Travail pratique # 1 - Expressions régulières et modèles N-grammes

Automne 2017

Proposé par Luc Lamontagne

Objectif :

- Introduction à l'analyse de textes et aux difficultés que cela comporte même pour accomplir des tâches relativement simples.
- Mener des expérimentations avec les expressions régulières pour extraire des informations à partir de textes semi-structurés.
- Comprendre comment construire un modèle N-grammes à partir d'un corpus de texte.
- Mettre en application ces modèles pour détecter la langue d'un document.

Instructions :

- Disponible le 14 septembre 2017.
- Ce travail sera noté sur 100 et vaut 20% de la note du cours.
- Rapport à remettre, via pixel, le 9 octobre 2017.
- Format de rapport accepté : PDF.
- Références : chapitres 2 et 4 de la 3^{e} édition du livre de Jurafsky et Martin.
- Langage de programmation autorisé: Python ou Java.
- Vous pouvez vous inspirer de logiciels disponibles sur le web. Mais je tiens à ce que votre code soit original pour ce travail. Autrement dit, ne pas utiliser les librairies telles que NLTK ou Stanford Core NLP. Je ferai une revue de votre code ☺

TÂCHE 1 – EXTRAIRE DES INFORMATIONS DU *WORLD FACTBOOK*

L'objectif de cette première tâche est de concevoir un logiciel/script qui extrait automatiquement, sans intervention humaine, des informations provenant du *World Factbook* (WFC)[1]. Le WFC est une banque de documents qui décrit chaque pays du monde selon différents volets tels que sa géographie, sa démographie, son économie ou ses effectifs militaires.

a) En utilisant la version 2016 du WFC[2], utilisez des expressions régulières pour repérer les informations suivantes :

[1] https://www.cia.gov/library/publications/the-world-factbook/

[2] Disponible sur le site du cours dans la section *Exercices et travaux*. Comme cette publication est souvent mise à jour, il est important d'utiliser la version sur le site du cours.

NATIONAL_ANTHEM	Le nom anglais de l'hymne national (par ex. *O Canada* pour le *Canada*)
LITERACY	Le pourcentage chez les hommes (par ex. *99%* au *Canada*)
EXPORTS	La plus récente valeur, en chiffre (par ex. *$12200000* pour *Anguilla*).
GDP_REAL_GROWTH_RATE	la valeur du pourcentage le plus récent (par ex. 3.5% pour *Argentina*)
GDP_PER_CAPITA	la plus grande valeur pour les années présentées (par ex. *$37900* pour *Andorra*).
NATURAL_HAZARDS	Des risques liés aux vents? (par *NO* pour *Angola*).
EXECUTIVE_BRANCH	Le nom du *chief of state* et son titre (par *ELIZABETH II*, *Queen*). Si le titre n'est pas mentionné, mettre *Unknown.*
DIPLOMATIC_REPRESENTATION_FROM_US	La rue et la ville de l'adresse postale (par ex. *9510 Tirana Place, Dulles* pour l'Albanie). Si non disponible, donnez le numéro de boîte postale et la ville (par ex. *G. P. O. Box 323, Dhaka*). Si le pays n'a pas de représentation, mettre *No_diplomatic_rep*.

Pour faciliter la correction, je vous demande d'observer les consignes suivantes :

- Python : nom de fichier = *wfb.py*, fonction = *get_wfb_info*(*pays*, *attribut*)
- Java : classe = *WorldFactbook*, méthode = *get_wfb_info*(String *pays*, String *attribut*)

b) Quelle performance obtenez-vous avec vos expressions? Pour mener votre évaluation, utiliser le fichier de test disponible sur le site du cours. Donnez les valeurs d'exactitude (*accuracy*) ou de précision-rappel (selon ce qui s'applique le mieux à votre analyse).

c) Discutez des principales erreurs commises par votre logiciel.

TÂCHE 2 – DÉTECTER LA LANGUE D'UN DOCUMENT

a) **Modèles de langue -** Construisez un logiciel qui construit des modèles **n-grammes de caractères (n = 1 à 3)** à partir d'un fichier textuel. Ces modèles n-grammes donnent la distribution des lettres, des signes de ponctuation et des caractères spéciaux qu'on retrouve dans un texte. Quelques précisions :

- Les modèles sont au niveau des caractères et non pas des mots.
- Il n'est pas nécessaire de segmenter les phrases.
- Il n'est pas nécessaire d'ajouter de symboles de début et de fin de phrase.
- Vous pouvez faire une tokenisation directement au niveau des caractères sans en faire une au niveau des mots.

b) **Lissage 1 -** Ajoutez à votre programme une fonction pour appliquer un lissage de Laplace (*add-delta smoothing*) à vos modèles.

c) **Lissage 2 -** Ajoutez à votre programme une fonction de lissage par interpolation linéaire (avec poids constant).

d) **Évaluation de modèle n-gramme -** Ajoutez une fonction pour estimer la perplexité d'un modèle.

e) **Détection de langue -** Pour cet exercice, nous allons pour construire des modèles n-grammes permettant de déterminer la langue d'un texte.
   a. En utilisant le code développé pour les sections a)-c), construisez des modèles N-grammes pour chacun des fichiers suivants (répertoire *corpus_entrainement*) :
      i. *french-training.txt*.
      ii. *english-training.txt*.
      iii. *espanol-training.txt*.
      iv. portuguese-training.txt
   b. Pour chacun des 20 fichiers du répertoire *corpus_test,* utiliser les modèles construits à l'étape précédente et un lissage de type *add-delta* pour déterminer la langue du fichier.
   c. Analyse :
      i. Présentez les résultats obtenus.
      ii. Faites varier la valeur de delta pour en mesurer l'impact sur les résultats. Discutez des résultats obtenus en indiquant clairement ce que vous observez et les conclusions que vous en tirez.
      iii. Répéter les étapes i et ii en appliquant un lissage par interpolation. Je vous laisse libre choix sur la manière de faire varier les poids de la fonction de lissage. Indiquez la combinaison de valeurs qui donne les meilleurs résultats.
   d. (Bonus 10%) Produisez un graphique, pour chaque langue, présentant la moyenne des valeurs de perplexité en fonction du nombre de caractères. Discutez des résultats obtenus.