

Python

Procesamiento de Lenguaje Natural

Segmentación de Textos



Rogelio Ferreira Escutia

Profesor / Investigador
Tecnológico Nacional de México
Campus Morelia



Segmentación de Textos

Tokenización (1)

- **Una vez que se ha hecho un pre-procesamiento del texto para poder eliminar algunos símbolos y caracteres no deseados, el siguiente paso es la segmentación del texto, es decir, separar el texto en palabras, donde cada palabra encontrada dentro del texto lo convertiremos a una lista de palabras importantes ó “tokens” (como se le denomina en inglés).**
- **Al proceso completo de pasar un texto a un conjunto de palabras importantes para nuestro análisis se le denomina “Tokenización” (que proviene de la palabra “Token” en inglés, y es mas conocido de esta manera).**

Tokenización (2)

- Para lograr encontrar palabras relevantes dentro de un texto y hacer la “tokenización”, se requiere eliminar las palabras que aporten poca relevancia a nuestro análisis (como son los artículos “el, la, los”, etc.) y a los cuales se les denominó “Stopwords” ó “palabras vacías” en español.
- Cada lenguaje tiene su propio conjunto de “stopwords” y se deberá tener una lista con las “stopwords” que se vayan a eliminar.
- En el caso de la biblioteca NLTK, ya se cuenta con esta lista, la cual está disponible para diferentes lenguajes.

NLTK (1)

- Lo primero será instalar nuestra lista de “stopwords” para utilizarse dentro de NLTK, para ello entraremos a nuestra línea de comandos (terminal) y escribiremos “python3” (ó solamente “python, de acuerdo al que se haya instalado) para invocar al intérprete de Python y se observará lo siguiente:

```
[rogelioferreiraescutia@Mac-mini-de-Rogelio ~ % python3 ]
Python 3.10.0 (v3.10.0:b494f5935c, Oct  4 2021, 14:59:19) [Clang 12.0.5 (clang-1
205.0.22.11)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> □
```

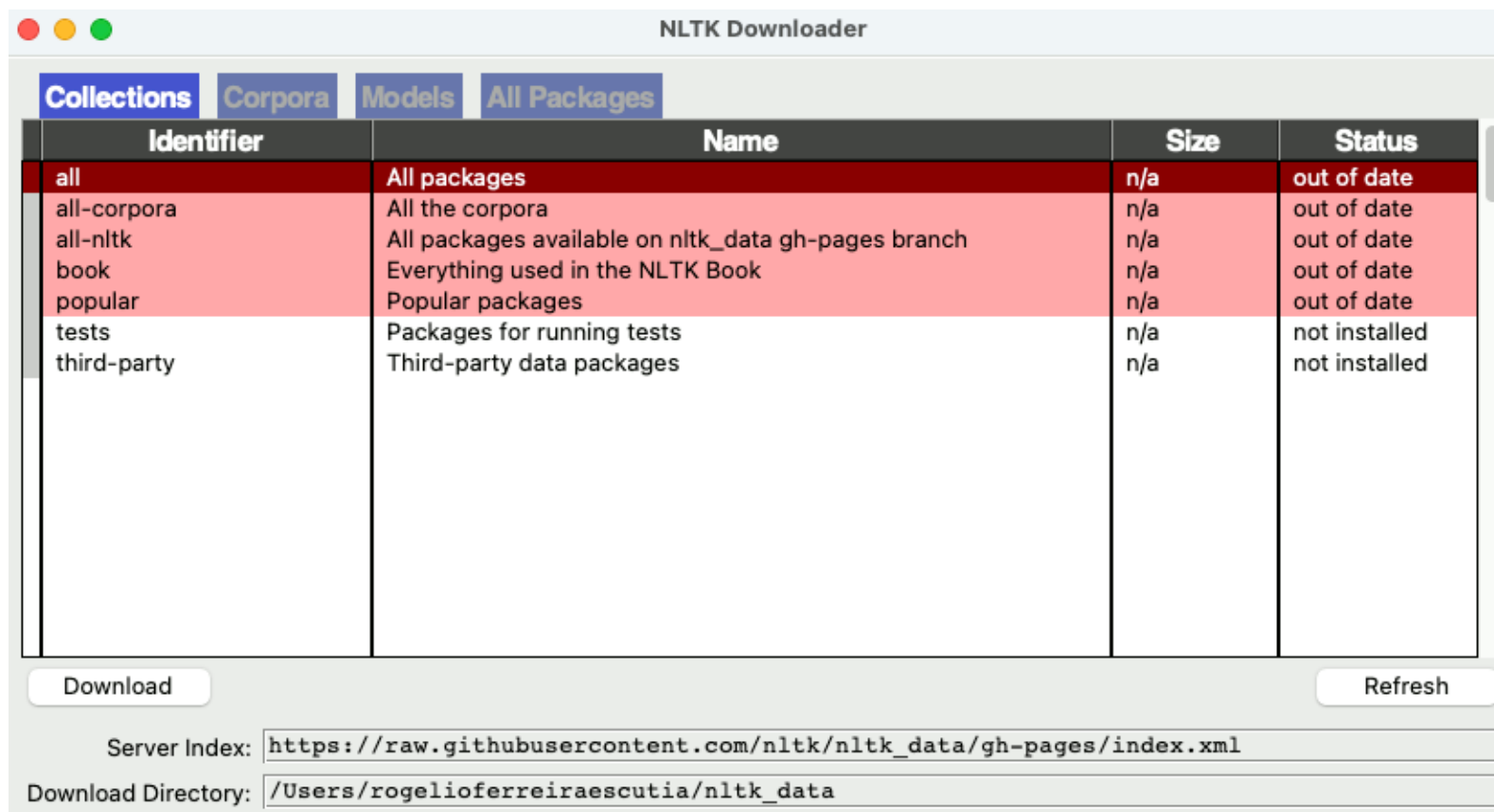
NLTK (2)

