pretraitement

May 18, 2021

1 Phase de pretraitement des données textuelles

Une fois les données nettoyées, on passe à leur prétraitement. En effet, on ne pourra pas passer les données brutes directement au modèle. Il faut passer par certaines étapes afin de lui permettre d'en retirer quelque chose.

[1]:

```python
import sys, re, inflect
import pandas as pd
from clean import clean_claimKG

import contractions as c
import nltk
from nltk.corpus import stopwords, wordnet
from nltk.stem.snowball import SnowballStemmer
from nltk.stem.lancaster import LancasterStemmer
from nltk.stem import WordNetLemmatizer
#nltk.download('all')
```

On commence par charger les données et les nettoyées. Voir Data presentation

[2]:

```python
file_name = "../data/claimKG.csv"

# Lecture du fichier
kg_origin = pd.read_csv(file_name)
kg= kg_origin.copy()
clean_claimKG(kg,inplace=True, verbose=True)
```

```
Taille du dataframe: (39218, 23)
Suppression des 10 columns suivant:
        -> Unnamed: 0
        -> claimReview_source
        -> claimReview_author
        -> claimReview_author_url
        -> creativeWork_author_name
        -> creativeWork_author_sameAs
        -> creativeWork_datePublished
        -> rating_bestRating
        -> rating_ratingValue
        -> rating_worstRating
```

```
Suppression de 5749 lignes en doubles.
Suppression de 5 lignes.
Taille finale: (33464, 13)
```

