1D1. Práctica: Cálculo de Información Mutua

Alumno: **Luis Fernando Izquierdo Berdugo**

Materia: **Procesamiento de Información**

Fecha: **10 de Septiembre de 2024**

Instrucciones:

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

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

Encontrar las asociaciones de palabras más significativas, usando la medida de información mutua para cada conjunto de datos. Cada conjunto de datos se deberá procesar por separado.

Los archivos son:

- `archivo_emojis_Proceso.csv`
- `archivo_emojis_Elpais.csv`
- `archivo_emojis_Elfinanciero.csv`

Usar el contenido de la columna ‘title’. Se pueden leer con:

```
data = pd.read_csv("<ruta>/archivo_emojis_Elfinanciero.csv")

text = data['title'].to_list()

text = " ".join(text)
```

Crear un notebook que calcule la información mutua de un texto. Usar una ventana de 2 palabras, para calcular las probabilidades conjuntas $p(x, y)$.

Incluir las asociaciones más importantes, respecto a la medida de información mutua para cada conjunto de datos proporcionado. Es decir, se deben de calcular todas las asociaciones entre el vocabulario único. Mostrar las diez asociaciones más importantes, de acuerdo al conjunto de datos proporcionado, indicar su índice de información mutua.

Preprocesamiento de los textos

Primero se descargarán los datos para las stopwords

In [35]:
```
from nltk.corpus import stopwords
import nltk
import ssl

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!
```

A continuación se crean las funciones para eliminar las stopwords y la de preprocesamiento general. Esta última efectúa los siguientes procesos:

- Pasar el texto a minúsculas
- Llamar la función que elimina stopwords
- Eliminar los acentos
- Sustituir los cambios de línea y de página por un espacio
- Eliminar caracteres especiales

In [36]:
```
import unicodedata
import re
import pandas as pd

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 pre(text):
    text = text.lower()
    text = remove_stopwords(text)
    text = ''.join(c for c in unicodedata.normalize('NFD', text) if unicodedata
    text = re.sub(r"[\t\n]", " ", text)
    text = re.sub(r"[;:,.\-\"‘/”(\)\[\]¿?¡!\{\}~<>|«»——\’\r\`\´]", "", text)
    return text
```

Con esto, se leen los archivos `csv` proporcionados y se preprocesan, siendo:

- `f1` el archivo correspondiente a "El Financiero".
- `f2` el archivo correspondiente a "El País".
- `f3` el archivo correspondiente a "Proceso".

In [37]:
```
data = pd.read_csv("archivo_emojis_Elfinanciero.csv")
f1 = data['title'].to_list()
f1 = " ".join(f1)
f1 = pre(f1)

data = pd.read_csv("archivo_emojis_Elpais.csv")
f2 = data['title'].to_list()
f2 = " ".join(f2)
f2 = pre(f2)
```

```
data = pd.read_csv("archivo_emojis_Proceso.csv")
f3 = data['title'].to_list()
f3 = " ".join(f3)
f3 = pre(f3)
```

Se crea la función que servirá para calcular la `Información Mutua`. Esta ejecuta lo siguiente:

- División del texto en una lista de palabras.
- Con la librería nltk, se usa la función `ngrams` para obtener una lista de bigramas.
- Conteo de los bigramas obtenidos.
- Creación de un dataframe que incluya como columnas los bigramas y la cantidad de veces que aparecen en el texto.
- Suma del total de bigramas del texto
- Adición de una columna para la probabilidad $P(X,Y)$ que es el conteo de cada bigrama entre el total de bigramas.
- Suma del total de las palabras en el texto.
- Adición de una columna para la probabilidad $P(X)$ que es el conteo de la primera palabra en cada bigrama entre el total de palabras.
- Adición de una columna para la probabilidad $P(Y)$ que es el conteo de la segunda palabra en cada bigrama entre el total de palabras.
- Adición de una columna para el cálculo de la Información Mutua, siguiendo la fórmula:

$$log_2(\frac{P(X,Y)}{P(X)*P(Y)})$$

In [38]:
```
from nltk import ngrams
from collections import Counter
import numpy as np

def infoMutua(text):
    palabras = text.split()
    bigramas = list(ngrams(palabras, 2))
    conteoBigramas = Counter(bigramas)
    df = pd.DataFrame(conteoBigramas.items(), columns=['bigrama', 'conteo'])
    totalBigramas = sum(conteoBigramas.values())
    df['p_xy'] = df['conteo'] / totalBigramas

     # Calcular las probabilidades marginales de cada palabra
    conteoPalabras = Counter(palabras)
    palabrasTotal = sum(conteoPalabras.values())
    df['p_x'] = df['bigrama'].apply(lambda bigrama: conteoPalabras[bigrama[0]]
    df['p_y'] = df['bigrama'].apply(lambda bigrama: conteoPalabras[bigrama[1]]

    # Calcular la información mutua
    df['infoMutua'] = df.apply(lambda row: np.log2(row['p_xy'] / (row['p_x'] *

    return df
```

Se obtienen los 10 bigramas de "El Financiero" con `mayor frecuencia` así como su índice `Información Mutua`.

