Prácticas - Recuperación de Información

Práctica 2 - Indexación de Textos

Iván Calle Gil Daniel López García Lothar Soto Palma
José Carlos Entrena Jiménez

Índice

1. Introducción **2**

2. Implementación **2**

2.1. Lectura de datos 2

2.2. Indexación 2

3. Ley de Zipf **2**

4. Trabajo en grupo **4**

1. Introducción

En esta práctica hemos realizado un programa en Java que utiliza el *stemmer* **snowball** para realizar un proceso de lexificación de un conjunto de ficheros de texto, que en este caso es la obra completa Don Quijote de la Mancha. Con esto, obtendremos un índice básico del texto y podremos comprobar si se cumple la ley de Zipf.

Este documento detalla el proceso de implementación, los resultados obtenidos y el trabajo realizado por todos los miembros del grupo.

2. Implementación

2.1. Lectura de datos

Al leer los ficheros de texto que vamos a tratar, y previo a la aplicación del *stemmer*, hemos de eliminar los signos de puntuación del fichero. Para esto, hemos utilizado una función llamada *removePunctuation*, que cambia las mayúsculas por minúsculas y elimina todos los caracteres no alfanuméricos, mediante el uso de una expresión regular. Así, nos aseguramos de que tratamos únicamente con palabras en minúscula sin ningún caracter especial.

Para el fichero con las palabras vacías no necesitamos ningún tipo de tratamiento, así que únicamente lo leemos. Como este fichero será el mismo independientemente del número de documentos que vayamos a indexar, lo leeremos una única vez y lo almacenaremos como variable de clase.

A la hora de leer los ficheros, el programa recibirá un *path* que, si es un directorio, recorreremos buscando todos los ficheros de texto que contenga, incluyendo aquellos que estén en subcarpetas, y almacenamos todo el texto en un *string*. De proporcionar el *path* a un fichero, simplemente indexaremos este.

2.2. Indexación

Una vez hemos leído los datos, utilizamos *StringTokenizer* para obtener los tokens sobre los que ir aplicando el *stemmer* uno a uno. Antes de llamar al método *stem*, hemos de comprobar que la palabra no esté en el conjunto de palabras vacías, que hemos leído previamente y tenemos almacenado en una tabla hash. Si es así, podemos continuar y obtener la raíz del token, la cual almacenaremos en una tabla hash en caso de ser nueva, o aumentaremos en uno el entero asociado a dicha raíz en la tabla hash.

3. Ley de Zipf

En los dos siguientes gráficos se muestran las frecuencias de todas las palabras del texto (Figura 1), y solo las 100 primeras (Figura 2).

