Crawling and structuring Open Data

from http://www.data.gouv.fr/

Arthur Imbert, Inria, June 21, 2017

https://gitlab.inria.fr/parietal/arthur_imbert/

What is Open data?

"Open data is the idea that some data should be **freely available** to everyone to use and republish as they wish, **without restrictions** from copyright, patents or other mechanisms of control"

==> Technical restrictions!

What is Open data?

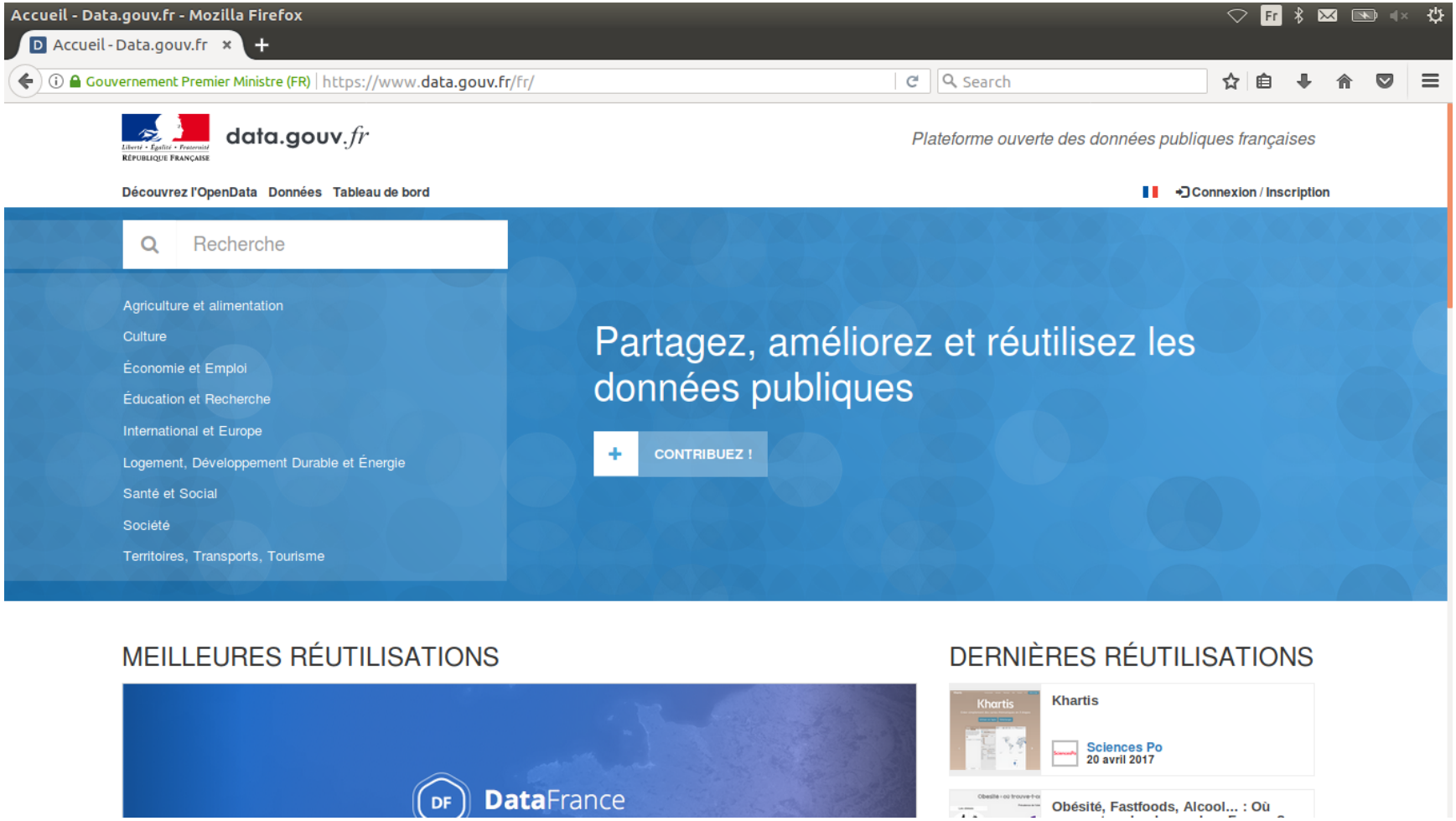
"Open data is the idea that some data should be **freely available** to everyone to use and republish as they wish, **without restrictions** from copyright, patents or other mechanisms of control"

==> Technical restrictions!

There are different levels of quality:

- data available on the web under an open license
- data available in a structured format
- data available in a non-proprietary open format
- data with Uniform Ressource Identifier
- data linked to another data

