Recherche d'information

« Rapport technique du projet : Indexation et évaluation des modèles de base de recherche d'information »

CHIKH Khadidja

Master 2 SII

Groupe 1

05/01/2019

Phase d'indexation:

Au fur et à mesure et selon le besoin de chaque étape, nous avons construit et sauvegardé les indexes suivants:

Index	Structure	Format	Temps d'indexation	Taille	Exemple
documents	dictionnaire de dictionnaires	Binaire (.pickle)	1.6948566436767578 secondes	1011 ko	{1:{"nice":2,..},...}
Fichier inversé	dictionnaire de dictionnaires	Binaire (.pickle)	0.0591588020324707 secondes	547 ko	{"nice":{1: 3,...},...}
-Indexe pondéré **-Fichier inversé pondéré**	-dictionnaire de dictionnaires -dictionnaire de dictionnaires	Binaire (.pickle)	1.1593577861785889 secondes (les deux ensembles)	Docs: 1564 ko Inversé: 1100 ko	//
requetes	dictionnaire	Binaire (.pickle)	0.016042232513427734 seconds	9ko	{1:"...",2:"..."}
requête-docs pertinents(donnés)	Dictionnaire de dictionnaires	Binaire (.pickle)	0.014073610305786133 secondes	5ko	{1:{2:1,3:1..}}
-Requête-rés-rsv -Requête-rapp-prec-fmesures(produit interne)	-dictionnaire de dictionnaire	Binaire (.pickle)	0.11726999282836914 secondes	491ko 2ko	{1:{2:1.3,..}..} {1:(0.25,0.4,0.3)..}
-Requête-rés-rsv -Requête-rapp-prec-fmesures (coef de dice)	-dictionnaire de dictionnaire -dictionnaire	Binaire (.pickle)	0.16142749786376953 secondes	491 ko 2ko	//
-Requête-rés-rsv -Requête-rapp-prec-fmesures (cosinus)	-dictionnaire de dictionnaire -dictionnaire	Binaire (.pickle)	0.14939522743225098 seconds	491 ko 2ko	//

-Requête-rés-rsv -Requête-rapp-prec-fmesures(jaccard)	-dictionnaire de dictionnaire -dictionnaire	Binaire (.pickle)	0.16042828559875488 secondes	491 ko 2ko	//
requetes-métriques -rapp-prec-fmesures	dictionnaire de dictionnaires	Binaire (.pickle)	72.3846185207367 secondes	12954ko	{1:{(1,0.2):0.5,..}..}
-req-bestsmetriques-rapp-prec-fmes -meilleures-des-meilleuresmetriques	-dictionnaire de dictionnaires -liste dynamique	Binaire (.pickle)	1.031782627105713 secondes	1546ko 1ko	- // -[(2,0.4),..]
-requêtes-taille-rappel-précision -taille-rappelmoyen-précisionmoyenne	-dictionnaire de dictionnaires -dictionnaire	Binaire (.pickle)	0.609616756439209 seconds	1133ko 55ko	-{1:{5:(3,0.6)..}...} - //

Fonctions d'accès :

Accès par numéro de document	Accès par mot
0.032053470611572266 secondes	0.02005171775817871 secondes

Points positifs:

-La construction des indexes en général est d'une vitesse acceptable.

-La construction des fichiers inversé occupe moins en temps et en taille que celle des indexes de documents(ie taille de la structure est minimisée, ceci est dû au fait que le nombre de mots>nombre de documents et les répétitions dans le cas d'index de documents)

-L'accès par numéro de document est rapide et encore moins que celui par mot.

Points négatifs:

-Le parcours de tout un index est plus lent (en secondes)

-La construction de certains indexes nécessitent le parcours complet de tout un autre index.

-La construction de l'index portant pour chaque requête, chaque combinaison de métriques le rappel, la précision et la f-mesure, a pris la durée la plus longue (>72s) et la taille la plus grande (12954ko), ceci est dû principalement au nombre important des valeurs possibles de "taille" pour chaque requête.

