

Laboratorio 2: Truncado y Lematización

Huerta Aguilar, Jesus., Huitzil Juárez, Guadalupe Quetzalli., Pérez Sánchez, Jasmine.,

Ruiz Ramírez, Gabino

Benemérita Universidad Autónoma de Puebla

Recuperación de la Información

Fecha: 23 enero de 2024

Resumen: En esta práctica se utilizó el lenguaje de programación “Python” para crear un código que, a partir de un texto dado, se realizará la separación del texto en tokens, eliminación de signos de puntuación y palabras vacías, se convertirán en minúsculas y se truncaran las palabras, para ello nos apoyamos del código aprendido en la práctica de laboratorio1 (split, string punctuation, re, stopwords) junto con los 3 métodos distintos que permiten truncar las palabras en sintaxis de código son snowball, wordnet, y porter, recordando que estas sintaxis vienen de la librería NLTK.

Palabras clave: tokens, palabras vacías, re, nltk, Python, snowball, wordnet, porter, truncar.

I. INTRODUCCIÓN

Recordemos que la recuperación de la información se refiere al proceso de obtener información relevante y útil a partir de un conjunto más grande de datos. Los algoritmos y técnicas utilizados en la recuperación de la información buscan devolver resultados relevantes y útiles de manera eficiente. En esta práctica continuaremos aprendiendo el proceso para recuperar la información de forma correcta, en este caso aprenderemos a truncar las palabras dado un texto, para ello analizaremos la forma en que muestran los resultados los distintos métodos (porter segmentación y lematización) para realizar este proceso.

II. OBJETIVO Y PLANTEAMIENTO DEL PROBLEMA

Aprender a preparar los textos para que sean de utilidad en el proceso de recuperación de información. Para ello deberás separar el texto en tokens, eliminarse los tokens inútiles (signos de puntuación, números), palabras vacías, convertir a minúsculas y truncar las palabras.

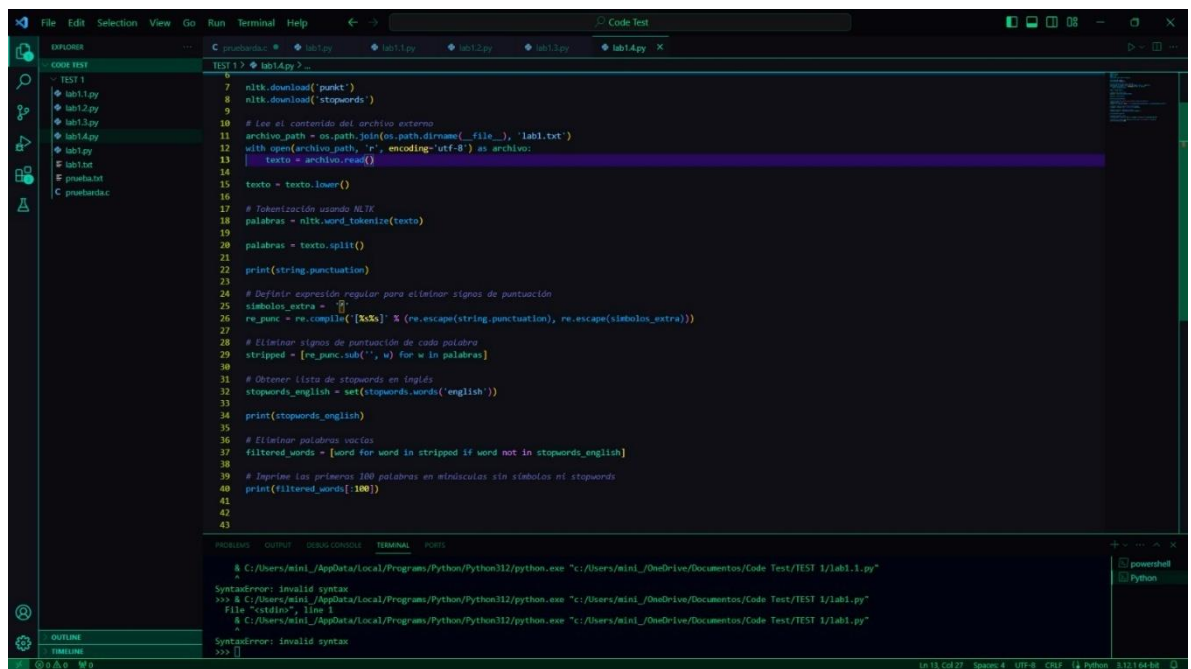
III. DESARROLLO EXPERIMENTAL

Para realizar esta práctica necesitaremos:

- Software Python
- Librería NLTK
- Libro de la página: <https://www.gutenberg.org/files/2591/2591-0.txt>

Una vez instalado el programa, la librería y descargado el libro procedemos a iniciar con la práctica:

1. Del texto conseguido, realizaremos el mismo proceso que en la práctica de laboratorio 1 es decir, separaremos el texto en tokens, eliminaremos signos de puntuación, colocaremos en minúsculas y eliminaremos las palabras vacías de las primeras 100 palabras:



```
1 # Importar librerías
2 import nltk
3
4 # Descargar los recursos necesarios
5 nltk.download('punkt')
6 nltk.download('stopwords')
7
8 # Leer el contenido del archivo externo
9 archivo_path = os.path.join(os.path.dirname(__file__), 'lab1.txt')
10 with open(archivo_path, 'r', encoding='utf-8') as archivo:
11     texto = archivo.read()
12
13 texto = texto.lower()
14
15 # Tokenización usando NLTK
16 palabras = nltk.word_tokenize(texto)
17
18 palabras = texto.split()
19
20 # Definir expresión regular para eliminar signos de puntuación
21 simbolos_extra = ''
22 re_punc = re.compile('[%s]' % (re.escape(string.punctuation), re.escape(simbolos_extra)))
23
24 # Eliminar signos de puntuación de cada palabra
25 stripped = [re_punc.sub('', w) for w in palabras]
26
27 # Obtener lista de stopwords en inglés
28 stopwords_english = set(stopwords.words('english'))
29
30 print(stopwords_english)
31
32 # Eliminar palabras vacías
33 filtered_words = [word for word in stripped if word not in stopwords_english]
34
35 # Imprimir las primeras 100 palabras en minúsculas sin símbolos ni stopwords
36 print(filtered_words[:100])
```

2. Del resultado obtenido en el paso anterior, procederemos a trunquear las palabras utilizando Porter Stemming, para ello lo llamaremos de la librería de NLTK, con ayuda de la siguiente sintaxis “from nltk.stem.porter import PorterStemmer”.
 - a. Primero crearemos un objeto tipo PorterStemmer, en seguida trunquemos palabra por palabra con ayuda del objeto creado y de la lista llamada “filtered_words”, así mismo el resultado lo iremos guardando en otra variable llamada “stemmed_words”.

```

65 # Imprime las primeras 100 palabras en minúsculas sin símbolos ni stopwords
66 print("\n///// TEXTO SIN PALABRAS VACIAS (BASE PARA TRUNCADAS, SNOWBALL STEMMER Y LEMMATIZER)\n")
67 print(filtered_words[:100])
68
69 stemmer = PorterStemmer()
70
71 stemmed_words = [stemmer.stem(word) for word in filtered_words]
72
73 # Imprime las primeras 100 palabras truncadas
74 print("\n///// TEXTO CON PALABRAS TRUNCADAS\n")
75 print(stemmed_words[:100])
76

```

3. Realizaremos una investigación del funcionamiento de *nlk.stem.SnowballStemmer()* y de *nlk.wordnet.WordNetLemmatizer()*
4. Una vez realizada la investigación ejecutaremos estas herramientas en el mismo código para observar la diferencia de estas:
 - a. Primero ejecutaremos *SnowballStemmer()* de la misma forma que el paso 2, es decir, crearemos nuestra variable objeto llamada *stemmer2* la cual llamara a la función *SnowballStemmer* donde añadiremos el idioma en el que estamos trabajando en este caso inglés; una vez creado el objeto podremos truncar palabra por palabra de la lista “*filtered_words*”, y el resultado lo iremos guardando en una nueva lista llamada “*stemmed_words2*”:

```

77 stemmer2 = SnowballStemmer('english')
78
79 stemmed_words2 = [stemmer2.stem(word) for word in filtered_words]
80
81 # Imprime las primeras 100 palabras SNOWBALL STEMMER
82 print("\n///// TEXTO SNOWBALL STEMMER\n")
83 print(stemmed_words2[:100])

