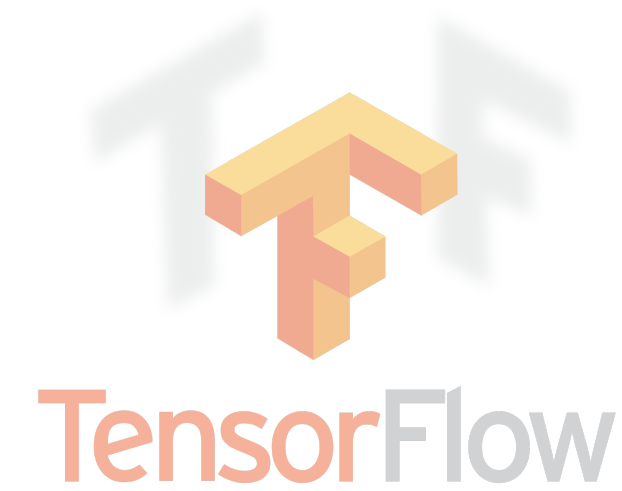


DOCKER

TEXT CLEANING & VECTORISATION

GOOGLE CLOUD PLATFORM

ML Python Libraries



Python Environment



Viz' Python Libraries



seaborn



Framework & Tool



TABLE OF CONTENTS

INTRODUCTION

TEXT CLEANING

TEXT VECTORISATION

INTRODUCTION

TEXT USE CASE IN ARTIFICIAL INTELLIGENCE

There are multiple applications of artificial intelligence on text data:

- **INFORMATION RETRIEVAL**: on text or content-based (*Google, Yahoo etc.*)
- **PATTERN RECOGNITION**: Information/Named extraction.
- **SENTIMENT ANALYSIS**: Marketing. Website comments.
- **TEXT GENERATION**: Chatbot. Newspaper article.
- **TEXT TRANSLATION**: Google Translate. DeepL.
- **DISAMBIGUATION**: Security.
- And many others...

Text processing does not always mean **Natural Language Processing (NLP)**

EXAMPLE: TEXT CLASSIFICATION

OBJECTIVE: Automate the categorisation of text product within discount website. Data come from datascience contest [website](#).

DIFFICULTIES:

- Text data require **processing** to used machine learning model on it.
- **Big** amount of data (15M of text description).
- Highly **unbalanced** classes.
- High number of classes (more than 5000).
- Real dataset that requires a lot of **cleaning**.

Train file contains 15.786.885 products. Answer of test file not furnished.

Three levels classification:

- 47 categories of level 1.
- 536 categories of level 2.
- 5789 categories of level 3.

Field	Type	Description
product id	String	Unique identifiant du produit
Catégorie 1	String	Catégorie de niveau 1
Catégorie 2	String	Catégorie de niveau 2
Catégorie 3	String	Catégorie de niveau 3
Description	String	Description produit
Libelle	String	Description courte
Marque	String	Marque du produit

DATA EXAMPLE

Categorie1	
ANIMALERIE - NEW	Lit Mijou, 48 × 37 Pouces, Crème - -imitation ...
ARME DE COMBAT - ARME DE SPORT	Réplique chargeur STI DUTY ONE (CPG1945) - Rép...
ART DE LA TABLE - ARTICLES CULINAIRES	Mugs Alchemy (king 13) (Taille unique) - Mugs...
ARTICLES POUR FUMEUR	E-PACK FRUITÉ 'EXPERT' (Titanium bleu - Mixte ...
AUTO - MOTO (NEW)	Tube de fourche Tarozzi KYMCO X-CITING 500 - 0...
BAGAGERIE	portefeuille porte cartes billets compagnon fe...
BATEAU MOTEUR - VOILIER	Echelle pour plateforme70086 - Fabrication i...
BIJOUX - LUNETTES - MONTRES	Seiko SFP599 Hommes Montre - Acheter Authentiq...
BRICOLAGE - OUTILLAGE - QUINCAILLERIE	Clé polygonale double contre-coudée - 20x22 - ...
CHAUSSURES - ACCESSOIRES	Bottes bi-matière à talons bleu - Zaza Pata -...
CONDITIONNEMENT	EMBALLAGE Ruban adhésif d'emballage PVC colle ...
CULTURE / JEUX	De Keenen Ivory Wayans avec Shannon Elizabeth,...
DECO - LINGE - LUMINAIRE	Cars Poster Reproduction Sur Toile, Tendue Sur...
DROGUERIE (NEW)	PERCHE TELESCOPIQUE SECURITY LOCK 3X2M - PERCH...
ELECTROMENAGER	Filtre metal antigraisse (x1) AD546BE11 AD546W...
ELECTRONIQUE	Bloc de jonction à fusible Contenu: 20 pc(s) p...
EPICERIE	Sel de Guérande aux épices, Verrine 150 gr - S...
HYGIENE - BEAUTE - PARFUM	Uriage AquaPRÉCIS Crème Confort 40 ml - Les mi...
INFORMATIQUE	Batterie Acer Aspire One 751H-52Yr - Li-Ion 11...

