NUBE DE PALABRAS

SISTEMAS DE BIG DATA

24/11/24 – IES Fernando Wirtz

Rafael Chamorro Maceiras

Fecha	Motivo del cambio
24/11/23	Versión inicial

Índice

Crear nube de etiquetas

Fai un pull ao repositorio de SBD e abre o notebook "nube_de_respostas.ipynb" da tarefa T1.2.

Executa o exemplo e logra que funcione seguindo as instruccións. Fai un documento en PDF no que se vexa a instalación de dependencias e a execución explicada do notebook.

Pandas y numpy

```
(base) rchamac@rchamac-HP-EliteBook-820-G1:~/Documentos/SBD/T1.2$ conda activate bigdata
(bigdata) rchamac@rchamac-HP-EliteBook-820-G1:~/Documentos/SBD/T1.2$ conda install numpy pandas
```

wordcloud

```
(bigdata) rchamac@rchamac-HP-EliteBook-820-G1:~/Documentos/SBD/T1.2$ conda install -c conda-forge wordcloud
Collecting package metadata (current_repodata.json): \
```

nltk

```
(bigdata) rchamac@rchamac-HP-EliteBook-820-G1:~/Documentos/SBD/T1.2$ conda install nltk
Collecting package metadata (current_repodata.json): |
```

Colle do foro da aula virtual as diferentes definicións de Big Data e méteas nun arquivo "defino.txt"

Defino.txt

```
1
2   Big data es el conjunto de tecnologías que obtiene, limpia y procesa datos de gran tamaño o qu
3   base de datos relacional, para poder trabajar con ellos de una manera “inteligente” .
4
5   El bigdata es una red que analiza e interpreta un gran volumen de datos
6
7   Chámase Big Data a grandes conxuntos de datos etiquetados e a o seu uso para diferentes aplica
8   Enténdese que o volume de datos é demasiado grande para ser xestionado de forma convencional.
9
10  Grandes cantidades de datos.
11
12  Big Data fai referencia a grandes cantidades de datos
13  dos cales se poden sacar estadísticas e funcionalidades dependentes destas (computación)
14  que sin chegar a ese tamaño non o permítirian.
15
16  Imaxe de Usuario eliminado
17
```

Cambia o código para que lea ese arquivo e xenere a nube de palabras a partir del: https://ellibrodepython.com/leer-archivos-python

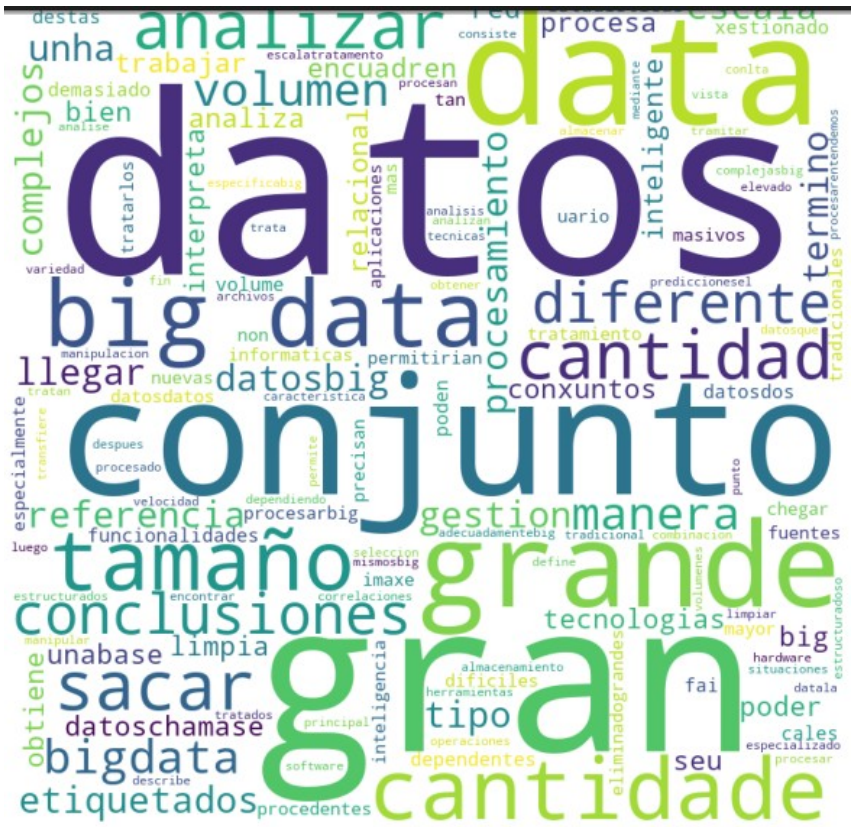
```
texto = ""
fichero = open('Defino.txt','r')
try:
    linea = fichero.readline()
    while linea != '':
        texto = texto + linea
        linea = fichero.readline()

finally:
    fichero.close()
```

