M2 SII
ANNEE : 2025/2026

RAPPORT DU PROJET

MODULE : RECHERCHE D'INFORMATIONS

Réalisé par les étudiantes :

- ANANE Dalia Khadidja
- ABDELLAZIZ Nassila
- AKMOUNE Feriel
- BENBOUCHAMA Fatima Zahra
- BOKHTACHE Khawla
- MOUSSAOUI Sarah
- THAROUMA Ryma

Table des matières

1. Introduction

L'objectif principal de ce projet est d'évaluer et de comparer différentes approches de modèles de recherche d'information sur un corpus réel de documents médicaux (MEDLINE).
Le travail s'appuie sur les implémentations réalisées au cours de ce module pour les divers modèles de recherche d'information, en appliquant une méthodologie d'évaluation rigoureuse basée sur des métriques standard telles que la précision, le rappel, le F1-score, le MAP, le nDCG, etc.
L'enjeu est double :

- Évaluer la performance de onze modèles de recherche (du modèle vectoriel aux modèles de langue) sur un benchmark standard.
- Comparer leurs résultats pour identifier le modèle le plus efficace dans un contexte réel.

Le projet comprend également le développement d'une interface utilisateur permettant de visualiser les résultats de recherche et les métriques d'évaluation, ainsi qu'une section sur l'apprentissage pour le classement (Learning to Rank).
À travers cette évaluation comparative, nous visons à renforcer notre compréhension des forces et faiblesses des différents modèles de RI, tout en consolidant nos compétences en implémentation, évaluation et analyse de systèmes de recherche

1.1. Organisation du travail et contributions des membres

Les différentes tâches du projet ont été réparties comme suit :

- Dalia Khadidja ANANE : Chargement du dataset et prétraitement
- Sarah MOUSSAOUI : Intégration des modèles et mise en place du pipeline expérimental
- Nassila ABDELLAZIZ : Métriques d'évaluation de base
- Fatima Zahra BENBOUCHAMA : Métriques d'évaluation avancées
- Feriel AKMOUNE : DCG, nDCG et Gain (%) et modèle LTR
- Ryma THAROUMA : Visualisation et graphiques
- Khawla BOKHTACHE : Interface utilisateur

2. Jeu de données et Prétraitement

2.1. Acquisition et analyse syntaxique du jeu de données

Les expérimentations ont été réalisées à partir du **jeu de données MEDLINE**, un benchmark standard largement utilisé en recherche d'information, fourni par l'Université de Glasgow.
Le jeu de données est distribué sous la forme d'une archive compressée (.tar.gz). Après décompression, les fichiers suivants ont été obtenus :

- **MED.ALL** : collection de 1 033 documents textuels (résumés médicaux), utilisée pour l'indexation.
- **MED.QRY** : ensemble de 30 requêtes servant à l'évaluation des systèmes.
- **MED.REL** : jugements de pertinence indiquant les documents pertinents pour chaque requête.
- **MED.REL.OLD** : fichier de pertinence obsolète, ignoré dans ce travail.

Seuls les fichiers MED.ALL, MED.QRY et MED.REL ont été utilisés dans la suite des expérimentations.

a) Analyse des documents
Le fichier MED.ALL a été analysé afin d'extraire, pour chaque document : un identifiant unique (doc_id) et le contenu textuel brut associé.
Les métadonnées non pertinentes pour la recherche d'information ont été ignorées. Les documents ont ensuite été stockés sous la forme d'une structure (dictionnaire) associant chaque identifiant à son texte brut.

b) Analyse des requêtes
Les requêtes contenues dans le fichier MED.QRY ont été traitées de manière similaire. Pour chaque requête, les éléments suivants ont été extraits : un identifiant de requête (query_id) et le texte de la requête.
Les requêtes ne sont pas indexées mais seront ultérieurement traitées avec le même pipeline de prétraitement que les documents.

c) Jugements de pertinence
Le fichier MED.REL a été analysé afin d'extraire les paires (query_id, doc_id) correspondant aux documents jugés pertinents pour chaque requête.
Ces informations ont été stockées sous forme d'une association entre chaque requête et l'ensemble des documents pertinents correspondants. Elles sont utilisées exclusivement lors de la phase d'évaluation et pour les modèles exploitant des informations de pertinence.

2.2. Prétraitement des textes

Un pipeline de prétraitement unique a été appliqué à l'ensemble des documents et des requêtes afin de garantir la cohérence des représentations textuelles.
Les étapes de prétraitement sont les suivantes :
a) Tokenisation :
Les textes ont été segmentés en unités lexicales à l'aide de la même expression régulière que celle utilisée lors des travaux pratiques précédents.
b) Mise en minuscules :
Tous les tokens ont été convertis en minuscules.
c) Suppression des mots vides (stopwords) :
Les mots fonctionnels fréquents ont été supprimés en utilisant la même liste de stopwords que précédemment.
d) Racinisation (stemming) :
Un algorithme de Porter a été appliqué afin de réduire les tokens à leur racine morphologique.

À l'issue de ce prétraitement, les documents et les requêtes sont représentés par des séquences de tokens normalisés. Les résultats ont été sauvegardés dans des fichiers.JSON afin d'éviter tout recalcul inutile.

2.3. Calcul des fréquences et statistiques de collection

À partir des documents prétraités, différentes statistiques nécessaires à l'indexation et aux modèles de recherche ont été calculées.

Fréquence des termes (TF):
Pour chaque document, la **fréquence des termes (TF)** a été calculée comme le nombre d'occurrences de chaque terme dans le document.
La fréquence maximale par document a également été enregistrée afin de permettre la normalisation des fréquences.
La fréquence normalisée est utilisée afin de réduire l'influence de la longueur des documents elle a été calculée selon la formule suivante :

$$TF_{norm}(t, d) = \frac{TF(t,d)}{TF_{max\ t' \in d}(t', d)}$$

Fréquence documentaire et IDF :
La **fréquence documentaire (DF)** d'un terme correspond au nombre de documents dans lesquels ce terme apparaît.
À partir de cette valeur, l'**inverse de la fréquence documentaire (IDF)** a été calculé conformément à la formule présentée dans le cours, et utilisé dans le calcul des pondérations TF–IDF.

$$idf_i = log\ (\frac{N}{ni} + 1)$$

2.4. Construction des structures d'indexation

Vocabulaire et index des documents
Un vocabulaire global a été construit en attribuant un indice unique à chaque terme distinct de la collection.
De la même manière, un index des documents a été créé afin d'associer chaque identifiant de document à une position de ligne dans les matrices de représentation.
Matrice document–terme
Une **matrice document–terme** a été construite où :

- chaque ligne représente un document ;
- chaque colonne représente un terme du vocabulaire ;
- chaque cellule contient la le poid TF-IDF du terme dans le document.

$$weight\ (ti,\ dj) = TF(ti,\ dj) \times IDF(ti)$$

Une version binaire de cette matrice, indiquant uniquement la présence ou l'absence des termes, a également été générée.
Ces matrices ont été stockées sous forme creuse (CSR) afin d'optimiser l'utilisation de la mémoire.
Index inversé
Un **index inversé** a été construit afin d'associer chaque terme à la liste des documents dans lesquels il apparaît, accompagnée du poid correspondant.
Les listes de postings ont été triées par identifiant de document afin d'assurer un accès déterministe et reproductible.

2.5. Entrées finales prêtes à l'emploi pour les modèles de recherche

À l'issue des différentes étapes de chargement, de prétraitement et d'indexation, l'ensemble des données a été transformé en représentations structurées directement exploitables par les modèles de recherche d'information et par les modules d'évaluation. Un ensemble complet de **statistiques de fréquence** a été calculé à partir de la collection, incluant la fréquence des termes par document, la fréquence maximale par document, les fréquences normalisées, la fréquence documentaire ainsi que les fréquences globales au niveau de la collection. Ces statistiques constituent les fondements des schémas de pondération TF, TF normalisé et TF–IDF, et sont nécessaires au bon fonctionnement des modèles vectoriels, probabilistes et de langage.

L'indexation de la collection repose sur la construction de plusieurs **structures fondamentales** :

- les documents et les requêtes prétraités, stockés sous forme de séquences de tokens associées à leurs identifiants respectifs, ainsi que les jugements de pertinence reliant chaque requête aux documents pertinents ;
- une matrice document–terme creuse (CSR) basée sur les poids des termes, représentant les fréquences des termes dans les documents ;
- une version binaire de la matrice document–terme, indiquant uniquement la présence ou l'absence des termes dans chaque document ;
- un vocabulaire global associant chaque terme distinct de la collection à un indice de colonne unique ;
- un index des documents permettant d'établir une correspondance explicite entre les identifiants réels des documents et leurs indices de lignes dans les matrices ;
- un index inversé reliant chaque terme à la liste des documents dans lesquels il apparaît, accompagné des valeurs de pondération correspondantes ;
- les longueurs individuelles des documents, exprimées en nombre de tokens après prétraitement, ainsi que la longueur moyenne des documents de la collection ;
- le nombre total de documents de la collection, utilisé dans le calcul des pondérations globales ;
- la fréquence maximale des termes par document, nécessaire à la normalisation des fréquences locales ;
- les fréquences des termes normalisées par document, permettant de réduire l'influence des documents très longs ;
- la fréquence documentaire de chaque terme, correspondant au nombre de documents dans lesquels il apparaît ;
- la fréquence totale de chaque terme à l'échelle de la collection, utilisée par les modèles probabilistes et les modèles de langage ;
- la fréquence totale des termes normalisée à l'échelle de la collection, facilitant certaines comparaisons statistiques globales.

Cette organisation garantit une **base commune, cohérente et reproductible**, assurant une comparaison équitable des performances des différents systèmes de recherche d'information évalués dans ce travail.

3. Modèles de Recherche d'Information Évalués

3.1 Modèles vectoriels

3.1.1. Vector Space Model (Cosine Similarity)

Le **Vector Space Model (VSM)** représente chaque document et chaque requête sous la forme de vecteurs dans un espace de termes. Chaque dimension correspond à un terme du vocabulaire, pondéré par un schéma de pondération (ici tf-idf).
La pertinence d'un document par rapport à une requête est mesurée par la **similarité cosinus**, qui correspond au cosinus de l'angle entre les deux vecteurs.
Formule :

$$\cos(\theta) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \, \|\vec{v}\|}$$

où u et v sont les vecteurs tf-idf du document et de la requête.
Paramètres utilisés :

- Poids des termes : tf-idf binaire pour la requête, tf-idf pour les documents.

Implémentation :

- Implémenté dans vsm_rank().
- Représente chaque document et requête comme un vecteur tf-idf, puis calcule la similarité cosinus pour le classement.

3.1.2. Latent Semantic Indexing (k = 100)

Le **LSI** réduit la dimensionnalité de l'espace vectoriel via **SVD (Singular Value Decomposition)**, capturant les relations sémantiques implicites entre termes et documents. Les documents et requêtes sont projetés dans un espace latent de dimension k.
Formules :

$$W_k \approx T_k \times S_k \times D_k$$

$$M = T_k \times S_k^{-1} \quad Q_{new} = Q^T \cdot M = Q^T \cdot T \cdot S^{-1}$$

où :

- T_k est la matrice des **termes (T)**,
- S_k est la matrice diagonale des **valeurs singulières**
- D_k est la matrice des **documents (D)**
- Q_{new} est la requête projetée dans l'espace talent.

Paramètres utilisés :

- Nombre de dimensions latentes : k=100

Implémentation :

- Implémenté avec train_lsi() et lsi_rank().
- Réduit la dimensionnalité via SVD (k=100), capture les relations sémantiques entre termes et documents, puis calcule la similarité cosinus dans l'espace latent.