DIFFICULTIES

- **Noise** linked to spelling, grammar, conjugaison mistake.
- **Non significant** terms.
- Mining of terms depends of **context**. (*Clothes / Dolls' clothes*)
- Different from one language to another.
- Transcription to machine learning.

Implies a lot of **cleaning**.

TEXT CLEANING

STEMMING

PROBLEM: Term can be written in different way (accentuation, genre, conjugaison, plurals, etc.) but still have the same meaning.

SOLUTION: replace word with their stems.

EXAMPLE:

- Épée, épée, épées, épée = epe
- vert, verts, vertes, vertes = vert
- mange, manger, mangez, mangent, = mang

Algorithm that generate stemming from words are **rules-based** and depends of the **language**.

The one used on nltk for French language is the [Snowball algorithm](#).

STOPWORDS

PROBLEM: Terms that are very common does not help to classify data and can even disturb the training.

SOLUTION: Most common words are removed. This words are called **stopword**.

EXAMPLE:

- **FRENCH:** 'au', 'aux', 'avec', 'ce', 'ces', 'dans', 'de', 'des', 'du', 'elle', 'en', 'et', 'eux', 'il', 'ils', 'je', 'la', 'le', 'les', 'leur', 'lui', 'ma', 'mais', 'me', 'même', 'mes', 'moi', 'mon', etc.
- **ENGLISH:** ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', etc.

OTHER CLEAN STEPS

- Remove **punctuation**, **number** or other **non-letter** symbol.
- Increment stopwords list with **domain** words.
- Removed technical noise (HTML code).
- Lower case.

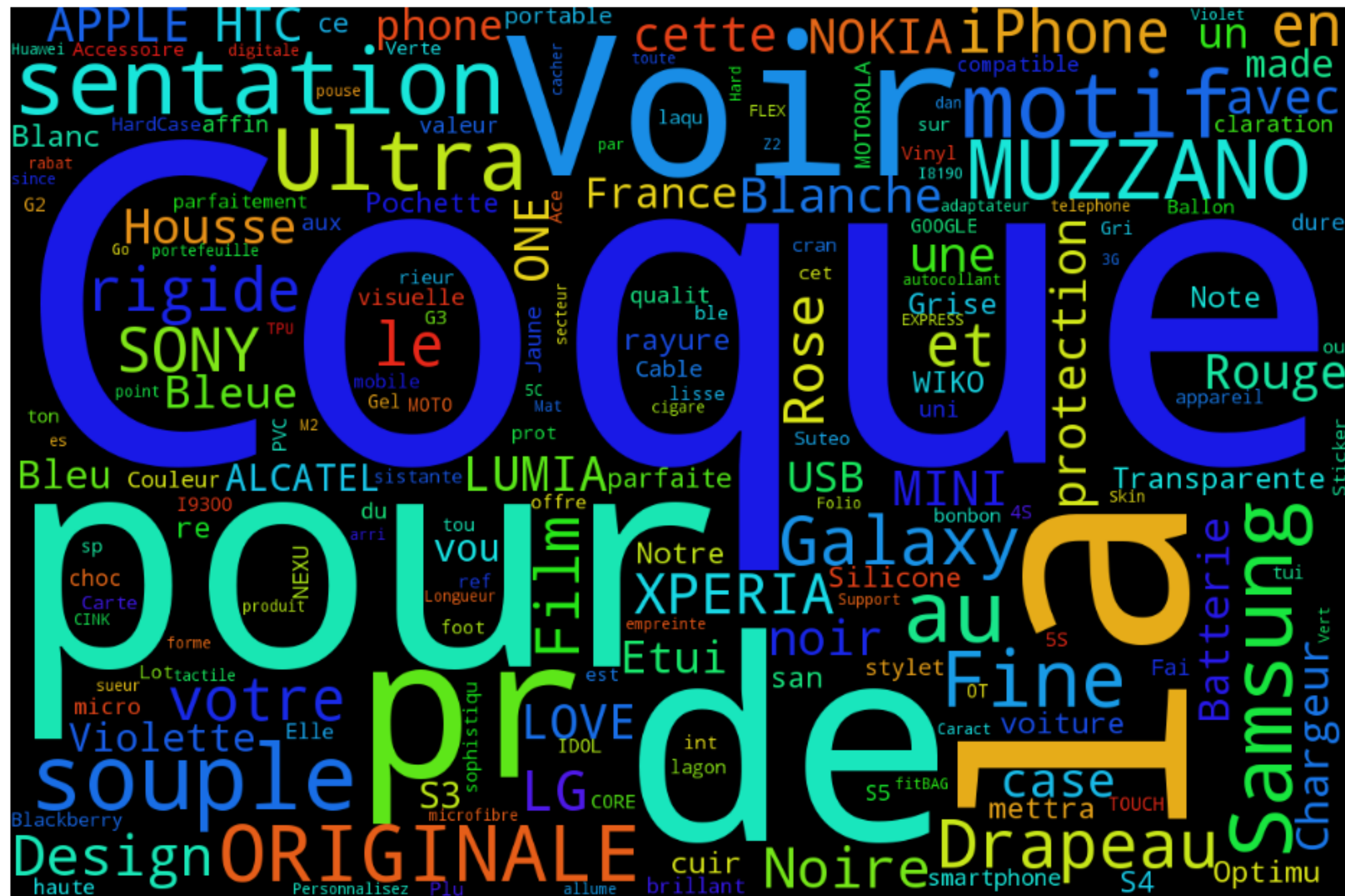
Most of these steps depend of the objectives you want to achieve.

