Université des Sciences et de la Technologie Houari Boumediene

Rapport technique de Projet n°1: Recherche D'information.

Réalise par :

LARABA Oussama

DJEZZAR Mohamed el Mehdi

MEGHNI Mohammed el Amine

Année Scolaire:

2020/2021

Spécialite :SII,M2,G02

1. Indexation

1.1 Indexation par document

L'indexation par document, permet de récupérer en temps constant, tous les mots d'un document précis, avec la fréquence de chaque mot dans le document.

Type de structure : dictionnaire (clé : index du document, valeur : dictionnaire (clé : mot, valeur : fréquence du mot dans le document))

Example : "1": { "preliminary": 1, "report": 1, "international": 1, "algebraic": 1, "language": 1, "perlis": 1, "samelson": 1},

Type de fichier : json	**Nombre d'entrées :** 3204 documents
Taille en octets : 147552	**Temps de génération** : 1.665 Seconde

1.1 Indexation par mot

L'indexation par mot, permet de récupérer en temps constant, tous les documents où un mot précis appartient, avec la fréquence du mot dans chacun de ces documents.

Cette phase n'est pas demandé dans le projet mais nous avons l'utiliser pour faciliter les autres parties de projet

Type de structure : dictionnaire (clé : mot, valeur : dictionnaire (clé : index du document, valeur : fréquence de mot dans le document))

Example : "international": { "1": 1, "99": 1, "414": 1, "690": 1, "975": 1, "1362": 1, "1476": 2, "2875": 1, "3184": 1 },

Type de fichier : json	**Nombre d'entrées :** 11391
Taille en octet : 5242968	**Temps de génération** : 0.124 Seconde

1.3 Fichier inverse par fréquence

Type de structure : dictionnaire (clé : tuple (mot, index du document), value : fréquence du mot dans le document)

Example : "('preliminary',1)": 1, "(, 'report',1)": 1, "('international',1)": 1.

Type de fichier : json	**Nombre d'entrées :** 90466
Taille en octets : 5242968	**Temps de génération** : 0.117 Seconde

1.4 Fichier inverse par poids

Type de structure : dictionnaire (clé : tuple (mot, index du document), valeur : poids du mot dans le fichier)

Example : "('preliminary', 1)": 2.2073650374690716, "('report', 1)": 1.7216244909023954

Type de fichier : json	**Nombre d'entrées :** 90466
Taille en octets : 5242968	**Temps de génération** : 2.425 Seconde

2. Modèle booléen

2.1 Evaluation des expressions booléennes

- Tokéniser la requête (transformer la requête en une liste de mots)
- Remplacer le mot de la requête par 1 si ce mot appartient au document au cours de traitement, 0 sinon.
- Les mots tels que **les parenthèses, and, or**, ne seront pas traités (ne seront pas remplacés par aucune valeurs).
- Passer la liste à la fonction ***eval*** du langage python afin de retourner une valeur booléenne.
- Si la valeur booléenne est vraie, alors le document sera choisi comme pertinent.

Example: report and light or (implementation and not algebraic) **Temp de response** : 0.596

3. Evaluation du modèle vectoriel

Nous avons retourné que les 5 premier document retourné.

Example : Computer Science

Résultat :

Inner product : Documents [3130, 3122, 3019, 2930,2899]	**Temps de réponse** : 0.0169
Cousine : Documents [1205, 1654, 1771, 2899, 3130]	**Temps de réponse** : 0.0239
Dice coefficient : Documents [1205, 1654, 1771, 57, 3010]	**Temps de réponse** : 0.0239
Jacard coefficient : Documents [1205, 1654, 1771, 57, 3010]	**Temps de réponse** : 0.0259

Nous avons remarque que l'ordonnancement des documents est différent ça affecte sur la précession de système et que Dice coefficient et Jacard coefficient ils ont le même ordre.

