Trabajo Práctico Nº 2

Recortes de noticias

Estructura de Datos

Integrantes: Álvarez Felipe
Buljubasic Martín
Rombolá Guido

Universidad Nacional De Tres de Febrero

Introducción

En este trabajo se recolectaron noticias de diferentes medios en formato XML a través del esquema RSS, para luego poder construir un índice invertido que permita obtener información de cantidad de apariciones y medios y secciones en donde se puede encontrar cada término. Gracias a este índice podemos, entre otras cosas, obtener respuestas a consultas booleanas y solicitar un ránking de las n palabras más mencionadas para un medio o sección específicos. Además, el programa permite contar la cantidad de noticias que se publicaron en un rango de fechas y ver las categorías más activas.

Descripción y decisiones de diseño

Se recolectaron noticias de Clarín, Télam, La Voz, Mendoza Online y El Litoral, de las secciones política, economía, últimas noticias, mundo y sociedad. La muestra de noticias que se utiliza actualmente supera las 5000.

El proyecto consta principalmente de cuatro clases:

- GetterDeNoticias: Se encarga de administrar todo lo relacionado con la recolección de noticias (crear los archivos xml en disco, actualizar el contenido, programar la tarea de actualización y etiquetar los archivos).
- GestorDeConsultas: Posee las funcionalidades principales que el usuario utiliza: ranking de palabras, categorías más activas, filtrar noticias por fecha, consulta booleana, etc.
- InvertedIndex: A partir de las noticias almacenadas, es capaz de generar el índice invertido.
- Consola: Controla la interacción con el usuario.

Se utilizó el método de compresión de índices block storage. Cada bloque almacena cuatro palabras. Éste método de compresión se compone de dos elementos, por un lado, un string que almacena todas las palabras antecedidas por su longitud, y por otro, una estructura auxiliar que es un diccionario que tiene como clave la posición del string donde comienza un bloque, y como valor una lista de identificadores únicos para cada palabra que constan de un dígito para indicar el medio a donde pertenece, otro dígito para indicar la sección, y cuatro dígitos para indicar el número de noticia (cada XML tiene sus respectivas noticias enumeradas, comenzando la más antigua por 1).

Los XML de las noticias sin comprimir ocupaban 3,7 MB. Luego de haber realizado la compresión por block storage, el string de palabras y la estructura auxiliar de todos los títulos y descripciones ocupa 931 KB, aproximadamente un 25% del

original. Si bien los XML originales contiene más elementos, como las etiquetas adicionales en cada noticia y las que identifican cada medio, la reducción es considerable.

Para construir este índice, se utilizó el esquema MapReduce:

- **Map:** genera una lista de pares intermedios (token, docId) para cada noticia, en donde el token es la palabra lematizada, y el docID es el identificador único del que hablamos previamente.
- **Reduce:** combina los resultados intermedios para generar la estructuras finales.

Para poder interactuar con el software, se deberá contar con las siguientes bibliotecas:

- nltk: se utiliza para lematizar las palabras.
- lxml: permite almacenar, recorrer, editar y realizar consultas xpath de los xml.
- schedule: facilita programar tareas de actualización de noticias en python.

Para mayor velocidad, el índice comprimido y la estructura auxiliar a los que se acceden desde las consultas por consola se encuentran serializados utilizando Pickle.

Conclusiones

Durante la construcción de este trabajo se han puesto en práctica muchos de los temas vistos durante la cursada de Estructura de Datos, tales como: manejo de archivos, recorrido de XML y consultas xpath y generación de índices invertidos y comprimidos, por lo que consideramos que es un buen ejercicio de integración, además de que nos ha permitido ahondar más en el uso de bibliotecas de Python que nos facilitan ciertas tareas.