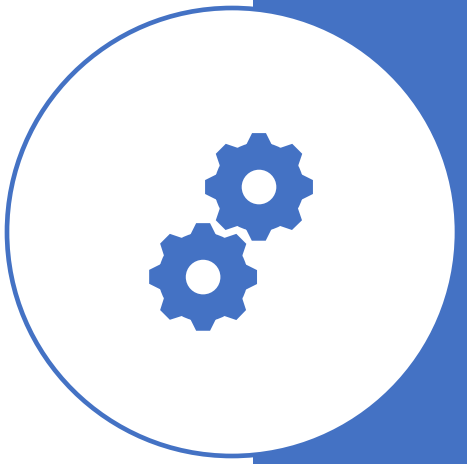
P5: Catégorisez automatiquement des questions

- Introduction
- Approche non supervisé
- Approche supervisé
- Démonstration API et synthèse



Problématique: Un système de suggestion de tags

Exemple:

title

How can I remove a key from a Python dictionary?

Asked 10 years, 6 months ago Modified 4 months ago Viewed 2.5m times

body

Is there a one-line way of deleting a key from a dictionary without raising a `KeyError`?

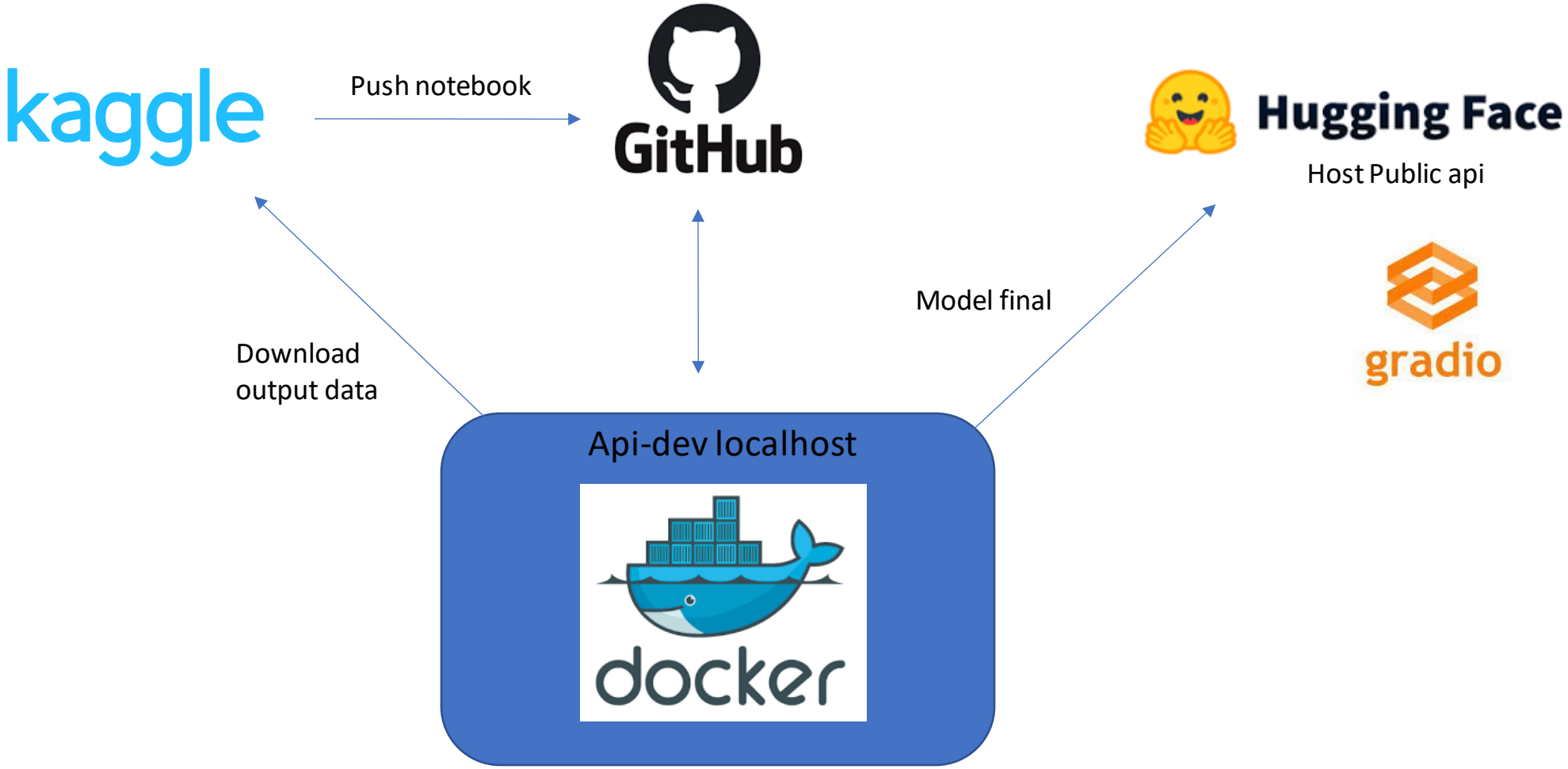
2724

```
if 'key' in my_dict:
    del my_dict['key']
```

