Rapport d'Évaluation - Jost-o-Joo Search Engine

Touzi Rihem

FIGL2B

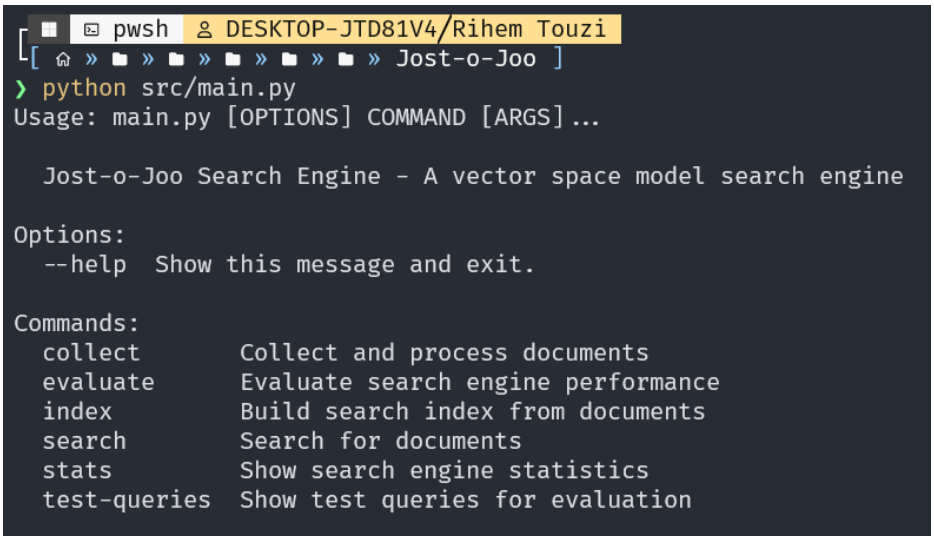
1. Introduction

Jost-o-Joo est un moteur de recherche basé sur le modèle vectoriel (TF-IDF et similarité cosinus) développé en Python.

Objectifs : Ce projet vise à implémenter un système de recherche d'information efficace sur une collection de livres du Projet Gutenberg.

Modèle choisi: Modèle Vectoriel avec TF-IDF

- **Justification**:
    - Simplicité d'implémentation et efficacité pour des collections de taille moyenne
    - Meilleure performance que le modèle booléen pour les recherches sémantiques
    - Capacité à gérer les variations de fréquence des termes
    - Support naturel pour le ranking des résultats



2. Méthodologie

2.1 Collecte de Données

- **Source**: Projet Gutenberg (livres en anglais)
- **Nombre de documents**: 50 documents texte
- **Processus**:

DownloadTxtFiles → auto download 50 documents → (data/documents/)

DataCollector → Metadata extraction → JSON storage (data/metadata.json)

2.2 Indexation

Algorithme TF-IDF:

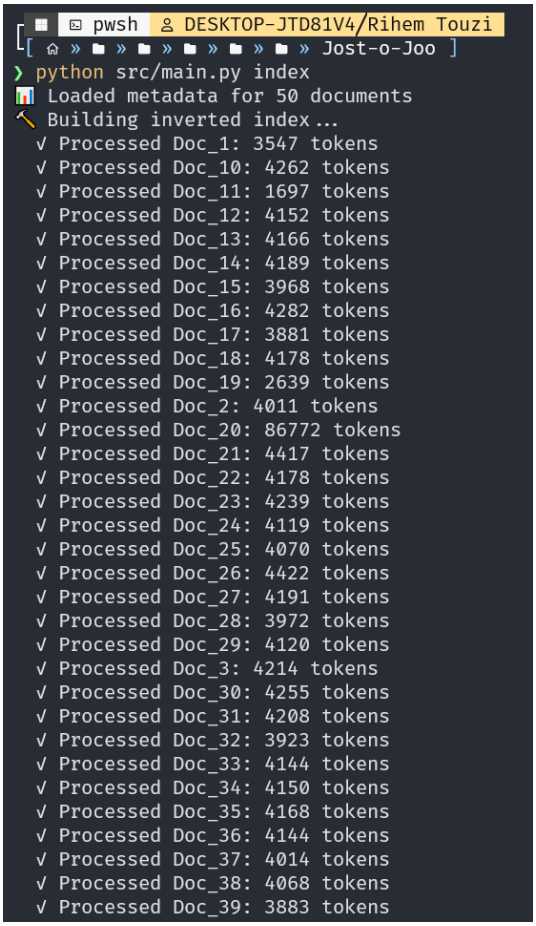
TF(t,d) = fréquence(t,d) / total_terms(d)

