UMAG Universidad de Magallanes | Facultad de Ingeniería

Diseño de Algoritmos - PR3: Algoritmos de Búsqueda de Patrones en Documentos

Diego Sanhueza[†], **Manuel González**[†] **y Claudio Matulich**[†]

[†]*Universidad de Magallanes*

13 de junio de 2025

Resumen

En este informe se presenta el desarrollo de un sistema de búsqueda de patrones en documentos, implementando algoritmos avanzados y estructuras de datos eficientes. El objetivo es analizar el rendimiento de estos algoritmos en términos de tiempo de ejecución, número de comparaciones y accesos a memoria, utilizando un conjunto de documentos preprocesados. Se implementaron varios algoritmos de búsqueda, incluyendo KMP y Boyer-Moore, y se desarrollaron estructuras de datos para indexación y recuperación de información. Además, se aplicaron técnicas de preprocesamiento de texto para mejorar la eficiencia y precisión de las búsquedas.

Índice

1. Introducción

En este informe se presenta el desarrollo de un sistema de búsqueda de patrones en documentos, implementando algoritmos avanzados y estructuras de datos eficientes. El objetivo es analizar el rendimiento de estos algoritmos en términos de tiempo de ejecución, número de comparaciones y accesos a memoria, utilizando un conjunto de documentos preprocesados.

2. Objetivos

- Implementar algoritmos avanzados de búsqueda de patrones en texto.
- Desarrollar estructuras de datos eficientes para indexación y recuperación de información.
- Analizar empíricamente la complejidad y rendimiento de los algoritmos.

- Aplicar los algoritmos a un problema práctico de procesamiento de documentos.
- Implementar técnicas de preprocesamiento de texto para mejorar la eficiencia de búsqueda.
- Documentar adecuadamente el código y los resultados del análisis.

3. Descripción del Proyecto

Explicación del problema a resolver, alcance y requisitos principales. El proyecto consiste en un sistema de búsqueda de patrones en documentos que permite a los usuarios buscar términos específicos dentro de un conjunto de archivos de texto y HTML. El sistema implementa varios algoritmos de búsqueda, estructuras de datos para indexación y técnicas de preprocesamiento de texto para mejorar la eficiencia y precisión de las búsquedas.

4. Requisitos Técnicos

Como requisitos técnicos, el sistema debe ser capaz de manejar archivos de texto y HTML, realizar búsquedas exactas y aproximadas, y proporcionar estadísticas sobre los documentos procesados. Además, debe contar con una interfaz de línea de comandos para facilitar su uso.

5. Diseño del Sistema

El sistema se divide en varias secciones clave:

5.1. Arquitectura del Sistema

Descripción general de la arquitectura del sistema, incluyendo módulos principales y su interacción.

5.2. Módulos Principales

- **Carga y Preprocesamiento de Documentos:** Módulo encargado de leer los archivos, extraer texto y metadatos, y aplicar técnicas de preprocesamiento.
- **Indexación de Documentos:** Módulo que construye y actualiza índices para facilitar búsquedas rápidas.
- **Motor de Búsqueda:** Implementa los algoritmos de búsqueda de patrones y maneja las consultas del usuario.
- **Análisis de Texto:** Proporciona estadísticas sobre los documentos, como palabras clave y similitud entre documentos.
- **Interfaz de Línea de Comandos:** Permite a los usuarios interactuar con el sistema a través de comandos específicos.

5.3. Algoritmos de Búsqueda de Patrones

En este proyecto se implementaron tres algoritmos principales para la búsqueda de patrones en texto:

- **KMP (Knuth-Morris-Pratt):** Utiliza un arreglo de prefijos (LPS) para evitar comparaciones redundantes y mejorar la eficiencia al buscar patrones. Se registran comparaciones, accesos a memoria y tiempo de ejecución, guardando los resultados en CSV para su análisis.
- **Boyer-Moore (mal carácter):** Implementado usando la heurística del mal carácter, permite saltos eficientes en el texto y reduce el número de comparaciones. También mide el rendimiento y almacena los resultados en un archivo CSV.
- **Shift-And:** Algoritmo basado en operaciones a nivel de bits, eficiente para patrones cortos (hasta 31 caracteres). Utiliza máscaras de bits para representar el estado de coincidencia y también registra métricas de rendimiento.

