Word2Vec - NLP

May 24, 2021

1 Tarea 15

1.1 Word2Vec

Miguel Angel Soto Hernandez

[103]:
```
import nltk
from nltk import word_tokenize, sent_tokenize
from nltk.corpus import stopwords, gutenberg
from nltk.stem.porter import *
nltk.download('gutenberg')
nltk.download('punkt')
nltk.download('stopwords')

import string

import gensim
from gensim.models.phrases import Phraser, Phrases
from gensim.models.word2vec import Word2Vec

from sklearn.manifold import TSNE

import pandas as pd
from bokeh.io import output_notebook, output_file
from bokeh.plotting import show, figure
%matplotlib inline
import numpy as np
```

```
[nltk_data] Downloading package gutenberg to /root/nltk_data...
[nltk_data]   Package gutenberg is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

[104]:
```
gutenberg.fileids()
```

[104]:
```
['austen-emma.txt',
 'austen-persuasion.txt',
 'austen-sense.txt',
 'bible-kjv.txt',
 'blake-poems.txt',
 'bryant-stories.txt',
 'burgess-busterbrown.txt',
 'carroll-alice.txt',
 'chesterton-ball.txt',
 'chesterton-brown.txt',
 'chesterton-thursday.txt',
 'edgeworth-parents.txt',
 'melville-moby_dick.txt',
 'milton-paradise.txt',
 'shakespeare-caesar.txt',
 'shakespeare-hamlet.txt',
 'shakespeare-macbeth.txt',
 'whitman-leaves.txt']
```

[105]:
```
texto_1_sents = gutenberg.sents('austen-emma.txt')
texto_2_sents = gutenberg.sents('carroll-alice.txt')
texto_3_sents = gutenberg.sents('shakespeare-macbeth.txt')
```

[106]:
```
print(texto_1_sents)
print(texto_2_sents)
print(texto_3_sents)
```

```
[['[', 'Emma', 'by', 'Jane', 'Austen', '1816', ']'], ['VOLUME', 'I'], ...]
[['[', 'Persuasion', 'by', 'Jane', 'Austen', '1818', ']'], ['Chapter', '1'],
...]
[['[', 'The', 'Tragedie', 'of', 'Macbeth', 'by', 'William', 'Shakespeare',
'1603', ']'], ['Actus', 'Primus', '.'], ...]
```

[107]:
```
stopwords_en = stopwords.words('english') + list(string.punctuation)
print(stopwords_en)
```

```
['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", '!', '"', '#', '$', '%', '&', "'", '(', ')', '*', '+', ',', '-',
'.', '/', ':', ';', '<', '=', '>', '?', '@', '[', '\\', ']', '^', '_', '`', '{',
'|', '}', '~']
```

[108]:
```
def sentencias_minusculas(sentences, stopwords):
  lower_sents = []
  for s in sentences:
    lower_sents.append([w.lower() for w in s if w.lower() not in \
                        stopwords])

  return lower_sents
```

[109]:
```
texto_1_limpio = sentencias_minusculas(texto_1_sents, stopwords_en)
texto_2_limpio = sentencias_minusculas(texto_2_sents, stopwords_en)
texto_3_limpio = sentencias_minusculas(texto_3_sents, stopwords_en)

print(len(texto_1_limpio))
print(len(texto_2_limpio))
print(len(texto_3_limpio))
```

```
7752
3747
1907
```

[110]:
```
def modelo_w2v(sentencias):
  model = Word2Vec(sentences= sentencias, size= 64, sg= 1, window= 10, iter= 5,
                   min_count= 10, workers= 4)
  return model
```

[111]:
```
modelo_1_w2v = modelo_w2v(texto_1_limpio)
modelo_2_w2v = modelo_w2v(texto_2_limpio)
modelo_3_w2v = modelo_w2v(texto_3_limpio)
```

[112]:
```
print(len(modelo_1_w2v.wv.vocab))
print(len(modelo_2_w2v.wv.vocab))
print(len(modelo_3_w2v.wv.vocab))
```

```
1324
782
189
```

[113]:
```
def promedio_w2v(modelo):
  suma = np.array(np.zeros(64))
  promedio = np.array(np.zeros(64))

  for s in modelo.wv.vocab:
    suma = suma + modelo.wv[s]

  for i in range(len(suma)):
    promedio[i] = suma[i] / len(modelo.wv.vocab)

