



**INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**

“Practica 1 – Normalización de texto”

-Tinoco Videgaray Sergio Ernesto

Grupo: 5BV1

Materia: Tecnologías del Lenguaje Natural

INSTITUTO POLITÉCNICO NACIONAL



ESCOM®

29/09/23

- **Introducción.**

En esta practica se va a realizar la normalización del texto de 2 documentos de texto, uno redactado en idioma español y otro en inglés.

La normalización forma parte del preprocesamiento del texto en la creación de un modelo de lenguaje natural ya que tiene como objetivo simplificar el diccionario o bien, la bolsa de palabras con la que se esta construyendo dicho modelo de lenguaje natural y en términos matemáticos, ayuda a reducir la dimensionalidad de un vector de características que va a representar cada documento dentro de un modelo de lenguaje, y de esta forma evitar algún tipo de sesgo que pueda reducir la eficiencia del modelo.

- **Desarrollo.**

A continuación, se describen los paquetes o módulos de Python con los que se va a realizar la implementación de dicha normalización de texto.

Paquetes.

Para el desarrollo de esta práctica se va a trabajar con los siguientes paquetes:

- TextBlob para realizar la normalización del texto de forma fácil.
- Pandas para el análisis exploratorio de datos.
- Matplotlib para la visualización de graficas como histogramas.
- Nltk se va a utilizar únicamente para importar una lista de stopwords tanto en el idioma inglés como en español.

- **Documento 1.**

Para el documento 1 se va a trabajar sobre un texto redactado en el idioma español.

Análisis exploratorio de texto

Antes de realizar la normalización del texto se va a realizar un análisis exploratorio para evaluar las condiciones del documento 1 e identificar los tokens más comunes en el diccionario.

a) Número total de tokens en el texto.

El número total de tokens en el texto corresponde con un total de 327 tokens como se muestra en la figura 1.



```
Total de tokens: 327
```

Figura 1. Impresión en consola del total de tokens en el documento 1.

b) Número de tokens únicos en el texto.

El número total de tokens únicos en el texto corresponde con un total de 190 tokens (figura 2).



```
Total de tokens unicos: 190
```

Figura 2. Impresión en consola del total de tokens únicos en el documento 1.

Como se puede observar el número total de tokens es 1.7 veces el número de tokens únicos. Lo que puede interpretarse como casi el doble de tokens únicos.

c) Histograma de los tokens.

A continuación, se presenta el histograma generado a partir de una muestra aleatoria de tokens correspondiente con el documento 1 (figura 3).

