Recuperación de la Información
Jordy Joaquin Cuan Robledo [1], Eduardo Cantorán Flores [1],
Rafael Pérez Aguirre [1]
[1] Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, México

Recuperación de Información

Lógica booleana

Es una lógica que se emplea (ya casi no, se sigue empleando aunque sólo para consultas muy avanzadas) al momento de buscar información dentro de los textos a pesar de ya ir en desuso dado las nuevas técnicas de recuperación de información como indexados.

Grep

Grep es un programa de Unix que se ejecuta desde la línea de comandos el cual nos permite introducir una línea o expresión regular, leer una lista de archivos e imprimir las líneas que hayan coincidido con nuestra búsqueda.

Técnica de matrices de incidencia

Las matrices de incidencia son tablas las cuales relacionan un documento con una palabra, y cada celda toma el valor de un cero o un uno, con esto sabemos si una palabra se encuentra presente en el archivo y nos permite realizar la consulta de lógica booleana con una mayor velocidad, el único inconveniente es que para cada palabra que no existe dentro del documento, se emplea un entero en memoria y produce un mal gaste de la memoria.

	Presentación.txt	**Contrato de trabajador.txt**	**Carta navideña.txt**
Nombre	1	1	1
Navidad	0	0	1
Empleado	0	1	0

¿Qué tan buenos son los documentos devueltos?
Hay dos maneras de medir la efectividad de un buscador:

- Precisión: Fracción de los documentos recibidos que tienen información relevante al usuario
- Recall: Fracción de los documentos relevantes en la colección que son devueltos

Las dos mediciones no suelen ser muy efectivas en cierto modo, así que podemos usar una más equilibrada que media entre ambas soluciones anteriores:

$$F_{\text{measure}} = \frac{0.5 \times P \times R}{P + R}$$

Índices Invertidos

En esta técnica, se tiene que decir en qué documento se encuentra cada término, para ello se define un diccionario el cual contiene todas las palabras encontradas dentro de cada documento y cada palabra tiene asociada una lista donde están almacenadas las referencias de cada documento donde haya aparecido esta palabra. A cada referencia de documento donde se haya dicha palabra se le llama posting, y a al arreglo donde están todos los documentos enlistados se le llama postings.

Metodología para Indexar documentos:

Para llevar a cabo la indexación de documentos, se llevan a cabo los siguientes pasos

1. En primer lugar tenemos que agrupar los documentos que procesaremos con el algoritmo.
2. Tokenizar el texto de los documentos, preprocesar el contenido.
3. Normalizar los resultados arrojados, como por ejemplo, si nos llegamos a encontrar la palabra ‘amigos’, almacenarla como ‘amigo’, con esto evitaremos tener que almacenar elementos con el mismo lemma.
4. Indexar, hacer el diccionario de listas. Para este paso, se extraen todas las palabras de cada documento, si la palabra aún no se encuentra dentro del diccionario, se añade como llave de nuestra tabla hash y como valor se le asigna una lista conteniendo un elemento que es el archivo donde se encontró. Si la palabra ya se encuentra dentro del diccionario, se toma el documento donde se encontró y este es aprendizado a la lista de dicha palabra. Así hasta haber recorrido todos los documentos.

Para hacer una consulta donde aparezca una palabra, lo único que se debe hacer es acceder al diccionario (tabla hash) y traer de vuelta la lista (postings), con ello sabremos que en esos documentos contienen la petición. Si se desean manejar consultas de dos o más elementos se debe de traer todos los postings de cada palabra y aplicarles un poco de teoría de conjuntos, ya sea intersección (and), union (or), resta, or exclusivo, etc según se desee.

Prácticas realizadas

Indexado de documentos locales

En este programa se aplican los puntos descritos en el algoritmo de la sección anterior. La implementación está hecha en python 2.7 bajo el sistema operativo de Mac OS X. Es un pequeño script que indexa cierto número de bibliografías recorriendo un árbol de direcciones

dentro de una dirección dada de la cual se extraen todos los nombres de los ficheros en una lista que se emplea dentro del ciclo que indexa estos documentos para conseguir el diccionario. Al finalizar, se pide al usuario introducir una consulta, si la consulta tiene una longitud de un elemento entonces se imprimen los postings, si tiene una longitud mayor, se procesan todos los postings de los elementos con un poco de lógica de conjuntos.

Indexado de la BUAP

Este trabajo era completamente opcional pero se implementó con el fin de reafirmar los conocimientos obtenidos. Consiste en un programa que se divide en dos etapas. La primer fase es un algoritmo de web scraping aplicado a la buap, esto quiere decir que solo se visitaran aquellos enlaces donde en su dominio aparezca la substring 'buap', de no ser así, simplemente se ignora el link y no se añade a la lista de urls. Para realizar el parsing de Html se utiliza una librería de python llamada BeautifulSoup que nos ayudará a procesar todo el contenido y sus tags. Al momento de visitar la página de la buap, se guarda la url como llave de un diccionario y se procede a limpiar lo que será el valor de esta llave, en este caso es todo el body/contenido devuelto por BeautifulSoup dentro de la página. Dentro de la limpieza se eliminan las etiquetas de bloques como *script* y *noscript*, a parte, se quitan los tags de formato mediante el uso de una expresión regular.

En la segunda fase del programa, se crea el diccionario indexado a partir de lo que se tiene en la tabla hash de urls con bodys. El resto del programa es completamente análogo a lo presentado en el programa que indexa los documentos locales.