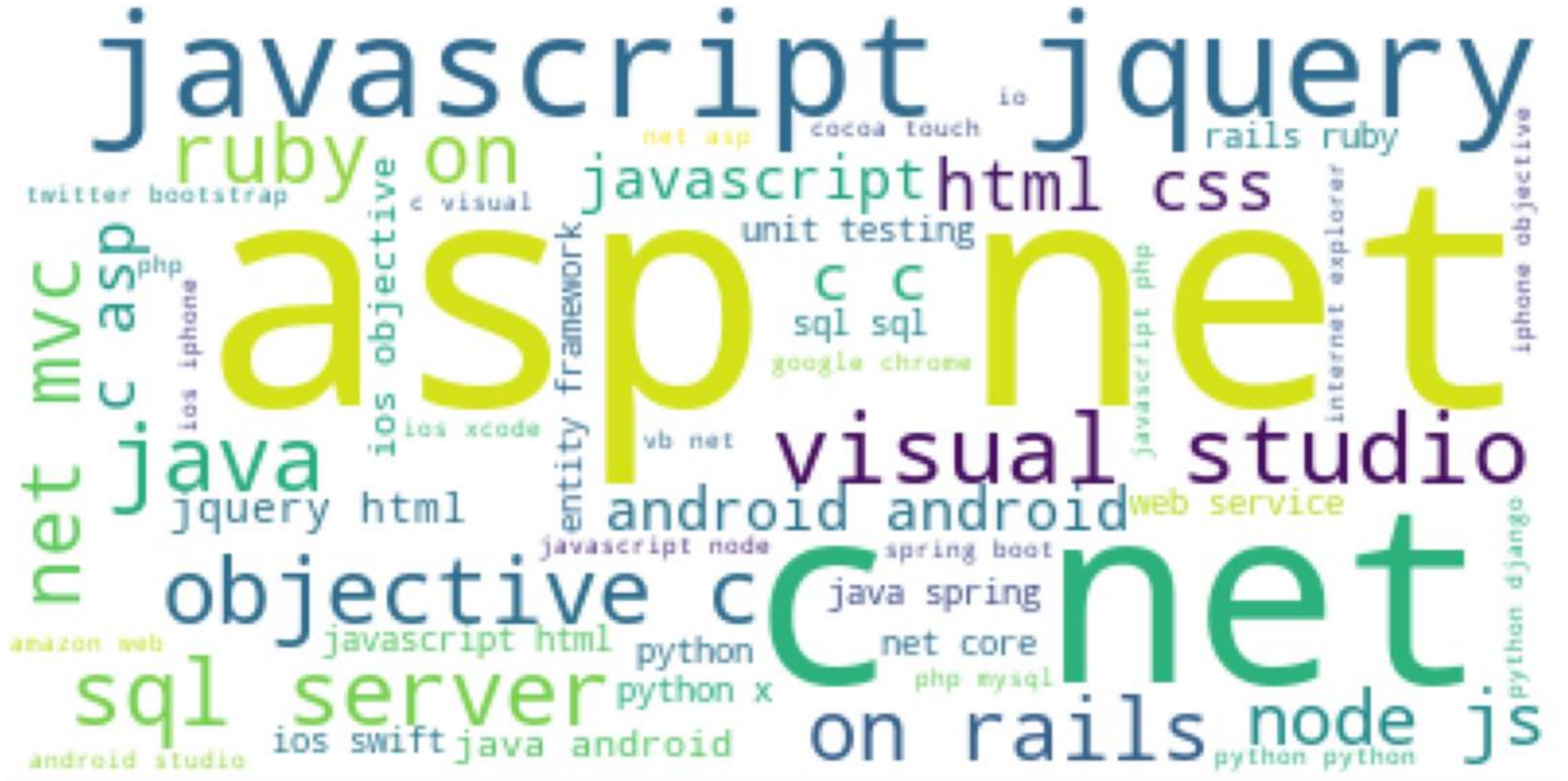
tags

python dictionary unset

Organisation



Tags

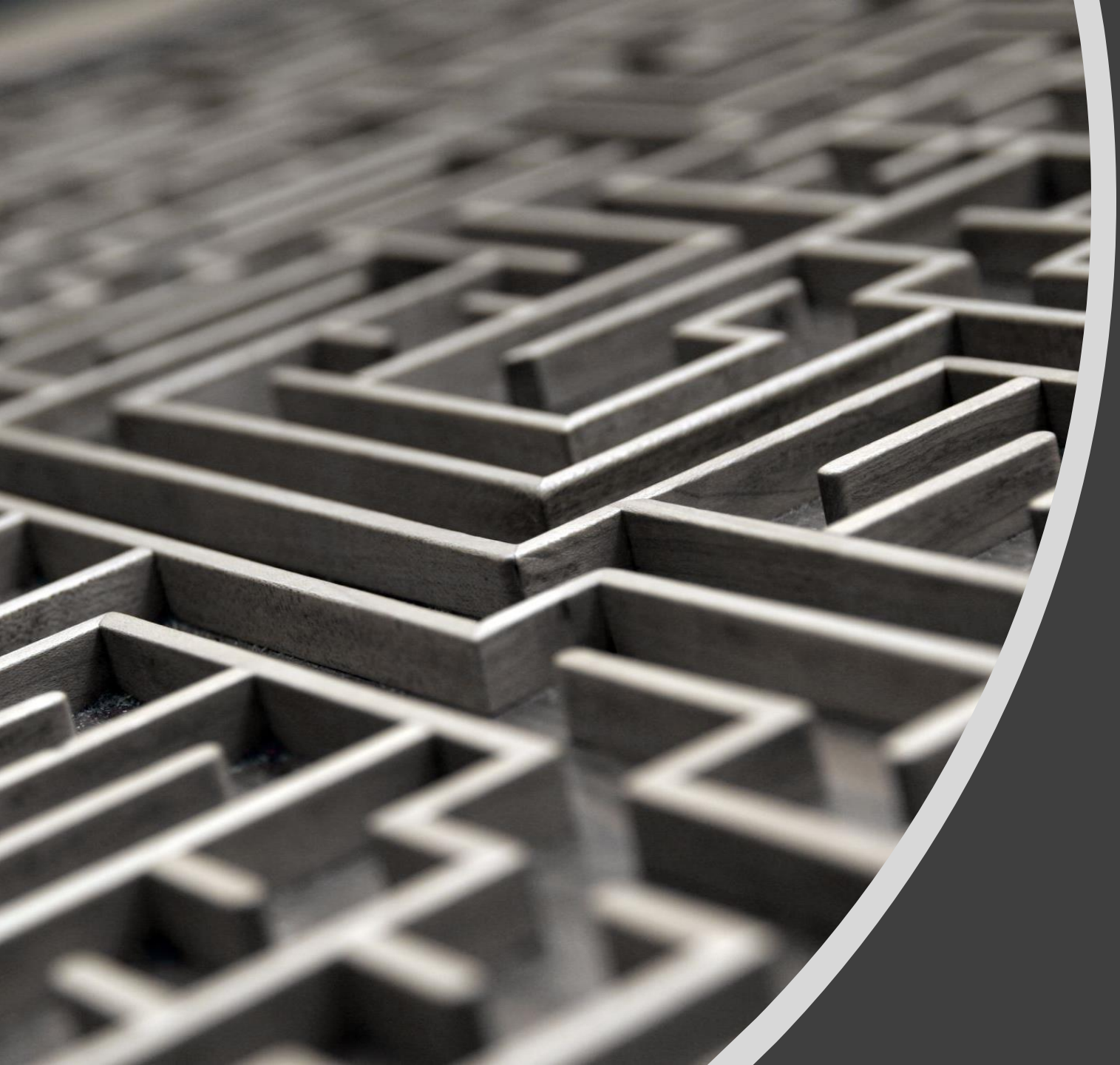

javascript jquery
asp net
ruby on
javascript
html css
java
visual studio
c net
objective c
sql server
on rails
node js
net mvc
c asp
jquery html
android android
java spring
net core
python
python x
ios swift
java android
unit testing
c c
sql sql
google chrome
ios objective
entity framework
ios xcode
vb net
javascript node
spring boot
web service
rails ruby
cocoa touch
twitter bootstrap
c visual
internet explorer
iphone objective
javascript php
ios iphone
php
amazon web
android studio
php mysql
python python
python django

