1 TD 4 : Deviner la langue d'un texte minuté

(correction page ??)

Abordé lors de cette séance	
programmation	fonction, listes, fichiers, dictionnaire
algorithme	histogramme, score

L'objectif est de distinguer un texte anglais d'un texte français sans avoir à le lire. Le premier réflexe consisterait à chercher la présence de mots typiquement anglais ou français. Cette direction est sans doute un bon choix lorsque le texte considéré est une œuvre littéraire. Mais sur Internet, les contenus mélangent fréquemment les deux langues : la présence de tel mot anglais n'est plus aussi discriminante. Il n'est plus aussi évident d'étiqueter un document de langue anglaise lorsque les mots anglais sont présents partout.

On ne cherche plus à déterminer la langue d'un texte mais plutôt la langue majoritaire. Il serait encore possible de compter les mots de chacune des langues à l'aide d'un dictionnaire réduit de mots anglais et français. La langue majoritaire correspondrait à celle dont les mots sont les plus fréquents. Mais construire un dictionnaire est d'abord fastidieux. Ensuite, il faudrait que celui-ci contienne des mots présents dans la plupart des textes. Il faudrait aussi étudier le problème des mots communs aux deux langues. Pour ces raisons, il paraît préférable d'étudier d'abord une direction plus simple quitte à y revenir plus tard.

Cette idée plus simple consiste à compter la fréquence des lettres. On s'attend à ce que certaines lettres soient plus fréquentes dans un texte anglais que dans un texte français.

Première demi-heure : fichiers, histogramme .

1) On s'inspire de ce qui a été fait au TD précédent : télécharger un texte [1] et le lire depuis un programme *Python*. C'est-à-dire écrire une fonction qui prend comme argument un nom de fichier et qui retourne une chaîne de caractères.

2) Toujours en s'inspirant du TD précédent, construire un histogramme comptant les occurrences de chaque lettre dans ce texte. C'est-à-dire écrire une fonction qui prend comme argument une chaîne de caractères et qui retourne un dictionnaire dont vous choisirez ce que seront les clés et les valeurs.

Seconde demi-heure : score .

3) Un texte inconnu contient 10 lettres I. Que pouvez-vous en conclure ? Pensez-vous que les fréquences de la lettre I dans un texte long et dans un texte court soient comparables ?

4) Ecrire une fonction qui normalise toutes les valeurs du dictionnaire à un.

```
def normalise (dico) :
    # faire la somme en une ligne avec la fonction sum

    # diviser
```

1. via le site http://www.gutenberg.org/ par exemple

```
    # fin
    return nouveau_dico
```

5) Appliquer votre fonction à un texte anglais et à un autre français, ... Que suggérez-vous comme indicateur pour distinguer un texte français d'un texte anglais ?

Troisième demi-heure : score et seuil

6) Choisir deux langues et calculer votre indicateur pour dix textes de chaque langue.

7) On suppose qu'on regroupe tous ces résultats en une matrice :

1. première colonne : la valeur de votre indicateur
2. seconde colonne : langue du texte

Que proposez-vous pour déterminer un seuil qui départage les deux langues selon votre indicateur ?

8) Comment proposez-vous d'implémenter la méthode que vous suggérez à la question précédente ? Une fonction... Ses paramètres... Son ou ses résultats...

Quatrième demi-heure : score et seuil

9) Implémenter la méthode suggérée. Si le résultat est constitué d'un unique nombre réel, on l'appelle le **score**.

10) Télécharger le fichier suivant : http://www.xavierdupre.fr/enseignement/tutoriels_data/articles.zip et le décompresser dans le même répertoire que votre programme. Ensuite, essayer le programme suivant.

```
import os
for fichier in os.listdir (".") :
    print fichier
```

Ecrire une fonction qui calcule votre score pour tous les textes décompressés ?

11) Est-ce que votre score marche dans tous les cas ?

Pour aller plus loin ou pour ceux qui ont fini plus tôt

12) Le score est ici un nombre unique généré à partir des documents. Admettons que nous disposons de deux scores, la fréquence de la lettre E et celle de la lettre W, comment les combiner pour obtenir un score meilleur que les deux pris séparément ?

Remarques

Ce problème s'inscrit dans un problème plus général de classification [2] [3] ou d'analyse discrimi-

2. http://en.wikipedia.org/wiki/Statistical_classification
3. http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

nante[4][5][6]. Il s'agit de déterminer un score, un indicateur numérique capable de déterminer automatiquement la langue d'un texte sans avoir à le lire. Ces indicateurs ne sont pas infaillibles, il sera toujours possible de le duper particulièrement sur des petits textes mais cela ne veut pas dire que ce score ne pourrait pas être utilisé pour estimer de façon grossière la quantité de pages internet dans chaque langue.

4. http://fr.wikipedia.org/wiki/Analyse_discriminante
5. http://fr.wikipedia.org/wiki/Analyse_discriminante_lin%C3%A9aire
6. http://cedric.cnam.fr/~saporta/discriminante.pdf