Le modèle booléen :

Forme de base	Parenthèses	Temps
La requête est une chaîne de caractères composée de termes reliés par un ou plusieurs opérateurs booléens: and, or ,not	Permises (avec ou sans)	0.11931896209716797 seconds variant selon la longueur de la requête

L'évaluation de la requête consiste à la parcourir en vérifiant d'abord si elle est bien écrite, ensuite l'existence de chaque terme dans l'index des documents, s'il existe alors il sera remplacé par 1 sinon par 0. Nous aurons à la fin une expression comme par exemple: 0 or 1 and (1 and 0) ,que nous injecterons à la fonction prédéfinie **eval** qui retourne le résultat final : 0 (aucun document pertinent) ou 1 (un ensemble de documents pertinents)

Le modèle vectoriel :

Comparaison entre les résultats des 4 formules:

Lors de la recherche des documents pertinents (tous les documents donnant des similarités>0, à l'aide des indexes résultats" et l'index des requêtes) pour chaque requête, avec chacune des 4 formules, nous avons remarqué une sorte de similarité dans l'ordre des documents retournés (la pertinence) entre les 3 formules (produit interne, coefficient de dice, jaccard) et une différence entre ces 3 et la formule de cosinus.

Les exemples suivants illustrent ce fait: Pour les 5 premiers documents retournés

Produit interne: **N°doc similarité** 2318 3.534009962711176 2542 3.1202035160404114 2268 2.9947751267805214 1653 2.6110198516289844 2634 2.535006078472707	**Produit interne:** **N°doc similarité** 2318 0.8005057686244686 2634 0.7535500871349442 2268 0.7301154894644689 1938 0.6862256729351953 3136 0.6601002101262123
Produit interne: **N°doc similarité** 2634 0.8623844709332684 2951 0.8562307758078082 2812 0.8427672973299751 2318 0.8076347029896331 1938 0.7744822590920627	**Produit interne:** **N°doc similarité** 2318 0.6673694192813925 2634 0.6045570538834203 2268 0.5749463698526153 1938 0.5223314680446985 3136 0.4926489392079019

Nous observons une stagnation pour les 3 formules, ceci est dû à manque d'exactitude dans les valeurs de similarité calculées par rapport à celle du cosinus.

Nous avons donc retenu la formule de cosinus pour les expérimentations suivantes.

Recherche des seuil et taille idéaux:

Nous avons procédé à une recherche par paramètrage comme pour les métaheuristiques. La taille est exprimée en nombre de documents et le seuil en pourcentage (0.1=10%). Nous avons calculé pour chaque requête, pour chaque combinaison: le rappel et la précision (à l'aide des indexes : résultats, docs pertinents et requêtes)

Pour procéder à la sélection, et comme nous ne favorisons aucune des deux, nous avons calculé la moyenne de chaque couple (rappel,précision) pour trouver un équilibre. Cette moyenne est la **F-mesure.** Nous avons sélectionné par la suite, les meilleures combinaisons pour chaque requête, selon la meilleure valeur de la F-mesure.

Exemple: Pour la requête n°1 ces combinaisons ont donné un meilleur résultat:

Taille,seuil	R	P	F-mesure
(8, 0.0)	0.6	0.375	0.4615384615384615
(8, 0.1)	0.6	0.375	0.4615384615384615
(8, 0.2)	0.6	0.375	0.4615384615384615
(8, 0.3)	0.6	0.375	0.4615384615384615
(8, 0.4)	0.6	0.375	0.4615384615384615
(8, 0.5)	0.6	0.375	0.4615384615384615
(8, 0.6)	0.6	0.375	0.4615384615384615

Les résultats montrent 7 combinaisons possibles pour obtenir la meilleure F-mesure, nous pouvons donc prendre **(8,0.6).**

Pour aller plus loin et par souci de curiosité, nous avons généralisé en cherchant la meilleure combinaison pour toutes les requêtes. Nous l'avons calculée en prenant la "meilleure" combinaison la plus répétée.

