RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Sorbonne Nouvelle
université des cultures

Lattice

Text mining

Statistiques textuelles - LZML041

Séance 3 : Python et Corpus

Chargé de cours : Loubna Serrar

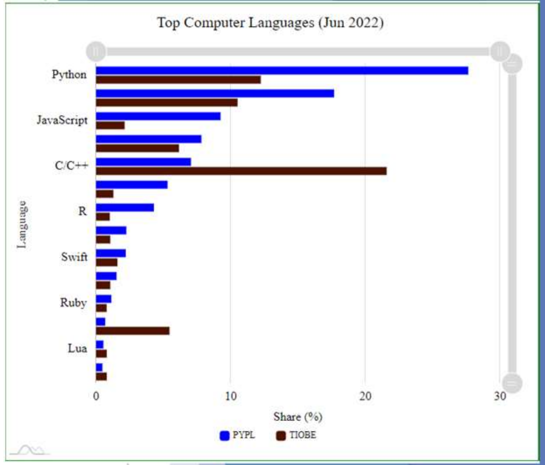
ANNÉE UNIVERSITAIRE 2022-2023

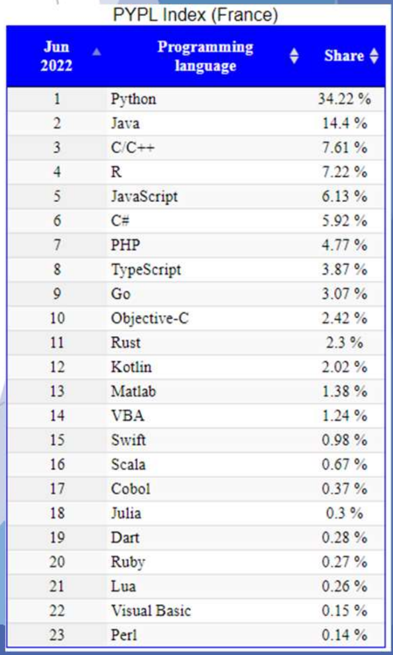
Description du cours

- 12 séances de 2h pour **s'initier au Text mining**
- Un mix de théorie et de pratique avec **l'outil Python/Jupyterlab**
- **Évaluation en controle continu** avec des tests courts toutes les 3 séances et un travail final à rendre en mai (50/50)
- **Plan des premières séances:**
  1. **Introduction Générale**
  2. **Introduction Python**
  3. **What is a corpus ?**
  4. **pre-processing techniques** (Cleaning? Segmentation, tokenization, part of speech tagging, syntactic parser, data annotation…)
  5. **Text to numbers** (Bag of words, frequency, TF-IDF, n-gramm model…

Why Python ?

- Python was created by Guido van Rossum, and first released on February 20, 1991. While you may know the python as a **large snake**, the name of the Python programming language comes from an old BBC television comedy sketch series called *Monty Python's Flying Circus*. See Guido introducing Python story here: https://www.youtube.com/watch?v=J0Aq44Pze-w
- Python is an open-source language, with a less complicated syntax than C or other object oriented languages, and it has a large set of libraries available to use !
- It benefits from :
  - Support for powerful Data Science libraries such as numpy, Scikit-Learn, matplotlib, TensorFlow and NLTK !
  - Provide tools for data collection, analysis, modeling, and visualization
  - Supports numerous file export and sharing options
  - Comes with a strong community for getting support !
- Python is omnipresent, and people use numerous platforms that we use today were developed with Python, including YouTube, Google Search and iRobot machines.




PYPL Index (France)

Jun 2022	Programming language	Share
1	Python	34.22 %
2	Java	14.4 %
3	C/C++	7.61 %
4	R	7.22 %
5	JavaScript	6.13 %
6	C#	5.92 %
7	PHP	4.77 %
8	TypeScript	3.87 %
9	Go	3.07 %
10	Objective-C	2.42 %
11	Rust	2.3 %
12	Kotlin	2.02 %
13	Matlab	1.38 %
14	VBA	1.24 %
15	Swift	0.98 %
16	Scala	0.67 %
17	Cobol	0.37 %
18	Julia	0.3 %
19	Dart	0.28 %
20	Ruby	0.27 %
21	Lua	0.26 %
22	Visual Basic	0.15 %
23	Perl	0.14 %

