Rapport Programmes de Modèles Linguistiques

Ces programmes ont pour objectif d'analyser de grands corpus, de factoriser les traitements, et d'évaluer les performances du modèle développé.

Le dossier est structuré en plusieurs parties : une partie dédiée au corpus et une autre aux programmes. Dans la partie "programmes", nous avons les exercices 1, 2, et 3. Quant à la partie "corpus" fournie par le professeur, elle contient des documents en 21 langues différentes, chacun étant divisé en un corpus d'apprentissage et un corpus de test.

L'objectif du TD4 est de développer un modèle capable d'analyser les mots de différentes langues pour en tirer des caractéristiques. Ce modèle pourra ensuite analyser de nouveaux textes et essayer de prédire la langue à partir des mots présents, en les comparant avec ceux des modèles de langue précédemment établis. Le terme « apprendre » est parfois utilisé pour ce processus, mais je préfère éviter ce terme anthropomorphique qui prête à la machine une capacité qu'elle n'a pas.

Fonctionnement du modèle

Le modèle fonctionne comme suit :
1) Il parcourt le corpus d'apprentissage et construit un dictionnaire répertoriant les mots par langue. La langue est identifiée dans le chemin d'accès du fichier à un index défini.
2) Le programme calcule ensuite les mots les plus fréquents pour chaque langue (les 10 mots les plus communs).
3) Enfin, il enregistre ces mots dans un fichier JSON, ce qui permet de structurer les données et de les conserver pour une utilisation future.

Fonctionnement du programme de test

Dans un second temps, un autre programme :
1) Charge le fichier JSON produit par le premier programme.
2) Parcourt à nouveau le corpus pour aller chercher les fichiers de test.
3) Pour chaque fichier de test, le programme identifie la langue réelle en l'extrayant du chemin du fichier.
4) Il crée un dictionnaire local avec les 10 mots les plus fréquents du fichier de test.

Le programme utilise ensuite ce dictionnaire de mots fréquents pour comparer le fichier de test avec chaque langue du modèle stocké dans le JSON. Il calcule l'intersection des mots fréquents entre le fichier de test et chaque langue du modèle pour identifier la langue qui partage le plus de mots communs avec le fichier de test. La langue ayant la plus grande intersection est choisie comme prédiction.

Une fois la prédiction effectuée pour chaque fichier, le programme évalue ses performances en calculant :
- L'exactitude globale (proportion de prédictions correctes),
- Des statistiques pour chaque langue : précision (bruit), rappel (silence), et F1-mesure (moyenne de la précision et du rappel).

Ces résultats sont ensuite affichés, sauvegardés dans un fichier JSON pour permettre des analyses ultérieures, et constituent l'évaluation finale du modèle de prédiction linguistique.

Résultats obtenus

Exercice 3 : Pour l’exercice 3, nous avons obtenu une performance globale d’environ 90 %, avec des scores (rappel, précision, mesure F1) oscillant entre 90 % et 99 %. Cela indique que, pour les mots entiers (tokens), ce programme fonctionne très bien.

Exercice 4 : Pour l’exercice 4, la performance globale est tombée autour de 2 %, ce qui est difficile à expliquer. De nombreuses langues affichent des scores de rappel, précision et mesure F1 à zéro. Cependant, en faisant varier le nombre de n-grammes, nous obtenons des résultats différents : avec moins d’ngrams, le score augmente jusqu’à 35-40 %, tandis qu’avec davantage (au-delà de 7 ou 8), les scores deviennent de plus en plus faibles. On pourrait expliquer cette baisse par le fait que, plus le nombre d’ngrams est faible, moins les langues sont distinctes les unes des autres. Inversement, en réduisant le nombre d’ngrams, on augmente le rappel, ce qui influence positivement le score de réussite global. Néanmoins, il est difficile d’interpréter ces résultats sans les comparer avec ceux de mes camarades. Il est possible qu’il y ait une erreur dans mon programme ou dans ma logique. J’ai ajouté la fonction n-gram à la partie 4 assez rapidement, donc des incohérences ont peut-être échappé à mon attention.

Difficultés rencontrées

- Établir la structure du modèle. Nous disposions de peu de ressources pour l’exercice 3, donc nous avons organisé plusieurs réunions pour définir notre approche.
- Manipuler plusieurs dictionnaires et variables demande une bonne organisation et des noms explicites pour les données et les variables.
- J’ai obtenu des résultats étonnants pour l’exercice 4 en utilisant les n-grammes.