Python

Procesamiento de Lenguaje Natural

Nubes de Palabras

Rogelio Ferreira Escutia

Profesor / Investigador
Tecnológico Nacional de México
Campus Morelia

Nubes de Palabras

Nubes de Palabras

- **Una forma más gráfica de representar y analizar es cuando se utilizan las “Nubes de Palabras” (“WordCloud” en inglés).**

- **Éstas nos representan de manera gráfica las palabras que mas se repiten en un texto y la importancia de acuerdo al tamaño de las letras que se imprimen en pantalla.**

Nube de Palabras (1)

- **Se hará la nube de palabras a partir del texto de una página Web (Wikipedia: Ciencia de Datos):**

Nube de Palabras (2)

- **Bibliotecas:**

```
# Bibliotecas a utilizar
import wikipedia                                # Extraer y procesar de Wikipedia
import re                                       # Manejo de expresiones regulares
import nltk                                     # Para procesar lenguaje natural
from nltk.tokenize import word_tokenize         # Para Tokenizar un texto
from nltk.corpus import stopwords               # Cargar las"Stopwords" del español
from wordcloud import WordCloud, STOPWORDS      # Crear nubes de ideas
import matplotlib.pyplot as plt                 # Graficar
```

Nube de Palabras (3)

- **Se seleccionará Wikipedia en Español, se cargará la página seleccionada y se imprimirá en pantalla**

```
# Seleccionar Wikipedia en Español
wikipedia.set_lang("es")

# Extraer información de una página de Wikipedia
texto = wikipedia.summary("Ciencia de Datos")

# Impresión completa del texto
print("\nTexto Completo:\n\n", texto)
```

Nube de Palabras (4)

- **Eliminamos símbolos y caracteres especiales y "tokenizamos" el texto:**

```
# Eliminar simbolos y caracteres especiales usando expresiones regulares
texto_sin_simbolos = re.sub(r'[^\w\s]', '', texto)

# Convertimos a tokens todo el texto y lo imprimimos en pantalla
tokens_de_mi_texto = word_tokenize(texto_sin_simbolos)
print('\nImpresión de todos los tokens del texto:\n\n', tokens_de_mi_texto)
print('\n Tokens Totales: ', len(tokens_de_mi_texto))
```

Nube de Palabras (5)

- **Cargamos las “StopWords” y hacemos el filtrado:**

```
# Cargamos las "stopwords" del español (las palabras que no nos aportan información)
palabras_vacias = set(stopwords.words('spanish'))

# Filtramos los tokens eliminando las "stopwords"
lista_final = []
for palabra in tokens_de_mi_texto:
    if palabra not in palabras_vacias:
        lista_final.append(palabra)
```

- **Imprimimos los Tokens:**

```
# Impresión final de las palabras relevantes del texto (Tokenizar)
print('\nLista Final eliminando las palabras vacías (no relevantes):\n\n', lista_final)
print("\nTotal de Tokens sin Stopwords: ", len(lista_final))
```

Nube de Palabras (6)

- **Convertimos toda nuestra lista de “tokens” en un solo texto, ya que para generar la nube (con la biblioteca “WordCloud”) se le tienen que dar los datos en este formato y lo imprimimos en pantalla:**

```
# Convertir la lista de tokens a un solo texto separados por un espacio
texto_final = " ".join(lista_final)

# Imprimir el texto final
print("texto final:\n", texto_final)
```

Nube de Palabras (7)

- **De acuerdo a los parámetros que nos pide la biblioteca “WordCloud”, configuramos la pantalla de salida de nuestra “Nube de Palabras” con los siguientes datos:**

```
# Formato de la "Nube de Ideas"
nube_de_ideas = WordCloud(
    width = 500,
    height = 500,
    random_state = 1,
    background_color = "salmon",
    colormap= "Pastel1",
    collocations = False,
    stopwords = STOPWORDS,
).generate(texto_final)
```

Nube de Palabras (8)

- **Y por último configuramos el gráfico y lo generamos:**

```
# Graficar (generar la "Nube de Ideas")
plt.imshow(nube_de_ideas, interpolation="bilinear")
plt.axis("off")
plt.show()
```

Rogelio Ferreira Escutia

Profesor / Investigador
Tecnológico Nacional de México
Campus Morelia

rogelio.fe@morelia.tecnm.mx

rogeplus@gmail.com

xumarhu.net

@rogeplus

https://www.youtube.com/channel/UC0on88n3LwTKxJb8T09sGjg

rogelioferreiraescutia