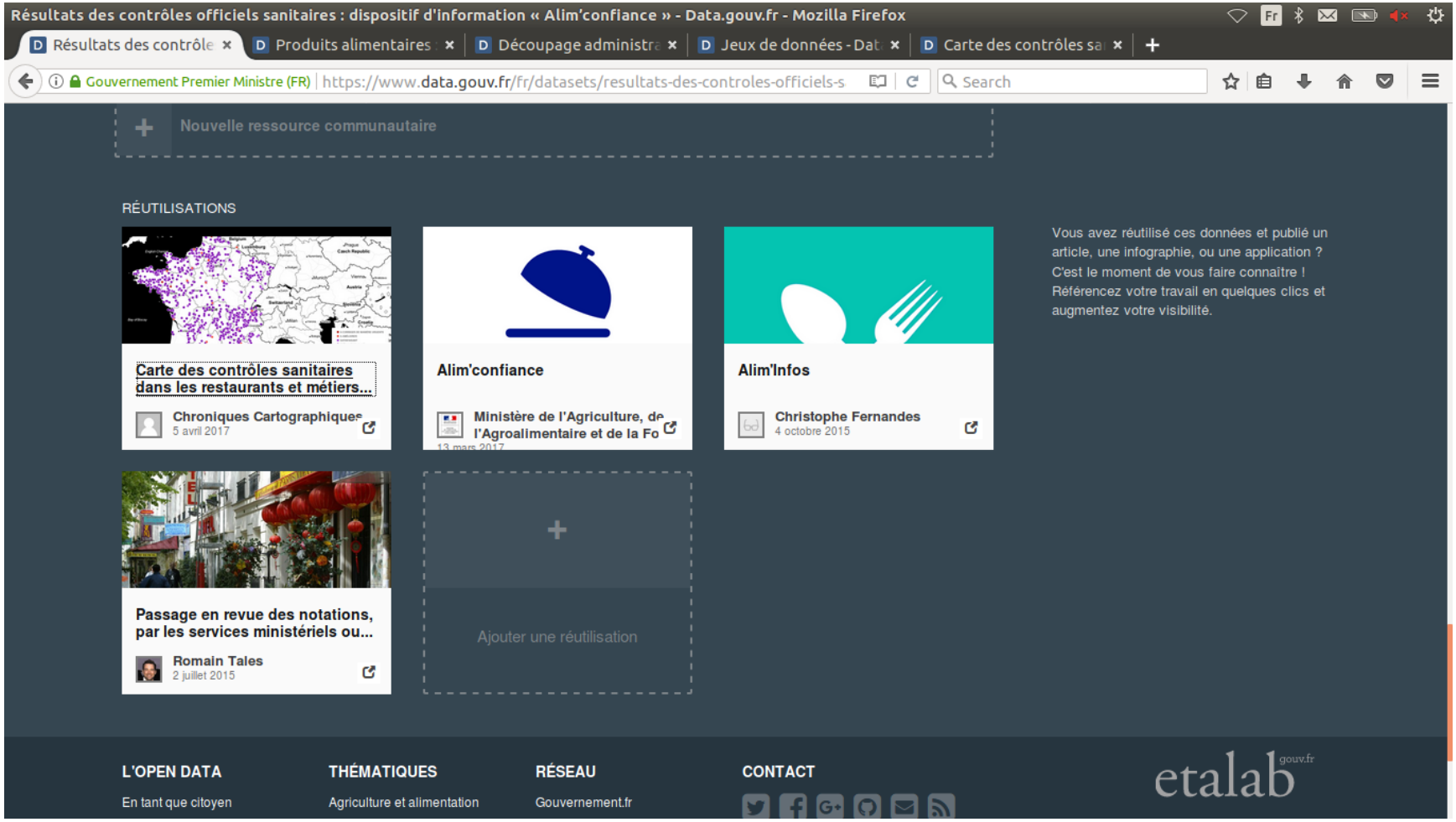
A government platform



https://www.data.gouv.fr/fr/

An example of dataset: food inspection reports

An example of reuse: Alim'confiance



- 25715 datasets

- 1260 organisations

- 1690 reuses

Collecting reliable data

Collecting data from data.gouv

There are dataset, producer, reuse catalogs... and an API!

Collecting data from data.gouv

There are dataset, producer, reuse catalogs... and an API!

With an http request, we can collect:

- the title of the page
- the name of the producer
- the creation date
- the granularity
- the frequency (time information)
- the tags
- the number of downloads
- a description
- an url to download the files

Collecting data from data.gouv

Metadata collected for 24617 pages

39405 files downloaded

269 Go of data

'Tablizing' everything

An example of file easy to clean

D	E	F	
ods_adresse	Code_postal	Libelle_commune	Sy
23 RUE DOMREMY	75013	Paris 13e Arrondissement	Sa
SAINT VICTOR SUR LOIRE	42230	Saint-Étienne	Sa
LD LE GRAND HAMEAU	14520	Sainte-Honorine-des-Pertes	Tr
8 Rue de Vaugirard	75006	Paris 6e Arrondissement	Sa
Rue de la Mécanique	27400	Louviers	Tr
SELE	20127	Bellegarde	Tr

- a tabular form
- a first row as header
- consistency of the data below the header

'Tablizing' everything

An example of file easy to clean

D	E	F	
ods_adresse	Code_postal	Libelle_commune	Sy
23 RUE DOMREMY	75013	Paris 13e Arrondissement	Sa
SAINT VICTOR SUR LOIRE	42230	Saint-Étienne	Sa
LD LE GRAND HAMEAU	14520	Sainte-Honorine-des-Pertes	Tr
8 Rue de Vaugirard	75006	Paris 6e Arrondissement	Sa
Rue de la Mécanique	27400	Louviers	Tr
SELE	20127	Bellegarde	Tr

- a tabular form
- a first row as header
- consistency of the data below the header

... still with some limitations

- no Uniforme Ressource Identifier for the adresses
- no Uniforme Ressource Identifier for the city names

'Tablizing' everything

An example of 'dirty data'

	A	B	C	D	E
	T79JNAIS : Répartition quotidienne des naissances vivantes				
	CHAMP : France métropolitaine, territoire au 31 décembre 2013				
		JOUR	01	02	03
	ANNÉE DE 1968 À 2013	*MOIS*			
2		Septembre	1,844	1,741	2,237
3		Octobre	2,393	2,471	2,327
4		Novembre	1,949	2,264	1,889
5		Décembre	1,948	1,768	2,197
6		Janvier	1,745	2,187	2,151
7		Février	2,173	1,879	1,778
8		Mars	2,222	1,872	1,712
9		Avril	1,741	2,129	2,167
0		Mai	1,890	2,281	2,291
1	2013	Juin	1,887	1,723	2,081
2		Juillet	2,298	2,337	2,370
3		Août	2,441	2,426	2,008
4		Septembre	1,818	2,138	2,327
5		Octobre	2,428	2,385	2,333
6		Novembre	1,830	1,834	1,868
7		Décembre	1,871	2,220	2,284
8					
9	Source : Insee, statistiques de l'état civil				
0					

'Tablizing' everything

Cleaning steps

- Is it a zipfile?
- Encoding and extension detections (`chardet` and `magic` libraries)
- Is it a json? A geojson?
- Different extensions (csv, pdf, json, etc.), different strategies

CSV

- Detect the delimiter from a sample (`from csv import Sniffer`)
- Detect the header

CSV

- Detect the delimiter from a sample (`from csv import Sniffer`)
- Detect the header

```
# we test if the first row could be a header
def is_header(file):
    # we test the consistency of the types over the rows

# we test the first rows of the file
for row in file:
    is_header(row)
```

Excel

	A	B	C	D	E
	T79JNAIS : Répartition quotidienne des naissances vivantes				
	CHAMP : France métropolitaine, territoire au 31 décembre 2013				
		JOUR	01	02	03
	ANNÉE DE 1968 À 2013	*MOIS*			
2		Septembre	1,844	1,741	2,237
3		Octobre	2,393	2,471	2,327
4		Novembre	1,949	2,264	1,889
5		Décembre	1,948	1,768	2,197
6	2013	Janvier	1,745	2,187	2,151
7		Février	2,173	1,879	1,778
8		Mars	2,222	1,872	1,712
9		Avril	1,741	2,129	2,167
0		Mai	1,890	2,281	2,291
1		Juin	1,887	1,723	2,081
2		Juillet	2,298	2,337	2,370
3		Août	2,441	2,426	2,008
4		Septembre	1,818	2,138	2,327
5		Octobre	2,428	2,385	2,333
6		Novembre	1,830	1,834	1,868
7		Décembre	1,871	2,220	2,284
8					
9	Source : Insee, statistiques de l'état civil				
0					

