[Files : 2](#_Toc91712643)

[Original files : 2](#_Toc91712644)

[Codes : 2](#_Toc91712645)

[Outputs: 2](#_Toc91712646)

[Explication conception 3](#_Toc91712647)

[Step N°1: split\_text 3](#_Toc91712648)

[Step N°2: remove\_character 3](#_Toc91712649)

[Step N°4: time format treatment 4](#_Toc91712650)

[Step N°5: row fusion according to “scientific\_title” or “title” column 4](#_Toc91712651)

[Step N°6: data replacement for “journal” column 5](#_Toc91712652)

[Step N°7: create new database based on individual word by the cleanded dataset obtained in the step N°6 5](#_Toc91712653)

[Step N°8: improve the database obtained from the step N°7 5](#_Toc91712654)

[Drugs quotes detection 6](#_Toc91712655)

# Files :

## Original files :

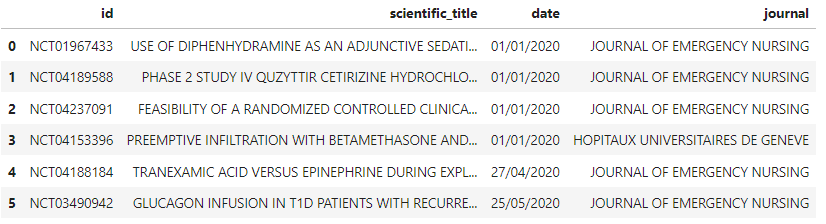
* 1.clinical\_trials.csv
* 1.drugs.csv
* 1.pubmed.csv
* 1.pubmed.json
* 1.test\_python.pdf

## Codes :

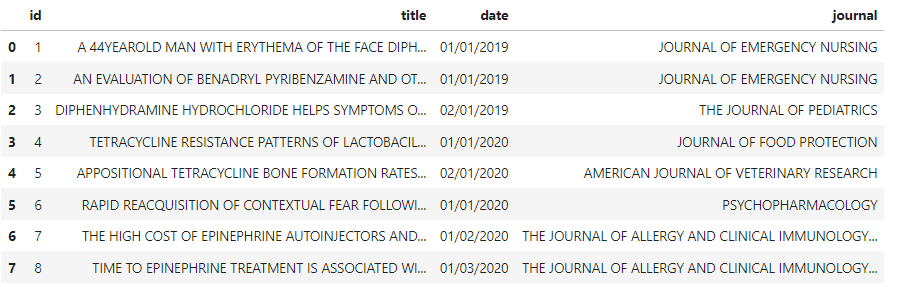
* pipeline.py : script executable
* conception.ipynb : conception of the pipeline.py

## Outputs:

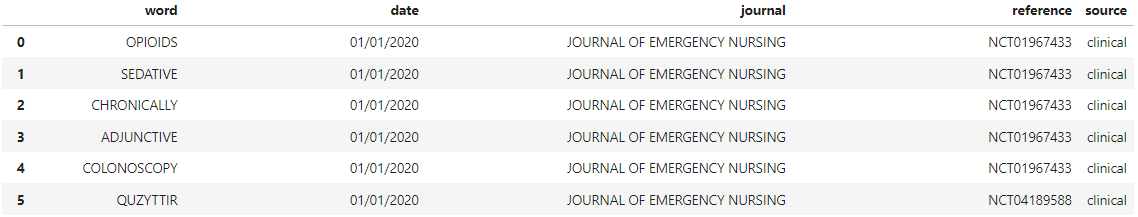
* cleaned\_data/clinical.csv: cleaned format of 1.clinical\_trials.csv



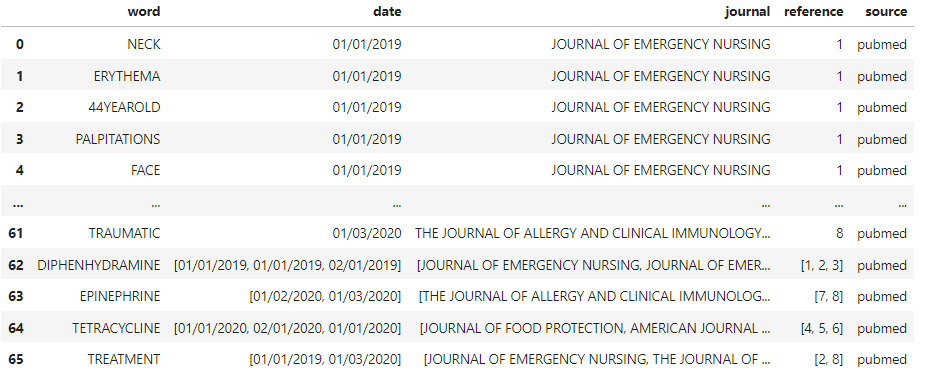
* cleaned\_data/pubmed.csv: cleaned format of 1. pubmed.csv