- Ya dentro del intérprete vamos a importar NLTK y cargaremos el “NLTK Downloader”, que es la herramienta gráfica para mostrar los “Corpus” y herramientas que se han descargado a nuestra computadora, y para ello escribimos lo siguiente en el intérprete:

```
[>>> import nltk  
[>>> nltk.download()
```

NLTK (3)

- Se abre una ventana modo gráfico y se observará la pantalla del “NLTK Downloader”, donde se encuentran todas las herramientas del NLTK que se han descargado de manera local a nuestra computadora:



The screenshot shows the NLTK Downloader application window. It features a tabbed interface with 'All Packages' selected. Below the tabs is a table with the following data:

Identifier	Name	Size	Status
all	All packages	n/a	out of date
all-corpora	All the corpora	n/a	out of date
all-nltk	All packages available on nltk_data gh-pages branch	n/a	out of date
book	Everything used in the NLTK Book	n/a	out of date
popular	Popular packages	n/a	out of date
tests	Packages for running tests	n/a	not installed
third-party	Third-party data packages	n/a	not installed

At the bottom of the window, there are two buttons: 'Download' and 'Refresh'. Below these buttons are two input fields: 'Server Index' with the value `https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml` and 'Download Directory' with the value `/Users/rogelioferreiraescutia/nltk_data`.

NLTK (4)

- Ya dentro del intérprete importamos NLTK y luego importamos las “stopwords” por medio de los siguientes instrucciones:

```
>>> import nltk
>>> nltk.download('stopwords')
[nltk_data] Downloading package stopwords to
[nltk_data]      /Users/rogelioferreiraescutia/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True_
```

- Para salir en cualquier momento del intérprete de Python tecleamos lo siguiente:
exit()

StopWords (1)

- Si queremos saber cuáles son las “stopwords” que utiliza NLTK para el idioma español, abrimos con un editor de texto el archivo “spanish” que se encuentra en el siguiente directorio:

/Users/rogelioferreiraescutia/nltk_data/corpora/stopwords/

- Donde el directorio “rogelioferreiraescutia” cambia de acuerdo al nombre del usuario que está usando la computadora.

StopWords (2)

- Podemos imprimir directamente las “StopWords” en pantalla con la siguiente instrucción:

```
cat /Users/rogerioferreiraescutia/nltk_data/corpora/stopwords/spanish
```

- Actualmente se tienen 313 “stopwords” consideradas para el idioma español.
- A partir de este paso ya estamos listos para “tokenizar” y eliminar las “stopwords” de un texto.

