Text mining

Statistiques textuelles - LZML041

Séance 4 : Corpus et pre-processing

Chargé de cours : Loubna Serrar

ANNÉE UNIVERSITAIRE 2022-2023

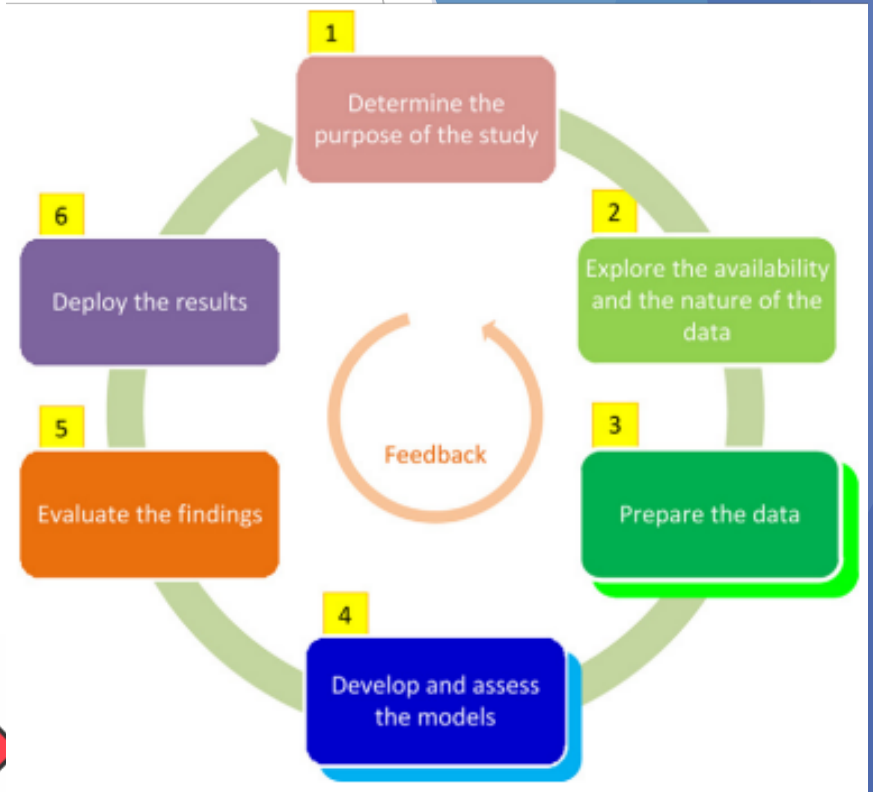
Description du cours

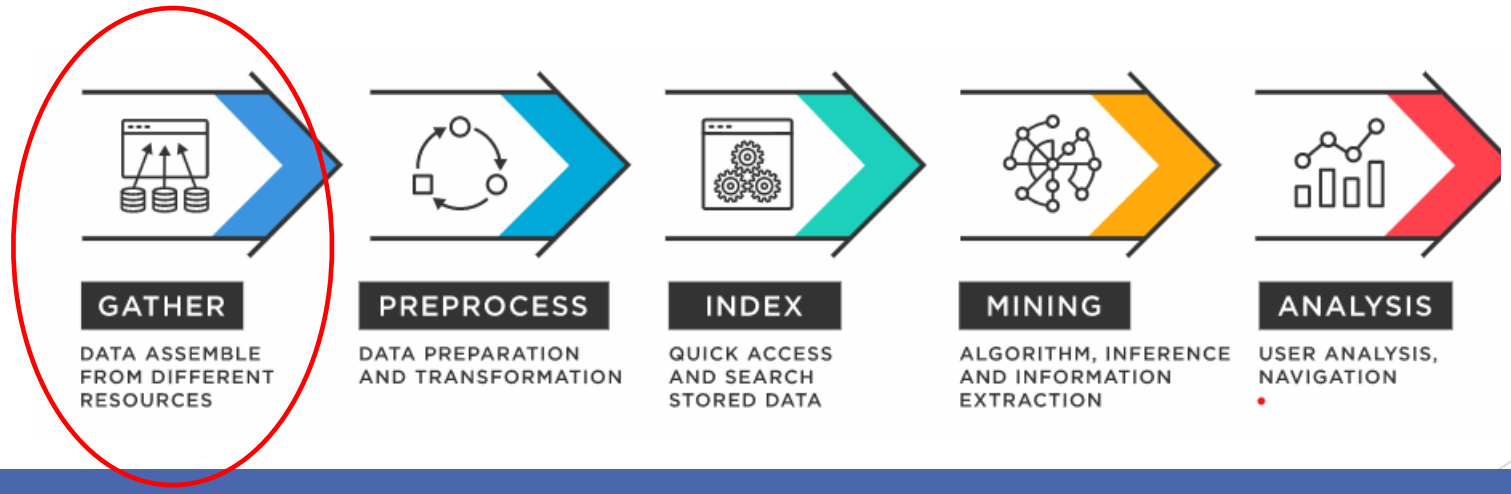
- 12 séances de 2h pour **s'initier au Text mining**
- Un mix de théorie et de pratique avec **l'outil Python/Jupyterlab**
- **Évaluation en controle continu** avec des tests courts toutes les 3 séances et un travail final à rendre en mai (50/50)
- **Plan des premières séances:**
  1. **Introduction 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…

