LETTRES SORBONNE UNIVERSITÉ

2022-2023

M1SOL023 Méthodologie de la Recherche en Langue et Informatique

Projet
Classification

Laurence Devillers et Nour El Houda Ben Chaabene, Sorbonne Université

Le projet en résumé

- Préparer des données pour la classification
- Utiliser différents algorithmes de classification
- Discuter et comparer des résultats
- Rapport intermédiaire pour le 13/12/2022
- Rendu final : Mardi 27/12/2022 sur Moodle (code + rapport)

Ce projet (rapport + code) constituera la moitié de votre note de contrôle continu et fera l'objet d'un oral pendant les examens de Janvier. Les doubles séances de TD du 22/11/2022 et du 06/12/2022 seront intégralement consacrées au suivi du projet de même que la séance de révision du 13/12/2022 (13h-14h30).

1 Objectif

Vous devrez traiter au moins deux des jeux de données proposées (liste provisoire) :

Tâche	Langue	Lien du jeu de données
Détection de polarité dans les tweets	FR	https://www.kaggle.com/datasets/hbaflast/french-twitter-sentiment-analysis
Détection de SPAM	EN	https://raw.githubusercontent.com/Balakishan77/Spam-Email-Classifier/master/spamham.csv
Classification thématique	EN	http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

L'objectif est de mettre en place une chaîne de traitement de classification automatique et d'en analyser les résultats.

Vous aurez à réaliser les étapes suivantes:

1. **Lire** le jeu de données pour récupérer les classes (y) et les instances (X)
2. **Vectoriser** les instances pour les exploiter avec différents classifieurs
3. **Séparer** entre jeu d'entraînement (train set) et jeu de test (test set)
4. **Mesurer** l'efficacité de ces différents classifieurs: exactitude, précision, rappel, F-mesure . . .
5. **Commenter** et **comparer** les résultats dans un document

C'est l'étape 3 qui constitue le cœur du projet, vous allez comme en TD démarrer avec un simple sac de mots (Bag of Words) avant de tester différentes manières de raffiner votre méthode :

- **Tests de classifieurs :** classifieurs bayésiens, arbres de décisions, forêts d'arbres aléatoires, SVM
- **Nettoyage :** enlever les mots outils, les mots rares . . .
- **N-grammes :** utilisation de n-grammes de mots ou de caractères en faisant varier la taille de N
- **Pondération :** donner une importance plus grande à certains mots grâce au tf-idf par exemple
- **Enrichissement :** ajouter les étiquettes morpho-syntaxiques (POS)

Pour voir l'influence de ces "raffinements", il faut regarder leur influence sur les résultats. Par exemple, si vous enlevez les mots outils c'est par ce que vous pouvez montrer que ça améliore les résultats.

Pour ne pas perdre votre temps à attendre devant votre ´ecran des résultats, faites attention au **temps de calcul**. Quelques conseils :

- Travaillez d'abord sur des échantillons pour vérifier que votre chaîne de traitement fonctionne bien
- Stockez les résultats de classification (par exemple des fichiers Json) de manière à pouvoir les consulter sans devoir lancer toute la chaîne de traitement
- Certains classifieurs sont plus gourmands en temps de calcul, en particulier les SVM, tenez en compte quand vous lancez vos expériences

2 Plan du rapport

C'est un plan type que nous proposons ci-dessous, vous pouvez vous en écarter dans une certaine mesure (en particulier le choix des titres) mais la plupart des éléments ci-dessous sont nécessaires.

