Implementación Del Algoritmo Rabin Karp Para Detección De Plagio En Documentos De Texto

1st Alex Helder Huancara,2st Ronaldo Quispe,3st Daniel Eduardo Sarco,4stRonaldinho Vega
Escuela profesional de Ingeniería Informatica y de Sistemas
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú
[1]helder.huancara@gmail.com,[2]ronld.qyr7@gmail.com,[3]jirenssj999@gmail.com,[4]dinho15971@gmail.com

***Abstract*—**
El plagio es una actividad que cada vez se incrementa dentro del ámbito académico, para evitar este problema se implementara una herramienta la cual nos ayudara a detectar el porcentaje de similitud entre dos documentos de texto ,donde como punto de partida tomaremos el algoritmo de Rabin Karp ,este algoritmo consiste en reconocer múltiples patrones de cadenas de texto para detectar similitudes entre ellas,para eso se utilizara múltiples estructuras de datos para la eficiencia del algoritmo.
***Index Terms*—Algoritmo Rabin Karp, Plagio, Similitud de textos.**

I. Introducción

En el mundo actual, las personas tienden a ser facilistas por los avances tecnológicos que nos proporcionan acceso a información variada en todo momento, esto sumado al miedo a reprobar determinadas materias, obtener bajas calificaciones , el escaso conocimiento de los derechos de autor y la propiedad intelectual motivan a los estudiantes a copiar o presentar de forma sustancial o total los trabajos, documentos, ideas, o las palabras de otro autor como si fueran de su propiedad. Este problema se conoce como plagio y es uno de los que más afecta a los académicos de los diferentes niveles educativos ya que disminuye la capacidad de crear e innovar un trabajo propio, por lo que es primordial garantizar la propiedad intelectual de cada autor.[1,2]

En muchas materias se les asigna la misma tarea a toda la clase y los estudiantes tienden a copiar información de Internet o de otros estudiantes. Esta tendencia del estudiante debe cambiar.Por lo que cualquier herramienta que pueda detectar dicho plagio es necesaria para solucionar este problema.

Mediante el uso de cadenas coincidentes podemos detectar el plagio en un documento de texto. Existen diversos algoritmos que se usan para la detección de plagios pero uno de los más populares y de uso frecuente es el algoritmo Rabin-karp.[3]

El algoritmo de Rabin Karp es un tipo de algoritmo de coincidencia de cadenas que se puede utilizar para varias búsquedas de patrones. Para reducir el tiempo de procesamiento y aumentar la precisión[1]. La ventaja del algoritmo Rabin-Karp en comparación con otros algoritmos de coincidencia de cadenas es la capacidad de buscar múltiples patrones de cadenas[6].

Este estudio intenta hacer coincidir cadenas de texto de documentos utilizando el algoritmo Rabin-Karp, para identificar plagios entre documentos.

II. Estado de Arte

El documento “Rabin Karp And Winnowing Algorithm For Statistics Of Text Document desarrolla una aplicación para detectar plagio de documentos de texto utilizando el algoritmo de Rabin-Karp y el algoritmo Winnowing. Estos documentos pasan por 3 fases de pre-procesamiento: eliminacion, filtrado y derivado, para después calcular el porcentaje de similitud entre 2 textos. [1]
Dita B. R. , Muhammad L. I., Rita B. Purba y Indra R. desarrollaron un sistema con el objetivo de analizar el plagio en el sistema E-learning, en el cual se aplica el algoritmo Rabin Karp basada en la web como herramienta para detectar plagio. [2].
Existe una aplicación que realiza una comparación del tiempo de procesamiento por el algoritmo de Rabin-Karp con un valor de k en diferentes k-Gram, donde se compara el valor de k-Gram y valor base principal para elegir el mejor valor de k. Los resultados experimentales muestran que no hay diferencia significativa al comparar los valores k-Gram y valor base principal. [3].
En la aplicación web desarrollada por Birnardi L. y Seng H. se determina el porcentaje de similitud entre documento de texto haciendo uso de los algoritmos de Rabin-Karp y Jaro-Winkler, donde se llegó a la conclusión que algoritmo de Rabin-Karp es más eficaz que el algoritmo de distancia de Jaro-Winkler. [4].
“Plagiarism Detection by using Karp-Rabin and String Match-

ing Algorithm Together" está centrado en la detección de plagio en las tareas prácticas (proyectos), así como en los documentos escritos que deben presentar los estudiantes en la universidad. Esta detección se hace utilizando el algoritmo Karp-Rabin and String Matching. [5]