How does Text Mining work?

- The six phases that naturally describes the data science life cycle of a data mining project:

1. Determining the purpose of the study: what is the question we want to answer?
2. Data understanding: what data do we have / need?
3. Data preparation: how do we organize the data for modeling?
4. Modeling: what modeling techniques should we apply?
5. Evaluation: which model best meets the study objectives?
6. Deployment: how do users access the results?





What is a corpus?

- Corpora are collections of related documents that contain natural language.
- A corpus can vary in size, from tweets to books, but they contain text (and sometimes metadata) inside of thousands of documents.

General

- WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.
- Google Blogger Corpus: Nearly 700,000 blog posts from blogger.com. The meat of the blogs contain commonly occurring English words, at least 200 of them in each entry.
- Enron Dataset: Over half a million anonymized emails from over 100 users. It's one of the few publically available collections of "real" emails available for study and training sets.
- SMS Spam Collection: Excellent dataset focused on spam. Nearly 6000 messages tagged as legitimate or spam messages with a useful subset extracted directly from Grumbletext.
- Recommender Systems Datasets: Datasets from a variety of sources, including fitness tracking, video games, song data, and social media. Labels include star ratings, time stamps, social networks, and images.

Specific:

- 20 Newsgroups: 20,000 documents from over 20 different newsgroups. The content covers a variety of topics with some closely related for reference. There are three versions, one in its original form, one with dates removed, and one with duplicates removed.
- The WikiQA Corpus: Contains question and sentence pairs. It's robust and compiled from Bing query logs. There are over 3000 questions and over 29,000 answer sentences with just under 1500 labeled as answer sentences.
- European Parliament Proceedings Parallel Corpus: Sentence pairs from Parliament proceedings. There are entries from 21 European languages including some less common entries for ML corpus.
- Jeopardy: Over 200,000 questions from the famed tv show. It includes category and value designations as well as other descriptors like question and answer fields and rounds.
- Legal Case Reports Dataset: Text summaries of legal cases. It contains wrapups of over 4000 legal cases and could be great for training for automatic text summarization.
- LibriSpeech: Nearly 1000 hours of speech in English taken from audiobook clips.

Litterature

- Project Gutenberg: Extensive collection of book texts. These are public domain and available in a variety of languages, spanning a long period of time.
- The Internet Archive, is building a digital library of Internet sites and other cultural artifacts in digital form.
- ABU : l'Association des Bibliophiles Universels ou ABU se propose de maintenir un corpus le plus vaste possible de textes numérisés en français du domaine public, représentatifs de la culture francophone.