```

- b. Ahora ejecutaremos *WordNetLemmatize()*, como en los pasos anteriores, pero ahora llamando la función desde una variable llamada *lemmatizer*, y el resultado guardándolo en una variable llamada *lemmatized_words*:

```

85 lemmatizer = WordNetLemmatizer()
86
87 lemmatized_words = [lemmatizer.lemmatize(word) for word in filtered_words]
88
89 # Imprime las primeras 100 palabras LEMMATIZER
90 print("\n///// TEXTO LEMMATIZER\n")
91 print(lemmatized_words[:100])

```

IV. DISCUSIÓN Y RESULTADOS

De acuerdo con los pasos de nuestra práctica, nuestros resultados fueron los siguiente:

1. En la salida de consola se muestran todas las palabras separadas, con los signos eliminados, sin palabras vacías y en minúsculas, tomando en cuenta las primeras 100 palabras, podemos observar que es el mismo resultado que obtuvimos en el laboratorio anterior.

```
Símbolo del sistema X + -

///// TEXTO SIN ESPACIOS

['The', 'Project', 'Gutenberg', 'eBook', 'of', 'Grims', 'Fairy', 'Tales', 'by', 'Jacob', 'Grimm', 'and', 'Wilhelm', 'Grimm', 'This', 'eBook', 'is', 'for', 'the', 'use', 'of', 'anyone', 'anywhere', 'in', 'the', 'United', 'States', 'and', 'most', 'other', 'parts', 'of', 'the', 'world', 'at', 'no', 'cost', 'and', 'with', 'almost', 'no', 'restrictions', 'whatsoever', 'You', 'may', 'copy', 'it', 'give', 'it', 'away', 'or', 'reuse', 'it', 'under', 'the', 'terms', 'of', 'the', 'Project', 'Gutenberg', 'License', 'included', 'with', 'this', 'eBook', 'or', 'online', 'at', 'www.gutenberg.org', 'If', 'you', 'are', 'not', 'located', 'in', 'the', 'United', 'States', 'you', 'will', 'have', 'to', 'check', 'the', 'laws', 'of', 'the', 'country', 'where', 'you', 'are', 'located', 'before', 'using', 'this', 'eBook', 'Title', 'Grims', 'Fairy', 'Tales']

///// SIGNOS DE PUNTUACION

!#$%&'()*+,-./:;<=>?@[\]^_`{|}~

///// TEXTO SIN SIGNOS DE PUNTUACION

['The', 'Project', 'Gutenberg', 'eBook', 'of', 'Grims', 'Fairy', 'Tales', 'by', 'Jacob', 'Grimm', 'and', 'Wilhelm', 'Grimm', 'This', 'eBook', 'is', 'for', 'the', 'use', 'of', 'anyone', 'anywhere', 'in', 'the', 'United', 'States', 'and', 'most', 'other', 'parts', 'of', 'the', 'world', 'at', 'no', 'cost', 'and', 'with', 'almost', 'no', 'restrictions', 'whatsoever', 'You', 'may', 'copy', 'it', 'give', 'it', 'away', 'or', 'reuse', 'it', 'under', 'the', 'terms', 'of', 'the', 'Project', 'Gutenberg', 'License', 'included', 'with', 'this', 'eBook', 'or', 'online', 'at', 'www.gutenberg.org', 'If', 'you', 'are', 'not', 'located', 'in', 'the', 'United', 'States', 'you', 'will', 'have', 'to', 'check', 'the', 'laws', 'of', 'the', 'country', 'where', 'you', 'are', 'located', 'before', 'using', 'this', 'eBook', 'Title', 'Grims', 'Fairy', 'Tales']

///// TEXTO EN MINUSCULAS

['the', 'project', 'gutenberg', 'ebook', 'of', 'grims', 'fairy', 'tales', 'by', 'jacob', 'grimm', 'and', 'wilhelm', 'grimm', 'this', 'ebook', 'is', 'for', 'the', 'use', 'of', 'anyone', 'anywhere', 'in', 'the', 'united', 'states', 'and', 'most', 'other', 'parts', 'of', 'the', 'world', 'at', 'no', 'cost', 'and', 'with', 'almost', 'no', 'restrictions', 'whatsoever', 'you', 'may', 'copy', 'it', 'give', 'it', 'away', 'or', 'reuse', 'it', 'under', 'the', 'terms', 'of', 'the', 'project', 'gutenberg', 'license', 'included', 'with', 'this', 'ebook', 'or', 'online', 'at', 'www.gutenberg.org', 'if', 'you', 'are', 'not', 'located', 'in', 'the', 'united', 'states', 'you', 'will', 'have', 'to', 'check', 'the', 'laws', 'of', 'the', 'country', 'where', 'you', 'are', 'located', 'before', 'using', 'this', 'ebook', 'title', 'grims', 'fairy', 'tales']

///// PRIMERAS 5 PALABRAS VACÍAS

['now', 'below', 'y', 'them', 'she's']

///// TEXTO SIN PALABRAS VACIAS

['project', 'gutenberg', 'ebook', 'grims', 'fairy', 'tales', 'jacob', 'grimm', 'wilhelm', 'grimm', 'ebook', 'use', 'anyone', 'anywhere', 'united', 'states', 'parts', 'world', 'cost', 'almost', 'restrictions', 'whatsoever', 'may', 'copy', 'give', 'away', 'reuse', 'terms', 'project', 'gutenberg', 'license', 'included', 'ebook', 'online', 'wwwgutenbergorg', 'located', 'united', 'states', 'check', 'laws', 'country', 'located', 'using', 'ebook', 'title', 'grims', 'fairy', 'tales', 'author', 'jacob', 'grimm', 'wilhelm', 'grimm', 'translators', 'edgar', 'taylor', 'marian', 'edwards', 'release', 'date', 'april', '2001', 'ebook', '2591', 'recently', 'updated', 'june', '28', '2021', 'language', 'english', 'character', 'set', 'encoding', 'utf8', 'produced', 'emma', 'dudding', 'john', 'bickers', 'dagny', 'david', 'widger', 'start', 'project', 'gutenberg', 'ebook', 'grims', 'fairy', 'tales', 'grims', 'fairy', 'tales', 'jacob', 'grimm', 'wilhelm', 'grimm', 'preparers']

C:\Users\Jesús Huerta Aguilar>
```