En los resultados se puede observar la presencia de varios nombres propios, tanto de personas como de lugares o programas televisivos. Debido a que El Financiero es un periódico mexicano que reporta acerca del día a día de la sociedad, es normal encontrar artículo que tienen de título nombres propios de personajes de la cultura como "Checo Perez", así como lugares que suelen ir relacionados como la "CDMX" y el "EdoMex".

In [39]:
```
df1 = infoMutua(f1)
df1.nlargest(10, 'conteo')
```

Out[39]:

	bigrama	conteo	p_xy	p_x	p_y	infoMutua
1425	(checo, perez)	340	0.001640	0.001925	0.001983	8.747547
113	(wall, street)	316	0.001524	0.001529	0.001529	9.348498
417	(cdmx, edomex)	273	0.001317	0.006387	0.002465	6.386286
17988	(xochitl, galvez)	253	0.001220	0.002007	0.001317	8.851192
415	(hoy, circula)	243	0.001172	0.001780	0.001177	9.127994
101	(metro, cdmx)	223	0.001076	0.001602	0.006387	6.716568
2976	(donde, cuando)	191	0.000921	0.001992	0.002402	7.588824
38833	(casa, famosos)	184	0.000888	0.001876	0.001090	8.761152
2974	(liga, mx)	179	0.000863	0.001042	0.000863	9.906508
2977	(cuando, ver)	179	0.000863	0.002402	0.001336	8.071467

Se obtienen los 10 bigramas de "El País" con `mayor frecuencia` así como su índice `Información Mutua`.

En este análisis se observa que este periódico está un poco más enfocado en las noticias de índole internacional (por lo que observamos "EE. UU.", "America Latina", "Javier Milei" y "Nueva York") y temas sociales (donde observamos "Inteligencia Artificial" y "Cambio Climático")

In [49]:
```
df2 = infoMutua(f2)
df2.nlargest(10, 'conteo')
```

Out[49]:

	bigrama	conteo	p_xy	p_x	p_y	infoMutua
144	(ee, uu)	488	0.001850	0.001850	0.001854	9.075285
931	(inteligencia, artificial)	470	0.001782	0.001888	0.001842	9.000678
97	(america, latina)	303	0.001149	0.001422	0.001221	9.370486
164	(millones, dolares)	169	0.000641	0.002112	0.000933	8.345807
155	(nueva, york)	135	0.000512	0.002024	0.000580	8.767707
1359	(anos, despues)	125	0.000474	0.004481	0.001209	6.450333
2603	(elon, musk)	112	0.000425	0.000440	0.000516	10.870887
0	(javier, milei)	105	0.000398	0.000519	0.001084	9.465318
11351	(cambio, climatico)	104	0.000394	0.001054	0.000459	9.671611
3691	(in, the)	101	0.000383	0.002131	0.004102	5.453276

Se obtienen los 10 bigramas de "Proceso" con `mayor frecuencia` así como su índice `Información Mutua`.

En estos resultados se observa la presencia de temas políticos mexicanos sobre todos los demás, teniendo la presencia de nombres de políticos mexicanos como "Lopez Obrador", "Xochitl Galvez", "Mario Delgado", "Peña Nieto", y también tópicos que han sido relevantes en el índole político como "Tribunal Electoral" y "Poder Judicial".

In [46]:
```
df3 = infoMutua(f3)
df3.nlargest(10, 'conteo')
```

Out[46]:

	bigrama	conteo	p_xy	p_x	p_y	infoMutua
653	(tribunal, electoral)	37	0.001080	0.001401	0.001809	8.735122
2257	(mil, millones)	32	0.000934	0.002626	0.002889	6.943617
787	(lopez, obrador)	25	0.000729	0.001401	0.000788	9.368833
2884	(xochitl, galvez)	25	0.000729	0.000817	0.000788	10.146441
748	(mario, delgado)	23	0.000671	0.000905	0.000759	9.933753
1010	(poder, judicial)	23	0.000671	0.001430	0.000817	9.166324
799	(millones, pesos)	22	0.000642	0.002889	0.001080	7.685449
4209	(inteligencia, artificial)	22	0.000642	0.000817	0.000700	10.131941
55	(plan, b)	21	0.000613	0.001021	0.000642	9.868430
1531	(pena, nieto)	20	0.000584	0.001313	0.000759	9.194462