III. METODOLOGÍA

Los proceso del sistema de verificación de plagio de documentos se muestra en la Figura 1.

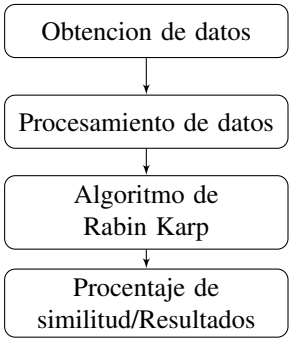

Fig. 1. Diagrama de flujo del sistema

A. Obtención de datos.

En este estudio, los datos utilizados como entrada son documentos de texto con la extensión .pdf.

B. Preprocesamiento de datos

Los usuarios ingresarán dos documentos, el documento original y el documento que se desea probar. Una vez que el sistema obtiene información de los documentos ingresados, el sistema empezara el proceso de preprocesamiento Los pasos del preprocesamiento se muestran en la Figura 2

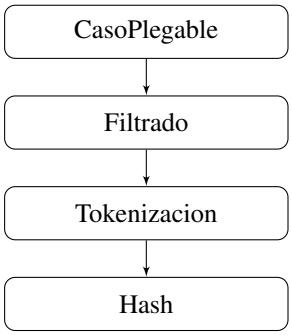

Fig. 2. Pasos del prepocesamiento

1) Caso plegable: Se convierten todas las letras en mayúsculas y se conserva solo el texto, eliminando formatos, imágenes, caracteres irrelevantes, etc. Obteniendo como resultado una cadena de texto.

2) Filtrado: El filtrado consiste en seleccionar palabras importantes, obtenidas de la tokenizacion, eliminando las palabras que se encuentran en una lista de palabras irrelevantes como conjunciones.

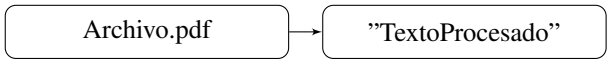

Fig. 3. Ejemplo del caso plegable

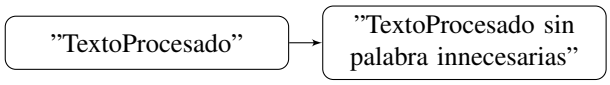

Fig. 4. Ejemplo de Filtrado

3) Tokenizacion: Tiene la función de cortar la cadena de entrada e función de cada palabra que la compone.

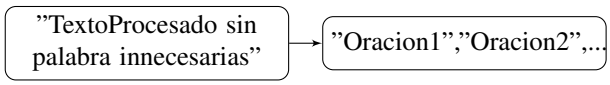

Fig. 5. Ejemplo de Tokenizacion

4) Hash: El proceso de ponderar cada palabra en un documento con un valor basado en una fórmula predeterminada.

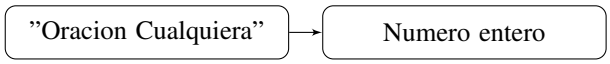

Fig. 6. Ejemplo de Hash de una Oracion

C. ALGORITMO RABIN KARP

El algoritmo de Rabin-Karp fue introducido por primera vez por Michael O. Rabin y Richard M. Karp en 1987. Este algoritmo utiliza tablas de matriz y métodos hash en la operación. Este método de hash se utiliza principalmente para aumentar la velocidad de búsqueda aumentando las pruebas de igualdad en el texto. Las características de este algoritmo son [1,2]:

1. Utilización de la función hash.
2. La fase de preprocesamiento utiliza la complejidad de tiempo O(m).
3. La fase de búsqueda su complejidad es O(mn).
4. Tiempo necesario: O(m + n),donde m es la longitud de cadena de búsqueda y n la longitud de la cadena a buscar.