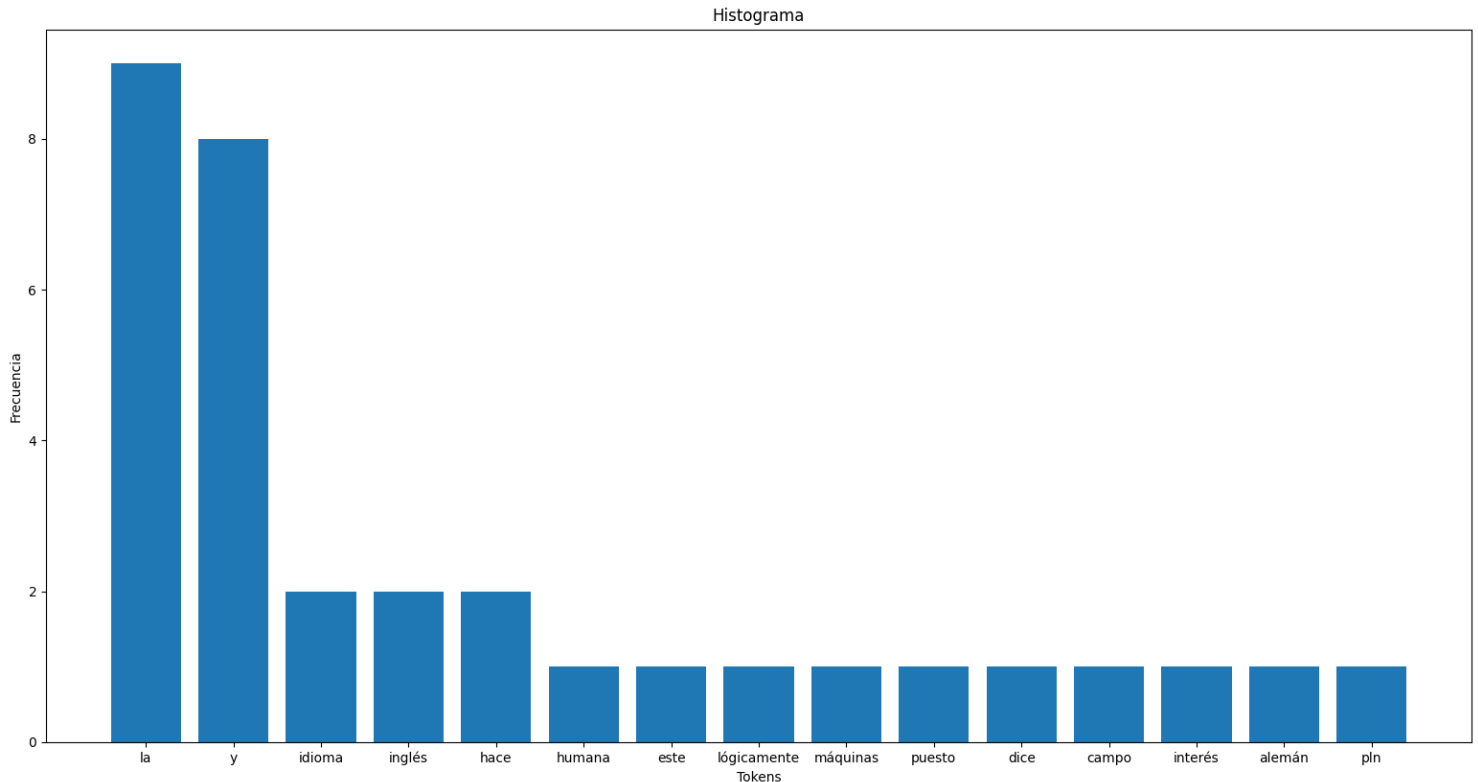


Figura 3. Histograma de los tokens del documento 1.

Como se puede observar en el histograma de tokens del documento 1, aparecen algunos conectores del idioma español como “este”, “la”, los cuales se definen como las “stopwords” o palabras “vacías” en el procesamiento del lenguaje natural, ya que no aportan significado alguno dentro de la semántica de un mensaje, aunque esto es dependiendo del contexto del mensaje. Pero en este caso las stopwords están generando ruido dentro del diccionario, lo que se puede traducir como un sesgo en términos matemáticos.

d) 10 tokens más comunes.

Se muestra una lista de los 10 tokens más frecuentes dentro del documento 1 (figura 4).


10 tokens mas frecuentes		
	tokens	freq
48	el	15
16	que	15
8	de	14
85	la	9
6	las	8
22	y	8
12	es	7
5	en	6
30	o	6
33	más	6

Figura 4. Impresión en consola de los 10 tokens más comunes en el documento 1.

Como se puede observar, aparecen otras stopwords como “el”, “que”, “de”, “la”, etc. Los cuales pueden producir un sesgo dentro de un modelo de procesamiento del lenguaje.

e) 10 tokens menos comunes.

Se muestra una lista de los tokens menos frecuentes dentro del documento 1 (figura 5).



10 tokens menos frecuentes		
	tokens	freq
70	embargo	1
73	resulta	1
82	cantidad	1
75	extremadamente	1
76	difícil	1
77	computadoras	1
78	tienen	1
79	lidiar	1
81	gran	1
179	7000	1

Figura 5. Impresión en consola de los 10 tokens menos comunes en el documento 1.

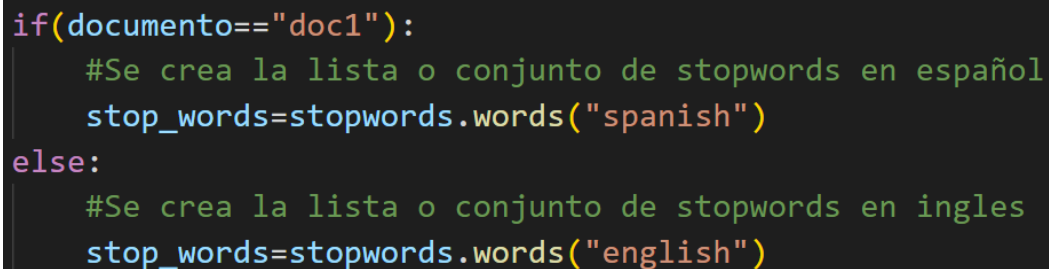
En contraste con los tokens más frecuentes, aquí no se muestra gran cantidad de stopwords que puedan generar un sesgo dentro del modelo.

Normalización de texto

En esta etapa se van a aplicar varios algoritmos al diccionario del documento 1 para reducir la cantidad de tokens y contrastar algunos resultados de lematización y stemming.

Primeramente, se van a remover las stopwords ya que representan un gran porcentaje dentro del diccionario y esto ayudara a optimizar los algoritmos que se aplicaran posteriormente.

