Anexo 10

Proyecto 10: Spinning de Texto

Mg. Luis Felipe Bustamante Narváez

Librerías

In [...
```
import numpy as np
import pandas as pd
import nltk
from nltk import word_tokenize
from nltk.tokenize.treebank import TreebankWordDetokenizer
import asyncio
from tqdm import tqdm
from colorama import Fore, Back, Style
import os
from itertools import islice
from IPython.display import display, Markdown
```

In [...
```
# Descargamos el conjunto de datos del tokenizador en español
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\luis_\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[... True

Cargamos los datos

In [...
```
# Es común que los archivos vengan codificados con ISO
path = 'data/data_larazon_publico_v2.csv'
path_utf = 'data/new_data.csv'
try:
    df = pd.read_csv(path, encoding='utf-8')
    print('Encoding utf-8')
except Exception:
    print('Encoding ISO-8859-1 a utf-8')
    df_iso = pd.read_csv(path, encoding='ISO-8859-1')
    df_iso.to_csv(path_utf, encoding='utf-8', index=False)
#await asyncio.sleep(3) #Espera 3 seg para abrir el nuevo archivo en espera de ser guardad
    df = pd.read_csv(path_utf, encoding='utf-8')
```

```
Encoding utf-8
```

In [...
```
df
```

Out[…

	Unnamed: 0	indi	cuerpo	titular
0	0	0	dos semanas después de su puesta de largo y pr...	el submarino s-80 ya flota
1	1	1	este viernes, el presidente del gobierno, pedr...	calviño y calvo alaban (sin darse cuenta) la g...
2	2	2	el ministro del interior, fernando grande-marl...	el geo de la policía tendrá una nueva sede en ...
3	3	3	son días muy duros para la familia de olivia y...	la madre de las niñas "sobran las palabras par...
4	4	4	sólo quedan 10 presos de eta por recibir los b...	sólo quedan 10 presos de eta por recibir el be...
...	...	...	...	...
58419	58420	18419	la comisión europea inició este un procedimien...	bruselas abre un expediente a españa por no de...
58420	58421	18420	el pleno de la asamblea de madrid ha aprobado ...	aprobado el proyecto de ley para que las mujer...
58421	58422	18421	la comisión de investigación parlamentaria del...	la comisión del alvia arranca escuchando a la ...
58422	58423	18422	erc y pdecat han calificado este jueves de "in...	erc y pdecat piden explicaciones a interior po...
58423	58424	18423	la junta de portavoces del congreso ha acordad...	el congreso aplaza la primera sesión de contro...

58424 rows × 4 columns

In […

```
df.head()
```

Out[…

	Unnamed: 0	indi	cuerpo	titular
0	0	0	dos semanas después de su puesta de largo y pr...	el submarino s-80 ya flota
1	1	1	este viernes, el presidente del gobierno, pedr...	calviño y calvo alaban (sin darse cuenta) la g...
2	2	2	el ministro del interior, fernando grande-marl...	el geo de la policía tendrá una nueva sede en ...
3	3	3	son días muy duros para la familia de olivia y...	la madre de las niñas "sobran las palabras par...
4	4	4	sólo quedan 10 presos de eta por recibir los b...	sólo quedan 10 presos de eta por recibir el be...

Creamos la Serie con las noticias

In […

```
# Tomamos solamente la columna abstract para crear una serie
textos = df['cuerpo']
```

In […

```
textos.head()
```

Out[…

```
0    dos semanas después de su puesta de largo y pr...
1    este viernes, el presidente del gobierno, pedr...
2    el ministro del interior, fernando grande-marl...
3    son días muy duros para la familia de olivia y...
4    sólo quedan 10 presos de eta por recibir los b...
Name: cuerpo, dtype: object
```

In […

```
#muestra de las noticias (solo las primeras 500 palabras)
textos[0][:500]
```

Out[…

```
'dos semanas después de su puesta de largo y presentación en sociedad, el primer submarino s-8
0 para la armada, el s-81 "isaac peral", ha entrado hoy en el agua tras una delicada y larga m
aniobra que se ha retrasado varios días por las condiciones meteorológicas. de esta forma, tra
s completar su construcción 17 años después de que arrancara el programa, navantia ha cumplido
otro importante hito.españa.submarino s-80 tras 17 años y 3.900 millones, el "isaac peral" ya
está aquíespaña.el comandante '
```

Probabilidades

Matriz de conteo

In […

