TP 1-2-3 - Indexation-Modèle-Evaluation

Exercice 1 – 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 Ecrire 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 – Modèles de RI

On considère la collection de documents et la liste des stopwords de l'exercice précédent. L'objectif dans cet exercice est d'estimer le score des documents pour la requête *"home sales top"*.

Q 2.1 On pensera à l'optimisation du calcul du score. Quels index faut-il interroger pour avoir un calcul du score pertinent ?

Q 2.2 Ecrire le code qui permet de calculer le score des documents à partir du modèle booléen.

Q 2.3 Ecrire le code qui permet de calculer le score des documents à partir du modèle vectoriel (produit scalaire) dans le cas d'une pondération tf.

Q 2.4 Ecrire le code qui permet de calculer le score des documents à partir du modèle de langue Jelineck-Mercer.

Exercice 3 – Evaluation en RI

L'objectif dans cet exercice est de mesurer la qualité de l'ordonnancement. Pour cela, on définit les jugements de pertinence suivant :

- Requête 1 "top sales" - Documents pertinents : 1
- Requête 2 "sales increase july" - Documents pertinents : 2 et 3 (avec 2 plus pertinent que 3)
- Requête 3 "new home"

Q 3.1 Calculer les mesures de précision, rappel et F-mesure au rang 2 ($P@2$, $R@2$ et $F@2$) pour chaque requête et ensuite leur moyenne sur l'ensemble des requêtes.

Q 3.2 Calculer la mesure de NDCG pour toutes les requêtes.