Evaluation : (Cousine comme fonction de mesure)

Pour chaque requête, nous devons déterminer le seuil de similarité et le nombre de document optimaux pour une bonne précision et rappel. Pour cela, nous allons boucler sur le seuil et le nombre de document pour trouver la combinaison idéale. La valeur de de paramètre « seuil » varie de 0 jusqu'à la similarité du document qui a la plus grande similarité avec un pas du 0.05. Le paramètre « nombre de documents à retourner » varie entre le nombre de document pertinents jusqu'au 100 avec un pas du 1 car dans notre ensemble des requêtes on a 51 comme max de document pertinent de la requête 25 de reponse des requêtes donc a partir de cette valeur la precesion sera affaiblé d'une manière important et donc on a choisi 100 comme max size.

Voilà les resultat :

Requête	seuil	taille	Rappel	Précession	f-mesure	Temps
1	0.05	9	40.0%	22.2%	28.5%	43.5 s
2	0.05	10	66.6%	20.0%	30.7%	8.5 s
3	0.05	71	33.3%	2.8%	5.1%	21.4 s
4	0.05	13	33.3%	30.7%	32%	80.1
5	0.05	21	50.0%	19.0%	27.5%	29.2 s
.	.	.	.	.	.	.

40	0.05	14	70.0%	50.0%	58.3%	112.7 s
.	.	.	.	.	.	.
64	0.05	15	100.0%	6.6%	12.5%	20.6 s

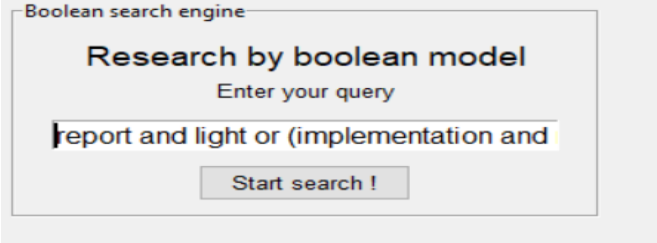
Nous avons remarque que tous les requêtes ont un seuil de 0.05 et pour les 5 premier requête on a moyenne f_mesure ≈ 24.2%.

4. Interface Graphique

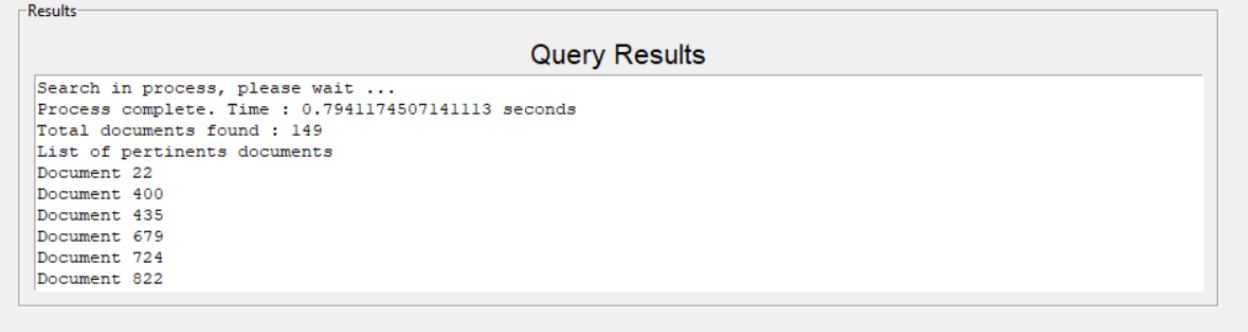
4.1 Modèle Boolean :

Il suffit d'entrer la requête boolean et appuyez sur le button start search.

Example : report and light or (implementation and not algebraic)

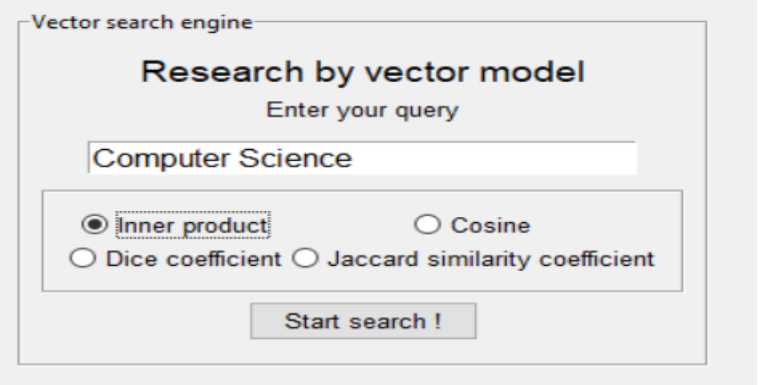


Le Résultat :