Installation Python

- Anaconda est un outil permettant d'installer Python : c'est-à-dire qu'en installant Anaconda, vous installerez Python, Jupyter Notebook et des dizaines de packages scientifiques, dont certains indispensables à l'analyse de données !
- Pour commencer, téléchargez la distribution Anaconda correspondant à votre système d'exploitation, en Python version 3 : https://www.anaconda.com/distribution/
- **Installez Anaconda sur Windows ou Mac**
  - Téléchargez l'installeur Windows ou macOS, et double-cliquez pour lancer l'installation.
  - Répondez aux différentes questions (les options par défaut suffisent !). Une fois l'installation terminée, vous pouvez vérifier que celle-ci s'est bien passée en lançant l'application Jupyter Notebook.
- **Lancez Jupyter**
  - Sous macOS, lancez Anaconda Navigator via le Launchpad.
  - Sous Windows, lancez Anaconda Navigator depuis votre liste de programmes.
- Vous pouvez maintenant ouvrir Jupyter Notebook ou lab !

Attention, si un souci : désinstaller Anaconda et recommencez ! Assurez-vous de créer Anconda dans votre disque C avec un nom simple en un seul mot ….

Jupyter Notebook

- Jupyter Notebook est une application web qui vous permet de stocker des lignes de code Python, les résultats de l’exécution de ces dernières (graphiques, tableaux, etc.) et du texte formaté. Cela peut être comparé à une page web, mais pas besoin d’une connexion internet !
- Jupyter Notebook est un outil puissant qui permet de :
  - d’écrire des petits bouts de code exécutable (appelés « cellules »), de les documenter (« commenter ») pour expliquer ce qu’ils font et d’afficher les données résultant de leur exécution.
  - stocker ces lignes de code Python, les résultats de l’exécution de ces dernières (graphiques, tableaux, etc.) et du texte formaté
  - de les partager facilement avec d’autres utilisateurs, et permettre des pratiques collaboratifs !
- Maintenant, ouvrez Jupyter Notebook : cela va ouvrir un nouvel onglet dans votre navigateur Internet appelé Home. Commencez par créer un nouveau notebook en cliquant sur New puis Python 3

Exercise 1: executer print('Hello World !’) dans votre première cellule

Premiers pas

- **Commentaires # :** Les langages de programmation proposent une notation pour insérer des *commentaires* dans le code, **c'est-à-dire du texte qui va être ignoré par l'ordinateur.**
- **Affichage print() :** un résultat peut-être imprimé à l'écran avec une commande print()
- **Variables() :** Une variable est un conteneur qui sert à stocker une valeur.
  - Les variables ont une durée de vie limitée (une variable ne vit généralement que le temps de l'exécution d'un script ou de la fonction qui l'a définie) ;
  - La valeur d'une variable peut varier, on peut la modifier durant la vie du script.
  - Pas besoin d'une déclaration un simple X=15 suffit

Erreur Code Python : NameError: name 'X' is not defined

- **Type() :** Pour connaître le type d'une variable, il suffit d'utiliser la fonction type().
  - les entiers (*integer* ou *int*),
  - les nombres décimaux que nous appellerons *floats*
  - les chaînes de caractères (*string* ou *str*).
  - il existe de nombreux autres types (par exemple, les booléens, les nombres complexes, etc.).

Ouvrez le notebook: 20230217_Intro.ipynb

Tests et boucles

- **Boucles :** Les boucles s'utilisent pour répéter plusieurs fois l'exécution d'une partie du programme.
  - Boucle bornée : Quand on sait combien de fois doit avoir lieu la répétition, on utilise généralement **une boucle for**.
  - Boucle non bornée : Si on ne connait pas à l'avance le nombre de répétitions, on choisit une **boucle while**.
- **Tests :** L'instruction if est la structure de test la plus simple. Sa syntaxe en Python fait intervenir la notion de bloc ou indentation.
  - Instruction If: L'instruction A n'est exécutée que si la condition est vérifiée (c'est-à-dire si elle prend pour valeur True).
  - Instruction if … else: Si on ne connait pas à l'avance le nombre de répétitions, on choisit une **boucle while**.

