



Estructuras de Datos

Fecha entrega: 31/05/2019

Bibliografía sugerida: MIR [1] Capítulos 8, Croft [2] Capítulo 5, MAN [3] Capítulos 4.

1. Codifique un script que procese documentos de un directorio y arme los índices que permitan soportar búsquedas booleanas. Grafique la distribución de tamaños de las *posting lists*. Calcule el *overhead* de su índice respecto de la colección. Calcule el overhead para cada documento. ¿Qué conclusiones se pueden extraer?
2. Codifique un script que empleando la estrategia TAAT sobre el índice creado en el ejercicio 1 y operaciones sobre conjuntos permita buscar por dos o tres términos utilizando los operadores *AND*, *OR* y *NOT*.
3. Utilizando el código anterior ejecute corridas con la colección Wiki-Small¹ y el siguiente subset de queries² y mida el tiempo de ejecución en cada caso. Para ello, utilice los siguientes patrones booleanos:
 - Queries $|q| = 2$
 - t_1 AND t_2
 - t_1 OR t_2
 - t_1 NOT t_2
 - Queries $|q| = 3$
 - t_1 AND t_2 AND t_3
 - $(t_1$ OR $t_2)$ NOT t_3
 - $(t_1$ AND $t_2)$ OR t_3

¿Puede relacionar los tiempos de ejecución con los tamaños de las listas? (pruebe con el índice en disco o cargándolo completamente en memoria antes). ¿Qué conclusiones se pueden extraer?

4. Asumiendo que tiene una cantidad limitada de memoria codifique un script que indexe una colección suficientemente grande y que requiera el volcado parcial a disco.
5. Modifique el script del ejercicio 1 para armar un archivo invertido posicional a nivel de palabra. Luego, implemente consultas con operadores de proximidad y búsquedas booleanas por frases.
6. Modifique el script 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.
7. Agregue *skip lists* a su índice del ejercicio 1 y ejecute un conjunto de consultas *AND* sobre el índice original y luego usando los punteros. Compare los tiempos de ejecución con los del ejercicio 2.
8. A partir de un conjunto de *posting lists* provistas³ realice un programa que arme el vocabulario utilizando un B+Tree por un lado y un archivo binario con la información de las postings y frecuencias por el otro. Use DGaps y agregue *skip-lists*.

¹<http://dg3rtljvitrle.cloudfront.net/wiki-small.tar.gz>

²http://www.labredes.unlu.edu.ar/sites/www.labredes.unlu.edu.ar/files/site/data/ri/queries_2y3t.txt

³<http://www.tyr.unlu.edu.ar/tallerIR/2014/data/dump10k.tar.gz>



9. A partir del archivo de palabras en inglés (words-en.txt) calcule el tamaño necesario para almacenarlo en memoria sin comprimir o como dictionary-as-string. Haga un notebook y calcule la distribución de longitudes de las palabras y las estadísticas básicas. Conviene tomar un valor máximo de palabra? Justifique.
10. Sobre la colección Dump10k escriba un programa que realice una evaluación TAAT y otro usando DATT. Compare los tiempos de ejecución para un conjunto de *queries* dados⁴. Separe su análisis por longitud de *queries* y de *posting lists*.
11. A partir de la colección Wiki-Small construya el índice invertido con información de frecuencias y comprímalo utilizando Elias-Gamma y Variable-Length Codes. Calcule tiempos de compresión/descompresión y tamaño resultante en cada caso. Realice dos experimentos, uno codificando con DGaps y otro sin codificar. Compare los tamaños de los índices resultantes.

Referencias

- [1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison-Wesley Publishing Company, USA, 2nd edition, 2008.
- [2] Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, USA, 1st edition, 2009.
- [3] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

⁴<http://www.labredes.unlu.edu.ar/sites/www.labredes.unlu.edu.ar/files/site/data/ri/queriesDump10K.txt.tar.gz>