TP3 : IDL_06 Language identification

Rapport de TP réalisé par **Viet Nguyen -- 20006303**.

Exercice 1 : Créer un modèle de langue

1. Dans le fichier `squelette.py` :
   - ☑ Créez deux variables `train_files_list` et `test_files_list` contenant respectivement les fichiers de train et de test del'ensemble du corpus.
   - ☑ Combien y a-t-il de fichiers de chaque type ?

     > Nombre de fichiers : 5988 Nombre de fichiers d'entraînement : 4752 Nombre de fichiers de test : 1232

2. Dans le fichier `squelette.py` :
   - ☑ Implémentez la fonction `read_file` qui prend en argument un chemin et qui renvoie la chaîne de caractères correspondant au contenu du fichier pointé par le chemin.

3. Dans le fichier `squelette.py` :
   - ☑ Implémentez la fonction `language_wc` (`wc` pour *word count*) qui prend pour argument une liste de fichiers, et qui renvoie un dictionnaire donc les clés sont les langues `l` et dont les valeurs sont des dictionnaires associant à chaque mot rencontré dans le corpus d'apprentissage de la langue `l` son effectif.

4. Dans le fichier `squelette.py` :
   - ☑ Implémentez la fonction `create_lmodels_wc` qui prend en paramètre le dictionnaire d'effectifs par langue créé précédemment, et qui renvoie un dictionnaire qui à chaque langue `l` associe la liste des 10 mots les plus fréquents dans cette langue.
   - ☑ Stockez le dictionnaire renvoyé dans une variable `lmodels_wc`.

Le résultats sont stockés dans le fichier `language_models.json`. Voici un extrait du résultat:


```
{
  "et": [
    "ja",
    "on",
    "Euroopa",
    "et",
    "ning",
    "ELi",
    "the",
    "and",
    "mis",
    "of"
  ],
```

```
  ...
}
```

Exercice 2 : Modèle de langue

1. Dans le fichier `squelette.py`, avec la fonction `get_most_frequent_words`:
    - ☑ Commencez par calculer les 10 mots les plus fréquents du texte et stockez-les dans une variable `most_frequent_test`.
2. Dans le fichier `squelette.py`, avec la fonction `lpredict`:
    - ☑ Calculez l'intersection entre cette liste de mots et les 10 mots les plus fréquents associés à chacun des modèles de langue existant (`lg`).

Le résultat est stocké dans le fichier `test_results.txt`. Voici un extrait du résultat:

```
File: corpus_multi/et/test/2009-05-08_celex_IP-09-723.et.html, Actual
Language: et, Predicted Language: et
File: corpus_multi/et/test/2009-02-20_celex_IP-09-300.et.html, Actual
Language: et, Predicted Language: et
```

Exercice 3 : Evaluation la méthode

- ☑ Stockez dans une variable `pred_results` (pour prediction results) la liste des couples `[language, predicted_language]` pour chacun des fichiers de test.
- ☑ Créez une fonction `evaluate_wc_model` qui prend en paramètre une liste de liste de résultats `pred_results` et qui renvoie un dictionnaire `results_dic` qui associe à chaque langue le nombre d'identification correctes et incorrectes réalisées par le modèle.

Voici un extrait du résultat:

```
Précision Globale: 97.24%
Résultats par Langue: {'et': {'TP': 55, 'FP': 3, 'FN': 1}, 'hu': {'TP': 55,
'FP': 0, 'FN': 1}, 'ro': {'TP': 55, 'FP': 3, 'FN': 1}, 'cs': {'TP': 54,
'FP': 4, 'FN': 2}, 'sl': {'TP': 54, 'FP': 0, 'FN': 2}, 'lt': {'TP': 54,
'FP': 0, 'FN': 2}, 'en': {'TP': 56, 'FP': 17, 'FN': 0}, 'sk': {'TP': 51,
'FP': 0, 'FN': 5}, 'it': {'TP': 55, 'FP': 0, 'FN': 1}, 'el': {'TP': 55,
'FP': 0, 'FN': 1}, 'fi': {'TP': 52, 'FP': 0, 'FN': 4}, 'da': {'TP': 55,
'FP': 0, 'FN': 1}, 'mt': {'TP': 55, 'FP': 0, 'FN': 1}, 'lv': {'TP': 54,
'FP': 0, 'FN': 2}, 'nl': {'TP': 55, 'FP': 0, 'FN': 1}, 'sv': {'TP': 55,
'FP': 0, 'FN': 1}, 'bg': {'TP': 55, 'FP': 0, 'FN': 1}, 'fr': {'TP': 55,
'FP': 0, 'FN': 1}, 'de': {'TP': 55, 'FP': 0, 'FN': 1}, 'es': {'TP': 55,
'FP': 0, 'FN': 1}, 'pl': {'TP': 53, 'FP': 0, 'FN': 3}, 'pt': {'TP': 55,
'FP': 0, 'FN': 1}}
```