EXAMPLE:

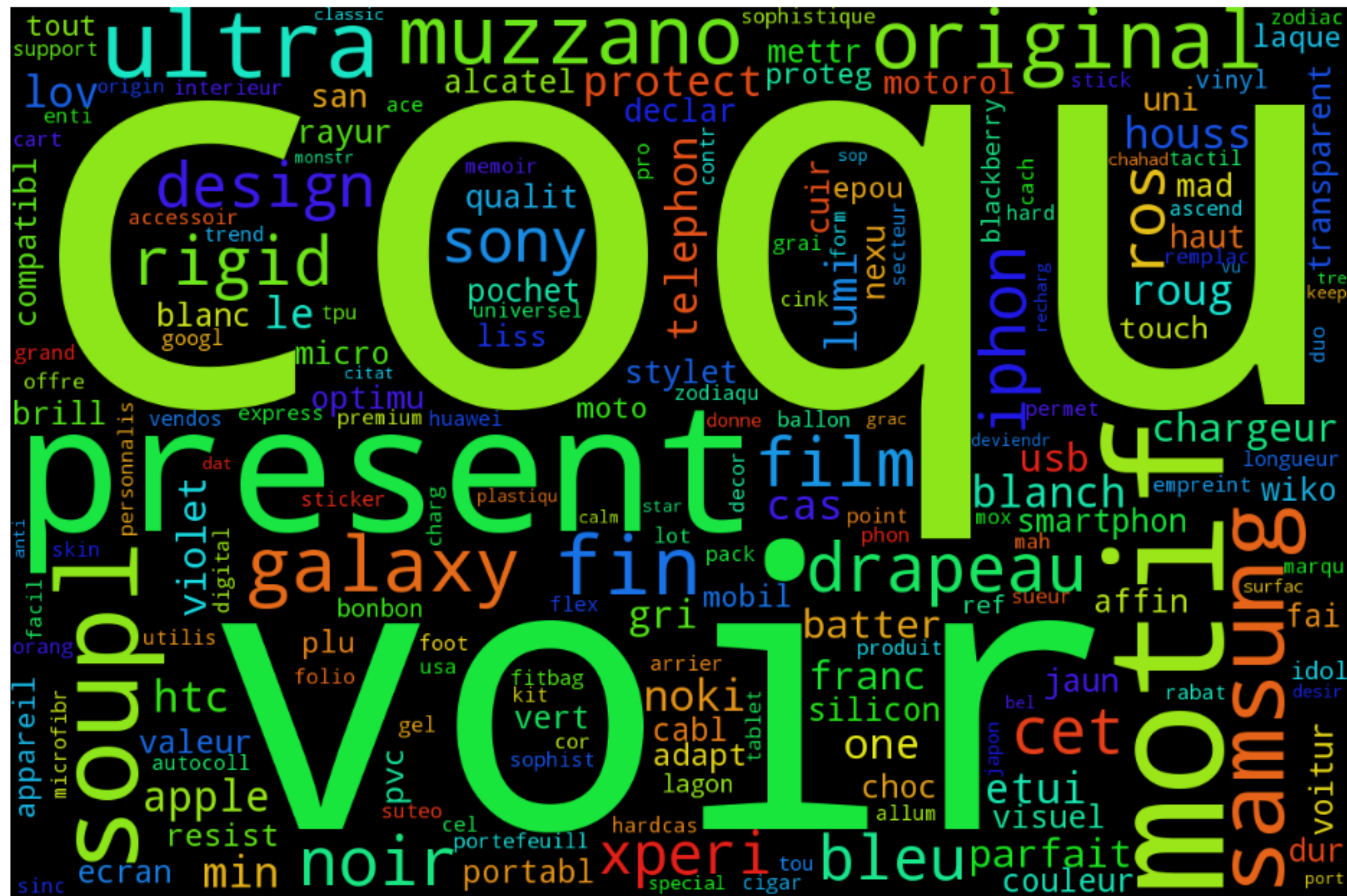
- Upper case can be kept for sentiment analysis.
- Cdiscount: Number can be removed for categorie 1's level and not categories 2 (xbox 360).
- etc.

