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

Julien Velcin, Université Lyon 2, Laboratoire ERIC

Janvier 2021

Exercice 4 : Structurer le corpus

Jusqu'à présent, le corpus n'est qu'un ensemble de documents. Afin de faciliter l'exploitation (interrogation, visualisation, navigation) de ce corpus, il est important de le structurer par exemple en regroupant les documents dans des catégories. Une solution consiste à une utiliser un algorithme de clustering de votre choix (k-means, modèle de mélange, etc.). Cet algorithme peut être utilisé à partir des différentes représentations vectorielles que nous avons vues en cours :

- espace des mots (avec les différents systèmes de pondération),
- espace de plongement (naïf, Doc2Vec, autres)
- espace thématique (par ex. avec LDA).

Vous pouvez essayez plusieurs solutions afin de comparer les résultats.

Exercice 5 : Visualisation du corpus

Cette étape consiste à proposer une ou plusieurs visualisation du corpus. Il s'agit par exemple de montrer :

- Les termes les plus employés dans le corpus, par exemple via des nuages de mots.
- Les co-occurrences de mots les plus observés.
- Les thématiques extraites par un algorithme comme lDA.
- Les catégories extraites à la section précédente (via les espaces de plongement et/ou les thématiques).

Une fonctionnalité intéressante serait de pouvoir sélectionner un ou plusieurs mots et de voir dans quelle partie du corpus (document, thématique, cluster) il(s) se situe(nt).

Exercice 6 : Etiqueter les catégories construites

Les catégories et/ou les thématiques ne sont pas toujours simples à interpréter. Pour aider à l'interprétation, on peut calculer pour chaque catégorie :

- les mots les plus fréquemment employés (que l'on pourrait représenter sous forme de nuage),
- les termes fréquents les plus intéressants, en utilisant par ex. des collocations,
- les termes les plus discriminants en pénalisant les termes employés dans trop de catégories ou thématiques.
- les documents les plus centraux à la catégories (par ex. proches du centre d'inertie ou de la moyenne).

Exercice 7 : Pour aller plus loin

Voilà plusieurs pistes qui vous permettront d'aller un peu plus loin dans la réalisation de cette application. Il n'est pas demandé de les explorer toutes : elles constituent des idées que vous pouvez plus ou moins développer.

7.1 Prise en compte de la structure Vous avez à votre disposition d'autres informations qui vous permettent de rapprocher deux documents lorsque : a) les articles partagent un ou plusieurs auteurs en commun, b) un article cite un autre article dans sa bibliographie, c) les articles ont été publiés dans le même journal ou la même conférence. La similarité entre deux documents peut donc se baser sur leur similarité textuelle (ce que nous avons fait précédemment) *mais également* sur d'autres informations de proximité. Vous pouvez utiliser une simple combinaison linéaire entre ces différentes sources d'information ou imaginer d'autres solutions.

7.2 Identification d'auteurs Une tâche intéressante consiste à essayer de trouver le nom des auteurs d'un article à partir de sa description textuelle. Cette tâche peut être définie comme un problème de recherche d'information dans laquelle on utilise un vecteur qui représente un auteur et on compare ce vecteur avec celui des documents. Une solution naïve consiste à place l'auteur au barycentre de vecteurs des articles qu'il a publié. Une autre solution serait d'utiliser Doc2Vec en utilisant comme tag le nom de l'auteur, ce qui permet de calculer des représentations d'auteur. La difficulté peut être de trouver une bonne manière d'évaluer la solution proposée, par exemple en calculant le rang moyen du ou des véritables auteurs dans la liste retournée par le système.

7.3 Utilisation de techniques avancées de plongement L'idée ici est de remplacer la représentation vectorielle sur le vocabulaire des mots par une représentation plus avancée (par ex. : InferSent, USE, SBert). L'objectif est clairement d'obtenir des espaces avec une meilleure estimation des similarités entre les documents.