Dependiendo del documento que se esté procesando se va a seleccionar un idioma para las stopwords (figura 6).



```
if(documento=="doc1"):
    #Se crea la lista o conjunto de stopwords en español
    stop_words=stopwords.words("spanish")
else:
    #Se crea la lista o conjunto de stopwords en ingles
    stop_words=stopwords.words("english")
```

Figura 6. Código del algoritmo que carga la lista de stopwords con base en el nombre del documento.

Por medio de un bucle for se va a recorrer la lista de tokens y se van a agregar a una segunda lista de tokens aquellas palabras que no se encuentren en la lista de stopwords. Es decir, se va a filtrar la lista de tokens por

medio de las stopwords. Posteriormente se va a comparar el tamaño del diccionario antes y después de remover las stopwords (figura 7 y 8).

```
print("Tamaño del diccionario antes de remover stop words:",len(all_tokens))
tokens = []
for t in all_tokens:
    if t not in stop_words:
        tokens.append(t)
print("Tamaño del diccionario despues de remover stop words:",len(tokens))
```

Figura 7. Código del algoritmo que elimina las stopwords del diccionario.

```
Tamaño del diccionario antes de remover stop words: 327
Tamaño del diccionario despues de remover stop words: 176
```

Figura 8. Impresión en consola de los tamaños del diccionario antes y después de remover stopwords.

Como se pudo observar, se redujo el tamaño del diccionario de 327 tokens a 176. Lo cual representa cerca del 54% del diccionario.

Posteriormente se van a remover los apostrofes ya que al igual que las stopwords, carecen de valor semántico al menos dentro del contexto de esta práctica (figura 9 y 10).

```
def remover_apostrofes(tokens):
    tokens_sin_apostorfes=[]
    apostrofes=['\'','\'','\'','\'']

    for t in tokens:
        if t not in apostrofes:
            tokens_sin_apostorfes.append(t)
    return tokens_sin_apostorfes
```

Figura 9. Código del algoritmo que elimina los apostrofes del diccionario.

```
Tamaño del diccionario antes de remover apostrofes: 176
Tamaño del diccionario despues de remover apostrofes: 170
```

Figura 10. Impresión en consola del tamaño del diccionario antes y después de remover apostrofes.

En este caso se eliminaron 6 apostrofes del diccionario.

De igual forma se van a remover los acentos del diccionario, remplazándolos por sus vocales correspondientes sin la tilde (figura 11).

```
def remover_acentos(tokens):
    tokens_sin_acentos=[]
    diccionario_conversion=str.maketrans({'á':'a','é':'e','í':'i','ó':'o'})

    for t in tokens:
        tokens_sin_acentos.append(str(t).translate(diccionario_conversion))
    return tokens_sin_acentos
```

Figura 11. Código del algoritmo que elimina los acentos del diccionario.

Posteriormente se realiza el etiquetado o Tagging (figura 12 y 13).

```
#POS tagging Etiquetado
tokens_tagged=tokens[:]
tokens_tagged=tokens_tagged.sentences[0].tags
print("Etiquetas asignadas por el POS tagging\n",tokens_tagged)
```

Figura 12. Código del algoritmo que aplica el POS tagging al diccionario.

```
Etiquetas asignadas por el POS tagging
[('Por', 'NNP'), ('general', 'JJ'), ('pensamos', 'NN'), ('complejidad', 'NP'), ('comportamiento', 'NN'), ('intuitivo', 'NN'), ('utiliza', 'JJ'), ('NN'), ('señales', 'NNS'), ('semanticas', 'VBP'), ('palabras', 'JJ'), ('NN'), ('facil', 'NN'), ('aprender', 'NN'), ('idioma', 'NN'), ('nuevo', 'miento', 'NN'), ('repetible', 'JJ'), ('entrenado', 'NN'), ('casi', 'NN'), ('sigue', 'NN'), ('conjunto', 'NN'), ('reglas', 'NN'), ('estricto', 'NN'), ('vos', 'VBP'), ('terminan', 'JJ'), ('femeninos', 'NNS'), ('caso', 'VBP'), ('anos', 'NN'), ('resulta', 'JJ'), ('natural', 'JJ'), ('extremadamente', 'VBP'), ('gran', 'JJ'), ('cantidad', 'NN'), ('datos', 'NN'), ('estructu', 'les', 'NNS'), ('falta', 'VBP'), ('contexto', 'JJ'), ('intencion', 'NN')]
```

Figura 13. Impresión en consola de las etiquetas asignadas en el POS tagging.

En este caso algunas etiquetas no coinciden con su significado gramatical como el caso de “fácil” que fue etiquetado como “Noun” o sustantivo cuando debería ser etiquetado como JJ o adjetivo. Esto quiere decir que algunas palabras no están siendo adaptadas a su significado gramatical de acuerdo al diccionario que maneja el paquete de TextBlob.