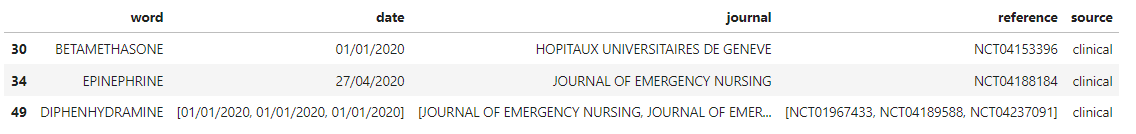
* cleaned\_data/drugs\_in\_clinical.csv: database of words in clinical.csv



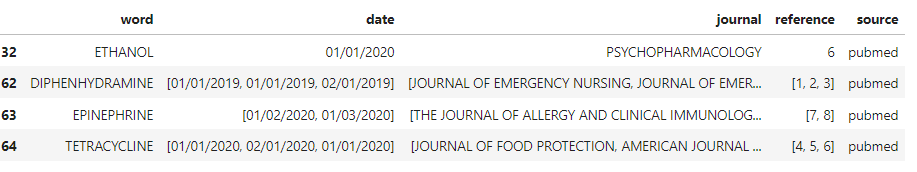
* cleaned\_data/drugs\_in\_pubmed.csv: database of words in pubmed.csv



* results/ search\_drugs\_inclinical.csv: drugs quoted by 1.drugs.csv in drugs\_in\_clinical.csv



* results/ search\_drugs\_inpubmed.csv: drugs quoted by 1.drugs.csv in drugs\_in\_pubmed.csv



# Explication conception

The data processing is composed of 8 steps and returns 2 dataframe (cleaned data + database : word – date – journal – reference – source)

## Step N°1: split\_text

* Standardize the string in upper and no accent format
* Slice string into word list

## Step N°2: remove\_character

* Remove all characters in each word if there are some
* Before:



* After:



### Step N°3: drop\_useless\_row

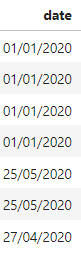
* It is supposed that if the “scientific\_title” or “title” is empty, there is no need to keep this row in dataset

## Step N°4: time format treatment

* Special case supplementary treatment: 1 January 2020 => 1 JANUARY 2020 => 1 01 2020
* Re-arrange all time data to one format in string: dd/mm/yyyy
* Before:

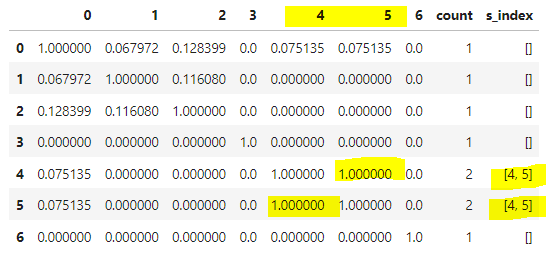


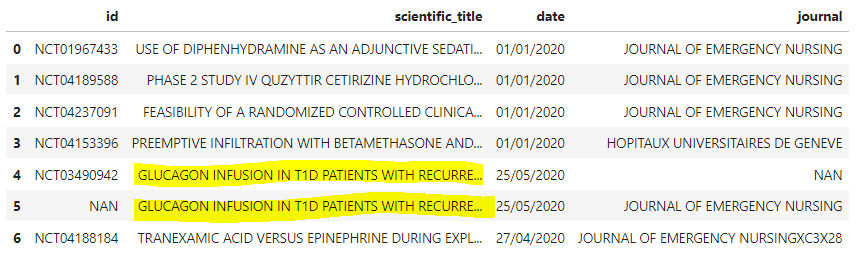
* After:



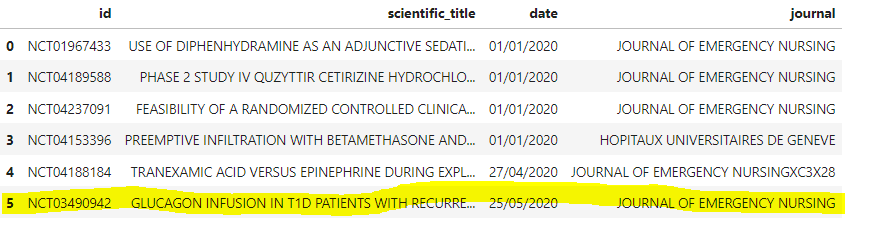
## Step N°5: row fusion according to “scientific\_title” or “title” column

