Fouille de données

HMIN326

Projet

Classification de documents par opinion

Encadrement :
Dino Ienco, Konstantin Todorov, Pascal Poncelet

Octobre 2020

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

Le corpus

Un jeu de données textuelles vous est mis à disposition sur Moodle. Il s'agit d'un corpus de 10000 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 les polarités.

Etape 1 : Transformation des données et vectorisation

WEKA prend en entrée des fichiers du type .arff. Le but ici est de transformer vos données de départ en fichiers .arff, compatibles avec Weka. Les programmes pourront être développés en Perl, Python, PHP, Java ou autres. Par la suite, les valeurs textuelles doivent être rendues numériques en utilisant une pondération fréquentielle (tf-idf, tf, ou autres). Cela peut se faire à l'aide de la fonction stringToWordVector, un filtre de WEKA. Pensez à la normalisation de vos vecteurs.
Il est possible d'utiliser Scikit Learn à la place de WEKA pour effectuer vos transformations et vectorisation.

Etape 2 : 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 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 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 (http://www.cis.unimuenchen.de/~schmid/tools/TreeTagger/) 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 votre corpus. 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, etc.).
Si vous utilisez Scikit Learn, il est possible d'utiliser les librairies NLTK ou Scapy pour faire l'analyse lexicale.

Etape 3 : Mise en oeuvre d'algorithmes de classification

La suite du travail consistera à utiliser Weka et 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
- Ensemble classifier

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.
Dans le cas de Scikit Learn, il est possible d'utiliser la fonction gridsearchCV pour tester différents classifieurs.

Etape 4 : 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 ?

Organisation

- Le travail s'effectuera en groupes de maximum 3 étudiants.

- Une soutenance orale de 20 minutes suivie de 10 minutes de questions est prévue à la fin du semestre. La soutenance a pour objectif de vous permettre de présenter vos approches, vos choix et de mettre en avant également l'analyse des résultats que vous avez obtenu. Il est inutile de perdre du temps lors de la présentation sur les données initiales (qui sont communes) ni sur la problématique du projet ou bien la théorie des méthodes utilisées.

- Le rendu final consiste en :
    1. Rapport de **max 10 pages décrivant les principaux pré-traitements et choix effectués ainsi que l'analyse de résultats. Ici aussi il n'est pas nécessaire de parler des données**

qui sont communes à tout le monde.
2. Les codes de l'ensemble des traitements automatiques