IFT2015 (Structures de données) - Automne 2014

Énoncé du T.P. #4
'Tries' et engins de recherche

François Major

Directives

Ce travail est à faire **seul ou en équipe de deux**. Il doit être remis sur Studium au plus tard **vendredi le 12 décembre à 23h55**. Vous devez seulement remettre vos fichiers sources. Les implémentations doivent être écrites en Python (version 3.3 ou plus).

En plus de la justesse de votre code, tout attribut généralement souhaitable dans un code de qualité (clarté, lisibilité, absence de duplication, nommage explicite, ...) pourra être pris en compte dans l'évaluation.

1

Vous devez réaliser le projet 13.57 du livre de référence du cours (page 617).

En guise de corpus de documents, vous devez utiliser (et télécharger) la collection *Reuters-21578*[1] (plutôt que les pages d'un site web). Dans le fichier tp4.zip, joint avec cet énoncé, vous trouverez quelques classes Python vous permettant d'accéder facilement aux documents de la collection et aux termes qu'ils contiennent (excluant les *stop words*). Les figures 1 et 2 donnent un bon aperçu de ce qu'il est possible de faire avec ces classes. Vous aurez besoin d'installer la librairie *lxml*[2] pour pouvoir les utiliser.

Votre engin de recherche doit utiliser un 'trie' compressé. Plus spécifiquement, vous devez implémenter un 'trie' qui utilise un espace proportionnel au nombre de termes dans l'index, comme celui illustré à la figure 13.13 du livre du cours.

Afin d'ordonner les résultats des requêtes de l'utilisateur, il vous est suggéré d'utiliser la représentation *TF-IDF*[3]. Ainsi, vous pourrez obtenir une représentation vectorielle des documents et des requêtes de l'utilisateur et vous pourrez ensuite comparer ces représentations à l'aide, par exemple, de la mesure de similarité cosine. Il existe de nombreuses ressources décrivant cette façon de faire. En guise d'exemple, l'article 'Vector space model' sur Wikipedia[4] fournit une bonne introduction.

Si vous le désirez, vous pouvez utiliser une librairie de calcul scientifique telle NumPy[5] pour simplifier vos calculs et réduire leurs temps d'exécution. Cette librairie vous permettrait, par exemple, de représenter des documents sous forme de vecteurs et/ou d'une matrice et d'effectuer plusieurs opérations sur ces représentations (norme, produit scalaire, produit matrice-vecteur, ...).

1. `http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`
2. `http://lxml.de`
3. `http://en.wikipedia.org/wiki/Tf-idf`
4. `http://en.wikipedia.org/wiki/Vector_space_model`
5. `http://www.numpy.org`

Enfin, étant donné la petite taille du corpus de documents utilisé dans ce TP, il n'est pas nécessaire de stocker les listes d'occurence des termes sur le disque tel qu'il est décrit à la section 13.5.4 du livre.

```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct  5 2014, 20:42:22)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from documents import ReuterDocuments, DocumentTerms
>>>
>>> docs = ReuterDocuments()
925 documents read from reut2-000.sgm.
930 documents read from reut2-001.sgm.
906 documents read from reut2-002.sgm.
926 documents read from reut2-003.sgm.
897 documents read from reut2-004.sgm.
924 documents read from reut2-005.sgm.
921 documents read from reut2-006.sgm.
890 documents read from reut2-007.sgm.
917 documents read from reut2-008.sgm.
904 documents read from reut2-009.sgm.
911 documents read from reut2-010.sgm.
912 documents read from reut2-011.sgm.
930 documents read from reut2-012.sgm.
637 documents read from reut2-013.sgm.
656 documents read from reut2-014.sgm.
924 documents read from reut2-015.sgm.
947 documents read from reut2-016.sgm.
940 documents read from reut2-017.sgm.
907 documents read from reut2-018.sgm.
902 documents read from reut2-019.sgm.
777 documents read from reut2-020.sgm.
460 documents read from reut2-021.sgm.
A total of 19043 documents were read.
>>>
>>> doc = docs[1234]
>>>
>>> doc.title()
'PANCANADIAN TO SELL BRITISH INTERESTS'
>>>
>>> doc.text()
'PanCanadian Petroleum Ltd>\nsaid it agreed to sell its working interest in its North Sea\nproperties and its
British unit, Canadian Pacific Oil and Gas\nof Canada Ltd, to Whitehall Petroleum Ltd, a private British\ncomp
any.\n    PanCanadian, 87 pct-owned by Canadian Pacific Ltd CP>,\nsaid it would receive 1.7 mln British pounds
 cash (3.5 mln\nCanadian dlrs) at closing, expected in two to three months.\n    It said the deal is subject t
o approval by regulators and\nits partners in the properties, which consist of exploration\nwells. It will als
o retain a royalty interest in the\nproperties.\n Reuter\n'
```

FIGURE 1

```
>>> terms = DocumentTerms(doc)
>>>
>>> for term, count in terms:
...     print(term, count)
...
sell 1
retain 1
gas 1
unit 1
87 1
private 1
wells 1
agreed 1
cash 1
months 1
company 1
petroleum 2
canadian 3
oil 1
closing 1
working 1
5 1
canada 1
north 1
deal 1
cp 1
exploration 1
pounds 1
royalty 1
interest 2
consist 1
dlrs 1
7 1
pacific 2
1 1
mln 2
receive 1
pct 1
two 1
approval 1
pancanadian 2
expected 1
sea 1
british 3
3 1
subject 1
owned 1
properties 3
whitehall 1
three 1
regulators 1
partners 1
```

FIGURE 2