# Manipulation d'une collection de test en RI

A. Mercier

janvier 2021

## Préambule

Ce TP est l'occasion d'utiliser le langage de programmation Perl https://www.perl.org qui permet la manipulation des chaines de caractères et des données textuelles. L'objectif est d'utiliser et de réaliser les scripts pour le traitement de la collection de test, l'indexation et l'accès aux documents.

La dernière section propose une ouverture pour aller plus loin où il vous est proposé de faire une recherche sur le "machine learning", de tester un SRI http://terrier.org/docs/v5.0/ ou d'utiliser une bibliothèque Python pour le traitement des données textuelles https://www.nltk.org/.

Le Tp se déroulera en 3 parties

— une présentation du langage Perl et le test de quelques scripts,
— la mise en place de la collection et l'utilisation de script pour les manipulations de base et,
— une dernière partie pour la réalisation de vos propres scripts.

Vous devez rédiger un compte-rendu de TP et fournir les codes de vos scripts dans une archive avec un fichier expliquant comment les utiliser. Pour les étudiants qui doivent préparer une soutenance sur ce TP, vous devez préparer une recherche sur le machine learning comme sujet d'ouverture.

## 1 Utilisation d'une collection de test

### 1.1 Etapes à suivre

Ce TP a pour objectif de vous faire *utiliser* et **écrire** des programmes concis en Perl pour effectuer des traitements simples sur le texte des documents d'une collection de test.

Pour rappel du cours, les étapes que vous allez mettre en place :

1. Nettoyage de la collection
   — ponctuation & accents
   — lemmatisation ou radicalisation
   — sac de termes
   — stop list / mots vides
2. Analyse de la collection
   — calcul de la fréquence des termes
   — fréquence documentaire
   — observation lois de Zipf, Heaps
   — terme le plus fréquent dans la collection, dans tous les documents, par document
3. Système de recherche d'information
   — indexation
   — création des fichiers inverses
   — implémentation des modèles de calcul
   — interrogation de la collection

**Les questions des sections suivantes permettent de répondre aux éléments indiqués ci-dessus. Les réponses doivent être rédigées et illustrées dans votre compte-rendu de TP. Vous pouvez à cet effet travailler avec 2/3 documents courts pour illustrer les résultats des programmes avant de donner les résultats pour la collection de test.**

### 1.2 La collection à utiliser

De nombreuses approches en recherche d'information utilisent des benchmark pour comparer les méthodes. Télécharger l'archive contenant la collection et récupérer le dossier complet.

— Observer les différents fichiers du répertoire cacm, combien y en a-t-il ?
— Quel est leur rôle pour une expérimentation en recherche d'information ?
— Le fichier cacm est structuré, à quoi correspondent les différentes balises ? et principalement .I ? .T ? .A ? .W ?

**La réponse à ces questions doit être rédigée dans votre compte-rendu, par exemple dans votre introduction.**

## 2 Traitement des données avec Perl

Deux exemples sont disponibles Téléphone et Test
L'extension d'un fichier perl est *.pl*.
L'exécution en ligne de commande : perl monProgramme.pl
Récupérer et utiliser les fichiers d'exemples.
Documenter votre code :
1. Modifier l'entête de chaque script en indiquant : les auteurs, le nom du fichier et les objectifs du programme.
2. Commenter le code.
Vous avez besoin d'un éditeur de code qui reconnait le langage perl, par exemple Geany ou notepad++ sur windows.
Vous pouvez un télécharger un environnement pour perl sous windows
cf. https://www.perl.org/get.html

## 3 Nettoyage de la collection de test

Tout d'abord, observer la langue utilisée. Quelle est la langue utilisée ? La collection est-elle multilangue, dans ce cas quel problème cela poserait-il ?

### 3.1 Préparation des fichiers

Les fichiers sources seront traités pour enlever les caractères inutiles, les mots vides, récupérer seulement le texte, etc. Nous utiliserons plusieurs versions de script pour réaliser cela, **prévoir** le nom de vos extensions de fichiers dans le tableau ci-dessous :

| | | |
|---|---|---|
| 1 | fichier extrait de cacm.all | pas d'extension fichiers nommés CACM-XX |
| 2 | fichier sans accent ni ponctuation | |
| 3 | fichier sans mot vide | |
| 4 | fichier sans pluriel | |

Pour l'archive à remettre, ne transmettre que les scripts avec les modes d'utilisations et pas les différentes version des documents de la collection qui prennent peu de place mais beaucoup de fichiers. **Les informations de ce tableau doivent être dans votre compte-rendu.**

## 3.2 Manipulation des documents

Écrire des scripts permettant de :

1. Créer un fichier par document.
   La collection avec un fichier par document est déjà fournie dans le répertoire Collection.
   Observer les documents et comparer au fichier cacm.all. Quelles sont les portions de texte qui ont été utilisées pour créer les documents ?
2. Pour commencer, créer un fichier par document ne contenant pas de caractères spéciaux ni d'accents, supprimer les espaces inutiles. Utiliser la collection de tous les documents (répertoire Collection) et le fichier Collection qui contient la liste des noms de fichiers de la collection CACM. Regarder le fichier d'exemple...

## 3.3 Analyse de la collection

Utiliser et écrire des scripts permettant de :

1. Compter le nombre d'occurrences de chaque mot dans la collection. Le parcours du fichier Collection permet de connaitre le nom de chaque document de la collection de test. Le parcours d'un fichier permet de calculer le nombre d'occurrence d'un même mot. Ce programme écrit dans un fichier de sortie les informations sous la forme : Rang du mot, Compte du mot et Mot trié par ordre croissant des rangs. Noter la taille du vocabulaire c'est-à-dire le nombre de mots différents de la collection. Le résultat devra apparaître dans votre compte-rendu de TP.
2. Calculer le nombre moyen d'apparitions du terme le plus fréquent dans les documents. Noter le résultat, il devra apparaître dans votre compte-rendu de TP. Vous pouvez ouvrir un fichier de sortie en mode ajout (»$filepath) et tester ce programme sur plusieurs termes très fréquents et moyennement fréquents.