Boucle for

Exemple d'utilisation :

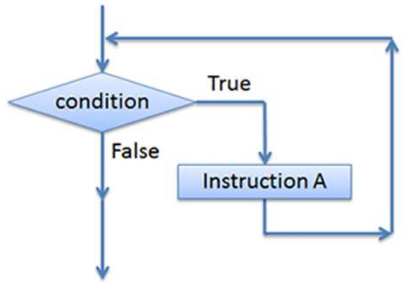
```
for i in [0, 1, 2, 3]:
    print("i a pour valeur", i)
```

Exécuter

Affichage après exécution :

```
i a pour valeur 0
i a pour valeur 1
i a pour valeur 2
i a pour valeur 3
```

```
while condition:
    Instruction A
```



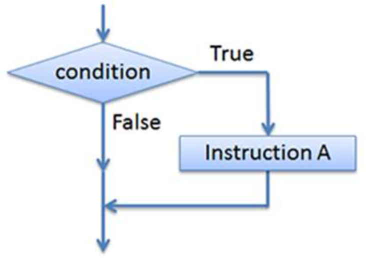
Exemple de programme :

```
x = 1
while x < 10:
    print("x a pour valeur", x)
    x = x * 2
print("Fin")
```

Instruction if

Syntaxe

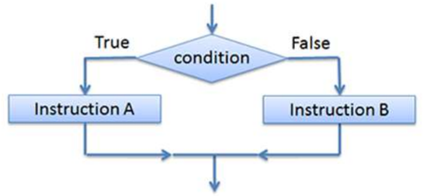
```
if condition:
    Instruction A
```



Instruction if … else

Syntaxe

```
if condition:
    Instruction A
else:
    Instruction B
```



Exemple où la condition est vraie :

```
x = 5
if x > 0:
    print(x, "est positif")
else:
    print(x, "est négatif ou nul")
print("Fin")
```

Ouvrez le notebook: 20230217_Intro.ipynb

Containers de données

- **Les tuples**: Il permet de créer une collection ordonnée de plusieurs éléments. Par exemple : a=(12,14,59)
- **Les listes []:** Sous Python, on peut définir une liste comme une *collection d'éléments séparés par des virgules, l'ensemble étant enfermé dans des crochets*. Exemple: ["girafe", 8, "souris", 0.18]
- Des listes de listes [[]]
- **Les dictionnaires {}:** Un dictionnaire en Python va aussi permettre de rassembler des éléments mais ceux-ci seront identifiés par une clé. On peut faire l'analogie avec un dictionnaire de français où on accède à une définition avec un mot. Exemple: dict_test = {"voiture": "quatre roues", "vélo": "deux roues"}
- Dataframe pandas: un dataframe (matrice) se comporte comme un dictionnaire dont les clefs sont les noms des colonnes et les valeurs sont des séries.

Ouvrez le notebook: 20230217_Intro.ipynb

Lets open Python / Jupyter !

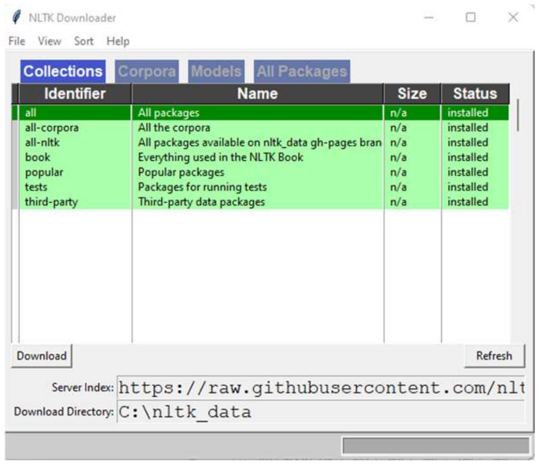
- **The Natural Language Toolkit (NLTK)** is an open source library for building Python programs that work with human language data for applying in statistical natural language processing (NLP). Online NLKT guide here.
- Installing **NLTK library :** do the following in your Jupyter notebook.