5.4. Estructuras de Datos para Indexación

Para facilitar búsquedas rápidas y eficientes, se utilizaron principalmente dos estructuras:

- **Índice invertido:** Permite asociar cada palabra con las posiciones donde aparece en los documentos, facilitando búsquedas exactas y consultas booleanas.
- **Tabla hash:** Usada para calcular la frecuencia de palabras y detectar palabras clave en los textos.

5.5. Técnicas de Preprocesamiento de Texto

El preprocesamiento es fundamental para mejorar la calidad de las búsquedas. Se aplicaron las siguientes técnicas:

- **Tokenización:** Separación del texto en palabras o tokens.
- **Normalización:** Conversión a minúsculas y eliminación de acentos/caracteres especiales.
- **Eliminación de stopwords:** Se eliminan palabras comunes que no aportan significado relevante (como .^{el}", "la", "de", etc.).

5.6. Framework de Análisis de Rendimiento

Para evaluar los algoritmos, se midieron las siguientes métricas en cada búsqueda:

- **Comparaciones:** Número de comparaciones realizadas entre caracteres.
- **Accesos a memoria:** Cantidad de veces que se accede a posiciones de texto o patrones.
- **Tiempo de ejecución:** Medido en milisegundos.

Los resultados se almacenan en archivos CSV para facilitar su análisis y comparación. Se usaron textos de prueba de diferentes tamaños y patrones variados.

6. Desarrollo del Sistema

6.1. Carga y Preprocesamiento de Documentos

El sistema permite cargar tanto archivos de texto plano como HTML. Para los archivos HTML, se realiza una limpieza básica para extraer solo el contenido relevante, eliminando etiquetas y caracteres innecesarios. Una vez cargado el texto, se aplica un preprocesamiento que incluye normalización (pasar todo a minúsculas y quitar acentos), tokenización (separar el texto en palabras) y eliminación de stopwords. Esto ayuda a que las búsquedas sean más precisas y rápidas, ya que se trabaja solo con la información relevante.

6.2. Indexación de Documentos

Para acelerar las búsquedas, se construye un índice invertido para cada documento. Este índice asocia cada palabra con las posiciones donde aparece en el texto, permitiendo encontrar rápidamente todas las ocurrencias de una palabra o patrón. Además, se utiliza una tabla hash para calcular la frecuencia de palabras y detectar palabras clave. Los índices pueden guardarse en disco para no tener que reconstruirlos cada vez que se realiza una búsqueda.

6.3. Motor de Búsqueda

El motor de búsqueda implementa varias funcionalidades:

- Búsqueda exacta de patrones usando los algoritmos KMP, Boyer-Moore y Shift-And.
- Búsqueda aproximada utilizando la distancia de Levenshtein, lo que permite encontrar palabras similares aunque tengan errores de tipeo.
- Consultas booleanas, combinando varios términos de búsqueda.
- Ranking de documentos según la cantidad de apariciones del patrón buscado.

El usuario puede elegir el algoritmo y el tipo de búsqueda desde la línea de comandos.

6.4. Análisis de Texto

El sistema también permite analizar los documentos para obtener estadísticas como la frecuencia de palabras clave, la cantidad de palabras únicas y la similitud entre documentos. Para la similitud, se compara el conjunto de tokens de cada documento, lo que ayuda a identificar textos relacionados o duplicados.

6.5. Interfaz de Línea de Comandos

La interacción con el sistema se realiza completamente desde la línea de comandos. Se pueden usar diferentes comandos y flags para seleccionar el algoritmo de búsqueda, indicar el archivo o carpeta a analizar, buscar palabras o frases, comparar documentos, mostrar rankings, entre otros. Además, se incluye un comando de ayuda que explica el uso básico del sistema.

7. Pruebas y Resultados Experimentales

Presentación de los experimentos, casos de prueba, resultados, tablas y gráficos comparativos.

8. Discusión y Conclusiones

Análisis de los resultados, dificultades, posibles mejoras y conclusiones finales.

9. Referencias

-
-
-