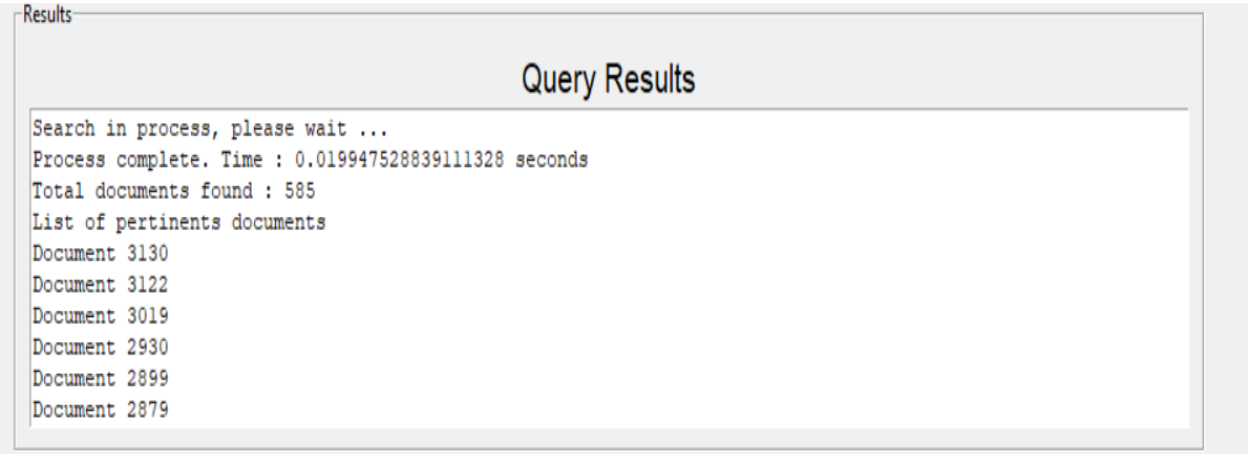
4.2 Modèle vectoriel :

Il y a un champ pour la requête et il y a aussi les radiobutton pour choisir la mesure d'evaluation.



Résultat :

Vous pouvez de voir tous la résultat il suffit de clique sur le texte area et appuyez sur le button ↓ du clavier