Excel

- Detect the number of columns
- Fill in the merged cells
- Detect a multiheader

Json

- Explore the json (recursive function)
- Flatten the json (`from pandas.io.json import json_normalize`)

Json

- Explore the json (recursive function)
- Flatten the json (`from pandas.io.json import json_normalize`)

```
# we explore the json until we find a good structure to flatten
def recursive(json):
    if ...
        # we test if the json is a list of dictionary
    else:
        # we keep searching that structure deeper in the json

# we flatten the json
right_structure = recursive(json)
df = json_normalize(right_structure)
```

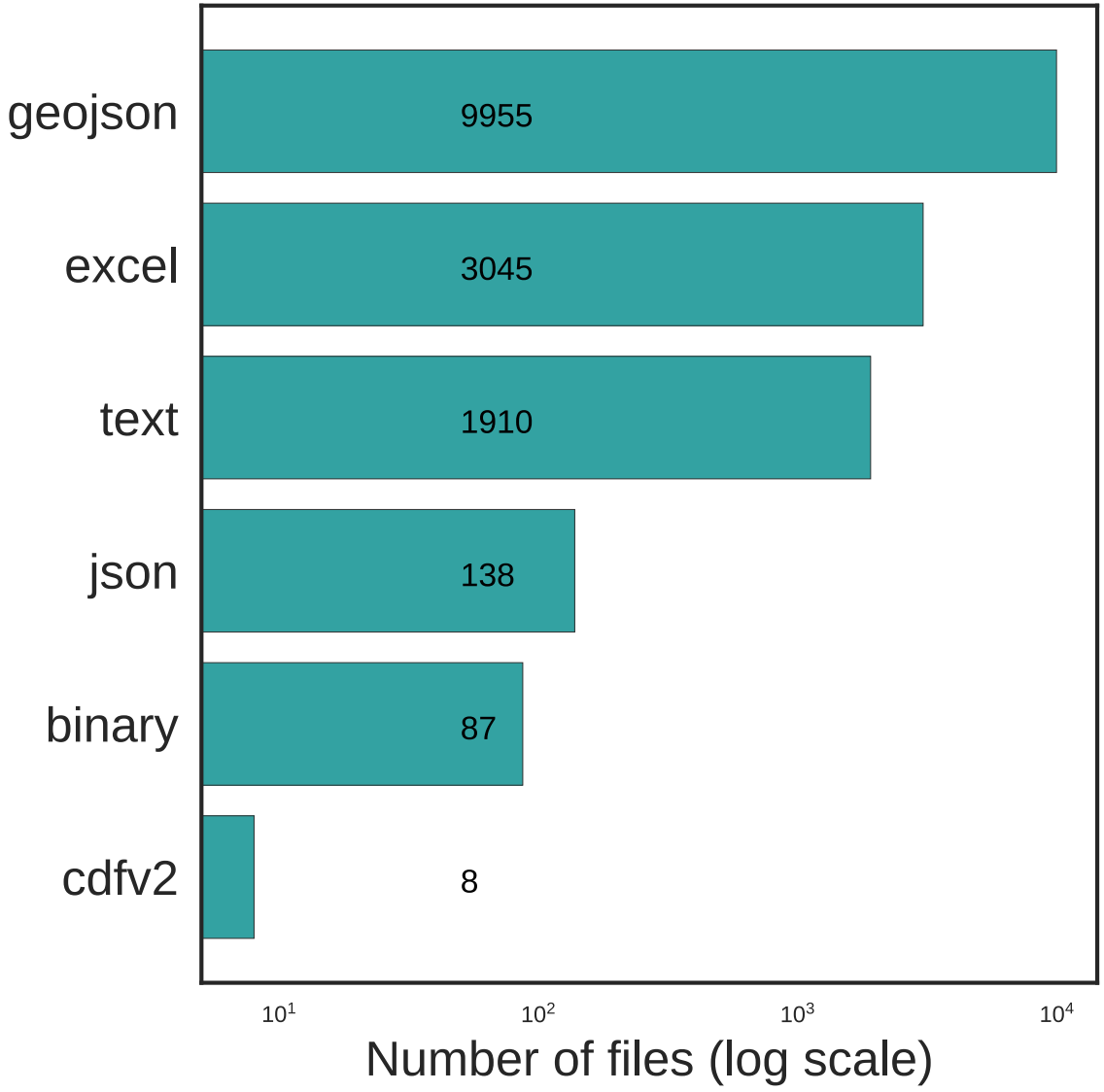
Json

- Explore the json (recursive function)
- Flatten the json (`from pandas.io.json import json_normalize`)

