Travail pratique # 2

Recherche d'information et classification de textes

IFT-7022

Proposé par Luc Lamontagne

Automne 2017

Instructions :

- Travail individuel.
- Objectif : Mener des expérimentations afin de vous familiariser avec les techniques de recherche d'information et de classification de textes.
- Utilisation de librairies externes: autorisée, mais contrainte pour chacune des tâches.
- Rapport et logiciel à remettre le 13 novembre.
- Ce travail est noté sur 100 et vaut 20% de la note de session.

1. Recherche d'information – Quels pays…?

Nous utiliserons pour cette tâche le *World Factbook* qui donne une description des différents pays du monde. L'objectif de cette tâche est de retrouver, à l'aide de technique de recherche d'information, les pays qui correspondent le mieux à des caractéristiques exprimées sous forme de requêtes.

Pour mener vos expérimentations, j'ai rendu disponible un fichier contenant une liste de requêtes (*liste_requetes.txt*) et un fichier indiquant les pays pertinents à chacune des requêtes (*jugements.txt*).

Pour créer votre index des différents pays, vous devez extraire du *World Factbook* les champs suivants :

- Le nom du pays
- Une description générale du pays (*Introduction/Background*)
- Des informations sur sa géographique (*Geography/Geography-note*)
- Une description sommaire de son économie (*Economy/Economy-overview*)

Contraintes :

- Choix de logiciel autorisés pour construire votre système: *Lucene*, *Solr* ou *ElasticSearch*.
- Des expressions régulières (package *re* en Python ou *java.util.regex* en Java) doivent être utilisées pour extraire les champs du *World Factbook*.
- Vous devez programmer vous-même la fonction permettant d'évaluer la mesure de *Mean Average precision* (MAP).

Les étapes à effectuer pour cette tâche sont les suivantes:

a) Téléchargez la plus récente version de *Lucene*, de *Solr* ou *ElasticSearch* . Voir les liens :
   a. Lucene - http://lucene.apache.org/
   b. Solr - http://lucene.apache.org/solr/
   c. ElasticSearch - https://www.elastic.co/products/elasticsearch.

b) Téléchargez, sur le site du cours, les fichiers nécessaires pour mener vos expérimentations.

c) Extraire du *World Factbook* les champs mentionnés précédemment pour chacun des pays.

d) Indexez chacun des pays comme un document en conservant chaque information dans des champs différents.

e) En utilisant le fichier de requêtes et le fichier de jugement de pertinence, estimez le *mean average precision (MAP)* de votre système de recherche d'information.

2. Classification de questions

À partir d'un corpus de questions disponibles sur le site du cours (fichier *questions.txt*), comparez la performance de 2 algorithmes d'apprentissage (ou plus) pour déterminer le type de chacune des questions. Le corpus contient quelques centaines de questions sous la forme *CLASSE texte de la question*. Par exemple :

QUANTITY How many feet are there in a mile ?

Les classes de questions que l'on retrouve dans le corpus sont: DEFINITION, PERSON, LOCATION, TEMPORAL et QUANTITY.

Consignes/Contraintes :

- Vous devez évaluer la performance d'un classificateur naïf bayésien. Je vous laisse libre choix pour l' (les) autre(s) algorithme(s).
- Les librairies logicielles autorisées sont *scikit-learn* (Python) et *MLlib* (Java ou Python). Pour tout autre choix, obtenir mon autorisation au préalable.
- Comme les questions sont tokénisées, je n'impose aucune exigence particulière pour le prétraitement des questions.

Comme il s'agit d'un problème multiclasses, vous pouvez soit construire un seul classificateur global qui discrimine parmi les 5 classes de questions ou des classificateurs individuels pour chacune des classes. Voir le chapitre 7 de la 3e édition du livre *Jurafski & Martin* et/ou la documentation de votre librairie logicielle pour plus d'informations.

Évaluez la performance des algorithmes en terme de précision, de rappel et d'exactitude. Discuter des principales sources d'erreur.

Au besoin, pour plus d'informations sur la classification de questions, vous pouvez vous référer aux articles suivants:

- Xin Li et Dan Roth (2002) *Learning Question Classifiers*, COLING'02. http://ucrel.lancs.ac.uk/acl/C/C02/C02-1150.pdf
- P. Blunsom, K. Kocik, J. Curran (2006) *Question Classification with Log-Linear Models*, SIGIR'06. http://www.it.usyd.edu.au/~james/pubs/pdf/sigir06qc.pdf

3. Classification - Analyse de sentiments

Cette tâche vise à classifier des critiques de livres selon leur polarité. Le corpus disponible sur le site du cours (*books.zip*) contient des critiques positives ou négatives (1000 de chaque type) que vous pourrez utiliser pour entraîner et évaluer des classificateurs binaires.

Tout comme pour la tâche précédente, vous devez :

- comparer un minimum de deux algorithmes d'apprentissage (au choix),
- évaluer la performance de ces algorithmes en terme de précision/rappel/exactitude
- discuter des principales sources d'erreur de vos algorithmes.

Consignes/Contraintes :

- Librairies logicielles autorisées : comme à la tâche 2.
- Vous devez faire le prétraitement de vos textes à l'aide d'un logiciel de traitement automatique de la langue (NLP). Les choix autorisés sont NLTK (Python) et Stanford CoreNLP (Java). Pour tout autre choix, obtenir mon approbation.
- Normalisation des mots et sélection des *features* : comparer les performances obtenues en normalisant vos textes à l'aide des critères suivants:
    - La fréquence des termes dans le corpus (par ex. retirer tous les mots dont la fréquence est inférieure à 1).

- Les mots outils (*stop words*). Plusieurs listes sont disponibles sur le Web, par ex. :
  - https://www.textfixer.com/tutorials/common-english-words.txt
  - http://xpo6.com/download-stop-word-list/
- En appliquant un *stemming* sur les mots. Plusieurs implémentations sont disponibles sur le Web, vous avez libre choix.
- La catégorie lexicale des mots (par ex. ne retenir que les noms, adjectifs, verbes et adverbes). Vous devez faire une analyse lexicale (*POS tagging*) des textes pour faire ce filtrage.

4. À remettre

- Votre projet et vos fichiers d'expérimentations (afin de nous permettre d'exécuter votre code dans les mêmes conditions que les vôtres).
- Un rapport qui décrit :
  - Vos choix d'outils et le code que vous avez développé;
  - Les expérimentations que vous avez menées;
  - Les résultats que vous avez obtenus;
  - Les conclusions que vous tirez de vos expérimentations;
  - Des instructions pour installer et exécuter vos projets.

5. Évaluation du travail

▪ Recherche d'information	30%
▪ Classification de questions	20%
▪ Analyse de sentiments	40 %
▪ Qualité du rapport	10 %