[2]:

```
      claimReview_author_name  \
0                      snopes
1                      snopes
2                      snopes
3                      snopes
4                      snopes
…                        …
39213           factcheck_afp
39214           factcheck_afp
39215           factcheck_afp
39216           factcheck_afp
39217           factcheck_afp

                               claimReview_claimReviewed  \
0      Finnish President Sauli Niinistö posted a vide…
1      A supporter of U.S. Rep. Alexandria Ocasio-Cor…
2      A photograph shows a bride and groom during a …
3         Canada legalized the medicinal use of cocaine.
4      In September 2019, U.S. President Donald Trump…
…                                                   …
39213  Rat meat from China is sold as boneless chicke…
39214     Mars will appear as big as the Moon on July 27
39215  Massachusetts repealed the Second Amendment of…
39216  Une photo montre une foule très dense sur une …
39217     Photos show the Croatian president in a bikini

      claimReview_datePublished  \
0                    2019-10-07
1                    2019-10-04
2                    2019-10-04
3                    2019-10-04
4                    2019-10-04
…                          …
39213                       NaN
39214                       NaN
39215                       NaN
39216                       NaN
39217                       NaN

                                         claimReview_url  \
0      https://www.snopes.com/fact-check/president-fi…
1      https://www.snopes.com/fact-check/babies-clima…
2      https://www.snopes.com/fact-check/handmaid-tal…
```

```
3      https://www.snopes.com/fact-check/medicinal-co…
4      https://www.snopes.com/fact-check/trump-autism…
…                                                  …
39213  https://factcheck.afp.com//no-massive-quantiti…
39214  https://factcheck.afp.com//no-mars-will-not-be…
39215  https://factcheck.afp.com//massachusetts-did-n…
39216  https://factcheck.afp.com//no-photo-does-not-s…
39217  https://factcheck.afp.com//images-croatian-pre…

                                              extra_body  \
0      On Oct. 2, 2019, a joint press conference at t…
1      An Oct. 3, 2019, town hall event in New York C…
2      In October 2019, a photograph supposedly showi…
3      On Sep. 20, 2019, Huzlers published an article…
4      We received multiple inquiries from readers in…
…                                                  …
39213  Claims that 1 million pounds of rat meat from …
39214  Mars will be as big as the Moon in the sky thi…
39215  Online reports claim that the US state of Mass…
39216  According to several posts on Facebook, a vira…
39217  As Croatia secured its place in the World Cup …

                                   extra_entities_author  \
0                                                     []
1                                                     []
2                                                     []
3                                                     []
4                                                     []
…                                                  …
39213                                                 []
39214                                                 []
39215  [{"id" : 58299742",""begin": 0,"end": 12,"enti…
39216  [{"id" : 7529378",""begin": 13,"end": 21,"enti…
39217                                                 []

                                     extra_entities_body  \
0      [{"id" : 33057",""begin": 46,"end": 57,"entity…
1      [{"id" : 645042",""begin": 33,"end": 46,"entit…
2      [{"id" : 50430110",""begin": 91,"end": 106,"en…
3      [{"id" : 7701",""begin": 96,"end": 103,"entity…
4      [{"id" : 4848272",""begin": 121,"end": 133,"en…
…                                                  …
39213  [{"id" : 11632",""begin": 131,"end": 162,"enti…
39214  [{"id" : 13586",""begin": 757,"end": 762,"enti…
39215  [{"id" : 18618239",""begin": 30,"end": 38,"ent…
39216  [{"id" : 7529378",""begin": 30,"end": 38,"enti…
39217  [{"id" : 5573",""begin": 3,"end": 10,"entity":…
```

```
                extra_entities_claimReview_claimReviewed  \
0      [{"id" : 1042690",""begin": 18,"end": 32,"enti…
1      [{"id" : 54885332",""begin": 22,"end": 45,"ent…
2      [{"id" : 50430110",""begin": 46,"end": 61,"ent…
3      [{"id" : 7701",""begin": 38,"end": 45,"entity"…
4      [{"id" : 4848272",""begin": 31,"end": 43,"enti…
…                                                 …
39213                                                 []
39214  [{"id" : 14640471",""begin": 0,"end": 4,"entit…
39215  [{"id" : 31655",""begin": 27,"end": 43,"entity…
39216                                                 []
39217                                                 []

      extra_entities_keywords  \
0                          []
1                          []
2                          []
3                          []
4                          []
…                       …
39213                      []
39214                      []
39215                      []
39216                      []
39217                      []

                                    extra_refered_links extra_tags  \
0      https://t.co/Oo5Q56ALAu,https://twitter.com/ia…        NaN
1      https://twitter.com/redsteeze/status/117991491…        NaN
2      https://twitter.com/God_loves_women/status/117…        NaN
3      https://web.archive.org/web/20191004171021/htt…        NaN
4      http://archive.is/ymlJP,http://archive.is/JgYP…        NaN
…                                                 …        …
39213  https://web.archive.org/web/20180725201453/htt…        NaN
39214  https://web.archive.org/web/20180725152030/htt…        NaN
39215  https://web.archive.org/web/20180720190537/htt…        NaN
39216  https://www.facebook.com/strangworldstrangerpe…        NaN
39217  https://twitter.com/TitansHomer/status/1015772…        NaN

                                            extra_title rating_alternateName
0      Did the President of Finland Post a Video Resp…                False
1      Did an AOC Supporter Suggest ‘Eating Babies’ t…              Mixture
2      Is This a Photo of a ‘Handmaid’s Tale’-Themed …         Miscaptioned
3      Did Canada Legalize the Medicinal Use of Cocaine?       Labeled Satire
4      Did Donald Trump Sign a $1.8 Billion Autism-Se…                 True
…                                                 …                 …
```

```
39213  No, massive quantities of rat meat are still n…                FALSE
39214  No, Mars will not be as big as the Moon in the…                 Hoax
39215  Massachusetts did not repeal the Second Amendm…           Misleading
39216  No, this photo does not show a crowded beach i…      Photo détournée
39217  Images of the Croatian president in a bikini: …   Misrepresentation

[33464 rows x 13 columns]
```

Ensuite, on sélectionne 10 assertions au hasard afin de les manipuler et de voir ce qu'on peut en faire.