2. El resultado del segundo paso obtuvimos lo siguiente:

- a. Podemos comparar el resultado de no tener las palabras truncadas con las que, si están, por ejemplo, observamos que elimina los sufijos y prefijos, siguiendo ciertas reglas como las palabras en terminación “sses” tendrán la terminación solo “ss” quitando la es, mientras que las que terminen en “ies” se convertirá en “i”, y así sucesivamente; por ejemplo en el texto del código: la palabra grims se convirtió en solo grimm; la palabra fairy ahora es fairi; la palabra united ahora es unit, entre otras.

```
Command Prompt X + -

['on', 'you've', 'which', 'in', 'other']

///// TEXTO SIN PALABRAS VACIAS (BASE PARA TRUNCADAS, SNOWBALL STEMMER Y LEMMATIZER)

['project', 'gutenberg', 'ebook', 'grims', 'fairy', 'tales', 'jacob', 'grimm', 'wilhelm', 'grimm', 'ebook', 'use', 'anyone', 'anywhere', 'united', 'states', 'parts', 'world', 'cost', 'almost', 'restrictions', 'whatsoever', 'may', 'copy', 'give', 'away', 'reuse', 'terms', 'project', 'gutenberg', 'license', 'included', 'ebook', 'online', 'wwwgutenbergorg', 'located', 'united', 'states', 'check', 'laws', 'country', 'located', 'using', 'ebook', 'title', 'grims', 'fairy', 'tales', 'author', 'jacob', 'grimm', 'wilhelm', 'grimm', 'translators', 'edgar', 'taylor', 'marian', 'edwards', 'release', 'date', 'april', '2001', 'ebook', '2591', 'recently', 'updated', 'june', '28', '2021', 'language', 'english', 'character', 'set', 'encoding', 'utf8', 'produced', 'emma', 'dudding', 'john', 'bickers', 'dagny', 'david', 'widger', 'start', 'project', 'gutenberg', 'ebook', 'grims', 'fairy', 'tales', 'grims', 'fairy', 'tales', 'jacob', 'grimm', 'wilhelm', 'grimm', 'preparers']

///// TEXTO CON PALABRAS TRUNCADAS

['project', 'gutenberg', 'ebook', 'grimm', 'fairi', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'ebook', 'use', 'anyon', 'anywher', 'unit', 'state', 'part', 'world', 'cost', 'almost', 'restrict', 'whatsoev', 'may', 'copi', 'give', 'away', 'reus', 'term', 'project', 'gutenberg', 'licens', 'includ', 'ebook', 'onlin', 'wwwgutenbergorg', 'locat', 'unit', 'state', 'check', 'law', 'countri', 'locat', 'use', 'ebook', 'titl', 'grimm', 'fairi', 'tale', 'author', 'jacob', 'grimm', 'wilhelm', 'grimm', 'translat', 'edgar', 'taylor', 'marian', 'edward', 'releas', 'date', 'april', '2001', 'ebook', '2591', 'recent', 'updat', 'jun', '28', '2021', 'languag', 'english', 'character', 'set', 'encod', 'utf8', 'produc', 'emma', 'dud', 'john', 'bickers', 'dagni', 'david', 'widger', 'start', 'project', 'gutenberg', 'ebook', 'grimm', 'fairi', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'prepar']
```


