Session 1: ElasticSearch and Zipf's and Heaps' laws

Q1 2023/2024

Àlex Domínguez Rodríguez

Mario Fernández Simón

Introducción

En esta práctica estudiaremos cómo se distribuyen las palabras de 3 tipos de documentos de diferente ámbito utilizando la ley de Zipf y la ley de Heaps.
En primer lugar, estudiaremos si la distribución rango-frecuencia de las palabras se ajusta a una ley potencial y estudiaremos sus parámetros.
A continuación, miraremos la relación entre el número total de palabras y el número de términos diferentes.

Preparación previa

Lo primero a realizar será la indexación de los 3 documentos utilizando *IndexFiles.py*.
A continuación, será necesario filtrar la lista de palabras que obtenemos con *CountWords.py*, para ello hemos utilizado 2 métodos.
En primer lugar, eliminaremos todos los caracteres especiales modificando el propio *CountWords.py* utilizando la función de python *isalpha* que permite eliminar caracteres especiales y números. Seguidamente, hemos utilizado la librería *enchant,* para quedarnos solo con aquellas palabras que pertenezcan al diccionario inglés, esto lo hemos implementado en los scripts *Zipf.py* y *Heap.py*.

Ley de Zipf

El objetivo es comprobar si la ley de Zipf se comporta como una ley potencial. En los documentos a estudiar hemos utilizado la función *curve_fit* de la librería *scpicy* de python. Esta función ajusta los valores de a, b y c de la función de Zipf para encontrar aquellos que mejor se adapten. Por un lado, la línea azul (“ideal”) representa como debería ser la distribución según los valores ideales de a, b y c. Por el otro, la línea roja (“data”) representa los valores que se obtienen de nuestros documentos.
Primero, hemos tomado una muestra de las 750 palabras que más aparecen para ver cómo se comportan. Más tarde comprobaremos si al aumentar ese número se

producen cambios. Los valores de a los hemos limitado entre 0.5 y 1.5 (sabiendo que el teórico es 1) y evitando que, b y c tengan valores negativos, con los cuales nos darían frecuencias negativas. Las gráficas obtenidas son:

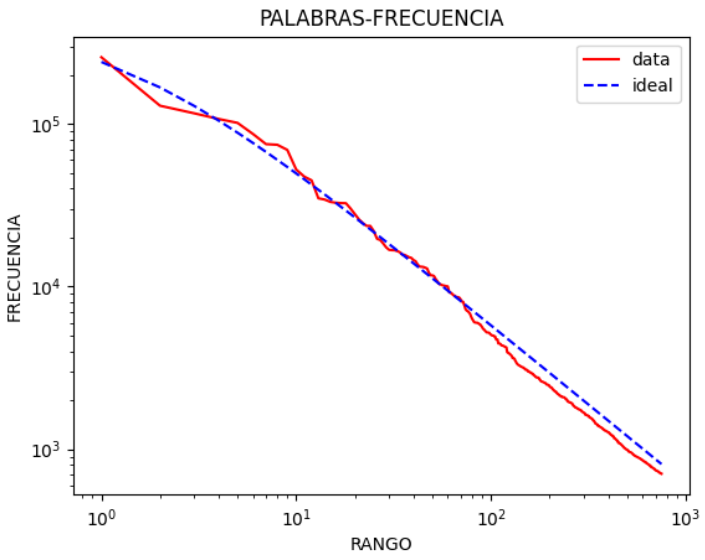

Figura 1. Gráfica logarítmica de 20Newsgroup.

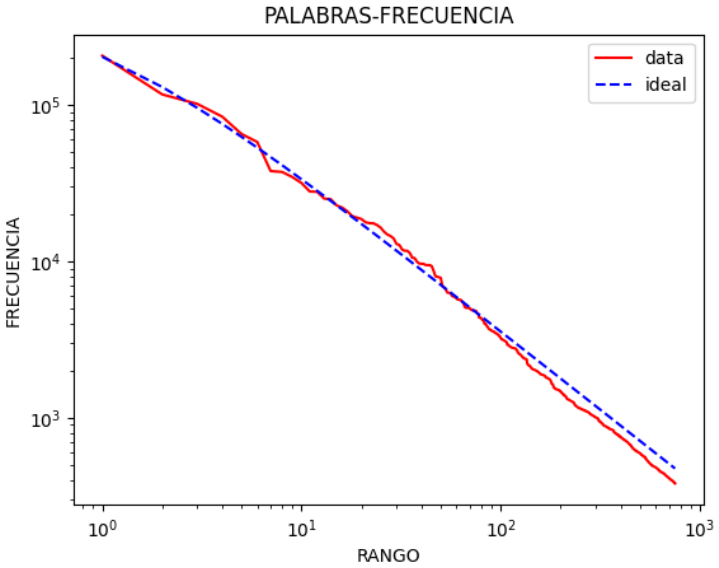

Figura 2. Gráfica logarítmica de novels.

