# Introduction to

# Elastic Search.

# Zipf ’s and Heaps’ laws

12-9-2022

He Chen

Daniel Muñoz

Grupo 12

**Metodología Zipf's law**

Inicialmente hemos seguido los pasos del documento, utilizando el elasticsearch y el indexFiles.py para indexar los archivos y el CountWords.py para que nos ponga en el archivo nombre_documento.txt (nombre_documento puede ser novels, news o arxiv_abs) dónde está ordenado ascendentemente por la primera columna que es la frecuencia y en la segunda la palabra. Después, con el script en python "borrar_basura.py" lo que hacemos es quitar los símbolos que no forman parte del inglés, números, puntos para eliminar urls y comparamos con un diccionario inglés para eliminar palabras que no pertenezcan. Este resultado lo guardamos en otro info_nombre_documento.txt donde ordenamos por frecuencia descendente. En la primera columna aparece la frecuencia y en la segunda la palabra. Finalmente ejectamos el script make_graph.py para crear una gráfica logarítmica de frecuencia-rango con los resultados obtenidos (línea continua) y la curva entrenada con la función curve fit para que de unos valores óptimos en los parámetros a, b y c (línea discontinua).

**Zipf's law**

La ley de Zipf es una ley empírica que relaciona frecuencia-rango con una ley potencial. Hemos usado la siguiente fórmula:

$$f = \frac{c}{(rank + b)^{a}}$$

Siguiendo los pasos explicados anteriormente hemos limitado las constantes en un intervalo que hemos considerado arbitrarios, donde:

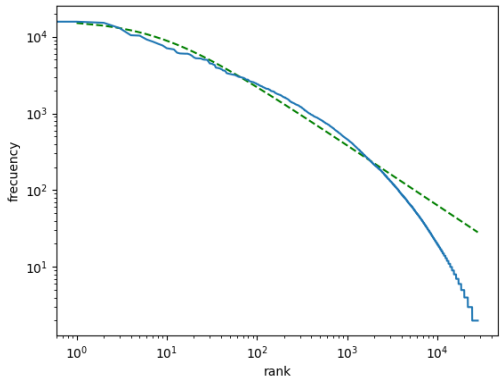
$a \in [0.7, 1.8]$ $b, c \in [-1000000, 1000000]$

Primero hemos calculado una gráfica borrando los stopwords, y otra con stopwords. Después analizamos la importancia de las palabras de frecuencia muy bajas en el resultado.

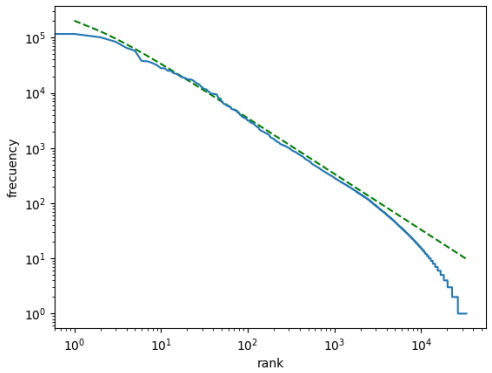
**Analisis Zipf's law**

Para el análisis de los resultados hemos usado como modelo el comportamiento de las gráficas explicado en el siguiente fichero:
https://upcommons.upc.edu/bitstream/handle/2117/180381/Ferrer-i-Cancho_and_Sole_JQL_2001.pdf?sequence=1&isAllowed=y
En la página 14 del documento podemos ver el comportamiento deseado. La gráfica empieza con una pequeña curva, seguido de una recta y finaliza con otra curva, las cuales representan diferentes power law.

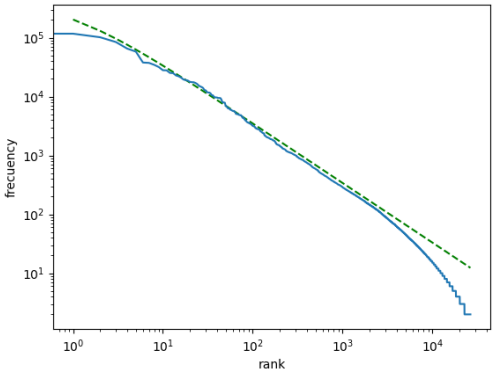
Las siguientes gráficas se han sacado usado el fichero novelas:




G1: gráfica borrando stop words




G2: gráfica con palabras de frecuencia 1



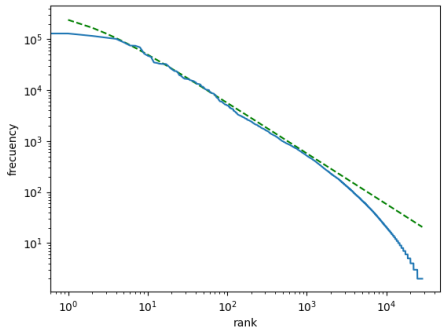

G3: gráfica con stop words y borrando palabras de frecuencia 1

| | a | b | c |
|---|---|---|---|
| G1 | 0.786 | 9.711 | 53175.023 |
| G2 | 1.013 | 0.827 | 374182.145 |
| G3 | 1.013 | 0.826 | 373897.795 |

