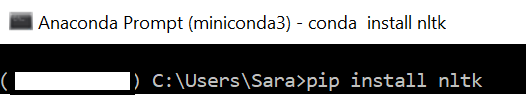
Détection de la langue :

Il existe plusieurs méthodes permettant la détection de la langue suite à l’analyse du contenu. Dans notre cas nous allons utiliser la méthode des stopwords et la méthode Tokenizer pour le premier code et pour le deuxième code nous allons utiliser la méthode TextCat, ces méthodes sont très simples et simplistes, et qui suffira pour donner un résultat satisfaisant dans la plupart des cas.

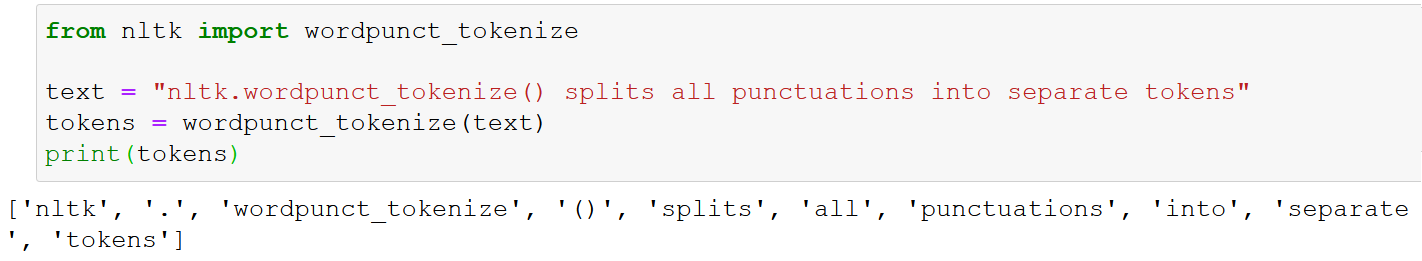
Pour commencer nous avons besoin d’installer la plateforme NLTK (Natural Language Toolkit).



Après installation, passer au code.

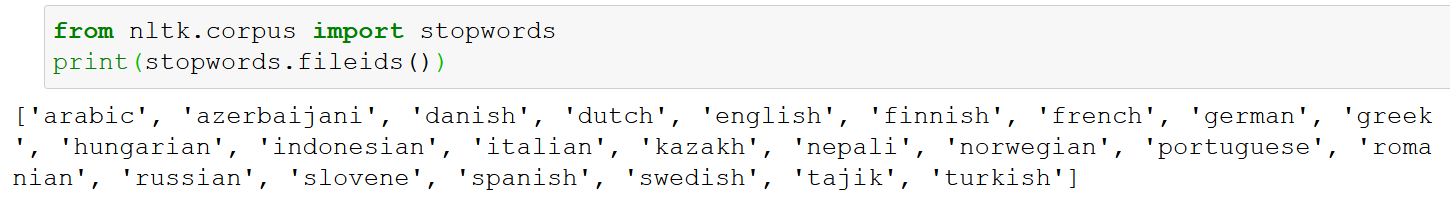
Programme 1 :

Tout d’abord, nous allons commencer par découper notre texte en mots. C’est ce qu’on appelle la tokenisation. L’idée est de prendre les phrases puis de les découper en tokens.

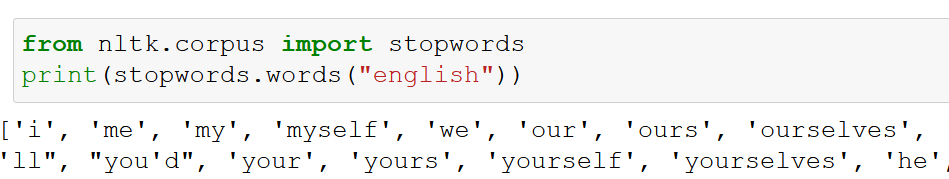


Par la suite on va aller pour découvrir les stopwords de chaque langue et les comparais avec les tokens de notre texte.

Voici la liste des langues pour lesquelles NLTK dispose d’une liste de stopwords :



Par exemple pour afficher les stopwords d’une langue il suffit de saisir le code suivant :

