

## 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 diverses façons de se procurer le corpus et de l’importer en Python. **Veillez utiliser la 2<sup>nde</sup>** :

1. Récupérer les [fichiers originaux](#) puis les traiter avec Scikit-learn ([exemple](#) 20 Newsgroups).
2. Utiliser la librairie NLTK qui fournit une version du corpus prête à l’emploi.

Voici quelques commandes à titre d’exemple (voir 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’, codées respectivement comme 1 et 0 (pour pouvoir appliquer facilement ‘f1-score’ en mode par défaut binaire avec ‘1’ comme classe positive).

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', sans utiliser les données de test.
  - Procéder par validation croisée (GridSearchCV) sur les données d'entraînement et à la fin indiquer clairement dans votre notebook les hyperparamètres optimaux trouvés (faites une conclusion sous forme de tableau). Utiliser comme métrique le score f1 de la classe 'grain'.
  - Veuillez explorer d'autres options pour CountVectorizer et TfidfTransformer que celles fournies dans l'exemple de Scikit-learn.
  - Comparer les résultats obtenus avec deux classifieurs : le modèle bayésien ComplementNB fourni dans l'exemple de Scikit-learn, et aussi un modèle SGDClassifier(loss='hinge'), qui correspond à un modèle SVM linéaire.
4. **Quel est le score f1 de votre meilleure configuration trouvée ci-dessus, cette fois-ci sur les données de test ?** Quels sont le rappel et la précision de la classe 'grain' ?