artificiel_tokenize

November 26, 2021

1 Tokenisation

La tokenisation consiste à découper un texte en *token*, le plus souvent des mots. Le notebook utilise un extrait d'un article du monde.

[1]:
```
from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

[1]:
```
<IPython.core.display.HTML object>
```

1.1 Tokenizer

[2]:
```
texte = """
Mardi 20 février, à la médiathèque des Mureaux (Yvelines), le chef de l'Etat a␣
 ↪accompagné
la locataire de la rue de Valois pour la remise officielle du rapport
sur les bibliothèques, rédigé par leur ami commun, l'académicien
Erik Orsenna, avec le concours de Noël Corbin, inspecteur général
des affaires culturelles. L'occasion de présenter les premières
mesures en faveur d'un « plan bibliothèques ".
"""
```

1.1.1 nltk

La librairie la plus connue pour faire du traitement du langage naturel est nltk (ou *Natural Language Toolkit*).

[3]:
```
from nltk.tokenize import word_tokenize
' - '.join(word_tokenize(texte))
```

[3]:
```
'Mardi - 20 - février - , - à - la - médiathèque - des - Mureaux - ( - Yvelines
- ) - , - le - chef - de - l - ' - Etat - a - accompagné - la - locataire - de -
la - rue - de - Valois - pour - la - remise - officielle - du - rapport - sur -
les - bibliothèques - , - rédigé - par - leur - ami - commun - , - l - ' -
académicien - Erik - Orsenna - , - avec - le - concours - de - Noël - Corbin - ,
- inspecteur - général - des - affaires - culturelles - . - L - ' - occasion -
de - présenter - les - premières - mesures - en - faveur - d - ' - un - « - plan
- bibliothèques - " - .'
```

[4]:
```
from nltk.tokenize import WordPunctTokenizer
to = WordPunctTokenizer()
' - '.join(to.tokenize(texte))
```

[4]:
```
'Mardi - 20 - février - , - à - la - médiathèque - des - Mureaux - ( - Yvelines
- ), - le - chef - de - l - ' - Etat - a - accompagné - la - locataire - de - la
- rue - de - Valois - pour - la - remise - officielle - du - rapport - sur - les
- bibliothèques - , - rédigé - par - leur - ami - commun - , - l - ' -
académicien - Erik - Orsenna - , - avec - le - concours - de - Noël - Corbin - ,
- inspecteur - général - des - affaires - culturelles - . - L - ' - occasion -
de - présenter - les - premières - mesures - en - faveur - d - ' - un - « - plan
- bibliothèques - ".'
```

[5]:
```
from difflib import context_diff, ndiff
print('\n'.join(context_diff(word_tokenize(texte),
                             to.tokenize(texte),
                             fromfile='word_tokenize',
                             tofile='WordPunctTokenizer')))
```

