TME 1 - Indexation

Exercice 1 – Exercice de compréhension : indexation d'un petit jeu de données

On considère les documents présentés lors du TD1 :

- Doc 1 : the new home has been saled on top forecasts
- Doc 2 : the home sales rise in july
- Doc 3 : there is an increase in home sales in july
- Doc 4 : july encounter a new home sales rise

Ainsi qu'une liste de mots vides : the, a, an, on, behind, under, there, in, on.

Q 1.1 Écrire le code qui à partir de la chaîne de caractère du document 1 : 1) sépare les mots, les transforme en minuscule, compte le nombre d'occurrences par mot dans le texte, 2) supprime les mots vides et 3) stocke le résultat sous la forme :

```
{'new': 1, 'home': 1, 'ha': 1, 'been': 1, 'sale': 1, 'top': 1, 'forecast': 1}
```

Remarque : pour la normalisation des termes, on s'aidera du fichier `porter.py`. Pour compter le nombre d'occurrences d'un terme, on utilisera la librairie `Counter` de `collection`.

Q 1.2 Réaliser les fichiers index et index inversé pour toute la collection de documents.

```
# fichier index :
{0: {'new': 1, 'home': 1, 'ha': 1, 'been': 1, 'sale': 1, 'top': 1, 'forecast':
    1},
1: {'home': 1, 'sale': 1, 'rise': 1, 'juli': 1},
2: {'is': 1, 'increas': 1, 'home': 1, 'sale': 1, 'juli': 1},
3: {'juli': 1, 'encount': 1, 'new': 1, 'home': 1, 'sale': 1, 'rise': 1}}

#fichier index inverse
{'new': {'0': '1', '3': '1'},
'home': {'0': '1', '1': '1', '2': '1', '3': '1'},
'ha': {'0': '1'},
'been': {'0': '1'},
'sale': {'0': '1', '1': '1', '2': '1', '3': '1'},
'top': {'0': '1'},
'forecast': {'0': '1'},
'rise': {'1': '1', '3': '1'},
'juli': {'1': '1', '2': '1', '3': '1'},
'is': {'2': '1'},
'increas': {'2': '1'},
'encount': {'3': '1'}}
```

Q 1.3 Modifier le code pour effectuer une pondération tf-idf.

Exercice 2 – Rechercher avec des index

Le but de cet exercice est d'implémenter des algorithmes de recherche efficace (TAAT et DAAT). Pour ce faire, vous reprendrez le code de l'exercice précédent qui permet de réaliser des structures d'index. Afin d'avoir des collections de documents et de questions plus importantes, vous utiliserez la librairie IR Datasets dont la documentation est accessible à l'adresse suivante : https://ir-datasets.com/. Cette librairie permet de charger un jeu de données, et de parcourir facilement documents et questions d'une collection de RI.

Vous pourrez utiliser la collection CRANFIELD pour commencer, et ensuite une collection de taille plus importante. Afin de vous faciliter la tâche, vous ne travaillerez qu'avec des structures en mémoire.

Vous mesurerez le temps nécessaire à la recherche pour les algorithmes ci-dessous (en moyennant sur 5 séries de recherches).

Q 2.1 Implémenter l'algorithme TAAT

Q 2.2 Implémenter l'algorithme DAAT

Q 2.3 Implémenter l'algorithme WAND