M2 Data Mining : TD text mining (1/2)

Julien Velcin, Université Lyon 2, Laboratoire ERIC

Janvier 2021

L'objectif principal de ces travaux dirigés est de mettre en pratique ce que nous avons vu en cours dans le cadre d'un projet suivant un fil rouge. C'est ce projet réalisé en binôme qui sera évalué, en complément de l'examen écrit final.

La commande qui vous est fixée consiste à développer un système de recherche d'information qui permet de naviguer efficacement dans un grand corpus de données textuelles. Il s'agira d'une base d'articles scientifiques publiés en Informatique et indexés dans la célèbre base DBLP (https://dblp.uni-trier.de) et quelques autres sources. En plus du titre, voire du résumé écrit en langage naturel, de l'article, vous disposez d'autres informations intéressantes : auteur(s) de l'article, la date, références, etc.

Pour parvenir à remplir cette commande, nous allons passer par un certain nombre d'étapes :

- EX 1 : acquérir les données depuis un *dump* des données au format `.json`.
- EX 2 : réaliser un certain nombre de prétraitements et construire un index sur les mots.
- EX 3 : mettre en place un moteur de recherche, notamment pour pouvoir sélectionner des sous-corpus.
- EX 4 : regrouper les documents par cluster et/ou thématique.
- EX 5 : proposer des visualisations du corpus dans un espace à faible dimension.
- EX 6 : étiqueter les catégories (clusters, thématiques) à l'aide d'expressions fréquentes / discriminantes.
- EX 7 : enrichir le système en suivant l'une des pistes proposées (ex. utilisation de la structure, scoring, etc.)

Il n'est pas toujours nécessaire d'accomplir toutes les étapes pour pouvoir continuer à avancer. Vous êtes également encouragés à essayer des méthodes qui ne sont pas données explicitement. La fin du sujet est plus libre, précisément pour vous laisser la possibilité d'expérimenter afin d'aller (un peu) plus loin que le cadre imposé au début du TD.

Exercice 1 : Acquisition des données

Le jeu de données considéré est tiré du Citation Network Dataset qui rassemble plusieurs millions d'articles : https://www.aminer.org/citation. Vous utiliserez en particulier la version 10 qui comprend : 3 079 007 articles et 25 166 994 citations, extraits le 27 octobre 2017. L'archive contient plusieurs (gros) fichiers au format `.json`. En utilisant la librairie Python du même nom, chargez ces données afin de les transformer dans un format `.csv` plus digeste en ne conservant que quelques informations : identifiant, titre et résumé (s'il existe).

Au vu de la taille et suivant les capacités de la machine que vous employez, vous pouvez vous restreindre à une année ou filtrer d'une manière de votre choix (par ex. sur la base d'un mot-clef dans le titre). L'objectif est d'avoir un jeu de données de taille suffisante mais traitable en temps raisonnable, imaginons entre 10k et 200k articles. Sauvegardez ce jeu de données dans un fichier `.csv` que vous pourrez ensuite facilement charger en mémoire pour les analyses futures.

Vous pouvez rencontrer des problèmes avec le caractère unicode nul (`'\$x00'`). Pour résoudre ce problème, vous pouvez le remplacer par un caractère vide grâce à la commande `replace`.

Exercice 2 : Construction d'un index sur les mots

Il s'agit à présent de construire un index sur un vocabulaire de mots extraits du corpus. En utilisant la librairie `scikit-learn` vue en cours, construisez ce vocabulaire en pensant à :

- supprimer les mots-outils,
- supprimer les mots trop fréquents,
- supprimer les mots trop rares.

Si vous le souhaitez, vous pouvez passer par une étape de normalisation des mots (*stemming*, lemmatisation).

Suite à cela, vous pouvez construire la matrice Documents x Termes en adoptant le schéma de pondération de votre choix. L'idéal serait de pouvoir tester plusieurs options (par ex. TF et TFxIDF) afin de pouvoir faire étudier l'impact de ce choix par la suite.

Afin de pouvoir visualiser facilement les vecteurs documents, intégrez et testez les fonctions vues en cours. Profitez-en pour calculer le nombre d'occurrences total des mots du vocabulaire et affichez les mots les plus fréquents.

Exercice 3 : Mise en place d'un moteur de recherche

L'objectif de cette partie est de construire un moteur de recherche maison à base de mots clefs. Pour cela, vous devrez suivre les étapes suivantes, comme nous l'avons vu en cours :

1. formuler une requête sous la forme d'une liste de mots,
2. construire un pseudo-document correspondant à la requête, càd un vecteur-requête dans le même espace que les documents,
3. comparer le vecteur-requête avec tous les vecteurs documents (càd les lignes de la matrice), par ex. avec une mesure cosinus,
4. trier le vecteur des scores qui en résultent,
5. afficher les documents qui ont obtenu les meilleurs scores.

Une fois que vous aurez réalisé ce processus dans son ensemble avec une version unique de prétraitement, essayez de faire varier les paramètres suivants : a) schéma de pondération (TF vs TFxIDF), b) taille du vocabulaire (par ex. 500 mots les plus fréquents vs. 5000 mots), c) avec/sans les mots-outils. Il est également intéressant de faire varier la mesure employée pour comparer les vecteurs (par ex. cosinus vs. distance euclidienne). N'oubliez pas de commenter les résultats obtenus.

Envoi du travail réalisé

La réalisation de cette première partie du TP (exercices 1 à 3) doit être envoyée à l'enseignant sous la forme d'un notebook suffisamment commenté. Attention, le notebook doit être autosuffisant et s'exécuter sur une machine équipée des principales librairies. Si vous avez besoin d'autres librairies, songez à le préciser pour qu'elle soit installée. Si vous passez par une solution en ligne, vous devrez vous assurer que l'enseignant puisse accéder à votre code et l'exécuter.

Bonne chance !