Exercice 4 : En caractères

1. Dans le fichier `squelette.py`
    - [x] Reprenez le code précédent en utilisant non plus les mots mais les n-grams de caractères : testez avec n allant de 1 à 4.
2. Dans le fichier `squelette.py`
    - [x] Proposez une comparaison entre les différentes méthodes testées.

J'ai testé avec n allant de 1 à 4. Voici un extrait du résultat:

```
language_models_n1.json creée
For n=1, Overall Accuracy: 38.56%
Results by Language: {'et': {'TP': 49, 'FP': 76, 'FN': 7}, 'hu': {'TP': 38,
'FP': 24, 'FN': 18}, 'ro': {'TP': 52, 'FP': 81, 'FN': 4}, 'cs': {'TP': 32,
'FP': 123, 'FN': 24}, 'sl': {'TP': 0, 'FP': 0, 'FN': 56}, 'lt': {'TP': 35,
'FP': 90, 'FN': 21}, 'en': {'TP': 31, 'FP': 185, 'FN': 25}, 'sk': {'TP': 0,
'FP': 0, 'FN': 56}, 'it': {'TP': 0, 'FP': 0, 'FN': 56}, 'el': {'TP': 55,
'FP': 0, 'FN': 1}, 'fi': {'TP': 0, 'FP': 0, 'FN': 56}, 'da': {'TP': 22,
'FP': 121, 'FN': 34}, 'mt': {'TP': 0, 'FP': 0, 'FN': 56}, 'lv': {'TP': 0,
'FP': 0, 'FN': 56}, 'nl': {'TP': 0, 'FP': 0, 'FN': 56}, 'sv': {'TP': 0,
'FP': 0, 'FN': 56}, 'bg': {'TP': 55, 'FP': 0, 'FN': 1}, 'fr': {'TP': 0,
'FP': 0, 'FN': 56}, 'de': {'TP': 28, 'FP': 0, 'FN': 28}, 'es': {'TP': 28,
'FP': 0, 'FN': 28}, 'pl': {'TP': 50, 'FP': 0, 'FN': 6}, 'pt': {'TP': 0,
'FP': 0, 'FN': 56}}
language_models_n2.json creée
For n=2, Overall Accuracy: 90.99%
Results by Language: {'et': {'TP': 54, 'FP': 3, 'FN': 2}, 'hu': {'TP': 54,
'FP': 0, 'FN': 2}, 'ro': {'TP': 51, 'FP': 8, 'FN': 5}, 'cs': {'TP': 51,
'FP': 17, 'FN': 5}, 'sl': {'TP': 51, 'FP': 10, 'FN': 5}, 'lt': {'TP': 53,
'FP': 0, 'FN': 3}, 'en': {'TP': 55, 'FP': 15, 'FN': 1}, 'sk': {'TP': 33,
'FP': 0, 'FN': 23}, 'it': {'TP': 47, 'FP': 0, 'FN': 9}, 'el': {'TP': 55,
'FP': 0, 'FN': 1}, 'fi': {'TP': 53, 'FP': 0, 'FN': 3}, 'da': {'TP': 53,
'FP': 23, 'FN': 3}, 'mt': {'TP': 55, 'FP': 0, 'FN': 1}, 'lv': {'TP': 55,
'FP': 0, 'FN': 1}, 'nl': {'TP': 50, 'FP': 4, 'FN': 6}, 'sv': {'TP': 39,
'FP': 0, 'FN': 17}, 'bg': {'TP': 55, 'FP': 0, 'FN': 1}, 'fr': {'TP': 54,
'FP': 0, 'FN': 2}, 'de': {'TP': 49, 'FP': 0, 'FN': 7}, 'es': {'TP': 48,
'FP': 1, 'FN': 8}, 'pl': {'TP': 53, 'FP': 0, 'FN': 3}, 'pt': {'TP': 53,
'FP': 0, 'FN': 3}}
language_models_n3.json creée
For n=3, Overall Accuracy: 90.26%
Results by Language: {'et': {'TP': 48, 'FP': 7, 'FN': 8}, 'hu': {'TP': 54,
