1B2. Práctica: Cálculo de entropía

Alumno: **Luis Fernando Izquierdo Berdugo**

Materia: **Procesamiento de Información**

Fecha: **3 de Septiembre de 2024**

Instrucciones:

Preprocesar el texto y convertirlo a minúsculas, quitar acentos y los siguientes caracteres:

`;:,.\\-\"'/()[]¿?¡!{}~<>|«»——’\t\n\r`

1. Calcular la entropía global a nivel de carácter de los documentos text_1 al text_5 de manera independiente.
2. Calcular la entropía global a nivel de palabra de los documentos libro_1 y libro_2 de manera independiente. Primero, sin quitar stopwords y luego quitándolas.

Preprocesamiento de los textos

Para esta actividad, se hizo el procesamiento siguiente:

- Pasado de los textos a minúsculas.
- Eliminación de caracteres especiales.

In [ ]:
```
import re

special = ";:,.\\-\"'/()[]¿?¡!{}~<>|«»——’\t\n\r"

file = open("text_1.txt", "r")
text_1 = file.read()
file.close()
text_1 = text_1.lower()
text_1 = re.sub(r"[;:,.\-\"'/\(\)\[\]¿?¡!\{\}~<>|«»——’\t\n\r]", "", text_1)

file = open("text_2.txt", "r")
text_2 = file.read()
file.close()
text_2 = text_2.lower()
text_2 = re.sub(r"[;:,.\-\"'/\(\)\[\]¿?¡!\{\}~<>|«»——’\t\n\r]", "", text_2)

file = open("text_3.txt", "r")
text_3 = file.read()
file.close()
text_3 = text_3.lower()
text_3 = re.sub(r"[;:,.\-\"'/\(\)\[\]¿?¡!\{\}~<>|«»——’\t\n\r]", "", text_3)

file = open("text_4.txt", "r")
text_4 = file.read()
file.close()
text_4 = text_4.lower()
text_4 = re.sub(r"[;:,.\-\"'/\(\)\[\]¿?¡!\{\}~<>|«»——’\t\n\r]", "", text_4)

file = open("text_5.txt", "r")
text_5 = file.read()
file.close()
text_5 = text_5.lower()
text_5 = re.sub(r"[;:,.\-\"'/\(\)\[\]¿?¡!\{\}~<>|«»——’\t\n\r]", "", text_5)
```

Preprocesamiento de los libros

En el caso de los libros, se harán dos procesamientos, uno que incluye quitar stopwords y otro que no, de cualquier manera, ambos siguen el siguiente proceso:

- Transformación a minúsculas
- Eliminación de acentos
- Eliminación de caracteres especiales

De igual forma, se descargan las stopwords del módulo nltk y se crea una función para eliminarlas. También se crea una función para eliminar los acentos.

In [ ]:
```
from nltk.corpus import stopwords
import nltk
import ssl
```

```
import unicodedata

try:
    _create_unverified_https_context = ssl._create_unverified_context
except AttributeError:
    pass
else:
    ssl._create_default_https_context = _create_unverified_https_context

nltk.download('stopwords')
_STOPWORDS = stopwords.words('spanish')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/izluis/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [ ]:
```
def remove_stopwords(text):
    text_nostop = []
    for word in text.split():
        if word in _STOPWORDS:
            continue
        else:
            text_nostop.append(word)
    return ' '.join(text_nostop)

def remove_accents(text):
    return ''.join(c for c in unicodedata.normalize('NFD', text) if unicodedata.category(c) != 'Mn')
```

In [ ]:
```
file = open("libro_1.txt", "r")
libro_1 = file.read()
file.close()
libro_1 = libro_1.lower()
libro_1_nostop = remove_stopwords(libro_1)
libro_1_nostop = remove_accents(libro_1_nostop)
libro_1_nostop = re.sub(r"[\t\n]", " ", libro_1_nostop)
libro_1_nostop = re.sub(r"[;:,.\-\"'/\(\)\[\]¿?¡!\{\}~<>|«»——’\r]", "", libro_1_nostop)
libro_1 = remove_accents(libro_1)
libro_1 = re.sub(r"[\t\n]", " ", libro_1)
libro_1 = re.sub(r"[;:,.\-\"'/\(\)\[\]¿?¡!\{\}~<>|«»——’\r]", "", libro_1)


file = open("libro_2.txt", "r")
libro_2 = file.read()
file.close()
libro_2 = libro_2.lower()
libro_2_nostop = remove_stopwords(libro_2)
libro_2_nostop = remove_accents(libro_2_nostop)
libro_2_nostop = libro_2_nostop.lower()
libro_2_nostop = re.sub(r"[\t\n]", " ", libro_2_nostop)
libro_2_nostop = re.sub(r"[;:,.\-\"'/\(\)\[\]¿?¡!\{\}~<>|«»——’\t\n\r]", "", libro_2_nostop)
libro_2 = remove_accents(libro_2)
libro_2 = re.sub(r"[\t\n]", " ", libro_2)
libro_2 = re.sub(r"[;:,.\-\"'/\(\)\[\]¿?¡!\{\}~<>|«»——’\r]", "", libro_2)
```

Inciso 1 - Entropía en Caracteres.

Para este caso, la variable aleatoria independiente será la probabilidad de ocurrencia de los caracteres presentes en el texto; entonces la probabilidad que se tomará es la frecuencia del caracter entre el total de los caracteres de cada texto.

Se crea una función para el conteo de caracteres. Esta irá por cada caracter del texto, añadirá uno al total de caracteres en el teto y también añadirá 1 al conteo de caracter individual.

In [ ]:
```
def charCount(text):
    conteo = {}
    total = 0
    for character in text:
        total += 1
        conteo[character] = conteo.get(character, 0) + 1
    return conteo, total
```

Se hace el cálculo de frecuencias por cada texto, así como se obtiene el total de caracteres por cada texto.

In [ ]:
```
frecuencia_txt1, total_txt1 = charCount(text_1)
frecuencia_txt2, total_txt2 = charCount(text_2)
frecuencia_txt3, total_txt3 = charCount(text_3)
frecuencia_txt4, total_txt4 = charCount(text_4)
frecuencia_txt5, total_txt5 = charCount(text_5)
```

Para el cálculo de la entropía, se tomará la frecuencia y el total de caracteres para buscar la probabilidad y también usar la fórmula de la entropía global:

$$H(x) = -\sum_{x \in X} p(x) log_2 p(x)$$

In [ ]:
```python
import math

