Guía de Clases Prácticas
Modelos de Recuperación de Información (y evaluación)
(Fecha entrega: 08/05/2020)

Ejericios 1, 2 ,3 y 4

Construcción de una Matriz Término Documento:

Dada la siguiente colección:

d1 = {casa naranja casa}
d2 = {mesa puerta casa casa}
d3 = {manzana naranja manzana}

La matriz término-documento se construye ubicando como columnas todos los documentos de la colección y como fila todos los términos (el vocabulario de la colección) :

	casa	naranja	mesa	manzana	puerta
D1					
D2					
D3					

Luego, cada celda de la matriz contiene el peso (wij) que se le asigne a cada término en cada documento. El valor de wij depende del tipo de ponderación que se usemos.

Para ver las diferentes ponderaciones y cálculos de similitud recomendamos seguir los ejemplos del libro “Introducción a la Recuperación de Información”

(http://www.tyr.unlu.edu.ar/tallerIR/2008/docs/Introduccion-RI-v9f.pdf)

Para la asignación de pesos y recuperación utilizando una matriz ver:

3.4.5 – Asignación de pesos
4.1.1 – El Modelo Booleano
4.1.3 – Modelo Vectorial
4.3 – Ejemplo completo de aplicación del modelo vectorial

Ejercicios 5 y 6

Procesamiento de la colección “wiki-small”

Al procesar la colección wiki-small hay que tener en cuenta que está organizada en subdirectorios dentro de otros directorios!. Podemos usar la librería *os.walk* que nos permite recorrer recursivamente todos los niveles de subdirectorios hasta encontrar un archivo.

Tambíen es importante utilizar el parámetro *errors=”ignore”* cuándo abrimos cada archivo con la función *open*.

Esta función utiliza un tipo de codificación cada vez que abre un archivo. Por defecto es *utf-8*.

Usar el parámetro *errors=”ignore”* hará que se ignoren los errores que puedan producirse si algún documento utiliza otra codificación que no sea utf8, y por lo tanto el programa podrá seguir adelante.

```
import os
from bs4 import BeautifulSoup

my_coleccion = "" #ruta a la coleccion wiki-small
for currentpath, folders, files in os.walk(my_coleccion):
   for file in files:
       path = os.path.join(currentpath, file)
       lines = open(path, errors='ignore').read() #es importante usar errors='ignore'

       soup = BeautifulSoup(lines,  'html.parser')
       text = soup.text.lower() #extraer el texto del html
       text = text.strip()
       …….
```

http://python-notes.curiousefficiency.org/en/latest/python3/text_file_processing.html
https://docs.python.org/3/library/codecs.html

Anexo - Terrier

Para resolver los ejercicios 5 y 9 es necesario utilizar el motor de búsqueda Terrier (v5.2). Pueden encontrar la documentación oficial del mismo en este enlace.

El software está escrito en Java, por lo que el único requisito que necesitan para ejecutarlo es tener el JRE 1.8.0 (o mayor) instalado en su máquina. Si revisan en la wiki las secciones *"Download Terrier"*, *"Step by step Unix Installation"* y *"Using Terrier"* van a poder fácilmente tener todo listo para empezar a utilizarlo.

En resumen, las acciones que van a tener que realizar con el software son:

1. Configuración inicial
2. Indexación de un corpus
3. Recuperación

4. Evaluación

1. Configuración de Terrier

Terrier se puede configurar desde los archivos de configuración que se encuentran en el directorio `etc/`, o haciendo un override de las propiedades que aparecen allí directamente desde la línea de comandos. Configurar Terrier significa decirle en dónde se encuentran los archivos de nuestro corpus, que tipo de colección vamos a utilizar (archivos en texto plano, TREC, XML's, etc), especificar el formato de los queries, si usamos o no stopwords, que modelo de recuperación vamos a utilizar (Booleano, TF-IDF, etc), entre otras cosas. Revisen la wiki de configuración aquí.