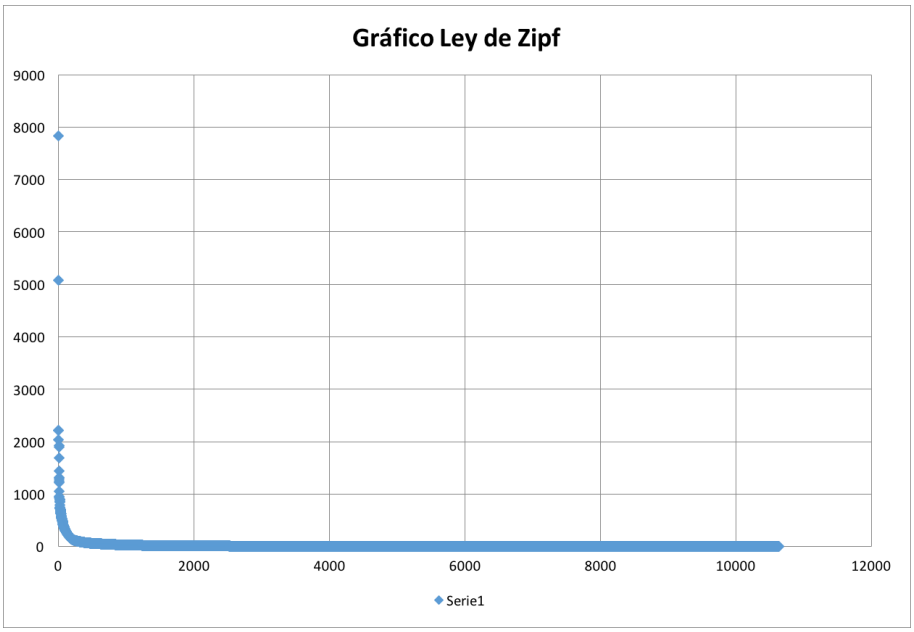

Figura 1: Ley de Zipf

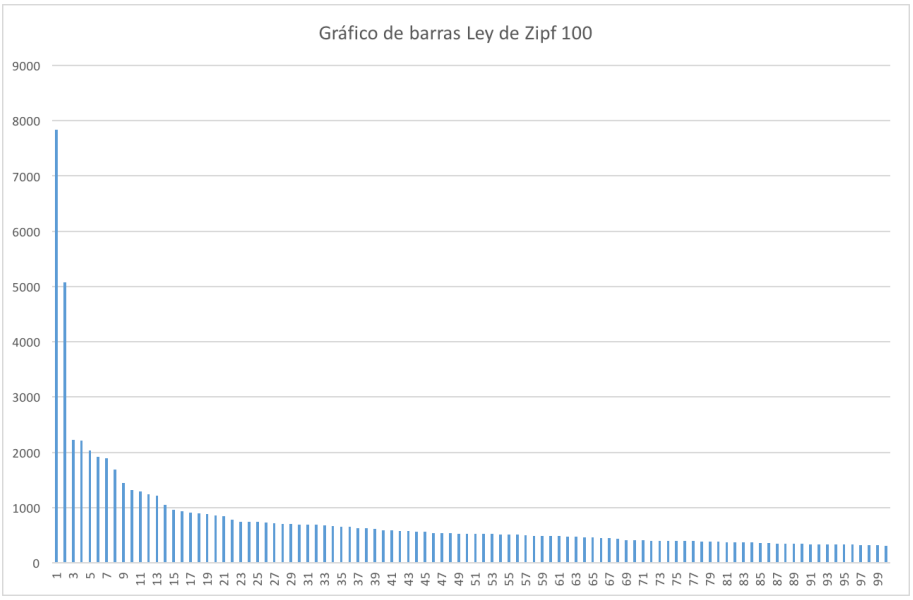

Figura 2: Ley de Zipf - 100 primeras palabras

Vemos ahora el gráfico log-log (Figura 3) con el ajuste lineal realizado sobre el gráfico, del cual obtenemos las constantes.

De esta gráfica podemos deducir las constantes de la Ley de Zifp: el valor de m es el de la pendiente de la curva, es decir, *m = -1'3905*, y el valor de k es deshaciendo el cambio logarítmico del término independiente, esto es, *k = 10^5'5063 = 320848'4902*.

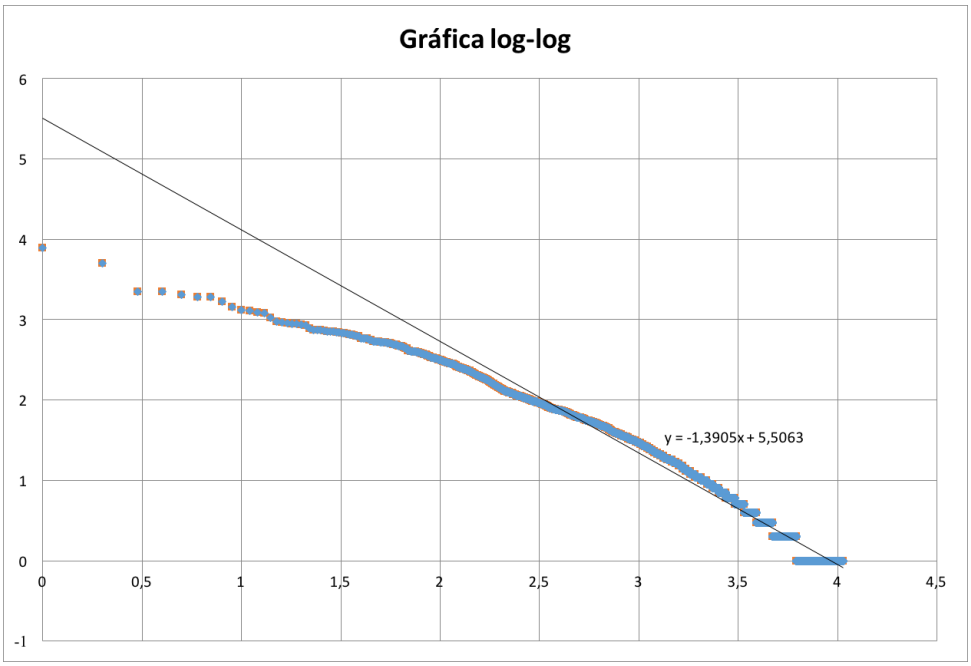

Figura 3: Ley de Zipf - Escala logarítmica con recta de regresión

4. Trabajo en grupo

En esta práctica hemos intentado que el trabajo de cada uno de los componentes fuera similar en cuanto a carga. El número de tareas a realizar no era numeroso, por lo que el reparto no ha sido demasiado grande, y consideramos que cada uno de los componente ha realizado un trabajo equiparable al de los demás. A continuación detallamos con más profundidad la aportación de cada miembro:

- Lothar Soto: Ha sido el encargado de la creación de los métodos de lectura de archivos y de la mayor parte del tratamiento de datos, como la eliminación de los signos de puntuación, además del ajuste lineal del gráfico log-log.
- Iván Calle: Se ha encargado de los métodos de lectura recursiva de un directorio, además de la implementación del proceso de lexificación.
- Daniel García: Ha llevado a cabo la generación de los gráficos para la Ley de Zipf, además de recopilar los datos obtenidos en el programa creando el código correspondiente, y volcarlos a los ficheros de datos.
- José Carlos Entrena: Encargado de la redacción del documento de entrega, además de revisar el código y hacer unas implementaciones mínimas.

Cabe destacar que la planificación del desarrollo de la práctica, junto a la discusión del mejor método para la implementación, fue realizada por todos los miembros del grupo en común, y solo después pasamos a llevar a cabo lo previamente discutido.