  return promedio
```

[114]:
```
modelo_1_avg = promedio_w2v(modelo_1_w2v)
modelo_1_avg
```

[114]:
```
array([ 5.52392038e-02, -1.26902341e-01, -3.73431573e-03, -2.00760517e-02,
       -7.30943360e-03, -3.32173511e-02, -2.18042993e-02,  4.04326921e-01,
        5.93801550e-02, -9.35255418e-02, -2.18070733e-02, -3.90048847e-02,
        7.10782188e-02, -2.15052229e-01,  4.10560180e-03,  4.01454125e-01,
       -9.91580343e-02,  9.14459083e-02,  1.07784566e-01, -1.15626071e-01,
       -1.01818313e-01,  9.37510247e-02,  1.42589532e-01,  8.15321414e-03,
       -2.44643050e-01,  1.99585682e-01,  4.23764420e-02,  5.05476589e-02,
       -1.26589724e-01,  1.99709376e-01, -1.00174890e-01, -5.19045797e-02,
       -2.97813717e-04,  2.85594495e-01,  3.65665775e-02,  4.59364300e-03,
        3.74848928e-01, -1.49324787e-01, -1.83798676e-01, -1.50291711e-01,
       -2.93644710e-01, -1.66183400e-01,  3.13575660e-01, -1.56795107e-01,
        1.60815713e-01,  2.43472150e-02, -3.26359691e-01,  5.64091321e-03,
        9.04393555e-02,  1.01131371e-01, -4.59289395e-02, -5.06231674e-02,
       -9.21433498e-02,  3.25849016e-02, -7.63569521e-02,  6.67381098e-02,
       -1.35537775e-01,  1.69715671e-01,  1.99330273e-01,  2.84800010e-01,
       -6.74266498e-02,  1.63026106e-03,  7.90108262e-02,  1.35278037e-01])
```

[115]:
```
modelo_2_avg = promedio_w2v(modelo_2_w2v)
modelo_2_avg
```

[115]:
```
array([ 2.59248193e-01, -1.84262789e-03, -1.58546257e-01,  2.11413401e-03,
       -1.70575061e-02, -4.57638342e-02,  1.50715873e-01,  1.70001633e-01,
        2.64427794e-01,  2.05608408e-02,  1.89534153e-01,  7.70949322e-02,
        2.13821971e-01, -8.23904827e-02, -3.18668578e-01,  2.09513731e-01,
       -1.28363508e-01,  1.30501270e-01,  1.36876569e-01,  6.89969379e-02,
       -2.16551469e-01, -1.90766349e-01,  1.03086592e-01, -6.55662101e-02,
        2.71989746e-02, -6.12296784e-02, -2.90217072e-01,  5.95721013e-02,
       -7.39448554e-02,  1.39867274e-01,  3.21203496e-01, -9.55838705e-02,
        2.72275100e-02, -2.14643496e-02,  1.91710538e-01, -5.24607057e-03,
        1.79155363e-01, -3.69843229e-02,  5.52920040e-03, -1.93452257e-01,
        2.14416656e-01, -2.22844202e-01, -9.60733824e-02,  3.60935143e-02,
        2.10276818e-01,  1.13931590e-01, -1.35399527e-01,  1.72135663e-01,
        8.32570556e-04,  1.49481944e-01, -4.56137994e-02,  1.87038267e-02,
       -3.16155623e-01, -7.07304687e-02,  1.58238301e-02,  1.06064503e-01,
```

```
        1.17979796e-04,  2.03973689e-01,  1.44392257e-01,  3.76324978e-01,
        2.19679174e-01, -9.52889148e-03,  1.44918919e-01, -3.70697320e-02])
```

[116]:
```
modelo_3_avg = promedio_w2v(modelo_3_w2v)
modelo_3_avg
```

[116]:
```
array([ 0.02951613,  0.02615745, -0.07069921,  0.02744029, -0.14414912,
        0.03168867,  0.11047114,  0.10380877,  0.08602747, -0.05591987,
       -0.08976793, -0.12190687,  0.14565501, -0.00330582,  0.08080185,
        0.05783781,  0.01613813,  0.07797912, -0.03789483,  0.13196988,
        0.08777385,  0.08051885,  0.00766932,  0.08972675,  0.02913374,
        0.00385416,  0.09662191, -0.10082769, -0.01137729,  0.08415734,
        0.02436703, -0.0179732 , -0.04244671,  0.02008461,  0.017092  ,
       -0.08562819,  0.05109498,  0.02149153,  0.11466543, -0.06374213,
       -0.00066256, -0.00210597,  0.14589805, -0.04279159, -0.06743392,
        0.00861533, -0.19379708, -0.05944377, -0.02074681,  0.11385792,
       -0.08299514, -0.04226911,  0.01321366, -0.01757633,  0.09033296,
        0.03381172,  0.02219296, -0.01025422, -0.08166346,  0.00710133,
        0.03302144, -0.01190986, -0.01933592,  0.09554933])
```

[117]:
```
def crear_tabla(nombre_texto, arreglo):
  df = pd.DataFrame()
  df['texto'] = [nombre_texto for i in range(len(arreglo))]
  df['valores_vector'] = [valor for valor in arreglo]
  return df
```

[118]:
```
tabla_1 = crear_tabla('texto_1', modelo_1_avg)
tabla_1.head()
```

[118]:

	texto	valores_vector
0	texto_1	0.055239
1	texto_1	-0.126902
2	texto_1	-0.003734
3	texto_1	-0.020076
4	texto_1	-0.007309

[119]:
```
tabla_2 = crear_tabla('texto_2', modelo_2_avg)
tabla_2.head()
```

[119]:

	texto	valores_vector
0	texto_2	0.259248
1	texto_2	-0.001843
2	texto_2	-0.158546
3	texto_2	0.002114
4	texto_2	-0.017058

[120]:
```
tabla_3 = crear_tabla('texto_3', modelo_3_avg)
tabla_3.head()
```

[120]:

	texto	valores_vector
0	texto_3	0.029516

