

1) Informations utiles :

- **Parser :** Construit un dictionnaire de document, cette classe est générale pour tout type de fichier texte.
- **IndexerSimple :** La méthode 'indexation' permet de construire « l'index », « l'index inverse » et « les liens hypertextes from et to ».
- **GridSearch :** Cette class permet d'initialiser une grille d'optimisation de paramètres.
- **Cross validation :** Cette class permet d'initialiser une validation croisée (KFold).
- **IRModel :** Contient les modèles vectoriel, de langue, et Okapi_bm25.
- **PageRank :** Dans la class PageRank.
- **EvalIRModel :** Pour évaluer une requête/requêtes avec une mesure/mesures.
- **Hits :** Cette class implémente l'algorithme Hits.
- **Search :** Pour le serveur, et pour changer le fichier (cacm ou cisi).

2) Le serveur :

Il faut installer les requierement.txt (pip install -r requierement.txt)

Il est hébergé sur : <https://billion-search.github.io/search>

Il faut lancer le serveur avec : « python app.py », avec le fichier app.py fournis

3) Répartition du travail :

Nous avons tout d'abord commencé à avancer ensemble en TME, ensuite chacun de son côté, on se réunissait pour comparer les avancées, ainsi que le travail effectué de chacun.

Nous nous envoyant les idées que nous avons trouvées intéressantes à rajouter.

Et pour la version finale on l'a faite ensemble (ce n'est pas partagé).

4) Bonus effectué :

Nous avons fait la majorité des bonus, seul la partie sur l'optimisation du parsing, nous l'avons pas abordé.

Nous avons aussi rajouté l'algorithme HITS

Nous avons rajouté une class Cross Validation ainsi que GridSearch

5) Difficultés rencontrées :

Nous avons plutôt géré ce qu'il fallait faire, et nous avons compris ce qu'il nous est demandé de faire.