Roua Alarji FIGL2A

# Rapport Final – Projet de Recherche d'Information

## 1- Introduction :

Dans un monde où les volumes de données textuelles augmentent rapidement, la capacité à retrouver des informations pertinentes dans un corpus devient essentielle. Les moteurs de recherche jouent un rôle clé en permettant d'extraire rapidement les documents utiles pour l'utilisateur .

**Objectifs :**

Le projet vise à :

- Créer un moteur de recherche capable d'indexer et de rechercher efficacement des documents.
- Implémenter et comparer deux modèles de Recherche d'Information : TF-IDF et BM25.
- Évaluer les modèles avec des métriques standards (précision, rappel, F1) pour mesurer leur performance.
- Fournir une interface web simple et fonctionnelle pour tester la recherche et afficher les résultats.

**Choix du modèle**

Deux modèles complémentaires ont été retenus :

- **TF-IDF** : mesure la fréquence des termes pondérée par leur importance dans le corpus. Simple et efficace, il constitue une base solide pour la recherche d'information.
- **BM25** : améliore TF-IDF en tenant compte de la longueur des documents e, offrant un classement plus précis des documents pertinents.

## 2-Méthodologie :

### 1- Collecte de données :

Les données ont été collectées automatiquement depuis des pages web fiables, principalement Wikipedia. Chaque document contient le titre, l'URL et le texte principal de la page.
Les documents sont stockés au format JSON et seules les pages contenant un texte suffisant ont été conservées.

Le corpus final comprend plus de 50 documents couvrant différents domaines : informatique, sciences, économie, santé et culture.

### 2. Indexation

Les documents ont été traités pour créer un index permettant des recherches efficaces. Les étapes principales sont :

- **Nettoyage et normalisation** : suppression des caractères spéciaux et mise en minuscule.
- **Tokenisation** : découpage du texte en mots ou termes significatifs.
- **Vectorisation** : représentation des documents sous forme de vecteurs, utilisant le modèle TF-IDF.

### 3. Modèle de Recherche d'Information (RI)

Deux approches ont été implémentées :

- **TF-IDF (Term Frequency – Inverse Document Frequency)** : mesure l'importance d'un terme dans un document par rapport à l'ensemble du corpus.
- **BM25 (Best Matching 25)** : un modèle probabiliste amélioré qui prend en compte la saturation de fréquence et la longueur des documents.

## 3. Résultats

### 3.1 Statistiques du corpus :

Le corpus utilisé pour notre moteur de recherche contient des documents collectés depuis Wikipédia à l'aide d'un script de scraping. Les principales statistiques sont :

**Nombre total de documents : 88**

**Nombre total de mots : 479,814**

**Moyenne de mots par document : 5,452.4**

### 3.2 Exemples de Recherches

Nous présentons ici des exemples représentatifs de requêtes avec les résultats obtenus par les deux modèles.

Exemple 1 : "intelligence artificielle"

| Rang | TF-IDF | Score | BM25 | Score |
|---|---|---|---|---|
| 1 | Intelligence artificielle | 0.5332 | Intelligence artificielle | 4.1661 |
| 2 | Intelligence émotionnelle | 0.2479 | Système multi-agents | 3.7841 |
| 3 | Intelligence économique | 0.2231 | Apprentissage automatique | 3.6288 |
| 4 | Système multi-agents | 0.1257 | Apprentissage automatique | 3.6288 |
| 5 | Apprentissage automatique | 0.0677 | Apprentissage profond | 3.3649 |

### Résultats TF-IDF

Le modèle TF-IDF parvient à identifier plusieurs documents contenant les termes « *intelligence* » et « *artificielle* », mais son classement reste imparfait.
Les premiers résultats incluent parfois :

- des documents où les termes apparaissent de manière isolée,
- des pages qui mentionnent l'IA comme une définition ou en introduction,

Ce comportement est typique de TF-IDF :
il privilégie la présence brute du terme, sans distinguer les documents où le concept est réellement approfondi.

Score F1 : 0.429, ce qui montre que les documents retrouvés sont corrects mais mal classés en termes de pertinence.

### Résultats BM25

Le modèle BM25, en revanche, fournit un classement beaucoup plus aligné avec l'intention de la requête.

Les documents placés en tête sont ceux qui :

- présentent une explication complète du concept d'intelligence artificielle,
- développent les notions essentielles (algorithmes, apprentissage, applications),
- utilisent les termes clés dans un contexte dense et significatif,

Score F1 : 0.571, nettement supérieur à celui de TF-IDF

## 3.3 Résultats d'évaluation :

On obtient après évaluation des différents requêtes les résultats suivants :

| Modèle | Précision | Rappel | F1-mesure |
|---|---|---|---|
| TF-IDF | 0.131 | 0.850 | 0.222 |
| BM25 | 0.221 | 0.943 | 0.313 |

- Le rappel élevé pour TF-IDF (0.85) indique que la plupart des documents pertinents sont retrouvés, mais la précision faible (0.131) montre qu'il y a beaucoup de documents non pertinents dans les résultats.

- BM25 améliore à la fois la précision et la F1-mesure, confirmant sa capacité à prioriser les documents réellement pertinents.

- L'amélioration F1 de 9.2 % (0.091)est significative et reflète une meilleure adéquation des résultats avec les attentes de l'utilisateur.

# 4. ANALYSE CRITIQUE

## 4.1 Forces du Système :

**1. Architecture modulaire et extensible**

- Séparation claire entre collecte, indexation et recherche

- Facilite l'ajout de nouveaux modèles ou fonctionnalités

**2. Double implémentation (TF-IDF et BM25)**

- Permet la comparaison objective des performances

-BM25 offre des résultats significativement meilleurs

-TF-IDF reste plus rapide pour des applications temps-réel

**3. Prétraitement robuste**

- Gestion des stopwords en français

- Tokenisation adaptée au corpus

**5. Interface utilisateur fonctionnelle**

- Résultats clairs avec scores et extraits

## 4.2 Limites du Système

**1. Taille limitée du corpus**

Impact : Risque de ne pas couvrir toutes les requêtes utilisateurs

**2. Absence de traitement sémantique**

- Pas de gestion des synonymes ("IA" ≠ "intelligence artificielle")

Impact : Requêtes formulées différemment donnent des résultats différents

### 4.3 Difficultés Rencontrées et Solutions

**Difficulté 1 : Stopwords insuffisants**

- Problème : Liste NLTK ne couvrait pas tous les mots vides
- Solution : Ajout de filtrage supplémentaire (mots < 3 caractères)
- Résultat : Vocabulaire plus significatif .

**Résultats vides pour certaines requêtes**

- Problème :
  Si la requête contient des mots absents du vocabulaire → score = 0.
- Solution :
  Nettoyage de la requête + expansion possible des termes.

### 4.4 Améliorations Possibles

**À court terme (faciles à implémenter) :**

1. **Expansion du corpus**
   - Collecter 200-500 documents supplémentaires
   - Diversifier les sources (articles scientifiques, blogs, actualités)
   - Équilibrer la représentation thématique
2. **Amélioration du prétraitement**
   - Lemmatisation (ex: "chercher", "cherché" → "chercher")
   - Stemming pour réduire les variations morphologiques
   - Détection et traitement des entités nommées
3. **Enrichissement de l'interface**
   - Filtres par date, source, longueur
   - Tri par pertinence, date, ou popularité
   - Export des résultats CSV, JSON
4. **Clustering et classification**
   - Regroupement automatique des résultats par thème
   - Étiquetage automatique des documents
5. **Personnalisation avancée**
   - Profil utilisateur avec historique de recherche
   - Recommandations contextuelles