Document et Web Sémantique - TP Moteur de recherche

L'objectif de ce TP est de compléter le code python d'un mini moteur de recherche Web.

Complement de cours

Nous avons vu en cours les modèles booléens et vectoriels pour la recherche d'information.

Nous avons vu que les avantages du modèle booléen sont sa simplicité et sa rapidité. Son inconvénient est qu'il ne permet pas de classer les documents pertinents.

Nous avons vu que les avantages du modèle vectoriel sont qu'il permet de classer les documents pertinents et de faire de la recherche par similarité. Son inconvénient est qu'il ne permet pas d'utiliser les opérateurs booléens dans les requêtes.

Pour palier à ce problème, Salton, Fox et Wu ont proposé un modèle booléen étendu qui permet d'identifier les documents pertinents au regard d'une requète booléenne utilisant les opérateurs Et, Ou et Non, mais aussi de les classer [1].

Ce modèle repose d'une part sur une représentation vectorielle normalisée des documents du corpus avec par exemple le tf.idf. Ensuite ce modèle repose sur le calcul d'une similarité entre une requête q et chaque document d_i du corpus tel que d_i est représenté par un vecteur de valeurs w_{ij} ($j \in 1..nb\_mots$).

Cette similarité est calculée de la manière suivante :

— si la requère est composée dans seul terme T se trouvant à la position t dans la matrice $tf.idf$, alors :

$$sim(q, d_i) = w_{it}$$

— si la requête est une conjonction (exp_g ET exp_d), alors :

$$sim(q, d_i) = 1 - \sqrt{\frac{(1 - sim(exp_g, d_i))^2 + (1 - sim(exp_d, d_i))^2}{2}}$$

— si la requête est une disjonction (exp_g OU exp_d), alors :

$$sim(q, d_i) = \sqrt{\frac{(sim(exp_g, d_i))^2 + (sim(exp_d, d_i))^2}{2}}$$

— si la requête est une négation (NON exp), alors :

$$sim(q, d_i) = 1 - sim(exp, d_i)$$

Dans le cadre de ce TP, lors de la recherche, nous utiliserons tout d'abord le modèle booléen pour identifier les documents pertinents. Puis nous utiliserons le modèle booléen étendu pour classer ces documents pertinents.

1. https://dl.acm.org/doi/pdf/10.1145/182.358466

Decsription de l'application

Le moteur de recherche, sujet de ce TP, est dans le répertoire `moteur_de_recherche_web`. Ce répertoire est un environnement poetry [2] pour installer les dépendances du projet. Pour cela, exécutez les commandes suivantes :

```
cd moteur_de_recherche_web
poetry install
poetry shell
```

Vous trouverez les modules et packages suivants :

— le script `main.py` le programme principal qui permet d'indexer le corpus et de rechercher des documents

— le package `moteur_de_recherche` qui contient les modules suivants :

  — `robot_web.py` qui propose les fonctions pour indexer des documents à partir d'un URL ;

  — `requeteur.py` qui propose les fonctions pour rechercher un document dans le corpus ;

  — `analyseur.py` qui propose l'analyseur syntaxique des requètes booléennes ;

  — `representation_ensembliste.py` qui propose les fonctions pour représenter les documents du corpus à l'aide d'un index inversé ;

  — `representation_vectorielle_tfidf.py` qui propose les fonctions pour représenter les documents du corpus à l'aide du tf.idf normalisé ;

  — `visiteur_modele_booleen.py` qui propose les fonctions pour visiter (*design pattern* visiteur) les noeuds de l'arbre syntaxique des requètes booléennes et ainsi réaliser les opérations ensemblistes correspondantes ;

  — `visiteur_modele_booleen_etendu.py` qui propose les fonctions pour visiter les noeuds de l'arbre syntaxique des requètes booléennes pour calculer la similarité entre une requête et un document du corpus.

Il y aussi quelques tests unitaires pytest qui vous permettront de tester le code de vos fonctions.

Travail à réaliser

1. Donnez les codes des fonctions du module `representation_ensembliste.py` ;
2. Donnez les codes des fonctions `_tf`, `_idf` du module `representation_vectorielle_tfidf.py` [3] ;
3. En vous inspirant des fonctions du module `visiteur_modele_booleen`, complétez le code des fonctions du module `visiteur_modele_booleen_etendu`.
4. Lancez le programme prinpal : `main.py` :
   (a) pour indexer le corpus de documents à partir de l'URL de la page d'accueil des pages créées par votre CICD ;
   (b) pour rechercher les documents répondants aux requêtes suivantes :
      — `delestre`
      — `delestre ET programmation`
      — `delestre ET programmation ET NON semestre`

2. `pip install poetry`
3. Attention contrairement au cours les documents représentent les lignes de la matrice, les termes les colonnes