  pip install nltk

  import nltk

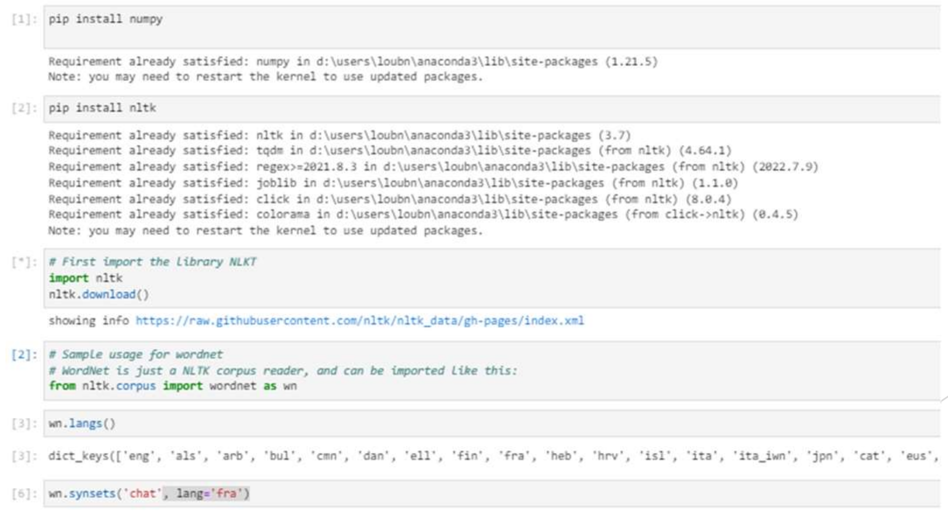
  nltk.download()

✓ A new window should open, showing the NLTK Downloader. Click on the File menu and select Change Download Directory. For central installation, set this to C:\nltk_data (Windows), /usr/local/share/nltk_data (Mac), or /usr/share/nltk_data (Unix). Next, download all.



Identifier	Name	Size	Status
all	All packages	n/a	installed
all-corpora	All the corpora	n/a	installed
all-nltk	All packages available on nltk_data gh-pages bran	n/a	installed
book	Everything used in the NLTK Book	n/a	installed
popular	Popular packages	n/a	installed
tests	Packages for running tests	n/a	installed
third-party	Third-party data packages	n/a	installed

Server Index: https://raw.githubusercontent.com/nlt
Download Directory: C:\nltk_data

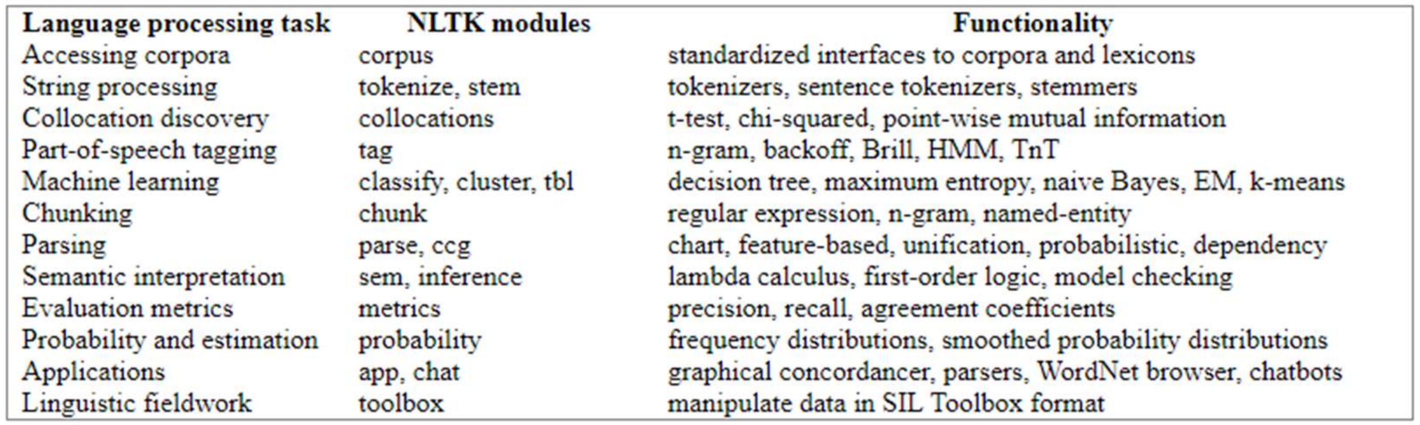


```
[1]: pip install numpy

     Requirement already satisfied: numpy in d:\users\loubn\anaconda3\lib\site-packages (1.21.5)
     Note: you may need to restart the kernel to use updated packages.

[2]: pip install nltk

     Requirement already satisfied: nltk in d:\users\loubn\anaconda3\lib\site-packages (3.7)
     Requirement already satisfied: tqdm in d:\users\loubn\anaconda3\lib\site-packages (from nltk) (4.64.1)
     Requirement already satisfied: regex>=2021.8.3 in d:\users\loubn\anaconda3\lib\site-packages (from nltk) (2022.7.9)
     Requirement already satisfied: joblib in d:\users\loubn\anaconda3\lib\site-packages (from nltk) (1.1.0)
     Requirement already satisfied: click in d:\users\loubn\anaconda3\lib\site-packages (from nltk) (8.0.4)
     Requirement already satisfied: colorama in d:\users\loubn\anaconda3\lib\site-packages (from click->nltk) (0.4.5)
     Note: you may need to restart the kernel to use updated packages.

[*]: # First import the Library NLKT
     import nltk
     nltk.download()

     showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml

[2]: # Sample usage for wordnet
     # WordNet is just a NLTK corpus reader, and can be imported like this:
     from nltk.corpus import wordnet as wn

[3]: wn.langs()

[3]: dict_keys(['eng', 'als', 'arb', 'bul', 'cmn', 'dan', 'ell', 'fin', 'fra', 'heb', 'hrv', 'isl', 'ita', 'ita_iwn', 'jpn', 'cat', 'eus',

[6]: wn.synsets('chat', lang='fra')
```