Mise en page

Vous pouvez faire la mise en page vous même mais vous pouvez aussi utiliser le modèle (template) du mémoire (Word/Writer ou latex) afin de prendre l'habitude de produire des documents "propres" avec en particulier :

- une table des matières
- des légendes à chaque tableau, figure . . .
- des mentions explicites de ce qui n'est pas de vous (citations, références. . .)
- une bibliographie

On ne juge pas un rapport simplement en nombre de pages, si vous n'avez que 9 pages mais qu'elles sont de qualité ce n'est pas très grave. Dès lors, nous ne voulons pas voir de "remplissage" et en particulier :

- pas de grandes captures d'écran injustifiées
- pas de sauts de pages intempestifs ou de grands "vides"
- pas de blocs de codes non commentés ou non contextualisés (= non décrits dans le corps du texte)

Introduction (1 page)

Que l'on sache de quoi vous parlez:

- Décrire ce que l'on cherche à faire (la tâche)
- Parler brièvement du type de données traité
- Avec quels outils travaillez vous ?
- Enoncer le plan

Etat de l'art (1 page mini)

C'est très important de donner des références, vous pouvez reprendre ce que vous avez vu en cours et ajouter des recherches personnelles (articles scientifiques en priorité mais aussi blogs et tutoriels pourquoi pas). Pour un projet comme celui-ci, l'état de l'art peut être court mais n'oubliez pas que c'est un cours de **Méthodologie**, il s'agit donc de prendre de bonnes habitudes pour votre mémoire du second semestre.

Vous pouvez parler :

- D'autres jeu de données pour la même tâche
- Des méthodes utilisées par d'autres (choix des caractéristiques, des classifieurs, etc)
- Des résultats obtenus par d'autres personnes sur ce même jeu de données

Jeu de données (corpus) (1 page mini)

Bien expliquer le(s) jeu(x) de données que vous avez choisi(s) :

- Quelle est la taille, quelles sont les classes
- Comment le jeu de données a été constitué, annoté
- Des statistiques sur le jeu de données :
  - Taille du corpus en nombre d'instances, phrases, mots . . .
  - Nombre d'instances par classes
  - Taille des jeux de *train* et de *test*
- Des exemples d'instances tirés du corpus pour mieux comprendre le contenu

Méthode (1 page mini)

- Quelles caractéristiques *(features)* utilisées ?
- Quels outils/méthodes pour les extraire ?
- Quels sont les différentes étapes de votre chaîne de traitement ?
- Quels sont les pré-traitements, pondérations, enrichissements utilisés ?

Résultats (2 pages mini)

Il s'agit de répondre (avec des résultats chiffrés et commentés) aux questions suivantes:

- Quels sont les caractéristiques les plus efficaces ?
- Quels classifieurs donnent les meilleurs résultats ?
- Quelles sont les instances qui résistent à la classification (exemples d'erreurs) ?

Pour présenter vos résultats (rappel, précision), n'hésitez pas à utiliser des tableaux (avec des l´egendes) et à commenter les différents résultats qui y sont présentés. Les **matrices de confusion** sont aussi des outils très utiles. Pour chaque expérience présentée, nous devons pouvoir savoir d'un coup d'œil :

- les caractéristiques utilisées
- les classes qui sont évaluées
- les classifieurs utilisés

Conclusion et Perspectives (1/2 page mini)

- Conclusion: rappeler ce qui a été fait, les résultats les plus importants (ce que l'on appelle les contributions)
- Perspectives: que pourriez vous faire pour améliorer les réesultat ou pour mieux les explorer. Pourrait-on appliquer votre méthode à d'autres jeux de données, d'autres langues . . . ?

Annexes (ne comptent pas dans les 10 pages requises)

Expliquez brièvement comment faire tourner votre code, en particulier s'il est découpé en plusieurs fichiers. Vous pouvez faire aussi figurer ici ce qui ne peut pas rentrer dans le rapport mais qui vous semble intéressant (exemples d'instances qui sont trop longs, etc).

Rendu

- Votre rapport en PDF uniquement
- Votre code python en **.ipynb** et **.py** (pas les jeux de données, nous les avons déjà),
- Si votre code est scindé en plusieurs fichiers, indiquez dans votre rapport (annexe) dans quel ordre ils doivent être exécutés