M2 Data Science

Renaissance de langues écrites

Par
LIAO Puwei
ZHANG Shurong

Tuteur Chriostophe Ambroise

Décembre 2021

Table des matières

1 Introduction

Dans le traitement du langage naturel, la reconnaissance linguistique ou la spéculation linguistique consiste à déterminer le langage naturel utilisé pour un contenu donné. Les méthodes de calcul du problème sont considérées comme un cas particulier de classification des textes et sont résolues par diverses méthodes statistiques.

Dans ce projet, nous allons d'abord construire un dataset qui contient 15 textes français et 15 lyrics anglais, et on donne les étiquettes à ces textes, 1 pour les textes anglais et -1 pour les textes français. Notre but de ce projet est de fabriquer les classifieurs (Naive bayes et Markovien) à partir des certaines textes du dataset comme training data pour distinguer la langue des textes et puis faire le test avec les textes restes. Finalement, on va évaluer leur performance par la méthode de validation croisée.

De plus, avec notre modèle Markovien, on peut créer un court texte d'au plus 1000 caractères enchaînant de manière aléatoire des phrase en français et en anglais tirées des textes initiaux. Nous pouvons trouver le passage entre français et anglais du texte fabriqué en utilisant l'algorithme Viterbi.

2 Base de textes

2.1 Création de Matrice f et X

Afin de créer le dataset D, nous avons d'abord mis les textes dans un vecteur "langue", puis nous avons crée un vecteur "class" pour les étiquettes des textes (1 pour anglais, -1 pour français). On combine ces deux vecteurs dans un data.frame.

Ensuite, nous avons écrit un programme de nettoyage du texte qui nous permet d'enlever les symboles qu'on n'aura pas besoin :

— - ; : ? . . . - ( ) , ' ' - 0 1 2 3 4 5 6 7 8 9 .!' etc

Et puis, on fabrique une matrice de fréquence f de taille $30 * 43$. Pour chaque ligne de f, on a le nombre de tous les lettres dans chaque texte. Les colonnes sont décrits par les lettres suivants :

"a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z"

'à','â','æ','ç','é','è','ê','ë','î','ï','ô','œ','ù','û','ü','ÿ'

Où il y a total 42 lettres à compter. Et la dernière colonne est le nombre d'espace dans chaque texte, ce qui est important pour distinguer les mots.

Après ça, nous avons calculé la fréquence de chaque lettre dans les différents textes et nous avons les normalisé par la fonction $log(1 + x), \forall x \in \mathbb{R}$. Avec cela, nous pouvons obtenir notre matrice X.

2.2 Histogrammes des 2 langues et commentaires

Enfin, nous traçons l'histogramme pour la log de fréquence des lettres dans la classe française et dans la classe anglaise en sommant tous les nombres de fois apparaissent pour chaque lettre de chaque langue,

et puis on le normalise.

Et voici les 2 histogrammes de log(1+fréquence des lettres des 2 langues correspondantes) où E indique l'espace :