Elimina tódolos símbolos de puntuación novos engandíndoos ao array correspondente no código.

```
punctuation=[]
for s in string.punctuation:
    punctuation.append(str(s))
sp_punctuation = ["¿", "¡", "“", "”", "…", ":", "", "»", "«",".","\n",">","/"]
```

Ejecutamos el código e al final tenemos nuestra nube de datos



Crear nube de palabras de una noticia

Colle unha noticia recente e LONGA dun periódico. Indica a URL. Cambia o código para coller directamente do medio o texto

Añado la URL de la noticia a la variable medioDigital para utilizarla en el http.request

```
ua = "Mozilla/5.0 (Linux; U; Android 2.2; en-us; Nexus One Build/FRF91) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/5
h = {"User-Agent": ua}

http = urllib3.PoolManager()


#medioDigital= "http://20minutos.es"
#medioDigital= "http://www.elpais.com"
#medioDigital= "http://www.elmundo.es"
#medioDigital= "http://www.lavozdegalicia.com"

medioDigital = 'https://www.lavozdegalicia.es/noticia/galicia/2023/11/22/gobierno-fija-servicios-minimos-72-alta-velocidad-65-media-dista
r = http.request('GET', medioDigital, fields=None, headers=h)
```

Me devuelve un objeto HTTPResponse

```
<urllib3.response.HTTPResponse at 0x7f9a4c50afa0>
```

Con la librería BeautifulSoup obtenemos el texto de la noticia

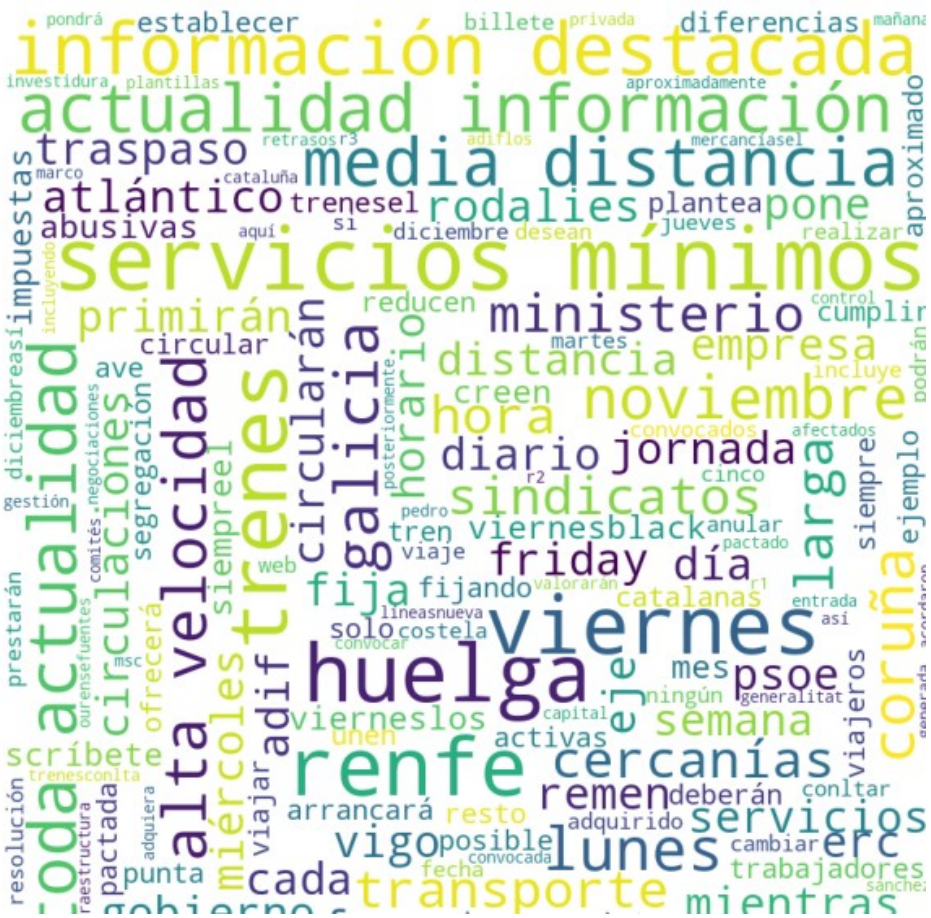
```
web_solotexto = BeautifulSoup(r.data).get_text()

salida = ''
#excluirlineas=4

for linea in web_solotexto.split('\n'):
    aux=linea.strip()
    if aux and len(aux) > 50:
        salida += aux + '\n'
        #if not aux.startswith('Comentarios ('):
        #    excluirlineas=excluirlineas-1
        #    if (excluirlineas < 0):
        #        vartext2 = vartext2 + aux + '\n'
```