Nous avons obtenu: (taille,seuil)=(5, 0.4)

Remarques:

Contrairement au modèle booléen qui peut est restrictif en terme de résultat, le modèle vectoriel retourne toujours un ensemble de documents à une certaine pertinence si au moins un mot de la requête est présent dans l'index. Le modèle vectoriel est aussi plus rapide et consomme approximativement le même temps de réponse pour n'importe quelle requête ceci est dû au fait d'utiliser le fichier inversé

(pondéré) qui permet moins d'accès qu'à l'utilisation de l'index des documents contrairement au modèle booléen.

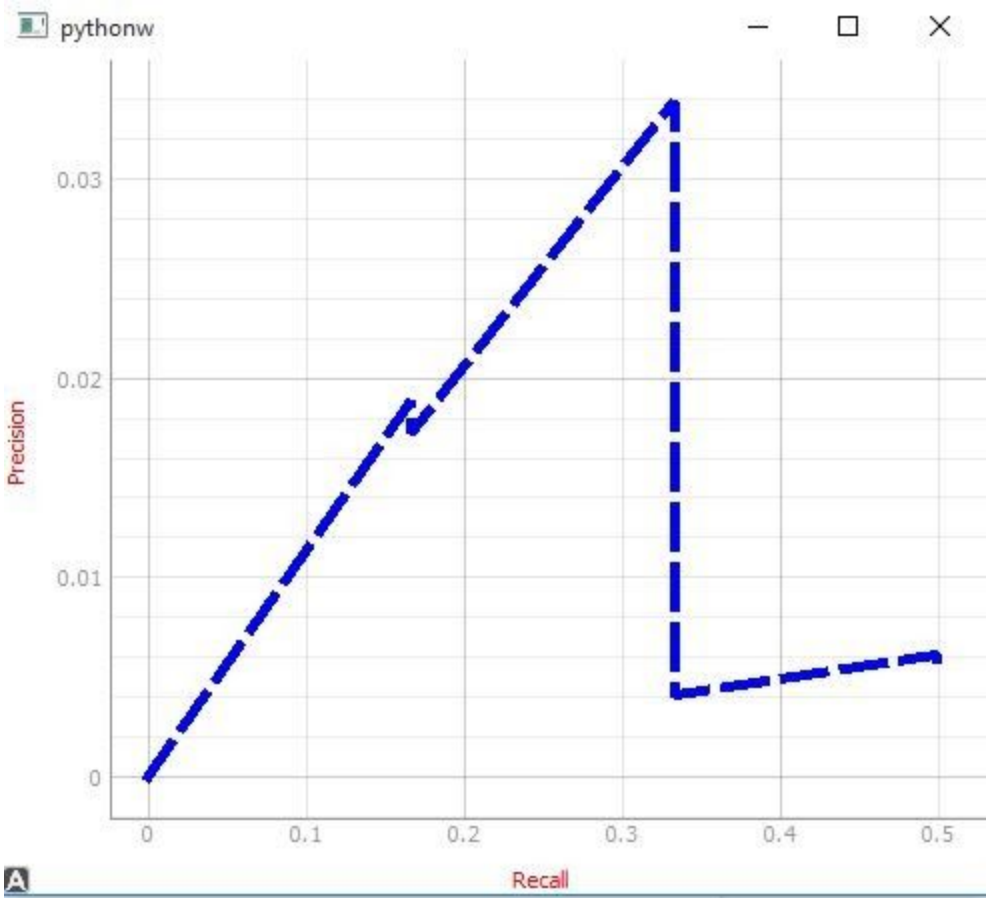
-L'enregistrement des indexes de façon permanente ainsi que l'utilisation de la structure "dictionnaire' (au lieu des listes par exemple) a permis d'expérimenter les différentes phases de manière rapide, structuré et souple.
-Meilleur rappel obtenu après fixation des métriques pour les requêtes: 66%
-Meilleure précision obtenue après fixation des métriques pour les requêtes: 54%

La courbe de rappel-précision:

Pour chaque requête, pour chaque taille (dépendante de sa longueur), nous avons calculé le rappel et la précision. Nous avons par la suite utilisé l'index de ces derniers pour calculer, pour chaque taille, le rappel et la précision moyens afin de les fixer (ainsi que lisser la courbe)