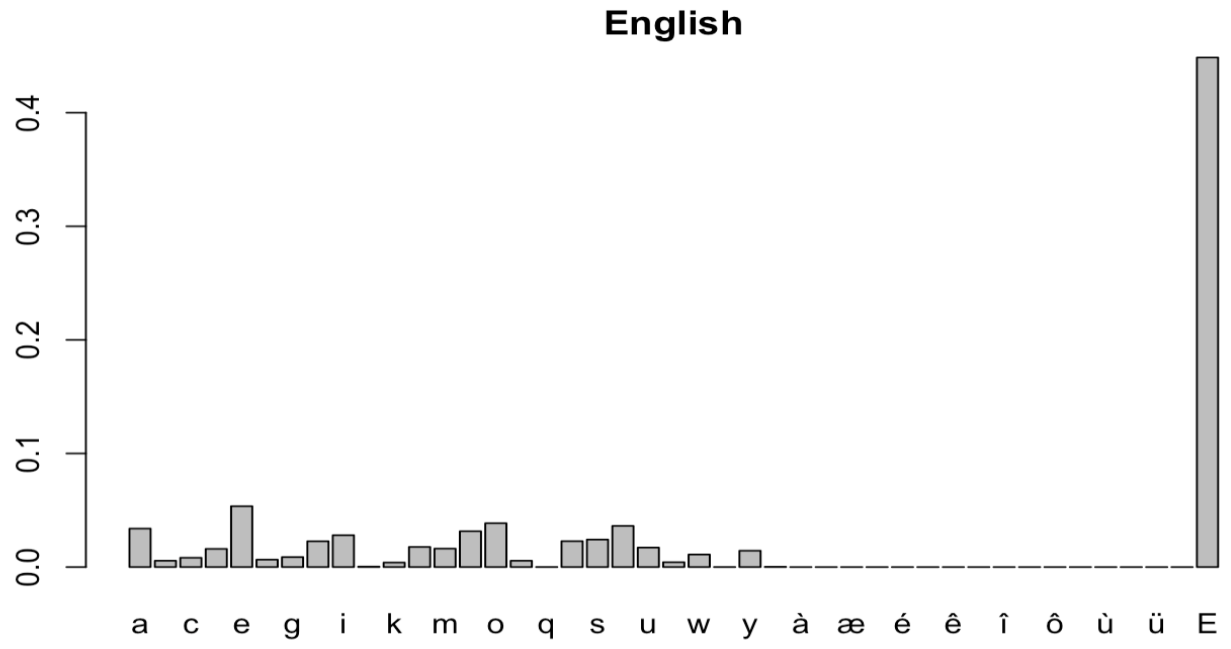

FIGURE 1 – English

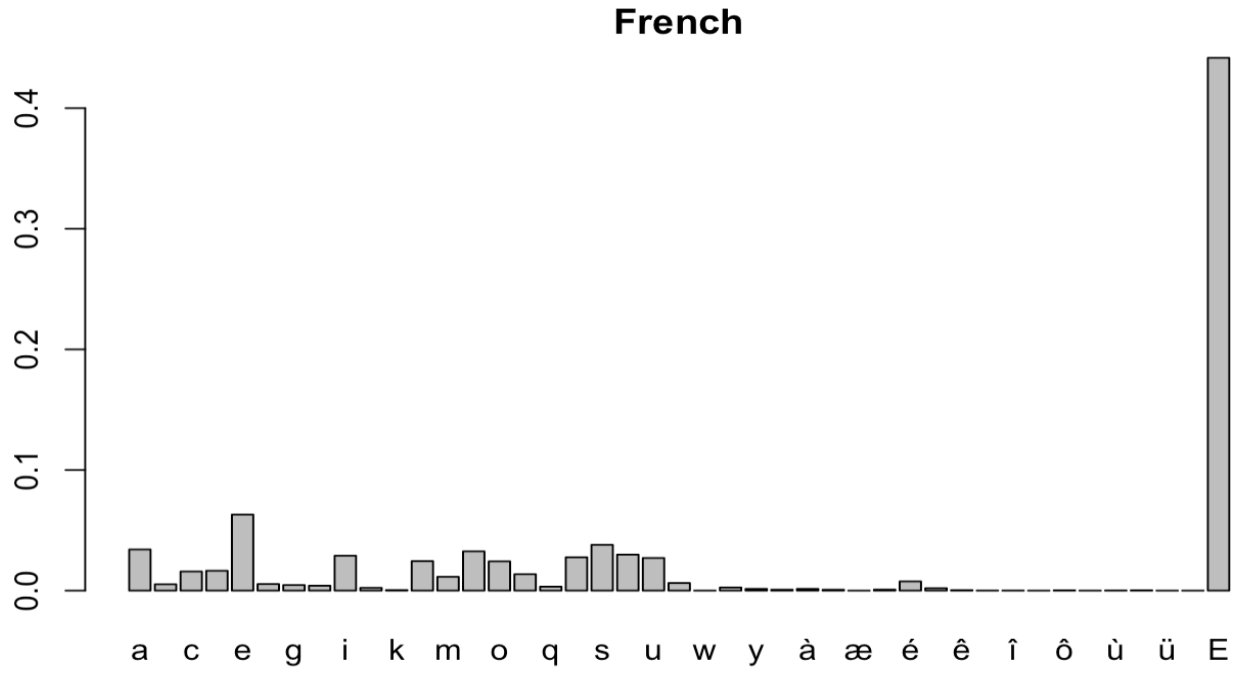

FIGURE 2 – French

D'après ces 2 histogrammes, visiblement que pour les lettres avec l'accent, ils n'ont aucune chance

d'apparaître dans un texte anglais, c'est pourquoi la log des fréquences de la colonne 'à' à la colonne 'ÿ' pour anglais est vide (sauf colonne 43 :espace). De plus, de 'a' à 'e' , les log de fréquences ressemblent beaucoup pour anglais et français, mais à partir du lettre 'f', les deux langues se diffèrent. Et finalement, on constate que la lettre 'e' s'apparaît le plus dans l'anglais et aussi dans le français, c'est un point commun intéressant pour ces 2 langues.

3 Classifieur de Bayes Naif

Nous allons étudier le classifieur de Bayes naif qui est un classifieur classique nous permet de classifier les textes en anglais ou en français. Les moyennes et les variances sont deux paramètres très importants pour ce classifieur. On va donc les estimer dans la première section.

3.1 Estimation des moyennes et variances de chaque classe

Pour estimer l'espérance d'anglais et l'espérance de français, on a calculé la moyenne de chaque lettre dans les textes anglais et les textes français qui nous renvoie ces résultats dans un vecteur $mean_{Fren}$ et un vecteur $mean_{Eng}$.

Pour estimer la variance d'anglais et la variance de français, on a calculé la variance de chaque lettre dans les textes anglais et les textes français qui nous renvoie ces résultats dans un vecteur var_{Fren} et un vecteur var_{Eng}.