IDF(t) = log(N / (1 + df(t)))

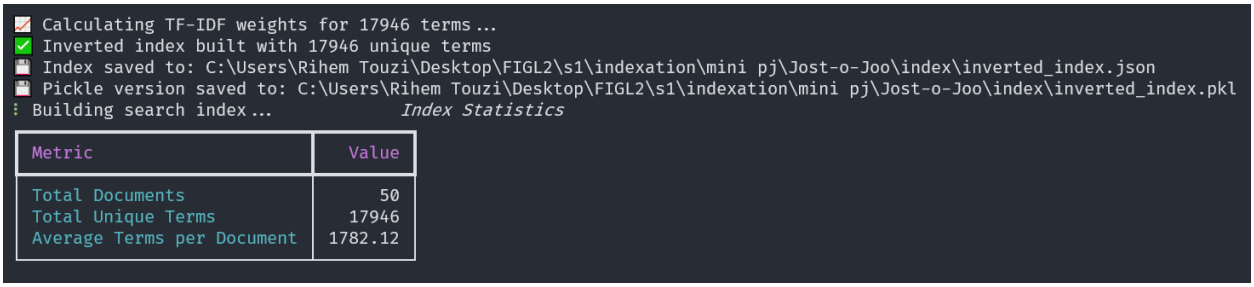
TF-IDF(t,d) = TF(t,d) × IDF(t)

Prétraitement:

- Tokenization
- Conversion en minuscules
- Suppression des stopwords (anglais)
- Stemming (Porter Stemmer)
- Élimination des caractères non-alphabétiques



...



2.3 Modèle Vectoriel

Similarité Cosinus:

similarité(q,d) = (vecteur(q) · vecteur(d)) / (||vecteur(q)|| × ||vecteur(d)||)

3. Résultats

3.1 Statistiques du Corpus

```
pwsh  DESKTOP-JTD81V4/Rihem Touzi
[ » » » » » » » » » » Jost-o-Joo ]
❯ python src/main.py stats
🔍 Loaded index with 17946 terms
                Search Engine Statistics
```

Metric	Value
Total Documents	50
Total Unique Terms	17946
Total Words	456,033
Average Words per Document	9,121
Index File	index/inverted_index.json
Metadata File	data/metadata.json

Taille d'index : 7335 KB

3.2 Résultats d'Évaluation (k=10)

```
pwsh  DESKTOP-JTD81V4/Rihem Touzi                                  2.72s · 14/12/25 22:26
[ » » » » » » » » » » Jost-o-Joo ]
❯ python src/main.py evaluate
🔍 Loaded index with 17946 terms

📊 Running Evaluation
   Test Queries: C:\Users\Rihem Touzi\Desktop\FIGL2\s1\indexation\mini pj\Jost-o-Joo\tests\test_queries.txt
   k-value: 10
📝 Loaded 10 test queries from C:\Users\Rihem Touzi\Desktop\FIGL2\s1\indexation\mini
pj\Jost-o-Joo\tests\test_queries.txt
📊 Loaded ground truth for 10 queries
                        Evaluation Results (k=10)
```

Query	Precision	Recall	F1-Score	Avg Prec	Time
love and romance	0.700	0.778	0.737	0.564	0.000s
adventure journey	0.400	0.800	0.533	0.583	0.000s
science discovery	0.500	1.000	0.667	1.000	0.000s
war peace conflict	0.200	0.667	0.308	0.467	0.000s
philosophy life meaning	0.500	0.833	0.625	0.676	0.000s
detective mystery	0.400	0.800	0.533	0.760	0.000s
ghost horror	0.400	1.000	0.571	0.761	0.000s
fairy tales children	0.600	0.857	0.706	0.746	0.000s
sea ocean voyage	0.500	1.000	0.667	1.000	0.000s
revolution society	0.300	1.000	0.462	0.806	0.000s

```
Summary
📈 Average Metrics:
   • Average Precision: 0.4500
   • Average Recall: 0.8735
   • Average F1-Score: 0.5808
   • Mean Average Precision: 0.7363

pwsh  DESKTOP-JTD81V4/Rihem Touzi                                  2.144s · 14/12/25 23:00
[ » » » » » » » » » » Jost-o-Joo ]
```