3. En la investigación aprendimos lo siguiente:

- a. El módulo nltk.stem.snowball en NLTK (Natural Language Toolkit) proporciona la implementación del algoritmo Snowball stemming. El stemming es el proceso de reducir una palabra a su raíz, eliminando afijos. El Snowball stemming es un algoritmo de stemming que ha sido desarrollado por Martin Porter.

- b. La clase WordNetLemmatizer en NLTK se utiliza para realizar lematización utilizando WordNet, una base de datos léxica del inglés. La lematización es el proceso de reducir las palabras a sus formas base o lemas.

4. Para el último paso conseguimos lo siguiente:

- a. El resultado del uso del método SnowballStemmer() podemos observar que el resultado es similar al método de porter, este método permite reducir las palabras a su forma raíz, como por ejemplo, la palabra united es ahora unit, la palabra states ahora es state, preparers se convirtió en prepar, entre otras.



```
Command Prompt

///// TEXTO SIN PALABRAS VACIAS (BASE PARA TRUNCADAS, SNOWBALL STEMMER Y LEMMATIZER)

['project', 'gutenberg', 'ebook', 'grimm', 'fairy', 'tales', 'jacob', 'grimm', 'wilhelm', 'grimm', 'ebook', 'use', 'anyone', 'anywhere', 'united', 'states', 'parts', 'world', 'cost', 'almost', 'restrictions', 'whatsoever', 'may', 'copy', 'give', 'away', 'reuse', 'terms', 'project', 'gutenberg', 'license', 'included', 'ebook', 'online', 'wwwgutenbergorg', 'located', 'united', 'states', 'check', 'laws', 'country', 'located', 'using', 'ebook', 'title', 'grimm', 'fairy', 'tales', 'author', 'jacob', 'grimm', 'wilhelm', 'grimm', 'translators', 'edgar', 'taylor', 'marian', 'edwardes', 'release', 'date', 'april', '2001', 'ebook', '2591', 'recently', 'updated', 'june', '28', '2021', 'language', 'english', 'character', 'set', 'encoding', 'utf8', 'produced', 'emma', 'dudding', 'john', 'bickers', 'dagny', 'david', 'widger', '', 'start', 'project', 'gutenberg', 'ebook', 'grimm', 'fairy', 'tales', '', 'grimm', 'fairy', 'tales', 'jacob', 'grimm', 'wilhelm', 'grimm', 'preparers']

///// TEXTO CON PALABRAS TRUNCADAS

['project', 'gutenberg', 'ebook', 'grimm', 'fairy', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'ebook', 'use', 'anyon', 'anywher', 'unit', 'state', 'part', 'world', 'cost', 'almost', 'restrict', 'whatsoev', 'may', 'copi', 'give', 'away', 'reus', 'term', 'project', 'gutenberg', 'licens', 'includ', 'ebook', 'online', 'wwwgutenbergorg', 'locat', 'unit', 'state', 'check', 'law', 'countri', 'locat', 'use', 'ebook', 'titl', 'grimm', 'fairy', 'tale', 'author', 'jacob', 'grimm', 'wilhelm', 'grimm', 'translat', 'edgar', 'taylor', 'marian', 'edward', 'releas', 'date', 'april', '2001', 'ebook', '2591', 'recent', 'updat', 'june', '28', '2021', 'languag', 'english', 'charact', 'set', 'encod', 'utf8', 'produc', 'emma', 'dud', 'john', 'bicker', 'dagni', 'david', 'widger', '', 'start', 'project', 'gutenberg', 'ebook', 'grimm', 'fairy', 'tale', '', 'grimm', 'fairy', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'prepar']

///// TEXTO SNOWBALL STEMMER

['project', 'gutenberg', 'ebook', 'grimm', 'fairy', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'ebook', 'use', 'anyon', 'anywher', 'unit', 'state', 'part', 'world', 'cost', 'almost', 'restrict', 'whatsoev', 'may', 'copi', 'give', 'away', 'reus', 'term', 'project', 'gutenberg', 'licens', 'includ', 'ebook', 'online', 'wwwgutenbergorg', 'locat', 'unit', 'state', 'check', 'law', 'countri', 'locat', 'use', 'ebook', 'titl', 'grimm', 'fairy', 'tale', 'author', 'jacob', 'grimm', 'wilhelm', 'grimm', 'translat', 'edgar', 'taylor', 'marian', 'edward', 'releas', 'date', 'april', '2001', 'ebook', '2591', 'recent', 'updat', 'june', '28', '2021', 'languag', 'english', 'charact', 'set', 'encod', 'utf8', 'produc', 'emma', 'dud', 'john', 'bicker', 'dagni', 'david', 'widger', '', 'start', 'project', 'gutenberg', 'ebook', 'grimm', 'fairy', 'tale', '', 'grimm', 'fairy', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'prepar']
```

