Lecture de flux RSS

Mot d'introduction

En conservant le mode de fonctionnement du TP précédent, c'est-à-dire en utilisant des branches sur git, et du python compatible avec bash, nous allons maintenant pouvoir commencer le projet du semestre.

La première étape est de pouvoir lire et manipuler les données fournies au format XML (RSS). Il faut pouvoir en extraire le texte à analyser ainsi que les métadonnées qui serviront au filtrage. Les métadonnées que l'on considérera sont :

— l'identifiant (id) de l'article ;
— la source : le nom du journal, qu'on peut approximer avec le nom du fichier ;
— le titre de l'article ;
— le contenu de l'article ;
— la date de l'article ;
— les catégories auxquelles appartient l'article.

Pour rappel

— un nouveau groupe gitlab vous a été attribué aléatoirement. Vous y trouverez un nouveau dépôt à cloner.
— la branche **main** sert à l'avancée du projet et des exercices, mais elle ne doit contenir que du code finalisé.
— une branche **doc** sert au rendu du journal de bord (un fichier markdown par semaine),
— chaque semaine, vous devrez créer des branches individuelles réservées au travail de chaque membre du groupe.
— un tag xxx-fin doit être utilisé pour indiquer qu'un exercice et terminé et un tag xxx-relu indiquera qu'il a été relu par un tiers et est prêt à être fusionné (**merge**). Un dernier tag indiquera que le travail sur la branche **main** est terminé.

Exercice 1 Découverte du RSS - traiter un unique fichier

Pour cet exercice, chaque membre doit écrire une partie différente du programme. Répartissez-vous les rôles entre r1, r2 et r3.

1. Téléchargez les archives de corpus depuis iCampus (celle de test et celle du corpus actuel). placez les **en dehors de votre dépôt git**.
2. Depuis la branche **main**, créez une nouvelle branche pour chacun(e) et basculez dessus. Créez un fichier `rss_reader.py` dans lequel chacun(e) sera chargé d'écrire une fonction qui lit le fichier xml dont le chemin est donné en argument, et qui retourne le texte et les métadonnées des *items* du flux RSS. Chacun(e) doit donc produire le même résultat, mais avec une méthode différente.

   r1 utilisera le module **re** et des expressions régulières

   r2 utilisera le module **etree** [1]

   r3 utilisera le module **feedparser**

   des fichiers de sortie d'exemple sont disponibles afin de vous donner des exemples de sorties attendues auxquelles vous pourrez comparez ce que vous produisez.
3. Ajoutez une fonction **main()** qui permet d'appeler votre méthode depuis bash.
4. Après avoir testé votre fonction, ajoutez un tag **xy-s3e1rN-fin**, à votre branche où xy sont vos initiales, e2 est le numéro d'exercice et N votre rôle.
5. relisez le code d'un autre membre. Demandez de discuter des améliorations nécessaires ou souhaitables. ajoutez le tag **xxxx-relu** lorsque vous considérez que le code est fini.
6. **En groupe**, vous fusionnerez les trois branches vers l'une des trois. Ajoutez ensuite une fonction pour appeler une des trois méthodes (elle prendra en argument le nom de la méthode et le chemin vers un fichier RSS). Adaptez la fonction main en conséquence (on doit pouvoir choisir la méthode depuis bash)
7. Après avoir testé sur quelques fichiers, fusionnez dans la branche **main**.

1. Pensez à bien lire la doc : `https://docs.python.org/3/library/xml.etree.elementtree.html#xml.etree.ElementTree.Element.remove`

Exercice 2 Traiter une arborescence de fichiers

Une fois que le code fonctionne sur un fichier à la fois, chacun devra proposer un moyen de lire l'ensemble de l'arborescence du corpus, en suivant les mêmes principes que pour l'exercice précédent :

r1 utilisera le module **pathlib**,[2] notamment son objet `Path`, mais sans utiliser la fonction **glob()**.

r2 utilisera le module **pathlib** et sa fonction **glob()**.

r3 utilisera le module **os**[3] (notamment `os.listdir` et `os.path`.

Il faudra adapter les tags à utiliser : on utilisera notamment le tag **xy-s3e2rN-fin** à votre branche où xy sont vos initiales, e2 est le numéro d'exercice et N votre rôle.

Après avoir testé sur le corpus, fusionnez dans la branche **main**. Ajoutez un tag **s3fin**.

Exercice 3 Mise à jour du journal de bord

Pour ce travail, chaque membre renseigne sa partie du journal de bord, qui sera hébergé sur la branche **doc**. Commentez

1. vos difficultés
2. vos solutions
3. les choix lors des *merges*

N'hésitez pas à ajouter quelques indications et conseils pour le groupe qui reprendra votre code la semaine suivante !

2. `https://docs.python.org/fr/3/library/pathlib.html`
3. `https://docs.python.org/fr/3/library/os.html`