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

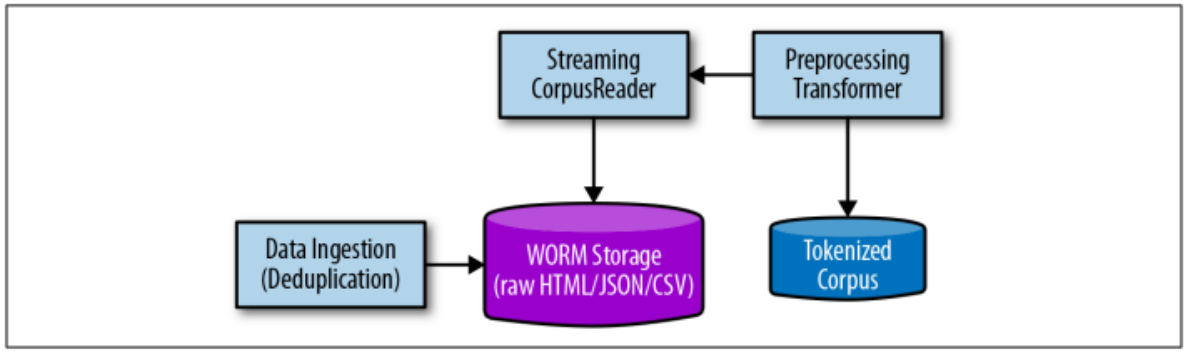
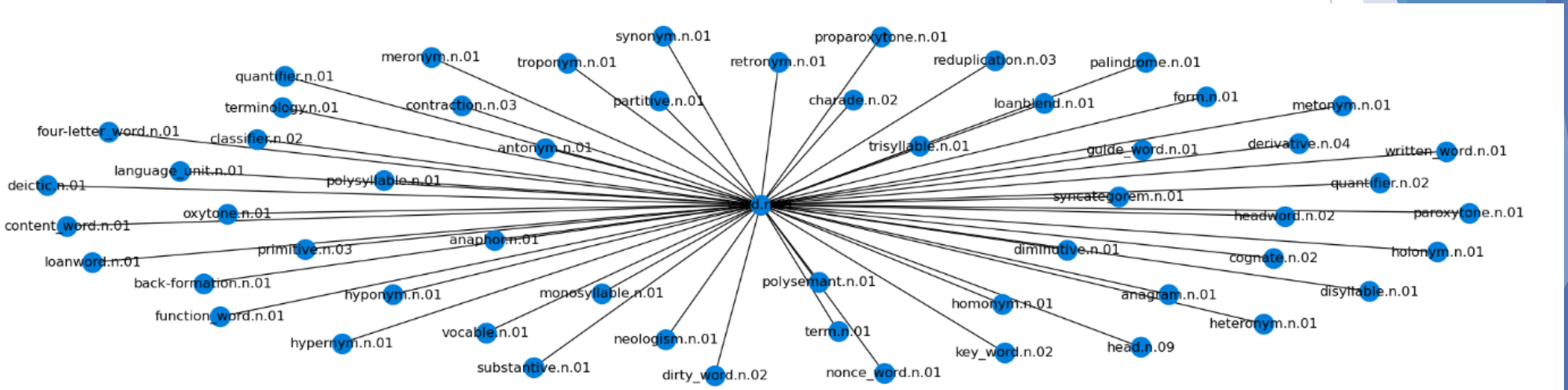


Figure 2-2. WORM storage supports an intermediate wrangling step

Lets do some Python !

Corpus example 1 : Wordnet

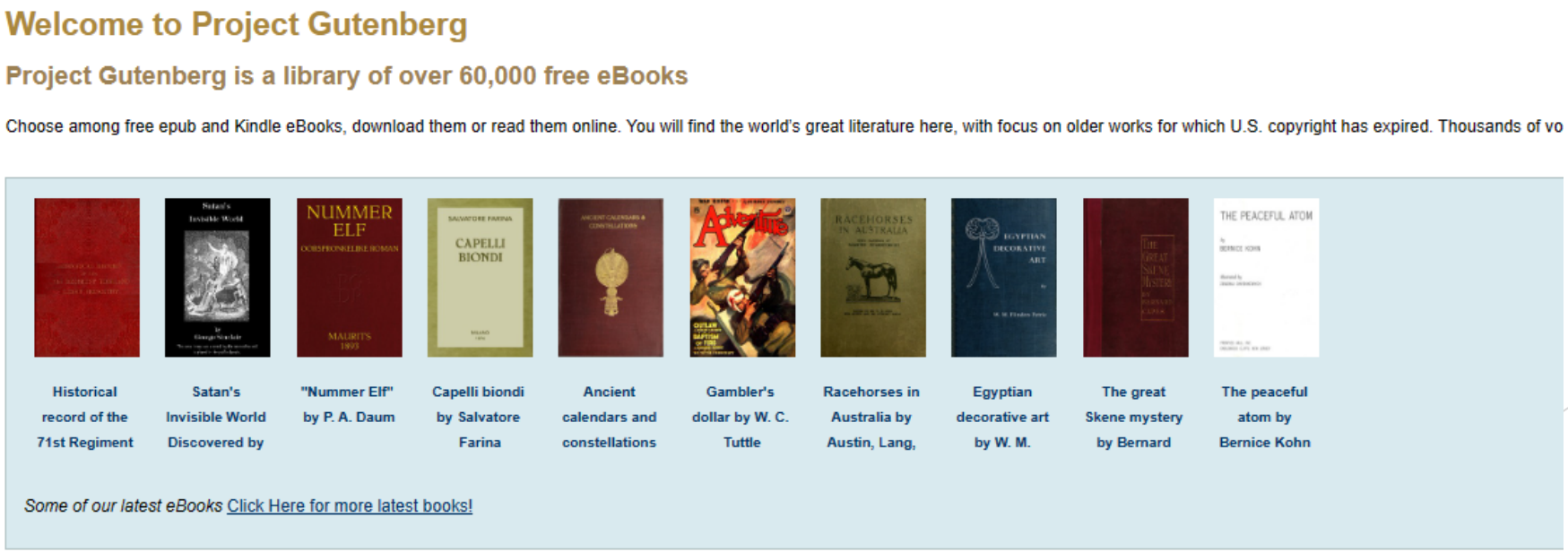
- WordNet, created by Princeton is a lexical database for English language. It is the part of the NLTK corpus.
- NLTK module includes the English WordNet **with 155 287 words and 117 659 synonym** sets that are logically related to each other.
- In WordNet, nouns, verbs, adjectives and adverbs are grouped 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 !

