

Laboratorio 4: Vocabulario

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

Ruiz Ramírez, Gabino

Benemérita Universidad Autónoma de Puebla

Recuperación de la Información

Fecha: 13 Febrero 2024

Resumen: En esta práctica se realizó un código en lenguaje Python, que permita obtener el vocabulario de un documento dado, con ayuda del resultado de las practicas de laboratorio 1, 2 y 3 en que se realizó el preproceso que requiere el documento para poder obtener correctamente lo deseado, es decir, tokenizaremos las palabras, eliminaremos signos y palabras vacías, convertiremos en minúsculas y las truncaremos. Al obtener el vocabulario podemos observar que son muchos términos encontrados, para eso crearemos otro documento que guarde el vocabulario reducido, en este caso, se eliminaron los términos que tenían solo 2 letras, de esta forma podremos observar que hay por ejemplo 100 términos menos al original.

Palabras clave: tokens, palabras vacías, re, nltk, Python, porter, lower, trincar, Split, stopwords, NPL, vocabulario, sorted.

I. INTRODUCCIÓN

La realización de este trabajo, encuadrado en el Laboratorio 4: Recuperación de Información, surge de la necesidad de consolidar y aplicar los conocimientos y habilidades adquiridos en los laboratorios previos sobre procesamiento de lenguaje natural y análisis de textos, con el objetivo de obtener un vocabulario detallado de la colección NPL. Este laboratorio específico se centra en la utilización de técnicas avanzadas para la extracción de vocabulario, implementando procesos de truncamiento y propuestas de reducción del vocabulario, que son esenciales para optimizar la búsqueda y recuperación de información dentro de grandes conjuntos de datos textuales.

El procesamiento de lenguaje natural (NLP) es una rama de la inteligencia artificial que juega un papel crucial en la interpretación y manipulación del lenguaje humano por parte de las máquinas, facilitando así la interacción entre humanos y computadoras. En este contexto, la colección NPL representa un conjunto de datos valioso para la exploración y

aplicación de técnicas de NLP, permitiendo a los estudiantes profundizar en el manejo de textos y el desarrollo de habilidades técnicas fundamentales para su futuro profesional.

Este laboratorio se propone no solo como una forma de aplicar conocimientos teóricos, sino también como un medio para entender la importancia del preprocesamiento de textos y la gestión eficiente del vocabulario en la recuperación de información. La elección de Python y NLTK como herramientas para este laboratorio refleja la relevancia de estas tecnologías en el campo del NLP, proporcionando a los estudiantes una base sólida en el uso de herramientas y librerías especializadas en el procesamiento de lenguaje.

II. OBJETIVO Y PLANTEAMIENTO DEL PROBLEMA

El objetivo principal de este trabajo es desarrollar y aplicar técnicas avanzadas de procesamiento de lenguaje natural para obtener y optimizar un vocabulario específico de la colección NPL, empleando para ello los conocimientos y herramientas adquiridos en los laboratorios anteriores. A través de este laboratorio, se busca profundizar en la comprensión y aplicación de métodos de preprocesamiento de texto, truncamiento, y técnicas de reducción de vocabulario, con el fin de mejorar la eficiencia en la recuperación de información dentro de grandes volúmenes de datos textuales.

III. DESARROLLO EXPERIMENTAL

Para realizar esta práctica necesitaremos:

- Software Python
- Librería NLTK
- Colección NPL de la página: https://ir.dcs.gla.ac.uk/resources/test_collections/npl/
- Laboratorios 1, 2 y 3.

1. Como primer paso crearemos una función que nos permite obtener el vocabulario de un documento txt:

```
def get_vocab(doc):
    doc = nltk.word_tokenize(doc.lower().replace(' ', ''))
    vocab = set(doc)
    return vocab

# Función para obtener el vocabulario de un documento
```

En seguida indicaremos en variables los documentos de los que haremos uso para obtener el vocabulario, y generaremos documentos en donde se guardaran los resultados obtenidos:

```

11 # Carpeta donde se encuentran los archivos .txt
12 directorio = r'C:\Users\mini_OneDrive\Documentos\Code Test\TEST 1\lab4'
13 directorio = r'C:\Users\mini_OneDrive\Documentos\Code Test\TEST 1\lab4\documentos'
14 output_file = r'C:\Users\mini_OneDrive\Documentos\Code Test\TEST 1\lab4\vocabularioTruncado.txt'
15 output_file_final = r'C:\Users\mini_OneDrive\Documentos\Code Test\TEST 1\lab4\vocabularioReducidoT.txt'
16

```

Ahora crearemos una lista donde se guarde el vocabulario obtenido:

```

17 # Lista para almacenar el vocabulario de todos los archivos
18 vocabulario_total = set()

