Trabajo Práctico
Estructuras de Datos para RI

Feha de Entrega: 01/10/2010

Bibliografía: [MIR] Capítulo 8, [SE] Capítulo 5, [MAN] Capítulo 4

1) Escriba un programa que tome un conjunto de documentos de un directorio, extraiga los términos y arme los índices que permitan soportar búsquedas mediante el modelo booleano. Utilice una lista de posteo sobre un archivo secuencial. (puede utilizar la librería Perl Tokenize). Luego, codifique un segundo programa que permita buscar por uno o dos términos.

2) Utilizando el programa anterior ejecute corridas con diferentes colecciones. Calcule los tamaños mínimos, máximos y promedio de las listas de posteo. ¿Qué utilidad tiene esta información? Calcule la relación de overhead de los índices respecto de la colección. Calcule el overhead para cada documento. Luego, determine mínimos, máximos y promedio. ¿Qué conclusiones se pueden extraer?

3) Agregue documentos a una colección (indexación incremental) y repita el ejercicio 2. Sus resultados: ¿Son consistentes con la ley de Heaps?

4) La indexación incremental sobre archivos invertidos es una operación costosa. ¿Por qué? ¿Cómo se puede realizar eficientemente?

5) Modifique el programa del ejercicio 1 para armar un archivo invertido posicional a nivel de palabra. Luego, implemente consultas con operadores de proximidad.

6) Escriba un programa que extraiga los términos (no elimine las palabras vacías) y arme el índice invertido. Luego, para el texto del Quijote, calcule la distribución de frecuencias por palabra y evalué – según Zipf y Luhn – los umbrales de corte y el conjunto de términos indexables. Con esta nueva información filtre el vocabulario y arme nuevamente el índice. ¿A qué tamaño y en qué proporción se redujo? Repita el ejercicio con una colección de textos científicos. ¿Qué diferencias encuentra? Justifique. Repita el ejercicio para textos en inglés.

7) Modifique el programa del ejercicio 1 para armar un archivo invertido con información de frecuencias. Luego, implemente consultas utilizando el modelo vectorial utilizando tres esquemas de ponderación y/o ranking diferentes.

8) Modifique su programa anterior para que realice indexación posicional y soporte búsquedas booleanas por frases.

9) Ejecute una consulta en un motor de búsqueda web como – por ejemplo – Google. Recupere solo las 20 primeras páginas HTML del resultado que posean texto (no solamente objetos incrustados como Flash). Escriba un programa que procese dichas páginas, elimine los tags HTML, extraiga el texto y arme un índice invertido con información de frecuencias. Utilice su programa del ejercicio 7 y ejecute la misma consulta y compare los rankings.