```
# Creamos el diccionario de probabilidad
# key: (w(t-1), w(t+1)), value: {w(t): count(w(t))}
probs = {}
```

In […

```python
# Separador
#variable de formato de la barra de progreso
bar_format_ = (f'{Back.WHITE}{Fore.GREEN}{{l_bar}}{{bar}}{Style.RESET_ALL} '
               f'{Fore.CYAN}{{n_fmt}}/{{total_fmt}} '
               f'[{{elapsed}}<{{remaining}}]{Style.RESET_ALL}'
            )
for doc in tqdm(textos, bar_format=bar_format_, desc='Creando matriz: '):
    #Separamos cada noticia por puntos
    lineas = doc.split('.')
    for linea in lineas:
        #Tokenizamos cada línea
        tokens = word_tokenize(linea, language='spanish')
        #Mostramos los tokens
        #print(tokens)   #Este proceso tarda bastante, se hace a modo de prueba
        #Condicionamos las palabras finales
        if len(tokens) >= 2:
            for i in range(len(tokens) - 2):
                t_0 = tokens[i]  #palabra anterior
                t_1 = tokens[i+1]  #palabra actual
                t_2 = tokens[i+2]  #palabra siguiente
                #Creamos la clave del diccionario
                key = (t_0, t_2)
                #preguntamos si la clave no está en el diccionario
                if key not in probs:
                    #asinamos una clave vacía
                    probs[key] = {}
                #preguntamos si la palabra actual no es una clave
                if t_1 not in probs[key]:
                    #asignamos valor inicial de 1 al diciconario de valores de las probs
                    probs[key][t_1] = 1
                else:
                    #sumamos el valor de aparición de la palabra actual
                    probs[key][t_1] += 1

#mostramos las líneas a modo de prueba
#lineas
```

```
100%|██████████ 58424/58424 [06:03<00:00]
```

In […

```python
# Mostramos el diccionario probs, pero solo una parte para hacer corto el proceso
dict(islice(probs.items(),1))
```

Out[…

```
{('dos', 'después'): {'semanas': 95,
  'años': 283,
  'días': 296,
  'meses': 208,
  'horas': 35,
  'siglos': 4,
  'minutos': 4,
  'décadas': 16,
  'elecciones': 2,
  'día': 3,
  'jornadas': 2,
  'legislaturas': 1,
  'domingos': 1,
  'negocios': 1,
  'pasiones': 1,
  'decenios': 1,
  'iniciativas': 1,
  'dispositivos': 1}}
```

In [...]
```
len(probs)
```

Out[...]
```
4875993
```

Normalización

In [...]
```
# Creamos una copia del diccionario para mantener los datos
d_probs = probs.copy()
#Recorremos las claves y los valores del diccionario probs
for key, d in tqdm(d_probs.items(), bar_format=bar_format_, desc='Normalizando: '):
    #sumamos los valores de repetición de cada una de las palabras
    total = sum(d.values())
    #Recorremos la clave y el valor del diccionario de los valores creado
    for k, v in d.items():
        d[k] = v / total
```