Tokenizar (1)

- **Bibliotecas a utilizar:**

```
# Bibliotecas a utilizar
import re                # Manejo de expresiones regulares
import nltk              # Para procesar lenguaje natural
from nltk.tokenize import word_tokenize # Para Tokenizar un texto
from nltk.corpus import stopwords    # Cargar las "Stopwords" del español
```

Tokenizar (2)

- **Definir texto a analizar:**

```
# Definir una frase
frase_gabriela_mistral = "Como soy reina y fui mendiga, ahora vivo en puro temblor de que me dejes, y te
pregunto, pálida, a cada hora: -¿Estás conmigo aún? ¡Ay, no te alejes!- Quisiera hacer las marchas sonriendo y
confiando ahora que has venido; pero hasta en el dormir estoy temiendo y pregunto entre sueños: -¿No te has ido?
_"

# Definir la frase a procesar
texto = frase_gabriela_mistral

# Impresión en pantalla de la frase seleccionada
print("\nTexto Completo:\n\n", texto)
```

- **Salida:**

Texto Completo:

```
Como soy reina y fui mendiga, ahora vivo en puro temblor de que me dejes, y te pregunto, pálida, a cada hora:
-¿Estás conmigo aún? ¡Ay, no te alejes!- Quisiera hacer las marchas sonriendo y confiando ahora que has venid
o; pero hasta en el dormir estoy temiendo y pregunto entre sueños: -¿No te has ido?-
```

Tokenizar (3)

- **Limpiar texto y Tokenizar:**

```
# Eliminar simbolos y caracteres especiales usando expresiones regulares
texto_sin_simbolos = re.sub(r'^\w\s', '', texto)
```

```
# Convertimos a tokens todo el texto y lo imprimimos en pantalla
tokens_de_mi_texto = word_tokenize(texto_sin_simbolos)
print('\nImpresión de todos los tokens del texto:\n\n', tokens_de_mi_texto)
print('\n Tokens Totales: ', len(tokens_de_mi_texto))
```

- **Salida:**

Impresión de todos los tokens del texto:

```
['Como', 'soy', 'reina', 'y', 'fui', 'mendiga', 'ahora', 'vivo', 'en', 'puro', 'temblor', 'de', 'que', 'me', 'dejes', 'y', 'te', 'pregunto', 'pálida', 'a', 'cada', 'hora', 'Estás', 'conmigo', 'aún', 'Ay', 'no', 'te', 'a lejes', 'Quisiera', 'hacer', 'las', 'marchas', 'sonriendo', 'y', 'confiando', 'ahora', 'que', 'has', 'venido', 'pero', 'hasta', 'en', 'el', 'dormir', 'estoy', 'temiendo', 'y', 'pregunto', 'entre', 'sueños', 'No', 'te', 'has', 'ido']
```

Tokens Totales: 55

Tokenizar (4)

- Cargar “StopWords” y eliminarlas de nuestro texto:

```
# Cargamos las "stopwords" del español (las palabras que no nos aportan información)
palabras_vacias = set(stopwords.words('spanish'))
```

```
# Filtramos los tokens eliminando las "stopwords"
lista_final = []
for palabra in tokens_de_mi_texto:
    if palabra not in palabras_vacias:
        lista_final.append(palabra)
```

Tokenizar (5)

- **Impresión Final:**

```
# Impresión final de las palabras relevantes del texto (Tokenizar)
print('\nLista Final eliminando las palabras vacías (no relevantes):\n\n', lista_final)
print("\nTotal de Tokens sin Stopwords: ", len(lista_final),"\n")
```

- **Salida:**

Lista Final eliminando las palabras vacías (no relevantes):

```
['Como', 'reina', 'mendiga', 'ahora', 'vivo', 'puro', 'temblor', 'dejes', 'pregunto', 'pálida', 'cada', 'hora', 'Estás', 'conmigo', 'aún', 'Ay', 'alejés', 'Quisiera', 'hacer', 'marchas', 'sonriendo', 'confiando', 'ahora', 'venido', 'dormir', 'temiendo', 'pregunto', 'sueños', 'No', 'ido']
```

Total de Tokens sin Stopwords: 30



Rogelio Ferreira Escutia

Profesor / Investigador
Tecnológico Nacional de México
Campus Morelia



rogelio.fe@morelia.tecnm.mx



rogeplus@gmail.com



xumarhu.net



[@rogeplus](https://twitter.com/rogeplus)



[https://www.youtube.com/
channel/UC0on88n3LwTKxJb8T09sGjg](https://www.youtube.com/channel/UC0on88n3LwTKxJb8T09sGjg)



[rogelioferreiraescutia](https://www.linkedin.com/in/rogelioferreiraescutia)