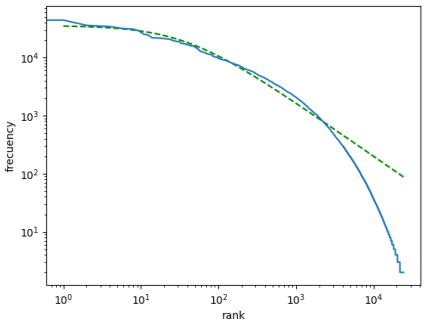
Como podemos ver, quitando los stopwords (G1) nos sale una gráfica con una curva con a = 0.786. La forma de la gráfica no es la que esperábamos, y el valor de la constante a se aleja mucho al valor teórico(1). Esto creemos que puede ser debido a que al eliminar las stopwords con una lista, puede haber algunas que no sean las más frecuentes y se encuentren por el centro.
En la gráfica G2 podemos ver el resultado teniendo en cuenta los stopwords. Esta vez la forma de la gráfica ya se acerca más al modelo que mencionamos antes (una curva seguido de una recta seguido de otra curva). Aunque en la última curva vemos escalones, y creemos que puede afectar la los valores de las constantes.

Para ello comparamos el resultado de quitar o no las palabras de frecuencia 1, que suponen un 20% de las palabras totales en la base de datos. No hemos quitado las palabras con frecuencia más alta ya que consideramos que quitar más de un 20% del total podría significar una parte importante del total de los datos.
Comparando las gráficas G2 y G3 vemos que casi no hay diferencia alguna en el resultado entre quitar o no palabras de baja frecuencia.

Quitando las palabras de baja frecuencia, aunque sean palabras que tienen significado, como hemos visto que no supone ningún cambio significativo, hemos decidido crear las gráficas de los otros dos grupos con las top words y sin las palabras de frecuencia 1.




G4: gráfica f-r de foros de internet
a = 0.9958  b = 1.3410  c = 559280.44




G5: gráfica f-r de textos científicos
a = 0.9240  b = 36.8488  c =  10^6

En la gráfica obtenida de foros de internet (G4), podemos observar que igual que la de novels (G3), se aproxima al modelo mencionado anteriormente y que el valor de a es prácticamente 1, qué es el valor que tiene en inglés. También, el parámetro a y la curva es muy similar. Podemos decir que tanto los foros como las novelas siguen muy bien la ley de Zipfs. En cambio, en los artículos científicos (G5) se ve una curva muy pronunciada y el valor de a es 0.9240 que está un poco más alejado de 1 de los otros. Esto puede ser debido al tipo de texto con el que estamos tratando, ya que este es más formal a diferencia de los dos anteriores que son más coloquiales.

## Heaps law

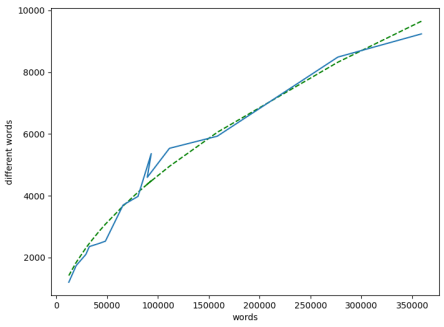
La ley de Heaps es una ley empírica que relaciona el número de palabras de un conjunto de documentos en función de la longitud de su longitud. Hemos usado la siguiente fórmula: $k * N^{\beta}$
donde k y β son constantes y N es el número de palabras diferentes. Nosotros los tomaremos como referencia, pero ampliando los rangos un poco más (k ∈ [1, 1000] y β ∈ [0, 2])

## Metodología Heaps law

Inicialmente contaremos las palabras de diferentes textos. Hemos escogido 16 páginas dependiendo del tamaño que ocupa (KB) en el .txt ya que hemos asumido que mientras más espacio ocupe más palabras habrá. Primero hemos utilizado el script Counting words para que nos deje en un .txt las palabras con su frecuencia. Después hemos utilizado el script borrar_basura.py comentado anteriormente para que nos deje las palabras que pertenecen al inglés. Finalmente hemos contado el número de palabras diferentes y el número de palabras totales con el script obtener_info.py que lo devuelve por la consola y lo hemos representado en una gráfica.

## Resultados Heaps law

Los valores obtenidos son: k = 6.62374771 β = 0.56944141.




En la gráfica vemos que al aumentar el número de palabras, el aumento del número de palabras diferentes va disminuyendo continuamente. Esto es lógico, ya que cuanto más números de palabras hay mayor probabilidad de que se repitan palabras. Si hiciéramos la gráfica con más datos veríamos que tiende a una constante, puesto que el número de palabras diferentes acaba variando muy poco.

## Conclusion Heaps law

Según lo que nos han explicado en teoría, el valor de la beta depende del tipo de texto y del lenguaje. En cuanto a los textos ingleses la beta está en 0.5 o inferior. Podemos decir que en nuestro caso, hemos conseguido un valor (0.569) bastante cercano al 0.5. Creemos que podríamos mejorar esta aproximación haciendo el experimento con más datos, ya que solo hemos usado 15 ficheros de páginas de novelas.