TRAITEMENT AUTOMATIQUE DU LANGAGE NATUREL
MASTER 2
-
IAA

CONCEPTION ET IMPLÉMENTATION D'UN ALGORITHME DE RECHERCHE D'INFORMATIONS
-
INTELLIGENCE ARTIFICIELLE

RAPPORT
JANVIER 2021

L'Unité d'Enseignement : IA et Langage
Enseignants : Madame CABRIO Elena et VILLATA Serena
Étudiant : Thomas GAUCI
Date de rédaction du dossier : JANVIER 2021

CONCEPTION :

Je ne m'attarde pas sur le partie de segmentation du document qui est fait par NLTK, utilisé déjà dans mes anciens TP.

Pour la partie de l'élimination des mots vides j'ai utilisé stopword de NLTK qui contient une liste plutôt bonne des mots non importants.

Pour ce qui est de la lemmatisation j'ai pris le choix de toujours utiliser NLTK qui propose une lemmatisation en fonction de la classe grammatical du mot (ce qui n'est pas le cas de tous les lemmatiser)

J'ai donc d'abord construit un dictionnaire reprenant tous les mots pour chaque document avec la fréquence d'apparition du mot dans le document en question.

Exemple :

mot	document	fréquence
"butter"	doc1	72
"desease"	doc1	24
"match"	doc2	1

Ensuite grâce à ce tableau j'ai pu créer la matrice d'incidence en notant pour un mot donné dans quel document il est présent.
Suite à cela j'ai créé l'index inversé grâce à la matrice.

Pour ce qui est des requêtes j'ai d'abord fait l'algo pour les requêtes complexes qui peuvent être analysées comme un enchaînement de mot avec des AND entre chaque mot.
antibody treatments -> antibody AND treatments
Puis j'ai fait les requêtes booléen, j'ai voulu utiliser mon algorithme des requêtes complexes pour cet algorithme.
En effet nous pouvons tout transformer en AND car finalement un NOT est juste un !AND. Je vais chercher les textes avec le meilleur score ou est le mot suivant NOT est soustraire ce score a mon score total pour chaque document
J'ai donc trié ma requête en trois tableaux distinct AND, NOT et OR

Exemple : NOT plasma AND infection AND restrictions OR pneumonia

Devient trois requête complexe :

- plasma AND infection AND restrictions
- pneumonia
- plasma (ATTENTION pour OR chaque mot = une requête)

Je récupère le résultat de ces trois requêtes en faisant pour chaque score (TF*IDF) de document :
score total = score_requeteAND - score_requeteNOT + score_requeteOR

Ce qui me donne un score total pour mon document.
Je soustrait à ma requête AND le score obtenu par ma requête NOT (car cela donne les documents ou le mot qu'on ne veut pas apparaît le plus) auquel j'ajoute aussi le score obtenu de la requête OR (comme si le OR était un bonus, si on a tout les mots avec AND c'est bien mais c'est encore mieux si on a ce mot x ou y).