- b. El método WordNetLemmatize() podemos observar que cambian algunas palabras regresando a su forma raíz como la palabra parts es part, grims es grim; aplicando para las palabras en plural, es decir las que terminan en s, como states es ahora state; sin embargo hay palabras que no cambiaron como preparers sigue siendo preparar; también observamos que las que están conjugadas en pasado las mantiene de la misma forma como por ejemplo: united sigue siendo united, included sigue siendo included, entre otras.

```

Command Prompt

///// TEXTO SIN PALABRAS VACIAS (BASE PARA TRUNCADAS, SNOWBALL STEMMER Y LEMMATIZER)

['project', 'gutemberg', 'ebook', 'grimm', 'fairy', 'tales', 'jacob', 'grimm', 'wilhelm', 'grimm', 'ebook', 'use', 'anyone', 'anywhere', 'united', 'states', 'parts', 'world', 'cost', 'almost', 'restrictions', 'whatsoever', 'may', 'copy', 'give', 'away', 'reuse', 'terms', 'project', 'gutemberg', 'license', 'included', 'ebook', 'online', 'wwwgutembergorg', 'located', 'united', 'states', 'check', 'laws', 'country', 'located', 'using', 'ebook', 'title', 'grimm', 'fairy', 'tales', 'author', 'jacob', 'grimm', 'wilhelm', 'grimm', 'translators', 'edgar', 'taylor', 'marian', 'edwardes', 'release', 'date', 'april', '2001', 'ebook', '2591', 'recently', 'updated', 'june', '28', '2021', 'language', 'english', 'character', 'set', 'encoding', 'utf8', 'produced', 'emma', 'dudding', 'john', 'bickers', 'dagny', 'david', 'widger', '', 'start', 'project', 'gutemberg', 'ebook', 'grimm', 'fairy', 'tales', '', 'grimm', 'fairy', 'tales', 'jacob', 'grimm', 'wilhelm', 'grimm', 'preparers']

///// TEXTO CON PALABRAS TRUNCADAS

['project', 'gutemberg', 'ebook', 'grimm', 'fairi', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'ebook', 'use', 'anyon', 'anywher', 'unit', 'state', 'part', 'world', 'cost', 'almost', 'restrict', 'whatsoev', 'may', 'copi', 'give', 'away', 'reus', 'term', 'project', 'gutemberg', 'licens', 'includ', 'ebook', 'online', 'wwwgutembergorg', 'locat', 'unit', 'state', 'check', 'law', 'countri', 'locat', 'use', 'ebook', 'titl', 'grimm', 'fairi', 'tale', 'author', 'jacob', 'grimm', 'wilhelm', 'grimm', 'translat', 'edgar', 'taylor', 'marian', 'edward', 'releas', 'date', 'april', '2001', 'ebook', '2591', 'recent', 'updat', 'june', '28', '2021', 'languag', 'english', 'character', 'set', 'encod', 'utf8', 'produc', 'emma', 'dud', 'john', 'bicker', 'dagni', 'david', 'widger', '', 'start', 'project', 'gutemberg', 'ebook', 'grimm', 'fairi', 'tale', '', 'grimm', 'fairi', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'prepar']

///// TEXTO SNOWBALL STEMMER