En este punto, lo primero que tienen que hacer es especificarle a Terrier cuál es el path al directorio en el que se encuentran los documentos de nuestra colección:

```
>> cd terrier-project-5.2
>> bin/trec_setup.sh <absolute-path-to-collection-files>
```

2. Indexación

Una vez configurado el path a nuestro corpus, ya estamos listos para indexarlo. Para eso, se debe utilizar el comando `batchindexing`:

```
>> bin/terrier batchindexing
```

Terrier procederá a indexar los archivos que se encuentran en el directorio especificado en el primer paso, utilizando los parámetros de configuración que se encuentran en los archivos dentro de `etc/`. Si queremos hacer algún override de alguno de estos parámetros, lo podemos hacer desde la

línea de comandos. Por ejemplo para indexar la wiki-small (que se trata de una SimpleFileCollection):

```
>> bin/terrier batchindexing -Dtrec.collection.class=SimpleFileCollection
```

Los índices generados se escribirán en el directorio `/var/index/`

3. Recuperación

Se puede hacer una recuperación (querying) de forma interactiva:

```
>> bin/terrier interactive
```

O en modo batch con el comando `batchretrieve`:

```
>> bin/terrier batchretrieve
```

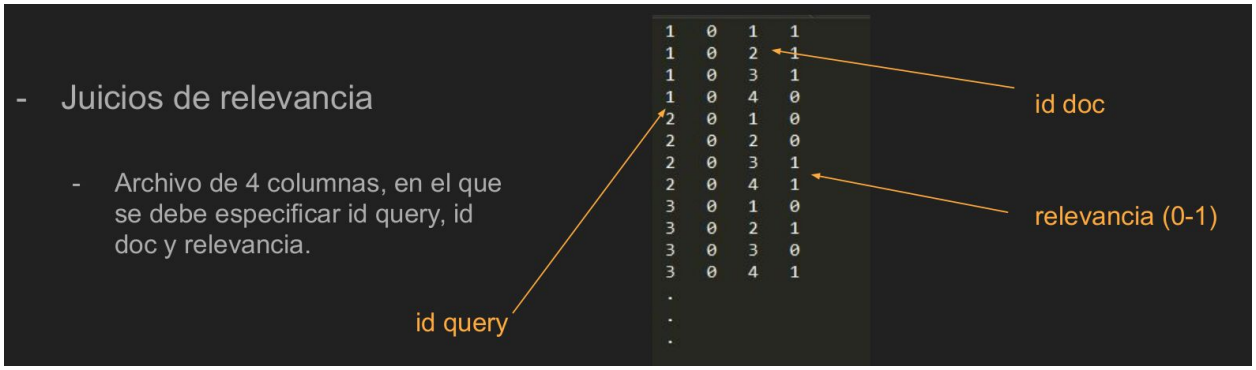
En modo batch, Terrier va a ejecutar los queries que provengan de un archivo de *topics.* Lean acá para más detalle.
Si desean cambiar el modelo de recuperación o el archivo de topics a utilizar, pueden ejecutar:

```
>> bin/terrier batchretrieve -Dtrec.model=BM25
>> bin/terrier batchretrieve -t ruta/absoluta/topics_file.txt
```

Luego de lanzar las consultas, los resultados se alojarán en el directorio `var/results/` en archivos con el formato .res, en donde van a poder encontrar las estadísticas de cada ejecución.

4. Evaluación

Finalmente, el último paso es realizar la evaluación de los resultados obtenidos. Para ello, es necesario preparar de antemano un archivo de juicios de relevancia (especificar qué documentos son relevantes para cada query):



Ejemplo de archivo de juicios de relevancia `.qrels.`
Significado:

- Para el query 1 los documentos con ID 1,2 y 3 son relevantes,
- Para el query 2 los documentos con ID 3 y 4 son relevantes,
- …

Una vez creado el archivo, procedemos a realizar la evaluación de lo que fue recuperado en el paso anterior con el comando `batchevaluate`:

```
>>bin/terrier batchevaluate -q /ruta/absoluta/qrels_file.txt
```

Los resultados de la evaluación se escribirán dentro del directorio `/var/results/` en archivos con el formato `.eval.` Allí van a poder encontrar algunas métricas de evaluación de cada ejecución, como precisión, MAP y recall.