université PARIS-SACLAY

M2 AIC

Recherche et Extraction d'Information

TP 5 : Similarité entre documents

L'objectif de ce TP est de parvenir à calculer la similarité entre deux documents, en utilisant les pondérations *tf.idf* calculées au TP 2 et la formule du cosinus :

$$sim(d_j, d_k) = \frac{\overrightarrow{d_j} \cdot \overrightarrow{d_k}}{|\overrightarrow{d_j}|.|\overrightarrow{d_k}|} = \frac{\sum_{i=1}^{n} w_{i,j}.w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

où $w_{i,j}$ est le poids du terme i dans le document j, tel que calculé (par exemple) par la méthode du *tf.idf*.

1. Dans une nouvelle classe Java, écrivez une méthode
   `public static double getSimilarity(File file1, File file2)`
   dont les deux arguments sont deux noms de fichiers `.poids` créés au TP 2 et représentant les termes pondérés contenus dans un document. La méthode devra renvoyer le score de similarité entre les deux documents en utilisant la formule ci-dessus.

   Parmi les deux techniques de normalisation (stemming, forme originale, le tout avec ou sans suppression des mots vides) mises en œuvre aux séances précédentes, choisissez celle qui vous semble la plus pertinente.

2. Écrivez ensuite une méthode
   `public static void getSimilarDocuments(File file, Set<File> fileList)`
   affichant les scores de similarité entre un document (dont le fichier `.poids` est passé en premier argument) et l'ensemble des documents du corpus. Le second argument représente le répertoire contenant l'ensemble des fichiers `.poids` du corpus.

   La liste devra être affichée par ordre décroissant de similarité des documents. Par exemple, pour le document `texte.95-1`, avec le *stemming* et la suppression des mots vides, on obtient la liste suivante :

```
texte.95-1.txt.poids      1.0
texte.95-2.txt.poids      0.2968277668834111
texte.95-97.txt.poids     0.22595508592309504
texte.95-3.txt.poids      0.20204549592848958
texte.95-4.txt.poids      0.1518870300777993
texte.95-95.txt.poids     0.06670154544020485
texte.95-60.txt.poids     0.06167813767683619
texte.95-74.txt.poids     0.05533692524208838
texte.95-33.txt.poids     0.05117282242032839
texte.95-69.txt.poids     0.04992824644524149
texte.95-100.txt.poids    0.04588571136756937
...                       ...
```

3. Comment peut-on utiliser les informations du fichier inverse (TP 3) et les méthodes créées aujourd'hui pour créer un moteur de recherche, et donc obtenir les documents les plus pertinents par rapport à une requête ?