[3]:
```
claims_text = list(kg['claimReview_claimReviewed'].sample(10))

for index, claim in enumerate(claims_text):
    print(index,"->",claim)
```

```
0 -> ""Michael Dukakis created jobs three times faster"" than Mitt Romney.
1 -> Says Pennsylvania charges a top income tax rate of 3 percent and Delaware
""has no state income tax at all.""
2 -> Says Hillary Clinton ""was literally present when we pressed the reset
button with Russia just a few months after Russia had invaded Georgia.""
3 -> The former Florida representative and wrote an angry column about 49ers
quarterback Colin Kaepernick on his personal blog.
4 -> Wisconsin's Lincoln Hills youth prison has a ""66 percent recidivism
rate,"" while ""states like Missouri, that have more of a regional model -- 8
percent.""
5 -> Since Oregon’s prescription-only law took effect, meth lab incidents have
dropped by 96 percent and meth-related arrests by 32 percent.
6 -> Members of Congress Refuse to Stand for Pledge of Allegiance
7 -> A 1942 Merrie Melodies cartoon includes a glimpse of Bug Bunny's phallus.
8 -> Says that Mitt Romney's response to the crisis in the auto industry was,
""Let Detroit go bankrupt.""
9 -> Says Democratic obstruction is the reason why ""many important positions in
government are unfilled.""
```

1.1 Tokenization

Découpage de l'assertion en Token (en mots).

[4]:
```
def tokenize(text):
    return nltk.word_tokenize(text)

tokenize(claims_text[8])
```

[4]:
```
['Says',
 'that',
 'Mitt',
 'Romney',
 "'s",
 'response',
 'to',
 'the',
 'crisis',
 'in',
 'the',
 'auto',
 'industry',
 'was',
 ',',
 '``',
 "''",
 'Let',
 'Detroit',
 'go',
 'bankrupt',
 '.',
 "''",
 "''"]
```

1.2 Mise en miniscule

La mise en miniscule peut-être util dans certain cas.

[5]:

```python
def lowercase(text):
    return text.lower()

lowercase(claims_text[2])
```

[5]:

```
'says hillary clinton ""was literally present when we pressed the reset button
with russia just a few months after russia had invaded georgia.""'
```

1.3 numbers to words

Transformer les nombres en mots.

NB: à utiliser avant `ponctuations` pour éviter de séparer `15.25` en `15` et `25`

[6]:

```python
def number_to_words(text):
    return inflect.engine().number_to_words(text)

#print(number_to_words("15.2"))

def number2words(text):
    tokens = tokenize(text)
    for i,m in enumerate(tokens):
        try:
            float(m)
        except ValueError:
            continue
        else:
            tokens[i] = number_to_words(m)
    return ' '.join(tokens)
print(claims_text[3])
print("\n")
print(number2words(claims_text[3]))
```

```
The former Florida representative and wrote an angry column about 49ers
quarterback Colin Kaepernick on his personal blog.


The former Florida representative and wrote an angry column about 49ers
quarterback Colin Kaepernick on his personal blog .
```

1.4 Traitement des contraction et ponctuation

La suppression des ponctuations peut avoir des conséquences sur le qualité du modèle, par exemple dans la détection des opinions. Il est préferable de traiter d'abord les contractions dans les phrases avant de supprimer les ponctuations.

[7]:
```
import contractions as c
def contractions(text):
    return c.fix(text)

print(contractions("couldn't"))

def ponctuations(text):
    return re.sub(r'[^\w\s]', ' ', text)

print(ponctuations(claims_text[5]))
```

```
could not
Since Oregon s prescription only law took effect  meth lab incidents have
dropped by 96 percent and meth related arrests by 32 percent
```

1.5 Stopwords

Supprimer les mots les plus fréquent de la langue. Dans notre cas:`the`, `a`, `an`, `in` …

NB: Dans les stopwords fournit par défault par NLTK contient les formes de négation.