REGULAR EXPRESSION

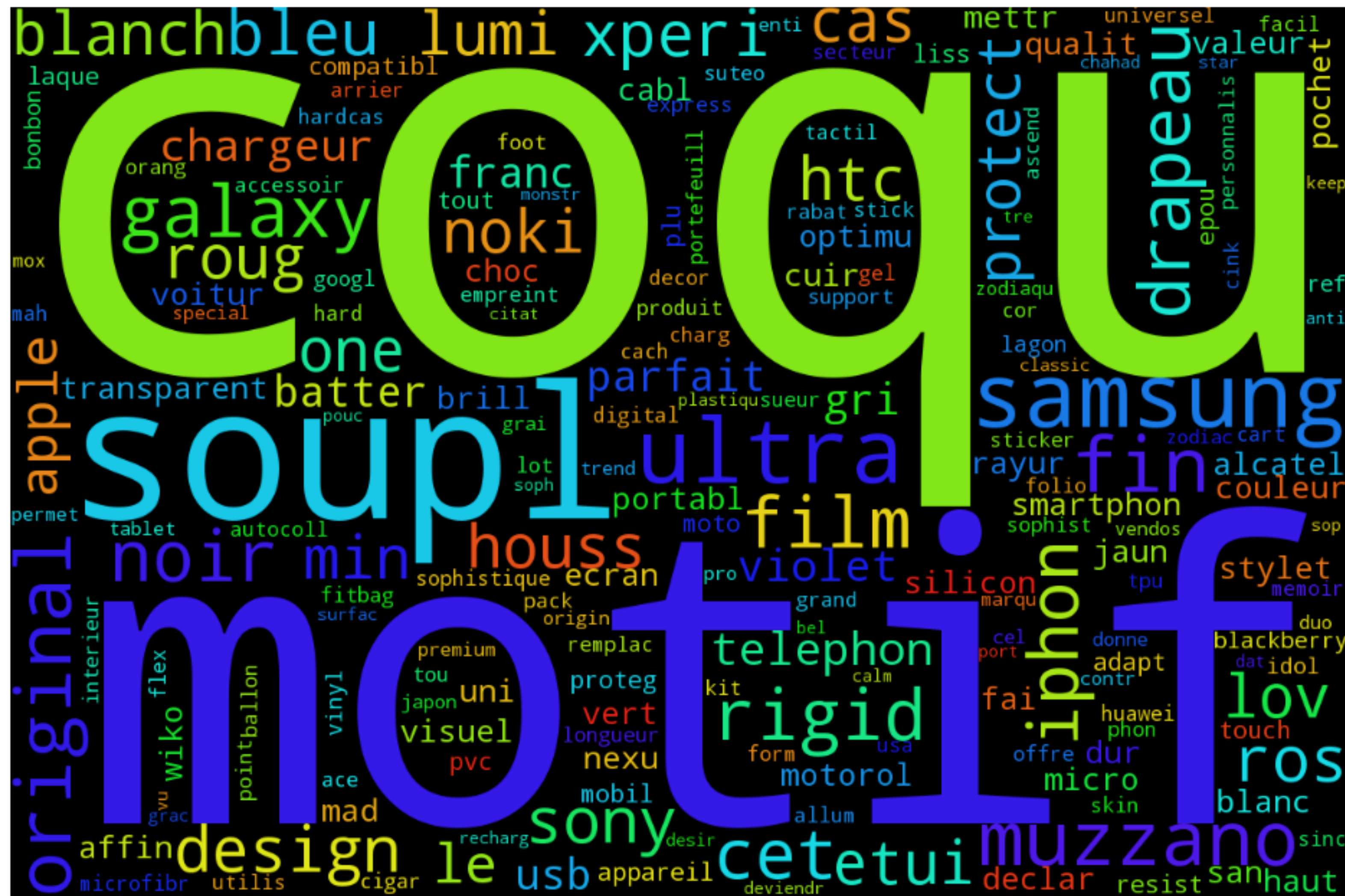
STOPWORD - CATEGORY - "TÉLÉPHONIE - GPS"