3.2 Modèles probabilistes

3.2.1. Classic BIR (avec et sans pertinence)

Le **Binary Independence Retrieval (BIR)** est un modèle probabiliste qui estime la probabilité qu'un document soit pertinent pour une requête, en supposant l'indépendance des termes et une représentation binaire de leur présence.

Formule sans pertinence :

$$RSV(d,q) = \sum_{i \in q \cap d} \log \frac{N - n_i + 0.5}{n_i + 0.5}$$

Formule avec pertinence (feedback) :

$$RSV(q,d) = \sum_{t_i \in (d \cap q)} \log \frac{\frac{r_i + 0.5}{R - r_i + 0.5}}{\frac{(n_i - r_i + 0.5)}{(N - n_i - R + r_i + 0.5)}}$$

où :

- N = nombre total de documents
- n_t = nombre de documents contenant le terme t
- R = nombre de documents pertinents connus
- r_t= nombre de documents pertinents contenant t

Implémentation :

- Fonction : bir_rank()
- Calcule le score RSV (Retrieval Status Value) basé sur la présence binaire des termes dans les documents.
- Avec pertinence : utilise la rétroaction sur les documents pertinents via qrels.

3.2.2. Extended BIR (avec et sans pertinence)

Le **Extended BIR** améliore le BIR classique en **pondérant chaque terme par son poids tf-idf** dans le document. La formule reste similaire au BIR classique, mais chaque logarithme est multiplié par le poids $w_{t,d}$ (tf-idf) du terme.

Implémentation :

- Fonction : ext_bir_rank()
- Amélioration du BIR classique : intègre le poids tf-idf des termes.
- Avec pertinence : prend en compte la rétroaction sur les documents pertinents.

3.2.3. BM25 (k = 1.2, b = 0.75)

Le **BM25** est un modèle probabiliste robuste prenant en compte la fréquence des termes et la longueur des documents.

Formule :

$$RSV_{BM25}(q,d) = \sum_{i \in q} \log \frac{N - n_i + 0.5}{n_i + 0.5} \times \frac{(k_1 + 1)\, tf_i}{k_1 \left((1-b) + b\frac{dl}{avdl}\right) + tf_i}$$

où :

- $tf_{t,d}$: fréquence du terme t dans le document d
- |d| : longueur du document
- avgdl: longueur moyenne des documents

Paramètres utilisés :

- $k_1 = 1{,}2$
- b=0.75

Implémentation :

- Fonction : bm25_score()
- Modèle probabiliste moderne et robuste, adapté aux collections de taille variable.

3.3 Modèles de langage

3.3.1. MLE

Le modèle de langage **MLE** estime la probabilité qu'une requête soit générée par un document à partir de la fréquence relative des termes.

$$\text{RSV(Q,d)} = P(Q \mid M_d) = \prod_{t \in Q} P(t \mid M_d) = \prod_{t \in Q} \frac{tf(t,d)}{|d|}$$

où :

- $tf_{t,d}$ est la fréquence du terme t dans le document,
- |d| est la longueur du document.

Implémentation :

- Fonction mle_score()
- Le score correspond au produit des probabilités des termes de la requête dans le document.

3.3.2. Add-1 (Laplace)

Le **lissage de Laplace** évite les probabilités nulles en ajoutant 1 à chaque fréquence (taille du vocabulaire |V| est ajouté au dénominateur).

Implémentation :

- Fonction laplace_score()
- Ce lissage garantit que tous les termes ont une probabilité non nulle.

3.3.3. Jelinek–Mercer (λ = 0.2)

Le **lissage Jelinek–Mercer** combine la probabilité d'un terme dans le document et dans l'ensemble de la collection :

$$P_{JM}(w \mid d) = \lambda\, P_{MLE}(w \mid d) + (1 - \lambda)\, P_{MLE}(w \mid C)$$

- λ=0.2
- $P_{MLE}(w,d)$ est équivalent au $P(Q,M_d)$
- |C| = longueur totale de la collection

Implémentation:

- Fonction : jm_score()
- Interpolation entre la probabilité dans le document et la probabilité dans la collection globale.
- Paramètre λ=0.2 pour pondérer le document à 20% et la collection à 80%.

3.3.4. Dirichlet

Le lissage Dirichlet pondère la probabilité du terme dans le document avec la probabilité de la collection selon un paramètre μ :

$$RSV(Q, d) = \prod_{w \in Q} P_{Dir}(w \mid d)$$

$$P_{Dir}(w \mid d) = \frac{tf(w, d) + \mu P_{MLE}(w \mid C)}{\mid d \mid + \mu}$$

où :

- μ=0.3×longueur moyenne des documents

Implémentation:

- Fonction : dirichlet_score()
- Modèle de langage bayésien robuste, équilibrant information locale (document) et globale (collection).

3.4. Exécution du pipeline expérimental

L'ensemble du pipeline d'évaluation des modèles de recherche d'information est orchestré par la fonction run_all_models.

Cette fonction assure l'exécution complète et automatisée de tous les modèles implémentés, garantissant une évaluation cohérente et reproductible.

Pour chaque requête de la collection, run_all_models applique successivement l'ensemble des modèles étudiés (modèles vectoriels, probabilistes et modèles de langage).

Chaque modèle calcule un score de pertinence pour tous les documents de la collection, puis génère un **classement (ranking)** des documents par ordre décroissant de score.

Les résultats produits par run_all_models incluent :

- les scores de pertinence associés à chaque document pour chaque requête,
- les listes de documents classés (ranked lists) pour chaque modèle,
- l'ensemble des résultats nécessaires à l'évaluation comparative des modèles.

Ainsi, run_all_models constitue le point central du système, assurant l'exécution du pipeline de bout en bout, depuis le calcul des scores jusqu'à la génération des classements finaux pour toutes les requêtes et tous les modèles.

4. Learning to Rank

L'apprentissage pour le réordonnancement (Learning to Rank - LTR) constitue une approche avancée en recherche d'information qui vise à optimiser le classement des documents en combinant intelligemment plusieurs modèles de recherche. Contrairement aux méthodes traditionnelles qui utilisent un seul algorithme de scoring, le LTR apprend à pondérer et à fusionner les scores de multiples modèles pour produire un classement final plus performant. Cette approche s'appuie sur des techniques d'apprentissage automatique supervisé, où le système apprend à partir d'exemples annotés de pertinence document-requête. Dans notre implémentation, nous avons développé un système LTR qui exploite les résultats de 11 modèles de recherche différents, transformant leurs scores individuels en features pour un modèle d'apprentissage qui prédit la pertinence optimale des documents.

4.1. Architecture du système

4.1.1 Collecte des données

- Source des labels : on exploite le fichier qrels.json contenant les documents pertinents pour chaque requête.
- Modèles utilisés : 11 modèles de recherche (BIR, BM25, LSI, VSM, modèles de langue, etc.)
- Format des données : Pour chaque (requête, document), collecte des scores de tous les modèles.

4.1.2 Création du dataset d'entraînement

La construction d'un dataset d'apprentissage de qualité représente une étape cruciale dans le processus LTR. Nous avons utilisé le fichier qrels.json comme source de vérité terrain, contenant les jugements de pertinence binaires pour chaque requête. Pour chaque paire (requête, document) pertinente, nous avons extrait les scores attribués par l'ensemble des 11 modèles de recherche implémentés précédemment. Pour équilibrer le dataset, nous avons également collecté des exemples négatifs en sélectionnant des documents bien classés par les modèles mais absents des jugements de pertinence. Cette approche a permis de créer un dataset structuré où chaque échantillon comprend l'identifiant de la requête, l'identifiant du document, un vecteur de 11 scores modèles, et un label de pertinence (1 pour pertinent, 0 pour non pertinent) de la forme suivante :

```
Dataset LTR = {
   "query_id": "1",
   "doc_id": "13",
   "features": [score_modèle1, score_modèle2, ..., score_modèle11],
   "relevance_label": 1 (pertinent) ou 0 (non pertinent)
}
```

4.1.3. Méthode adoptée

Le système développé repose sur une approche Learning to Rank (LTR) supervisée de type pointwise, visant à apprendre une fonction de score permettant de classer les documents par ordre de pertinence pour chaque requête.

L'architecture globale du système est structurée en plusieurs modules indépendants et complémentaires :

- **Chargement et préparation des données**

Les données sont chargées à partir d'un fichier JSON contenant, pour chaque couple requête–document, un vecteur de caractéristiques issu des modèles de recherche d'information ainsi qu'un label de pertinence binaire.
Les données sont ensuite organisées de manière à permettre un traitement au niveau des requêtes, conformément aux bonnes pratiques en recherche d'information.

- **Séparation entraînement / test par requêtes**

Afin d'éviter toute fuite d'information, la séparation des données est effectuée au niveau des requêtes. Toutes les paires requête–document associées à une même requête appartiennent exclusivement soit à l'ensemble d'apprentissage, soit à l'ensemble de test. Cette stratégie garantit une évaluation plus réaliste du modèle de ranking.

- **Gestion du déséquilibre des classes**

Le dataset présente un fort déséquilibre entre documents pertinents et non pertinents.
Pour y remédier, un sous-échantillonnage de la classe majoritaire est appliqué lors de l'entraînement, en conservant l'ensemble des exemples pertinents et en limitant le nombre d'exemples non pertinents selon un ratio contrôlé. Cette étape permet d'améliorer la capacité du modèle à détecter les documents pertinents.

- **Modèle LTR pointwise**

Le modèle est une implémentation pointwise de Learning-to-Rank (LTR), ce qui signifie qu'il traite chaque paire requête-document indépendamment pour prédire une probabilité de pertinence.
Le modèle utilisé est une régression logistique, entraînée par descente de gradient, dont l'objectif est d'estimer directement la probabilité de pertinence d'un document pour une requête donnée. Cette probabilité est ensuite exploitée comme score de ranking.
Afin d'assurer la stabilité numérique et la robustesse de l'apprentissage, plusieurs mécanismes sont intégrés :

- une normalisation des caractéristiques, permettant d'harmoniser les échelles des différentes sources de scores ;
- une fonction sigmoïde numériquement stable, limitant les problèmes d'overflow lors du calcul des probabilités, qui transforme le score linéaire combiné ($z = w \cdot x + b$) en une probabilité comprise entre 0 et 1 et qui classe ensuite selon un seuil de décision qui par défaut est à 0.5 pour décider entre classe 0 (non pertinent) et 1 (pertinent);
- une fonction de coût basée sur la log-loss, enrichie par une régularisation L2 afin de limiter le sur-apprentissage ;
- un ajustement adaptatif du taux d'apprentissage, combiné à un arrêt précoce, garantissant une convergence stable du modèle.

Le modèle apprend à prédire la probabilité de pertinence d'un document pour une requête donnée.

- **Phase d'entraînement et d'évaluation**

Le modèle est entraîné sur l'ensemble d'apprentissage équilibré, puis évalué sur l'ensemble de test à l'aide de métriques de classification telles que l'accuracy, le F1-score, ainsi que le rapport de classification.
Ces métriques permettent d'analyser le compromis précision–rappel, particulièrement important dans un contexte de ranking.

- **Génération du ranking**

Une fois entraîné, le modèle est appliqué à l'ensemble des documents de chaque requête.
Les probabilités prédites sont utilisées comme scores LTR, permettant de classer les documents par ordre décroissant de pertinence. Un rang est ensuite attribué à chaque document.

4.2. Résultats et comparaison avec les modèles classiques

Les expérimentations ont été menées sur un dataset composé de 30 990 couples requête–document, répartis sur 30 requêtes distinctes. La séparation entre les ensembles d'apprentissage et de test a été réalisée au niveau des requêtes, avec respectivement 24 requêtes pour l'entraînement et 6 requêtes pour l'évaluation, garantissant ainsi une évaluation réaliste du système de ranking.

