Programmation de spécialité (python)
TD 8 : interface avec Jupyter-Notebook

Julien Velcin

2024-2025

Partie 1 : Démarrage

L'objectif de cette première partie consiste à reproduire la mise en place d'un corpus à partir d'un nouveau jeu de données et à le faire dans le cadre d'un carnet de notes (*notebook*). Il faut veiller ici à réussir à réutiliser le code et les classes qui ont été produits dans les TD précédents, notamment en les important de manière appropriée. Répondez aux questions suivantes.

1.1 Récupérez le jeu de données intitulé `discours_US` fourni au format CSV et chargez-le grâce à la librairie pandas. N'hésitez pas à aller y jeter un coup d'oeil avec un simple éditeur de texte.

1.2 Vérifiez la distribution des auteurs des discours à l'aide de la fonction `value_counts()`.

1.3 Importez votre classe `Corpus` (cf. TD 4), créez un objet et ajoutez-y les documents du jeu de données. Les textes étant très long, découpez les discours en phrases, chacune des phrases constituera alors un document qui doit être géré par votre classe `Document`.

1.4 Testez votre objet corpus en faisant quelques recherches à l'aide de vos fonctions `search` et `concorde` (cf. TD 6).

Partie 2 : Utilisation de votre moteur de recherche

Il faut à présent pouvoir interroger le corpus avec un moteur de recherche à base de mots clefs (cf. TD 7). Répondez aux questions suivantes.

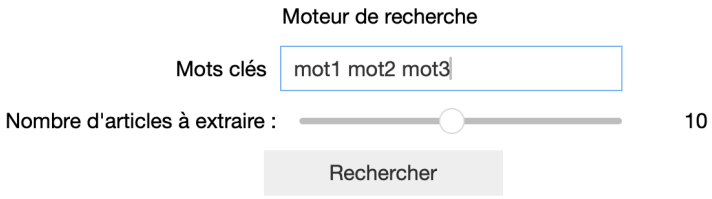
2.1 Importez votre classe `SearchEngine` et initialisez-la à l'aide de votre objet corpus.

2.2 Testez votre fonction `search` avec plusieurs requêtes pour vous assurer que tout fonctionne comme vous le souhaitez.

2.3 Ajoutez l'affichage d'un compteur dans la fonction de recherche afin de pouvoir visualiser la progression de la boucle dans le tableau des données. Pour cela, vous utiliserez la fonction `tqdm` (cf. https://tqdm.github.io).

Partie 3 : Petite interface

L'objectif est ici de proposer une interface visuelle simple basée sur des *widgets* qui peuvent être inclus dans le carnet de notes. N'hésitez pas à vous référer à la documentation (par ex. ici https://ipywidgets.readthedocs.io/en/8.1.2/examples/Widget%20Basics.html) et à des exemples trouvés sur Internet. Voici l'interface que l'on va essayer de construire :



Pour y parvenir, répondez aux questions suivantes.

3.1 Commencez par créer les premiers objets graphiques simples : l'objet "Label" (le titre), "Text" (chargé de recueillir les mots clefs) et "IntSlider" (pour indiquer le nombre de documents à retourner).

3.2 Pour afficher ces éléments graphiques, il suffit d'utiliser des objets `VBox` (placement vertical) et `HBox` (placement horizontal) qu'il faut donner à la fonction `display` :

3.3 Créez un objet `Output` qui va recevoir la sortie de la recherche qu'il faudra afficher et pensez à l'ajouter à l'interface précédente.

3.4 Ajoutez un objet `Button` qui déclenche la fonction personnalisée `clique_bouton` qu'il vous reste à écrire. Pour cela, il faut passer par la fonction `on_click` du widget.

3.5 Ecrivez la fonction `clique_bouton` qui lance la recherche en utilisant les différents éléments récupérés grâce aux widgets (les mots de la requête, le nombre de documents à retourner). Cette fonction doit appeler la fonction `search` de votre moteur de recherche et afficher le résultat dans l'objet `Output` via l'instruction `with` (cf. https://ipywidgets.readthedocs.io/en/8.1.2/examples/Output%20Widget.html#). A noter qu'il faut penser à vider la sortie à chaque appel de la fonction grâce à la commande `clear_output()` appelée sur l'objet.

3.6 Testez votre interface en faisant varier les mots clefs de la requête et le nombre de documents en sortie.

3.7 Pour aller plus loin sur cette interface, vous pouvez ajouter une recherche qui prend en compte un certain nombre de filtres. Il s'agit par exemple d'un filtre sur l'auteur du texte ou sur la date.