Python Toolkit (NLTK)

- **The Natural Language Toolkit (NLTK)** is an open source library for building Python programs that work with human language data for applying in statistical natural language processing (NLP). Online NLKT guide here.
- It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Language processing tasks and corresponding NLTK modules with examples of functionality



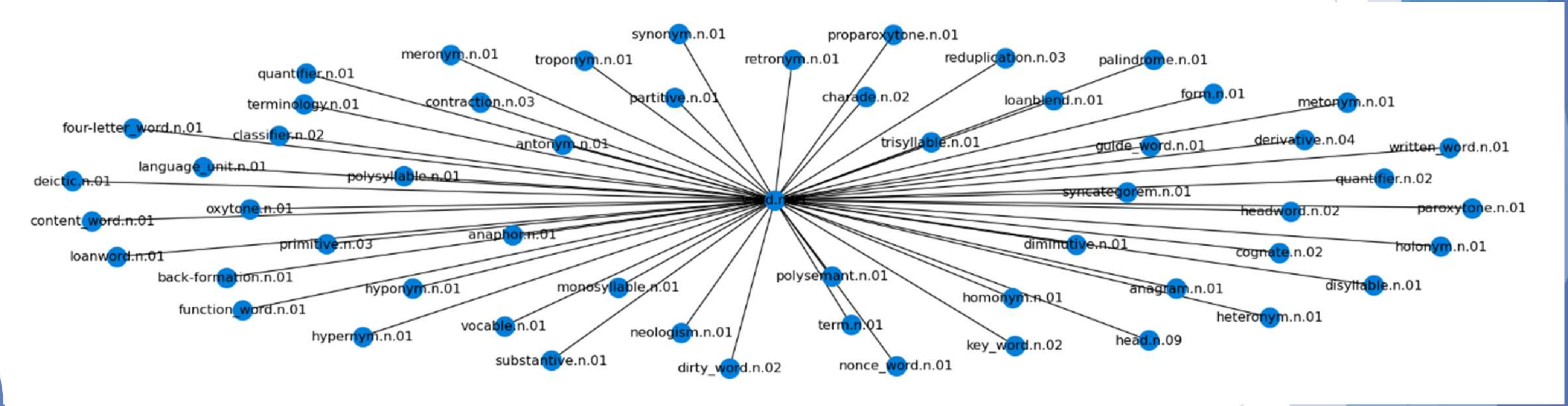
Language processing task	NLTK modules	Functionality
Accessing corpora	corpus	standardized interfaces to corpora and lexicons
String processing	tokenize, stem	tokenizers, sentence tokenizers, stemmers
Collocation discovery	collocations	t-test, chi-squared, point-wise mutual information
Part-of-speech tagging	tag	n-gram, backoff, Brill, HMM, TnT
Machine learning	classify, cluster, tbl	decision tree, maximum entropy, naive Bayes, EM, k-means
Chunking	chunk	regular expression, n-gram, named-entity
Parsing	parse, ccg	chart, feature-based, unification, probabilistic, dependency
Semantic interpretation	sem, inference	lambda calculus, first-order logic, model checking
Evaluation metrics	metrics	precision, recall, agreement coefficients
Probability and estimation	probability	frequency distributions, smoothed probability distributions
Applications	app, chat	graphical concordancer, parsers, WordNet browser, chatbots
Linguistic fieldwork	toolbox	manipulate data in SIL Toolbox format