Corpus example 2 : Guttenberg

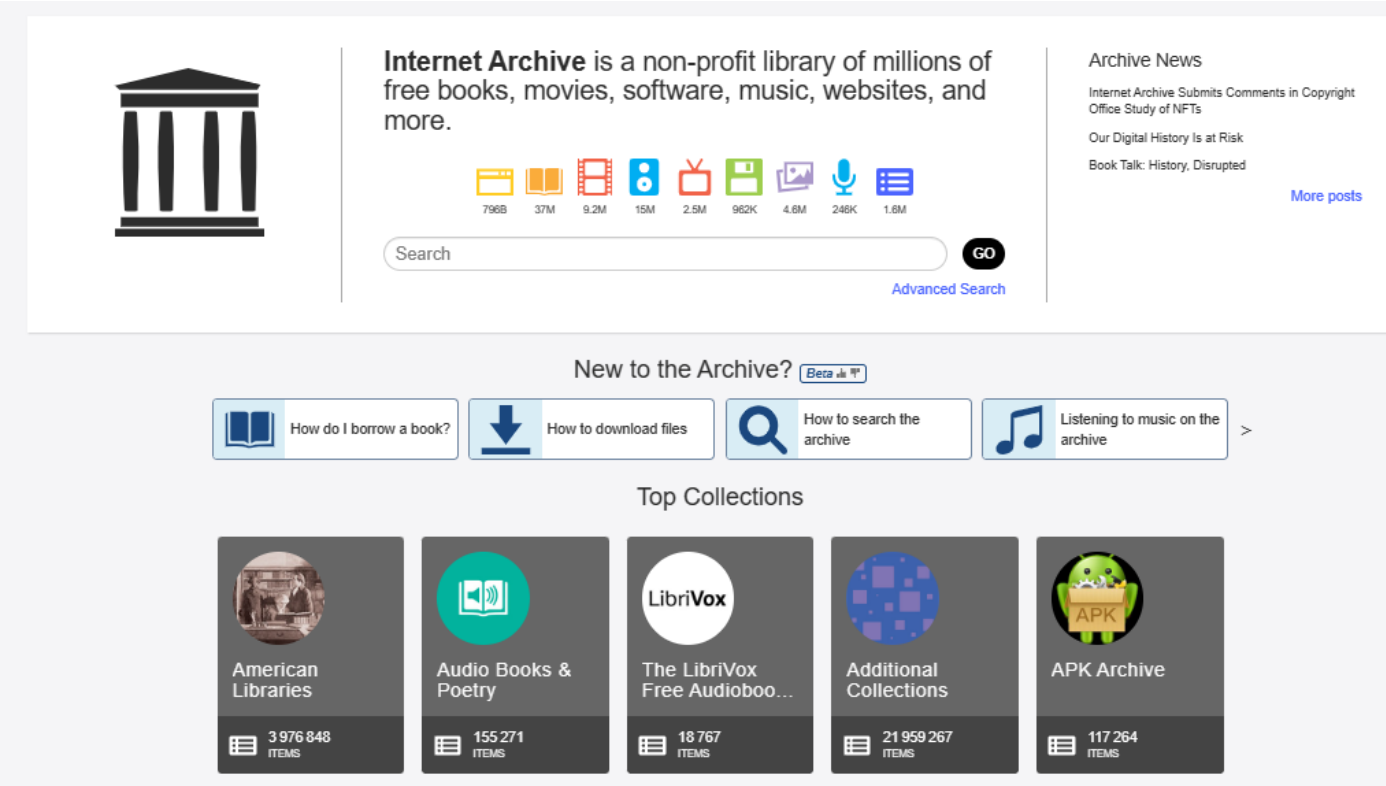
- The Gutenberg dataset represents a corpus of over 15,000 book texts, their authors and titles, all available on : https://www.gutenberg.org/
- The text data itself can be downloaded using the gutenberg_download.py script, which will parse the metadata file, download the text data for each book and 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 !



