Proyecto RI

RECUPERACIÓN DE LA INFORMACIÓN

Autores:

Isabel Becerra Losada

Francisco Javier Molina Rojas

1. Módulos

Nuestro proyecto se divide en dos módulos principales, cada uno de estos en su archivo correspondiente:

CargaArchivos.java

Es un fichero ejecutable, que tiene como objetivo realizar las siguientes tareas:

1. Procesamiento de ficheros: se toman los ficheros que se hayan almacenado en el directorio "/temp", y a estos se les realiza una serie de operaciones con el fin de obtener la máxima información de cada uno de ellos. Destacamos de estas operaciones la eliminación de signos de puntuación, palabras con longitud = 1 y la realización de Stemming (reducir la palabra eliminando prefijos y sufijos). Este último paso, a requerido de la utilización de la librería "opennlp", y en específico de la clase PotterStemmer.
2. Creación de índice invertido: añadiremos de forma única cada palabra de los documentos a un índice invertido, en el que se almacenará el valor IDF de dicho término, junto al TF de cada documento. La construcción se llevará a cabo utilizando la estructura de datos hash-map. En adicción, se almacenará un JSON que contenga toda la información del índice invertido. Esto lo hacemos para no tener que ejecutar el programa cada vez que tengamos que realizar una consulta.
3. Guardado de ficheros procesados: los ficheros (junto a su texto) procesados, se almacenarán en "/temp1" con la visión de poder mejorar el sistema añadiendo un control sobre versiones de los propios documentos. La idea sería solo actualizar los valores de los ficheros o índices que sean modificados.
4. Normalización de vectores documentos: para ayudar a la búsqueda en el índice, se almacenará un JSON con los valores de los documentos normalizados. Para ello, se usará la estructura de datos hash-map.

Para más información relativa a la implementación de las tareas ya descritas, recomendamos la lectura del fichero "CargaArchivos.java" o del JavaDoc proporcionado. Para este último, abre el directorio "javaDoc" y haz abre el archivo "index.html".

Main.java

Es un fichero ejecutable, que tiene como objetivo realizar las siguientes tareas:

1. Carga del índice y de los documentos normalizados: antes de nada, y usando la librería GSON, leemos y cargamos en hash-maps, los JSONs generados en el módulo anterior, para poder tener el índice cargado en memoria.
2. Búsqueda de un término: en esta operación se le pedirá a l usuario que ingrese un término que desee buscar, y a su vez, un número límite de documentos a mostrar. Para realizar la búsqueda, primero preprocesaremos la consulta (similar a la tarea 1 del módulo anterior). Posteriormente, obtendremos el vector normal de la consulta, para ello y sabiendo que solo tenemos un término, su valor será el siguiente $|\vec{q}| = \sqrt{(tf * IDF)^2}$ como $tf = 1$, debido a que el termino solo aparece una vez en la consulta, entonces $|\vec{q}| = \sqrt{IDF^2} = IDF$. Con la consulta normalizada, ahora solo queda aplicar la fórmula para obtener los cosenos (la puntuación) de cada documento para dicha consulta. Una vez obtenido el coseno de cada documento, se ordena de mayor a menor y se muestra el nombre del documento, su puntuación (coseno) y su contenido.

3. Búsqueda AND/OR: esta tarea sigue un proceso similar a la anterior en cuanto a los cálculos y obtención de los documentos (teniendo en cuenta que ahora puede haber más de un término). Si seleccionamos OR, se mostrarán de mayor a menor puntuación, los documentos que contengan al menos uno de los términos introducidos. Si seleccionamos AND, se eliminarán aquellos documentos que no contengan todos y cada uno de los términos de la consulta, mostrándose así los documentos que contienen todos los términos.
4. Salida del sistema: permite al usuario terminar la ejecución del programa.

Para más información relativa a la implementación de las tareas ya descritas, recomendamos la lectura del fichero "Main.java" o del JavaDoc proporcionado. Para este último, abre el directorio "javaDoc" y haz abre el archivo "index.html".

2. Recomendaciones

- Uso de IDE: se recomienda importar el proyecto en un IDE para Java, como puede ser Eclipse o InteliJ.
- Añadir librerías al proyecto: Cuando crees el proyecto, es posible que no estén añadidas las librerías necesarias para el funcionamiento de este (estas son GSON y opennlp). Si esto ocurre, se encuentran en el directorio "/lib" para que puedan ser añadidas (debes añadir el archivo .jar correspondiente a GSON y para la librería de stemming en opennlp, añada el archivo contenido en la siguiente ruta: "/lib/apache-opennlp-2.3.1/lib/ opennlp-tools-2.3.1.jar"
- Adición de documentos en "/temp": se recomienda no añadir una cantidad excesiva de documentos a procesar. Con 200 o 300 podrás tener un resultado optimo y ver el funcionamiento de la herramienta.
- Ejecución en un orden lógico: se debe tener un índice para poder realizar la búsqueda, por lo que, ejecute el primer modulo antes que el segundo.

3. Bibliografía

- *Gson*. (2024). [Java]. Google. https://github.com/google/gson (Obra original publicada en 2015)
- *PorterStemmer (Apache OpenNLP Tools 2.3.1 API)*. (s. f.).https://opennlp.apache.org/docs/2.3.1/apidocs/opennlp-tools/opennlp/tools/stemmer/PorterStemmer.html
- Diapositivas de clase