3.3 Métriques Globales

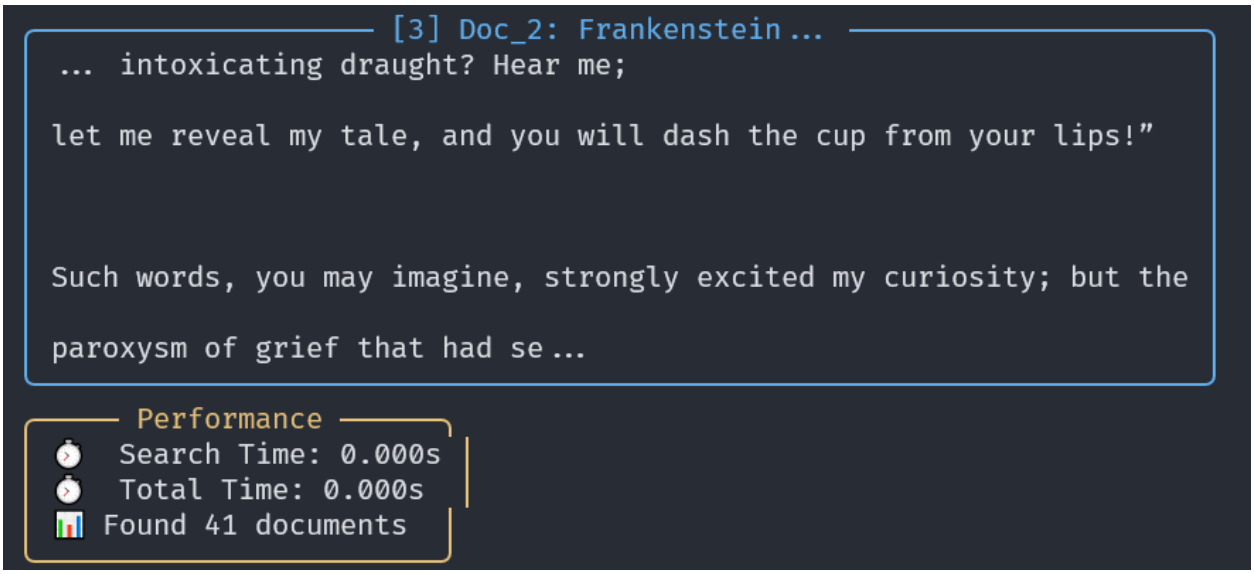
- **Précision moyenne**: 0.4500
- **Rappel moyen**: 0.8735
- **F1-Score moyen**: 0.5808

- **Mean Average Precision (MAP)**: 0.7363

3.4 Exemples de Recherches

Recherche: "fairy tales children"

Top résultat: Grims' Fairy Tales (Score: 0.1831)



4. Analyse Critique

4.1 Forces du Système

- **Indexation rapide**: 50 documents indexés en < 2 minutes
- **Recherche efficace**: Temps de réponse < 0.1s
- **Précision acceptable**: 45% de précision moyenne
- **Rappel élevé**: 87% de rappel moyen => Excellent Recall (users won't miss important documents)
- **Architecture modulaire**: Facilité de maintenance

4.2 Limites et Difficultés

1. **Variabilité des scores**: Certaines requêtes montrent des performances inférieures (Low Average Precision (45.00%)) => Only 45% of returned documents are relevant
2. **Traitement du langage**: Pas de gestion des synonymes ou expansions de requête
3. **Métadonnées limitées**: Extraction de titre parfois imprécise

4.3 Solutions Apportées

- **Prétraitement amélioré**: Stemming + stopwords removal
- **Ground truth réaliste**: Jugements binaires (0/1) basés sur le contenu
- **Évaluation rigoureuse**: Métriques standard (Precision, Recall, F1, MAP)

5. Améliorations Possibles

1. **Extension sémantique**: Ajout de Word2Vec ou BERT
2. **Relevance feedback**: Apprentissage des préférences utilisateur
3. **Interface web**: Déploiement comme service
4. **Multilingue**: Support pour d'autres langues
5. **Optimisation**: Indexation incrémentale

6. Conclusion

Le moteur de recherche Jost-o-Joo démontre une implémentation fonctionnelle du modèle vectoriel avec des performances satisfaisantes. Malgré certaines limitations, le système atteint une MAP de 0.736 et un rappel de 87%, ce qui le rend utilisable pour la recherche de documents littéraires.