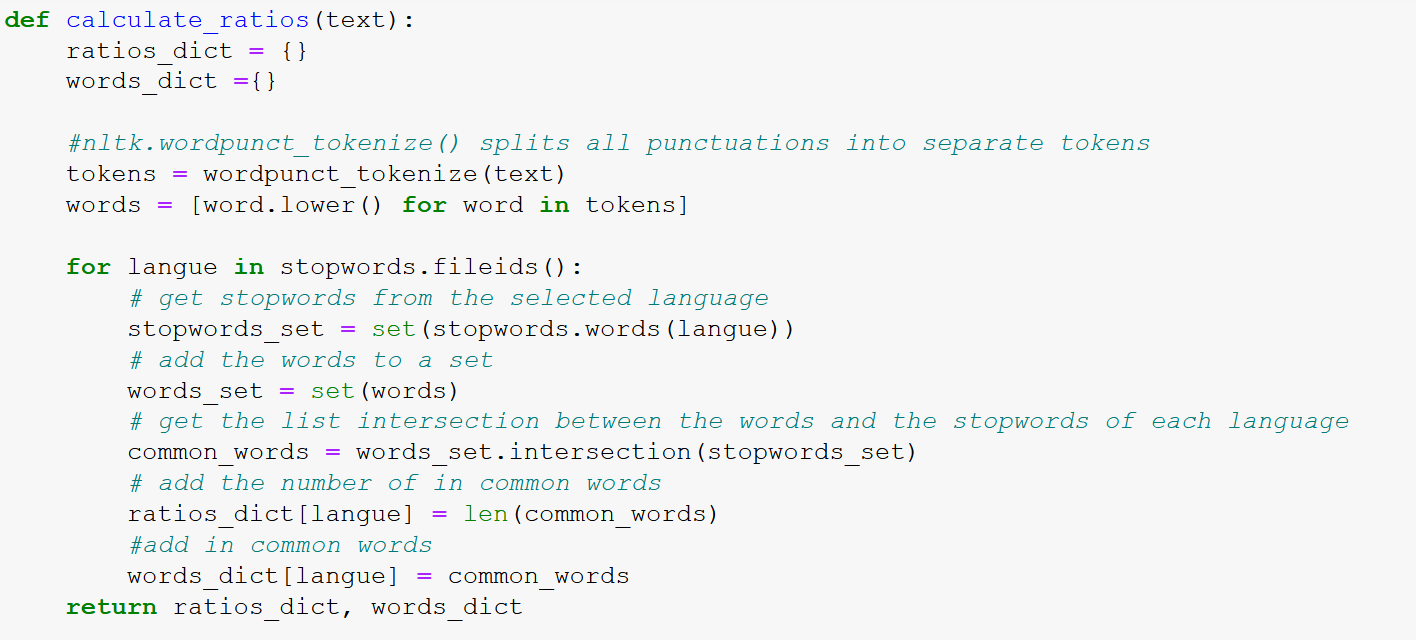


Allons au programme d’analyse :

1.

Dans le code suivant on va utiliser la fonction d’intersection pour détecter les stopwords en commun avec les tokens de notre texte.

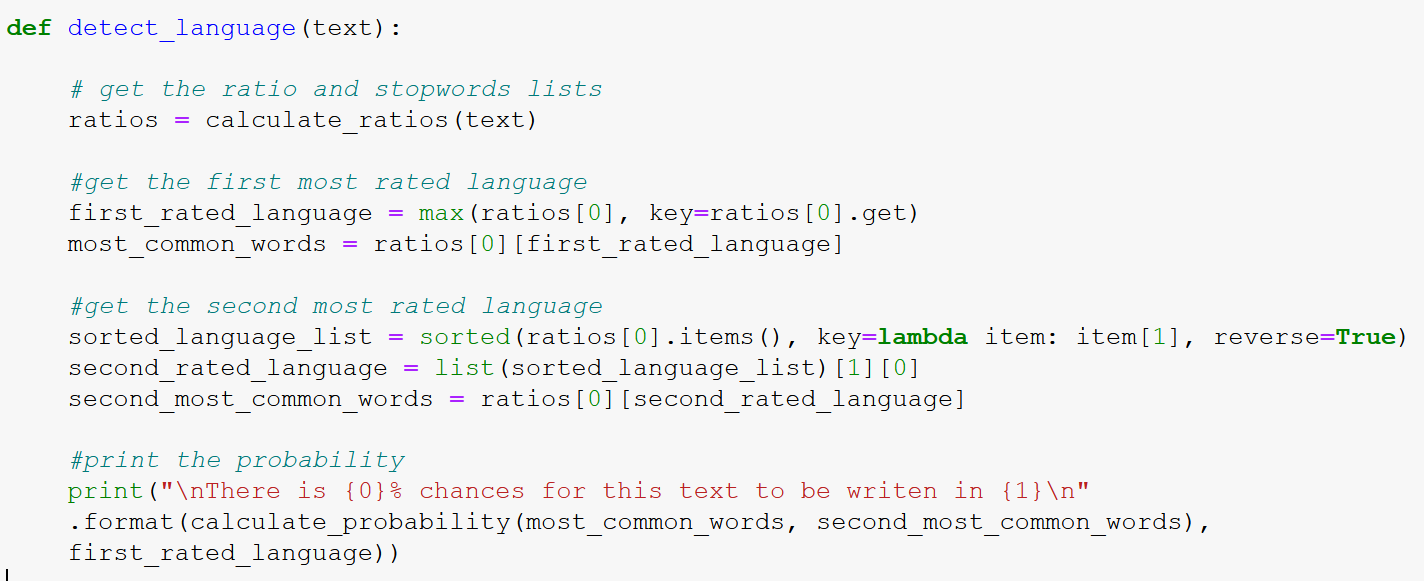
Par la suite on va calculer la longueur de chaque liste des stopwords.