3.2 Création classifieur de Bayes Naif

Nous allons créer un classifieur de Bayes Naif à la main pour faire une prédiction du texte si c'est en français ou en anglais. Pour faire un test de ce classifier à la main, nous pouvons choisir X textes qui sont choisis aléatoirement comme les données d'entraînement, et les 30-X textes restes comme les données de test. Ensuite, on rappelle que l'algorithme se base sur le théorème de Bayes qui nous dit que :

$\mathbb{P}(A|B) = \mathbb{P}(\mathrm{B}|A)$*P(A)/P(B) $\propto \mathbb{P}(\mathrm{B}|A)$*P(A)

Où A : la langue du texte ; B : le texte

On a ainsi que $\mathbb{P}(\mathrm{B}|A) = \prod_{i\in\mathbb{E}} \mathbb{P}(\mathbb{C}_i|A)$

Où $\mathbb{C}$: les vocabulaires ; E : l'espace des vocabulaires.

Ensuite, nous pouvons aussi calculer la probabilité initiale pour chaque langue.

$\mathbb{P}_{initial}(French)$ = le nombre de textes français / le nombre de textes total

$\mathbb{P}_{initial}(English)$ = le nombre de textes anglais / le nombre de textes total

On a alors la probabilité de prédiction pour l'anglais et pour le français :

$\mathbb{P}(Fren|B) = \mathbb{P}_{initial}(French) * \prod_{i\in\mathbb{E}} \mathbb{P}(\mathbb{C}_i|French)$

$\mathbb{P}(Eng|B) = \mathbb{P}_{initial}(English) * \prod_{i\in\mathbb{E}} \mathbb{P}(\mathbb{C}_i|English)$

Finalement, on compare ces deux dernières probabilités et puis on prend la plus grande probabilité pour notre prédiction de langue.