- **Analyse des performances du modèle LTR**

Le dataset présente un fort déséquilibre entre documents pertinents et non pertinents, avec seulement 2,2 % de documents pertinents. Afin de pallier ce déséquilibre, un sous-échantillonnage de la classe majoritaire a été appliqué lors de l'apprentissage.
Après entraînement, le modèle LTR obtient une accuracy de 0,94 et un F1-score de 0,36 sur l'ensemble de test. Bien que l'accuracy soit élevée, cette métrique reste peu représentative dans un contexte de recherche d'information fortement déséquilibré. Le F1-score, ainsi que l'analyse du rappel, fournissent une évaluation plus pertinente du comportement du modèle.
Le rapport de classification met en évidence un rappel élevé pour la classe pertinente (0.82), indiquant que le modèle parvient à identifier la majorité des documents pertinents. En revanche, la précision reste plus faible (0.23), ce qui s'explique par la volonté du modèle de privilégier le rappel, stratégie couramment adoptée en recherche d'information afin d'éviter l'exclusion de documents potentiellement pertinents en amont du ranking.

```
Rapport de classification:
              precision    recall  f1-score   support

Non pertinent       1.00      0.95      0.97      6090
    Pertinent       0.23      0.82      0.36       108
```

- **Comparaison avec les modèles classiques de recherche d'information**

Contrairement aux modèles classiques utilisés individuellement (tels que VSM Cosine, LSI, modèles de langage ou BIR), le modèle LTR proposé ne se base pas sur un seul critère de pertinence. Il apprend automatiquement une combinaison linéaire optimisée des scores produits par ces différents modèles.
L'analyse des poids appris par le modèle montre que certains modèles classiques contribuent davantage au score final. En particulier, LSI_k100 apparaît comme la caractéristique la plus influente, suivi par VSM Cosine et LM Laplace. Ces résultats confirment que les modèles sémantiques et statistiques jouent un rôle central dans l'estimation de la pertinence, tout en bénéficiant de la complémentarité apportée par les autres approches.

- **Impact sur le ranking**

Les probabilités prédites par le modèle LTR sont utilisées comme scores de classement, permettant d'ordonner les documents pour chaque requête. Les résultats montrent que les documents pertinents tendent à apparaître en tête du classement, comme illustré par les premiers rangs obtenus pour plusieurs requêtes.
Ainsi, le modèle LTR permet non seulement d'améliorer la détection des documents pertinents, mais également de produire un classement plus cohérent que celui obtenu par l'utilisation isolée d'un modèle classique.

- **Synthèse**

En résumé, les résultats obtenus montrent que l'approche LTR proposée offre une amélioration qualitative du ranking en exploitant de manière conjointe plusieurs modèles de recherche d'information. Bien que les métriques de classification traduisent un compromis précision–rappel typique des systèmes IR, l'utilisation d'un modèle LTR permet de dépasser les limites des approches classiques prises individuellement, en fournissant un score de pertinence global mieux adapté au classement des documents. Il convient toutefois de noter que ces résultats ont été obtenus sur un jeu de données de taille relativement limitée, tant en nombre de requêtes qu'en diversité de jugements de pertinence. Néanmoins, les performances observées restent cohérentes et illustrent le potentiel de l'approche LTR, qui pourrait être davantage renforcée par l'utilisation de datasets plus larges et plus variés.

5. Métriques d'Évaluation

5.1 Métriques classiques

5.1.1. Précision

La précision (Precision) mesure la proportion de documents pertinents parmi l'ensemble des documents retournés par le système. Elle évalue donc la capacité du modèle à fournir des résultats exacts, en limitant le nombre de documents non pertinents présentés à l'utilisateur.
La précision est définie par la formule suivante :

$$\text{Précision} = \frac{|\text{Documents pertinents retrouvés}|}{|\text{Documents retrouvés}|}$$

Une valeur élevée de la précision indique que la majorité des documents retournés par le système sont pertinents. Cette métrique est particulièrement importante lorsque l'utilisateur souhaite éviter les résultats non pertinents.

5.1.2. Recall

Le rappel (Recall) mesure la proportion de documents pertinents retrouvés par le système par rapport au nombre total de documents pertinents existants dans la collection. Il permet d'évaluer la capacité du modèle à couvrir l'ensemble des documents pertinents. Le rappel est défini par la formule suivante :

$$\text{Recall} = \frac{|\text{Documents pertinents retrouvés}|}{|\text{Documents pertinents existants}|}$$

Un rappel élevé signifie que le système parvient à récupérer une grande partie des documents pertinents, même si cela peut parfois se faire au détriment de la précision.

5.1.3. F1-Score

Le F1-Score est une métrique synthétique qui combine la précision et le rappel en une seule mesure. Il correspond à la moyenne harmonique de ces deux métriques et permet d'obtenir un compromis entre exactitude et exhaustivité. Le F1-Score est défini comme suit :

$$\text{F1-Score} = \frac{2 \times \text{Précision} \times \text{Recall}}{\text{Précision} + \text{Recall}}$$

Cette métrique est particulièrement utile lorsque l'on souhaite équilibrer l'importance de la précision et du rappel, notamment dans des contextes où les classes sont déséquilibrées ou lorsque les erreurs de type faux positifs et faux négatifs ont un impact comparable.

5.2 Métriques basées sur le rang

5.2.1. Precision@K (P@K)

La précision à K (Precision@K) est une métrique d'évaluation qui mesure la proportion de documents pertinents parmi les *K* premiers documents retournés par un système de recherche d'information. Elle permet d'évaluer la capacité du modèle à fournir des résultats pertinents dans les premières positions du classement, qui sont généralement les plus consultées par l'utilisateur.
La Precision@K est définie par la formule suivante :

$$P@K = \frac{|Rel_K|}{K}$$

RelK = nombre de documents pertinents parmi les K premiers résultats.
Dans ce travail, nous avons utilisé deux valeurs de K, à savoir K = 5 et K = 10, afin d'évaluer la qualité des résultats en tête de classement pour chaque modèle.

5.2.2. R-Precision

La R-Precision est une métrique qui mesure la précision obtenue après avoir récupéré *R* documents, où *R* correspond au nombre total de documents pertinents pour une requête donnée. Elle permet ainsi d'évaluer la capacité du système à récupérer l'ensemble des documents pertinents sans fixer arbitrairement une valeur de *K*.
La R-Precision est définie comme suit :

$$R\text{-Precision} = \frac{|Rel_R|}{R}$$

R = nombre total de documents pertinents pour la requête.
Cette métrique est particulièrement utile lorsque le nombre de documents pertinents varie d'une requête à une autre.

5.2.3. Reciprocal Rank (RR) et Mean Reciprocal Rank (MRR)

Le Reciprocal Rank (RR) mesure l'efficacité d'un système de recherche en prenant en compte la position du premier document pertinent dans la liste des résultats. Plus ce document apparaît tôt dans le classement, plus la valeur du RR est élevée.
Il est défini par la formule suivante :

$$RR = \frac{1}{rank_{first}}$$

rankfirst est le rang du premier document pertinent retrouvé.
Le Mean Reciprocal Rank (MRR) correspond à la moyenne du RR calculé sur l'ensemble des requêtes. Cette métrique permet d'évaluer globalement la rapidité avec laquelle un système retourne un premier document pertinent.

5.3 Métriques globales

5.3.1. Average Precision (AP) et Mean Average Precision (MAP)

L'Average Precision (AP) évalue la qualité globale du classement pour une requête donnée en prenant en compte la position de tous les documents pertinents dans la liste de résultats. Elle correspond à la moyenne des valeurs de précision calculées à chaque rang où un document pertinent est retrouvé.

$$AP = \frac{1}{|Rel|} \sum_{k=1}^{N} P(k) \cdot rel(k)$$

La Mean Average Precision (MAP) est ensuite obtenue en calculant la moyenne de l'AP sur l'ensemble des requêtes :

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} AP(q)$$

où *Q* représente l'ensemble des requêtes. Le MAP est une métrique largement utilisée car elle combine à la fois la précision et le rappel tout en tenant compte de l'ordre de classement.

5.3.2. Courbe Precision–Recall interpolée

La courbe Precision–Recall interpolée permet d'analyser le compromis entre la précision et le rappel pour un système de recherche d'information. Pour chaque niveau de rappel standard (0.0, 0.1, …, 1.0), la précision interpolée correspond à la meilleure précision obtenue pour un rappel supérieur ou égal à ce niveau.

$$P_{interp}(r) = \max_{r' \geq r} P(r')$$

Cette interpolation permet de lisser la courbe et de faciliter la comparaison entre différents modèles de recherche d'information.

5.4 Métriques basées sur le gain

5.4.1. DCG

Discounted Cumultative Gain (DCG) est une mesure de la qualité du classement en recherche d'information en cumulant leur pertinence, la pertinence d'un document en position "i" est divisée par un facteur logarithmique croissant avec "i" selon la formule suivante :

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2(i)}$$

- où rel_i est le score de pertinence du document à la position "i" et p est la profondeur d'évaluation qui représente le nombre de documents à considérer.

Pour obtenir la meilleure performance possible pour une requête donnée avec son ensemble de documents pertinents disponibles, on calcule L'IDCG qui est la valeur maximale possible du DCG, obtenue en classant les documents dans l'ordre de pertinence parfaite (du plus pertinent au moins pertinent).

$$IDCG_p = rel_{1\ (le\ plus\ pertinent)} + \sum_{i=2}^{p} \frac{rel_{i\ (trié\ décroissant)}}{log_2(i)}$$

5.4.2. NDCG

le DCG pose problème, une requête avec beaucoup de documents pertinents aura naturellement un DCG plus élevé contrairement à une requête avec peu de documents pertinents qui aura un DCG plus faible. Le NDCG est la version normalisée du DCG qui permet de comparer la qualité de classement entre différentes requêtes, sur une échelle standard de 0 à 1.

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

5.4.3. Gain (%)

Le gain est une mesure pour comparer quantitativement deux systèmes de recherche ou de classement.

$$Gain\ (\%) = \frac{val(A) - val(B)}{val(B)} \times 100$$

- Val(A) = valeur de la métrique (nDCG, MAP, etc.) pour le système A.
- Val(B) = valeur de la métrique pour le système B.

Un gain positif indique que le système B surpasse le système A pour la métrique considérée, tandis qu'un gain négatif montre que le système B est moins performant que le système A. Par exemple, un gain moyen de -20 % suggère que, en moyenne, le système B obtient des résultats inférieurs de 20 % par rapport à A,

tandis qu'un gain de 10 % signifie une amélioration moyenne de 10 %. Cela permet d'évaluer rapidement quelles méthodes sont plus efficaces dans le classement des documents.

6. Résultats Expérimentaux

6.1 Résultats pour les requêtes

6.1.1 Tableau comparatif des performances des modèles

Le tableau (1) suivant présente les résultats moyens obtenus par les différents modèles de recherche d'information sur les requêtes, selon les diverses métriques.