[8]:
```
stopwords.words('english')
stopwords.words('french')

def remove_stopwords(text_tokenized,language='english'):
    stop_words = set(stopwords.words(language))
```

```python
    return [w for w in text_tokenized if not w in stop_words]

print(claims_text[8])
claim = ponctuations(claims_text[8])
claim = tokenize(claim)
claim = remove_stopwords(claim)
print(' '.join(claim))
```

```
Says that Mitt Romney's response to the crisis in the auto industry was, ""Let
Detroit go bankrupt.""
Says Mitt Romney response crisis auto industry Let Detroit go bankrupt
```

1.6 Stemmatisation

Le stemmatisation (racinisation en français) vise à garder la racine du mot. La racine d'un mot correspond à la partie du mot restante une fois que l'on a supprimé son (ses) préfixe(s) et suffixe(s), à savoir son radical. Plusieurs variantes d'un terme peuvent ainsi être groupées dans une seule forme représentative.

Il existe plusieurs algorithmes de stemmatisation, celui utiliser ici est `SnowBall Stemmer`. Mais il existe aussi `Lancaster Stemmer` qui est considérer comme plus agresif.

[10]:
```python
def stem(text_tokenized,␣
 ↪language='english',stemmer_name='snowball',verbose=False):
    if stemmer_name == 'snowball':
        if verbose:
            print('Snowball stemmer used!')
        stemmer = SnowballStemmer(language=language)
    elif stemmer_name == 'lancaster':
        if language != 'english':
            print("LancasterStemmer do not suport "+language, file=sys.stderr)
            raise ValueError()
        stemmer = LancasterStemmer()
    return [stemmer.stem(term) for term in text_tokenized]

claim = claims_text[0]
print(claim)
claim = ponctuations(claim)
claim = tokenize(claim)
claim = stem(claim,stemmer_name='snowball',verbose=True)
print(' '.join(claim))
```

```
""Michael Dukakis created jobs three times faster"" than Mitt Romney.
Snowball stemmer used!
michael dukaki creat job three time faster than mitt romney
```

1.7 Pos-tagging

L'étiquetage morpho-syntaxique est le processus qui consiste à associer aux mots d'un texte les informations grammaticales correspondantes comme la partie du discours, le genre, le nombre, etc.

[12]:

```python
def pos_tag(words):
    return nltk.pos_tag(words)

claim = claims_text[8]
print(claim)
claim = ponctuations(claim)
claim = tokenize(claim)
pos = pos_tag(claim)
pos
```

```
Says that Mitt Romney's response to the crisis in the auto industry was, ""Let
Detroit go bankrupt.""
```

[12]:

```
[('Says', 'VBZ'),
 ('that', 'IN'),
 ('Mitt', 'NNP'),
 ('Romney', 'NNP'),
 ('s', 'NN'),
 ('response', 'NN'),
 ('to', 'TO'),
 ('the', 'DT'),
 ('crisis', 'NN'),
 ('in', 'IN'),
 ('the', 'DT'),
 ('auto', 'NN'),
 ('industry', 'NN'),
 ('was', 'VBD'),
 ('Let', 'NNP'),
 ('Detroit', 'NNP'),
 ('go', 'VB'),
 ('bankrupt', 'JJ')]
```

1.8 Lemmatisation

La stemmatisation et la lemmatisation sont deux notions proches, mais il y a des différences fondamentales. La lemmatisation a pour objectif de retrouver le lemme d'un mot, par exemple l'infinitif pour les verbes. La racinisation consiste à supprimer la fin des mots, ce qui peut résulter en un mot qui n'existe pas dans la langue.

NB: La lemmatisation foncionnent beaucoup mieux si chaque mot vient avec son tag parts-of-speech (POS).