Finalmente se aplica la lematización y el stemming de los tokens del diccionario y se comparan en una tabla (figura 14 y 15).

```
#Lematizar
tokens_lematizados=tokens.lemmatize()

#Stemming
tokens_stemming=tokens.stem()
```

Figura 14. Código del algoritmo que aplica la lematización y el stemming al diccionario.

utiliza	utiliza	utiliza
transmitir	transmitir	transmitir
informacion	informacion	informacion
significados	significados	significado
señales	señales	señal
semanticas	semanticas	semantica
palabras	palabras	palabra
signos	signos	signo
imagenes	imagenes	imagen
Se	Se	se
dice	dice	dice
facil	facil	facil

Figura 15. Impresión en consola de los tokens en su forma original, lema y stemming.

Como se pudo observar en la lematización y el stemming, algunas palabras o tokens, conservan su estructura mientras que a otras se les omiten los sufijos como el caso de “significados/significado”, “señales/señal”, “imágenes/imagen”, etc. Esto se debe a que la lematización se realiza con base en un diccionario mientras que el stemming únicamente omite los sufijos y no está necesariamente asociado a un diccionario.

Análisis exploratorio posterior a normalización

Una vez realizada la normalización del diccionario se va a realizar nuevamente un análisis exploratorio para determinar que tan eficiente es el nuevo diccionario con respecto del original.

a) Número total de tokens en el texto.

El número total de tokens en el texto corresponde con un total de 170 tokens (figura 16).

Total de tokens: 170

Figura 16. Impresión en consola en consola del total de tokens del documento 1 después del proceso de normalización.

b) Número de tokens únicos en el texto corresponde con 149 tokens únicos (figura 17).

Total de tokens unicos: 149

Figura 17. Impresión en consola en consola del total de tokens únicos del documento 1 después de la normalización.

c) Histograma de los tokens.

Se vuelve a generar un histograma obtenido a partir de una muestra aleatoria de tokens (figura 18).

