Nombre: Gabriel Muñoz Marcelo Callisaya
CI: 9873103

DAT 261 - Procesamiento del lenguaje natural

Tarea 2.

1. Considera el texto: Rosa blanca: Los alemanes que intentaron derrocar a Hitler disponible en: https://www.bbc.com/news/magazine-21521060
2. Tokenizar las palabras con NLTK
3. Tokenizar las palabras con spaCy
4. Determinar la diversidad léxica
5. Comparación: Analiza las diferencias. ¿Cómo operan NLTK y SpaCy?

Tokenización con NLTK:

Para la tokenización por palabras con NLTK, se importa la función `word_tokenize` de `nltk.tokenize`. Para manejar mejor el texto, se lo almacena en un archivo `textoBBC.txt` el cual se leerá en el programa de python.

```
from nltk.tokenize import word_tokenize

with open('textoBBC.txt', 'r', encoding='utf-8') as archivo:
    texto = archivo.read()

palabras = word_tokenize(texto)
print(palabras)
```

La ejecución parcial del programa es:

```
['Seventy', 'years', 'ago', 'today', ',', 'three', 'German',
'students', 'were', 'executed', 'in', 'Munich', 'for', 'leading',
'a', 'resistance', 'movement', 'against', 'Hitler', '.', 'Since',
'then', ',', 'the', 'members', 'of',

...

'up', 'to', 'the', 'age', 'of', '86', '.', 'Her', 'friend',
'Alexander', 'Schmorell', 'was', 'made', 'a', 'saint', 'by', 'the',
'Russian', 'Orthodox', 'church', 'in', '2012', '.', 'He', 'would',
'have', 'laughed', 'out', 'loud', 'if', 'he', "'d", 'known', ',',
'says', 'Furst-Ramdohr', '.', 'He', 'was', "n't", 'a', 'saint', '-',
'he', 'was', 'just', 'a', 'normal', 'person', '.']
```

Como se puede ver, la función `word_tokenize` también tokeniza símbolos como comas, puntos y guiones, aunque cuando un guión no está separado por espacios, se toma en cuenta como parte de la palabra; también toma por separado las contracciones (wasn’t)

Tokenización con SpaCy:

Para la tokenización con SpaCy, se usa un modelo «blank» en inglés, se lee el mismo archivo `textoBBC.txt`.

```
import spacy
nlp = spacy.blank("en")

with open('textoBBC.txt', 'r', encoding='utf-8') as archivo:
    texto = archivo.read()

doc = nlp(texto)

for token in doc:
    print(token)
```

La ejecución parcial del programa es:

```
Seventy
years
ago
today
,
three
German
students
were
executed
in
Munich
for

...

Ramdohr
.
He
was
n't
a
saint
-
he
was
just
a
normal
person
.
```

SpaCy imrpime los tokens en líneas separadas, pero esta es una diferencia únicamente visual. Como con NLTK, SpaCy reconoce símbolos como comas, puntos y guiones.

Diversidad léxica:

Para encontrar la diversidad léxica, se sigue la fórmula:

$$\text{Diversidad léxica} = \frac{\text{Tipos}}{\text{Tokens}}$$

Donde los tipos son como los tokens pero sin repeticiones y la cantidad de tokens no puede ser cero.

Se usaron programas de python para determinar la cantidad de tipos, tanto con NLTK como con SpaCY.

En NLTK:

```
import nltk
from nltk.tokenize import word_tokenize

with open('textoBBC.txt', 'r', encoding='utf-8') as archivo:
    texto = archivo.read()

Tokens = word_tokenize(texto.lower())

tipos = set(Tokens)

tokens = len(Tokens)

ttr = len(tipos) / tokens if tokens > 0 else 0

print(f"Tipos: {len(tipos)}")
print(f"Tokens: {tokens}")
print(f"Diversidad léxica: {ttr:.4f}")
```

Corrida:

```
Tipos: 410
Tokens: 1062
Diversidad léxica: 0.3861
```

En SpaCy:

```
import spacy

nlp = spacy.blank("en")

with open('textoBBC.txt', 'r', encoding='utf-8') as archivo:
    texto = archivo.read()
```

```
doc = nlp(texto.lower())

Tokens = [token.text for token in doc if not token.is_space]

tipos = set(Tokens)

tokens = len(Tokens)
ttr = len(tipos) / tokens if tokens > 0 else 0

print(f"Tipos: {len(tipos)}")
print(f"Tokens: {tokens}")
print(f"Diversidad léxica: {ttr:.4f}")
```

Corrida:

```
Tipos: 412
Tokens: 1098
Diversidad léxica: 0.3752
```

En porcentajes, los resultados son:

- NLTK: 38.61%
- SpaCy: 37.52%

Comparación:

Aunque ambas librerías mostraron resultados similares, hubieron algunas diferencias pequeñas en la cantidad de tipos y tokens, lo cual afecto minimamente a la diversidad léxica.

Las diferencias se deben a la forma de tokenizar los textos por parte de cada librería. En el texto, se habla de Liselotte Furst-Ramdohr, cuyo apellido lleva un guión, en NLTK, se considera a su apellido como un solo token, mientras que en SpaCy, se consideran como tres (“Furst”, “-”, “Ramdohr”). Cuando un guión está rodeado por espacios, ambas librerías lo tratan igual, como un token aparte, pero si no está con espacios, se genera la diferencia entre librerías.

En el texto aparecen otros guiones sin espacios, como `99-year-old`, `best-known` u `Oscar-nominated`. Estos casos son los responsables de la discrepancia entre la cantidad encontrada de tipos y tokens entre ambas librerías.

En conclusión, ambas librerías manejan la tokenización de una forma muy similar, aunque con pequeñas diferencias con casos especiales como con los guiones, los resultados al calcular la diversidad léxica son prácticamente iguales.

El código usado para tokenizar fue adaptado de https://www.geeksforgeeks.org/nlp/tokenization-using-spacy-library/ y https://www.geeksforgeeks.org/nlp/spacy--for-natural-language-processing/