Petit Hack avec Python

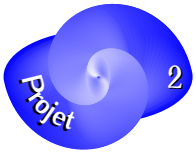
Hacking

Exercice 1

decryptage de fichiers zip securises

On peut crypter des fichiers zip avec un passwd. Le but est de faire une petite attaque en force brute pour décrypter un zip de ce type en ayant un dictionnaire de mots employés possibles. (Un dictionnaire très court sera fourni dans votre répertoire data). On essaye simplement tous les mots possibles jusqu'à trouver le bon.

- Ecrire une fonction de profil `unzipFic(file,pass)` qui extrait le contenu du fichier avec le passwd donné en argument,renvoie une exception sinon. On utilisera par exemple le module `zipfile`
- Ecrire un `main` avec des arguments en utilisant le module `optparse` qui permet d'indiquer sur la ligne de commande un dictionnaire (-d), un fichier (-f) et qui va essayer tous les mots du dictionnaire pour décrypter le fichier zip.
- Testez sur les fichiers zip fournis.
- Le passwd du fichier tressecret.zip n'est pas dans le dictionnaire fourni mais est tout de même un mot commun de la langue française, peut-être avec une flexion. Comment faire ?
- Faire un programme Python permettant de trouver le contenu de ce fichier.

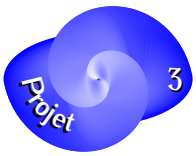


Classification par Genre

Exercice 2

NLTK et classification

Voir http://nltk.org/book3/ch06.html qui décrit comment démarrer une classification supervisée visant à identifier des éléments de Genre en anglais. Reprendre les exemples en essayant de les adapter au Français. (paragraphes 1.1 et 1.2)

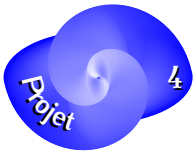


Catégorisation de documents RSS

Exercice 3

Etude de flux RSS

Etude de Flux RSS d'actualités. Prendre quelques sources importantes en français (Lemonde, AFP, Yahoo.fr, etc.) en essayant de catégoriser les articles selon la méthode décrite dans http://nltk.org/book3/ch06.html paragraphes 1.3 à 1.6



Outils d'aide à l'annotation de fichiers

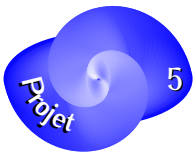
Exercice 4

Annotation

Aide fichier à l'évaluation d'outils d'annotation automatique : Comparer le fichier annoté manuellement et le fichier annoté automatiquement. On va ensuite calculer automatiquement quatre valeurs :

1. Vrais positifs
2. Vrais négatifs : balise pertinente mais pas balisé
3. Faux positifs : balisé mais ne doit pas l'être
4. Faux négatifs

On calcule ensuite : Précision = vrais positifs / (vrai positifs + faux positifs) Rappel = vrais positifs / (vrai positifs + faux négatifs) Fmesure : 2 PR/P+R Puis on transformera le texte en tableau pour pouvoir l'utiliser dans le logiciel de datamining Weka



Google Trends

Exercice 5

Google Trends

Etude lexicale des recherches sur Google trends : http://www.google.fr/trends/ Catégorisation. Dispertion dans le temps. Graphes. On peut peut-être utliser :https://pypi.python.org/pypi/pyGTrends