```

Creando un ciclo que permite leer los documento txt indicando la carpeta donde se encuentran los documentos en este caso en variable anteriormente defina se llama directorio, con una condicional que indique que solo lea los archivos con extensión txt. En seguida creamos la ruta que indica finalmente el documento del cual obtendremos el vocabulario, una vez creada la ruta, procederemos a obtener el vocabulario de dicho documento, para así guardar el resultado en la lista creada anteriormente llamada vocabulario total:

```

20 # Recorremos todos los archivos en el directorio
21 for archivo in os.listdir(directorio):
22     if archivo.endswith('.txt'):
23         ruta_archivo = os.path.join(directorio, archivo)
24         vocabulario_archivo = obtener_vocabulario(ruta_archivo)
25         vocabulario_total.update(vocabulario_archivo)

```

Por último, ordenaremos alfabéticamente el vocabulario obtenido con ayuda de la función sorted:

```

27 # Ordenamos el vocabulario alfabéticamente
28 vocabulario_ordenado = sorted(vocabulario_total)
29

```

2. Volveremos a recorrer la colección para que esta vez calcule la cantidad de términos encontrados para el documento de vocabulario.

```

31 for archivo in os.listdir(directorio):
32     if archivo.endswith('.txt'):
33         ruta_archivo = os.path.join(directorio, archivo)
34         palabras_archivo = obtener_vocabulario(ruta_archivo)
35         cantidad_palabras = len(palabras_archivo)

```

Para el vocabulario reducido, ocupamos la condición de que si hay palabras que tengan solo 2 letras no sean guardadas en una nueva lista, es decir serán eliminadas del vocabulario.

```

38 # Eliminamos palabras de 2 caracteres o menos
39 vocabulario_filtrado = [palabra for palabra in vocabulario_ordenado if len(palabra) > 2]
40

```

Por último, identificaremos la cantidad de términos que hay en el vocabulario ya reducido:

```

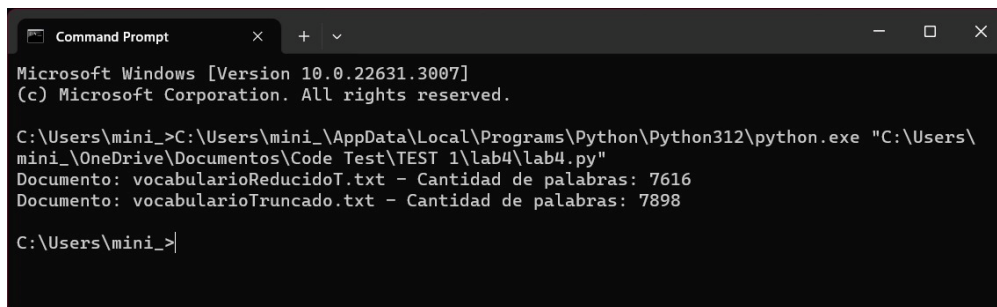
46 for archivo in os.listdir(directoriado):
47     if archivo.endswith('.txt'):
48         ruta_archivo = os.path.join(directoriado, archivo)
49         palabras_archivo = obtener_vocabulario(ruta_archivo)
50         cantidad_palabras = len(palabras_archivo)
51         print(f"Documento: {archivo} - Cantidad de palabras: {cantidad_palabras}")
52

```

IV. DISCUSIÓN Y RESULTADOS

En esta práctica logramos obtener los siguientes resultados:

En nuestra ejecución en la terminal nos indica la cantidad de términos encontrados para el primer documento con el vocabulario inicial, y al mismo tiempo la cantidad de términos encontrados para nuestro segundo documento con el vocabulario reducido, indicando que solo hay una diferencia 282 términos que cuentan con 2 letras.



```

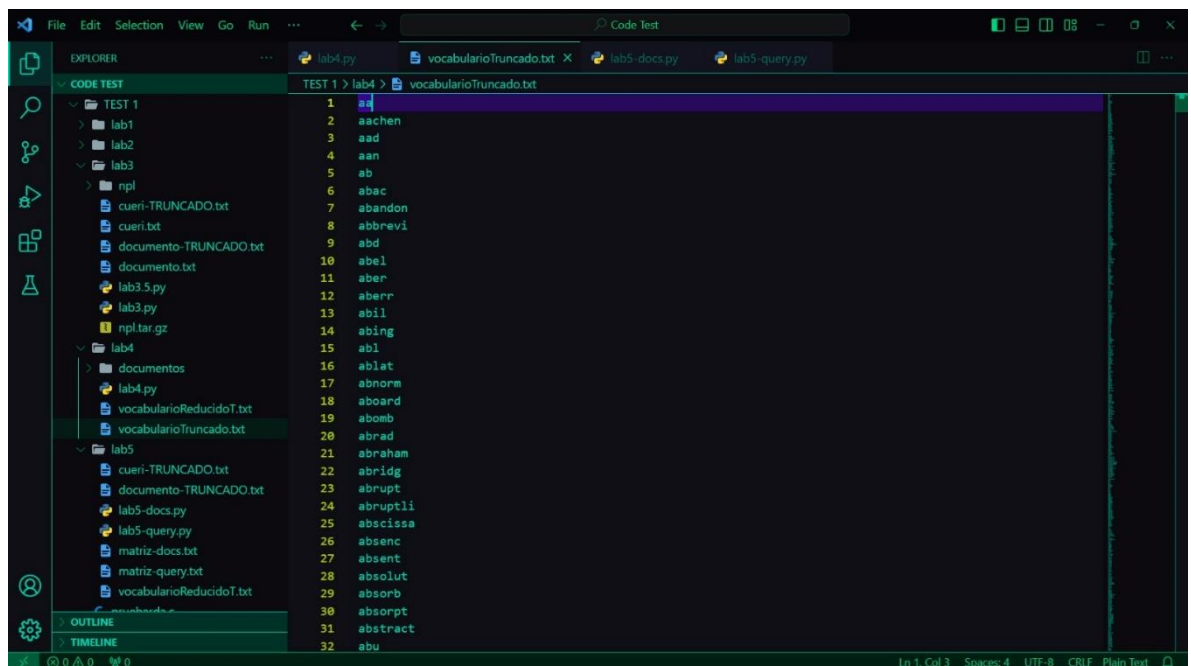
Microsoft Windows [Version 10.0.22631.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mini_>C:\Users\mini\AppData\Local\Programs\Python\Python312\python.exe "C:\Users\mini_OneDrive\Documentos\Code Test\TEST 1\lab4\lab4.py"
Documento: vocabularioReducidoT.txt - Cantidad de palabras: 7616
Documento: vocabularioTruncado.txt - Cantidad de palabras: 7898

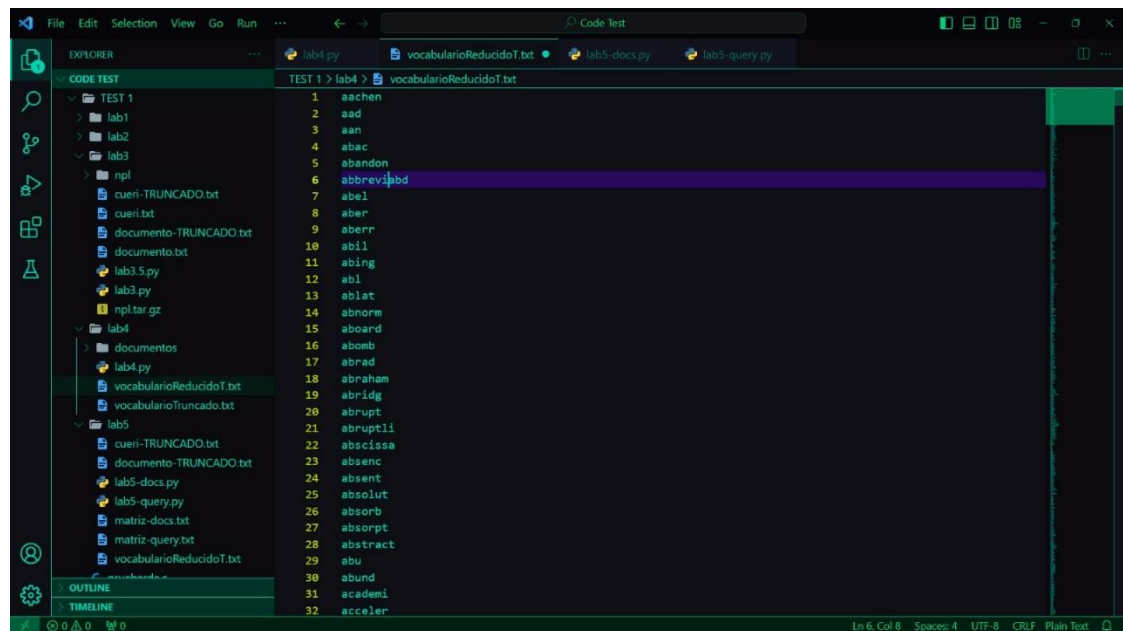
C:\Users\mini_>

```

1. Como primer resultado, podemos observar que esta ordenado de forma alfabéticamente el vocabulario, de igual forma identificamos que hay palabras que solo tienen 2 letras, y que el vocabulario cuenta con 7898 términos:



2. Como segundo resultado observamos el vocabulario reducido, con tan solo 7616 términos y al mismo tiempo nos percatamos de que ya no se encuentran las palabras que tiene solo 2 letras:



V. CONCLUSIONES

El trabajo realizado en el Laboratorio 4: Recuperación de Información ha permitido abordar de manera práctica y efectiva el desafío de optimizar un vocabulario extraído de la colección NPL, aplicando técnicas avanzadas de procesamiento de lenguaje natural. A través de los pasos seguidos y los resultados obtenidos, hemos podido consolidar y expandir nuestro conocimiento y habilidades en el campo del NLP, especialmente en lo que respecta a la recuperación de información y la gestión de vocabulario.

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>