prétraitement effectué

- Stop words
- Lemma , lower
- Séparation code

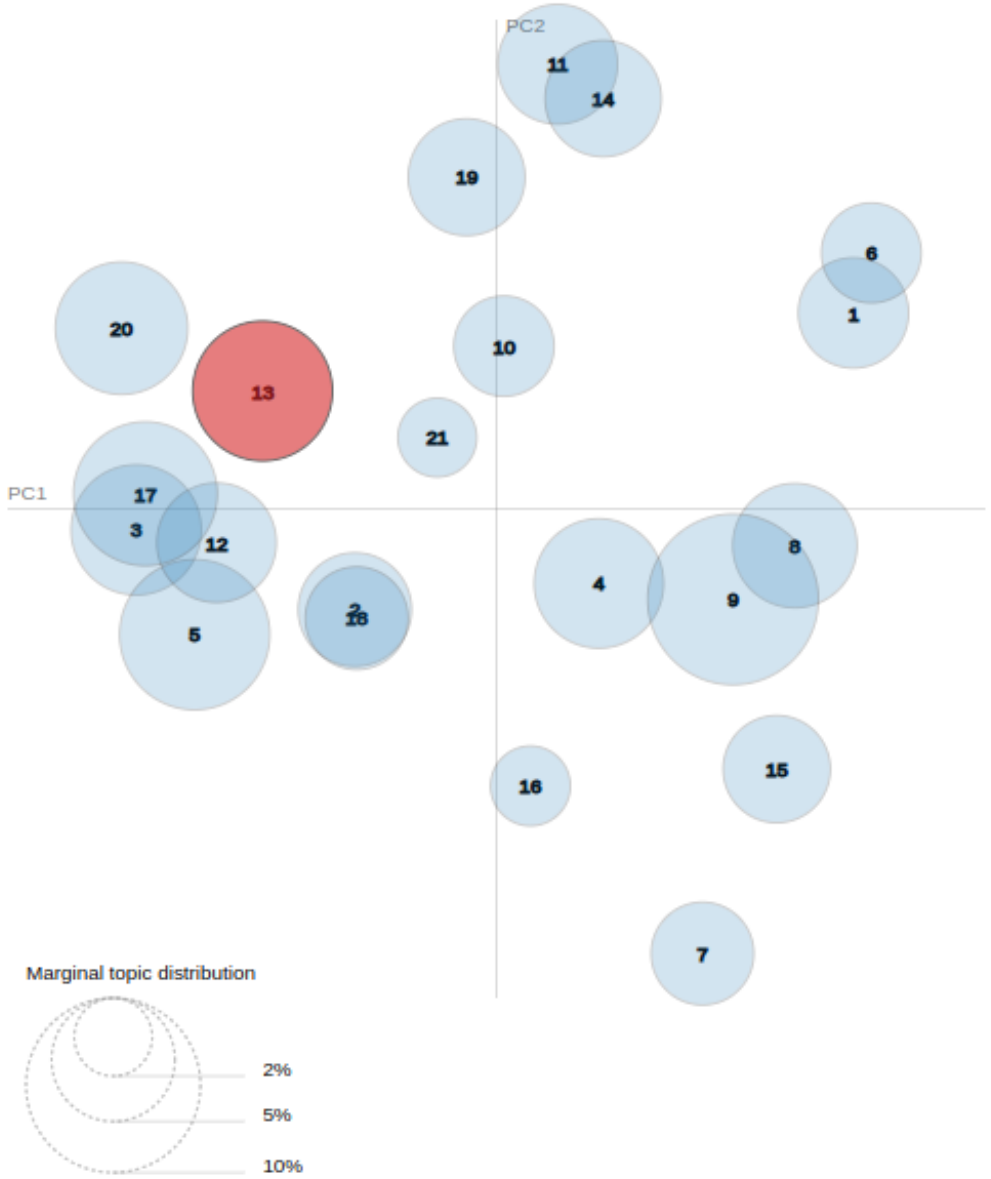
Exemple

Token	Lemma	Stopword	PartOfSpeech
How	how	True	('SCONJ',)
do	do	True	('AUX',)
I	I	True	('PRON',)
undo	undo	False	('VERB',)
the	the	True	('DET',)
most	most	True	('ADV',)
recent	recent	False	('ADJ',)