```
Normalizando: 100%|██████████ 4875993/4875993 [00:21<00:00]
```

In [...]
```
# Mostramos el diccionario d_probs, pero solo una parte para hacer corto el proceso
dict(islice(d_probs.items(),1))
```

Out[...]
```
{('dos', 'después'): {'semanas': 0.09947643979057591,
  'años': 0.2963350785340314,
  'días': 0.3099476439790576,
  'meses': 0.21780104712041884,
  'horas': 0.03664921465968586,
  'siglos': 0.004188481675392671,
  'minutos': 0.004188481675392671,
  'décadas': 0.016753926701570682,
  'elecciones': 0.0020942408376963353,
  'día': 0.0031413612565445027,
  'jornadas': 0.0020942408376963353,
  'legislaturas': 0.0010471204188481676,
  'domingos': 0.0010471204188481676,
  'negocios': 0.0010471204188481676,
  'pasiones': 0.0010471204188481676,
  'decenios': 0.0010471204188481676,
  'iniciativas': 0.0010471204188481676,
  'dispositivos': 0.0010471204188481676}}
```

In [...]
```
len(d_probs)
```

Out[...]
```
4875993
```

Ejemplo de Detokenización

Permite volver a unir los tokens en frases, por ejemplo:

In [...]
```
detokenizar = TreebankWordDetokenizer()
ejemplo = 'Bootcamp de Inteligencia Artificial'
print(f'Frase original: {ejemplo}')
token_ejemplo = word_tokenize(ejemplo, language='spanish')
print(f'Frase tokenizada: {token_ejemplo}')
detoken_ejemplo = detokenizar.detokenize(token_ejemplo)
print(f'Frase Detokenizada: {detoken_ejemplo}')
```

```
Frase original: Bootcamp de Inteligencia Artificial
Frase tokenizada: ['Bootcamp', 'de', 'Inteligencia', 'Artificial']
Frase Detokenizada: Bootcamp de Inteligencia Artificial
```

Spinner

In [...

```python
# Función de prueba para una palabra random
def sample_word(d):
    p0 = np.random.random()
    cumulative = 0
    for key, p in d.items():
        cumulative += p
        if p0 < cumulative:
            return key
```

In [...

```python
# Función spinner para una línea
# CADA COMENTARIO DONDE ESTÁ EL RETURN ES UN EJEMPLO PARA IR ANALIZANDO EL CÓDIGO
def spin_line(linea, imp):
    tokens = word_tokenize(linea, language='spanish')
    i = 0
    salida = [tokens[0]]
    #return salida #ejemplo de ejecución --- comentar
    if len(tokens) >= 2:
        while i < (len(tokens) - 2):
            t_0 = tokens[i]   #palabra anterior
            t_1 = tokens[i+1] #palabra actual
            t_2 = tokens[i+2] #palabra siguiente
            #creamos la clave
            key = (t_0, t_2)
            #creamos el diccionario de distribución
            p_dist = d_probs[key]
            #i = 1100000   #Para desbordar el while ----- comentar
            #return p_dist   #ejemplo de ejecución  ---- comentar
            #Cuando el diccionario tenga más de una palabra y un spinning del x%
            if len(p_dist) > 1 and np.random.random() < 0.3:
                #selecciona una palabra al azar de la función de prueba de palabras
                middle = sample_word(p_dist)
                #i = 1100000   #Para desbordar el while ----- comentar
                #return middle   #ejemplo de ejecución  ---- comentar

                #Validamos si deseamos mostrar la palabra de cambio automáticamente
                # Si imp es True, muestra el texto cambiado
                # Si imp es False, muestra la palabra actual y el cambio que sugiere
                if imp:
                    #agregamos la palabra nueva en la posición t_1
                    salida.append(middle)
                    #agregamos la palabra t_2, que va al final
                    salida.append(t_2)
                    #movemos el cursor 2 posici. para que no haga 2 spin en 2 pal. seguidas
                    i += 2
                else:
                    #agregamos a la salida la palabra t_1, es decir la que queremos cambiar
                    salida.append(t_1)
                    #agregamos, para visualizar, la palabra por la que nos va a cambiar
                    salida.append('<' + middle + '>')
                    #agregamos la palabra t_2, que va al final
                    salida.append(t_2)
        #movemos el cursor dos posici. para que no haga dos spin en 2 pal seguidas
                    i += 2
            #en caso que el diccionario sea <= 1 o que el random no entre al spinner
            else:
    #agregamos la palabra siguiente y ubicamos el cursor en la siguiente palabra
```