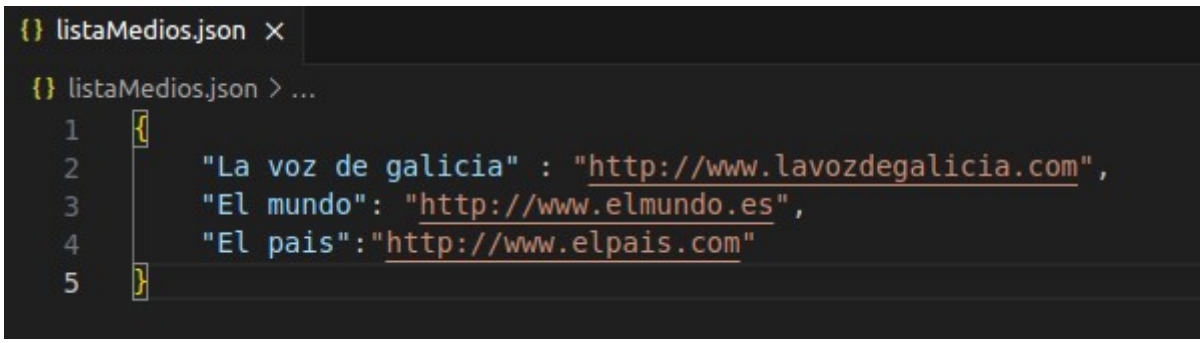
```
'El Gobierno fija servicios mínimos del 72 % en alta velocidad y del 65 % en med
```

Una vez que ya tenemos el texto la aplicación funciona como en el apartado anterior



Crear nube de palabras de titulares

Guardamos en un archivo JSON la lista de diarios para hacer el scrapping



Abrimos el archivo con los datos y guardamos las url's de los medios en una lista como lo teníamos en el ejercicio anterior

```
import json
listaMedios = []
with open("listaMedios.json","r") as j:
    lista_Medios = json.load(j)


for medio in lista_Medios.keys():
    #print(lista_Medios[medio])
    listaMedios.append(lista_Medios[medio])
```

* Para esta tarea hemos refactorizado el código en funciones .

```
noticias = obtener_noticias(listaMedios)
#noticias = quitar_HTML(noticias)
noticias = limpiar_texto(noticias)
lista_palabras =generar_lista_palabras(noticias)
df = contar_palabras(lista_palabras)
#funciones.plot_bar(data=df, top=5)
generar_nube_palabras(noticias)
✓ 7.2s
```

Para obtener los titulares enviamos la lista de medios a la función obtener_noticias(lista), esta recorre la lista obteniendo los titulares de cada diario (en este caso obtuve los h4 de los periódicos, pero sería necesario comprobar en cada caso la etiqueta correspondiente), devolviendo un string con todos los titulares

```python
def obtener_noticias(lista_medios):
    ua = "Mozilla/5.0 (Linux; U; Android 2.2; en-us; Nexus One Build/FF
    h = {"User-Agent": ua}
    http = urllib3.PoolManager()
    web_solotexto=""
    for item in lista_medios:
        r = http.request('GET',item,fields=None,headers=h)
        sopa = BeautifulSoup(r.data, "html.parser")
        titulares = sopa.find_all('h4')
        for titular in titulares:
            web_solotexto += titular.get_text().strip()

    salida = ''

    for linea in web_solotexto.split('\n'):
        aux=linea.strip()
        if aux and len(aux) > 50:
            salida += aux + '\n'

    return salida
```

El resto de la aplicación funciona como lo expuesto anteriormente.