local	local	False	('ADJ',)
commits	commit	False	('NOUN',)
in	in	True	('ADP',)
Git	Git	False	('PROPN',)
?	?	False	('PUNCT',)

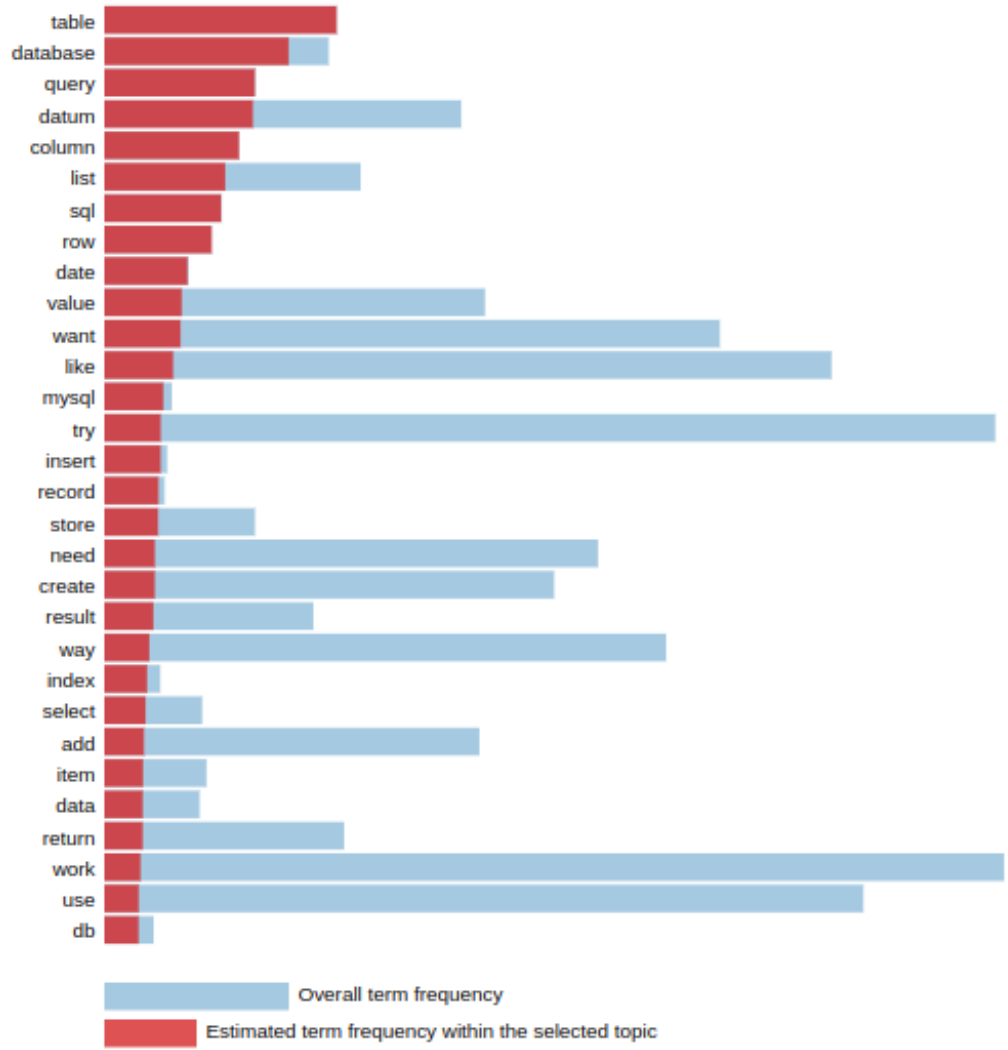


Approche non supervisée Latent Dirichlet Allocation LDA

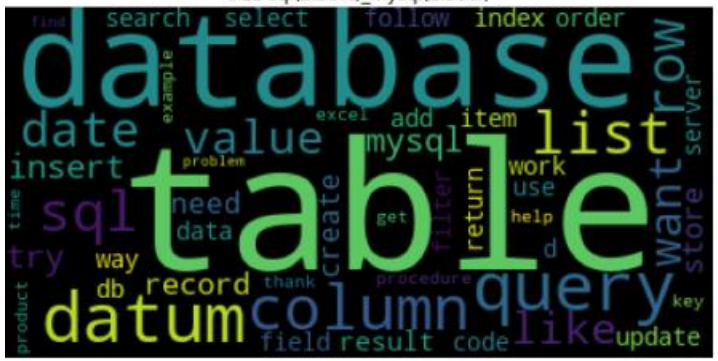
Intertopic Distance Map (via multidimensional scaling)



