Introduction à NLTK avec Exercices corrigés

La **Natural Language Toolkit (NLTK)** est une bibliothèque Python incontournable pour le traitement du langage naturel (NLP). Elle est largement utilisée dans le milieu académique pour son approche pédagogique et sa richesse en ressources linguistiques.

# Installation de NLTK

Avant de commencer, assurez-vous d'avoir installé NLTK :

pip install nltk

# Importation de NLTK et téléchargement des ressources

import nltk

nltk.download('all') # Télécharge tous les corpus et modules nécessaires

*Note : Le téléchargement de toutes les ressources peut prendre du temps et de l'espace disque. Vous pouvez télécharger des modules spécifiques si nécessaire.*

**Principales fonctionnalités de NLTK**

1. **Tokenisation**
2. **Étiquetage morpho-syntaxique (Part-of-Speech Tagging)**
3. **Lemmatisation et racinisation (Stemming)**
4. **Analyse syntaxique (Parsing)**
5. **Reconnaissance des entités nommées**
6. **Analyse de sentiments**

Nous allons explorer ces fonctionnalités avec des exemples et des Exercices corrigés.

# Tokenisation

La tokenisation consiste à découper un texte en unités linguistiques appelées tokens (mots, phrases, etc.).

# Exemple :

from nltk.tokenize import word\_tokenize, sent\_tokenize texte = "Bonjour! Comment allez-vous? Je vais bien, merci." # Tokenisation en mots

mots = word\_tokenize(texte, language='french') print("Tokens de mots :", mots)

# Tokenisation en phrases

phrases = sent\_tokenize(texte, language='french') print("Tokens de phrases :", phrases) **Explication :**

* **word\_tokenize** : Découpe le texte en mots.
* **sent\_tokenize** : Découpe le texte en phrases.

# Exercice 1 :

**Énoncé :**

Tokenisez le texte suivant en mots et comptez le nombre d'occurrences de chaque mot. texte = "Le chat dort sur le canapé. Le chien joue avec le chat."

# Étiquetage morpho-syntaxique (POS Tagging)

Attribuer à chaque mot son rôle grammatical (nom, verbe, adjectif, etc.).

# Exemple :

from nltk import pos\_tag

from nltk.tokenize import word\_tokenize texte = "Le chat noir dort sous le soleil."

mots = word\_tokenize(texte, language='french')

# POS Tagging nécessite un modèle de langue; par défaut, NLTK utilise l'anglais. # Pour le français, on peut utiliser 'spacy' ou entraîner un modèle spécifique.

# Pour l'exemple, nous utiliserons l'anglais. # Exemple en anglais

texte\_en = "The black cat sleeps under the sun." mots\_en = word\_tokenize(texte\_en)

tags = pos\_tag(mots\_en) print("Étiquettes POS :", tags)

# Exercice 2 :

**Énoncé :**

Étiquetez morpho-syntaxiquement le texte suivant en anglais et identifiez les noms communs (NN). texte = "She sells sea shells on the sea shore."

1. **Lemmatisation et racinisation Racinisation (Stemming)** Réduire les mots à leur racine. **Exemple :**

from nltk.stem import SnowballStemmer stemmer = SnowballStemmer('french')

mots = ["manger", "mangé", "mangeais", "mangeront"] racines = [stemmer.stem(mot) for mot in mots] print("Racinisation :", racines)

# Lemmatisation

Trouver le lemme (forme canonique) d'un mot.

# Exemple :

La lemmatisation en français n'est pas directement supportée par NLTK, mais on peut utiliser spacy. import spacy

nlp = spacy.load('fr\_core\_news\_sm')

doc = nlp("Les chats mangent des souris.") lemmes = [token.lemma\_ for token in doc] print("Lemmatisation :", lemmes)

# Exercice 3 :

**Énoncé :**

Racinez les mots suivants et commentez les résultats : mots = ["courir", "courait", "courrons", "couru"]

# Analyse syntaxique (Parsing) Arbres syntaxiques

Créer des structures arborescentes pour représenter la syntaxe d'une phrase.

# Exemple :

from nltk import CFG

from nltk.parse import RecursiveDescentParser # Grammaire simplifiée

grammaire = CFG.fromstring(""" S -> NP VP

NP -> Det N VP -> V NP

Det -> 'le' | 'la'

N -> 'chat' | 'souris' V -> 'mange'

""")

phrase = ['le', 'chat', 'mange', 'la', 'souris'] parser = RecursiveDescentParser(grammaire) for arbre in parser.parse(phrase):

print(arbre)

arbre.draw() # Affiche l'arbre graphiquement

# Exercice 4 :

**Énoncé :**

Construisez une grammaire pour la phrase "le chien court" et affichez l'arbre syntaxique.

# Reconnaissance des entités nommées

Identifier les noms de personnes, lieux, organisations, etc.

# Exemple :

import nltk

from nltk import ne\_chunk, pos\_tag, word\_tokenize texte = "Barack Obama était le président des États-Unis." # Tokenisation et POS Tagging

mots = word\_tokenize(texte) tags = pos\_tag(mots)

# Reconnaissance des entités

entites = ne\_chunk(tags, binary=False) print(entites)

*Note : La reconnaissance des entités nommées avec NLTK est optimisée pour l'anglais.*

# Exercice 5 :

**Énoncé :**

Sur le texte suivant en anglais, identifiez les entités nommées. texte = "Apple is looking at buying U.K. startup for $1 billion."

# Analyse de sentiments

Évaluer le sentiment exprimé dans un texte (positif, négatif, neutre).

# Exemple avec TextBlob (extension de NLTK) :

from textblob import TextBlob

texte = "C'est une excellente journée!" blob = TextBlob(texte)

sentiment = blob.sentiment.polarity # Entre -1 (négatif) et 1 (positif) print("Sentiment :", sentiment)

*Note : TextBlob supporte mieux l'anglais pour l'analyse de sentiments.*

# Exercice 6 :

**Énoncé :**

Utilisez TextBlob pour analyser le sentiment du texte suivant en anglais : texte = "I am not happy with the service."