

Introduction à la fouille de textes (LZST001)

Plurital M1 - Semestre 2

Yoann Dupont yoann.dupont@sorbonne-nouvelle.fr

2022 – 2023

Descriptif de projet repris de celui déjà existant créé par Loïc Grobol.

Objectif : entraîner des classifieurs par apprentissage automatique et comparer les performances de différents algorithmes de classification sur une tâche de votre choix et un corpus que vous aurez constitué.

Consignes

- Projet à rendre le 7 mai 2023 à 23h59 *au plus tard* par rendu sur icampus.
- Projet collectif, par groupe de 2
- Aucune restriction sur la langue utilisée pour le corpus (vos comptes-rendus qui doivent être en français ou en anglais)
- Idéalement, votre corpus devrait comporter **au moins une centaine** de documents par classe (aucune limite supérieure). Cette exigence peut éventuellement être relâchée mais parlez m'en avant
- De manière générale, si vous avez un problème ou des question sur quoi que ce soit qui a trait à ce projet ou le cours en général, **contactez moi**, ne restez pas coincées ou dans l'embarras.

Le rendu devra comporter :

- Un compte-rendu traitant les points suivants :
 - Les objectifs du projet
 - Les données et ressources utilisées (origine, format, statut juridique) et les traitements opérés sur celles-ci. Pensez à donner un résumé du nombre de documents par classe et à préciser la taille des documents sur lesquels vous travaillez.
 - La méthodologie (comment vous vous êtes réparti le travail, comment vous avez identifié les problèmes et les avez résolus, différentes étapes du projet)
- **Les expériences réalisées**
 - Précisez les réglages des paramètres et le mode de calcul des performances. Vos expériences doivent être **reproductibles**
 - Vous devez tester au moins deux des trois principaux algorithmes de classification vus en cours (J48, Naive Bayes et SVM)
 - Évidemment, rien ne vous empêche d'utiliser des algorithmes que nous n'avons pas vu en cours
 - Les résultats des expériences et une discussion de ces résultats
- Les données utilisées (ou un échantillon si le volume est important)

- Tous les scripts et outils que vous avez développés (le cas échéant) pour ce projet **en format utilisable** et avec des instructions suffisantes pour que je puisse les tester. (Par exemple ne donner que des screenshots, c'est non)

Pour nous faciliter la vie à toutes

- Le compte-rendu doit être au format pdf et le nom de fichier doit être `<Prénom1>_<Nom1>-<Prénom2>_<Nom2>-fdt2023.pdf`, par exemple `Alice_Dupont-Bob_Martin-fdt2023.pdf`. Pour éviter les éventuels problèmes d'encodage ou de plateformes, merci d'utiliser une version de vos noms et prénoms en alphabet latin (désolé)
- Votre rendu doit être contenu entièrement dans une archive au format zip, 7zip ou tar+gz, nommée avec le format `<Prénom1>_<Nom1>-<Prénom2>_<Nom2>-fdt2023.zip` (avec l'extension qui convient)

Assurez-vous que tous vos fichiers textes sont encodés en utf-8.

Conseils

- N'hésitez pas à motiver vos choix dans le compte-rendu
- Écrivez ! Tenez un carnet : vos questions, un compte-rendu de vos discussions, les problèmes rencontrés, tout est bon à prendre et cela vous aidera à rédiger la documentation finale.
- Utilisez au maximum des outils de collaboration, selon vos goûts et vos compétences, par exemple :
 - [Framapad](#)
 - [Dropbox](#)
 - [Github](#) ou [Gitlab](#)
- Respectez **soigneusement** la méthode décrite dans le cours, en particulier :
- Assurez vous bien de ne pas tester sur le train
- Assurez vous que les indices évidents sont retirés de vos documents, par exemple si vous faites de la classification de recette par type et que vous avez laissé un en-tête qui contient le type c'est pas super
- En général **méfiez vous comme la peste des résultats trop beaux pour être vrais**

Ressources linguistiques

Pour constituer vos corpus de travail

- Des ressources linguistiques exploitables librement et facilement sont disponibles sur
 - [Ortolang](#)
 - [Clarin](#).
 - Vous pouvez aussi aller voir du côté de l'API twitter pour récupérer des données (qui ne sont pas nécessairement uniquement linguistiques)
- [Wikisource](#) et évidemment [Wikipédia](#) proposent des textes sous licences libres comportant déjà des métadonnées
- Vous pouvez réaliser des aspirations de sites webs avec [Gromoteur](#). Les corpus de presse se prêtent bien aux tâches de classification
- Si les corpus de tweets vous intéressent, n'hésitez pas à regarder du côté de l'API Twitter

Outils

Script de vectorisation

Un [script de vectorisation](#) basique vous est fourni pour générer des fichiers ARFF. Il se lance avec

```
python3 vectorisation.py chemin/du/corpus chemin/du/fichier/de/sortie
```

Où "chemin/du/corpus" est le chemin vers un dossier contenant un sous-dossier par classe, chaque sous-dossier contenant un fichier par document de cette classe (avec l'extension ".txt"). Par exemple

```
mon_corpus
├── culture
│   ├── t1.txt
│   ├── t2.txt
│   ├── t3.txt
│   └── t4.txt
└── societe
    ├── t5.txt
    ├── t6.txt
    └── t7.txt
```

Pour vectoriser un corpus en utilisant un vocabulaire prédéfini (par exemple pour vectoriser le corpus de test à partir du corpus d'entraînement), utilisez l'option "lexicon" comme ceci

```
python3 vectorisation.py --lexicon chemin/vers/train.arff
chemin/vers/corpus/test chemin/du/fichier/de/sortie
```

Vous pouvez aussi voir le détail des options disponibles en lançant

```
vectorisation.py --help
```

N'hésitez pas à adapter ce script pour l'adapter à vos besoins: une meilleure segmentation, un calcul des fréquences relatives ou des TFIDF

Autres ressources

- Si programmer ne vous effraie pas, allez voir du côté de [scikit-learn](#) qui propose des versions faciles à utiliser des algorithmes vu en cours.
- Si vous voulez réaliser des traitements linguistiques plus sophistiqués sur vos données, allez voir [spacy](#), [nltk](#) ou [CoreNLP](#). Évidemment, n'utilisez pas les classifieurs de documents déjà inclus directement comme entrée de vos classifieurs à vous

Exemples de sujets

- Classer des poésies par mouvement littéraire (classicisme, pléiade et romantisme)
- Retrouver le type de plat (entrée, plat ou dessert) de textes de recettes de cuisine
- Déterminer la polarité (favorable ou défavorable) de commentaires clients sur un site marchand
- Classer des textes de chansons par auteur
- Identifier les textes issus de sites classés comme sectaires par la Miviludes
- Retrouver la catégorie (sport ou cinéma) d'articles du journal en ligne *Le Monde*

Un exemple de dossier complet est [disponible](#) pour vous donner une idée de ce qui est attendu. **Il ne s'agit que d'un exemple et vous êtes parfaitement libre de présenter vos résultats autrement, de faire des expériences différentes**