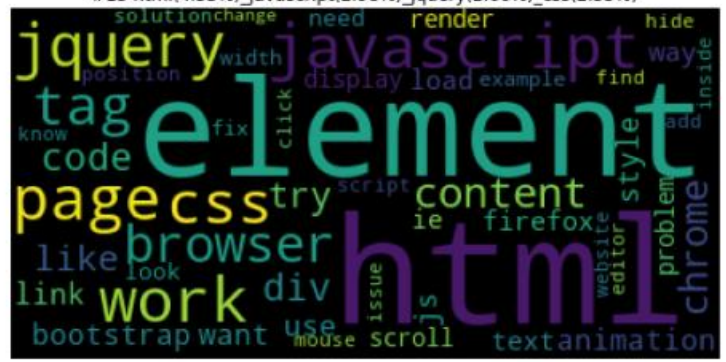
Top-30 Most Relevant Terms for Topic 13 (6.4% of tokens)



#12 sql(2.58%)_mysql(1.30%)

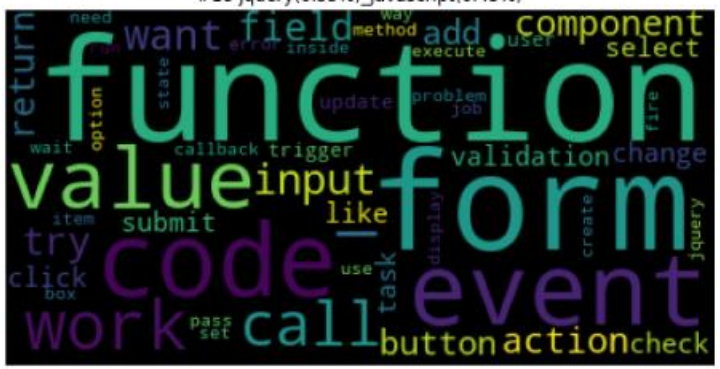


#13 html(4.33%)_javascript(2.98%)_jquery(2.66%)_css(2.55%)

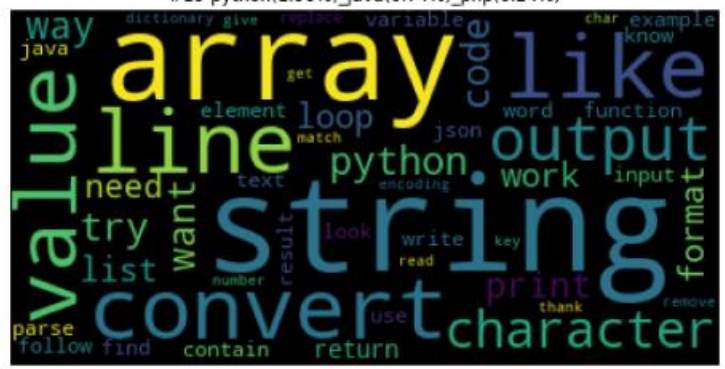


#14 android(0.28%)

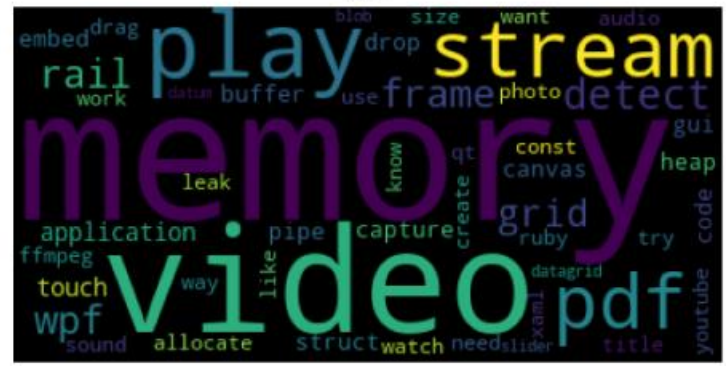
#18 jquery(0.55%)_javascript(0.43%)



#19 python(1.90%)_java(0.74%)_php(0.24%)



#20



Python sur 2612

	python(2.82%) linux(0.22%) ios(0.20%)	python(0.66%) c++(0.21%)	c(3.90%) .net(0.63%) c++(0.22%)	html(0.54%) python(0.45%)	c++(0.29%)	html(0.26%) php(0.21%)	c++(1.15%) c(0.60%)	sql(3.08%) mysql(1.72%) php(0.53%)	asp.net(0.36%)	python(0.38%) linux(0.35%)
count	900	843	417	409	358	353	334	318	289	281
mean	15	14	12	11	13	11	11	12	12	10
std	12	9	10	6	10	7	8	8	8	5
min	5	5	5	5	5	5	5	5	5	5
50%	11	11	9	9	10	9	9	9	10	8
max	144	84	126	41	105	63	60	62	49	35

Exemple

Tokens question:

process array process array piece c++ code show behavior reason sort datum region make loop time sort take time pass array need calculate array think language compiler anomaly try java result thought sorting bring datum cache array generate code sum term order matter relate follow q&a effect compiler option

