Proyecto 1: Buscador de Documentos Basado en Índices Invertidos

Objetivo

Implementar un sistema básico de búsqueda de términos en documentos, utilizando un índice invertido construido con listas enlazadas. Este proyecto busca aplicar los conocimientos de estructuras de datos vistas hasta ahora para representar de manera eficiente la relación entre palabras y documentos.

Descripción del problema

Para realizar una búsqueda sobre una colección de documentos (archivos de texto, documentos PDF, páginas HTML, tweets, etc) se pueden crear índices, con el fin de acelerar el proceso. Una forma de hacerlo es mediante una estructura de datos conocida como Índice Invertido, estructura central en los sistemas de búsqueda de texto. Permite localizar de manera eficiente los documentos que contienen un término (palabra) dado. Cada documento puede ser una página web, archivo pdf, archivo word, etc.

La Figura 1 muestra un ejemplo de índice invertido. En la parte superior se presenta una colección de tres documentos cada uno con una lista de palabras. En la parte inferior se construye un índice invertido: para cada término que aparece en la colección, se mantiene una lista de posteo (implementada como lista enlazada) que almacena los identificadores de los documentos donde el término aparece. Este índice debe ser representado mediante listas enlazadas, tanto para las palabras como para las listas de documentos asociadas.

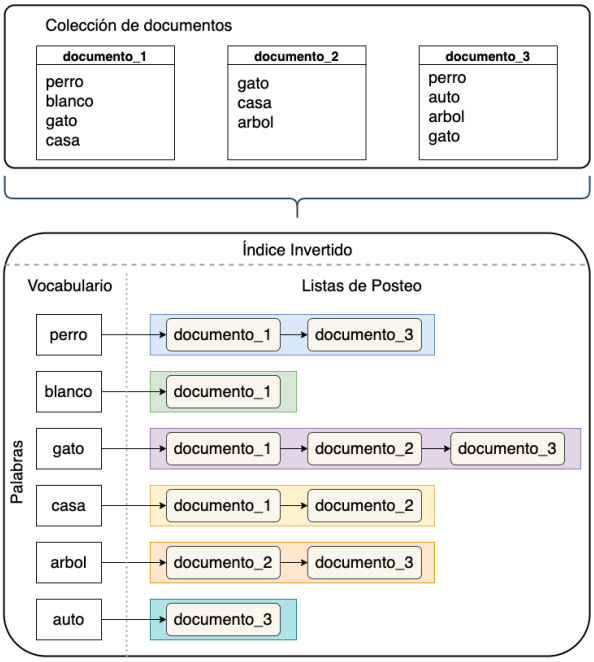

Figura 1: Ejemplo de búsqueda con tres términos.

Su tarea será replicar este comportamiento: dado un conjunto de documentos, deberás construir el índice invertido correspondiente usando listas enlazadas implementadas por ti mismo(a) en C o C++.

Es habitual que en los documentos de la colección se encuentre un conjunto de palabras muy comunes y que aparecen en casi todos los documentos, a estas palabras se las denomina *stopwords* o palabras basura (normalmente preposiciones, artículos) que son descartadas del índice debido a que no aportan a discriminar documentos en una búsqueda. En ocasiones, las palabras ofensivas también son tratadas como *stopwords* y se remueven del índice.

Con el índice invertido construido, resulta muy fácil y rápido efectuar una búsqueda:

- Primero, se obtienen las listas de posteo para cada término (o palabra) de la consulta.
- Luego, se realiza una intersección de las listas de posteo, obteniendo aquellos documentos que aparecen todas las listas, tal como se observa en Figura 2 y Figura 3.

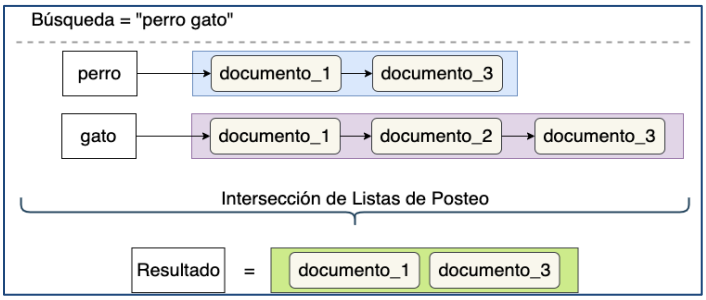

Figura 2: Ejemplo de búsqueda con dos términos.

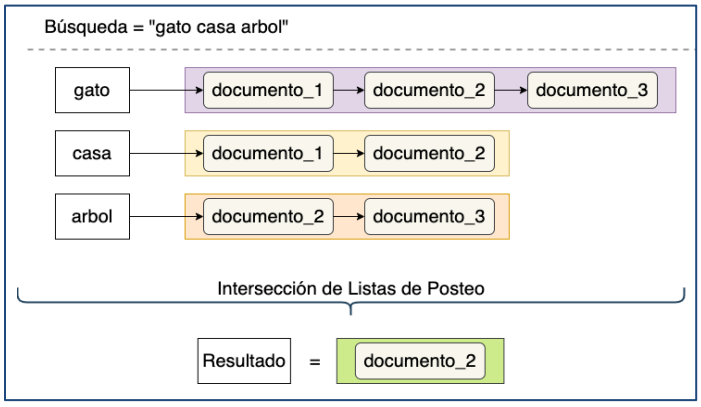

Figura 3: Ejemplo de búsqueda con tres términos.

Importante:

Queda estrictamente prohibido hacer uso de LLMs (como chatGPT, DeepSeek o cualquier otro) para la generación de código de su proyecto. Su uso será sancionado con nota mínima y acciones reglamentarias.

