Université de la MANOUBA

ISAMM – Licence IMM – Techniques d'Indexation et de Référencement –

Responsable : Chiraz Trabelsi

Mini-projet 2 — Articles scientifiques

(Mini Project 2 — Scientific Abstracts Retrieval)

Corpus / Dataset : arXiv Open Metadata (title+abstract)

Domaine / Domain : Physique / Informatique

🎯 Objectifs pédagogiques / Learning Objectives

- Construire un index inversé et un modèle vectoriel TF-IDF.

- Exécuter des requêtes et classer les documents (cosinus).

- Évaluer : Precision@k, Recall@k, MAP.

- Appliquer le feedback de pertinence (Rocchio).

- Étudier l'impact du prétraitement (ablation).

🧠 Compétences visées / Skills Developed

Indexation textuelle, pondération TF-IDF, métriques IR, expérimentation, analyse critique.

⚙️ Pré-requis / Prerequisites

Python 3.x • scikit-learn • nltk • numpy • pandas • matplotlib

🧩 Énoncé & Consignes / Specification & Tasks

1) Index inversé : construire {terme : {doc_id : tf}} et sauvegarder l'index (JSON).

2) Modèle TF-IDF : vectoriser les documents (stopwords langue du corpus, min_df adapté).

3) Requêtes : exécuter au moins 8 requêtes représentatives :

quantum field theory; semiconductor laser; graph neural network; cosmic microwave background; optical cavity; spintronics; superconducting qubits; photonic integrated circuits

4) Évaluation : calculer P@5, P@10, Recall@10, AP, MAP (définir la pertinence).

Note : Pertinence proxy : catégorie arXiv du document correspondant au sous-domaine de la requête.

5) Feedback (Rocchio) : sélectionner D+ / D−, mettre à jour la requête, ré-évaluer.

6) Ablation : comparer au moins 4 pipelines (stopwords ON/OFF, Porter stemming ON/OFF).

💻 Démarrage (Python) / Starter Code

Chargement / Loading :

```
import pandas as pd
# Charger un CSV arXiv (title, abstract, categories)
df = pd.read_csv('arxiv_sample.csv')
docs = (df['title'].fillna('') + '. ' + df['abstract'].fillna('')).tolist()
```

Vectorisation TF-IDF :

```
from sklearn.feature_extraction.text import TfidfVectorizer
vec = TfidfVectorizer(stop_words='english', min_df=3)
X = vec.fit_transform(docs)
```

Classement par similarité cosinus :

```
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np
def rank_query(q, k=10):
    qv = vec.transform([q])
    sims = cosine_similarity(qv, X).ravel()
    top = np.argsort(sims)[::-1][:k]
    return list(zip(top, sims[top]))
```

Rocchio (feedback) :

```
import numpy as np
def rocchio(q_vec, Dpos, Dneg, a=1.0, b=0.75, g=0.15):
    qp = a*q_vec
    if Dpos.shape[0]: qp = qp + b*Dpos.mean(axis=0)
    if Dneg.shape[0]: qp = qp - g*Dneg.mean(axis=0)
    return qp
```

🔎 Questions d'analyse (réponses à rédiger) / Analysis Questions

Question (FR/EN)	Réponse étudiante / Student answer
1️⃣ Définir la pertinence et ses limites. / Define relevance and its limits.	…
2️⃣ Catégorie(s) dominante(s) pour 2 requêtes clés et pourquoi ? / Dominant categories and why?	…
3️⃣ Configuration offrant la meilleure MAP et justification. / Best MAP configuration and justification.	…
4️⃣ Effet moyen de Rocchio (Δ P@10, MAP). / Average effect of Rocchio.	…
5️⃣ Deux faux positifs récurrents + analyse. / Two recurring false positives + analysis.	…
6️⃣ Deux pistes d'amélioration simples. / Two simple improvement ideas.	…
7️⃣ Count vs TF-IDF : différences observées. / Count vs TF-IDF differences.	…
8️⃣ Impact de min_df sur vocabulaire et MAP. / Impact of min_df.	…
9️⃣ Trois termes TF-IDF saillants et leur rôle. / Three salient TF-IDF terms and role.	…
1️⃣0️⃣ Limites majeures du modèle vectoriel. / Major limitations of vector space model.	…

📘 Livrables / Deliverables

- Notebook/script Python commenté ; index JSON ; résultats (CSV).

- Tableaux P@5, P@10, Recall@10, MAP ; avant/après Rocchio.

- 2 pagee d'analyse répondant aux 10 questions.

🧱 Barème / Grading (100 pts)

Index+TF-IDF 30 • Évaluation 25 • Rocchio 15 • Ablation 20 • Rapport 10

📜 Intégrité & Ressources / Integrity & Resources

Citez les corpus et bibliothèques utilisés. Code non cité = 0.