STOPWORD - CATEGORY - “TÉLÉPHONIE - GPS”



STOPWORD - CATEGORY - "TÉLÉPHONIE - GPS"



LIBRARIES FOR TEXT PROCESSING

- **NLTK** (*python*): language processing (stemming, stop words)
- **Lucene** (*java*): text indexation and information retrieval
- **BeautifulSoup**: clean html text.

VECTORISATION

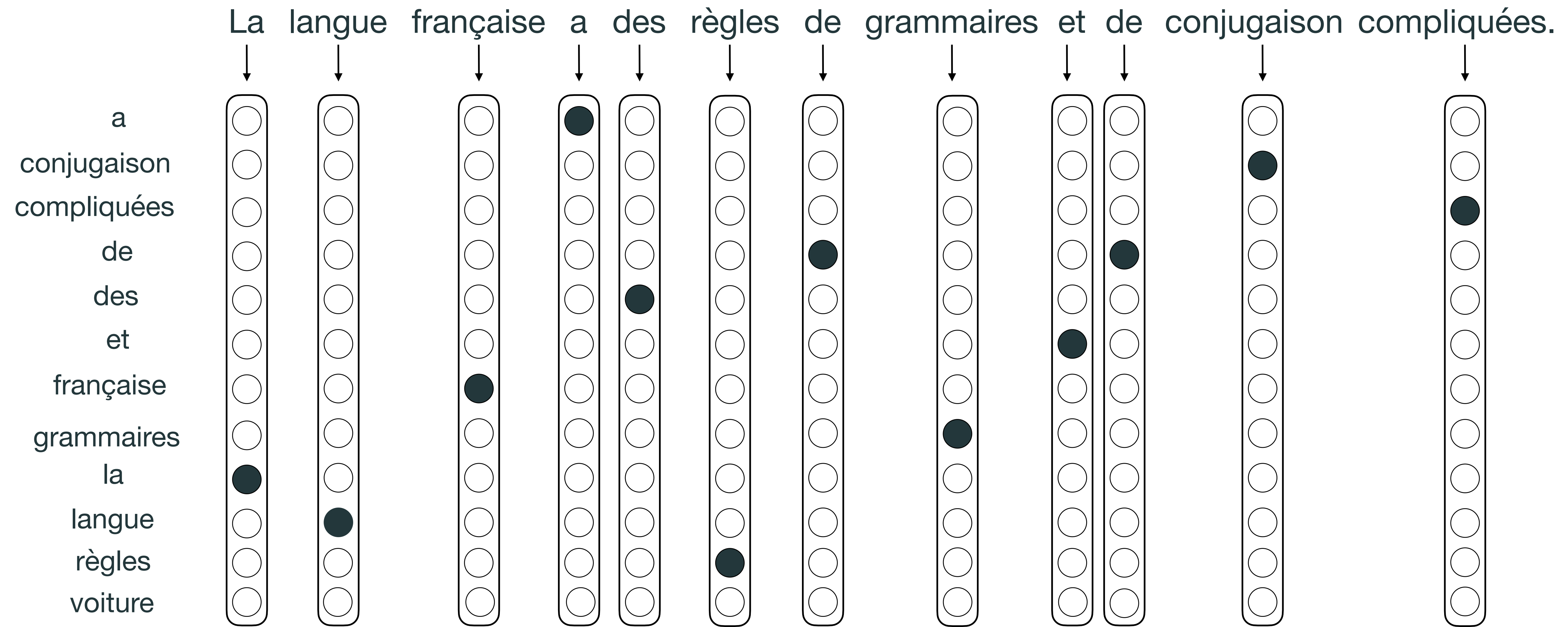
OBJECTIVES AND DIFFICULTIES

- Transform text to numerical data to be used in AI algorithms.
- Manage high number of features. Example:
 - 21.543 lines on category “TELEPHONIE - GPS”
 - 24.486 unique words -> 8384 after cleaning.
- Choose significant words

Two types of solutions:

- Frequency based : **Vectorizer**
- Learning based : **Word Embedding** (See corresponding course).