```
*** word_tokenize

--- WordPunctTokenizer

***************

*** 9,16 ****

  Mureaux
  (
  Yvelines
! )
! ,
  le
  chef
  de
--- 9,15 ----

  Mureaux
  (
  Yvelines
! ),
  le
  chef
  de
***************

*** 77,81 ****

  «
  plan
  bibliothèques
! "
! .
--- 76,79 ----

  «
  plan
```

bibliothèques
! ".

1.1.2 gensim

La documentation de la librairie nltk est assez longue et ce n'est pas la plus simple d'accès. gensim est une autre option plus récente.

[6]:
```
from gensim.utils import tokenize
" - ".join(tokenize(texte, deacc=True, lower=True))
```

[6]:
```
'mardi - fevrier - a - la - mediatheque - des - mureaux - yvelines - le - chef -
de - l - etat - a - accompagne - la - locataire - de - la - rue - de - valois -
pour - la - remise - officielle - du - rapport - sur - les - bibliotheques -
redige - par - leur - ami - commun - l - academicien - erik - orsenna - avec -
le - concours - de - noel - corbin - inspecteur - general - des - affaires -
culturelles - l - occasion - de - presenter - les - premieres - mesures - en -
faveur - d - un - plan - bibliotheques'
```

[7]:
```
from gensim.utils import tokenize
" - ".join(tokenize("chiffres 20 ch20", deacc=True, lower=True))
```

[7]:
```
'chiffres - ch'
```

1.1.3 spacy

Un dernier module a vu le jour spacy. On suit l'exemple présenté dans spacy-101. Il faut télécharger un paquet de ressource depuis spacy-models. **Note :** sous Windows, il faut faudra ruser et installer le module fr_core_news_sm vous même (et bidouiller le fichier setup.py).

[8]:
```
import spacy
nlp = spacy.load('fr_core_news_sm')
# Il faut exécuter ceci depuis la ligne de commande au moins une fois.
# python -m spacy download fr_core_news_sm
```

[9]:
```
doc = nlp(texte)
```

[10]:
```
' - '.join(t.text for t in doc)
```

[10]:
```
'\n - Mardi - 20 - février - , - à - la - médiathèque - des - Mureaux - ( -
Yvelines - ) - , - le - chef - de - l' - Etat - a - accompagné - \n - la -
locataire - de - la - rue - de - Valois - pour - la - remise - officielle - du -
rapport - \n - sur - les - bibliothèques - , - rédigé - par - leur - ami -
commun - , - l' - académicien - \n - Erik - Orsenna - , - avec - le - concours -
de - Noël - Corbin - , - inspecteur - général - \n - des - affaires -
culturelles - . - L' - occasion - de - présenter - les - premières - \n -
mesures - en - faveur - d' - un - « - plan - bibliothèques - " - . - \n'
```

[11]:
```
' - '.join(t.text for t in doc if t.is_alpha)
```

[11]:
```
'Mardi - février - à - la - médiathèque - des - Mureaux - Yvelines - le - chef -
de - Etat - a - accompagné - la - locataire - de - la - rue - de - Valois - pour
- la - remise - officielle - du - rapport - sur - les - bibliothèques - rédigé -
par - leur - ami - commun - académicien - Erik - Orsenna - avec - le - concours
```

