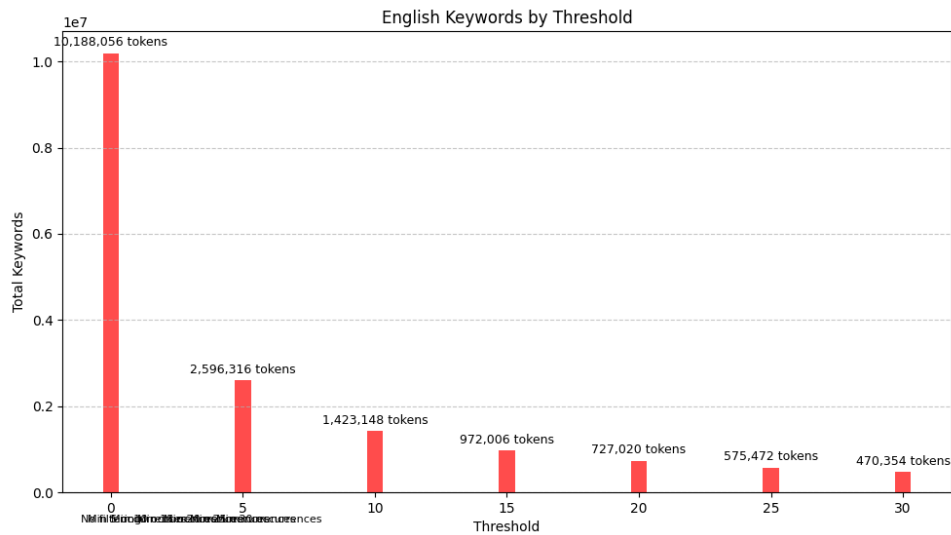


FIGURE 2 – Distribution des mots-clés en anglais

6.2. Stratégies de Mise en Cache

- Cache multi-niveaux pour les différentes parties de l'application
- Mise en cache des réponses API
- Mise en cache des données de voisinage
- Mise en cache des calculs de centralité
- Clés de cache uniques basées sur les paramètres de requête
- Délais d'expiration de 24 heures pour les éléments mis en cache

7. Défis Techniques et Solutions

7.1. Goulots d'Étranglement et Optimisations

- **Problème Initial** : Requêtes lentes de lecture/écriture en base de données
- **Solution** :
 - Préfiltre des tokens à faible occurrence
 - Optimisation des requêtes
 - Mise en place de mécanismes de mise en cache
- **Complexité de Calcul**
 - Limitation du calcul de centralité aux 30 nœuds les plus connectés
 - Restriction à 20 voisins pour les nœuds avec de nombreux voisins
 - Mise en place de contrôles de délai d'attente (2 secondes)

7.2. Compromis Techniques

- Arbitrage entre exhaustivité et performance
- Choix de seuils pour réduire la complexité computationnelle
- Approche itérative d’optimisation des algorithmes

8. Construction du Graphe de Similarité

Un graphe est construit où chaque nœud représente un livre et les arêtes sont créées selon la distance de Jaccard. Si la similarité entre deux livres est supérieure à un seuil (0.4), une arête est ajoutée.

La distance de Jaccard est définie comme :

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

Le graphe final contient 1099 nœuds et 21478 arêtes. L’algorithme parcourt chaque paire de livres et compare leurs ensembles de mots-clés pour établir les connexions. Les relations sont ensuite stockées dans la base de données sous forme de table ‘Neighbors’.

9. Méthodes de Classement et Recherche Avancée

9.1. Recherche Avancée

Deux types de recherche sont implémentés :

- Recherche simple par mot-clé.
- Recherche avancée avec expressions régulières.

9.2. Classement des Résultats

Les résultats sont classés selon :

- Closeness centrality : mesure l’inverse de la somme des distances minimales d’un nœud aux autres.
- Betweenness centrality : compte le nombre de chemins minimaux passant par un nœud.
- Cosine similarity : calculée via une API qui récupère les mots-clés et applique la similarité de cosinus sur les vecteurs TF-IDF pré-calculés.

L’utilisation d’un cache pour stocker les requêtes, pour gagner du temps de calcul.

9.3. Suggestions de Livres

Les suggestions sont générées via :

- Les voisins immédiats du graphe de Jaccard.
- Les livres les plus populaires selon le nombre de téléchargements.

10. Détails sur la Similarité Cosinus

La similarité cosinus permet de comparer les livres en fonction de leurs vecteurs TF-IDF.

$$\text{similarité}(A, B) = \frac{\sum_i A_i B_i}{\sqrt{\sum_i A_i^2} \times \sqrt{\sum_i B_i^2}} \quad (2)$$

Le processus se déroule comme suit :

1. Extraction des mots-clés du livre cible.
2. Représentation de ces mots-clés sous forme de vecteurs pondérés (TF-IDF).
3. Comparaison avec tous les autres livres et classement des résultats.

L'API `/data/books/keywords/cosine-similarity/` permet d'obtenir ces résultats en quelques millisecondes.

11. Résultats et Performance

Des tests ont été effectués pour comparer les performances des différentes méthodes :

Méthode	Temps d'exécution (s)
Cosine Similarity (API)	0.04
Closeness	1.32
Betweenness	1.44
Download count	1.5

TABLE 1 – Temps de calcul des différentes méthodes de classement

12. Conclusion

Ce rapport a mis en avant le travail sur le projet du moteur de recherche. Si vous voulez plus de points techniques, veuillez vous référer au README.

Les principales réalisations de ce projet incluent :

- Un moteur de recherche de livres basé sur la bibliothèque Gutenberg
- Des algorithmes avancés de recherche et de suggestion
- Une architecture modulaire combinant backend Django et frontend Next.js
- Des méthodes innovantes d'analyse de similarité entre livres

Le projet démontre la possibilité de créer un système de recherche intelligent et performant, capable de suggérer des livres de manière contextuelle et rapide.