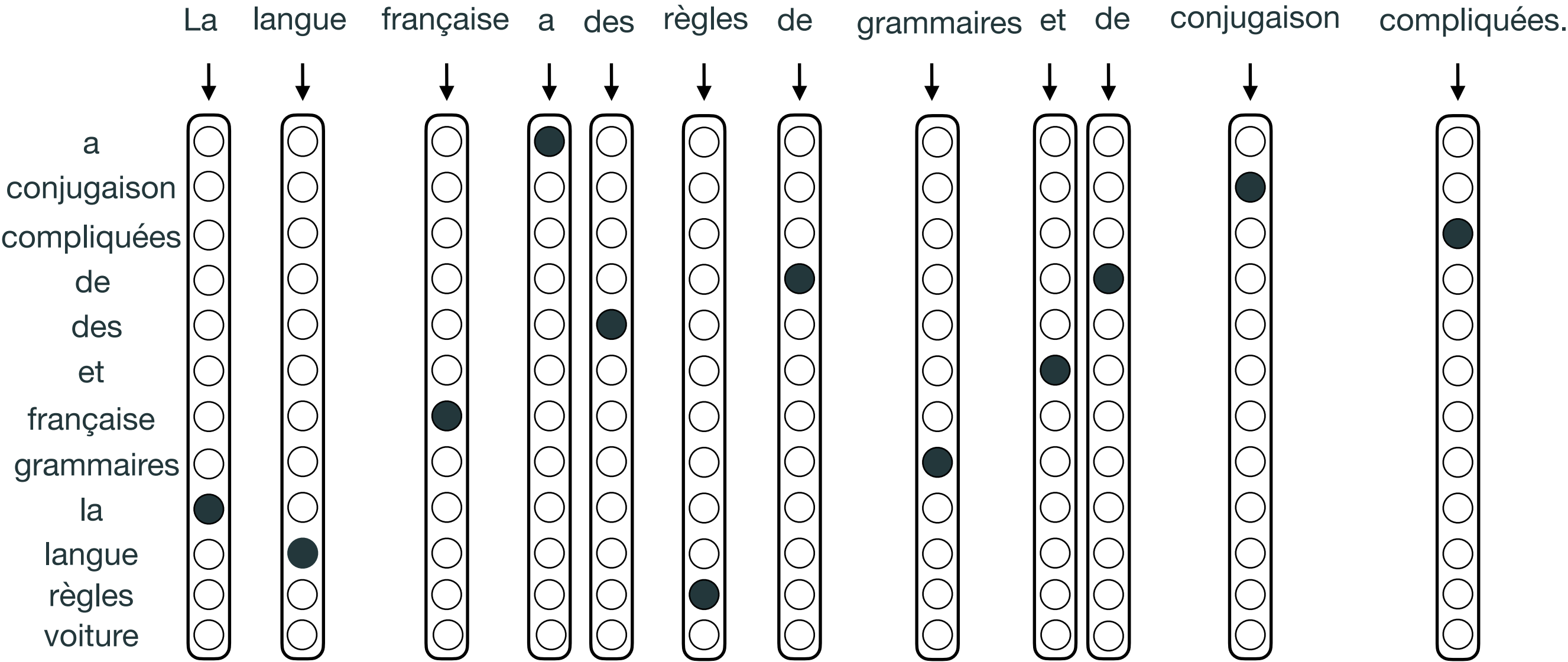
ONE-HOT-ENCODER



Dictionary size: $V = 12$

COUNT & BINARY VECTORISER

How to convert OHE encoding to 'sentences' encoding ?



COUNT VECTORISER

a	1
conjugaison	1
compliquées	1
de	2
des	1
et	1
française	1
grammaires	1
la	1
langue	1
règles	1
voiture	0

BINARY VECTORISER

a	1
conjugaison	1
compliquées	1
de	1
des	1
et	1
française	1
grammaires	1
la	1
langue	1
règles	1
voiture	0

Limitation: all words have the same weights

TF-IDF

Assign a **weight** to word, a token or an association of words in a **document** regarding to a **corpus of document**.

