

# Programmation de spécialité (python)

## TD 6 : analyse du contenu textuel

Julien Velcin

2024-2025

Dans ce TD, nous allons aller plus loin dans l'analyse du contenu textuel de nos documents.

### Partie 1 : travail sur les expressions régulières

**1.1** Nous allons commencer par utiliser la librairie `re` (*regular expression*). Ajoutez une fonction `search` à la classe `Corpus`, qui retourne les passages des documents contenant le mot-clef entré en paramètre. Pour cela, il vous faudra travailler sur une unique chaîne de caractères qui concatène l'intégralité des chaînes (cf. TD 3). L'idéal serait de ne pas avoir à construire cette chaîne à chaque appel de la fonction `search`, mais une seule fois au moment du premier appel.

**1.2** Ajoutez une fonction `concorde` à la classe `Corpus`, qui construit un concordancier pour une expression donnée. Il s'agit d'une légère modification de la fonction `search`. La taille du contexte est fixée par un paramètre en entrée. Vous ferez appel à la librairie `re` et à la librairie `panda` afin de stocker et retourner les résultats obtenus dans un tableau qui doit ressembler à ce qui suit :

| contexte gauche       | motif trouvé | contexte droit        |
|-----------------------|--------------|-----------------------|
| ...oilà un exemple de | texte        | trouvé au milieu d... |
| ...euxième exemple de | texte        | trouvé ailleurs ma... |

### Partie 2 : quelques statistiques

Nous allons maintenant implémenter une méthode affichant plusieurs statistiques textuelles sur le corpus (appelée `stats`). Elle doit afficher:

- Le nombre de mots différents dans le corpus
- Afficher les n mots les plus fréquents (n est un paramètre)

Pour cela, vous devrez suivre les instructions suivantes :

**2.1** Tout d'abord, vous devez implémenter une fonction `nettoyer_texte` qui prend une chaîne de caractères en entrée et lui applique une chaîne de traitements. Il faut à minima implémenter les traitements suivants : mise en minuscules (via la fonction `lower()`), remplacement des passages à la ligne `\n`. Vous pouvez aussi remplacer les ponctuations et les chiffres à l'aide d'expressions régulières appropriées.

**2.2** En bouclant sur les documents de votre corpus, vous devez construire vous même le *vocabulaire* qui sera utilisé pour décrire les textes de vos documents. Pour cela, vous utiliserez la fonction `split` en considérant plusieurs types de délimitation possible pour l'anglais (espace, tabulation, signe de ponctuation...). Le vocabulaire doit être stocké dans un dictionnaire, mais vous pouvez passer avant par un ensemble (`set`) afin d'éliminer facilement les doublons.

**2.3** Pour finir, il faut compter le nombre d'occurrences de chacun des mots de votre vocabulaire en parcourant à nouveau la liste de vos documents. Parcourir deux fois le corpus n'est évidemment pas le plus efficace, donc n'hésitez pas à chercher une solution qui vous évite cette perte de temps. L'idéal est de construire un tableau `freq` avec la librairie `pandas`.

**2.4** Vous pouvez enrichir le tableau `freq` en ajoutant une colonne indiquant le nombre de documents (*document frequency*) qui contiennent chacun des mots. Ce n'est pas la même chose que le nombre d'occurrences total des mots (*term frequency*).