**Cours :** Traitement Automatique du Langage Naturel
**Promo :** Master Polytech
**Auteur :** Amir HAZEM
**Date :** Novembre 2019

# TP : Traitement du langage naturel

## Préambule :

Le but de l'ensemble des 6 séances de TP est de réaliser une chaîne complète de traitement automatique de documents écrits. Le cas d'usage sera l'extraction de relations sémantiques. À partir d'une liste de mots, il est demandé de trouver pour chaque mot les mots qui lui sont sémantiquement liés.

Pour ce faire, on s'appuiera sur les données suivantes: http://www.casmacat.eu/corpus/news-commentary.html (raw text files)

## Travail à faire :

Dans ce TP, nous nous intéresserons dans un premier temps à l'étape de prétraitement des données. Il vous est demandé de réaliser un script python qui permettra de générer à partir d'un corpus brute, une version tokenisée. À partir de cette version tokenisée, vous devez ajouter les fonctions de lemmatisation et d'étiquetage morphosyntaxique.

Dans un second temps, nous nous intéresserons à l'analyse du contenu des corpus. Ainsi, il vous est demandé de remplir un tableau de statistiques sur les données du corpus tel que : la fréquence des n-grammes (n=[1, 5]), le nombre de noms, de verbes, d'adjectifs, la longueur moyenne des phrases en nombre de mots et de caractères, etc. Enfin, nous nous intéresserons aux expressions régulières pour extraire certaines informations, par exemple : les urls, les dates, la ponctuation ( !, ?), etc. Ces informations seront à rajouter à votre tableau de statistiques.

# Exercice 1 :

Récupérer et consulter le corpus d'articles journalistiques (http://www.casmacat.eu/corpus/news-commentary.html (raw text files)).

Une fois le corpus téléchargé et consulté, effectuez les tâches suivantes :

## 1- Tokenisation

Chaque phrase du corpus d'entraînement devra être tokenisée.

Votre script prendra en entrée un fichier (par exemple fichier1.txt) et produira en sortie un fichier tokenisé (par exemple : fichier1**.tok**)

Vous pouvez utiliser la fonction de tokenisation (*nltk.word_tokenize(phrase)*) en vous appuyant sur la librairie NLTK.

## 2- Lemmatisation

Pour chaque fichier du corpus, et à partir du corpus préalablement tokenisé, fournir une version lemmatisée du corpus.

Votre script prendra en entrée par exemple le fichier fichier1**.tok** et produira en sortie le fichier fichier1**.lem**

### 3- Étiquetage morpho-syntaxique

Effectuer l'étiquetage morpho-syntaxique sur les fichiers tokenisés du corpus d'entraînement. Pour ce faire, vous vous appuierez sur la fonction pos_tag de la librairie NLTK.
Installer TreeTagger (http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/), un outil qui permet entre autres la tokenisation et l'étiquetage morphosyntaxique sur plusieurs langues.

### 4- Filtrage des mots outils

À partir des fichiers .tok et .lem générer une version ou chaque phrase aura été préalablement filtrée, c'est-à-dire, chaque phrase ne devra contenir que les mots pleins (porteur de sens). Vous vous appuierez sur le fichier stopwords.txt que vous trouverez ici
**(**http://www.amirhazem.ovh/teaching.html**)**

## Exercice 2 :

Afin d'appréhender au mieux n'importe quelle tâche en traitement automatique du langage , l'observation des données constitue un élément-clé, c'est pourquoi il vous est demandé dans ce qui suit de produire un ensemble d'observations statistiques sur les données d'entraînement.

### 1- Statistiques sur les données

Extraire à partir du corpus d'entraînement les informations suivantes :

- la fréquence des n-grammes (n=[1, 5])
- le nombre de noms, de verbes, d'adverbes et d'adjectifs.
- le nombre de phrases contenant des points d'exclamation et d'interrogation.
- la longueur des phrases en nombre de mots et de caractères.
- le TfxIdf sur les n-grammes (n=[1,2]).
- l'information mutuelle de chaque bigramme

### 2- Expressions régulières

Extraire à l'aide d'expressions régulières les informations suivantes :
- le titre (HEADLINE) de l'article
- le nom de l'auteur de l'article
- les chiffres, les nombres, les urls et les dates
- à l'aide de certains marqueurs tels que la majuscule ou l'étiquetage morphosyntaxique, extraire les entités nommées du corpus d'entraînement.

Compléter vos tableaux de statistiques avec les informations extraites grâce aux expressions régulières.

## Exercice 3 :

### 1- Représentation vectorielle

Représenter chaque mot plein du corpus par un vecteur de contexte en explorant plusieurs tailles de fenêtres contextuelles ainsi que plusieurs mesures de cooccurrences.
- taille de fenêtre w=3, 5 et 10
- mesure de cooccurrence :
    - → fréquence,
    - → information mutuelle
    - → information mutuelle positive
    - → Tf-Idf
    - → lissage de Laplace puis fréquence
    - → lissage de Laplace puis information mutuelle,

→ lissage de Laplace puis information mutuelle positive

- Calculer le cosinus pour chaque couple de mots du corpus
- Analyser les résultats obtenus

## 2- Plongements de mots

Dans ce qui suit nous allons utiliser word2vec et fastText pour construire des modèles par plongements de mots. Le but est de comparer les différents modèles en les évaluant sur la tâche d'extraction de relations sémantiques.

Pour word2vec, utilisez la librairie `gensim`. Le tutoriel de prise en main est disponible sur ce lien https://rare-technologies.com/word2vec-tutorial/
Pour fastText appuyez vous sur https://fasttext.cc/docs/en/unsupervised-tutorial.html
ou avec python ici : https://github.com/facebookresearch/fastText/tree/master/python

1- Entraîner un modèle word2vec (cbow et skipgram) ainsi qu'un modèle fastText (cbow et skipgram) en essayant plusieurs paramétrages (taille de fenêtre, dimension, etc.)
2- Calculer le cosinus pour chaque couple de mots du corpus.
3-Analyser les résultats obtenus en utilisant les données de wordsim353 :
http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/

## À rendre

- Un script contenant les différentes fonctions implémentées.
  Chaque fonction devra être commentée et accompagnée d'un exemple d'exécution.
- Un compte rendu succin contenant les résultats statistiques extraits à partir de votre script
  Chaque résultat mis dans le tableau doit pouvoir être recalculé via votre script.
  Veillez à ce que chaque fonction réponde à une seule requête (question)
- Une analyse des résultats obtenus via les différents modèles utilisés.