Dans la section suivante, on fera une validation croisée plus sérieuse pour vérifier la performance de ce classifier.

3.3 Validation croisée

On évalue la performance à l'aide de la méthode de la validation croisée(K-fold) avec $K = 10$, et voici nos résultats obtenus :

```
le nombre de classification correste est: 4 / 4
le nombre de classification correste est: 4 / 4
le nombre de classification correste est: 4 / 4
le nombre de classification correste est: 3 / 3
le nombre de classification correste est: 2 / 2
le nombre de classification correste est: 3 / 3
le nombre de classification correste est: 2 / 2
le nombre de classification correste est: 2 / 2
le nombre de classification correste est: 3 / 3
le nombre de classification correste est: 3 / 3
```

FIGURE 3 – Résultat de la validation croisée Naif Bayes classifier

On est donc très satisfait avec notre classifieur Naif Bayesian qui a un accuarcy 100 pourcent. Maintenant, on peut donc utiliser ce classifieur pour faire la prédiction pour un texte si c'est en français ou en anglais.

4 Classifieur Markovien

Nous allons étudier le classifieur Markovien qui est un autre classifieur classique nous permet de classifier les textes en anglais ou en français. La matrice de transition A_k et la probabilité d'états initials π_k sur l'ensemble des symboles sont deux paramètres très importants pour ce classifieur. On va donc les estimer dans la première section.

4.1 Estimation de A_k et π_k

Pour le classifieur Markovien, on va d'abord estimer le paramètre Π_k (loi initial) pour la classe anglaise et la classe française. Pour ce faire, nous allons d'abord mettre l'espace au début de chaque texte puis nous allons calculer la probabilité de chaque lettre qui suit l'espace. C'est à dire que la probabilité pour

chaque lettre qui est le premier lettre d'un mots. On peut aussi estimer l'autre paramètre A_k (matrice de transition) à l'aide de la programme du bigram. On aura une loi initiale et une matrice de transition pour chacune classe.

On remarque qu'on a remplacé les espaces du texte par 0 dans notre programme puisque cela nous permet de faire le calcul numérique.

4.2 Théorie de classifieur Markovien

Ensuite, nous allons créer un classifieur Markovien pour distinguer la langue du texte. Pour ce faire, on fait un test d'hypothèse,

$$H_0 \ : \ le\ texte\ est\ en\ français$$

$$Contre$$

$$H_1 \ : \ le\ texte\ est\ en\ anglais$$

Pour le modèle de statistique, on a un texte comme une réalisation d'une chaîne de Markov $X = (X_0, X_1, X_2.....X_n)$ de loi initial π et de matrice de transition A. Et on a le choix de statistique du test T pour l'empirique de loi initial et l'empirique de la matrice de transition. Pour le test de statistique T, on a :

$$H_0 : T = \begin{pmatrix} \pi \\ A \end{pmatrix} = \begin{pmatrix} \pi_{french} + Y_1 \\ A_{french} + Z_1 \end{pmatrix} \in \mathbb{R}^n \times \mathbb{M}_{n*n} \tag{1}$$

$$H_1 : T = \begin{pmatrix} \pi \\ A \end{pmatrix} = \begin{pmatrix} \pi_{english} + Y_2 \\ A_{english} + Z_2 \end{pmatrix} \in \mathbb{R}^n \times \mathbb{M}_{n*n} \tag{2}$$

$$Y_1 \sim \mathbb{N}(0, \sigma_1^2)\ avec\ \sigma_1^2\ est\ la\ variance\ empiriques\ de\ \pi_{french}$$

$$Z_1 \sim \mathbb{N}(0, \Sigma_1^2)\ avec\ \Sigma_1^2\ est\ la\ variance\ empiriques\ de\ A_{french}$$

$$Y_2 \sim \mathbb{N}(0, \sigma_2^2)\ avec\ \sigma_2^2\ est\ la\ variance\ empiriques\ de\ \pi_{english}$$

$$Z_2 \sim \mathbb{N}(0, \Sigma_2^2)\ avec\ \Sigma_2^2\ est\ la\ variance\ empiriques\ de\ A_{english}$$