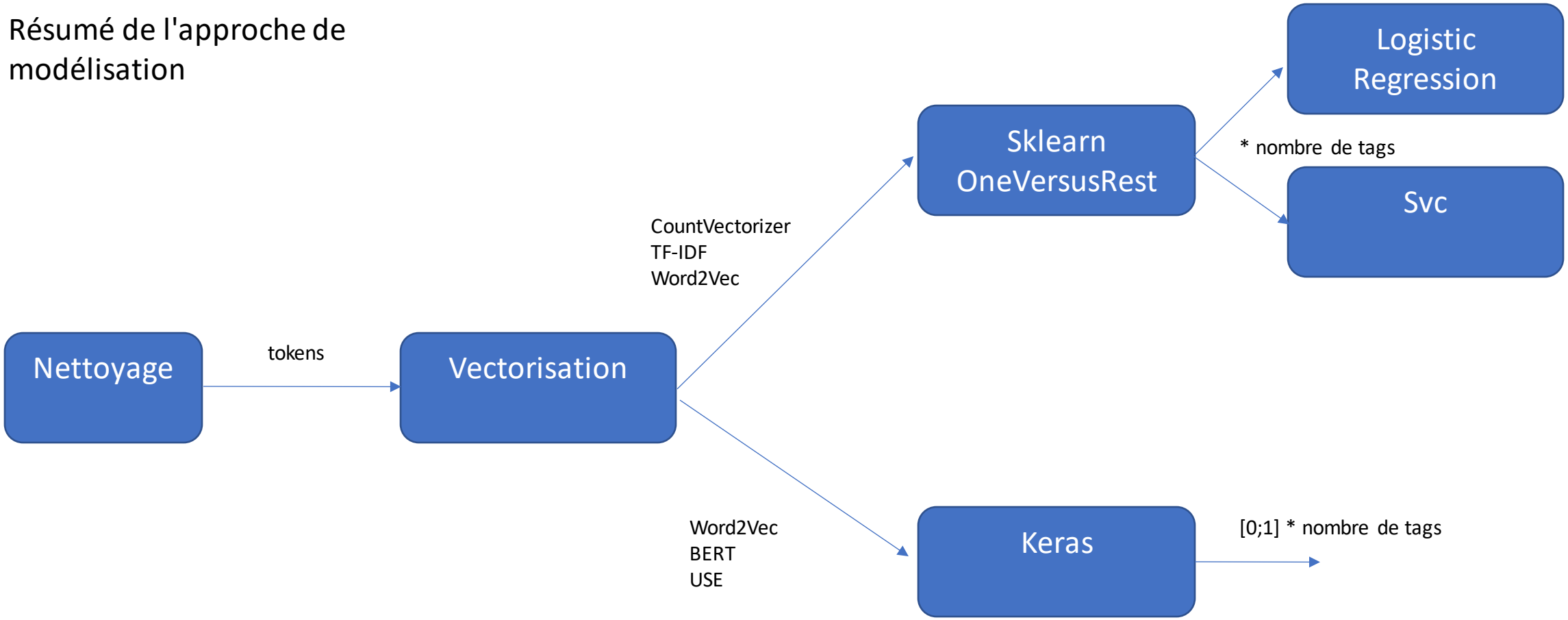
Topics:

- java(0.53%)_linux(0.39%): 10.93%,
- linux(4.08%): 7.49%,
- jquery(0.30%): 13.47%,
- '': 21.21%