Tableau 1. Comparaison des performances des modèles

Modèle	MAP	MRR	P@5	P@10	r-Precision	DCG@20	nDCG@20
BIR (sans pertinence)	0.4749	0.7854	0.6867	0.5833	0.4589	50.4137	1
BIR (avec pertinence)	0.1056	0.2946	0.2067	0.1700	0.1254	2.2366	0.8666
Extended BIR (sans pertinence)	0.5024	0.8062	0.6333	0.6100	0.5329	38.8841	1
Extended BIR (avec pertinence)	0.1529	0.4098	0.2800	0.2333	0.1847	2.8192	1
BM25	0.4777	0.8753	0.6600	0.5800	0.4964	31.0145	0.9
VSM (Cosine)	0.5139	0.9444	0.7333	0.6333	0.5117	1.6287	0.6666
LSI (k = 100)	**0.6463**	0.9278	**0.8467**	**0.7667**	**0.6235**	**4.6789**	**1**
LM – MLE	0.0590	0.1167	0.0667	0.0633	0.0309	0.0003	0.6666
LM – Laplace	0.4563	0.9043	0.6667	0.5600	0.4428	1.032×10^{-8}	0.6666
LM – Jelinek–Mercer	0.3559	0.5818	0.4800	0.4333	0.3476	1.414×10^{-5}	1
LM – Dirichlet	0.3537	0.5957	0.4667	0.4200	0.3353	4.209×10^{-5}	1

Interprétation

▪ **Précision**

La précision mesure la proportion de documents pertinents parmi l'ensemble des documents retrouvés. Dans nos résultats, la précision moyenne obtenue est égale à **0.02245**.

Cette valeur est **constante pour tous les modèles**, car le nombre total de documents retrouvés correspond au nombre total de documents de la collection, tandis que le nombre de documents pertinents **varie d'une requête à une autre**. Toutefois, comme tous les modèles retournent l'ensemble de la collection, la précision obtenue est **identique pour tous les modèles**. Ainsi, la précision ne dépend pas du modèle utilisé et ne permet pas de différencier leurs performances.

▪ **Recall**

Le recall évalue la capacité du système à retrouver l'ensemble des documents pertinents. Dans notre configuration expérimentale, tous les documents pertinents sont systématiquement retrouvés, ce qui

conduit à une valeur de **recall constante et égale à 1** pour l'ensemble des modèles. Cette métrique ne fournit donc pas d'information discriminante sur la qualité des modèles évalués.

▪ **F1-Score**

Le F1-score combine la précision et le recall afin de fournir une mesure synthétique des performances. Dans nos résultats, le **F1-score est égal à 0.04380** et demeure constant pour tous les modèles. Cette constance s'explique par le fait que la précision et le recall sont eux-mêmes constants. Par conséquent, le F1-score ne permet pas non plus de distinguer les performances des différents modèles dans ce cadre expérimental.

- Une solution consiste à **introduire un seuil sur le score de pertinence** afin de ne conserver que les documents dont le score dépasse une valeur minimale, ou à limiter explicitement le nombre de documents retournés en appliquant une stratégie de **Top-K retrieval**. Ces approches permettent de réduire le nombre de documents non pertinents présentés à l'utilisateur et rendent les métriques classiques, telles que la précision, le recall et le F1-score, à nouveau discriminantes. En particulier, l'utilisation d'un seuil est naturellement adaptée aux modèles probabilistes et aux modèles de langage, tandis que le Top-K est couramment employé avec des modèles vectoriels comme **VSM_Cosine**.

▪ **Mean Average Precision (MAP)**

Le MAP mesure la qualité globale du classement en tenant compte de la position de **tous les documents pertinents**.

Un MAP élevé indique que les documents pertinents apparaissent majoritairement en tête de classement. Dans nos résultats, les modèles probabilistes avancés tels que **BM25** et les **modèles de langage avec lissage** obtiennent généralement les meilleures valeurs de MAP, ce qui confirme leur efficacité sur des collections textuelles réelles comme MEDLINE.

▪ **Mean Reciprocal Rank (MRR)**

Le MRR évalue la rapidité avec laquelle un système retourne **le premier document pertinent**.

Un MRR élevé signifie que l'utilisateur trouve rapidement une réponse pertinente.

Les modèles exploitant la fréquence des termes et le contexte global, notamment **BM25** et **LSI**, présentent de bonnes performances selon cette métrique, contrairement aux modèles binaires simples sans information de pertinence.

▪ **Precision@5 et Precision@10**

Les métriques **P@5** et **P@10** mesurent la précision dans les premières positions du classement, qui sont les plus importantes du point de vue utilisateur.

Les résultats montrent que :

- **BM25** et **Extended BIR avec pertinence** obtiennent les meilleures précisions,
- les modèles sans information de pertinence affichent des performances plus faibles, notamment pour P@5.

Cela indique que l'intégration d'informations statistiques et de pertinence améliore significativement la qualité des premiers résultats.

▪ **R-Precision**

La R-Precision permet d'évaluer la capacité du système à récupérer l'ensemble des documents pertinents pour une requête donnée.

Les modèles utilisant les jugements de pertinence (**BIR avec pertinence**, **Extended BIR avec pertinence**) obtiennent des scores plus élevés, montrant une meilleure couverture des documents pertinents.

▪ **DCG@20**

Le DCG@20 met en évidence des différences importantes entre les modèles en termes de distribution des scores dans les premières positions du classement. Les modèles de type BIR et ExtendedBIR obtiennent les valeurs de DCG les plus élevées, ce qui s'explique par le fait qu'ils produisent des scores numériques élevés et fortement différenciés pour les documents classés en tête. À l'inverse, les modèles de langue (LM) génèrent des scores très faible, ce qui conduit mécaniquement à des DCG proches de zéro. Dans ce contexte, le DCG@20 doit être interprété comme un indicateur descriptif de l'amplitude des scores plutôt que comme une mesure directe de pertinence. Les résultats montrent que BM25 se situe entre les modèles probabilistes BIR et les modèles vectoriels (VSM, LSI). Son DCG@20 relativement élevé traduit une bonne capacité à différencier les documents les plus importants dans le top-20, tout en produisant des scores plus modérés que BIR. Les modèles vectoriels, notamment VSM Cosine et LSI, présentent des DCG plus faibles, ce qui suggère une distribution de scores plus homogène, avec moins de contraste entre documents fortement et faiblement classés.

▪ **NDCG@20**

La majorité des modèles obtiennent un nDCG@20 égal à 1, ce qui indique que le classement observé est identique au classement idéal tel qu'il est défini dans l'expérimentation. Cette situation s'explique par le fait que le nDCG est calculé à partir des scores produits par les modèles eux-mêmes (qui produisent des scores triés dans un ordre décroissant), sans recours à des jugements de pertinence externes, donc le DCG est toujours égale au IDCG. Le nDCG ne mesure donc pas ici la qualité réelle du ranking, mais uniquement la cohérence interne du modèle avec ses propres scores, une solution serai de prendre le top 20 des documents pertinents dans un ordre croissant des scores pour calculer le DCG.

6.1.2 Tableau comparatif des performances des modèles selon le gain

Le tableau (2) ci-dessous présente les gains moyens (%) de nDCG@20 obtenus lors de la comparaison des différents modèles de recherche d'information sur les 10 premières requêtes. Ces gains illustrent l'amélioration ou la perte de performance relative d'un modèle par rapport à un autre.

Tableau 2. Tableau comparatif des performances des modèles selon le gain

A B	BIR (sans pertinence)	BIR (avec pertinence	BM 25	Extended BIR (sans pertinence)	Extended BIR (avec pertinence)	LM-Dirichlet	LM-Laplace	LM-JM	LM-MLE	LSI (k=100)	VSM (cosine)
BIR (sans pertinence)	/	/	/	/	/	/	/	/	/	/	/
BIR (avec pertinence)	-10	/	/	/	/	/	/	/	/	/	/

BM25	0	10	/	/	/	/	/	/	/	/	/
Extended BIR (sans pertinence)	0	10	0	/	/	/	/	/	/	/	/
Extended BIR (avec pertinence)	0	10	0	0	/	/	/	/	/	/	/
LM-Dirichlet	-20	-10	-20	-20	-20	/	/	/	/	/	/
LM-Laplace	0	10	0	0	0	20	/	/	/	/	/
LM-JM	-20	-10	-20	-20	-20	0	-20	/	/	/	/
LM-MLE	-90	-80	-90	-90	-90	-70	-90	-70	/	/	/
LSI (k=100)	0	10	0	0	0	20	0	20	90	/	/
VSM (cosine)	0	10	0	0	0	20	0	20	90	0	/

-> Les gains de nDCG@20 montrent que les modèles BIR et ExtendedBIR sont globalement stables et performants, avec peu de variations même lorsque la pondération par pertinence est ajoutée. BM25 obtient des résultats comparables à BIR, tandis que les modèles de langage classiques comme LM_MLE, LM_Dirichlet ou LM_JelinekMercer présentent des gains fortement négatifs sur certaines requêtes, indiquant leur difficulté à classer correctement les documents pertinents. LM_Laplace reste compétitif, et les modèles vectoriels LSI et VSM atteignent presque systématiquement un nDCG parfait, ce qui explique l'absence de gains lorsqu'ils sont comparés à BIR ou entre eux. Cependant, les valeurs extrêmes (+/-100%) sur certaines requêtes montrent que nDCG peut être très sensible aux échecs totaux et n'est pas toujours fiable pour comparer de façon globale des modèles avec des comportements très différents. En résumé, BIR, ExtendedBIR, BM25, LSI et VSM dominent la performance, mais l'interprétation des gains doit rester prudente en raison des limites de la métrique.

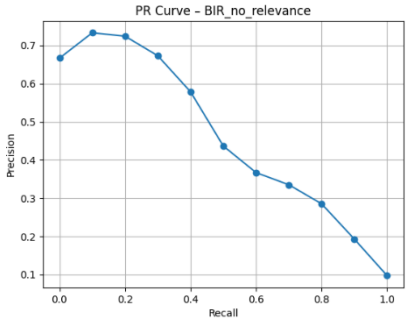
-> Nous tenons à préciser que le détail des résultats de gains nDCG pour chaque requête est disponible, et que nous avons également calculé les gains correspondants pour le DCG. L'ensemble de ces informations

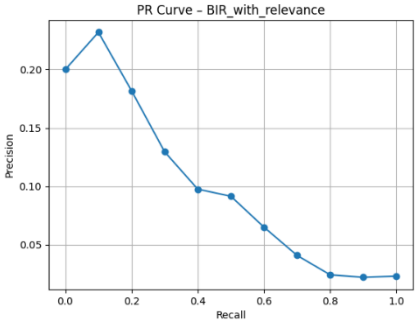
est consultable dans le fichier
SourceCode\evaluation_results\evaluation_results_dcg_ndcg_gain\comparison_report.json.

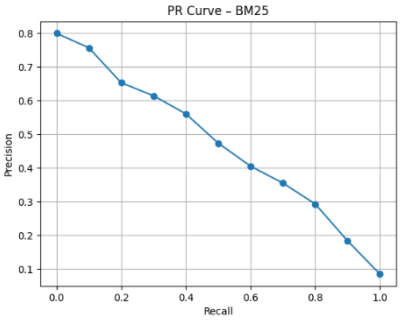
Remarque importante :
Dans la section des résultats expérimentaux, nous avons présenté les valeurs moyennes des métriques par modèle pour l'ensemble des requêtes, afin d'éviter un affichage trop volumineux pour les 30 requêtes. Cependant, les résultats détaillés pour chaque requête sont disponibles dans les fichiers JSON correspondants.