'FP': 3, 'FN': 2}, 'ro': {'TP': 54, 'FP': 7, 'FN': 2}, 'cs': {'TP': 50,
'FP': 25, 'FN': 6}, 'sl': {'TP': 52, 'FP': 1, 'FN': 4}, 'lt': {'TP': 50,
'FP': 10, 'FN': 6}, 'en': {'TP': 56, 'FP': 16, 'FN': 0}, 'sk': {'TP': 30,
'FP': 4, 'FN': 26}, 'it': {'TP': 53, 'FP': 0, 'FN': 3}, 'el': {'TP': 53,
'FP': 0, 'FN': 3}, 'fi': {'TP': 52, 'FP': 0, 'FN': 4}, 'da': {'TP': 55,
'FP': 13, 'FN': 1}, 'mt': {'TP': 55, 'FP': 0, 'FN': 1}, 'lv': {'TP': 41,
'FP': 0, 'FN': 15}, 'nl': {'TP': 54, 'FP': 0, 'FN': 2}, 'sv': {'TP': 44,
'FP': 0, 'FN': 12}, 'bg': {'TP': 55, 'FP': 0, 'FN': 1}, 'fr': {'TP': 54,
'FP': 4, 'FN': 2}, 'de': {'TP': 52, 'FP': 0, 'FN': 4}, 'es': {'TP': 54,
```

```
'FP': 3, 'FN': 2}, 'pl': {'TP': 46, 'FP': 0, 'FN': 10}, 'pt': {'TP': 50,
'FP': 0, 'FN': 6}}
language_models_n4.json creée
For n=4, Overall Accuracy: 82.95%
Results by Language: {'et': {'TP': 52, 'FP': 96, 'FN': 4}, 'hu': {'TP': 48,
'FP': 2, 'FN': 8}, 'ro': {'TP': 54, 'FP': 13, 'FN': 2}, 'cs': {'TP': 44,
'FP': 30, 'FN': 12}, 'sl': {'TP': 42, 'FP': 8, 'FN': 14}, 'lt': {'TP': 38,
'FP': 17, 'FN': 18}, 'en': {'TP': 55, 'FP': 20, 'FN': 1}, 'sk': {'TP': 29,
'FP': 0, 'FN': 27}, 'it': {'TP': 54, 'FP': 1, 'FN': 2}, 'el': {'TP': 45,
'FP': 0, 'FN': 11}, 'fi': {'TP': 28, 'FP': 0, 'FN': 28}, 'da': {'TP': 51,
'FP': 0, 'FN': 5}, 'mt': {'TP': 54, 'FP': 0, 'FN': 2}, 'lv': {'TP': 36,
'FP': 0, 'FN': 20}, 'nl': {'TP': 53, 'FP': 0, 'FN': 3}, 'sv': {'TP': 50,
'FP': 0, 'FN': 6}, 'bg': {'TP': 41, 'FP': 0, 'FN': 15}, 'fr': {'TP': 51,
'FP': 0, 'FN': 5}, 'de': {'TP': 53, 'FP': 0, 'FN': 3}, 'es': {'TP': 53,
'FP': 3, 'FN': 3}, 'pl': {'TP': 45, 'FP': 0, 'FN': 11}, 'pt': {'TP': 46,
'FP': 0, 'FN': 10}}
```

=> La précision globale est la plus élevée pour n=2, suivi par n=3, n=4, et n=1. La précision globale est la plus faible pour n=1. Cela est dû au fait que les n-grams de caractères sont plus informatifs que les mots. Cependant, la précision globale pour n=2 est seulement légèrement plus élevée que pour n=3. Cela signifie que les n-grams de caractères de longueur 2 sont suffisamment informatifs pour identifier la langue.

Exercice 5 : Apprentissage

Dans le fichier `exercice_5.py` :

- [x] Préparation des données
- [x] Vectorisation données
- [x] Classification et évaluation
- [x] Faire jouer les paramètres