Cours PCD – Labo 6 : Extraction d'attributs à partir de textes en vue de la classification

Objectifs

- Étudier les bénéfices apportés par différents attributs textuels pour la classification.
- Réaliser automatiquement une recherche de paramètres optimaux.
- Trouver le meilleur score pour l'identification de dépêches parlant de céréales (en anglais, 'grain') dans le sous-ensemble de test du corpus Reuters-21578.

Prise en main du corpus Reuters-21578

Le corpus Reuters-21578 contient un total d'environ 20'000 dépêches de l'agence Reuters des années 1990, réparties dans un certain nombre de classes, par exemple 'grain', 'wheat', 'crude' ou 'money-fx'. Le corpus a été souvent utilisé pour comparer des méthodes de classification, et nous utiliserons ici une partie nommée « ApteMod » avec 7769 dépêches pour l'entraînement et 3019 pour le test. Chaque dépêche peut appartenir à une ou plusieurs catégories, mais dans ce labo, **nous étudierons seulement la classe 'grain'**.

Il existe plusieurs façons de se procurer le corpus et de l'importer en Python. Veuillez utiliser la 3e.

1. http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html (fichiers originaux)
2. Utiliser un modèle de code fourni par Scikit-learn
3. Utiliser la librairie NLTK (à installer préalablement) qui fournit une version du corpus prête à l'emploi. Voici un exemple inspiré de la documentation :

```
from nltk.corpus import reuters
print(reuters.readme())
print(reuters.fileids()[150:155])
print('training files : ', len([fid for fid in reuters.fileids() if fid[:5] == 'train']))
print('testing files  : ', len([fid for fid in reuters.fileids() if fid[:4] == 'test']))
print('total files    : ', len(reuters.fileids()))
print(reuters.words('test/15120')[:200])
print(reuters.categories('test/15120'))
```

Tâches et questions

1. Importer le corpus Reuters-21578 dans un *notebook* Jupyter grâce à NLTK. Bien identifier les documents d'entraînement (*train*) et ceux d'évaluation (*test*). N'utiliser que les premiers pour la recherche des paramètres optimaux.

2. Transformer les données (*train* et *test*) en *DataFrames* de *pandas* avec deux colonnes : le texte complet de la dépêche (réassemblé à partir des *tokens* de NLTK avec `' '.join()`) et la catégorie 'grain' ou non-'grain', codée comme 0 ou 1 par exemple.

3. Adapter le code fourni par Scikit-learn pour la classification de 20 Newsgroups (un notebook est également disponible) afin de trouver les *meilleurs hyperparamètres* pour la classification des dépêches de Reuters-21578 en 'grain' et non-'grain'.

    - Procéder par validation croisée (`GridSearchCV`) sur les données d'entraînement et à la fin indiquer clairement dans votre notebook les hyperparamètres optimaux trouvés.
    - Ne pas hésiter pas à explorer d'autres d'options pour `CountVectorizer` et `TfidfTransformer` que celles fournies dans l'exemple de Scikit-learn.
    - Utiliser seulement le classifieur `SGDClassifier` fourni dans l'exemple, qui correspond à un modèle SVM linéaire.

4. **Quel est le score sur les données de test de la meilleure configuration que vous avez obtenue ci-dessus ?** Dans la mesure où il s'agit d'une tâche de classification binaire, il suffit de donner le pourcentage de dépêches correctement classées (en 'grain' et en non-'grain'). Comment se compare votre score avec le score « majoritaire », obtenu si on classe toutes les dépêches comme non-'grain' ?