[13]:

```python
def get_wordnet_pos(treebank_tag):
    if treebank_tag.startswith('J'):
        return wordnet.ADJ
    elif treebank_tag.startswith('V'):
        return wordnet.VERB
    elif treebank_tag.startswith('N'):
        return wordnet.NOUN
    elif treebank_tag.startswith('R'):
        return wordnet.ADV
    else:
        return None

def lem(words, pos_tag=[]):
    lemmatizer = WordNetLemmatizer()
    if len(words) == len(pos_tag):
        lem = []
        for w, pos in zip(words,pos_tag):
            if pos is None:
               lem.append(lemmatizer.lemmatize(w))
            else:
               lem.append(lemmatizer.lemmatize(w, pos=pos))
        return lem
    else:
        return [lemmatizer.lemmatize(w) for w in words]

claim = claims_text[8]
print(claim)
claim = ponctuations(claim)
claim = tokenize(claim)

pos = pos_tag(claim)

claim = lem(claim,)
print(' '.join(claim))
```

```
Says that Mitt Romney's response to the crisis in the auto industry was, ""Let
Detroit go bankrupt.""
Says that Mitt Romney s response to the crisis in the auto industry wa Let
Detroit go bankrupt
```

1.9 Fonction de prétraitement

La fonction de prétraitement va utiliser une combinaison des fonctions citées plus haut. L'idée est donc d'entraîner le modèle avec quelques combinaisons pour voir leurs effets sur le processus d'apprentissage. Cette fonction va prendre en paramètre un texte brut et donner en résultat une liste de token, potentiellement accompagné avec leur tag POS.

Dans le cas pratique, nous avons mis en œuvre une classe TextPreTraitement qui sera plus facile à utiliser qu'une fonction. Notamment, quand on va utiliser les Pipeline.

