

# Vocabulary Challenge

Fausto Morales

[ffmogbaj@gmail.com](mailto:ffmogbaj@gmail.com)

## 1. Trabajo realizado

En este ejercicio se retó para que procesara el texto de “Los miserables” parte 1 del url:

<https://cdn.pruebat.org/recursos/recursos/libros/pdf/Los-miserables.pdf> donde se busca:

- Estandarizar las palabras de su contenido.
- Crear un vocabulario.
- Almacenar el vocabulario en formato parquet.
- Generar estadísticas.

Para realizar el challenge se decidió emplear el lenguaje de Python, el código se encuentra en: Vocabulary.ipynb

Se importaron librerías y posteriormente se definieron las rutas relativas para realizar el ejercicio.

Para poder leer el pdf se realizó la siguiente función:

```
pdf_to_csv(pdf_path, csv_path):
```

Esta función nos generó el archivo: “Los-miserables.csv” Donde todo el pdf se volvió en un csv separado por líneas y comas. En la figura 1 se muestra un fragmento del resultado csv.

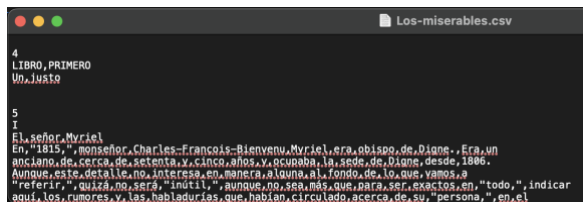


Figura 1.- Extracto del resultado csv generado por la función pdf\_to\_csv

Una vez con ello empezamos la primera actividad: Estandarización y creación de vocabulario en memoria. Para ello realizamos la función:

```
def csv_clean(input_csv_path, output_csv_path):
```

La cual además regresará el vocabulario.

Con la cual barrimos todo el csv y definimos la función:

```
def limpiar_texto(texto):
```

Para poder estandarizar. Aplicamos funciones para volver todas las palabras en minúsculas y no se tomaran como palabras distintas, así como incluimos una expresión regular para quedarnos sólo con caracteres válidos para el idioma español:

```
re.sub(r'[^a-záéíóúñ]+', '', texto)
```

Y creamos el elemento de “vocabulary” para almacenar todas las palabras distintas y en caso de duplicarse sumar la cuenta del número de palabras.

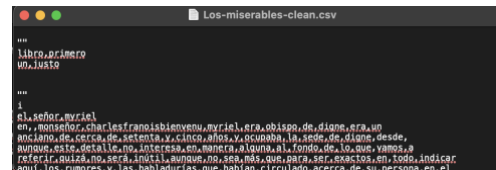


Figura 2.- Extracto del resultado csv generado por la función csv\_clean

En esta función almacenamos el csv limpio en el archivo: “Los-miserables-clean.csv”. Se puede ver un fragmento de la estandarización en la Figura 2.

Dado que con esta misma función realizamos el vocabulary con las palabras distintas y su contador de repetición en el texto se procedió a almacenar la información en un archivo .parquet a través de un dataframe siguiendo el siguiente código:

```
vocab_df.to_parquet(parquet_path,  
engine='pyarrow')
```

El resultado de este paso es el archivo Los-miserables.parquet.

Adicionalmente para poder obtener estadísticas se generaron las siguientes funciones:

Para obtener el número de palabras distintas:

```
len(vocabulary)-1
```

Para obtener el total de palabras distintas ignorando el vacío:

```
sum(value for key, value in  
vocabulary.items() if key != "")
```

Mientras que para obtener las 100 palabras más frecuentes se aplicó:

```
# Ordenar el diccionario por frecuencia (de  
mayor a menor)  
sorted_vocabulary_desc =  
sorted(vocabulary.items(), key=lambda item:  
item[1], reverse=True)  
  
# Imprimir las 100 palabras más frecuentes  
print("Las 100 palabras más frecuentes:")
```

```
for word, freq in sorted_vocabulary_desc[:100]:
    print(f"Palabra: {word}, Frecuencia: {freq}")
```

Mientras que para obtener las 100 palabras menos frecuentes se aplicó una lógica similar:

```
# Ordenar el diccionario por frecuencia (de menor a mayor)
sorted_vocabulary_asc = sorted(vocabulary.items(),
key=lambda item: item[1])

# Imprimir las 100 palabras menos frecuentes
print("\nLas 100 palabras menos frecuentes:")
for word, freq in sorted_vocabulary_asc[:100]:
    print(f"Palabra: {word}, Frecuencia: {freq}")
```

## 2. Resultados

A partir de este código que puede ser consultado a detalle en: Vocabulary.ipynb

Se obtuvo:

- Que el número total de palabras distintas es de: 13,255 sin considerar vacíos y manteniendo acentos.
- Que el número total de palabras eliminando vacíos y conservando acentos es: 109,276
- Se obtuvieron las 100 palabras más frecuentes, por mencionar algunas:  
Palabra: “de” se obtuvo frecuencia: 5,324  
Palabra: “la”, Frecuencia: 3,918  
Palabra: “que”, Frecuencia: 3,505  
Palabra: “y”, Frecuencia: 3,123  
Palabra: “el”, Frecuencia: 3,081
- Se obtuvieron las 100 palabras menos frecuentes, por mencionar algunas:  
Palabra: “dominio”, Frecuencia: 1  
Palabra: “prohibida”, Frecuencia: 1  
Palabra: “ajenos”, Frecuencia: 1  
Palabra: “zúrich”, Frecuencia: 1  
Palabra: “carso”, Frecuencia: 1
- También se generaron los siguientes archivos a partir del PDF:  
Los-miserables.csv  
Los-miserables-clean.csv  
Los-miserables.parquet

## 3. Conclusiones

Se aprendió a hacer los pasos iniciales para poder arrancar a preparar la data de tipo texto para hacer uso de modelos que puedan procesar texto.

El tiempo dedicado para esta actividad fue de algunas 4 horas por 2 días, entre código y reporte.

Se generó un vocabulario a partir de un libro y se extrajeron principales características como lo son: Número total de palabras, el número total de palabras distintas, así como palabras con mayor y menor frecuencia.

Este ejercicio se realizó en Jupyter, Python y empleando Mac.

Debido a la carga de trabajo laboral de esta semana no fue posible realizar el Addendum.