Jupyterlab keyboard shortcuts

- Command Mode : press Esc to enable)
  - ✓ F : find and replace
  - ✓ Ctrl-Shift-F : open the command palette
  - ✓ Ctrl-Shift-P : open the command palette
  - ✓ Enter : enter edit mode
  - ✓ P : open the command palette
  - ✓ Shift-Enter : run cell, select below
  - ✓ Ctrl-Enter : run selected cells
  - ✓ Alt-Enter : run cell and insert below
  - ✓ **Y : change cell to code**
  - ✓ M : change cell to markdown
  - ✓ R : change cell to raw
  - ✓ 1 : change cell to heading 1
  - ✓ 2 : change cell to heading 2
  - ✓ 3 : change cell to heading 3
  - ✓ 4 : change cell to heading 4
  - ✓ 5 : change cell to heading 5
  - ✓ 6 : change cell to heading 6
  - ✓ K : select cell above
  - ✓ Up : select cell above
  - ✓ Down : select cell below
  - ✓ J : select cell below
  - ✓ Shift-K : extend selected cells above
  - ✓ Shift-Up : extend selected cells above
  - ✓ Shift-Down : extend selected cells below
  - ✓ Shift-J : extend selected cells below
  - ✓ Ctrl-A : select all cells
  - ✓ A : insert cell above
  - ✓ B : insert cell below
  - ✓ X : cut selected cells
  - ✓ C : copy selected cells
  - ✓ Shift-V : paste cells above
  - ✓ V : paste cells below
  - ✓ Z : undo cell deletion
  - ✓ D,D : delete selected cells
  - ✓ Shift-M : merge selected cells, or current cell with cell below if only one cell is selected
  - ✓ Ctrl-S : Save and Checkpoint
  - ✓ S : Save and Checkpoint
  - ✓ L : toggle line numbers
  - ✓ O : toggle output of selected cells
  - ✓ Shift-O : toggle output scrolling of selected cells
  - ✓ H : show keyboard shortcuts
  - ✓ I,I : interrupt the kernel
  - ✓ 0,0 : restart the kernel (with dialog)
  - ✓ Ctrl-V : Dialog for paste from system clipboard
  - ✓ Esc : close the pager
  - ✓ Q : close the pager
  - ✓ Shift-L : toggles line numbers in all cells, and persist the setting
  - ✓ Shift-Space : scroll notebook up
  - ✓ Space : scroll notebook down

Corpus example 1 : Wordnet

- WordNet, created by Princeton is a lexical database for English language. It is the part of the NLTK corpus.
- NLTK module includes the English WordNet **with** 155 287 words and 117 659 synonym sets that are logically related to each other.
- In WordNet, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms called Synsets. All the synsets are linked with the help of conceptual-semantic and lexical relations. Its structure makes it very useful for natural language processing (NLP).
- In information systems, WordNet is used for various purposes like word-sense disambiguation, information retrieval, automatic text classification and machine translation. One of the most important uses of WordNet is to find out the similarity among words.



Lets do some Python !

