Fouille de textes

https://perso.limsi.fr/grouin/inalco/

Master 2 Ingénierie Linguistique (INaLCO), parcours *Ingénierie Multilingue* et *TeTraDom*.

Syllabe et syllabation

Correction

Proposition de correction : syllabation, nombre de syllabes, et schéma syllabique. Passage du texte à traiter en argument. Les espaces sont sur fond jaune pour une meilleure visualisation.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# python3 syllabation.py "auteur partir secteur applique apprendre pompier obstacle"

import glob
import re
import string
import sys

phrase = sys.argv[1]
phrase = phrase.lower()
phrase = phrase.translate(str.maketrans('', '', string.punctuation))

# Réduction des espaces multiples et de fin de phrase puis tokénisation
phrase = re.sub(' +', ' ', phrase)
phrase = re.sub(' $', '', phrase)
mots = re.split(' ', phrase)

for mot in mots:
    token = mot

    # Cas particuliers : qu- (quand), 'h' (ch, sh, ph), voyelles nasales, 'd' final,
    # semi-voyelles (yod, 'w'), réduction des consonnes géminées
    mot = re.sub('qu', 'k', mot)
    mot = re.sub('[cs]h', 'S', mot)
    mot = re.sub('ph', 'f', mot)
    mot = re.sub('h', '', mot)
    mot = re.sub('([aeiou]i?)([mn]t?)([^aeéèêiîoôu]|$)', r'~\1\3', mot)
    mot = re.sub('d$', '', mot)
    mot = re.sub('ill', 'Y', mot)
    mot = re.sub('ie', 'Ye', mot)
    mot = re.sub('oi', 'Wi', mot)
    motif = '(bb|dd|ff|ll|mm|nn|pp|rr|ss|tt)'
    if re.search(motif, mot):
        geminees = re.findall(motif, mot)
        mot = re.sub(motif, geminees[0][:1], mot)

    # Ajout d'espace après chaque groupe de voyelles
    mot = re.sub('([aàâeéèêëiîïoôöuùûüyÿ]+)', r'\1 ', mot)
    # Rattachement des consonnes finales à la syllabe précédente (dans)
    mot = re.sub(' ([bcdfghjklmnpqrstvwxz]+)$', r'\1', mot)
    # Rattachement des fins de mots avec 'e' muet à la syllabe précédente (baleine)
    mot = re.sub(' ([bcdfghjklmnpqrstvwxz]+)(e|es)\s?$', r'\1\2', mot)
    # "un groupe de deux consonnes se sépare en deux syllabes,
    # sauf si la seconde consonne est un [r] ou un [l] ou une semi-voyelle"
    mot = re.sub('\s?([bcdfghjklmnpqrstvwxz])([bcdfghjkmnpqstvwxz])', r'\1 \2', mot)
```

```
# "un groupe de trois consonnes avec un [s] au milieu subit une coupe syllabique après le [s]"
mot =re.sub('([bcdfghjklmnpqrtvwxz])s([bcdfghjklmnpqrtvwxz])', r'\1s \2', mot)
# Séparation des diphtongues (coïncidence, duo, Noël, vidéo)
mot = re.sub('(é|o|u)(a|ë|ï|o|ü|ÿ)', r'\1 \2', mot)
# Suppression des marques de pluriel ou espace finale (sauf 'des')
# et infinitif ou verbes à la 2ème du pluriel ('prier' vs. 'fier'...)
mot = re.sub('(\w\w)(e|es|s)\s?$', r'\1', mot)
mot = re.sub('(er|ez)\s?$', 'e', mot)

# Découpage en syllabes (après réduction des espaces multiples et
# suppression des espaces finales) et décompte
mot = re.sub('\s+', ' ', mot)
mot = re.sub('\s+$', '', mot)
syllabes = re.split(' ', mot)
nombre = len(syllabes)

# Production du schéma syllabique
schema = mot
schema = re.sub('[bcdfghjklmnpqrstvwxzS]', 'C', schema)
schema = re.sub('[aàâeéèêëiîïoôöuùûüyÿ]+', 'V', schema)

# Affichage
print (token, '\t', mot, '\t', schema, '\t', nombre)
```

❧ **Exercice :** vérifier ce script sur les tweets et poèmes précédents. Retrouve t-on des alexandrins (vers de 12 pieds) sur les quatrains et tercets ? Si vous n'obtenez pas le bon nombre, est-ce que la prise en compte des 'e' muets en finale devant une consonne permet d'atteindre les douze pieds ? Effectuer une évaluation de la qualité de la syllabation et du décompte de syllabes. Poursuivre sur une phonétisation basique.