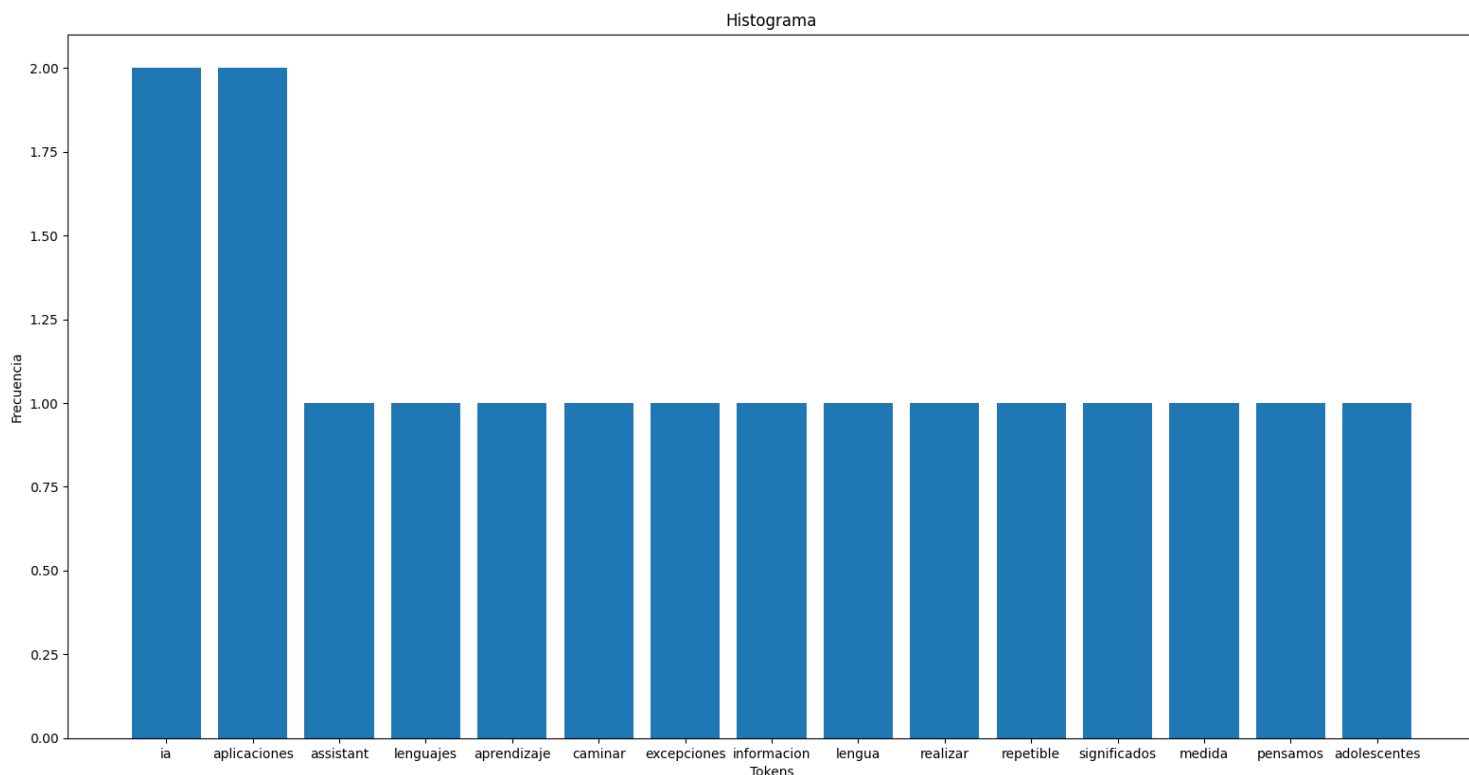


Figura 18. Histograma de tokens del documento 1 después de la normalización.

Como se pudo observar no aparece ninguna stopword, apostrofe, o palabra acentuada dentro de la muestra y las palabras con mayor frecuencia no sufrieron ningún cambio en su estructura.

d) 10 tokens más comunes.

Se despliega una lista de los tokens más frecuentes después de la normalización (figura 19).

10 tokens mas frecuentes		
	tokens	freq
98	lenguas	5
46	natural	3
44	humanos	2
65	ia	2
25	trata	2
119	mundo	2
97	uso	2
22	idioma	2
127	google	2
33	reglas	2

Figura 19. Impresión en consola de los 10 tokens más comunes del documento 1 después de realizar la normalización.

Como se puede observar en la tabla, no aparece ninguna stopword, apostrofe, o palabra acentuada que pueda causar algún sesgo dentro del modelo y cada token se encuentra en su forma original.

e) 10 tokens menos comunes (figura 20).


```

10 tokens menos frecuentes
      tokens  freq
51      gran    1
50     lidiar    1
49   computadoras  1
48     dificil    1
47 extremadamente  1
45     resulta    1
43     embargo    1
42        sin    1
41   problema    1
143      7000    1

```

Figura 20. Impresión en consola de los 10 tokens menos comunes del documento 1 después de realizar la normalización.

De igual forma no aparece ninguna stopword en los tokens menos frecuentes.

- Documento 2.

Para el documento 2 se va a trabajar sobre un texto redactado en el idioma inglés.

Análisis exploratorio de texto

Antes de realizar la normalización del texto se va a realizar un análisis exploratorio para evaluar las condiciones del documento 1 e identificar los tokens más comunes en el diccionario.

- f) Número total de tokens en el texto.

El número total de tokens en el texto corresponde con un total de 352 tokens (figura 21).

```
Total de tokens: 352
```

Figura 21. Impresión en consola del total de tokens en el documento 2.

- g) Número de tokens únicos en el texto.

El número total de tokens únicos en el texto corresponde con un total de 207 tokens (figura 22).

```
Total de tokens unicos: 207
```

Figura 22. Impresión en consola del total de tokens únicos en el documento 2.

Como se puede observar el número total de tokens es 1.7 veces el número de tokens únicos. Lo que puede interpretarse como casi el doble de tokens únicos.

h) Histograma de los tokens.

A continuación, se presenta el histograma generado a partir de una muestra aleatoria de tokens correspondiente con el documento 2 (figura 23).