```
                salida.append(t_1)
                i += 1
        # si ya estamos en la última palabra a poner a prueba
        if i == len(tokens) - 2:
            #agregamos la última palabra al diccionario
            salida.append(tokens[-1])
    # retornamos la salida detokenizada ya que es una lista ESTE NO SE COMENTA, ES EL FIN
        detoken = detokenizar.detokenize(salida)
        return detoken
```

In [...
```
# Función spinner para recorrer el documento
def spin_document(doc, imp):
    lineas = doc.split('.')
    output = []
    for linea in lineas:
        if linea:
            new_line = spin_line(linea, imp)
        else:
            new_line = linea
        output.append(new_line)
    #corregimos el posible error de tener cadenas en None
    try:
        return '\n'.join(output)
    except Exception:
        return '\n'.join(filter(None, output))
```

In [...
```
#Código para pruebas de creación
#spin_document('dos años después cómo están')
#spin_line('dos años después cómo están')
```

Texto (noticia) de prueba para el modelo

In [...
```
#Recordemos qué tenía nuestro df textos
textos.head()
```

Out[...
```
0    dos semanas después de su puesta de largo y pr...
1    este viernes, el presidente del gobierno, pedr...
2    el ministro del interior, fernando grande-marl...
3    son días muy duros para la familia de olivia y...
4    sólo quedan 10 presos de eta por recibir los b...
Name: cuerpo, dtype: object
```

In [...
```
#seleccionamos un índice cualquiera de alguna noticia del df textos
i = np.random.choice(textos.shape[0])
display(Markdown('---'))
display(Markdown(f'**Índice seleccionado:** {i}'))
display(Markdown('---'))
#tomamos el texto que se encuentra en dicho índice
doc = textos.iloc[i]
#Recortamos el texto, solo para mostrarlo; no se altera el texto inicial
doc_recortado = doc.split() #separamos el texto en palabras
doc_recortado = ' '.join(doc_recortado[:100])
display(Markdown(f'**Texto seleccionado:**'))
print(f'{doc_recortado}...')
display(Markdown('---'))



#Generamos el Spinning Article - Text
```

```
imp = True
new_doc = spin_document(doc, imp)



#Recortamos el nuevo texto, solo para mostrarlo; no se altera el texto generado por el spin
new_doc_recortado = new_doc.split() #separamos el texto en palabras
new_doc_recortado = ' '.join(new_doc_recortado[:100])
display(Markdown(f'**Texto Spinning:**\n\n'))
print(f'{new_doc_recortado}...')
display(Markdown('---'))
```

Índice seleccionado: 35576

Texto seleccionado:

```
un centenar de taxistas se han concentrado a las 8,00 horas de este viernes en la entrada del c
ementerio de la almudena de madrid y, pasada esta hora, tenían bloqueada la zona. esta acción s
e produce en el marco de las protestas que está protagonizando el colectivo desde el pasado lun
es por la regulación del sector. se trata de la primera acción del día, que ira acompañada de o
tras como el comienzo de la huelga de hambre que van a iniciar 16 compañeros a partir de las 1
0,00 horas de hoy en los alrededores de ifema, centro de operaciones de...
```

Texto Spinning:

```
un centenar de taxistas se han concentrado a las 2,00 horas de este campo en la entrada del cem
enterio de la almudena de presentarlo y la pasada esta hora, tenían bloqueada la zona esta cifr
a se produce en el toque de las protestas que está protagonizando el colectivo según el próximo
lunes por una regulación del sector se trate de la primera acción del día después que ira acomp
añada de otras sin el comienzo de la historia de hambre que maltratan a iniciar 16 compañeros a
cambio de las 10,00 horas de hoy mantener los sistemas de ifema, instructor de...
```

Errores de tipo NoneType - Análisis

Cuando existe un valor None en el output de la función spin_document, no se puede definir el nuevo texto sugerido. Para solucionar, basta con filtrar el output antes de hacer el join.

```
'\n'.join(filter(None, output))
```

Conclusiones

El Article Spinning, permite realizar cambios de palabras con el fin de brindar otra opción a un texto ya construido y cambiarle sus palabras de modo que conserve la idea contextual, pero con otro estilo de escritura. El uso de N-Grams através de las cadenas de Markov, permiten utilizar las probabilidades de ocurrencia de una palabra cuando ésta se encuentra en medio de dos palabras previamente entrenadas. Aunque el modelo es bueno, se requiere de un filtro de fuentes más preciso de un tema en específico, pero este es un sencillo ejemplo que nos deja el desafío de usar Spinning Text dentro de NPL.

Mg. Luis Felipe Bustamante Narváez