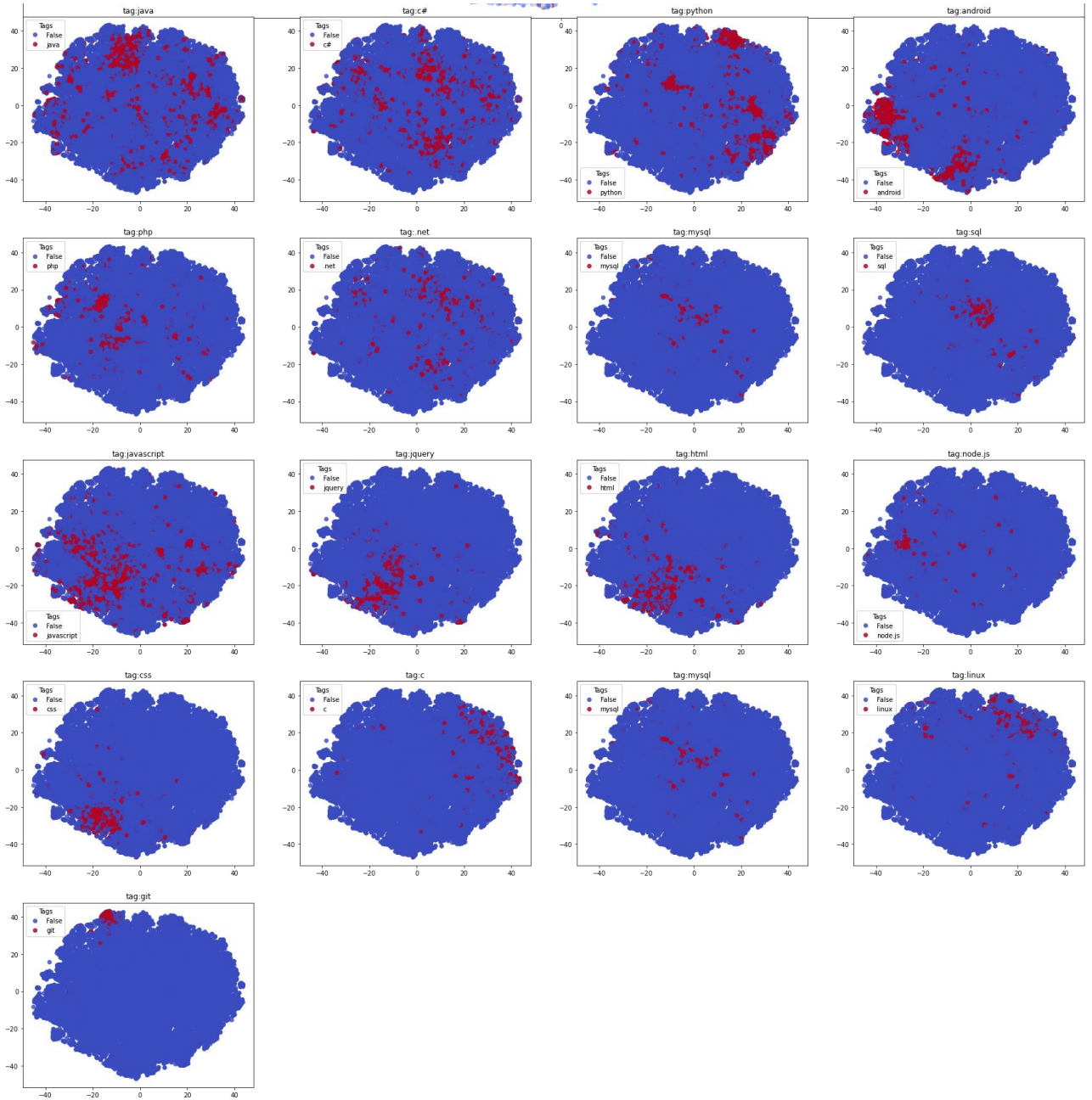
Approche supervisée

- One versus Rest :
  - SVC
  - Logistic Regression
- Word2Vec
- BERT
- USE

Résumé de l'approche de modélisation

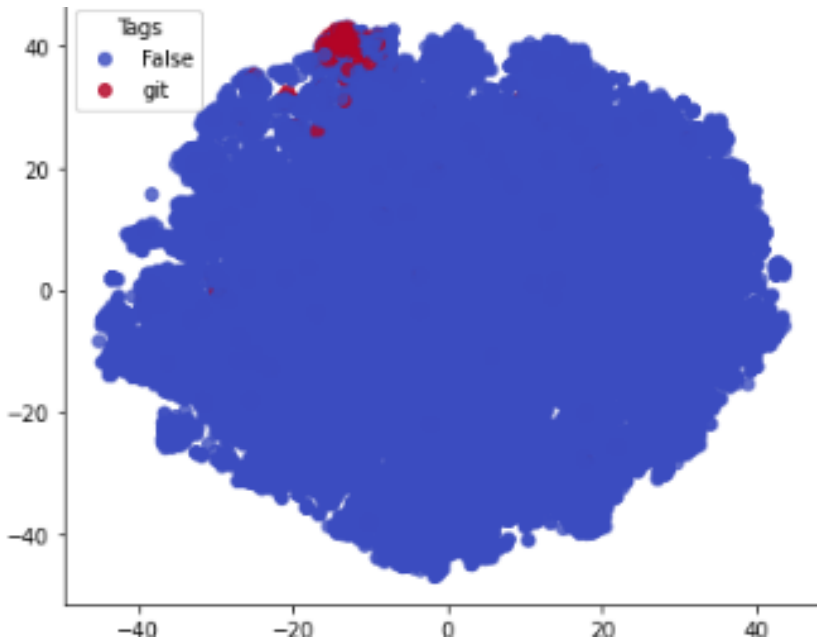


Visualisation TSNE embeddings USE

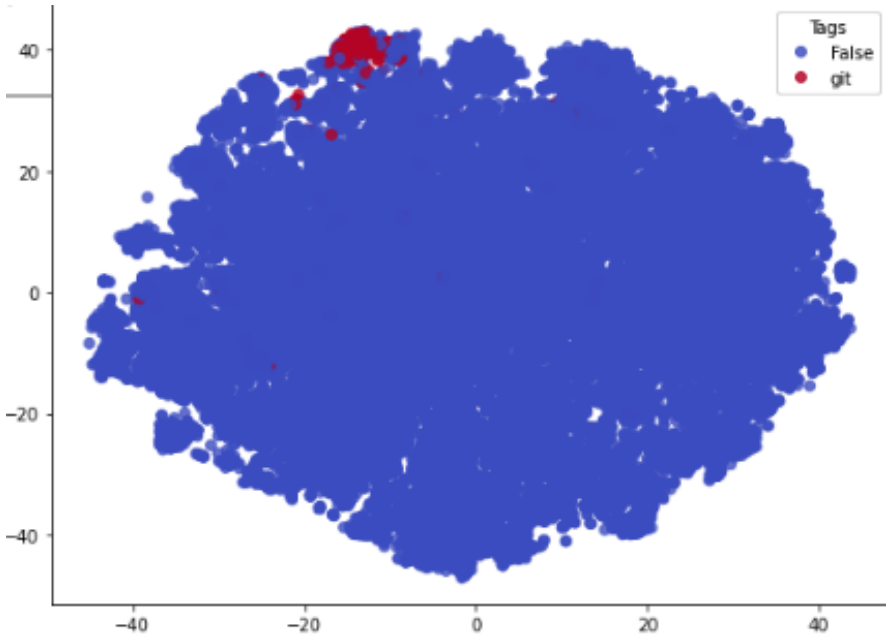


GIT

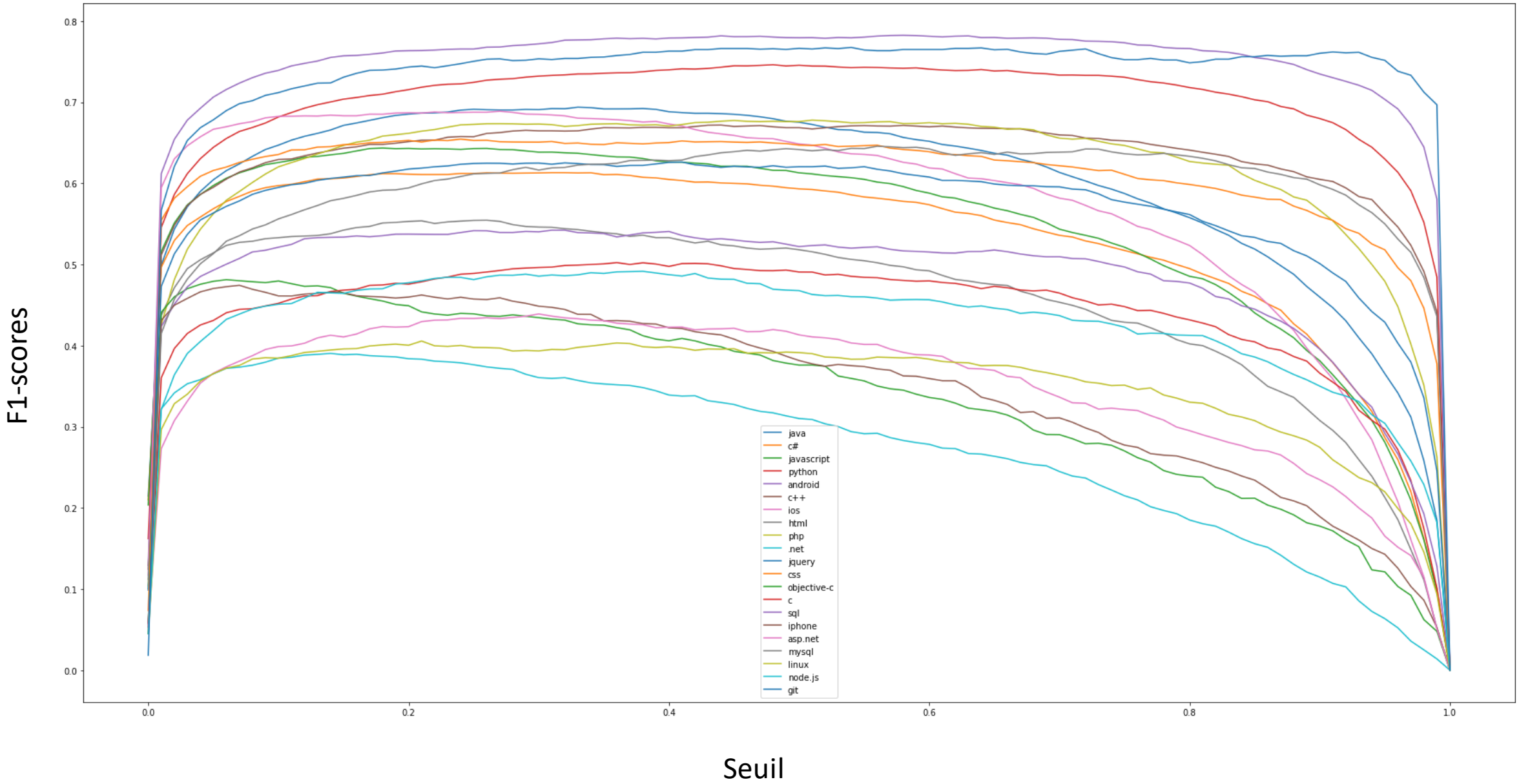
Réel



Prédit



Seuil optimal de chacun des tags pour le f1-scores
USE avec 21tags



F1-scores avec limite optimale

Global microavg : 61.44% 0.32	java : 69.39% 0.33	c# : 61.32% 0.31	javascript : 64.35% 0.18	python : 74.62% 0.48	android : 78.25% 0.58
c++ : 67.18% 0.44	ios : 68.89% 0.27	html : 55.46% 0.26	php : 67.79% 0.51	.net : 39.05% 0.14	jquery : 62.63% 0.41
css : 65.45% 0.24	objective-c : 48.13% 0.06	c : 50.23% 0.36	sql : 54.23% 0.31	iphone : 47.43% 0.07	asp.net : 43.90% 0.3
	mysql : 64.57% 0.56	linux : 40.56% 0.21	node.js : 49.16% 0.38	git : 76.75% 0.54	

Model Word2Vec

Layer (type)	Output Shape	Param #
text_vectorization (TextVect	(None, 256)	0
embedding (Embedding)	(None, 256, 512)	72620544
global_average_pooling1d (Gl	(None, 512)	0
dense (Dense)	(None, 256)	131328