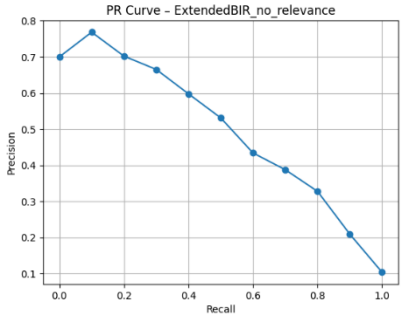
7. Analyse Graphique

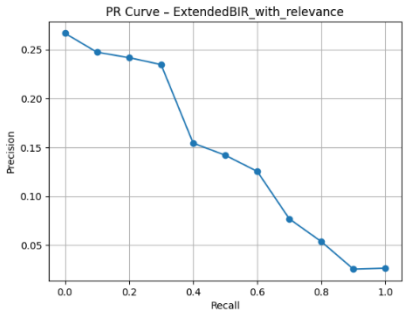
7.1 Courbes Precision–Recall

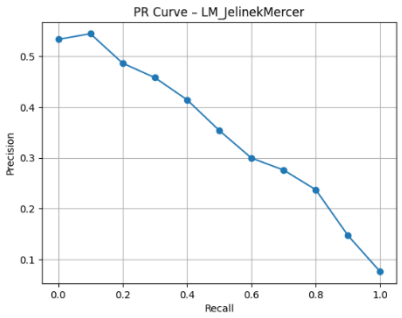


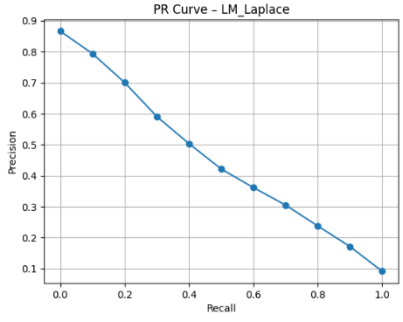


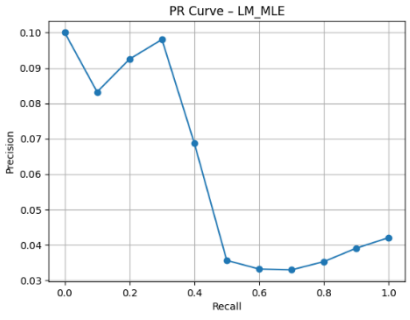


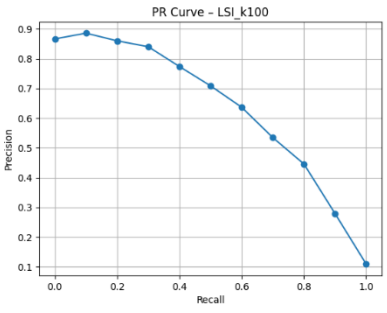


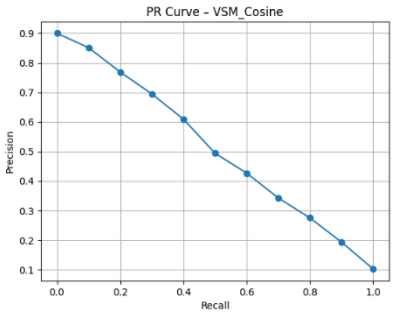


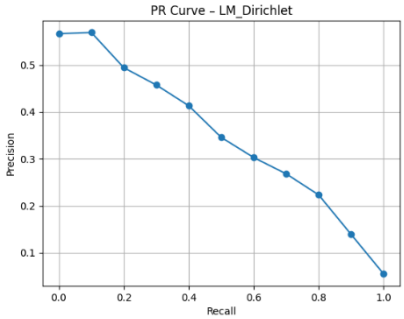




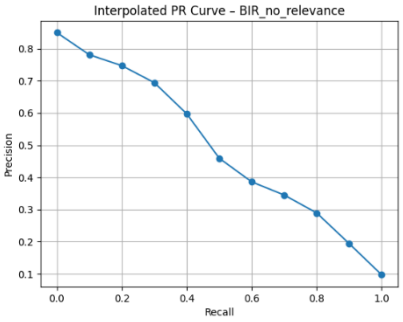


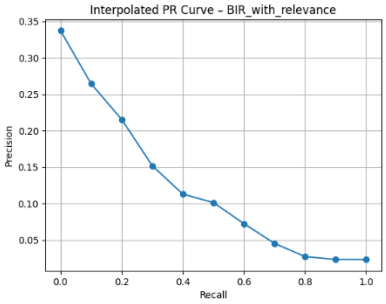


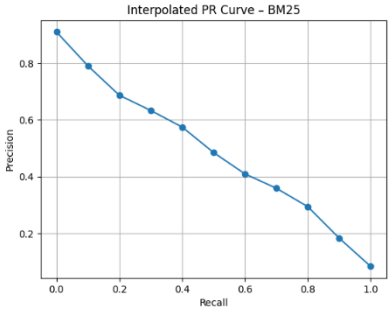


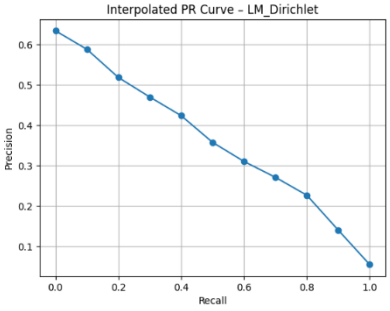


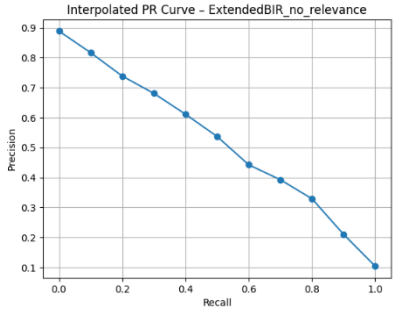
7.2 Courbes Precision–Recall interpolées

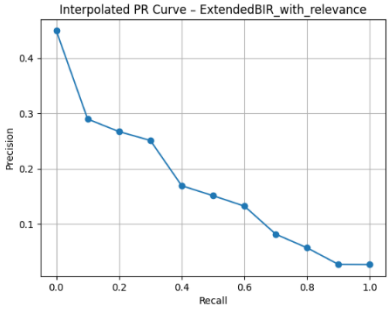


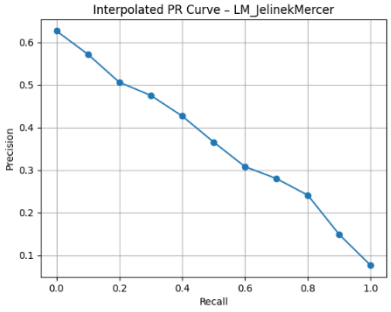


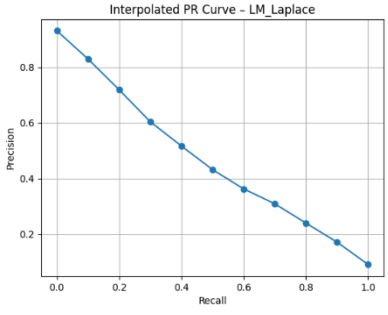


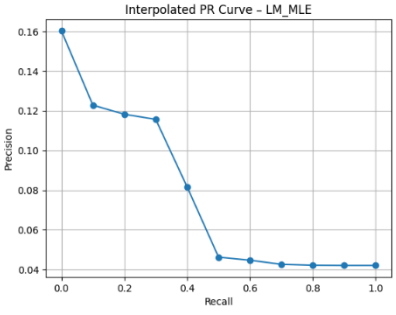


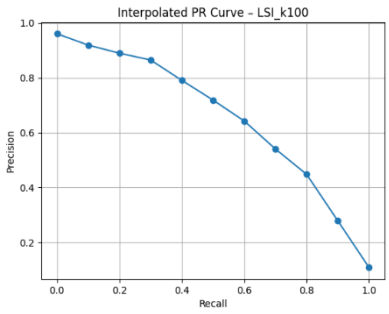


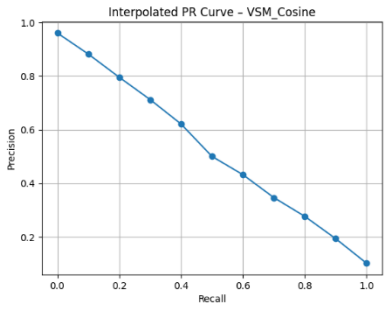












7.3 Comparaison visuelle des modèles

- **Interprétation des courbes Precision–Recall**

Les courbes Precision–Recall illustrent le compromis entre la précision et le rappel pour les différents modèles de recherche d'information évalués.
De manière générale, on observe que la précision diminue lorsque le rappel augmente, ce qui est attendu, car inclure davantage de documents dans les résultats conduit souvent à intégrer des documents moins pertinents.
Les modèles LSI_k100 et VSM_Cosine présentent les meilleures performances globales, avec des courbes situées plus haut que les autres, indiquant une forte capacité à retourner des documents pertinents dès les premiers rangs. Ces résultats confirment leur efficacité en termes de classement.
Les modèles BM25, ExtendedBIR_no_relevance et LM_Laplace offrent des performances intermédiaires, avec une décroissance progressive de la précision lorsque le rappel augmente, ce qui traduit un comportement relativement équilibré.
En revanche, les modèles LM_MLE, BIR_with_relevance et ExtendedBIR_with_relevance affichent des courbes nettement plus basses, révélant une précision faible sur l'ensemble des niveaux de rappel. Cela indique une difficulté à positionner correctement les documents pertinents en tête de classement.
Ces courbes confirment ainsi les résultats obtenus à l'aide des métriques globales (MAP, MRR, P@K), et mettent en évidence la supériorité des modèles basés sur la sémantique latente et la similarité vectorielle dans ce contexte expérimental.

- **Interprétation des courbes Precision–Recall interpolées**

Les courbes Precision–Recall interpolées permettent d'évaluer la qualité du classement de manière plus stable en considérant, pour chaque niveau de rappel standard, la **meilleure précision atteinte pour un rappel supérieur ou égal**. Cette interpolation réduit l'effet des fluctuations locales observées dans les courbes PR classiques.
Les résultats montrent que **LSI_k100** et **VSM_Cosine** dominent clairement les autres modèles, avec des courbes situées très haut sur l'ensemble des niveaux de rappel. Ces modèles conservent une précision élevée même lorsque le rappel augmente, indiquant une excellente capacité à positionner les documents pertinents en tête de classement.
Les modèles **BM25**, **ExtendedBIR_no_relevance** et **LM_Laplace** présentent des performances intermédiaires, avec une décroissance progressive et régulière de la précision. Leur comportement reste globalement stable et cohérent sur l'ensemble des niveaux de rappel.
À l'inverse, **LM_MLE**, **BIR_with_relevance** et **ExtendedBIR_with_relevance** affichent des courbes nettement plus basses, traduisant une précision faible dès les premiers niveaux de rappel. Ces résultats

suggèrent une moins bonne exploitation de l'information de pertinence ou un lissage insuffisant du modèle probabiliste.
Dans l'ensemble, les courbes interpolées confirment les conclusions tirées des métriques quantitatives (MAP, MRR, P@K), en mettant en évidence la supériorité des modèles sémantiques et vectoriels dans ce cadre expérimental

8. Interface Utilisateur et visualisation

8.1. Introduction générale

Ce chapitre décrit l'implémentation de l'interface utilisateur du Système de Recherche d'Information (Information Retrieval System) développée à l'aide du framework Streamlit.
L'objectif principal de cette interface est de permettre :

- la sélection interactive de requêtes et de modèles de recherche,
- l'affichage des résultats classés,
- l'évaluation des performances via différentes métriques,
- la comparaison visuelle entre plusieurs modèles,
- l'analyse des courbes Precision–Recall.

L'interface a été conçue pour être interactive, intuitive et personnalisable, avec un support du mode sombre et clair**.**

8.2. Configuration générale de l'application

La page est configurée pour :

- un affichage en pleine largeur pour exploiter tout l'espace écran,
- une barre latérale ouverte par défaut,
- un titre explicite affiché dans l'onglet du navigateur.