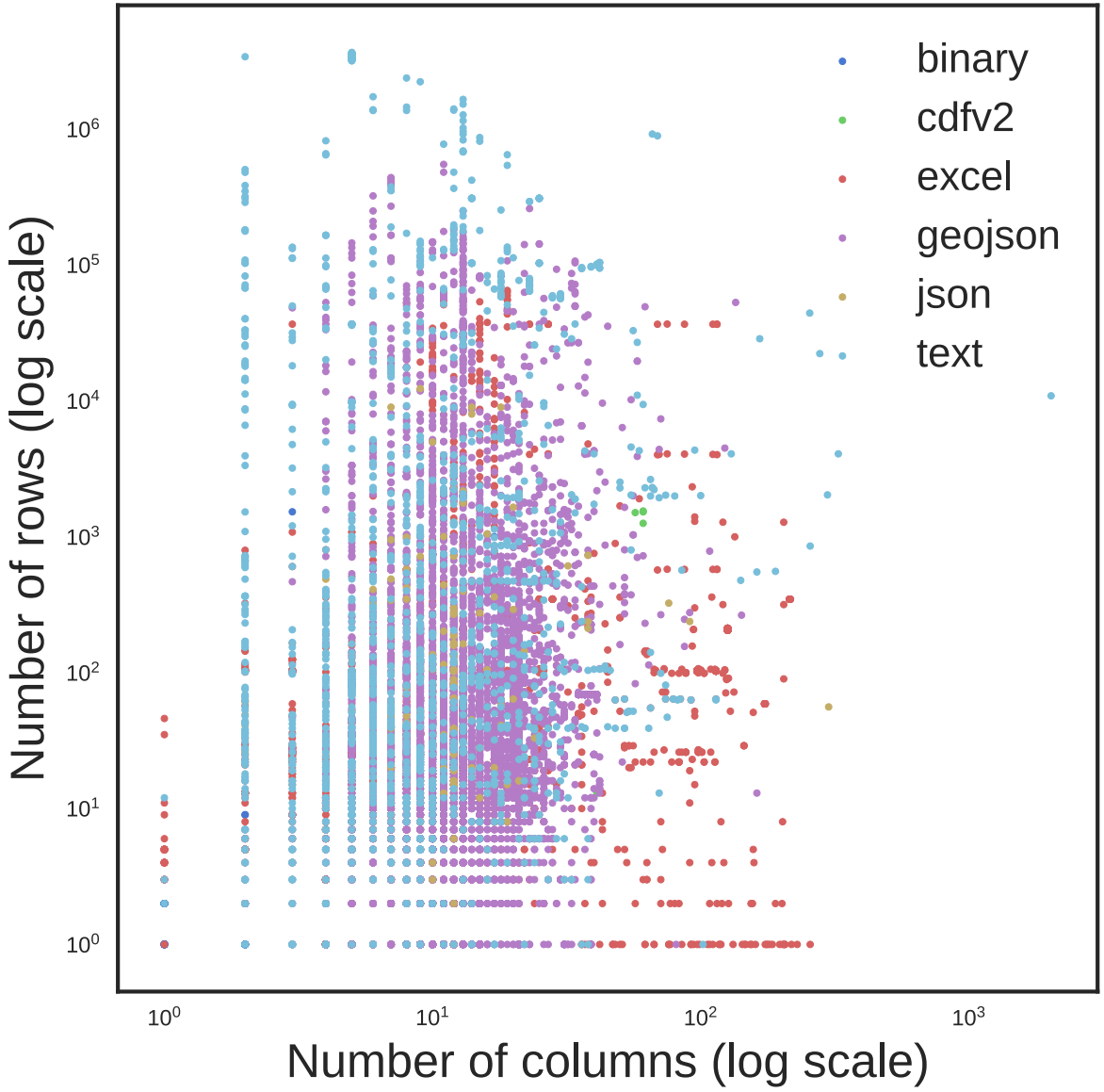
```
# we explore the json until we find a good structure to flatten
def recursive(json):
    if ...
        # we test if the json is a list of dictionary
    else:
        # we keep searching that structure deeper in the json

# we flatten the json
right_structure = recursive(json)
df = json_normalize(right_structure)
```

- What about XML?

Results


geojson
9955
excel
3045
text
1910
json
138
binary
87
cdfv2
8
10^1
10^2
10^3
10^4
Number of files (log scale)

Results


binary
cdfv2
excel
geojson
json
text
Number of rows (log scale)
Number of columns (log scale)
10^0
10^1
10^2
10^3
10^4
10^5
10^6

Learning semantic structure

Preprocessing

- Tokenization

Count words which contain characters only

=> Count_matrix [n_files, n_words]

Preprocessing

- Tokenization
- Normalization

Weight each file to compensate for varying file sizes

Preprocessing

- Tokenization
- Normalization
- Content, header and metadata

$$\text{total} = 0.5\ \text{content} + 0.25\ \text{header} + 0.25\ \text{metadata}$$

Preprocessing

- Tokenization
- Normalization
- Content, header and metadata
- Stemming and unstemming

Reduce each word to its root form:

continuer -> continu
continuant -> continu

Preprocessing

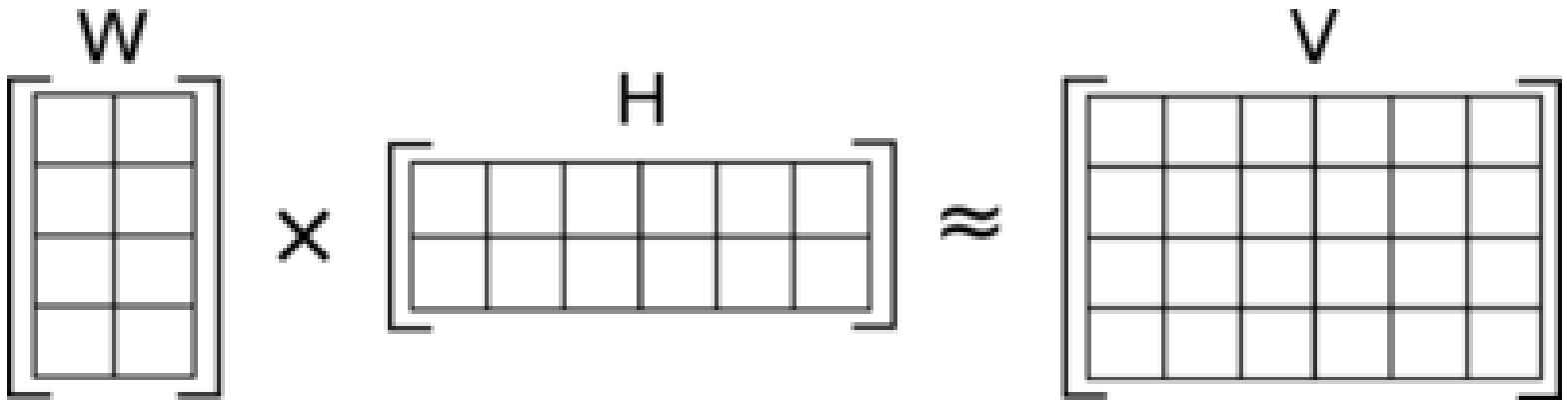
- Tokenization
- Normalization
- Content, header and metadata
- Stemming and unstemming
- TFIDF