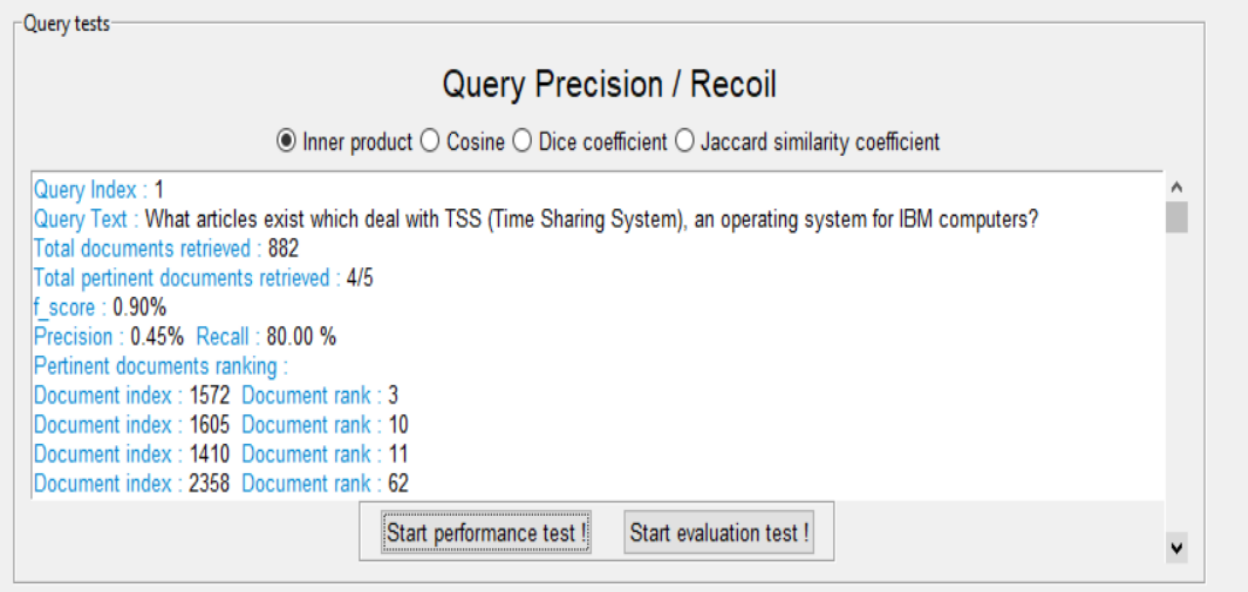


4.3 Evaluation et Performance des résultat

4.3.1 Performance :

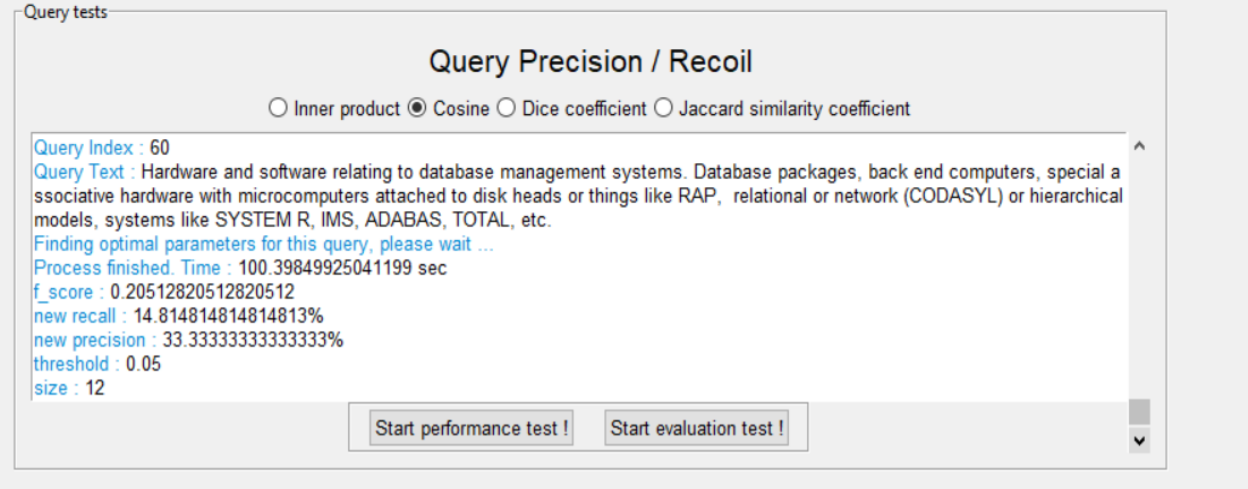
Dans cette phase on a fait l'evaluation de performance avec des paramatre par defaut (seuil = 0 taille = nombre des documents returne).

Et donc pour lancé il faut choisir une fonction de mesure et cliquer sur start performance test.



4.3.2 Evaluation :

Cette phase prend un temp pour trouvé les paramaters idéal pour chaque requêtes.



5. Comparaison entre le modèle booléen et vectoriel

5.1 Modèle booléen

Le modèle booléen, dont son grand avantage, est la facilité de l'implémentation, cependant, durant ce projet, nous pouvons remarquer que :

- La correspondance exacte entre les mots de la requête ainsi que les documents, peut soit ne rien sélectionner, ou de sélectionner un grand nombre de documents (possibilité de précision nulle).
- La syntaxe de la requête est presque non compréhensible p≈our presque tous les utilisateurs.
- Tous les mots de la requête ont le même poids.
- Les documents sont récupérés sont degrés de pertinence.
- Les termes sont considérés comment indépendants.

5.2 Modèle vectoriel

Le modèle vectoriel est basé sur des notions mathématiques exactes, c'est le modèle de base des modèles avancés, cependant, nous pouvons remarquer que :

- Plus le corpus est grand, plus l'espace vectoriel devient plus important, plus le temps de calcul devient couteux.
- Les termes sont considérés comment indépendants.