Comme vous voyez dans le code la fonction retourne une liste avec les nombres des stopwords en communs et la liste des stopwords.

2.

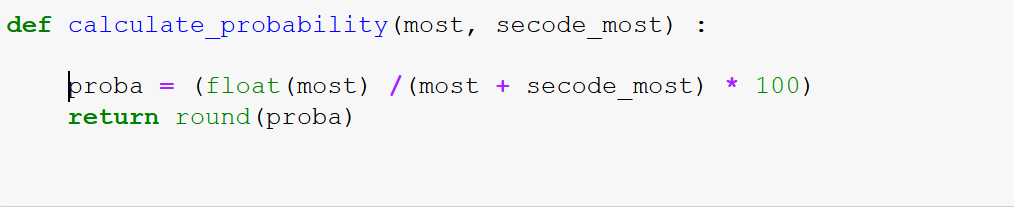
Dans le deuxième code on va extraire la liste des stopwords ayant le maximum de mots en commun avec le texte et qui sera considérer la langue la plus proche à la langue du texte.



Et pour que la réponse soit pertinente nous allons calculer la probabilité du choix de langue.

3.

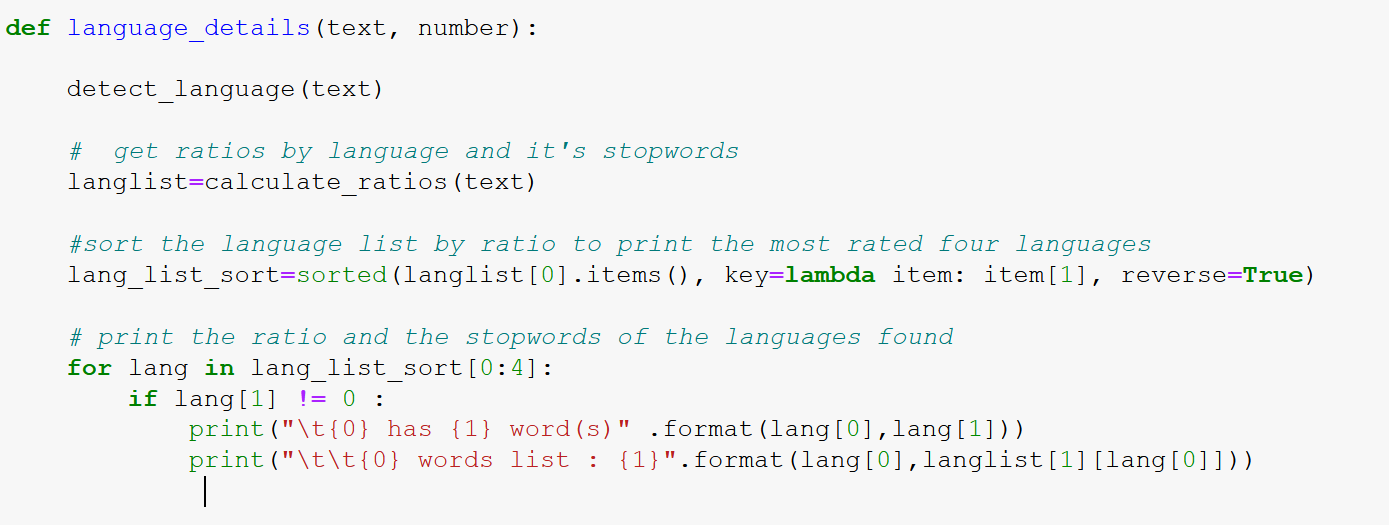
Pour le calcul de la probabilité nous utilisons le nombre de mots de la première et la deuxième langue qui ont le plus de mots en commun avec le texte, voir le code suivant :



4.

Le programme va donnée comme résultat la probabilité de la langue considérée celle dans laquelle le texte est créé aussi que le détail des stopwords trouvé dans chaque langue afin de savoir si le texte contient plusieurs langues.

Le code suivant récupère la liste des stopwords trouvés et les affiches :



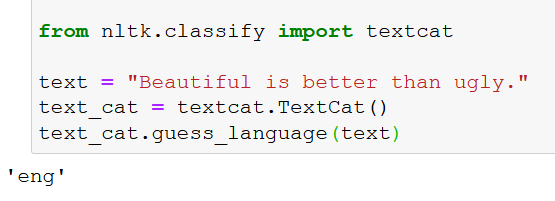
Exemple du programme :



Programme 2 :

Dans cette partie nous allons utiliser la fonction guess\_language de TextCat.

Les principaux points d'entrée de guess language prennent tous une seule chaîne en entrée et renvoient un identifiant de langue. La chaîne doit être du texte Unicode ou UTF-8.



Pour commencer il faut découper notre texte en phrases par la fonction tokenise :

La fonction tokenize divise une chaîne en sous-chaînes en se fractionnant sur la chaîne spécifiée.



Voici le code du deuxième programme :



Pour accéder au code voici le lien sur github : <https://github.com/FatimaZahrae814/LanguageIdentifier/blob/master/language%20identification_last.py>