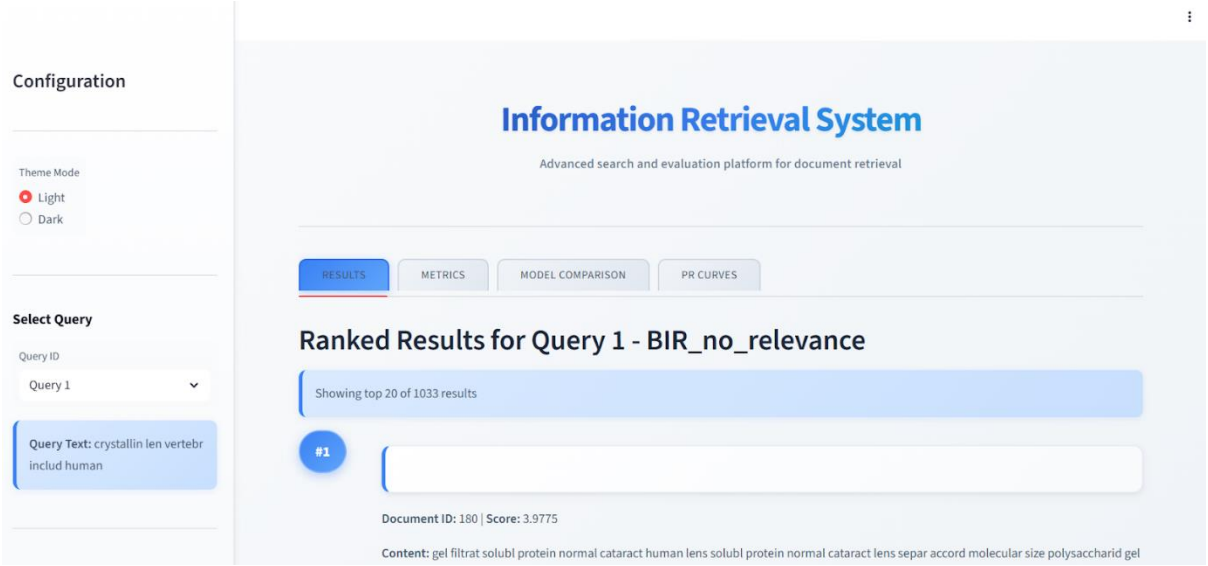


Figure 1 : Page d'accueil de l'application Streamlit

8.3. Gestion du thème : mode sombre et mode clair

L'interface prend en charge deux thèmes :

- Mode sombre : arrière-plan sombre, textes clairs, cartes et résultats avec effets d'ombre, meilleure lisibilité en environnement peu lumineux.
- Mode clair : couleurs claires et professionnelles, mise en valeur des graphiques, design adapté à un usage académique.

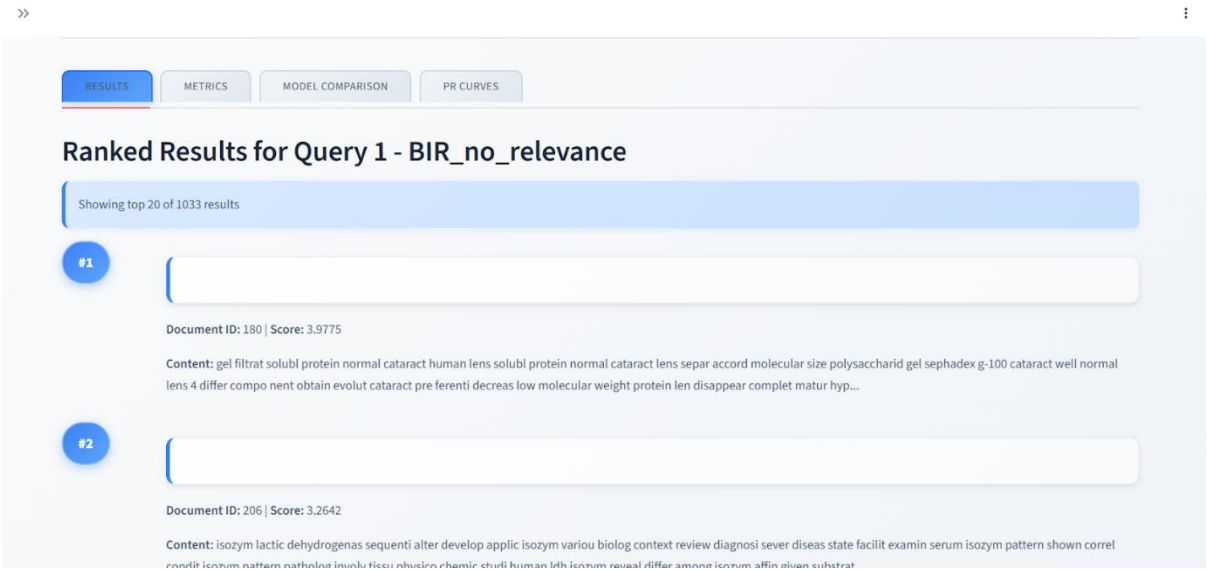


Figure 2 : Interface en mode clair

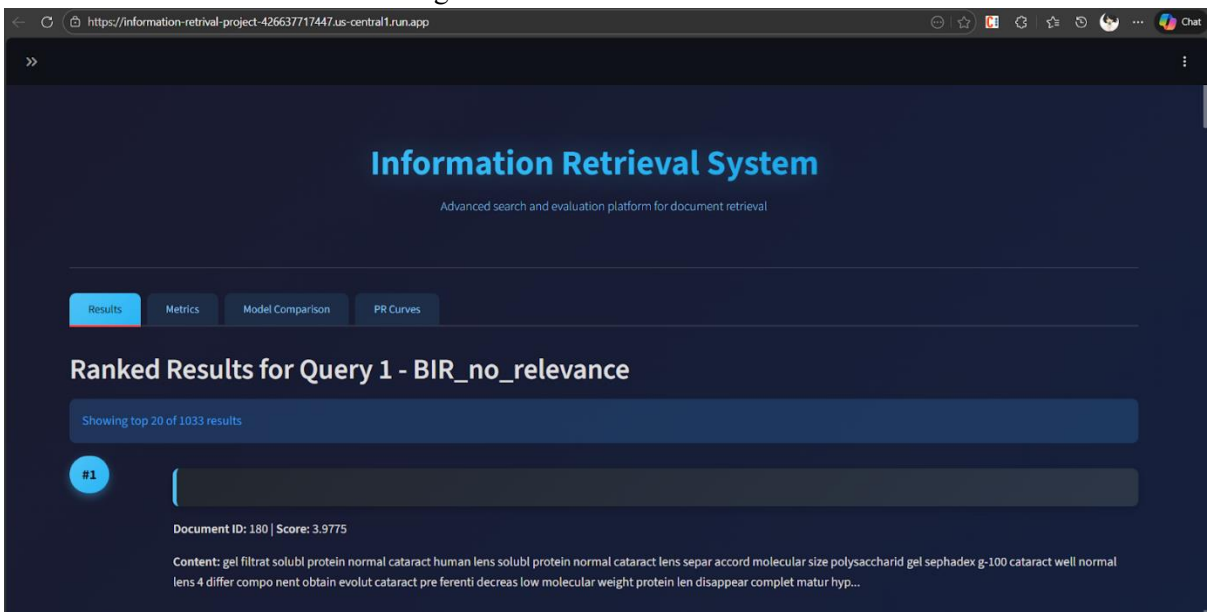


Figure 3 : Interface en mode sombre

8.4. Barre latérale – Configuration utilisateur

8.4.1 Sélection de la requête

L'utilisateur peut choisir une requête parmi celles disponibles. Le texte de la requête sélectionnée est affiché dynamiquement.

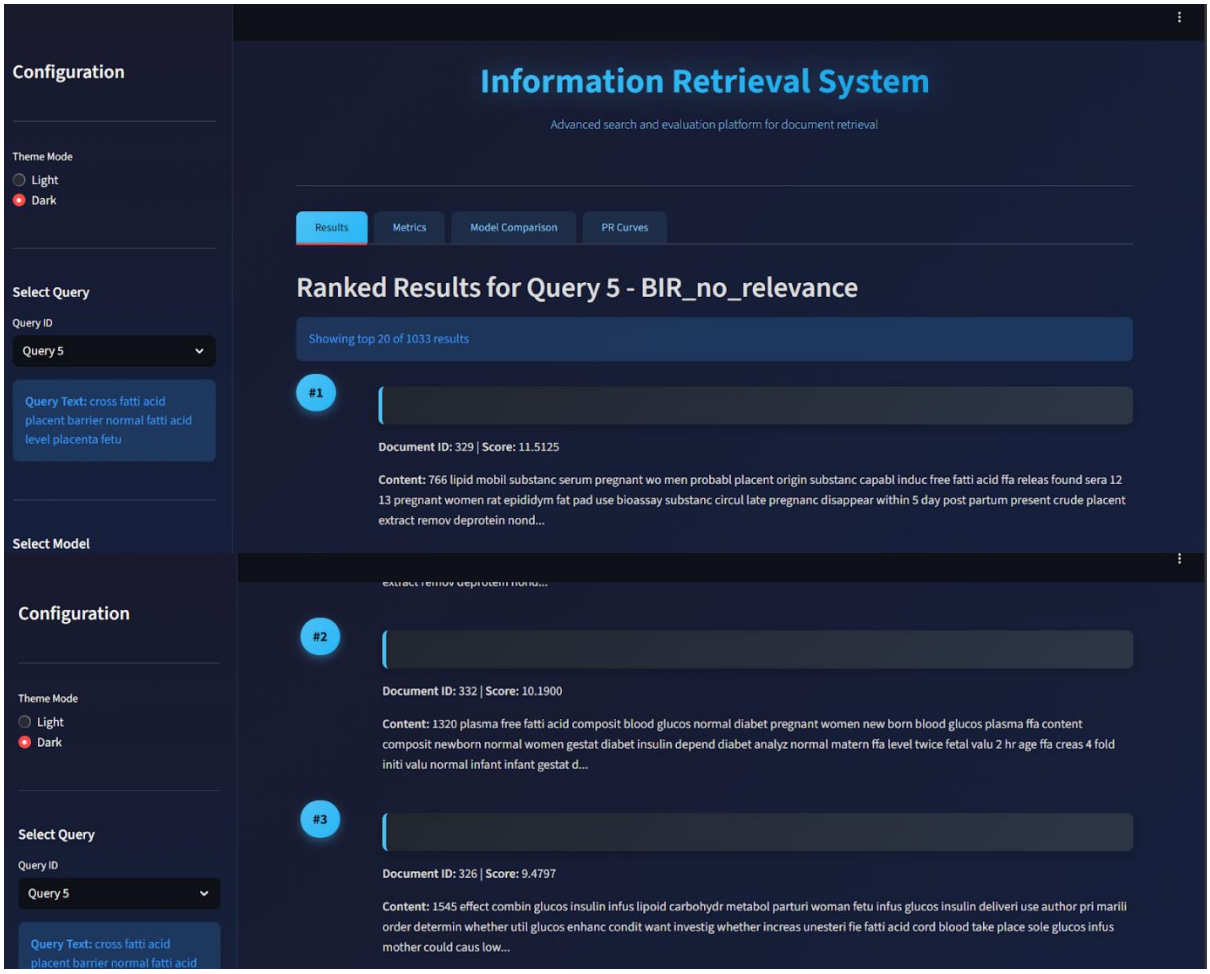


Figure 4 : Sélection d'une requête et affichage de son contenu

8.4.2 Sélection du modèle de recherche

Les modèles de recherche (BM25, VSM, LM, LSI, BIR, etc.) sont sélectionnables.