Weight each word frequency by its inverse document frequency

=> A word relatively frequent in a specific file will be discriminant

Topic modeling

NMF



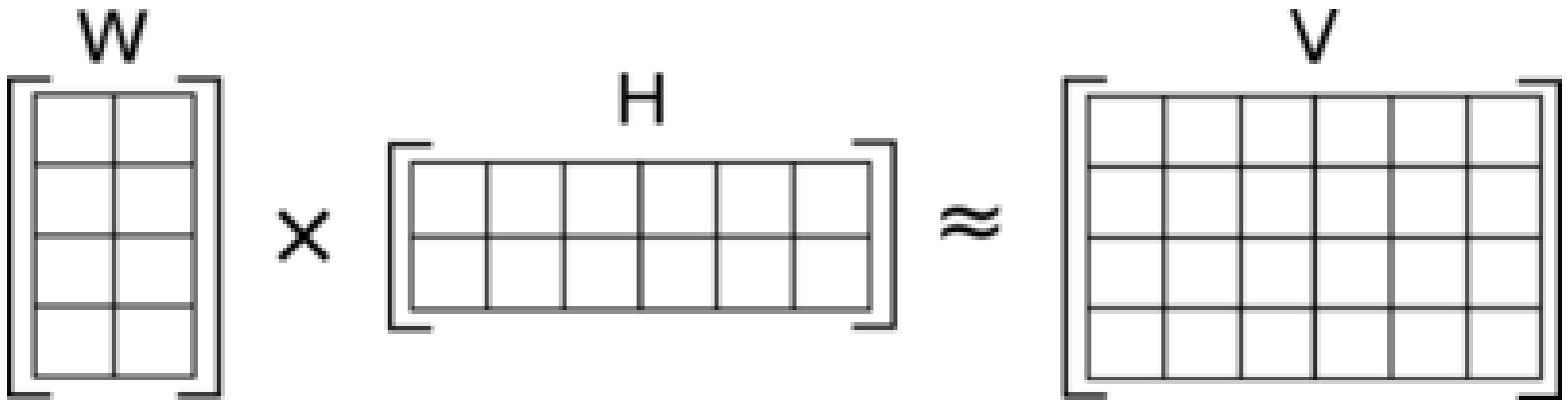
V is a TFIDF matrix [n_files, n_words]

W is a matrix [n_files, n_topics] => files embedding

H is a matrix [n_topics, n_words] => wordcloud per topic

Topic modeling

NMF



V is a TFIDF matrix [n_files, n_words]

W is a matrix [n_files, n_topics] => files embedding

H is a matrix [n_topics, n_words] => wordcloud per topic

n_topics = 20

A reliable embedding ?

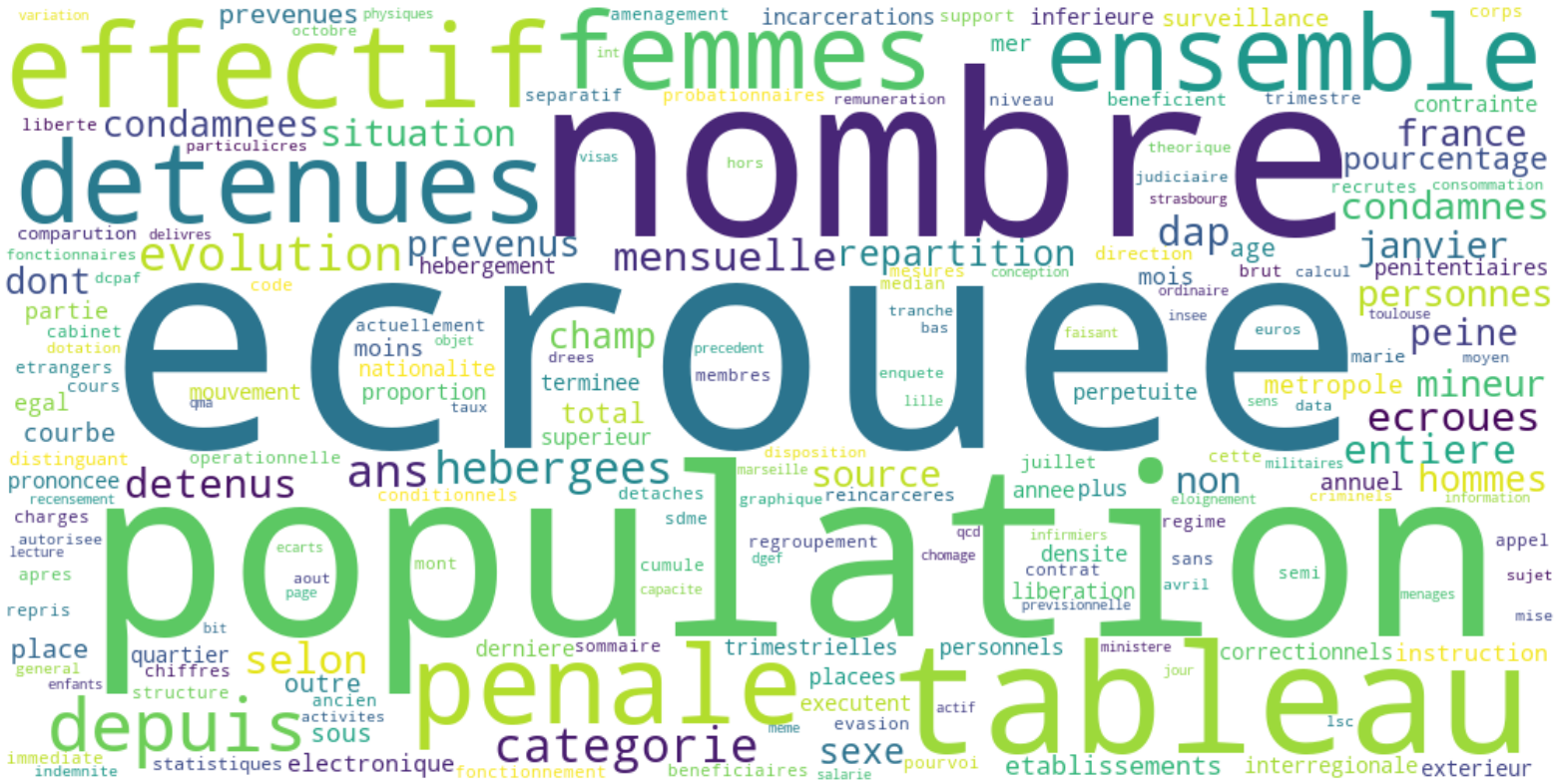
A reliable embedding ?

