LA TOKENISATION :

La tokenisation est l'opération de segmenter un acte langagier en unités "atomiques" : les tokens. Les tokenisations les plus courantes sont le découpage en mots ou bien en phrases. Ainsi la tokenisation en mots de la phrase Tout le monde attendait une véritable furia rouge. nous donnerait les composants mots : « Tout », « le », « monde », « attendait », « une », « véritable », « furia », « rouge », .. la tokenisation est surtout utilisée lors des prétraitements afin d'identifier des unités de "haut niveau" sur lesquelles porteront l'analyse.

Tokenisation avec NLTK :

Utilisation en Python, et notamment le package NLTK, pour tokeniser en mots. Il est nécessaire d'ouvrir une console python et d'importer nltk à l'aide de la commande import nltk.

NORMALISATION :

(https://www.science-emergence.com/Articles/Supprimer-les-accents-dun-string-en-python/)

Normaliser une chaîne de caractère (supprimer des accents ainsi que certains caractères spéciaux). Pour gérer ces caractères spéciaux :

- Le type unicode :

Il attribue à chaque caractère un code unique. Mais surtout, il couvre plusieurs dizaines de langues avec tous leurs symboles ou glyphes. C'est un codage quasi universel des lettres ou syllabes, chiffres ou nombres, symboles divers, signes diacritiques et signes de ponctuation. D'où son nom : UNICODE. Pour créer un texte unicode sous Python, il faut l'écrire entre guillemets précédés de la lettre **u** : u"ceci est une chaine unicode" ou bien u'Encore un texte unicode'.

source : (http://www.geoinformations.developpement-durable.gouv.fr/fichier/pdf/Python_-_Gerer_les_caracteres_accentues_dans_les_textes_cle213b6c.pdf?arg=177830528&cle=31ca778b2d398b06d33455d7df94aa4653ea3726&file=pdf%2FPython_-_Gerer_les_caracteres_accentues_dans_les_textes_cle213b6c.pdf)

LEMMATISATION :

Désigne l'analyse lexicale du contenu d'un texte regroupant les mots d'une même famille. Chacun des mots d'un contenu se trouve ainsi réduit en une entité appelée lemme (forme générale). La lemmatisation regroupe les différentes formes que peut revêtir un mot, soit : le nom, le pluriel, le verbe à l'infinitif, etc.

La lemmatisation d'une forme d'un mot consiste à en prendre sa forme canonique. Celle-ci est définie comme suit :

- Pour un verbe : ce verbe à l'infinitif,
- Pour les autres mots : le mot au masculin singulier.

En informatique, il est difficile pour un programme de savoir que « eussions eu » et « avoir » sont deux facettes d'un même terme. La lemmatisation est donc une opération préliminaire pour la reconnaissance des mots d'une phrase.

STEMMING :

La racine est le processus de réduction d'un mot dans sa racine, c'est-à-dire sa forme racine. La forme racine n'est pas nécessairement un mot en elle-même, mais elle peut être utilisée pour générer des mots en concaténant le bon suffixe.

Par exemple, les mots « fish, fishes and fishing » tous se transforment en « fish », ce qui est un mot correct. De l'autre côté, les mots «study, studies and studying » se trouvent dans « studi », qui n'est pas un mot anglais.

Le plus souvent, les algorithmes de racine (aka stemmers) sont basés sur des règles pour l'effacement des suffixes.

Exemple en Python : (https://gist.github.com/bonzanini/b4336bbdae6e45d594e9)

```
from nltk import pos_tag

from nltk.tokenize import word_tokenize

from nltk.stem import PorterStemmer, WordNetLemmatizer

stemmer = PorterStemmer()

lemmatiser = WordNetLemmatizer()

print("Stem %s: %s" % ("going", stemmer.stem("going")))

print("Stem %s: %s" % ("gone", stemmer.stem("gone")))

print("Stem %s: %s" % ("goes", stemmer.stem("goes")))

print("Stem %s: %s" % ("went", stemmer.stem("went")))
```

Comparatif des deux méthodes : (http://textminingonline.com/dive-into-nltk-part-iv-stemming-and-lemmatization)

Stemming et Lemmatization sont les méthodes de traitement de texte, le but est de réduire les formes flexionnelles et parfois les formes dérivées d'un mot à une forme de base commune.

- *Le processus de stemming ne génère pas un mot réel, mais une forme racine.*
- *De l'autre côté, le lemmatiseur génère des mots réels, mais sans informations contextuelles il n'est pas capable de distinguer les noms et les verbes, donc le processus de lemmatisation ne change pas le mot.*

TYPES DE MESURES : on distingue deux types :

- Multi-Token : (Jaccard)

Peut gérer les chaines composées, sont moins sensibles aux échanges de mots ex : PeupleAfrique / AfriquePeuple.

- Single-Token : (Levenstein, Jaro, Identity)

Peut gérer des chaines avec de légères variations d'orthographe.