CENTRO DE ENSEÑANZA TÉCNICA Y SUPERIOR

Profesional

Carrera: I.C.C.

3er Semestre

MATERIA: Análisis de Algoritmos

PROFESORA: Vanessa López

TÍTULO: Detector de Plagio *(Metodo Rabin-Karp)*

PRESENTA:

Ángel Ramírez - *#37487*

Carlos Gutierrez - *#32725*

Diego Varela - *#37957*

Esteban Colautti - *#37267*

Tijuana, B.C., 16 de septiembre del 2024

INTRODUCCION

Definición de Problema

Queremos llevar a cabo este proyecto con la intención de facilitar a las empresas o instituciones que necesiten identificar y reconocer el plagio, ya que es un problema común en muchos entornos, tanto laborales como académicos, siendo crucial para el buen funcionamiento y desarrollo de estas entidades. Con la creciente cantidad de contenidos digitales y la facilidad para copiar información, el plagio se ha vuelto más frecuente y representa un desafío para la integridad y reputación de las organizaciones. Por esta razón, es esencial contar con una herramienta que permita detectar de manera rápida y precisa las similitudes entre documentos, garantizando así la originalidad y el respeto por el trabajo intelectual.

Entrada esperada:

- **Documentos de texto**: El sistema recibirá como entrada uno o más documentos que serán comparados entre sí para detectar posibles plagios.
- **Fragmento de texto específico**: También se podrá buscar coincidencias de un fragmento específico de texto dentro de uno o más documentos.

Salida esperada:

- **Informe de similitud**: El sistema creará una serie de advertencias que indiquen los fragmentos del texto en los documentos comparados que coinciden entre sí.
- **Porcentaje de similitud**: Un porcentaje indicará el nivel de coincidencia entre los documentos analizados.
- **Detalle de coincidencias**: Se mostrará un fragmento de la zona que se considere plagio o coincidencia.

Comportamiento del sistema:

1. **Lectura de los documentos**: El sistema toma los documentos de entrada, los convierte a una representación interna de texto y los segmenta en fragmentos (subcadenas).
2. **Aplicación del algoritmo de Rabin-Karp**: Para cada fragmento de texto, el sistema calculará un hash. Comparará los hashes de los fragmentos de texto de los documentos para detectar coincidencias.
3. **Detección de plagio**: Si los hashes coinciden, se verificará el contenido real de los fragmentos para confirmar que no es una coincidencia accidental de los valores hash.
4. **Generación de resultados**: Si se detecta una coincidencia, se generará un reporte que incluya las secciones coincidentes, el porcentaje de similitud y las posiciones exactas de las coincidencias en ambos documentos.

Objetivos

Detectar plagio: Identificar coincidencias entre fragmentos de texto en dos o más documentos para identificar si se ha copiado algún contenido sin autorización o citas apropiadas.
Eficiencia: El sistema debe ser capaz de comparar texto de manera eficiente utilizando el algoritmo de Rabin-Karp.
Precisión: Reducir coincidencias accidentales de frases y el posible plagio no detectado.

Investigación preliminar y selección del algoritmo:

Problemas Previos

Con respecto a los problemas previos que encontramos con respecto a este problema, el principal y más detallado de todos fue el encontrado en el libro “Introduction to Algorithms”, donde en el capítulo 11 se menciona un programa con el mismo objetivo el cual utiliza el método Rabin-Karp para poder realizar esta detección.

Algoritmos similares

Plagscan es una aplicación similar a lo que queremos hacer, esta aplicación detecta tres coincidencias en palabras que sean consecutivas para de esa manera encontrar el plagio a pesar de que contenga sinónimos, ya al final del proceso se usa IA para identificar las citas, las coincidencias que no tienen nada que ver con plagio y contenido de su lista blanca para dar resultados más completos, sistema para la indexación se basa en Apache solr que es un motor de búsqueda vertical lo cual lo hace mas rapido que un motor de búsqueda general al buscar las cosas de manera más centrada.

Selección de Algoritmo

El método que se seleccionó a utilizar para este proyecto sería el de las “Hash Tables”, y más específicamente el método Rabin-Karp para la detección de plagio, esto debido a que en la investigación realizada sobre proyectos previos de este tema se vio que este método era el más comúnmente utilizado en estos programas de detección de plagio.

Además de lo anterior, una de las principales razones por la cual se seleccionó el método de Hash Tables para este proyecto fue debido a que este programa principalmente realizará búsqueda de coincidencias entre textos, cuestión que el método Hash realiza de manera más eficiente que los demás métodos disponibles para este proyecto. Esto lo realiza por medio de guardar partes del texto en llaves dentro de un índice el cual nos permitiría comparar el texto insertado en el programa con otros dentro de la base de datos del mismo y comparar estos mismo, al final ver cuantas veces se repitieron ciertas cuestiones en el mismo orden y al final entregar el porcentaje de plagio del texto insertado.

Complejidad Esperada

Con respecto a la complejidad de tiempo para el programa sería que, según lo investigado, las funciones Hash tienen una para el peor de los casos una complejidades de log(n), mientras que en el mejor sería de 1. Con respecto a la memoria, ésta sería del doble del original al realizar una copia donde se comparará los valores con la de la base de datos.

REFERENCIAS:

- Cormen, T., Leiserson, C., Stein, Clifford, Rivest, R., (2022), "*Introduction to Algorithms*", Cuarta Edición.