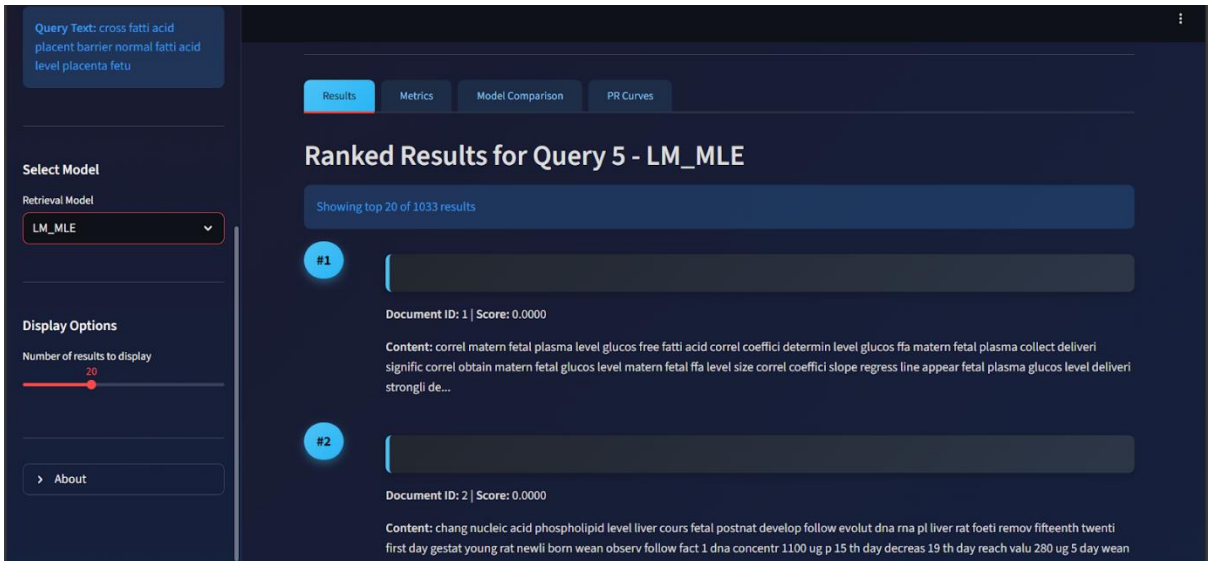
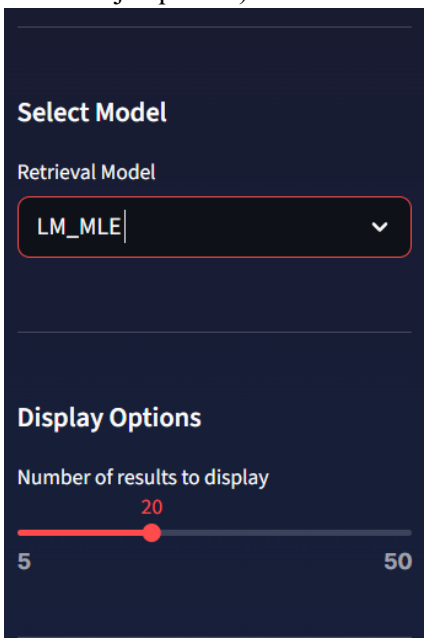


Figure 5 : Sélection du modèle de recherche

8.4.3 Paramètres d'affichage

L'utilisateur définit le nombre de documents à afficher (Top-K) pour contrôler la quantité d'information visible. (pour une simulation nous avons fait jusqu'a 50)



8.5. Affichage des résultats classés

8.5.1 Classement des documents

Les résultats sont présentés sous forme de liste ordonnée incluant :

- rang du document,
- identifiant,

- score de pertinence,
- extrait du contenu

Chaque document est présenté dans une carte visuelle avec badge de rang.
pour requete 6 modèle BM25 nous avons obtenue :

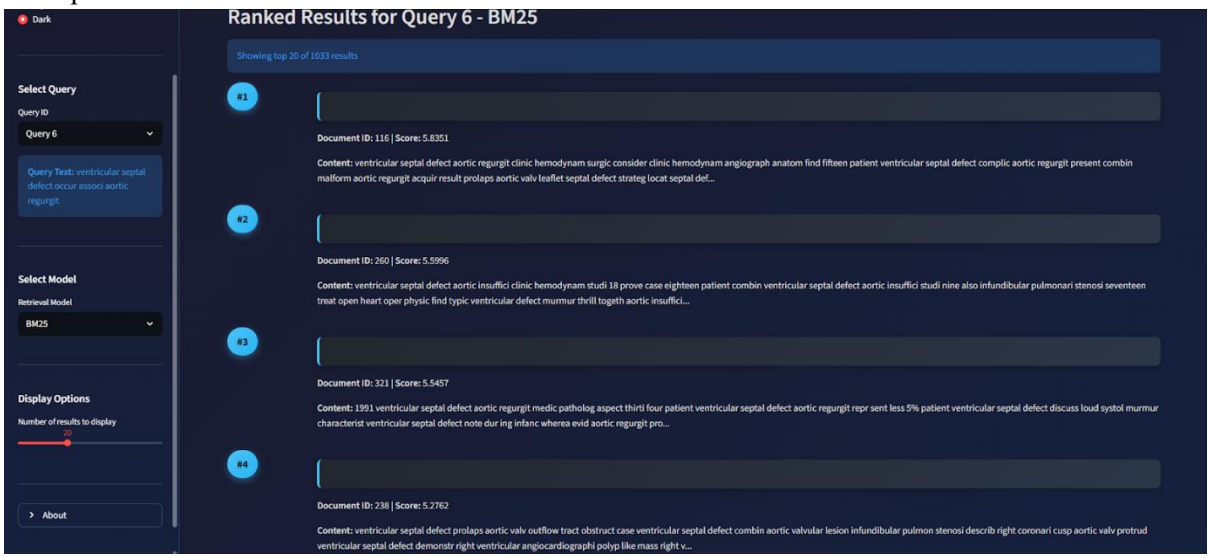


Figure 6 : Résultats classés pour une requête donnée

8.5.2 Objectif fonctionnel

Cette vue permet :

- d'analyser la qualité du classement,
- de comparer visuellement les scores,
- d'interpréter la pertinence des documents retournés.

8.6. Affichage des métriques d'évaluation

8.6.1 Métriques standards

Métriques affichées :

- Precision
- Recall
- F1-score
- R-Precision

- P@5, P@10

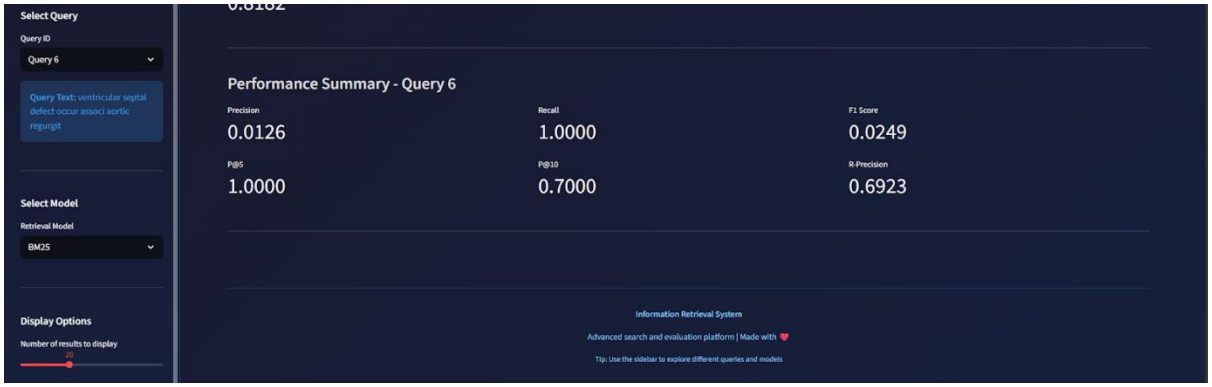


Figure 7 : Métriques standards pour la même requête (requête 5 modèle BM25)

8.6.2 Métriques avancées

Métriques avancées :

- MAP (Mean Average Precision)
- MRR (Mean Reciprocal Rank)
- Precision@K
- Évaluations globales et par requête

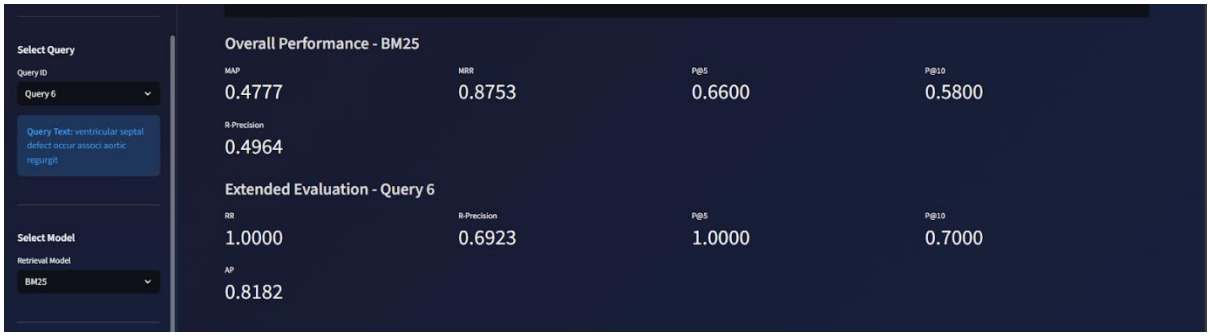


Figure 8 : Métriques avancées globales et par requête

8.6.3 Distribution des scores

Graphique en barres montrant la distribution des scores pour les 20 premiers documents.

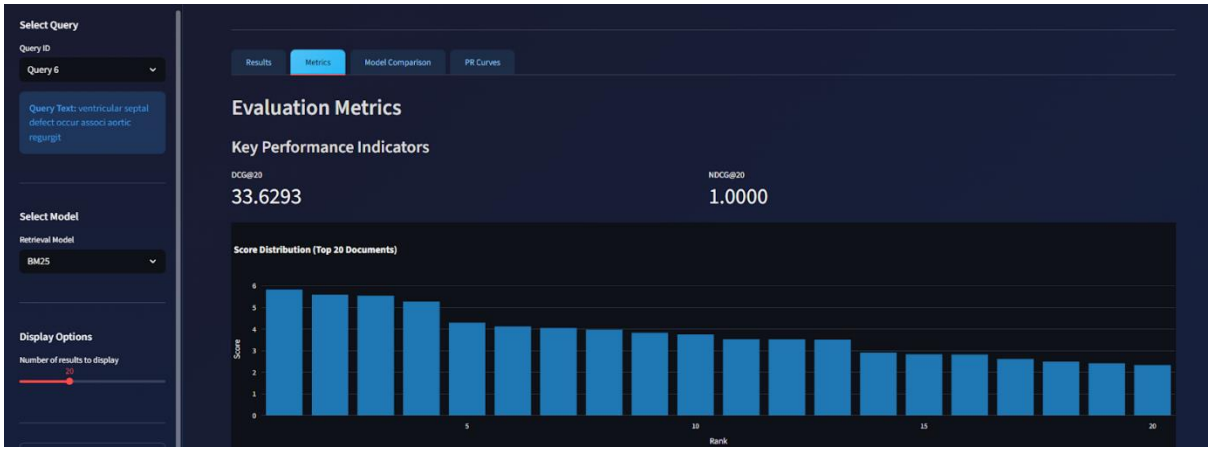


Figure 9 : Distribution des scores des documents

8.7. Comparaison des modèles

8.7.1 Tableau comparatif

Tableau regroupant toutes les métriques pour chaque modèle :