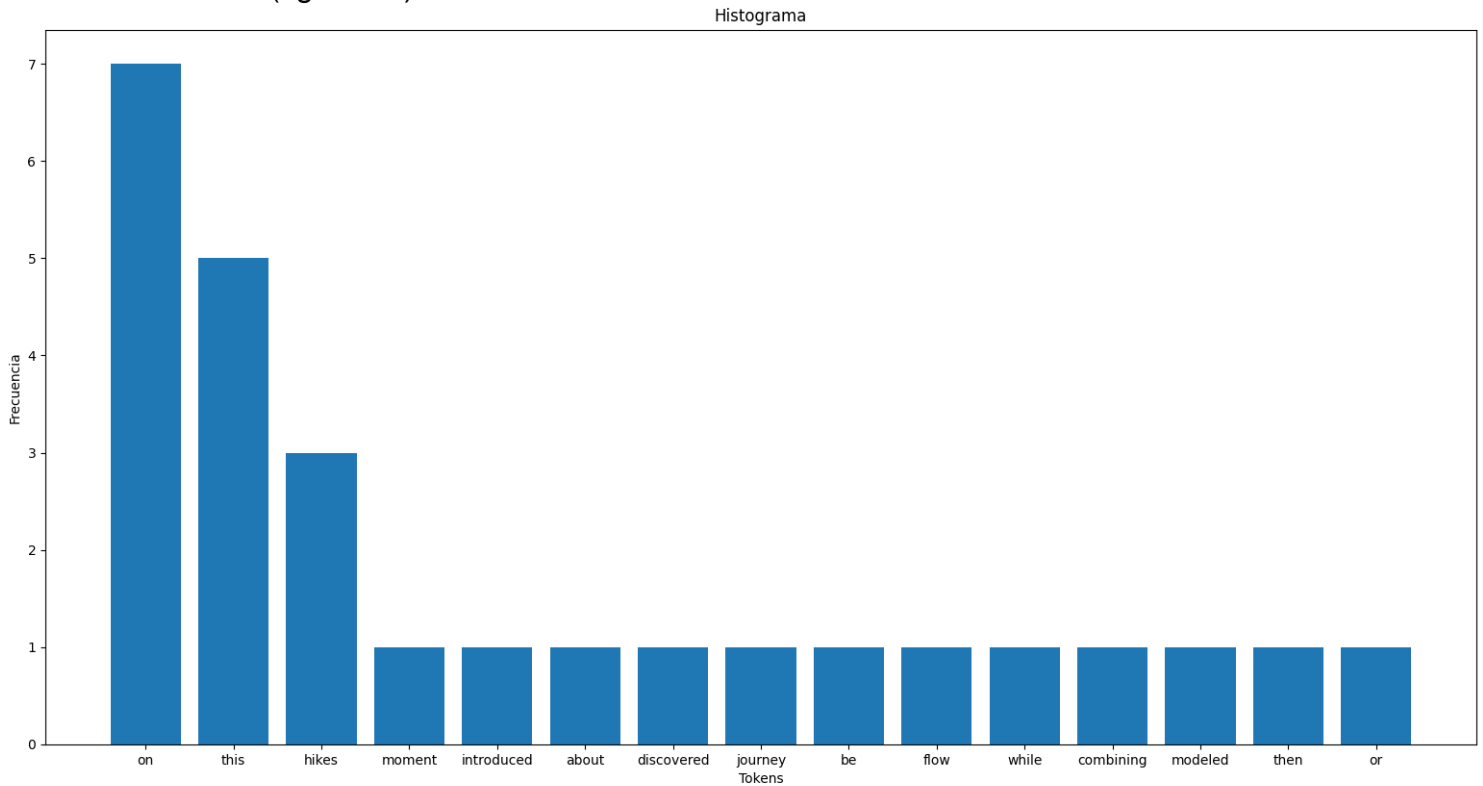


Figura 23. Histograma de los tokens del documento 2.

Como se puede observar en el histograma de tokens del documento 2, aparecen algunos conectores del idioma español como “on”, “this”, los cuales se definen como las “stopwords” o palabras “vacías” en el procesamiento del lenguaje natural, ya que no aportan significado alguno dentro de la semántica de un mensaje, aunque esto es dependiendo del contexto del mensaje. Pero en este caso las stopwords están generando ruido dentro del diccionario, lo que se puede traducir como un sesgo en términos matemáticos.

i) 10 tokens más comunes.

Se muestra una lista de los 10 tokens más frecuentes dentro del documento 2 (figura 24).

10 tokens mas frecuentes		
	tokens	freq
23	the	19
26	of	16
39	and	14
10	i	11
30	a	10
43	to	8
7	in	8
63	on	7
32	that	7
4	language	6

Figura 24. Impresión en consola de los 10 tokens más comunes en el documento 2.

Como se puede observar, aparecen otras stopwords como “the”, “of”, “and”, “in”, etc. Los cuales pueden producir un sesgo dentro de un modelo de procesamiento del lenguaje.

j) 10 tokens menos comunes.

Se muestra una lista de los tokens menos frecuentes dentro del documento 2 (figura 25).

10 tokens menos frecuentes		
	tokens	freq
82	origin	1
91	landscapes	1
83	elements	1
84	at	1
85	moment	1
86	they	1
87	occurred	1
88	while	1
89	walking	1
198	wonder	1

Figura 25. Impresión en consola de los 10 tokens menos comunes en el documento 2.

En contraste con los tokens más frecuentes, aquí no se muestran tantas stopwords que puedan generar un sesgo dentro del modelo.

