TD 1 - REGEXP ET STATISTIQUE TEXTUELLE

LANGAGE NATUREL - C. RAMISCH

1. Expressions régulières

Étant donné un texte en entrée, écrivez les commandes `grep` ou `sed` qui permettent de :

(1) Trouver toutes les lignes qui ne commencent pas par une majuscule

(2) Trouver toutes les lignes qui finissent par un point d'intérrogation

(3) Trouver tous les mots contenant exactement 7 lettres

(4) Trouver toutes les dates

(5) Transformer tous les caractères "A" en "a"

(6) Séparer toutes les virgules des mots qui les précèdent

(7) Remplacer toutes les URL par un marqueur `<URL>`

2. Statistique textuelle

```
elle achète une robe bon marché
elle trouve un manteau sur internet
je mange un poulet sur un marché
il achète un poulet et une tomate
```

(1) Calculez le nombre de mots N (tokens) et le nombre de mots différents (types) apparaissent dans ce texte. Quel est le TTR ?

(2) Combien de mots apparaissent une seule fois, et quel est la proportion que cela représente ?

(3) Dessinez un rankplot, c'est-à-dire, un graphique contenant :
- sur l'axe x, le rang de chaque mot trié en ordre décroissant de nombre d'occurrences. Par exemple, le mot le plus fréquent aura un rang 1, le deuxième le plus fréquent 2, etc
- sur l'axe y, le nombre d'occurrences $c(w_i)$ correspondant au mot de rang i (vous pouvez obtenir les comptes ci-dessous, exercice 3).

(4) Si on estime que la probabilité d'un mot est son nombre d'occurrences $c(w_i)$ divisé par le nombre total de mots N, triez les phrases $s_1 =$ *elle achète un poulet*, $s_2 =$ *un poulet elle achète*, $s_3 =$ *poulet poulet poulet poulet* et $s_4 =$ *il achète un poulet* par ordre de probabilité en multipliant les probabilités individuelles des mots, c'est-à-dire $p(s) = \prod_{i=1}^{4} p(w_i)$.

3. Commandes bash/awk

Supposez que vous avez récupéré un corpus au format `txt` dans un fichier nommé `corpus.txt`, tokenisé avec des espaces entre les mots. Votre objectif est d'écrire un petit script bash qui dessine l'histogramme de fréquences des mots dans ce corpus. Un histogramme est un graphique qui indique combien de mots différents apparaissent 1 fois, 2 fois, etc. dans le corpus.

(1) Donnez la suite de commandes bash qui permet de (a) mettre un mot par ligne, (b) trier les mots et (c) compter les doublons consécutifs de façon à obtenir, pour chaque mot w, son nombre d'occurrences $c(w)$. Redirigez le résultat de ces commandes dans un fichier nommé `vocab-comptes.txt`. Par exemple, pour le corpus de l'exercice 2, ces commandes génèrent le fichier `vocab-comptes.txt` suivant :

```
2 achète
1 bon
2 elle
1 et
1 il
1 internet
1 je
1 mange
1 manteau
2 marché
2 poulet
1 robe
2 sur
1 tomate
1 trouve
4 un
2 une
```

(2) À partir du fichier `vocab-comptes.txt`, donnez la suite de commandes bash qui permet de (a) récupérer uniquement la colonne contenant les nombres d'occurrences $c(w)$ de chaque mot (b) trier ces nombres d'occurrences et (c) compter le nombre d'occurrences de chaque nombre d'occurrences pour obtenir un compteur de nombre d'occurrences $c(c(w))$. Redirigez le résultat de ces commandes dans un fichier nommé `histo.txt`. Par exemple, pour le corpus de l'exercice 2, ces commandes génèrent le fichier `histo.txt` suivant :

```
10 1
6 2
1 4
```

(3) Écrire un programme `awk` qui prend en entrée l'histogramme généré par les commandes ci-dessus et affiche un graphique avec des barres horizontales correspondant aux compteurs de nombres d'occurrences. Par exemple, pour le corpus de l'exercice 2, ce programme doit afficher le graphique suivant :

```
1 **********
2 ******
4 *
```