## 3.4 Utilisation des résultats

Ploter (gnuplot) ou utiliser un tableur pour créer un graphique visualisant la fréquence des mots (ordonnée) en fonction du rang (abscisse). Quel constat faites-vous ?

Rechercher de la documentation sur la loi de Zipf.

Améliorer votre graphique en sélectionnant quelques mots et le rang/fréquence associé.

## 3.5 Calcul des valeurs classiques

De nombreuses fonctions de correspondance utilisent les valeurs de fréquence des termes dans les documents et de fréquence documentaire (nombre de documents qui contient un terme).

1. Créer un fichier par document en filtrant les mots vides.
2. Créer un fichier contenant le vocabulaire de la collection. Le vocabulaire de la collection est sauvegardé dans un fichier (un mot par ligne).
   Nom de ce fichier :
   Taille du vocabulaire :
3. Créer un fichier sauvegardant le df (document frequency) pour chaque document. Les valeurs de fréquence documentaire sont sauvegardées dans un fichier de la forme #mot #df (un mot par ligne).
   Nom de ce fichier :
   Est-ce qu'un terme est dans tous les documents ? Si oui, que peut-on faire ?
4. Créer un fichier pour la représentation vectorielle des documents sous forme binaire. (Exemple : 10 termes pour le vocabulaire, le document contient les termes 1,3,7, on représente ce document sur une ligne par 1 :1 3 :1 7 :1).

Réaliser une version améliorée pour une représentation avec la fréquence des termes *tf*
Réaliser une troisième basée version sur *tf* . *idf* en utilisant les données du fichier des *df*.

5. Créer l'index inversé des termes à partir du vocabulaire.

### 3.6 Construction de fichier inversé

Si besoin, vous pouvez utiliser la méthode suivante pour la construction du fichier inverse en 3 étapes 1. Extraction des paires d'identifiants (terme, doc), passe complète sur la collection 2. Tri des paires suivant les id. de terme, puis les id. de docs 3. Regroupement des paires pour établir, pour chaque terme, la liste des docs

A condition que la collection puisse tenir en mémoire ce qui est le cas ici.

## 4 Modification des fichiers de la collection

1. Noter la taille du vocabulaire avec et sans les mots vides
2. Vérifier la loi de Zipf sur la collection sans les mots vides
3. Eliminer les pluriels et relancer les programmes précédant sur la nouvelle version de la collection
4. Noter la taille du vocabulaire
5. Vérifier la loi de Zipf sur la collection sans les pluriels

Si vous avez terminé, créer les scripts pour interroger la collection de test avec une requête, définir le vecteur de la requete, et accéder aux fichiers d'index.

## 5 Travail à faire pour l'exposé

Machine learning Positionnement par rapport à la RI Traitement des données volumineuses

## 6 Pour aller plus loin

### 6.1 Utilisation d'un système de recherche d'information

— Récupérer l'archive du SRI terrier.
http://terrier.org/docs/v5.1/
— Installer le SRI.
— Suivre le tutoriel pour (i) lancer la procédure d'indexation et (ii) de recherche sur la collection test proposée.
http://terrier.org/docs/v5.1/quickstart_experiments.html
http://terrier.org/docs/v5.1/configure_retrieval.html

### 6.2 Avec un autre langage de programmation

La plupart des scripts en Perl vous ont été donnés.

Utiliser Python et la bibliothèque nltk pour réaliser tout le processus de préparation de la collection. Réaliser en python scripts des questions précédentes pour le traitement des documents de la collection.

— Installation sous windows
https://www.guru99.com/download-install-nltk.html#1
— Tutoriel de l'outil de traitement de la langue
https://www.nltk.org/book/

Autre possibilité, utiliser JAVA et la bibliothèque Lucène.

# 7 Bibliographie

## 7.1 RI


1. Recherche d'information Applications, modèles et algorithmes - Data mining, décisionnel et big data Massih-Reza Amini, Eric Gaussier https://www.eyrolles.com/Informatique/Livre/recherche-d-information-9782212673760/
2. Data Science - Cours et exercices - 08/2018 - Eyrolles Massih-Reza Amini, Renaud Blanch, Marianne Clausel, Jean-Baptiste Durand, Eric Gaussier, Jérôme Malick, Christophe Picard, Vivien Quéma, Georges Quénot https://www.eyrolles.com/Informatique/Livre/data-science
3. Chiaramella Yves, Mulhem Philippe, « La recherche d'information. De la documentation automatique à la recherche d'information en contexte », Document numérique, 2007/1 (Vol. 10), p. 11-38 https://www.cairn.info/revue-document-numerique-2007-1-page-11.htm
4. Livre de C. J. van RIJSBERGEN http://www.dcs.gla.ac.uk/Keith/Preface.html
5. Cours de Jian-Yun Nie http://www-labs.iro.umontreal.ca/~nie/IFT6255/
6. Cours de M. Boughanem https://www.irit.fr/~Mohand.Boughanem/Enseignements_RI.php


## 7.2 Perl


Traduction documentation en français http://perl.mines-albi.fr/DocFr.html
Formation Perl - Sylvain Lhullier http://formation-perl.fr/guide-perl.html
Documentation Perl français http://mongueurs.net/ressources/documentation.html
Cours de F. Torre http://www.grappa.univ-lille3.fr/~torre/Enseignement/Cours/perl.php