Normalización de texto

Al igual que el documento 1, en esta etapa se van a aplicar varios algoritmos al diccionario del documento 2 para reducir la cantidad de tokens y contrastar algunos resultados de lematización y stemming.

naturalist	naturalist	naturalist
professional	professional	profession
life	life	life
book	book	book
modeled	modeled	model
metaphysic	metaphysic	metaphys
I	I	i
know	know	know
best-the	best-the	best-th
flow	flow	flow
ideas	idea	idea
observations	observation	observ
arise	arise	aris
spontaneously	spontaneously	spontan
humans	human	human

Figura 29. Impresión en consola de los tokens en su forma original, lema y stemming.

En este caso se puede observar que en la lematización y el stemming algunas palabras o tokens, conservan su estructura mientras que a otras se les omiten los sufijos como el caso de “ideas/idea”, “spontaneously/spontan”, “professional/profession”, etc. Esto se debe a que la lematización se realiza con base en un diccionario mientras que el stemming únicamente omite los sufijos y no está necesariamente asociado a un diccionario.

Análisis exploratorio posterior a normalización

Una vez realizada la normalización del diccionario del documento 2 se va a realizar nuevamente un análisis exploratorio para determinar qué tan eficiente es el nuevo diccionario con respecto del original.

a) Número total de tokens en el texto.

El número total de tokens en el texto corresponde con un total de 192 tokens (figura 30).

Total de tokens: 192

Figura 30. Impresión en consola en consola del total de tokens del documento 1 después del proceso de normalización.

b) Número de tokens únicos en el texto.

El número de tokens únicos corresponde con 154 tokens únicos (figura 31).

Total de tokens unicos: 154

Figura 31. Impresión en consola en consola del total de tokens únicos del documento 1 después de la normalización.

c) Histograma de los tokens.

Se vuelve a generar un histograma obtenido a partir de una muestra aleatoria de tokens (figura 32).