On a alors

$$\mathbb{E}(T) = \begin{cases} \pi_{french} \in \mathbb{R}^n \\ A_{french} \in \mathbb{M}_{n*n} \end{cases} \quad ; \quad \mathbb{E}(T) = \begin{cases} \pi_{english} \in \mathbb{R}^n \\ A_{english} \in \mathbb{M}_{n*n} \end{cases}$$

On va maintenant tracer le point (π, A) dans l'espace métrique $\mathbb{R}^n * \mathbb{M}_{n*n}$, on va regrouper ces points en deux par calculer la plus courte distance entre ces points et les deux points centrés de chaque groupe qui sont trouvés grâce aux données d'entraînements et les modèles de la classe anglaise et de la classe française.

4.3 classifieur Markovien en R

Pour aboutir ce but, le but de créer le classifieur Markovien, nous allons d'abord créer 6 fonctions en R qui sont basées sur la fonction classifieur Markovien :

— A_empirique_Eng() : Calcul de matrice de transition A empirique par training texte en anglais

— A_empirique_Fren() : Calcul de matrice de transition A empirique par training texte en français

— Pi_empirique_Eng() : Calcul de loi initial π empirique par training texte en anglais

— Pi_empirique_Fren() :Calcul de loi initial π empirique par training texte en français

— A_test() :Calcul de matrice de transition A empirique par un texte de test données

— Pi_test() :Calcul de loi initial π empirique par un texte de test données

Basées sur ces 6 fonctions, on est capable de programmer le classifier Markov, le principe est d'utiliser la partie théorique de classifieur Markov, c'est à dire, l'idée principale est de comparer la distance entre (π_{test}, A_{test}) et $(\pi_{English}, A_{English})$ et la distance entre (π_{test}, A_{test}) et $(\pi_{French}, A_{French})$ et puis on en choisit le plus proche comme la classification du texte de test, enfin on vérifie si le résultat est bon. Ainsi, on va vérifier la performance de classifier Markov dans la prochaine section par validation croisée.

4.4 Validation croisée

De même processus de la section précédente Naïve Bayes, on réalise une validation croisée avec K=10, et voici la performance que l'on a obtenu :

```
le nombre de classification correste est: 3 / 3
le nombre de classification correste est: 3 / 3
le nombre de classification correste est: 3 / 3
le nombre de classification correste est: 2 / 2
le nombre de classification correste est: 4 / 4
le nombre de classification correste est: 4 / 4
le nombre de classification correste est: 2 / 2
le nombre de classification correste est: 3 / 3
le nombre de classification correste est: 2 / 2
le nombre de classification correste est: 4 / 4
```

FIGURE 4 – Résultat de la validation croisée CM classifier

On est donc très satisfait avec notre classifieur Markov qui a un accuarcy 100 pourcent. Maintenant, on peut donc utiliser ce classifieur pour faire la prédiction pour un texte si c'est en français ou en anglais.

5 Décodage de langue par Viterbi

De plus, nous pouvons simuler la chaîne de Markov d'anglais où la chaîne de Markov Français qui permet de retourner une chaîne de caractère en anglais où en Français à partir des textes initiaux. Mais afin d'étudier le passage entre ces deux langues, on a collé tous les textes puis on a crée la matrice de transition A et la probabilité d'état initale π et puis relancer notre classifieur Markovien pour avoir un texte qui contient des lettres en français et en anglais. On a aussi crée la matrice A_{total} de taille 2*2 qui nous indique la probabilité du passage entre français et anglais, la matrice B de taille 43*2 qui nous indique la probabilité pour chaque lettre dans les textes français et anglais et on peut trouver la π_{total} avec la propriété de loi stationnaire. A l'aide de notre algorithme de Viterbi, on peut constater que c'est très rare d'avoir un changement de langue sauf au premier pas.