```
- de - Noël - Corbin - inspecteur - général - des - affaires - culturelles -
occasion - de - présenter - les - premières - mesures - en - faveur - un - plan
- bibliothèques'
```

On voit que la tokenisation des apostrophes est différente et qu'on a plus d'information sur chaque token.

[12]:
```
el = doc[1]
```

[13]:
```
el.text, el.is_alpha
```

[13]:
```
('Mardi', True)
```

1.2 Supprimer les stopwords

1.2.1 nltk

Le module nltk fournit une liste de stopwords. Il suffit de supprimer tous les mots dans cette liste.

[14]:
```
from nltk.corpus import stopwords
' - '.join(stopwords.words('english'))
```

[14]:
```
"i - me - my - myself - we - our - ours - ourselves - you - you're - you've -
you'll - you'd - your - yours - yourself - yourselves - he - him - his - himself
- she - she's - her - hers - herself - it - it's - its - itself - they - them -
their - theirs - themselves - what - which - who - whom - this - that - that'll
- these - those - am - is - are - was - were - be - been - being - have - has -
had - having - do - does - did - doing - a - an - the - and - but - if - or -
because - as - until - while - of - at - by - for - with - about - against -
between - into - through - during - before - after - above - below - to - from -
up - down - in - out - on - off - over - under - again - further - then - once -
here - there - when - where - why - how - all - any - both - each - few - more -
most - other - some - such - no - nor - not - only - own - same - so - than -
too - very - s - t - can - will - just - don - don't - should - should've - now
- d - ll - m - o - re - ve - y - ain - aren - aren't - couldn - couldn't - didn
- didn't - doesn - doesn't - hadn - hadn't - hasn - hasn't - haven - haven't -
isn - isn't - ma - mightn - mightn't - mustn - mustn't - needn - needn't - shan
- shan't - shouldn - shouldn't - wasn - wasn't - weren - weren't - won - won't -
wouldn - wouldn't"
```

[15]:
```
' - '.join(stopwords.words('french'))
```

[15]:
```
'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
- ne - nos - notre - nous - on - ou - par - pas - pour - qu - que - qui - sa -
se - ses - son - sur - ta - te - tes - toi - ton - tu - un - une - vos - votre -
vous - c - d - j - l - à - m - n - s - t - y - été - étée - étées - étés - étant
- étante - étants - étantes - suis - es - est - sommes - êtes - sont - serai -
seras - sera - serons - serez - seront - serais - serait - serions - seriez -
seraient - étais - était - étions - étiez - étaient - fus - fut - fûmes - fûtes
- furent - sois - soit - soyons - soyez - soient - fusse - fusses - fût -
fussions - fussiez - fussent - ayant - ayante - ayantes - ayants - eu - eue -
eues - eus - ai - as - avons - avez - ont - aurai - auras - aura - aurons -
aurez - auront - aurais - aurait - aurions - auriez - auraient - avais - avait -
avions - aviez - avaient - eut - eûmes - eûtes - eurent - aie - aies - ait -
ayons - ayez - aient - eusse - eusses - eût - eussions - eussiez - eussent'
```

[16]:
```
st = set(stopwords.words('french'))
' - '.join(w for w in word_tokenize(texte) if w not in st)
```

[16]:
```
'Mardi - 20 - février - , - médiathèque - Mureaux - ( - Yvelines - ) - , - chef
- ' - Etat - a - accompagné - locataire - rue - Valois - remise - officielle -
rapport - bibliothèques - , - rédigé - ami - commun - , - ' - académicien - Erik
- Orsenna - , - concours - Noël - Corbin - , - inspecteur - général - affaires -
culturelles - . - L - ' - occasion - présenter - premières - mesures - faveur -
' - « - plan - bibliothèques - " - .'
```

1.2.2 gensim

[17]:
```
from gensim.parsing.preprocessing import STOPWORDS
" - ".join(STOPWORDS)
```

[17]:
```
'co - again - alone - doesn - two - one - kg - con - latter - own - thin -
however - along - hers - further - first - had - three - de - because -
nevertheless - ours - using - thereafter - mostly - ten - same - down - just -
every - above - anyway - already - five - she - both - ourselves - twenty -
anywhere - former - before - third - else - herein - across - next - full -
should - fifty - whenever - the - fire - hereupon - take - could - became -
where - therefore - mill - about - more - perhaps - almost - never - whole -
anyhow - was - wherein - hasnt - myself - nobody - me - whether - found - be -
out - fifteen - becoming - as - becomes - we - please - nowhere - another - our
- this - amount - then - enough - somewhere - being - very - while - behind -
over - only - get - between - most - he - last - put - have - besides - himself
- ltd - still - none - am - yourselves - whence - latterly - us - doing - what -