- t : a word or and association of words.
- d : a document.
- D : a corpus of document.

DEFINITION: TF-IDF general formula.

$$\text{tfidf}(t, d) = \text{tf}(t, d) \times \text{idf}(t, D)$$

- $\text{tf}(t, d)$: *Term-Frequency*. Number of occurrence of token t in document d .
- $\text{idf}(t, D)$: *Inverse-Document-Frequency*. Importance of token t in the corpus D .

TF FORMULA

The $tf(t, d)$ general formula is defined as the number of occurrence t in document d .

$$tf(t, d) = f_{t,d}$$

This definition is used in **scikit-learn** python library and **MLlib** spark library.

However there exist some variations:

Binary	0,1
Logarithmique normalisation	$1 + \log(f_{t,d})$
max normalisation	$0.5 + 0.5 \times \frac{f_{t,d}}{\max_{t' \in d} f_{t',d}}$
max normalisation (0.5)	$0.5 + 0.5 \times \frac{f_{t,d}}{\max_{t' \in d} f_{t',d}}$

IDF FORMULA

The $idf(t, D)$ change from an implementation to another.

$\log(\frac{N_D}{DF(t, D)})$	
$\log(\frac{N_D + 1}{DF(t, D) + 1})$	MLlib (Spark)
$\log(\frac{N_D + 1}{DF(t, D) + 1}) + 1$	Scikit-learn (Python)

- N_D : Number of documents.
- $DF(t, D)$: Number of documents in which terms t appears.

DIMENSION ISSUE

BINARY VECTORISER

a	1
conjugaison	1
compliquées	1
de	1
des	1
et	1
française	1
grammaires	1
la	1
langue	1
règles	1
voiture	1

V=11

BINARY VECTORISER

a	1
	.
conjugaison	1
	.
compliquées	1
	.
de	1
	.
des	1
	.
et	1
	.
française	1
	.
grammaires	1
	.
la	1
	.
langue	1
	.
règles	1
	.
voiture	0

V=10.000

- Vectors are very big
- Dimensions gros quickly

HASHING [WEINBERGER AND AL, 2009]

Vectorise descriptions while reduce features space

$$X \implies \phi$$

Vector of size V , unknown until computing all vocabulary.

Vector of size n_{hash} fixed.

- Deterministic function.
- Only one pass on data to build the vector.
- Unbiased cross product: $\mathbb{E}[\langle \phi(x), \phi(x') \rangle] = \langle x, x' \rangle$

HASHING [WEINBERGER AND AL, 2009]

HASHED FEATURE MAP

$$\phi_j^{\xi, h}(x) = \sum_{i \text{ s.t. } h(i)=j} \xi(i)x_i$$

Where

$$\begin{aligned} h: \mathbb{N} &\rightarrow \{1, \dots, nhash\} \\ i &\mapsto j = h(i) \end{aligned}$$

And

$$\begin{aligned} \xi: \mathbb{N} &\rightarrow \{1, \dots, -1\} \\ i &\mapsto j = \xi(i) \end{aligned}$$