['project', 'gutemberg', 'ebook', 'grimm', 'fairi', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'ebook', 'use', 'anyon', 'anywher', 'unit', 'state', 'part', 'world', 'cost', 'almost', 'restrict', 'whatsoev', 'may', 'copi', 'give', 'away', 'reus', 'term', 'project', 'gutemberg', 'licens', 'includ', 'ebook', 'online', 'wwwgutembergorg', 'locat', 'unit', 'state', 'check', 'law', 'countri', 'locat', 'use', 'ebook', 'titl', 'grimm', 'fairi', 'tale', 'author', 'jacob', 'grimm', 'wilhelm', 'grimm', 'translat', 'edgar', 'taylor', 'marian', 'edward', 'releas', 'date', 'april', '2001', 'ebook', '2591', 'recent', 'updat', 'june', '28', '2021', 'languag', 'english', 'character', 'set', 'encod', 'utf8', 'produc', 'emma', 'dud', 'john', 'bicker', 'dagni', 'david', 'widger', '', 'start', 'project', 'gutemberg', 'ebook', 'grimm', 'fairi', 'tale', '', 'grimm', 'fairi', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'prepar']

///// TEXTO LEMMATIZER

['project', 'gutemberg', 'ebook', 'grimm', 'fairy', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'ebook', 'use', 'anyone', 'anywhere', 'united', 'state', 'part', 'world', 'cost', 'almost', 'restriction', 'whatsoever', 'may', 'copy', 'give', 'away', 'reuse', 'term', 'project', 'gutemberg', 'license', 'included', 'ebook', 'online', 'wwwgutembergorg', 'located', 'united', 'state', 'check', 'law', 'country', 'located', 'using', 'ebook', 'title', 'grimm', 'fairy', 'tale', 'author', 'jacob', 'grimm', 'wilhelm', 'grimm', 'translator', 'edgar', 'taylor', 'marian', 'edwardes', 'release', 'date', 'april', '2001', 'ebook', '2591', 'recently', 'updated', 'june', '28', '2021', 'language', 'english', 'character', 'set', 'encoding', 'utf8', 'produced', 'emma', 'dudding', 'john', 'bicker', 'dagny', 'david', 'widger', '', 'start', 'project', 'gutemberg', 'ebook', 'grimm', 'fairy', 'tale', '', 'grimm', 'fairy', 'tale', 'jacob', 'grimm', 'wilhelm', 'grimm', 'preparers']

C:\Users\mini_>

```

V. CONCLUSIONES

La aplicación de los algoritmos de Porter, segmentación y lematización reveló diferencias en la forma en que cada método aborda la reducción de palabras a su forma base. Porter demostró ser un proceso más completo, observando más resultados al momento de truncarlas, regresándolas a su forma raíz y simplificándolas. Stemmer, por otro lado, mostró una moderación en la truncación, manteniendo un equilibrio entre la reducción y la comprensión semántica. En cambio, la lematización proporcionó resultados menos cercanos a los esperados. De esta forma, esta práctica nos permitió diferenciar los diferentes métodos para trincar las palabras, para así escoger el método más eficiente para utilizar en los siguientes códigos en los que necesitemos para la recuperación de información.

VI. BIBLIOGRAFÍA

1. NLTK. (2022). SnowballStemmer. Recuperado de <https://www.nltk.org/api/nltk.stem.SnowballStemmer.html?highlight=stopwords>
2. NLTK. (2022). WordNetLemmatizer. Recuperado de <https://www.nltk.org/api/nltk.stem.WordNetLemmatizer.html?highlight=wordnet>