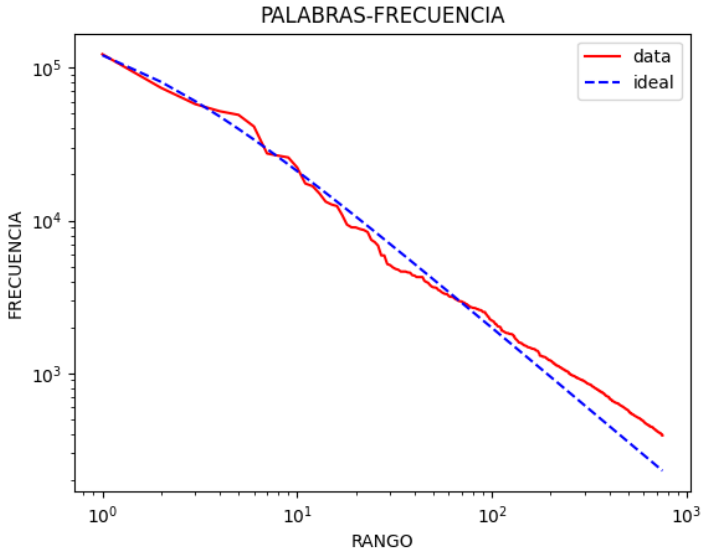

Figura 3. Gráfica logarítmica de arxivs.

	A	B	C
NEWS	0.979	1.245	5.315e+05
NOVELS	1.003	0.789	5.315e+05
ARXIVS	1.074	1.202	2.8247e+05

Tabla 1. Tabla de valores de a, b y c para los 3 tipos de archivos.

Se puede observar como en todos los casos la gráfica obtenida se ajusta bastante bien a los valores ideales siguiendo una distribución de ley potencial. En concreto, para los valores de a, en los 3 casos se obtienen valores muy cercanos al teórico (a=1).
En general, para los 2 primeros tipos de textos la gráfica obtenida se ajusta mejor que para el último que es de tipo científico. Esto se puede ver conforme aumenta el rango, la pendiente empieza a distanciarse del valor "ideal". El motivo por el cual esto sucede es por el tipo de palabras que hay en cada conjunto de textos.

Por último, probamos de no limitar el número de palabras a 750. Las gráficas que obtuvimos cada vez se parecían más a una curva que a una recta y los parámetros también variaron. Con lo cual, se puede observar que los parámetros no solo varían dependiendo del tipo de palabras sino que también lo hacen dependiendo del número de palabras que haya.

Ley de Heaps

La ley de Heaps explica la relación entre la cantidad de tokens (número total de palabras) y types (total de palabras distintas). El estudio de esta ley lo hemos hecho solo sobre el conjunto de novelas, ya que al ser un lenguaje más formal es plausible que se adapte a la ley.

En primer lugar, hemos creado grupos de 5, 10, 15, 20, 25 y 30 novelas. Los grupos son acumulativos entre ellos, el de 5 en el de 10, el de 10 en el de 15, etc.
Al igual que en el apartado anterior, hemos utilizado la función *curve_fit* para encontrar los valores de K y ß. Además, hemos probado a limitar de varias maneras

los valores para intentar ajustar los datos en la gráfica, pero al final, hemos comprobado que para encontrar los mejores valores no era necesario poner esos límites. Finalmente, se ha obtenido la siguiente función:

$$f(n) = 159*N^{0.816}$$

El valor esperado para la ß para textos en inglés es alrededor del 0.5. El nuestro es significativamente mayor (muy parecido al del Quijote), es posible que sea debido a que se han usado novelas de distintos autores que a su vez tienen distinto vocabulario y maneras de expresarse. También es posible que otro criterio u otro orden al escoger las novelas hubiera afectado a los resultados.

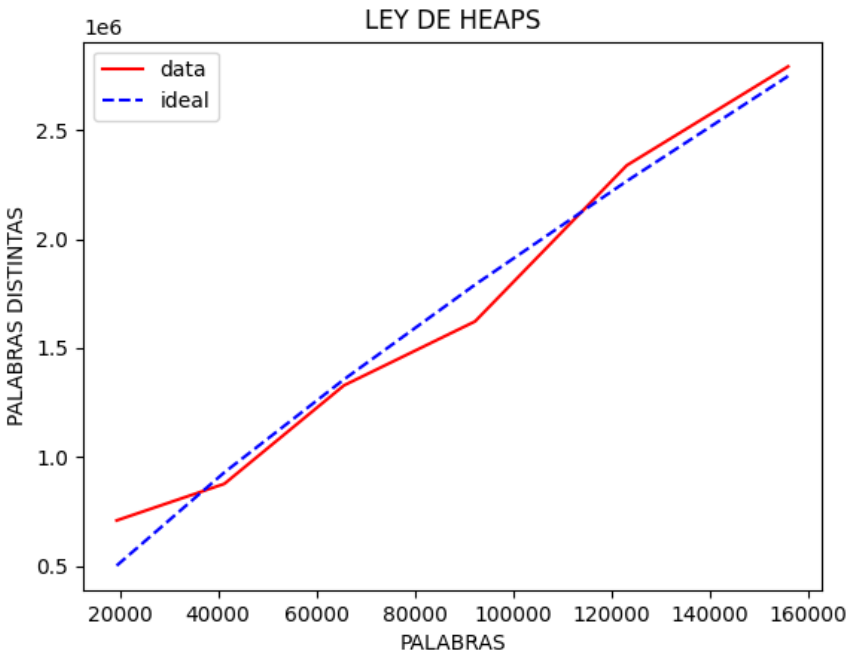

Figura 4: Gráfica logarítmico de la Ley de Heap

Pese a que nuestra ß es un poco mayor de lo normal, si observamos la gráfica vemos que los datos estudiados se adaptan más o menos bien a lo esperado por la ley y, por tanto, podemos concluir que nuestra muestra de datos cumple la ley de Heaps.