Requisitos funcionales mínimos

1. Procesar un conjunto de archivos de texto para construir el índice invertido.
2. Eliminar *stop words* al momento de construir el índice y de las consultas que haga el usuario (se proporcionará un listado).
3. Representar el índice usando listas enlazadas (por ejemplo, una lista de palabras, cada una con una sublista de documentos donde aparece).
4. Permitir al usuario consultar una o varias palabra(s) y obtener la lista de documentos en los que aparecen., Si una consulta de usuario considera varias palabras, deberá buscar en los índices correspondientes y, como resultado, entregar la intersección de los documentos de las listas.
5. Implementar funciones básicas como:
    a. Creación del índice.
    b. Inserción de documentos sin repetición.
    c. Búsqueda de términos.

Requisitos funcionales Optativos

1. Mostrar la frecuencia de aparición de cada término en cada documento (5 Pts extra).
2. Comprimir mediante incrementos los ID de los documentos en la lista de posteo (20 Pts extra).

Restricciones

- Debe usarse lenguaje C o C++.
- No se permite el uso de estructuras de datos avanzadas no vistas en clase (como árboles, hash maps, etc.) para las listas de posteo, deben ser implementadas por Ud.
- Todo el manejo del índice debe realizarse en memoria utilizando listas enlazadas implementadas por los estudiantes.
- Sin embargo, para la construcción de la tabla “Vocabulario”, se permite el uso de *Maps*.

Entrega

- Fecha de entrega: Martes 6 o Miércoles 7, según corresponda a su sección.

Rúbrica de Evaluación

1. Correctitud y funcionamiento (45 puntos):
    a. Procesamiento correcto de documentos: 10 pts
    b. Eliminación de palabras vacías (stop words): 5 pts
    c. Construcción del índice invertido: 10 pts
    d. Búsqueda de términos individuaes: 10 pts
    e. Búsqueda de múltiples términos (realizar intersección de resultados parciales): 10 pts.
2. Uso de estructuras de datos (20 puntos):

a. Definición de estructuras de datos adecuadas: 10 pts
b. Implementación de listas enlazadas en sublistas: 10 pts

3. Organización y calidad del código (15 puntos):
   a. Modularización y claridad: 5 pts
   b. Comentarios y nombres descriptivos: 5 pts
   c. Uso correcto de memoria dinámica: 5 pts
4. Informe y documentación (20 puntos):
   a. Descripción de estructuras: 7 pts
   b. Explicación de funciones clave: 7 pts
   c. Ejemplos de ejecución: 6 pts
5. *Elementos Optativos (25 Pts adicionales).*

Ejemplo de uso del sistema de búsqueda

Documentos de entrada:

Documento 1:, La simulación es una herramienta poderosa para estudiar sistemas complejos

Documento 2: Los modelos de simulación permiten representar el comportamiento de un sistema real

Documento 3: Un sistema puede analizarse mediante simulación o mediante observación directa

Stop Words:

Stop words: *la, los, de, un, una, es, para, o, el*

Índice invertido correspondiente (simplificado):

```
analizarse → [doc3]
comportamiento → [doc2]
complejos → [doc1]
directa → [doc3]
herramienta → [doc1]
mediante → [doc3]
modelos → [doc2]
observación → [doc3]
permite → [doc2]
puede → [doc3]
representar → [doc2]
simulación → [doc1, doc2, doc3]
sistema → [doc2, doc3]
sistemas → [doc1]
estudiar → [doc1]
```

Consultas y salidas esperadas:

Ejemplo 1:

Entrada: simulación
Salida: La palabra "simulación" aparece en los documentos: ***doc1, doc2, doc3***

Ejemplo 2:

Entrada: sistema simulación
Salida: Los documentos que contienen todas las palabras son: ***doc2, doc3***

Conjunto de Datos:

Para el desarrollo del proyecto, considere la colección de datos que se detalla a continuación. El archivo `gov2_pages.dat` es un subconjunto de documentos de la colección GOV2 de documentos de la Web[1].,

Cada línea del archivo corresponde al contenido de una página web. Cada línea contiene los elementos separados por los caracteres "||" (doble barra vertical). Después de la última ocurrencia de "||" vienen los términos del contenido de cada página, y lo que lo antecede corresponde a su URL.

Por ejemplo, la primera línea del archivo (correspondiente a la URL) contiene:

```
http|| sgra|| jpl|| nasa|| gov|| JPL Sgra web Site JPL Sgra Web Site The US Space VLBI project web site has moved Click HERE to go to the new US Space VLBI project web site The following web pages are now available from this site Old US space VLBI project web site Project science web page Last updated Thu Sep 16 17 24 48 PDT 1999
```

Donde:

- La URL de la página corresponde a:
  - `http|| sgra|| jpl|| nasa|| gov||`
- Y el contenido de la página es:
  - `JPL Sgra web Site JPL Sgra Web Site The US Space VLBI project web site has moved Click HERE to go to the new US Space VLBI project web site The following web pages are now available from this site Old US space VLBI project web site Project science web page Last updated Thu Sep 16 17 24 48 PDT 1999`

Adicionalmente, se dispone de un log de consultas reales (archivo `Log-Queries.dat`) que puede utilizar para probar su funcionamiento. Las *stop words* se encuentran en el archivo `stopwords_english.dat`

Instrucciones de Entrega:

Deberá subir tanto el informe como el código junto a todas las indicaciones que permitan su compilación y uso en Aula Virtual.

El profesor, discrecionalmente, podría citarles para realizar una revisión en persona del proyecto.

---

[1] **Advertencia**: Los archivos de datos de la colección corresponden a datos de páginas reales de la Web, por lo tanto, podrían contener palabras que algunas personas pueden considerar como ofensivas, se recomienda discreción.