TRAVAIL PRATIQUE # 1 - EXPRESSIONS RÉGULIÈRES ET MODÈLES N-GRAMMES

Automne 2018

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 ou non structurés.
- Comprendre comment construire des modèles N-grammes à partir d'un corpus de texte.
- Mettre en application ces modèles pour reconstruire des phrases grammaticales.

INSTRUCTIONS :

- Disponible le 17 septembre 2018.
- Ce travail sera noté sur 100 et vaut 25% de la note du cours.
- Rapport : À remettre, via pixel, le 12 octobre 2018.
- Format du rapport : PDF.
- Références : Chapitres 2 et 3 de la 3^{e} édition du livre de Jurafsky et Martin.
- Langage de programmation: Python.
- Textes pour accomplir les tâches : disponibles sur le site Web du cours.
- Vous pouvez vous inspirer de logiciels disponibles sur le Web. Je tiens cependant à ce que votre code soit original (à moins d'indication contraire). Je ferai une revue de votre code ☺

TÂCHE 1 – LA CUISINE, C'EST SIMPLE !

a) Construisez un programme Python avec des expressions régulières pour repérer les aliments et les quantités de chaque item du fichier *ingredients.txt*. Votre code doit contenir une fonction `get_ingredient` qui retourne l'ingrédient et sa quantité. Par exemple,

```
get_ingredient("2 cuillères à café de poudre à pâte")
```

devrait retourner la paire

```
"poudre à pâte", "2 cuillères à café"
```

Quelle performance obtenez-vous avec vos expressions régulières? Évaluez la proportion de bonnes réponses que vous obtenez en utilisant le fichier *ingredients_solutions.txt*. Et discutez dans votre rapport des principales erreurs commises par votre logiciel.

b) Construisez un programme Python avec des expressions régulières pour repérer les temps de cuisson et les températures de cuisson des instructions du fichier *recettes.txt*. Analysez vos résultats et évaluez la précision et le rappel de votre programme.

N.B. : La partie b) est optionnelle si *Équipe = 1 personne*

TÂCHE 2 – COMME LE DISAIT LE PROVERBE…

Pour cette tâche, vous allez construire des modèles de langue qui vous aideront à compléter des proverbes incomplets en ajoutant les bons mots aux bons endroits. Par exemple, étant donné la phrase incomplète:

*aide-toi, le *** t'aidera*

et la liste de mots candidats [*médecin*, *ciel*, *professeur, whisky*], votre logiciel doit retourner la séquence la plus probable qui, souhaitons-le, sera " *aide-toi, le ciel t'aidera* ".

a) **Modèles de langue -** Construisez un logiciel Python qui entraîne des modèles **N-grammes de mots (N = 1 à 3)** à partir du fichier *proverbes.txt*. Quelques précisions :
   - Un proverbe = 1 ligne du fichier.
   - Il n'est pas strictement nécessaire de segmenter les phrases. De plus, j'ai retiré la ponctuation à la fin des proverbes.
   - Il n'est pas nécessaire d'ajouter des symboles de début et de fin de phrase (au choix).
   - Vous pouvez utiliser une librairie externe (par ex. NLTK) pour tokeniser les mots des proverbes.
   - Aucune normalisation n'est appliquée aux mots. Tous les caractères sont en minuscule.

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 une fonction de lissage par interpolation linéaire avec poids constant ou *stupid backoff* (au choix). Uniquement pour les équipes de 2-3 personnes !

d) **Estimation -** Ajoutez des fonctions pour estimer le logarithme de la probabilité (*logprob*) et la perplexité d'une séquence de mots à l'aide d'un modèle.

e) **Compléter les proverbes :**
   - En utilisant un lissage de type *add-delta*, évaluez la performance de vos modèles N-grammes à l'aide du fichier de test :
     - *test1.txt* si Équipe = 1 personne
     - *test2.txt* si Équipe = 2 ou 3 personnes
   - Répétez la même expérience avec un lissage par interpolation linéaire ou *stupid backoff* (uniquement pour les équipes de 2-3 personnes).
   - Analyses à inclure dans votre rapport :

i. Présentez les résultats obtenus. Les modèles modélisent-ils bien le langage utilisé dans les proverbes ? Quel est l'impact de la longueur de l'historique?
ii. Lissage de Laplace : 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. Pour les équipes de 2-3 personnes : répétez les étapes i et ii en appliquant un lissage par interpolation (ou *stupid backoff*). Je vous laisse libre de choisir la manière de faire varier les poids de la fonction de lissage (par ex. *grid search* ou aléatoire). Indiquez la combinaison de valeurs qui donne les meilleurs résultats.