Lets do some Python !

Corpus example 3 : Internet Archive

- Internet Archive is a non-profit library of millions of free books, movies, software, music, websites, all available on : https://archive.org/
- Today the archive contains: 735 billion web pages, **41 million books and texts**, 14.7 million audio recordings, 8.4 million videos, and 4.4 million images



Corpus example 4 : Twitter

- Twitter is a rich source of data. Analyzing the tweets can give you important and interesting insights about what people are talking about, their opinions towards a particular topic/brand and some general trends in media.
- To get started, you'll need to do the following things:
  - Set up a Twitter account if you don't have one already.

Option 1: Tweepy Library	Option 2: snscrape Library
◦ Tweepy is one of the most popular Python libraries to set up access with Twitter. It is a great tool for simple automation, creating Twitter bots, or a small school project. However, Tweepy has a scraping limit of 3200 tweets and the farthest time you can go in a week. There is no access to historical data. ◦ 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.	◦ snscrape is a scraper for social networking services (SNS). It scrapes things like user profiles, hashtags, or searches and returns the discovered items, e.g. the relevant posts. ◦ The following services are currently supported: ▪ Facebook: user profiles, groups, and communities (aka visitor posts) ▪ Instagram: user profiles, hashtags, and locations ▪ Mastodon: user profiles and toots (single or thread) ▪ Reddit: users, subreddits, and searches (via Pushshift) ▪ Telegram: channels ▪ Twitter: users, user profiles, hashtags, searches, tweets (single or surrounding thread), list posts, and trends ▪ VKontakte: user profiles ▪ Weibo (Sina Weibo): user profiles
See : https://docs.tweepy.org/en/stable/	See: https://github.com/JustAnotherArchivist/snscrape

Lets do some Python !

Data Pre-processing

- Data preparation is one of the most important steps in Text Mining. **This step will determine the quality of the results** we can expect !
- For a standard Text Mining project, we spend **almost 80% of our time collecting** and cleaning the data.
- Data pre-processing consists of multiple sub-steps that helps clean the noise and prepare the data for your study.
- Most of the below steps are generic and need to be adapted to your corpus and the purpose of your study !

Cleaning data	Normalising data
○ Remove Unicode Strings and Noise	○ Spelling Correction
○ Remove/Replace URLs, User Mentions and Hashtags	○ Replace Negations with Antonyms
○ Non-Letter characters: numbers, emojis, or hash marks.	○ Handling Capitalized Words
○ Remove/Replace Slang and Abbreviations	○ Lowercase
○ Remove/Replace Contractions	○ Tokenization
○ Remove/Replace Numbers	○ Remove Stopwords (ex: the, and….)
○ Remove/Replace Repetitions of Punctuation	○ Stemming
○ Remove Punctuation	○ Lemmatizing
○ Handling Capitalized Words / Lowercase	
○ Replace Elongated Words (ex: hahahaaaa, ‘Duuuuude, that's awful,’”)	

Lets do some Python !

Devoir numéro 2 pour 5pts

- Objectif : création de votre premier corpus
- Consignes :
  - Le corpus doit contenir au moins 70000 tweets
  - Le corpus doit compter au moins 5 ans d'historique
  - Préciser votre sujet de recherche: tweets contenant un terme précis (ex: Bitcoins), ou émis d'un compte particulier (ex: @Elonmusk)
- À rendre pour le 10 mars 2023 :
  - Votre corpus au format csv nomenclature: Tweets_SujetXXX_prenom_NOM
  - à poster sur GitHub : https://github.com/lserrar/LZML041/tree/main/TD/TD02