Prérequis

NLTK doit être installé, ainsi que les modules brown et wordnet, et tagsets, accessibles sous python :

```
>>> import nltk
>>> nltk.download()
```

⇨ sélectionner l'onglet « Corpora » puis installer les modules « brown » et « wordnet » , et sous l'onglet « Models », le module « tagsets »

L'aide sur les modules est accessible via >>> help(brown) ou help(wordnet) etc…

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():
...
```

NB : Dans tout le TP, on ignorera les formes qui n'ont aucun synset.
NB : pour accélérer la mise au point, mettez au point votre programme en ne lisant que 1000 premières formes du corpus

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'un couple forme.catégorie :

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

1.2 Répétez l'opération, cette fois en prenant en compte les catégories morpho-syntaxiques. Pour obtenir le jeu de catégories du corpus Brown : nltk.help.brown_tagset()

(nécessite l'installation du module nltk « tagsets », sous l'onglet Models)
Chacune de ces catégories devra être convertie en étiquette WordNet (faire une conversion grossière, en utilisant les premières lettres des catégories brown).

1.3 Fournissez les nombres moyens de synsets pour les x formes les plus fréquentes de chaque catégorie, avec x=100, 1000 et 10000. 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). Que pouvez-vous dire de ces nombres moyens?

Bonus : utilisez matplotlib.plot() pour tracer la courbe des nb moyens de synsets pour les x formes les plus fréquentes, avec x en abscisses allant de 1 à 1000.

[1] la **médiane** d'un ensemble de valeurs (ici les nbs de synsets des lemmes) est la valeur *m* telle que le nb de valeurs de l'ensemble supérieures ou égales à *m* est égal au nb de valeurs inférieures ou égales à *m*.

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, disponible dans NLTK (wordnet.path_similarity). Exemple :

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

Utiliser d'abord le premier sens pour chaque paire de mots, puis les paires de sens qui maximisent la similarité.

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, et pour le 1.3

3 Bonus : 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.