[49]:
```
def pretraitement(text_brut):
    text = ponctuations(text_brut)

    text_tokenized = tokenize(text)

    text_tagged = pos_tag(text_tokenized)

    text_lematized = lem(text_tokenized,[get_wordnet_pos(p[1]) for p in␣
 ↪text_tagged])

    return [(text_lematized[i],p) for i,(w,p) in enumerate(text_tagged)]

for claim in claims_text:
    print("Claim: ",claim)
    pre = pretraitement(claim)
    for p in pre:
        print(p)
    print("\n")
```

```
Claim:  In 2005 and 2007, "" Joe Straus received a 100 percent rating by NARAL
(the National Abortion and Reproductive Rights Action League).""
('In', 'IN')
('2005', 'CD')
('and', 'CC')
('2007', 'CD')
('Joe', 'NNP')
('Straus', 'NNP')
('receive', 'VBD')
('a', 'DT')
('100', 'CD')
('percent', 'NN')
('rating', 'NN')
('by', 'IN')
('NARAL', 'NNP')
('the', 'DT')
('National', 'NNP')
('Abortion', 'NNP')
('and', 'CC')
('Reproductive', 'NNP')
('Rights', 'NNP')
('Action', 'NNP')
('League', 'NNP')


Claim:  Says that except for Donald Trump, ""every other major party nominee""
for the past 40 years has released their tax returns.
('Says', 'VBZ')
```

```
('that', 'WDT')
('except', 'IN')
('for', 'IN')
('Donald', 'NNP')
('Trump', 'NNP')
('every', 'DT')
('other', 'JJ')
('major', 'JJ')
('party', 'NN')
('nominee', 'NN')
('for', 'IN')
('the', 'DT')
('past', 'JJ')
('40', 'CD')
('year', 'NNS')
('have', 'VBZ')
('release', 'VBN')
('their', 'PRP$')
('tax', 'NN')
('return', 'NNS')


Claim:  Says that 500,000 federal workers -- one-fourth of the federal workforce
-- make more than $100,000 a year.
('Says', 'VBZ')
('that', 'IN')
('500', 'CD')
('000', 'CD')
('federal', 'JJ')
('worker', 'NNS')
('one', 'CD')
('fourth', 'JJ')
('of', 'IN')
('the', 'DT')
('federal', 'JJ')
('workforce', 'NN')
('make', 'VBP')
('more', 'JJR')
('than', 'IN')
('100', 'CD')
('000', 'CD')
('a', 'DT')
('year', 'NN')


Claim:  A Clinton Foundation cargo ship arriving from Africa was raided and
found to contain ""illegal contraband"" in the form of foreign refugees,
narcotics, weapons, and illegal fruits.
```

```
('A', 'DT')
('Clinton', 'NNP')
('Foundation', 'NNP')
('cargo', 'NN')
('ship', 'NN')
('arrive', 'VBG')
('from', 'IN')
('Africa', 'NNP')
('be', 'VBD')
('raid', 'VBN')
('and', 'CC')
('find', 'VBN')
('to', 'TO')
('contain', 'VB')
('illegal', 'JJ')
('contraband', 'NN')
('in', 'IN')
('the', 'DT')
('form', 'NN')
('of', 'IN')
('foreign', 'JJ')
('refugee', 'NNS')
('narcotic', 'NNS')
('weapon', 'NNS')
('and', 'CC')
('illegal', 'JJ')
('fruit', 'NNS')


Claim:  ""North Carolina last year was second in the nation in overdose deaths""
('North', 'NNP')
('Carolina', 'NNP')
('last', 'JJ')
('year', 'NN')
('be', 'VBD')
('second', 'JJ')
('in', 'IN')
('the', 'DT')
('nation', 'NN')
('in', 'IN')
('overdose', 'JJ')
('death', 'NNS')


Claim:  A photograph shows a female wolf protecting a male's throat during a
fight.
('A', 'DT')
('photograph', 'NN')
```

```
('show', 'VBZ')
('a', 'DT')
('female', 'JJ')
('wolf', 'NN')
('protect', 'VBG')
('a', 'DT')
('male', 'JJ')
('s', 'NN')
('throat', 'NN')
('during', 'IN')
('a', 'DT')
('fight', 'NN')


Claim:  Harry Reid Was Injured in a Fight With His Brother
('Harry', 'NNP')
('Reid', 'NNP')
('Was', 'NNP')
('Injured', 'NNP')
('in', 'IN')
('a', 'DT')
('Fight', 'NN')
('With', 'IN')
('His', 'PRP$')
('Brother', 'NN')


Claim:  ""While (Barack) Obama preaches ‘we are our brother’s keeper,’ his
brother and aunt live in real poverty in Kenya.""
('While', 'IN')
('Barack', 'NNP')
('Obama', 'NNP')
('preach', 'VBZ')
('we', 'PRP')
('be', 'VBP')
('our', 'PRP$')
('brother', 'NN')
('s', 'NN')
('keeper', 'IN')
('his', 'PRP$')
('brother', 'NN')
('and', 'CC')
('aunt', 'NN')
('live', 'VBP')
('in', 'IN')
('real', 'JJ')
('poverty', 'NN')
('in', 'IN')
```

```
('Kenya', 'NNP')


Claim:  An inattentive janitor caused several deaths in a hospital when he
disconnected patients' life support systems to plug in a floor polisher.
('An', 'DT')
('inattentive', 'JJ')
('janitor', 'NN')
('cause', 'VBD')
('several', 'JJ')
('death', 'NNS')
('in', 'IN')
('a', 'DT')
('hospital', 'NN')
('when', 'WRB')
('he', 'PRP')
('disconnect', 'VBD')
('patient', 'NNS')
('life', 'NN')
('support', 'NN')
('system', 'NNS')
('to', 'TO')
('plug', 'VB')
('in', 'IN')
('a', 'DT')
('floor', 'NN')
('polisher', 'NN')


Claim:  Says Vince Polistina is ""collecting nearly $70,000 in taxpayer-funded
salaries -- plus a government pension.""
('Says', 'VBZ')
('Vince', 'NNP')
('Polistina', 'NNP')
('be', 'VBZ')
('collect', 'VBG')
('nearly', 'RB')
('70', 'CD')
('000', 'CD')
('in', 'IN')
('taxpayer', 'NN')
('funded', 'JJ')
('salary', 'NNS')
('plus', 'CC')
('a', 'DT')
('government', 'NN')
('pension', 'NN')
```