Programmation et projet encadré - L8TI005

Flux RSS

Crédits supports : Serge Fleury

Pierre Magistry pierre.magistry@inalco.fr
Yoann Dupont yoann.dupont@sorbonne-nouvelle.fr

2024-2025

Université Sorbonne-Nouvelle
INALCO
Université Paris-Nanterre

Objectifs Généraux

Objectif du projet

Objectif global

Suivre les thématiques et les expressions qui ont fait l'actualité, au fil du temps dans les publications de différents journaux en ligne.

Moyens mis en œuvre

- git(lab)
- flux RSS
- analyse morphosyntaxique et extraction de motifs
- Cartographie des thèmes
  topic modeling

Planning prévisionnel

BàO 1 – gestion des données et du code

- Git avancé et GitLab
- Extraction (récursive) de texte depuis le XML

BàO 2 – enrichir les données

- filtrage sur métadonnées (création de sous-corpus)
- Analyse automatique
  - lemmatisation
  - morphosyntaxe et dépendances syntaxiques

BàO 3 – analyse

- Extraction de motifs syntaxiques
- *Topic Modeling* sur les données récoltées en comparant des sous-corpus.

BàO 4 – visualisation

- mise en forme des sorties (tabulaire/XML) pour la visualisation
- rapport en HTML

Déroulement du travail (1/2)

Données

- fournies sur iCampus, une large archive à explorer
- à vous de les filtrer/nettoyer/restructurer

Bonnes pratiques

- journal de bord détaillé (les suivants vous remercieront)
- avancer progressivement (traiter un seul fichier xml, puis un dossier, pas toute l'arborescence dès le départ)
- NE PAS METTRE LE CORPUS SUR LE GIT !
  à chacun de le télécharger depuis iCampus.
  Vos programmes prendrons son chemin en **argument**.

Déroulement du travail (2/2)

On gardera (presque) le même fonctionnement

- toujours trois façons de faire proposées ou trois tâches différentes
- chacun(e) code une solution sur une branche personnelle, ajoute un tag (pm-ex1r3-fin) pour indiquer un travail terminé
- chacun(e) relit la solution d'un(e) **autre** et ajoute un tag (S3-pm-ex1r2-relu) pour indiquer un travail relu
- après concertation, une branche est choisie pour être fusionnée (*merge*) vers la branche *main*.
- les choix sont expliqués dans le journal de la semaine

Votre mission

Objectifs du jour

Cette semaine (nouveau groupe)

- Se familiariser avec les données
- Identifier ce que l'on veut extraire:
  distinguer **données** (texte) et **metadonnées**
- extraire le texte d'un fichier RSS
- gérer d'abord un fichier, puis parcourir l'arborescence pour extraire le texte de tous les fichiers

La semaine suivante

- faire attention aux doublons (entre autres)
- proposer des options (ou arguments) pour cibler certaines sous-partie spécifiques du corpus (se préparer à construire des sous-corpus)

Semaine du 24

Nettoyage pour laisser la place au groupe suivant.

Du 12/02 au 17/02 - Fichier unique puis arborescence

Du 12/02 au 17/02 – lecture d'un unique fichier

1. télécharger le corpus sur iCampus et le placer dans un dossier `Corpus`
2. écrire un programme python pour extraire et afficher différents attributs (titre, description, date, etc.) de chaque article dans ce fichier. Un ensemble de fichiers prétraités sont également fournis sur iCampus pour comparer vos sorties avec celles qui sont attendues.
   trois stratégie possible :
   - R1: utiliser le module **re** (https://docs.python.org/3/library/re.html)
   - R2: utiliser le module **etree** (https://docs.python.org/fr/3/library/xml.etree.elementtree.html)
   - R3: utiliser le module **feedparser** (https://feedparser.readthedocs.io/en/latest/)
3. **le programme doit pouvoir être appelé en ligne de commande** avec le fichier xml donné en argument. On doit aussi pouvoir **importer la fonction de lecture d'un fichier** dans un autre programme python.
4. relisez le code de vos camarades et choisissez-en un à fusionner dans la branche principale

Du 12/02 au 17/02 – lecture d'une hiérarchie de fichiers

Il faudra ensuite effectuer le même traitement sur tous les fichiers, ou d'une sous-collection. Plusieurs stratégies à explorer:

- R1 : utiliser le module **pathlib** *sans* utiliser la fonction glob()
- R2 : utiliser le module **pathlib** *en utilisant* la fonction glob()
- R3 : utiliser le module **os** (os.path, os.listdir)
- (niveau bonus: charger tout dans BaseX et tout faire en bash et XQuery)

Du 19/02 au 02/03
filtrage

19/02 – filtrage

On souhaite reprendre le code précédent et l'enrichir avec la possibilité de filtrer les documents par **date**, par **flux** ou par **catégorie** (métadonnées trouvées dans le XML des flux RSS).

point importants !

1. votre code doit appeler celui des semaines précédentes
2. sélectionner **une ou plusieurs** catégories
3. bien réfléchir à où placer le filtrage !