Tags : faits_constates, police, criminalite

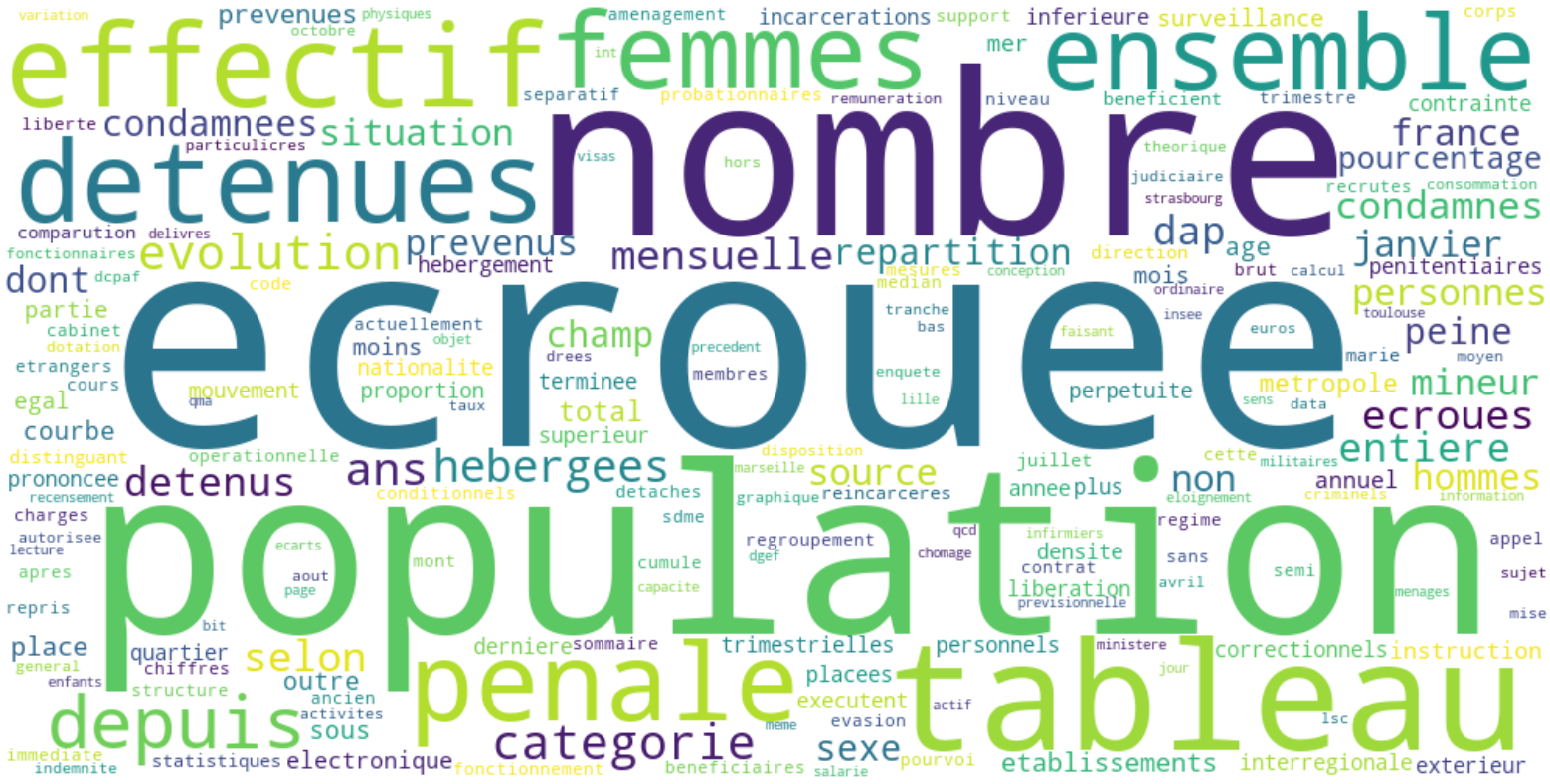
Producer : Observatoire national de la délinquance et des réponses pénales, Ministère de la Justice, Ministère de l'Intérieur

Extension : excel, geojson, text

A reliable embedding ?



A reliable embedding ?