- DCG@20
- NDCG@20
- MAP
- MRR
- Precision / Recall / F1

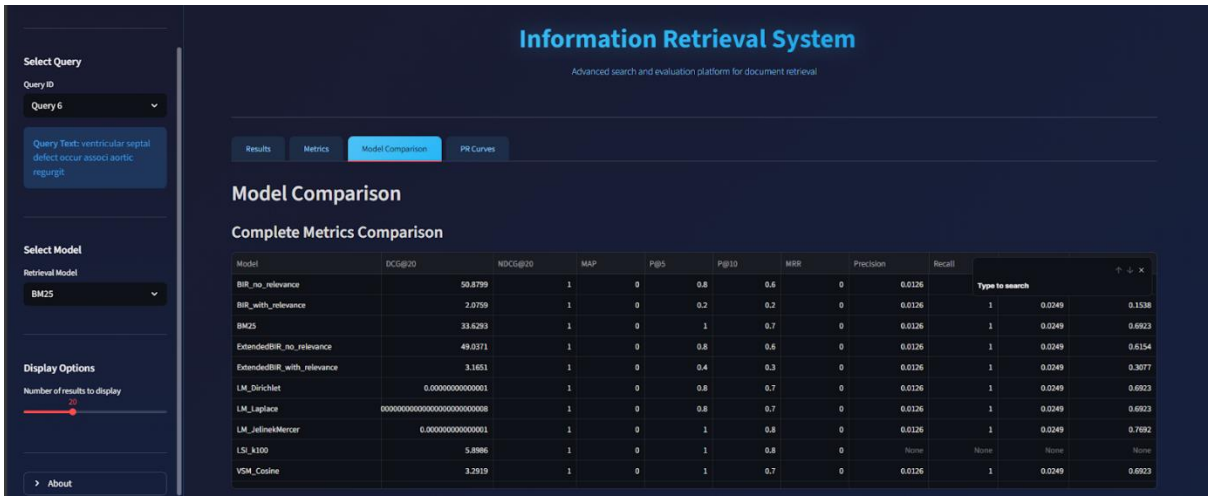


Figure 10 : Tableau comparatif des modèles

8.7.2 Visualisation graphique

Graphiques représentant :

- DCG / NDCG
- MAP / MRR / P@10
- Precision / Recall / F1

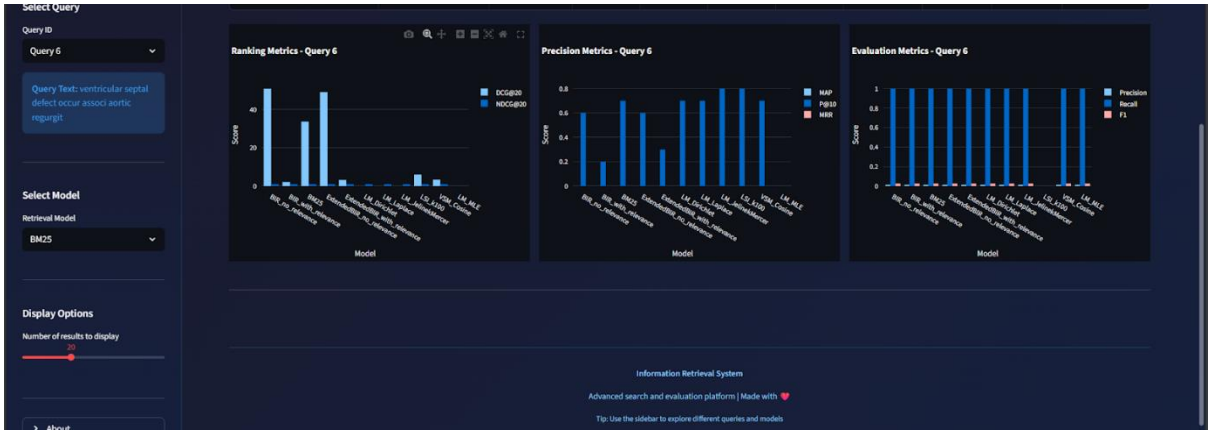


Figure 11 : Comparaison graphique des modèles

8.8. Courbes Precision–Recall

Deux types de courbes PR sont affichées :

- courbe PR standard
- courbe PR interpolée

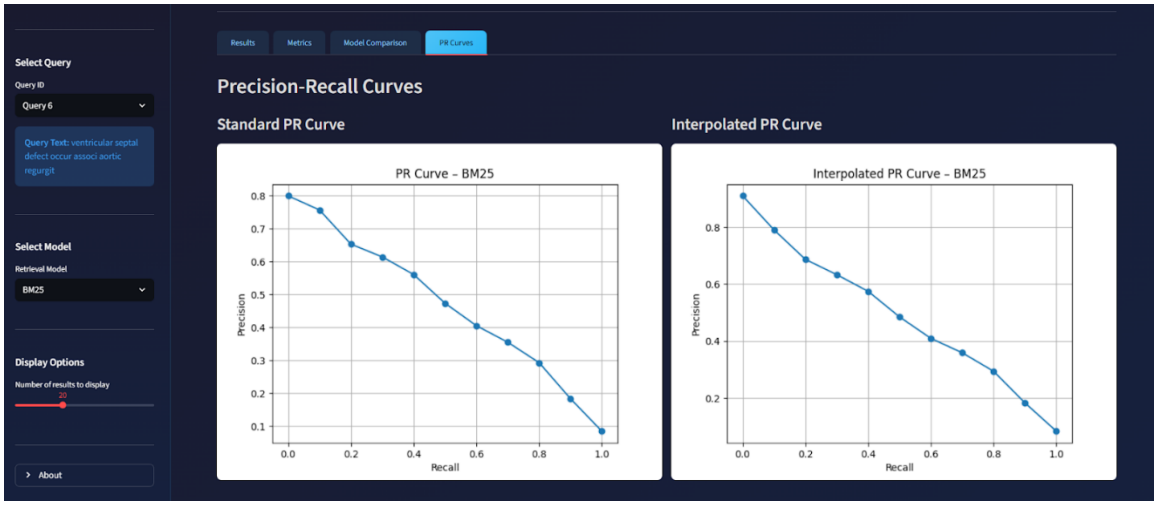


Figure 12 : Courbe Precision–Recall standard & Recall interpolée

8.9. Déploiement de l'application

L'application a été déployée en utilisant une architecture basée sur la conteneurisation et le cloud. Développée avec Streamlit, elle est encapsulée dans un conteneur Docker, garantissant portabilité et cohérence de l'environnement d'exécution.

Le déploiement est automatisé via **Google Cloud Build**, qui construit l'image du conteneur et la stocke dans le registre de Google Cloud. Cette image est ensuite déployée sur **Google Cloud Run**, une plateforme serverless qui exécute l'application sans nécessiter la gestion d'infrastructure serveur.

Cloud Run assure une montée en charge automatique, une haute disponibilité et un accès sécurisé via une URL publique. Cette configuration rend l'application accessible en ligne, facilement partageable et adaptée à un usage académique ou professionnel.

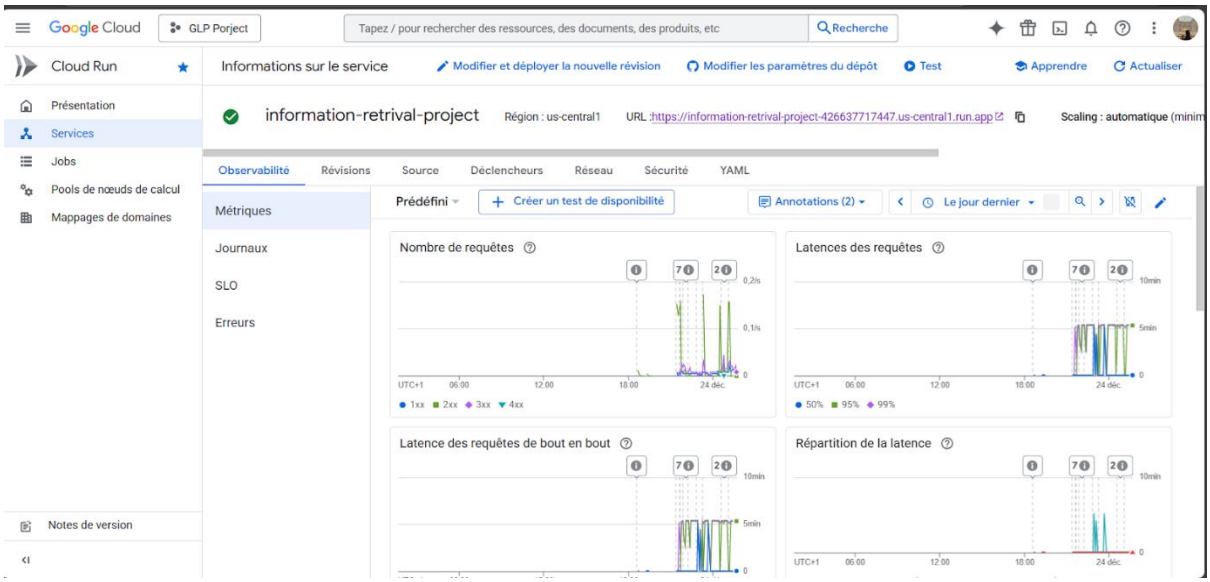


Figure 13 : Google Cloud Run interface

➔ **Lien de l'interface :** https://information-retrival-project-426637717447.us-central1.run.app/

8.10. Conclusion

L'interface Streamlit permet :

- exploration interactive des résultats de recherche,
- évaluation complète des performances des modèles IR,
- comparaison visuelle claire et interprétable,
- expérience utilisateur moderne grâce à la personnalisation du thème.

L'architecture modulaire et l'utilisation de fichiers JSON assurent une extensibilité facile du système

9. Conclusion

9.1. Synthèse des résultats

Ce projet a permis d'évaluer et de comparer onze modèles de recherche d'information sur le corpus MEDLINE, en suivant un pipeline rigoureux de prétraitement, d'indexation et d'évaluation. Les résultats expérimentaux montrent que les modèles sémantiques et vectoriels — en particulier LSI (k=100) et VSM Cosine — obtiennent les meilleures performances en termes de MAP, P@K et qualité du classement, confirmant leur capacité à capturer des relations pertinentes entre termes et documents. Les modèles probabilistes tels que BM25 et Extended BIR offrent également des résultats solides, notamment lorsqu'ils intègrent des informations de pertinence. En revanche, les modèles de langage simples (MLE) et les approches binaires sans feedback présentent des limitations significatives dans ce contexte. Par ailleurs, l'implémentation d'un système Learning to Rank (LTR) a démontré la possibilité d'améliorer le classement en combinant les scores de plusieurs modèles, avec un rappel élevé sur les documents pertinents. Enfin, l'interface Streamlit développée permet une visualisation interactive et complète des résultats, facilitant l'analyse et la comparaison des modèles.

9.2. Limites et perspectives

Plusieurs limites ont été identifiées au cours de ce travail. Tout d'abord, l'évaluation repose sur un corpus de taille modeste (1 033 documents et 30 requêtes), ce qui peut limiter la généralisation des résultats. De plus, certaines métriques classiques (précision, rappel, F1) se sont avérées peu discriminantes dans notre configuration expérimentale, car tous les modèles retournent l'intégralité de la collection. Par ailleurs, le système LTR, bien que prometteur, souffre du déséquilibre important entre documents pertinents et non pertinents, affectant sa précision. Enfin, le prétraitement et l'indexation réalisés ne prennent pas en compte les aspects sémantiques profonds ou le contexte étendu des documents.

Pour poursuivre ce travail, plusieurs pistes d'amélioration peuvent être envisagées : l'expérimentation sur des corpus plus larges et variés, l'intégration de modèles neuronaux (BERT, SBERT) pour une représentation contextuelle, l'optimisation du seuillage et du top-K pour rendre les métriques classiques plus informatives, et l'enrichissement du système LTR avec des features supplémentaires et des techniques avancées de rééquilibrage des classes. Enfin, l'interface utilisateur pourrait être étendue avec des fonctionnalités de recherche en temps réel et des visualisations encore plus interactives.