Extraction de mots clés et de termes

3 novembre 2020

—Lexicologie, terminologie, dictionnairique—

Exercices.

1. Téléchargez le script exo1.4.py ou (de préférence) utilisez le vôtre s'il donne des résultats identiques.

2. Complétez exo1.4.py avec une fonction de normalisation "normalize(word)" qui prend un mot en entrée et
   — transforme les majuscules en minuscules
   — coupe les ponctuations attachées
   Evitez d'utiliser nltk ou d'autres librairies avec une fonction existante. Utilisez le module *re* pour la substitution avec une expression régulière. Appelez la fonction à l'endroit approprié dans exo1.4.py (attention : le nombre des types de mots peut changer) et ré-exécutez. Faites en sorte qu'il ne reste pas de mots qui correspondent à des strings vides après la suppression des ponctuation, ni parmi les mots unigram, ni parmi les composants de bigrams.

3. Complétez exo1.4.py de façon à ignorer les mots et les bigrams avec une fréquence inférieure à 5. (Attention : N et B changent encore).

4. Nous remplaçons la fréquence par une mesure d'association, la PMI :

$$PMI(a,b) = log\frac{P(a,b)}{P(a) \times P(b)} \tag{1}$$

   où la probabilité d'un bigram P(x,y) est estimé à partir de sa fréquence relative dans le corpus :

$$P(x,y) = log\frac{freq(x,y)}{\sum_{bigram} freq(bigram)} = log\frac{freq(x,y)}{B} \tag{2}$$

   et la probabilité d'un mot P(x) est estimé :

$$P(x) = log\frac{freq(x)}{\sum_{x} freq(x)} \tag{3}$$

   Complétez le programme pour calculer la PMI de chaque bigram et afficher les bigrams en ordre décroissant de PMI.
   Pour le calcul de log vous pouvez utiliser *math.log* du module *math*, qui calcule par défaut le logarithme naturel (ln). Utilisez le log à base 10. Aide :

$$log_a(b) = \frac{ln(b)}{ln(a)} \tag{4}$$