dense_1 (Dense)	(None, 21)	5397

Total params: 72,757,269
Trainable params: 72,757,269
Non-trainable params: 0

Model BERT

Layer (type)	Output Shape	Param #	Connected to
input_word_ids (InputLayer)	[(None, 512)]	0	
input_mask (InputLayer)	[(None, 512)]	0	
input_type_ids (InputLayer)	[(None, 512)]	0	
keras_layer (KerasLayer)	[(None, 768), (None,	109482241	input_word_ids[0][0] input_mask[0][0] input_type_ids[0][0]
tf.__operators__.getitem (Slici	(None, 768)	0	keras_layer[0][1]
dense (Dense)	(None, 512)	393728	tf.__operators__.getitem[0][0]
dropout (Dropout)	(None, 512)	0	dense[0][0]
dense_1 (Dense)	(None, 128)	65664	dropout[0][0]
dropout_1 (Dropout)	(None, 128)	0	dense_1[0][0]
dense_2 (Dense)	(None, 21)	2709	dropout_1[0][0]

Total params: 109,944,342
Trainable params: 462,101
Non-trainable params: 109,482,241

Synthèse scores

Model	micro avg	macro avg	weighted avg	samples avg
Word2Vec keras	0.7	0.68	0.7	0.55
USE	0.66	0.61	0.64	0.49
TFIDF OVR LogisticRegression	0.6	0.56	0.59	0.4
Word2vec OVR SVC	0.5	0.44	0.48	0.41
BERT	0.45	0.42	0.46	0.32
naive tag dans le texte	0.45	0.41	0.42	0.34
TFIDF OVR SCV	0.45	0.3	0.4	0.37