1 Ambiguïté moyenne en synsets

On utilise le corpus Brown (Francis et Kucera, 1964), un corpus américain d'environ 1 million de mots. La version tokénisée et taggée du corpus est accessible dans NLTK:

```
>>> from nltk.corpus import brown
>>> for w in brown.tagged_words():
...
```

1.1 Mesurer les nombres moyen et médian[1] de synsets par mot (par type et non pas par occurrence) dans le corpus Brown. On ignorera dans un premier temps la catégorie associée aux formes fléchies.

Pour récupérer les synsets d'une forme, ou d'une forme et d'une pos :

```
>>> from nltk.corpus import wordnet
>>> s1 = wordnet.synsets('dog')
```

1.2 Répétez l'opération, cette fois en prenant en compte les catégories morpho-syntaxiques. Pour information, le jeu de catégories du corpus Brown s'obtient de la manière suivante:

```
>>> nltk.help.brown_tagset()
```

Chacune de ces catégories devra être convertie en étiquette WordNet.

1.3 Fournissez les nombres de sens des 10, 100, puis 10000 mots les plus fréquents. On s'intéressera exclusivement aux mots de catégorie ouverte (n, v, adj et adv, c'est-à-dire 'n','v','a' et 'r' dans wordnet ; il est également intéressant d'observer ces distributions pour chaque catégorie séparément). Que pouvez-vous dire de ces distributions?

2 Similarité WordNet

2.1 Construire une matrice de similarité entre les 10 noms communs les plus fréquents dans Brown (parmi ceux ayant au moins un synset). Utiliser tout d'abord la mesure de similarité basée sur les chemins. Utiliser d'abord le premier sens pour chaque paire de mots, puis les paires de sens qui maximisent la similarité.

Cette mesure est implémentée dans NLTK et peut être appelée de la manière suivante:

[1] la **médiane** d'un ensemble de valeurs est la valeur *m* telle que le nombre de valeurs de l'ensemble supérieures ou égales à *m* est égal au nombre de valeurs inférieures ou égales à *m*.

```
>>> from nltk.corpus import wordnet
>>> s1 = wordnet.synsets('dog')[0]
>>> s2 = wordnet.synsets('cat')[0]
>>> wordnet.path_similarity(s1,s2)
0.20000000000000001
```

2.2 Répéter l'exercice avec la mesure de similarité de Resnik décrite en cours (res_similarity dans NLTK : voir **help(wordnet.res_similarity)**). Pour pouvoir utiliser cette métrique, vous devrez préalablement collecter les contenus informationnels sur base d'un corpus (utiliser le corpus Brown):

```
>>> ic_dict = wordnet.ic(brown)
```

2.3 Répéter l'exercice avec la mesure de similarité de Lin (lin_similarity dans NLTK).

2.4 Comparer, si possible de manière critique, les matrices obtenues avec les différentes mesures.

Suggestion : installez et utilisez le module PrettyTable pour l'affichage des matrices de similarité

3 Similarité distributionnelle

3.1 Définir une fonction qui calcule pour un ensemble de mots donnés , le ``vecteur contexte'' de chacun de ces mots: un vecteur dont les dimensions sont tous les mots du corpus et dont les valeurs sont 1 si les deux mots co-occurrent dans une fenêtre de 5 mots, et 0 sinon.

Vous pouvez accéder au corpus phrase à phrase en utilisant brown.sents()
(voir help(brown))

Appliquer cette fonction pour obtenir les vecteurs contexte des 10 noms communs les plus fréquents.

3.2 À partir de la fonction précédente, on peut définir une mesure de similarité entre deux vecteurs-contextes. Implémenter la similarité définie comme le cosinus de 2 vecteurs.

3.3 Utiliser cette similarité pour construire une nouvelle matrice entre les 10 noms communs précédemment utilisés. Comparer avec les matrices précédentes.