Machine à inventer des mots

Dans ce projet, on veut programmer un algorithme qui permettra d'inventer de nouveaux mots.

On pourra commencer par écrire un algorithme inventant des nouveaux mots aléatoires, *i.e.* avec les lettres tirées aléatoirement uniformément.

On va améliorer notre machine de la façon suivante, en deux étapes :

- Analyse de la langue française : à partir d'un dictionnaire, un algorithme va créer une table de fréquence de suites de deux lettres ; par exemple, à la case a, b, on calcule la probabilité que la lettre a soit suivie de la lettre b.

  On pensera à ajouter les probabilités d'être en début ou fin de mot.
- Création des mots : à partir de la table précédente, on pourra facilement inventer des mots, en choisissant la lettre suivante en fonction des probabilités calculées.

5 Analyse fréquentielle

Dans cette partie, on analysera le fichier `dico.txt` à récupérer sur le site. Le fichier est à enregistrer dans le même dossier que le fichier Python, et on utilisera la commande

```
dico = [line.rstrip("\n") for line in open("dico.txt")]
```

pour créer la liste `dico`, contenant la liste des mots du dictionnaire sous forme de chaîne de caractères.

Une première lecture de cette liste permettra de créer l'alphabet (l'ensemble des caractères utilisés). On rajoutera à cet alphabet les symboles < (début de mot) et > (fin de mot).

On créera alors un dictionnaire `d` :

- dont les clefs sont les lettres de l'alphabet
- dont les valeurs sont des dictionnaires, dont les clefs sont les lettres de l'alphabet, et les valeurs sont la probabilité que ces deux lettres se suivent.

Par exemple, la valeur de `d[a][b]` contiendra la probabilité qu'après un a on ait un b.

6 Création de mots

En utilisant le dictionnaire précédent, on pourra alors créer des mots en suivant la loi de probabilité créée.

7 Améliorations

On pourra limiter la taille des mots créés en éliminant ceux dont la longueur est trop importante (par exemple, les mots de plus de quinze caractère).

Pour que les mots précédents soient plus proches de la langue française, on pourra améliorer l'algorithme précédent en calculant la probabilité de suites de trois lettres ; le dictionnaire créé aura alors une profondeur de trois, `d[a][b][c]` contenant la probabilité que c suive les lettres ab.