Ambiguity and Uncertainty in Language

- Lexical Ambiguity : The ambiguity of a single word is called lexical ambiguity. For example, treating the word **silver** as a noun, an adjective, or a verb.
- Syntactic Ambiguity : This kind of ambiguity occurs when a sentence is parsed in different ways. For example, the sentence “The man saw the girl with the telescope”. It is ambiguous whether the man saw the girl carrying a telescope or he saw her through his telescope.
- Semantic Ambiguity : This kind of ambiguity occurs when the meaning of the words themselves can be misinterpreted. In other words, semantic ambiguity happens when a sentence contains an ambiguous word or phrase. For example, the sentence “The car hit the pole while it was moving” is having semantic ambiguity because the interpretations can be “The car, while moving, hit the pole” and “The car hit the pole while the pole was moving”.
- Anaphoric Ambiguity : This kind of ambiguity arises due to the use of anaphora entities in discourse. For example, the horse ran up the hill. It was very steep. It soon got tired. Here, the anaphoric reference of “it” in two situations cause ambiguity.
- Pragmatic ambiguity : Such kind of ambiguity refers to the situation where the context of a phrase gives it multiple interpretations. In simple words, we can say that pragmatic ambiguity arises when the statement is not specific. For example, the sentence “I like you too” can have multiple interpretations like I like you (just like you like me), I like you (just like someone else dose).

Corpus example 2 : Guttenberg

- The Gutenberg dataset represents a corpus of over 15,000 book texts, their authors and titles, all available on : https://www.gutenberg.org/
- The text data itself can be downloaded using the gutenberg_download.py script, which will parse the metadata file, download the text data for each book and save the results as a csv file. The final csv file containing the book texts, the authors, the titles and the categories will have a size of around 5 GB. https://www.kaggle.com/datasets/mateibejan/15000-gutenberg-books
- But we will use another code to select only the books we choose !

Corpus example 3 : Twitter

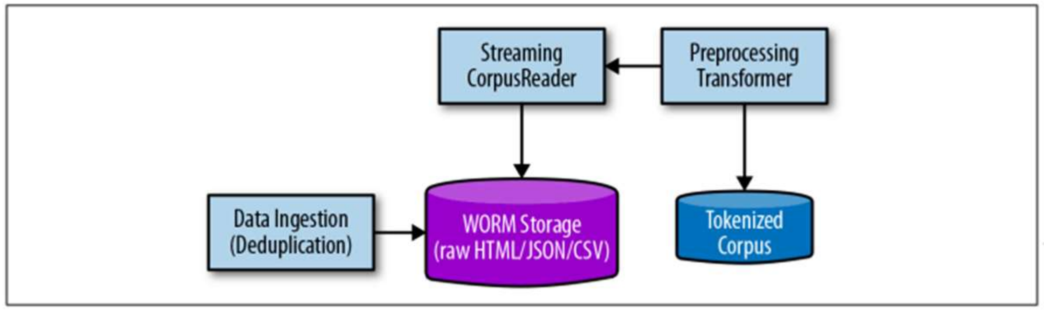
- You can use the Twitter **RESTful API** to access data about both Twitter users and what they are tweeting about.
- To get started, you'll need to do the following things:
  - Set up a Twitter account if you don't have one already.
  - Using your Twitter account, you will need Apply to the Twitter Developer Account.
  - **Setup up a project** using this link. You would be asked to provide the project name, use case (similar to what you did while applying for the developer account), and a project description.
  - Once you've finished the preceding steps, you need to next **create an App**. It will be within the project you created in the previous step. *Important: the name of the app you're creating must not be duplicated or else you may receive an error.*
  - On the next screen, you'll be presented with **keys & tokens** i.e. API Key, API Key Secret, and Bearer Token. *Important: Please save these on your local machine, you will be using it later.*
  - Import the tweepy package.

```
import os
import tweepy as tw
import pandas as pd
```

See : https://www.earthdatascience.org/courses/use-data-open-source-python/intro-to-apis/twitter-data-in-python/