seemed - does - cannot - onto - upon - noone - name - beyond - front - except -
whereby - thereupon - although - throughout - whatever - unless - other - either
- regarding - moreover - become - not - that - each - call - eg - among - whom -
anyone - made - used - computer - amongst - detail - off - amoungst - his -
their - empty - namely - mine - if - quite - wherever - yet - nor - whereafter -
him - some - nine - didn - themselves - together - everyone - they - yourself -
around - hence - un - interest - cry - why - who - done - say - thick - make -
sixty - eleven - beside - everything - been - anything - has - of - ie -
describe - and - rather - per - up - those - part - but - otherwise - thence -
give - now - keep - seems - well - least - to - too - serious - might - inc - km
- itself - meanwhile - her - few - whereas - find - neither - via - formerly -
at - etc - must - when - within - see - somehow - eight - often - four - various
- do - sincere - whereupon - your - such - don - twelve - on - with - were -
hereafter - or - you - herself - towards - my - for - go - them - by - in -
nothing - all - can - afterwards - under - thus - will - once - which - any -
indeed - bottom - system - whoever - from - against - whose - many - thereby -
show - elsewhere - without - how - everywhere - an - during - much - are - here
- below - would - a - cant - beforehand - always - hundred - sometime - due -
bill - though - someone - since - after - others - six - forty - yours - back -
move - ever - into - less - did - may - toward - i - no - so - it - couldnt -
several - its - side - these - even - seem - whither - really - therein - top -
until - something - thru - re - there - is - than - hereby - through - fill -
sometimes - also - seeming'
```

1.2.3 spacy

Encore plus simple avec spacy où chaque token contient l'information souhaitée.

[18]:
```
docw = nlp(texte)
' - '.join(t.text for t in docw if t.is_stop)
```

[18]:
```
'à - la - des - le - de - l' - a - la - de - la - de - pour - la - du - sur -
les - par - leur - l' - avec - le - de - des - L' - de - les - en - d' - un'
```

[19]:
```
' - '.join(t.text for t in docw if not t.is_stop)
```

[19]:
```
'\n - Mardi - 20 - février - , - médiathèque - Mureaux - ( - Yvelines - ) - , -
chef - Etat - accompagné - \n - locataire - rue - Valois - remise - officielle -
rapport - \n - bibliothèques - , - rédigé - ami - commun - , - académicien - \n
- Erik - Orsenna - , - concours - Noël - Corbin - , - inspecteur - général - \n
- affaires - culturelles - . - occasion - présenter - premières - \n - mesures -
faveur - « - plan - bibliothèques - " - . - \n'
```

1.3 Autres modules

- textblob, textblob-fr : Simplified Text Processing
- corpora, pycorpora : corpus de texte
- regex4dummies : expression régulière pour extraire des informations dans un texte

Il existe une quantité de modules différentes. Lorsque les sources sont connues et très utilisées comme wikipedia.

1.4 n-grams

Petit intermède : après un découpage en mots, on ne considère plus l'ordre avec une approche *bag-of-words*. Si l'information contenu par l'ordre des mots s'avère importante, il faut considérer un découpage en couple de mots (bi-grammes), triplets de mots (3-grammes)…

[20]:
```
from nltk.util import ngrams
generated_ngrams = ngrams(word_tokenize(texte), 4, pad_left=True, pad_right=True)
list(generated_ngrams)[:7]
```

[20]:
```
[(None, None, None, 'Mardi'),
 (None, None, 'Mardi', '20'),
 (None, 'Mardi', '20', 'février'),
 ('Mardi', '20', 'février', ','),
 ('20', 'février', ',', 'à'),
 ('février', ',', 'à', 'la'),
 (',', 'à', 'la', 'médiathèque')]
```

1.5 Versions utilisées pour ce notebook

spacy s'est montré quelque peu fantasques cette année avec quelques erreurs notamment celle-ci : ValueError: cymem.cymem.Pool has the wrong size, try recompiling. Voici les versions utilisées…

[21]:
```
def version(module):
    try:
        ver = getattr(module, '__version__', None)
        if ver is None:
```

```
            ver = [_ for _ in os.listdir(os.path.join(module.__file__, '..', '..')) \
                   if module.__name__ in _][-1]
        return ver
    except Exception as e:
        return str(e)
```

[22]:
```
import os
import thinc
print("thinc", version(thinc))
import preshed
print("preshed", version(preshed))
import cymem
print("cymem", version(cymem))
import murmurhash
print("murmurhash", version(murmurhash))
import plac
print("plac", plac.__version__)
import spacy
print("spacy", spacy.__version__)
```

```
thinc 7.4.1
preshed preshed-3.0.2.dist-info
cymem cymem-2.0.2.dist-info
murmurhash murmurhash-1.0.2.dist-info
plac 0.9.6
spacy 2.3.2
```

[23]:
```

```