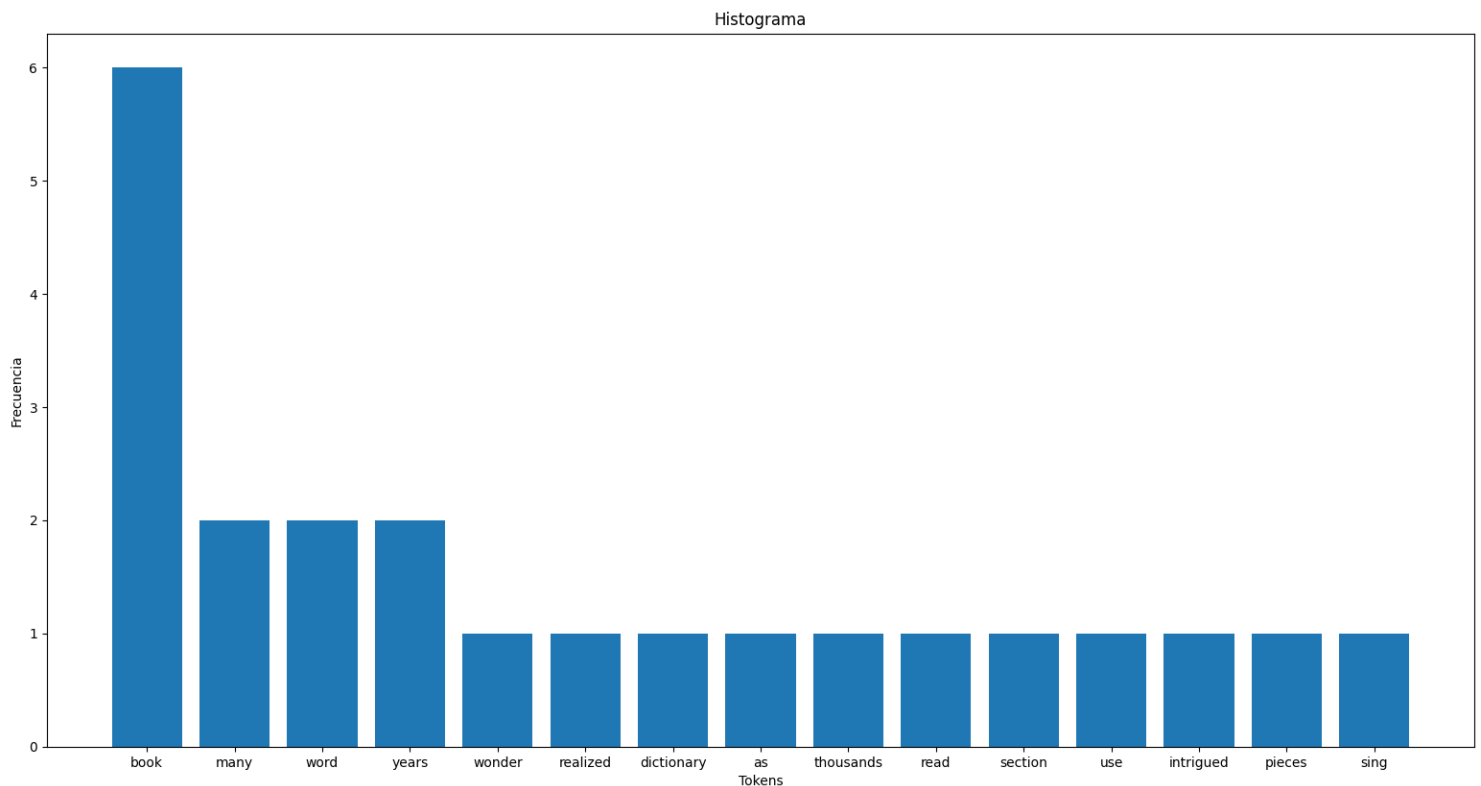


Figura 32. Histograma de tokens del documento 2 después de la normalización.

En este caso ya no aparece ninguna stopword, apostrofe, o palabra acentuada dentro de la muestra y las palabras con mayor frecuencia no sufrieron ningún cambio en su estructura.

d) 10 tokens más comunes.

Se despliega una lista de los tokens más frecuentes después de la normalización (figura 33).

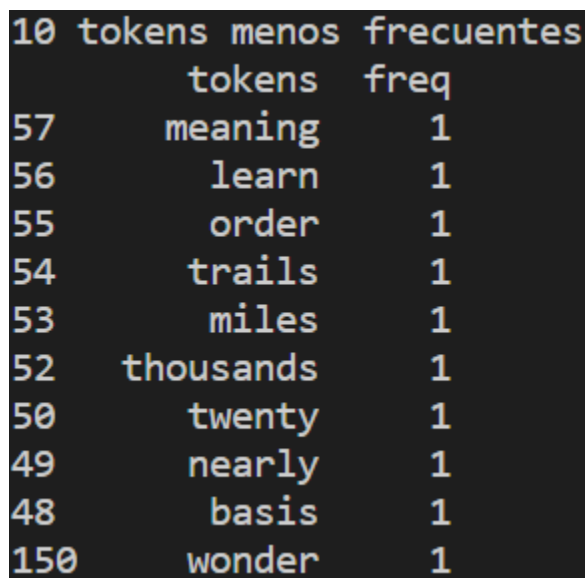
```
10 tokens mas frecuentes
tokens  freq
6       i      11
11      book   6
3       language 6
123     ideas  4
44      hikes  3
90      english 3
41      word   2
63      wild   2
29      land   2
30      this   2
```

Figura 33. Impresión en consola de los 10 tokens más comunes del documento 2 después de realizar la normalización.

Se puede observar que no aparece ninguna stopword, apostrofe, o palabra acentuada dentro que pueda causar algún sesgo dentro del modelo y cada token se encuentra en su forma original.

e) 10 tokens menos comunes.

Se muestran los 10 tokens menos comunes en el nuevo diccionario (figura 34).



10 tokens menos frecuentes		
	tokens	freq
57	meaning	1
56	learn	1
55	order	1
54	trails	1
53	miles	1
52	thousands	1
50	twenty	1
49	nearly	1
48	basis	1
150	wonder	1

Figura 34. Impresión en consola de los 10 tokens menos comunes del documento 2 después de realizar la normalización.

De igual forma no aparece ninguna stopword, apostrofe, o palabra acentuada en los tokens menos frecuentes.

Análisis y conclusiones

En esta práctica se realizó el proceso que corresponde con el preprocesamiento de un modelo de lenguaje natural, más específicamente con la etapa de normalización de texto, en este caso de un diccionario que surge a partir de un texto que puede estar en español o en inglés.

Como se pudo observar en los análisis antes y después de normalizar, hubo una reducción de stopwords que representaba cerca del 54% en el caso del documento 1 y 56% del diccionario para el caso del documento 2, teniendo así un promedio del 55% para todo el corpus de esta práctica. Lo que implica un diccionario más reducido y en términos matemáticos, un vector de características con una dimensionalidad menor a la que se tenía en un inicio. Lo cual resulta más eficiente y preciso a la hora de querer trabajar sobre este corpus en un modelo de lenguaje natural.

Ya en la práctica, se debería realizar un análisis más profundo dependiendo del objetivo que se tenga a la hora de realizar un procesamiento de lenguaje, pero independientemente del objetivo o el contexto que se tenga acerca del texto con el que se este trabajando, algo que no puede faltar es la parte de la normalización de texto ya que ayuda a reducir y simplificar el tamaño del diccionario con el que se está trabajando de forma considerable e incluso puede ayudar a contextualizar mejor el texto en si como en la etapa del etiquetado o la lematización.