* If several rows concern the same article title, it needs to fusion them into only one row
* Similarity between 2 strings calculated by TfidfVectorizer



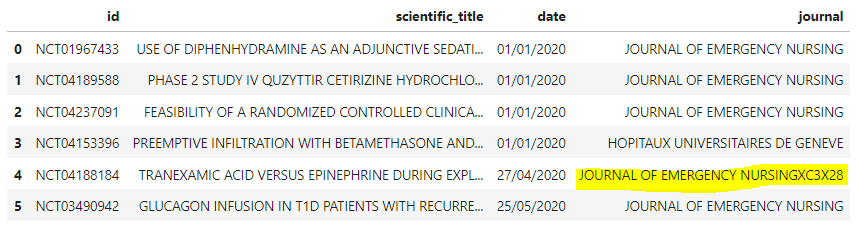


* If the similarity > 0.9: fusion



## Step N°6: data replacement for “journal” column

* Polluted data:



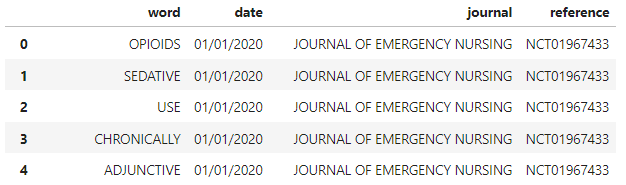
* Different than the step N°5 which is possible to merge different data, in this case, some wrong data has to been corrected
* The idea is firstly detecting the similar journal name by SequenceMatcher
* For a group of journal names who are considered similar, their names will be changed to the name of the majority:

example: ['Apple', 'Apple', 'Apple', 'Aapple', 'Apples'] => ['Apple'] x5

* Output: cleaned final dataset

## Step N°7: create new database based on individual word by the cleanded dataset obtained in the step N°6

* Firstly, remove all stop words: ‘OF’, ‘AN’, ‘A’, …
* Secondly: remove all “digital” words: ‘2’, ‘3’, …
* Create the database like following

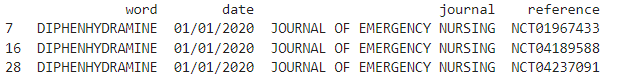


* Remove all duplicates

## Step N°8: improve the database obtained from the step N°7

* One word could be quoted at different time and in different journal, the aim of this step is to grouped them together
* Example:

Before: DIPHENHYDRAMINE quoted in 3 different articles (different references) at the same time in the same journal

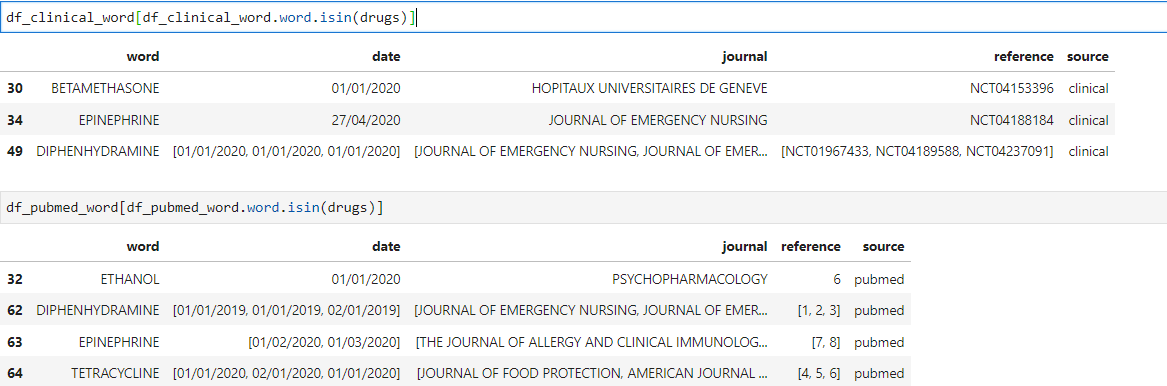


After: in one row



# Drugs quotes detection

Once the 2 databases corresponding to clinical\_trial and pubmed have been done, it needs a simple filtration to search the drug names’ quotes:



* Run the script pipeline.py
* Results in csv files could be founded in folder “results”