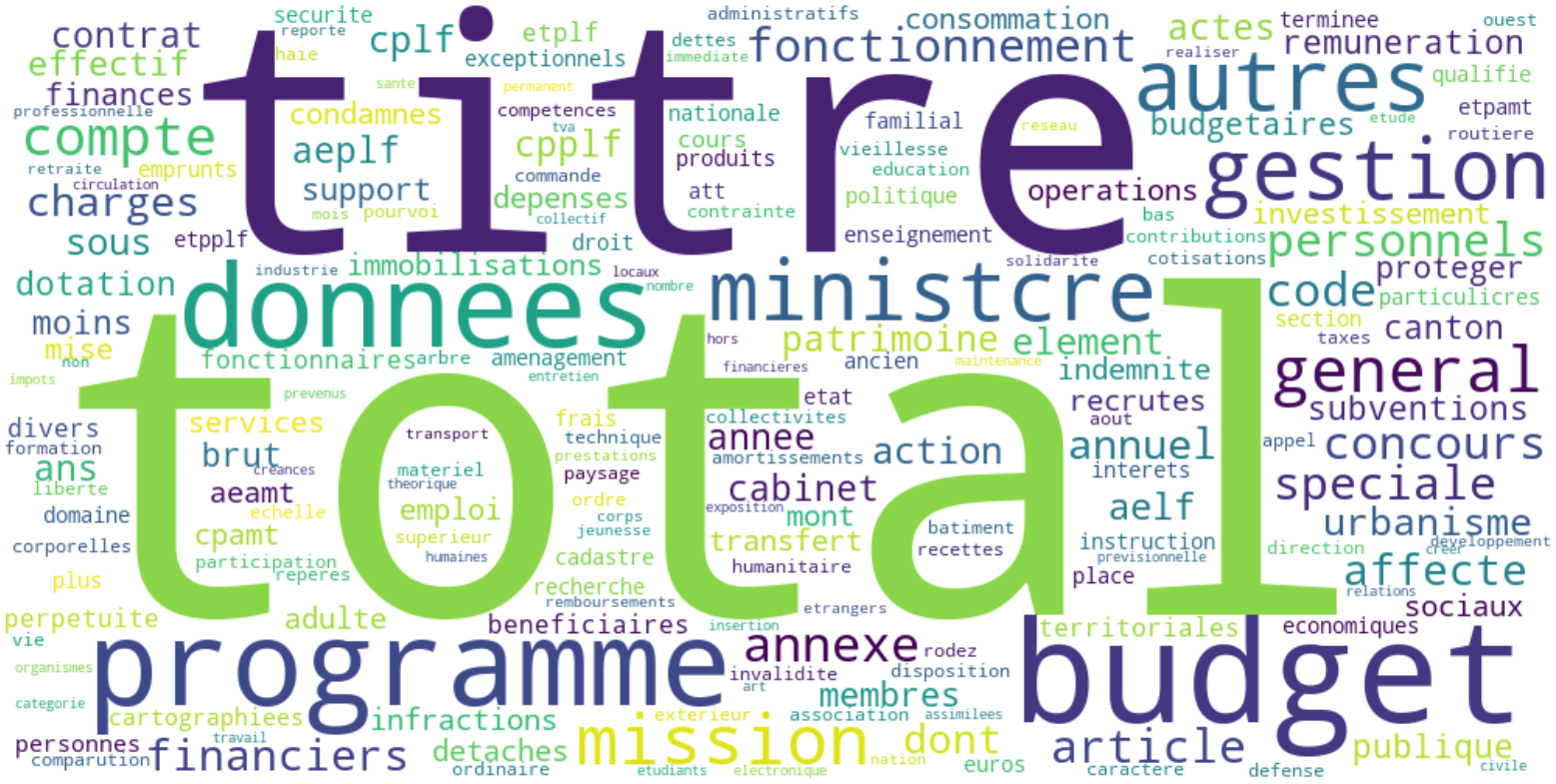


Tags : amenagements_de_peine, ecroues, immigration

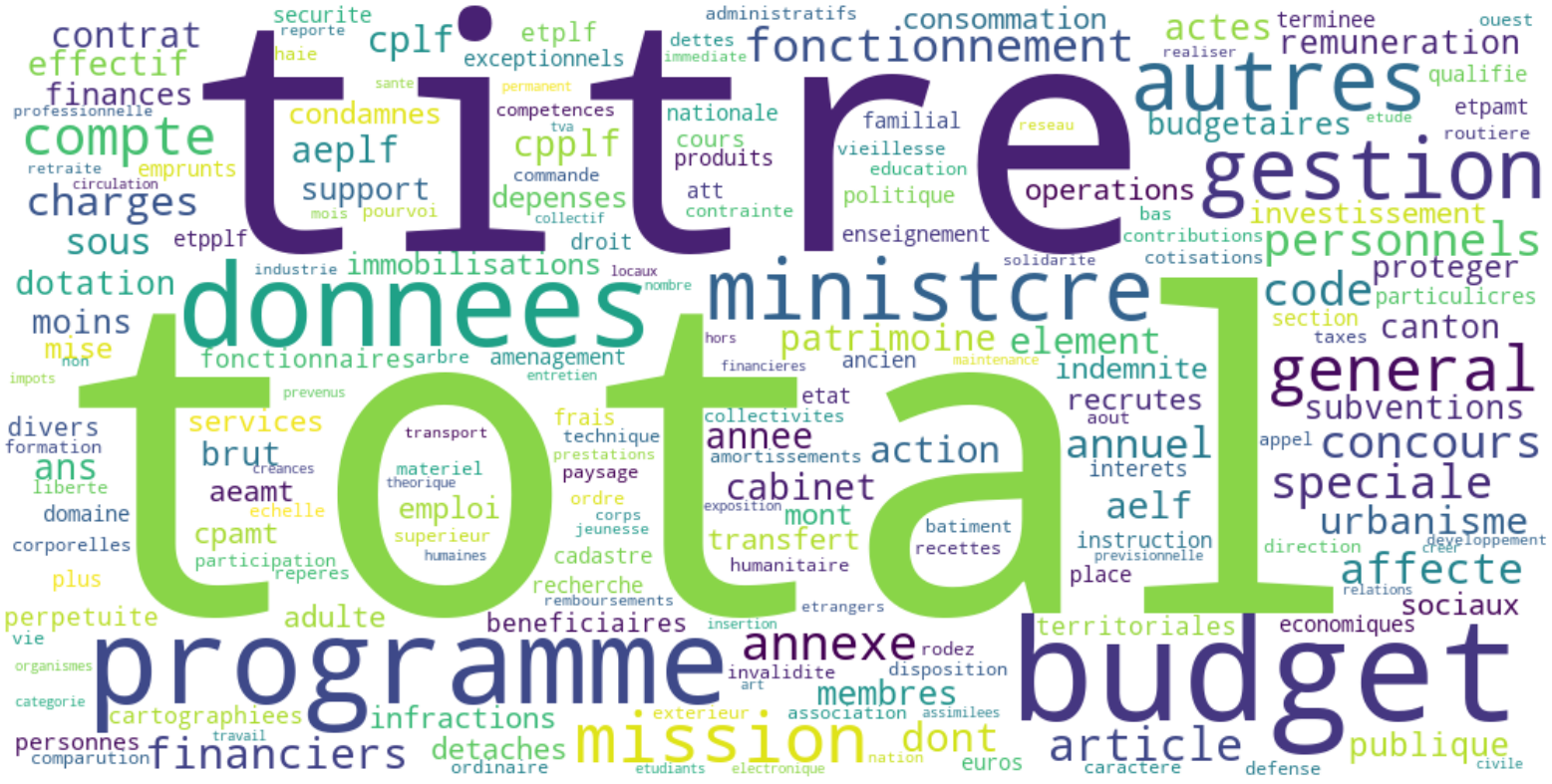
Producer : Ministère de la Justice, Ministère des finances et des comptes publics, Ministère de l'Intérieur

Extension : excel, text, geojson

A reliable embedding ?



A reliable embedding ?