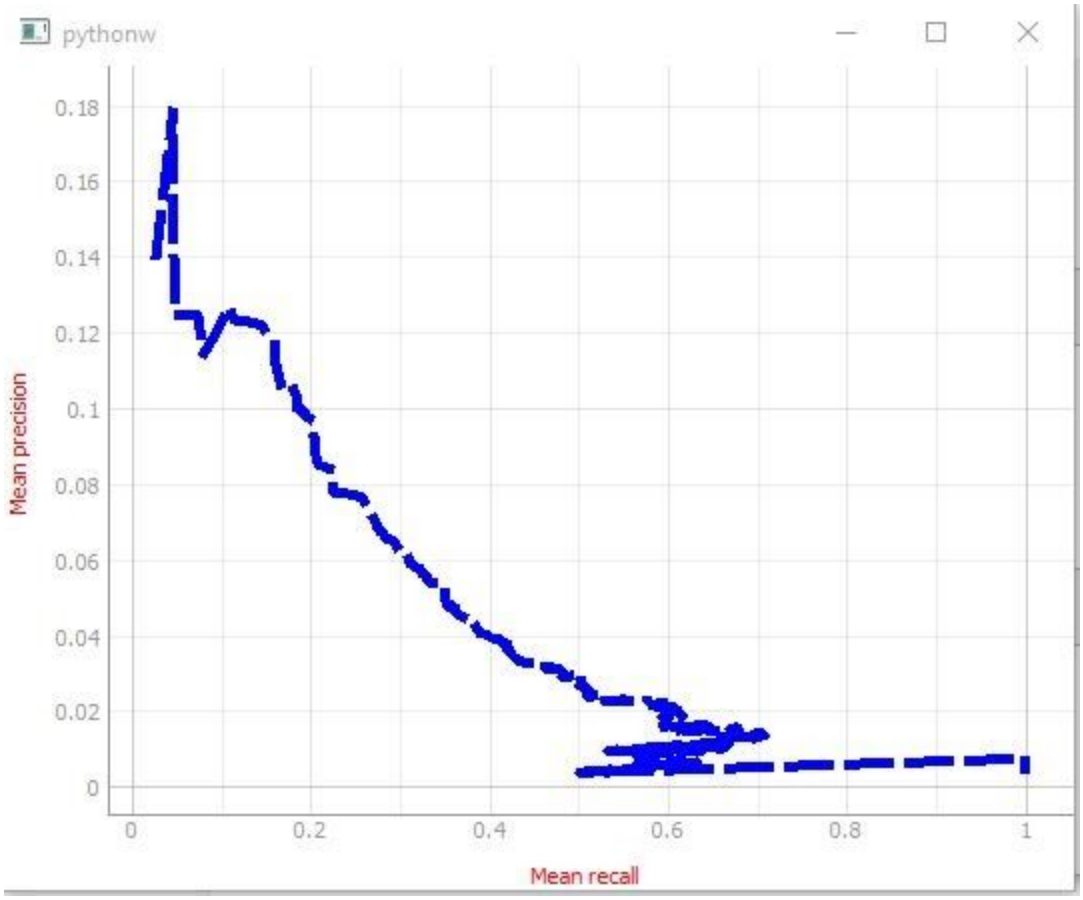
Les exemples suivants illustrent les résultats pour la requête N°3:

Courbe rappel-précision



La courbe monte jusqu'à la plus haute pique aux valeurs des deux métriques ~ (0.33,0.034) (le point d'équilibre) à la taille= 59 ensuite redescend jusqu'à ~(0.33,0.04) et commence à remonter doucement.

Courbe rappelmoyen-précisionmoyenne



La courbe monte jusqu'à la plus haute pique aux valeurs des deux métriques ~ (0.046,0.18) à la taille=2 ensuite redescend jusqu'à (0.5<0.02) et devient presque stable.

Résultats:
Pour taille=1:
Rappel moyen (minimum)=2.7159049599272792 % précision moyenne (maximale)=14.0625 %
Pour taille=2417:
Rappel moyen(maximum)=100 % précision moyenne (minimum)= 0.4964832436905255 %