Programmation de Modèles Linguistiques (I)

(L5SOPROG L3 Sciences du Langage)

2024-2025

TD3 Analyse en caractères III

Caroline Koudoro-Parfait, Sorbonne Université.
Crédit : Gaël Lejeune, Sorbonne Université

Objectifs

- Réutiliser des fonctions
- Retrouver des résultats connus
- Généraliser des procédés

L'objectif de ce TD est de faire des statistiques sur les caractères et leurs positions dans les mots.

Unigrammes : En ré-utilisant les textes utilisés dans le TD précédent calculez la fréquence de chaque caractère en français, en anglais et dans les autres langues du corpus. Comparez les résultats avec les fréquences indiquées sur la page `http://norvig.com/mayzner.html` (section: Letter Counts).

Unigrammes : Calculez la fréquence de chaque caractère par rapport à sa position dans le mot (pour les 7 premières positions dans le mot) pour le français et pour l'anglais. Pour la langue anglaise, comparez les résultats avec les fréquences mentionnées sur la page web mentionnée ci-dessus (section: Letter Counts by Position Within Word).

Bigrammes : Pour les deux textes, calculez la fréquence de chaque paire de caractères. Pour la langue anglaise, comparez les résultats avec les fréquences mentionnées sur la page mentionnée ci-dessus (section: Two-Letter Sequence (Bigram) Counts).

n-grammes : procédez de même pour d'autres valeurs de n

Exercice 1 : Calculer des fréquences de caractères

Pour ce TD nous pouvons ré-utiliser des fonctions vues dans les TDs précédents. En premier lieu la fonction `lire_fichier` et la fonction `decouper_mots`.

En effet, pour avoir des stats sur l'utilisation des caractères nous avons besoin d'ouvrir chaque texte comme une chaîne de caractère. Puis nous allons ré-exploiter nos connaissances sur les **dictionnaires Python** (ou **tableau associatifs**) et enfin trier nos résultats par ordre décroissant avec la fonction `sorted`.

Testez le code ci-dessous (après avoir bien chargé les fonctions `lire_fichier` et `decouper_mots` du dernier TD)

```
import glob, re
for chemin in glob.glob("data/*"):#Adapter si besoin
  print(chemin)#Si pas de chemin qui s'affiche, c'est pas bon !
  chaine = lire_fichier(chemin)
  nom_langue = chemin[5:7]
  dic_caracteres = {}
  for carac in chaine:
    if carac not in dic_caracteres:
      dic_caracteres[carac]=1
    else:
      dic_caracteres[carac]+=1
  print(dic_caracteres)
```

Vous remarquerez que l'affichage n'est pas très lisible, on va donc trier par ordre de fréquence, remplacez la dernière ligne par la suite d'instructions suivantes :

```
  liste_tri = [] #on va stocker dans une liste pour trier
  for caractere, effectif in dic_caracteres.items():
    liste_tri.append([effectif, caractere])#Pour le tri sur l'effectif
  liste_tri = sorted(liste_tri)
  print(liste_tri)
```

Ce qui nous donne :

```
[[1, '#'], [1, '%'], .... [3473, 'b'], [3579, 'g'],
 [3759, 'j'], [4119, 'h'], [5395, 'f'], [7005, 'v'],
 [8826, ','], [8961, "'"], [40669, 'n'], [47188, 's'],
 [88677, 'e'], [108773, ' ']]
```

On remarque que le plus fréquent c'est l'espace (" "), ce qui ne nous apprend pas grand chose. On va donc l'enlever, on va faire de même avec d'autres caractères de ponctuation. Pour ce faire on va faire un filtrage du dictionnaire `dic_caracteres` **(à vous de réfléchir où le placer)**:

```
  print(len(dic_caracteres))#AVANT
  for ponctuation in [" ", ",", "'", ":", "\n"]:
    #NB: vous pouvez ajouter des ponctuations
    del dic_caracteres[ponctuation]#on reture
  print(len(dic_caracteres))#APRES
```

Maintenant on peut ranger les résultats par langue en créant **(à vous de le faire au bon endroit)** un dictionnaire `dic_langues` qui aura pour clé le code langue et pour valeur les caractères de la langue classés par fréquence. On ajoute à la fin de la boucle le code suivant :

```
  dic_langues[nom_langue] = liste_tri
```

En sortie de boucle, on exploite cette structure de données de la façon suivante :

```
liste_langues = dic_langues.keys()
print("\t".join(liste_langues))
print("-"*30)
```

```
for cpt in range (1, 10):
  l = []
  for langue in liste_langues:
    l.append(dic_langues[langue][-cpt][1])
  print("\t".join(l))
```

Exercice 2 : Calculer les fréquences de caractères par position

En exploitant le code du TD2 et celui de l'exercice précédent calculez la fréquence de chaque caractère par rapport à sa position dans le mot (pour les 7 premières positions dans le mot) pour le français et pour l'anglais. Pour la langue anglaise, comparez les résultats avec les fréquences mentionnées sur la page web mentionnée ci-dessus (section: Two-Letter Sequence (Bigram) Counts).

Exercice 3 : (BONUS) Calculer des fréquences de bigrammes

Adapter l'Exercice 1 pour calculer la fréquences des bigrammes (paire de caractères) pour le français et pour l'anglais. Pour la langue anglaise, comparez les résultats avec les fréquences mentionnées sur la page web mentionnée ci-dessus (section: Letter Counts by Position Within Word).

Le code ci-dessous vous permettra l'extraction de n-grams d'un texte. **(à vous de le faire au bon endroit)**.

```
def get_grams(texte, n):
    debut = 0
        n_grams = []

    for i in range(len(texte)-n+1):
        n_grams.append(texte[debut:n+i])
        debut+=1

    return n_grams

phrase = "une phrase longue avec du texte"
phrase_ngrams= get_grams(phrase, 2)

print(phrase_ngrams)
```

Ce qui nous donne :

```
['un', 'ne', 'e ', ' p', 'ph', 'hr', 'ra', 'as', 'se',
'e ', ' l', 'lo', 'on', 'ng', 'gu', 'ue', 'e ', ' a',
'av', 've', 'ec', 'c ', ' d', 'du', 'u ', ' t', 'te', 'ex']
```

Devoir

Vous produirez un document d'une page (à l'aide du traitement de textes de votre choix) environ incluant :

- des statistiques sur votre corpus (nombre de caractères et de mots pour chaque langue)
- les résultats obtenus (fréquences de caractères, fréquences de caractères par rapport à sa position, etc.)
- quelques phrases de conclusion (qu'est-ce qui était attendu, qu'est-ce qui est inattendu...)

Vous déposerez sur Moodle une archive zip nommée NUMETU.zip (où NUMETU est votre numéro d'étudiant) et contenant :

- Votre code exporté au format Python `.py` (et pas ipynb)
- le PDF du document que vous avez produit

Date limite : indiquée sur le Moodle !