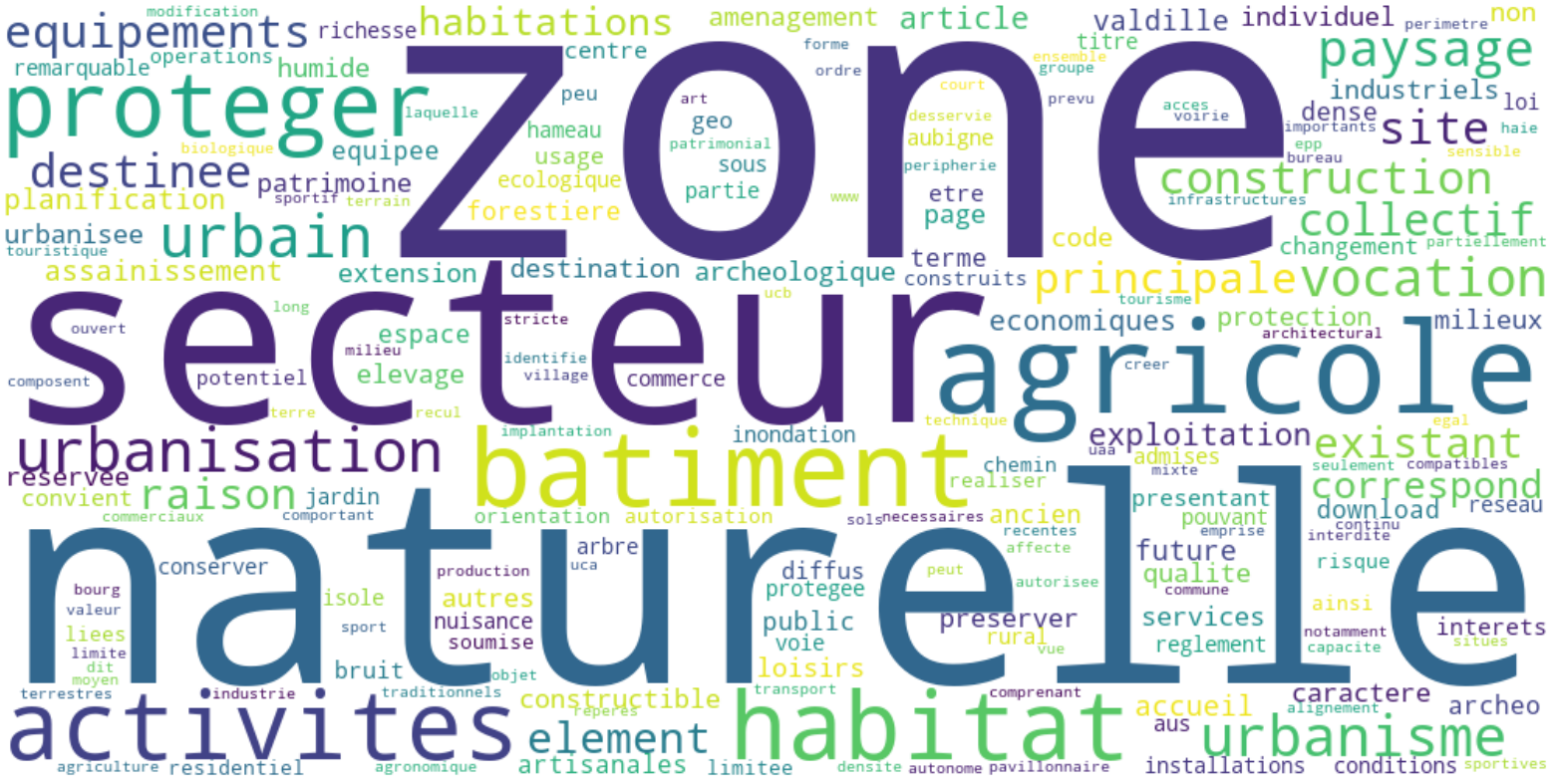


Tags : finances_publiques, planning_cadastre, usage_des_sols

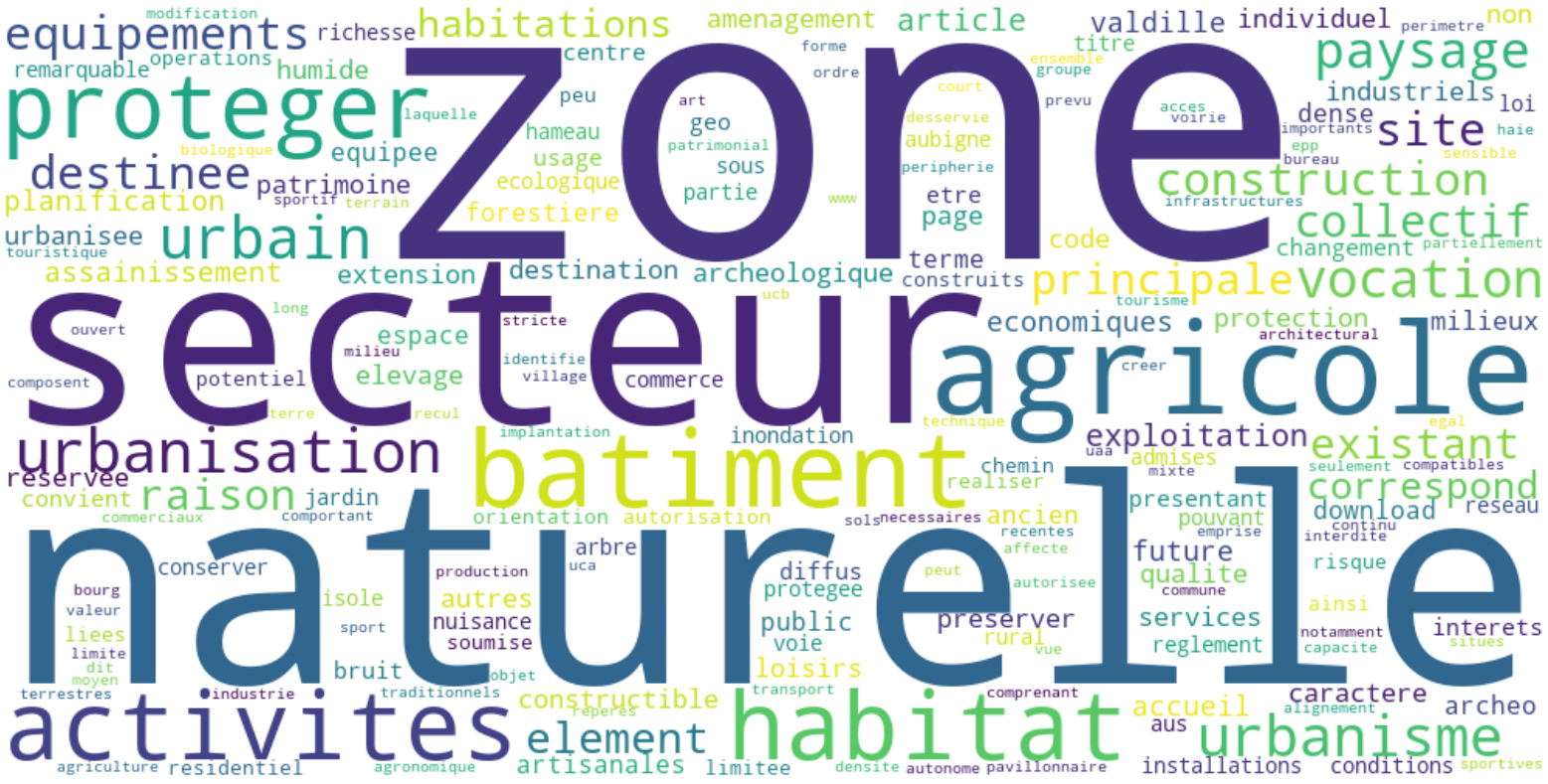
Producer : Ministère des finances et des comptes publics, Ministère de l'Intérieur, Ministère de la Justice

Extension : text, excel, geojson

A reliable embedding ?



A reliable embedding ?



Tags : planning_cadastre, usage_des_sols, plu

Producer : Direction Départementale des Territoires de *

Extension : geojson, excel, text

What are the next steps ?

- Keep working on the embedding
- Find new ways to visualize the relevance of our embedding
- Explore new kind of relations between the files (mutual geographic space, datetime)


KEEP
CALM
AND
OPEN
DATA