Corpus Data Management

- Usually, Text mining studies are applied to corpora containing thousands or tens of thousands of documents comprising gigabytes of data. We can also assume that it will require a preliminary steps of cleaning and pre-processing…
- The simplest and most common method of organizing and managing a text-based corpus is to store individual documents in a file system on disk. By maintaining each document as its own file, corpus readers can seek quickly to different subsets of documents and processing can be parallelized, with each process taking a different subset of documents.
- Data products often employ write-once, read-many (WORM) storage as an inter-mediate data management layer between ingestion and pre-processing as shown in Figure 2-2. WORM stores provide streaming read accesses to raw data in a repeatable and scalable fashion, and pre-processed data can be reanalysed without reingestion, allowing new hypotheses to be easily explored on the raw data format



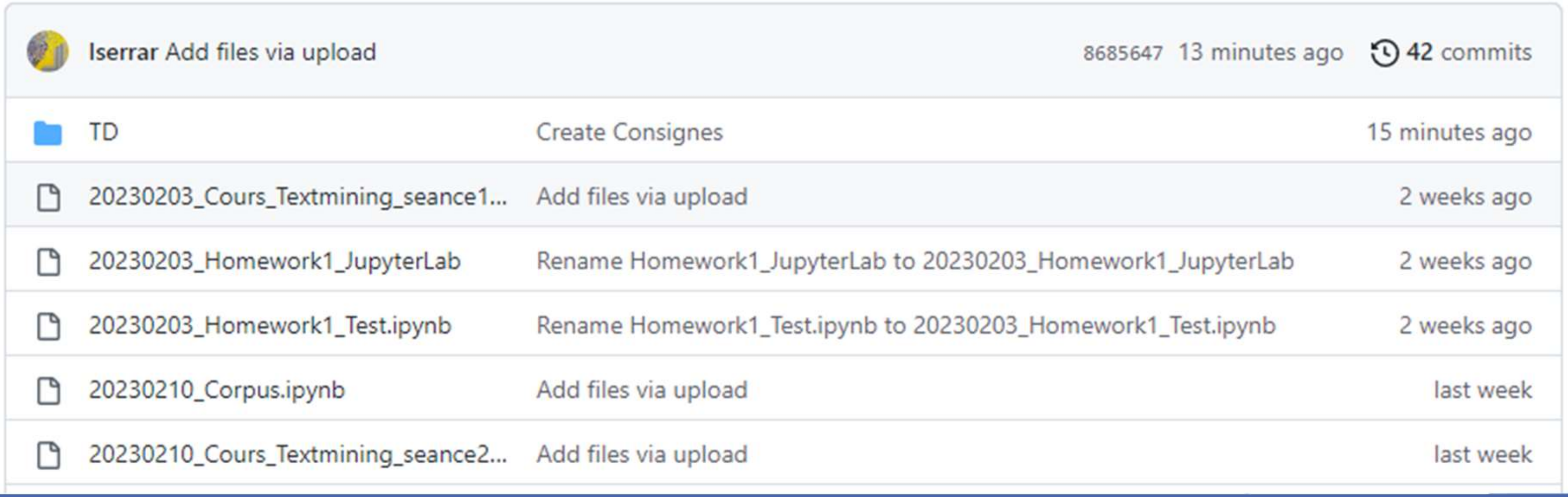
Streaming CorpusReader
Preprocessing Transformer
Data Ingestion (Deduplication)
WORM Storage (raw HTML/JSON/CSV)
Tokenized Corpus

Figure 2-2. WORM storage supports an intermediate wrangling step

Lets do some Python !

For next week :

- Repository in GitHub : https://github.com/lserrar/LZML041
- Créer un compte personnel Github, et m'envoyer votre identifiant sur loubna.serrar@gmail.com
- Vous recevrez une invitation de Github pour vous joinder au groupe que vous devez accepter.
- Les slides du cours et les notebooks y seront disponible pour chaque séance



lserrar Add files via upload 8685647 13 minutes ago 42 commits

TD	Create Consignes	15 minutes ago
20230203_Cours_Textmining_seance1...	Add files via upload	2 weeks ago
20230203_Homework1_JupyterLab	Rename Homework1_JupyterLab to 20230203_Homework1_JupyterLab	2 weeks ago
20230203_Homework1_Test.ipynb	Rename Homework1_Test.ipynb to 20230203_Homework1_Test.ipynb	2 weeks ago
20230210_Corpus.ipynb	Add files via upload	last week
20230210_Cours_Textmining_seance2...	Add files via upload	last week