def calcEntr(freq, total):
    entr = 0
    for char in freq.values():
        prob = char/total
        entr -= (prob * math.log2(prob))
    return entr
```

Ya con la función de entropía, se hace el cálculo de esta para cada uno de los textos

In [ ]:
```python
entropia_txt1 = calcEntr(frecuencia_txt1, total_txt1)
entropia_txt2 = calcEntr(frecuencia_txt2, total_txt2)
entropia_txt3 = calcEntr(frecuencia_txt3, total_txt3)
entropia_txt4 = calcEntr(frecuencia_txt4, total_txt4)
entropia_txt5 = calcEntr(frecuencia_txt5, total_txt5)

print(f"La entropía a nivel de carácter del Texto 1 es {entropia_txt1}")
print(f"La entropía a nivel de carácter del Texto 2 es {entropia_txt2}")
print(f"La entropía a nivel de carácter del Texto 3 es {entropia_txt3}")
print(f"La entropía a nivel de carácter del Texto 4 es {entropia_txt4}")
print(f"La entropía a nivel de carácter del Texto 5 es {entropia_txt5}")
```

```
La entropía a nivel de carácter del Texto 1 es 3.1681820267348035
La entropía a nivel de carácter del Texto 2 es 3.188483436190213
La entropía a nivel de carácter del Texto 3 es 3.3242783480628573
La entropía a nivel de carácter del Texto 4 es 3.2153472642292873
La entropía a nivel de carácter del Texto 5 es 3.265906719965582
```

Inciso 2 - Entropía en palabras

El procesamiento será bastante similar al de la entropía en caracteres, la principal diferencia será que la variable aleatoria independiente será la probabilidad de ocurrencia de una palabra en el texto.

Debido a lo anterior, se creó una función que cuenta el total de palabras en el texto y la frecuencia de cada una de ellas.

In [ ]:
```python
def wordCount(text):
    words = re.findall(r'\w+', text)
    conteo = {}
    total = 0
    for word in words:
        total += 1
        conteo[word] = conteo.get(word, 0) + 1
    return conteo, total
```

Con la función anterior, se calcula la frecuencia y total de los libros 1 y 2, esto en el caso de tener stopwords y también cuando no las tienen

In [ ]:
```python
frecuencia_libro1, total_libro1 = wordCount(libro_1)
frecuencia_libro2, total_libro2 = wordCount(libro_2)
frecuencia_libro1_nostop, total_libro1_nostop = wordCount(libro_1_nostop)
frecuencia_libro2_nostop, total_libro2_nostop = wordCount(libro_2_nostop)
```

Finalmente, se usará la función de entropía generada previamente para hacer este cálculo en los libros con y sin stopwords.

In [ ]:
```python
entropia_libro1 = calcEntr(frecuencia_libro1, total_libro1)
entropia_libro2 = calcEntr(frecuencia_libro2, total_libro2)
entropia_libro1_nostop = calcEntr(frecuencia_libro1_nostop, total_libro1_nostop)
entropia_libro2_nostop = calcEntr(frecuencia_libro2_nostop, total_libro2_nostop)


print(f"La entropía a nivel de palabras del Libro 1 es {entropia_libro1}")
print(f"La entropía a nivel de palabras del Libro 2 es {entropia_libro2}")
print(f"La entropía a nivel de palabras del Libro 1 sin stopwords es {entropia_libro1_nostop}")
print(f"La entropía a nivel de palabras del Libro 2 sin stopwords es {entropia_libro2_nostop}")
```

```
La entropía a nivel de palabras del Libro 1 es 9.21154132057109
La entropía a nivel de palabras del Libro 2 es 9.61096251407057
La entropía a nivel de palabras del Libro 1 sin stopwords es 11.069996852967849
La entropía a nivel de palabras del Libro 2 sin stopwords es 11.653170296257365
```

Conclusión

Como se puede observar en el caso de la **entropía a nivel de carácter**, todos los textos tienen una entropía similar; la cual indica una alta variabilidad en el uso de los caracteres y que no tienen una secuencia predecible, no sigue patrones obvios.

Para el caso de **entropía a nivel de palabras**, se puede observar que los libros son similares en entropía en sus dos estados: con y sin stopwords. Un valor de entropía alto para ambos casos indica una gran variedad de palabras en ambos libros, las cuales no se repiten con frecuencia y no tienen una secuencia predecible. Al quitar las stopwords el valor aumenta ya que se quitan palabras que no tienen un peso significativo y tambié una gran cantidad de palabras repetidas, lo cual aumenta la variabilidad.