Extraction de connaissances à partir de données
HMIN208

Projet
Classification de documents par opinion

Encadrement : Dino Ienco, Pascal Poncelet, Andon Tchechmedjiev, Konstantin Todorov
Février 2018

Le but de ce projet consiste à mettre en oeuvre et évaluer des méthodes de classification de documents exprimant des opinions.

Les programmes pour réaliser ce projet peuvent être écrits en Java, Python, Php, R. Vous pouvez utiliser également les algorithmes de classification de Weka, Python (Sickit-learn) ou R. Vous préciserez explicitement les bibliothèques utilisées.

Le corpus

Un jeu de données textuelles est mis à votre disposition sur Moodle. Il s'agit d'un corpus d'à peu près 8000 documents contenant des avis d'internautes sur des films. A chaque document est associé sa polarité selon l'avis (+1 : positif, -1 : négatif). Le fichier des documents est formaté dans un tableau cvs (un avis par ligne), un autre fichier csv contient les polarités d'avis par document (-1/+1). Une correspondance directe existe entre les numéros des lignes des documents et des polarités.

La classification de documents textuels nécessite de transformer les documents en vecteurs de mots/tokens pour ensuite pouvoir utiliser un classifieur. En fonction des librairies utilisées pour la réalisation de votre projet il existe de nombreuses possibilités. Par exemple dans Weka la fonction stringToWordVector a pour objectif de transformer automatiquement le contenu du document dans un vecteur de mots. Il existe bien entendu de nombreux filtres disponibles dans cette fonction : stop-list, nombre de mots, choix de la mesure (booléens, tf-idf, etc).

Etape 1 : Prétraitements des documents

Vous utiliserez les différents types de données d'entrée selon les prétraitements. Le but est d'utiliser vos textes avec différentes informations, en préparant au moins 3 versions du corpus :

(1) Textes bruts (avec ou sans suppression de stop-words),
(2) Textes lemmatisés,
(3) Textes lemmatisés avec analyse morphosyntaxique (à l'aide, par exemple, de l'outil Tree-tagger vu en cours).

Pensez à la possibilité d'appliquer des prétraitements personnalisés selon vos besoins et votre corpus (e.g., liste de stop-words personnalisée). Avec l'outil Tree-tagger[1] vous pouvez ajouter à chaque mot sa catégorie grammaticale et enrichir l'espace des descripteurs et ainsi comprendre si cette information peut aider (ou non) à classer vos documents. Attention au format d'entrée utilisé par Tree-tagger. Vous pouvez également vous intéresser à d'autres types de connaissances linguistiques (par exemple, la terminologie, la sémantique, l'usage d'un dictionnaire de mots polarisés, etc.).

[1] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

Etape 2 : Mise en oeuvre d'algorithmes de classification

La suite du travail consistera à utiliser soit Weka, Sickit Learn ou Ret à évaluer rigoureusement les résultats de classification obtenus en prenant en entrée les différents corpus préparés dans l'étape précédent. Rappelons que de nombreuses approches d'apprentissage peuvent alors être utilisées pour la classification de textes :

- K plus proches voisins,
- Arbres de décisions,
- Naïve Bayes,
- Machines à support de vecteurs
- Les règles d'association

Paramétrage : Pour chaque méthode de classification, il existe plusieurs paramètres à choisir, tels que le paramètre K de l'algorithme des KPPV, le noyau pour les SVM, le support pour les règles, *etc*.

Etape 3 : Analyse

Une analyse complète de la qualité de la classification selon les différents types d'entrées et types de prétraitements par modèle de classification et paramétrage doit être proposée. Autrement dit, les combinaisons différentes de modèle + paramètres + pondération + type de données d'entrée donneront des performances différentes. A vous de les comparer et configurer votre fonction de classification pour qu'elle soit le plus performante possible sur les données de test en proposant une analyse approfondie de vos résultats.

Remarque 1 : Le thème de la classification des textes laisse penser que certains types de mots peuvent se révéler particulièrement discriminants (par exemple, les adjectifs pour la classification d'opinion). Une discussion sur l'influence de tels marqueurs morphosyntaxiques sera bienvenue.

Remarque 2 : Différents traitements (par exemple, pondérations, algorithmes de fouille de données comme l'extraction des règles d'association) ont été proposés par les encadrants du projet. Vous pourrez vous en inspirer pour présenter des résultats complémentaires aux résultats de classification.

Remarque 3 : Attention à la négation : est-ce qu'une opinion contenant le mot “génial” est forcément positive...? Comment traiter ce problème ?

Etape 5 : Challenge

Un seconde jeu de données de plus petite taille contenant des avis uniquement (sans polarités) vous sera fourni peu avant l'examen. Il sera formaté de la même façon que les données que vous avez utilisé précédemment. Le but de ce challenge est d'évaluer le ou les meilleurs classifieurs (avec leurs paramètres, descripteurs et prétraitements associés) que vous avez construit auparavant sur ces nouvelles données. Une comparaison des résultats obtenus par les différents groupes sera effectué. Les trois meilleurs groupes pourront bénéficier de points complémentaires dans l'évaluation du TP. Attention, l'objectif de ce challenge n'est pas de modifier vos classifieurs dans la mesure où les résultats seront ceux qui vous seront rendus. Par contre il est important d'analyser les résultats obtenus.

Organisation

Le travail s'effectuera en groupes de 4 à 5 étudiants.
Une soutenance orale de 20 minutes suivie de 10 minutes de question est prévue à la fin du semestre. La soutenance a pour objectif de présenter vos approches, vos choix et de mettre en avant également l'analyse des résultats que vous avez obtenu. Lors de la présentation vous présenterez également les résultats obtenus dans le challenge et discuterez des résultats (meilleurs, moins bons, pourquoi, etc). Il est inutile de perdre du temps lors de la présentation sur les données initiales (qui sont communes) ni sur la problématique du projet.
Le rendu final, une semaine avant la soutenance, consiste en :

1. Rapport de max 15 pages
2. Les codes de l'ensemble des traitements automatiques
3. Les résultats du challenge