Module *Text Mining*
Sujets des projets de fin de module
François Role – novembre 2019

Independent sets

Etant donné un corpus, l'objectif est de décrire chaque document par un ensemble de termes diversifiés ayant de fortes valeurs TF-IDF.
Chaque document est prétraité pour détecter des termes (mots ou groupes de mots)[1]. On sélectionne les termes ayant une valeur supérieure à un seuil fixé (paramètre du programme). On crée un graphe de similarité G entre ces termes. On calcule ensuite plusieurs *maximal independent sets* de G comprenant le terme t ayant le score TF-IDF le plus élevé[2].
Parmi les *maximal independent sets* ainsi créés, on identifie celui dont la somme des poids (les poids des sommets sont ici les valeurs TF-IDF des termes) est maximale. Un programme est ensuite créé qui permet pour un document donné (passé comme argument) d'afficher son graphe G avec les sommets appartenant au meilleur *independent set* en rouge et les autres en bleu.

Les données à utiliser sont NG20 facilement accessible par Scikit, et Classic3 disponible à l'adresse :

https://mycloud.mi.parisdescartes.fr/s/HwpG4bmaRp3sSCC

Rendu : un notebook Python

Co-évolution de termes

Le sujet consiste à implémenter l'approche décrite dans l'article :

Analyse visuelle de la co-évolution des termes dans les thématiques Twitter. Lambert Pépin et al. *EGC*, 2014

qui s'inscrit dans le contexte de l'étude de l'évolution des thématiques dans le temps. On donne ci-dessous un rapide résumé des objectifs de l'article.
La période étudiée est découpée en n sous-périodes. Chaque terme est représentés par des « vecteur d'évolution » dont chacune des composantes correspond au score du terme (sa fréquence) sur cette période. On peut ainsi calculer la similarité « temporelle » de chaque paire de termes. Le but est de détecter des groupes de termes évoluant conjointement.

Les données à utiliser sont les titres de DBLP que vous découperez en 20 périodes correspondant aux 20 dernières années.

Rendu : un notebook Python

Mécanisme d'attention pour le *text mining*

Le travail consiste à écrire une version Keras du code :

https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html

Vous devez essayer de suivre le plus fidèlement possible le code existant.

Rendu : un notebook Python et un document décrivant toutes les différences entre les deux implémentations (ce qui n'a pas pu être reproduit à l'identique et pourquoi)

1 Vous pouvez utiliser Gensim pour cela.

2 Un *maximal independent set* est un sous ensemble de sommets de G mutuellement non adjacents. Vous pouvez utiliser https://networkx.github.io/documentation/stable/reference/algorithms/generated/networkx.algorithms.mis.maximal_independent_set.html

Active Learning

Le travail consiste à implémenter un algorithme d'*active learning* pour la classification de textes selon le pseudo-code ci-dessous :

Input : ensemble de données étiquetées **L**, et ensemble de données non étiquetées **U**
Output : ensemble de données étiquetées **L** UNION **U**
Tant que **U** non vide

1. Entraîner un classifieur **C** (par exemple, un simple classifieur bayesien naif) en utilisant **L**
2. Utiliser **C** pour prédire les données **U**
3. Sélectionner dans **U** n instances et former un ensemble **I** contenant ces instances
4. Demander à l'expert d'étiqueter **I**
5. Faire **L** = **L** UNION **I** et **U** = **U** MOINS **I**

L'expert devra être simulé par le programme.
La technique de sélection utilisée au point 3 sera l'*Uncertainty Sampling*.
On utilisera les algorithmes de classification implémentés dans Scikit.
A chaque étape on affichera une métrique d'évaluation (utilisant les implémentation de Scikit).

Les données à utiliser sont :

le sous-ensemble NG5 du corpus NG20 accessible via Scikit. NG5 comprend les classes :

'rec.motorcycles',
'rec.sport.baseball',
'comp.graphics',
'sci.space',
'talk.politics.mideast'

Le corpus Classic3 accessible à l'adresse :

https://mycloud.mi.parisdescartes.fr/s/HwpG4bmaRp3sSCC

ATTENTION : pour NG5, on utilisera l'option remove=('headers', 'footers', 'quotes') pour une évaluation plus réaliste

Rendu : un notebook Python