```
1  texto_3        0.026157
2  texto_3       -0.070699
3  texto_3        0.027440
4  texto_3       -0.144149
```

[121]:
```
tabla_general = pd.concat([tabla_1, tabla_2, tabla_3], ignore_index=True)
tabla_general.shape
```

[121]:
```
(192, 2)
```

[122]:
```
tabla_pivoteada = tabla_general.groupby(['texto',␣
↪'valores_vector'])['valores_vector']\
                  .agg(['count']).reset_index()\
                  .pivot(index='valores_vector', columns='texto',␣
↪values='count')

tabla_pivoteada.columns.name = None
tabla_pivoteada = tabla_pivoteada.fillna(0)
tabla_pivoteada.head()
```

[122]:

valores_vector	texto_1	texto_2	texto_3
-0.326360	1.0	0.0	0.0
-0.318669	0.0	1.0	0.0
-0.316156	0.0	1.0	0.0
-0.293645	1.0	0.0	0.0
-0.290217	0.0	1.0	0.0

[123]:
```
from scipy.spatial.distance import cosine

def similitud_coseno(a,b):
  distancia = cosine(a,b)
  return 1-distancia

tabla_pivoteada.corr(method=similitud_coseno)
```

[123]:

	texto_1	texto_2	texto_3
texto_1	1.0	0.0	0.0
texto_2	0.0	1.0	0.0
texto_3	0.0	0.0	1.0

[124]:
```
!wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
from colab_pdf import colab_pdf
colab_pdf('Word2Vec  - NLP.ipynb')
```

```
File colab_pdf.py already there; not retrieving.


WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

^C
```

```
[NbConvertApp] Converting notebook /content/drive/My Drive/Colab
Notebooks/Word2Vec  - NLP.ipynb to pdf
[NbConvertApp] Writing 45738 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: [u'xelatex', u'./notebook.tex',
'-quiet']
Traceback (most recent call last):
  File "/usr/local/bin/jupyter-nbconvert", line 8, in <module>
    sys.exit(main())
  File "/usr/local/lib/python2.7/dist-packages/jupyter_core/application.py",
line 267, in launch_instance
    return super(JupyterApp, cls).launch_instance(argv=argv, **kwargs)
  File "/usr/local/lib/python2.7/dist-packages/traitlets/config/application.py",
line 658, in launch_instance
    app.start()
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/nbconvertapp.py", line
338, in start
    self.convert_notebooks()
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/nbconvertapp.py", line
508, in convert_notebooks
    self.convert_single_notebook(notebook_filename)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/nbconvertapp.py", line
479, in convert_single_notebook
    output, resources = self.export_single_notebook(notebook_filename,
resources, input_buffer=input_buffer)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/nbconvertapp.py", line
408, in export_single_notebook
    output, resources = self.exporter.from_filename(notebook_filename,
resources=resources)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/exporters/exporter.py",
line 179, in from_filename
    return self.from_file(f, resources=resources, **kw)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/exporters/exporter.py",
line 197, in from_file
    return self.from_notebook_node(nbformat.read(file_stream, as_version=4),
resources=resources, **kw)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/exporters/pdf.py", line
178, in from_notebook_node
    rc = self.run_latex(tex_file)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/exporters/pdf.py", line
149, in run_latex
    self.latex_count, log_error)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/exporters/pdf.py", line
129, in run_command
    out, _ = p.communicate()
  File "/usr/lib/python2.7/subprocess.py", line 475, in communicate
    stdout = _eintr_retry_call(self.stdout.read)
  File "/usr/lib/python2.7/subprocess.py", line 125, in _eintr_retry_call
```

```
    return func(*args)
KeyboardInterrupt
```

[124]:
```
'File Download Unsuccessful. Saved in Google Drive'
```

[ ]:
```

```