Programmation de spécialité (python)
TD 3 : acquisition de données

Julien Velcin

2023-2024

L'objectif principal de cette suite de travaux dirigés est de mettre au point un moteur de recherche d'information maison. Il ne s'agit donc pas d'utiliser l'une des (nombreuses) librairies existantes mais d'implémenter sa *propre solution*. C'est ainsi l'occasion de mettre en œuvre les connaissances acquises dans le cours tout en acquérant une meilleure compréhension des techniques souvent cachées derrière les librairies "prêtes à l'emploi" (par ex. scikit-learn ou nltk).

Partie 1 : chargement des données

Cette première partie consiste à extraire des documents à partir d'une source externe. Ces documents s'organisent en une collection qu'on appelle un *corpus*. On considère plusieurs sources d'information que l'on aimerait croiser sur un sujet de votre choix que nous appellerons **thématique**. Cette thématique devra pouvoir être simplement identifiée par un ou des mots clés, par exemple : "Coronavirus", "football", "climate" (nous travaillerons en anglais).

Nous allons d'abord simplement récupérer le contenu textuel. Commencez par créer une liste *docs*. Chaque entrée de la liste sera un document (donc un texte pour le moment). Les documents viendront de *deux* sources : Reddit et Arxiv. Vous récupérerez les documents liés à votre thématique au moyen de mots-clés, comme indiqué ci-dessous.

1.1 Reddit: à l'aide de la librairie `praw`, interrogez l'API comme expliqué dans le site web suivant. Quels sont les champs disponibles ? Quel est le champ contenant le contenu textuel ? Alimentez la liste *docs* avec ce texte uniquement. Vous pouvez déjà vous débarrasser des sauts de ligne ($\backslash n$) en les remplaçant par des espaces.

1.2 Arxiv: à l'aide de la librairie `urllib`, interrogez l'API d'Arxiv en vous aidant de l'aide en ligne. Parser les résultat grâce à la libraire `xmltodict`. Quels sont les champs disponibles ? Quel est le champ contenant le contenu textuel ? Alimentez la liste *docs*.

Partie 2 : sauvegarde des données

L'objectif est de ne pas avoir à interroger les APIs à chaque fois qu'on exécute notre programme. Pour cela :

2.1 Créez un tableau de type `DataFrame` de la librairie `pandas`. Instanciez ce tableau avec les textes qui ont été récupérés depuis les APIs. Vous utiliserez trois colonnes : une première colonne contenant l'identifiant unique du texte (par ex. un simple entier naturel), une deuxième colonne contenant le texte et une troisième colonne contenant son origine (reddit ou arxiv).

2.2 Sauvegardez ce tableau sur le disque dans un fichier de type `.csv` avec la tabulation `\t` comme séparateur.

2.3 Ajoutez le code permettant de charger directement ce tableau en mémoire, sans avoir à passer par l'interrogation des APIs.

Partie 3 : premières manipulation des données

Nous allons manipuler simplement notre corpus.

3.1 Affichez la taille de votre corpus, càd le nombre de documents.

3.2 Pour chaque document, affichez le nombre de mots et de phrases. Pour cela, vous utiliserez la fonction `split` en considérant que les mots sont séparés par des espaces et les phrases par des points. Il s'agit bien sûr d'une simplification, des techniques plus avancées ont été développées en TAL.

3.3 Supprimez de votre corpus les documents trop petits, ici qui contiennent moins de 20 caractères.

3.4 Créez une unique chaîne de caractère contenant tous les documents grâce à la fonction `join`. Cette chaîne de caractère vous sera utile dans la suite des TDs.