Classification et prétraitement de textes

Proposé par Luc Lamontagne
Automne 2018

Instructions :

- Travail en équipe de 1 à 3 personnes.
- Objectifs :
  - Se familiariser avec un logiciel d'apprentissage automatique.
  - Prétraiter des textes à l'aide de techniques de traitement de la langue (par ex. tokenisation, *stemming*, lemmatisation, analyse grammaticale).
  - Mener des expérimentations pour construire des classificateurs de textes.
  - Évaluer et analyser les résultats que vous obtenez.
- Utilisation de librairies externes:
  - Logiciels autorisés: Python, Scikit-learn, NLTK, Stanford CoreNLP (avec interface Python).
  - Exclusion : package NLTK.sentiment
  - Obtenir mon autorisation pour tout autre choix. Me consulter en cas de doute.
- Rapport et logiciel à remettre le 13 novembre.
- Ce travail est noté sur 100 et vaut 25% de la note de session.

1. Analyse de sentiment

Cette tâche vise à classifier des critiques de produits selon leur polarité (positive ou négative). Le corpus disponible sur le site du cours contient des critiques positives ou négatives (1000 de chaque type) que vous utiliserez pour entraîner des classificateurs binaires et évaluer leur performance.

Votre travail consiste à comparer un minimum de deux algorithmes d'apprentissage pour accomplir cette tâche : naïf bayésien (*naive bayes*) et régression logistique. Vous devrez évaluer la performance de ces algorithmes en terme de précision/rappel et d'exactitude (*accuracy*) et présenter une analyse de vos résultats.

De plus, vous devez faire le prétraitement de vos textes à l'aide d'un logiciel de traitement automatique de la langue (TALN/NLP). Les prétraitements à effectuer portent sur des techniques que nous avons étudiées dans le cours:

- La **normalisation** de mots :
  - En appliquant un ***stemming*** sur les mots. Quelques implantations sont disponibles dans NLTK. Sinon plusieurs implémentations sont disponibles sur le Web. Vous avez libre choix.
  - En effectuant une **lemmatisation** des mots. Par ex. voir la classe *WordNetLemmatizer* de NLTK.
- La **sélection d'attributs** (*features*) :
  - Éliminer les mots dont la **fréquence** dans le corpus est faible (par ex. retirer tous les mots dont la fréquence = 1).
  - Retirer les **mots outils** (*stop words*). Voir NLTK ou Scikit-learn.
  - Ne garder que les mots appartenant à des **classes ouvertes** (c.-à-d. les noms, adjectifs, verbes et adverbes). Vous devez faire une analyse grammaticale (*POS*

tagging) des textes pour identifier ces mots.

- **Valeurs** d'attributs - 3 options sont possibles :
  - Les **comptes** de mots dans chacun des textes.
  - La **présence** des mots dans chacun des textes (valeur 0-1).
  - La valeur **tf-idf** de chaque mot (disponible dans Scikit-learn).
- **Autres attributs** :
  - Le **nombre de mots** dont la polarité est positive ou négative. Vous pouvez utiliser SentiWordnet (NLTK) ou un autre lexique pour estimer cet attribut.

Les configurations minimales à évaluer dans vos expérimentations sont les suivantes :

	Équipe de 1 personne	**Équipe de 2-3 personnes**
Classificateurs	*Naive bayes*, régression logistique	*Naive bayes*, régression logistique
Normalisation	*Stemming*, aucune normalisation	*Stemming*, lemmatisation, aucune normalisation
Sélection d'attributs	Tous les attributs, avec classes ouvertes seulement, sans les mots outils	Tous les attributs, avec filtrage en fréquence, sans les mots outils, avec classes ouvertes seulement
Valeurs d'attributs	Compte de mots	Compte, présence et tf-idf
Autres attributs	Nombre de mots positifs/négatifs	Nombre de mots positifs/négatifs

Analysez clairement dans votre rapport le comportement de chacune des configurations et identifiez l'option qui vous semble la plus intéressante (ou les options le cas échéant).

2. Identification de la langue d'un texte

L'objectif de cette tâche est de construire un classificateur qui **identifie la langue d'un texte** à partir de **sous-séquences de caractères**. Les attributs que vous utiliserez sont les séquences de lettres, de signes de ponctuation et de caractères spéciaux qu'on retrouve dans un texte. Par exemple, pour déterminer la langue de la phrase[1] :

This_is_an_example.

vous pourriez utiliser comme attributs les trigrammes de caractères suivants:

[thi, his , is_, s_i, _is , is_, s_i, _is , is_, s_a, _an , …, mpl, ple , le.]

Je vous laisse libre choix sur le type d'algorithmes à utiliser pour construire un classificateur supervisé. Comme à la tâche précédente, je vous demande au minimum de comparer **les performances de 2 algorithmes d'apprentissage** (par ex. *naive bayes* et régression logistique). Pour entraîner votre module d'identification de langue, vous trouverez sur le site du cours des fichiers pour 4 langues : français, anglais, espagnol et portugais. Vous trouverez également sur le site des fichiers pour évaluer les performances de votre module.

Les fonctions de prétraitement à appliquer sur les textes sont :

- segmenter les textes en phrase. Il n'est pas nécessaire de mettre des symboles de début et de fin de phrases comme on le ferait pour construire des modèles N-grammes.
- convertir les caractères en minuscule.
- extraire les sous-séquences de caractères (c.-à-d. les n-grammes) pour obtenir les attributs de vos classificateurs.

[1] Les espaces ont été remplacés par des _ pour améliorer la lisibilité.

Présentez dans votre rapport les résultats que vous obtenez avec des sous-séquences de 1, 2 et 3 caractères.

Pour les équipes de 2-3 personnes, produisez un graphique qui présente les performances de votre module en fonction du nombre de caractères lu dans un texte (par ex. 5, 10, 15, 20, etc.).

3. 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 votre code.

4. Évaluation du travail

Analyse de sentiment	60%
Identification de langue	30%
Qualité du rapport	10%

Les critères pour évaluer les 2 premiers items sont: la conformité des travaux avec l'énoncé, une bonne méthodologie pour mener les expérimentations, les configurations qui sont évaluées, une présentation claire et une analyse pertinente des résultats.

5. Liens utiles

Scikit-learn : http://scikit-learn.org/stable/index.html

Quelques classes/fonctions utiles : MultinomialNaiveBayes, LogisticRegression, CountVectorizer, TfidfTransformer, TfidfVectorizer, cross_val_score, cross_validate, métriques d'évaluation.

NLTK : https://www.nltk.org