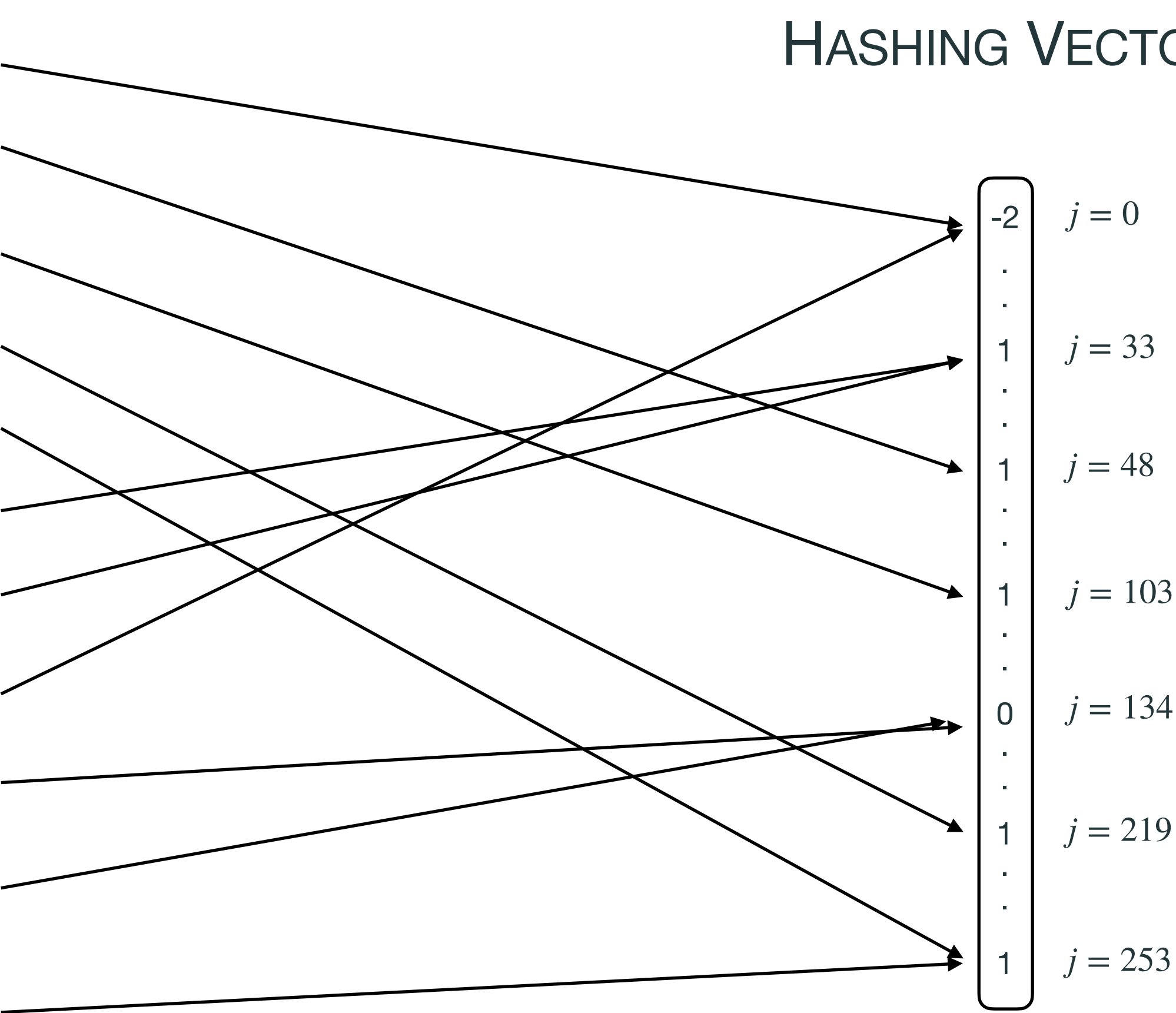
HASHING [WEINBERGER AND AL, 2009]

BINARY VECTORISER

a, i=0	1	$h(0) = 0, \xi(0) = -1$
conjugaison, i=1248	1	$h(1248) = 48, \xi(1248) = 1$
compliquées, i=1603	1	$h(1603) = 103, \xi(1603) = 1$
de, i=2019	1	$h(2019) = 219, \xi(2019) = 1$
des, i=2053	1	$h(2053) = 253, \xi(2053) = 1$
et, i=3033	1	$h(3033) = 33, \xi(3033) = 1$
française, i=3853	1	$h(3853) = 33, \xi(3853) = -1$
grammaires, i=14500	1	$h(4500) = 0, \xi(4500) = -1$
la, i=5234	1	$h(5234) = 134, \xi(5234) = 1$
langue, i=5834	1	$h(5834) = 134, \xi(5834) = -1$
règles, i=7453	1	$h(7453) = 253, \xi(7453) = 1$
voiture, i=9023	0	

$V = 10.000$

HASHING VECTOR



$n_{hash} = 300$

APPLICATION OF VECTORISATION FOR LEARNING

- **Hashing** and then **TF-IDF** are applied on training dataset.
- Same **hashing** function are used on test dataset.
- **TF** value between a word t and a document d are recomputed for the test dataset.
- **IDF** terms computed during training are re-used.

N-GRAMS

PROBLEMS

Some words does not have the same sense according to the context where it used.

- *Short de bain* \neq *short* \neq *bain*

SOLUTION

We consider not only the word (*unigram*) but also succession of two (*bigram*) or more words (*n-gram*).

- Solve language ambiguity.
- Explosion of vectors size. Example:
 - For 21.543 lines of categorise "TELEPHONIE - GPS"
 - 8.384 unigrams, 50.012 bigram, 90.854 trigram..