De hecho, la función hash almacena la forma de una cadena en otra forma, una enumeración para que una determinada cadena tenga su propio valor de enumeración (único). Debido a que una cadena solo tiene un valor de enumeración, esto es lo que usa el algoritmo Rabin-Karp para acelerar la búsqueda de cadenas en tablas hash. Usando un método como este, habrá una pérdida en la búsqueda en un texto largo, porque en el texto largo habrá la misma numeración de cadena aunque la cadena de destino sea diferente. Por tanto, es necesario realizar una mayor verificación del contenido de la cadena. En realidad, esto puede consumir más tiempo si ocurre en una sub cadena larga. Pero una buena función hash garantiza que esta falta sea poco común, por lo que el tiempo medio

de búsqueda con este método es relativamente bueno.

El algoritmo de Rabin Karp acelera las pruebas de similitud de patrones de sub cadenas en el texto utilizando la función hash. El algoritmo de Rabin-Karp es un algoritmo de coincidencia de cadenas que utilizará la función hash como una comparación entre la cadena de búsqueda (m) con la sub cadena en el texto (n). La función hash proporciona un método simple para evitar comparaciones cuadráticas de números de caracteres en muchos casos o situaciones. En lugar de comprobar cada posición del texto cuando se produce la coincidencia de patrones, es más eficaz realizar comprobaciones solo si el texto que se procesa tiene similitudes con el patrón. Para comprobar la similitud entre estas dos palabras se utiliza la función hash (Fernando, 2009). Si 2 cadenas son idénticas, el valor hash será el mismo, por lo que la búsqueda de cadenas se puede derivar calculando el valor hash del patrón y luego buscando un patrón con el mismo valor hash en los datos de entrada. Si los valores hash son los mismos, se realizará una comparación de los caracteres. Si los resultados de los dos no son los mismos, entonces la sub cadena se desplazará hacia la derecha para evitar la complejidad de tiempo O(n^2), puesto que usaremos será de izquierda a derecha para el proceso de comparación. El cambio se lleva a cabo tanto como (n-m) veces. El cálculo eficiente de los valores hash al cambiar afectará el rendimiento de este algoritmo[2].

D. Procentaje de similitud

el siguiente paso es calcular el porcentaje de similitud de los dos documentos usando los Hash de los dos documentos. La fórmula utilizada es la sigue:

$$P = \frac{2*SH}{(THA+THB)} * 100\%$$

P = porcentaje de similitud
SH = Hash idénticos
THA = Hash total en el documento A
THB = Hash total en el documento B

E. Modelado del problema

El hash es el valor más importante del algoritmo de Rabin-Karp. El resultado del hashletras de k-gramo con un cierto número de bases se obtiene multiplicando el valor ASCII pornúmeros predeterminados donde la base es prima. El método Rabin-Karp tiene provisiones si dos cadenas sonlo mismo, entonces el valor hash debe ser el mismo.

El calculo del hash se realiza hasta que se cumplen todas las oraciones de la lista. Las siguientes tablas 1 y 2 son ejemplos de comparación de documentos después de que los valores hash sea adquirido. El valor hash de la primera tabla se calculará mediante el valor hash de la segunda tabla.

F. Diseño del Aplicativo

El aplicativo fue desarrollado el el sistema operativo de Windowns 10, en el lenguaje de programación C-sharp , haciendo uso del framework .NET 4.7.2. El aplicativo tiene 2 opciones de comparacion, la primera es seleccionando 2 archivos pdf desde el explorador de archivos de windows, la segunda forma es escribiendo 2 textos. En cualquiera de los casos se considerara plagio si el porcentaje de similitud es mayor o igual al 30 %.

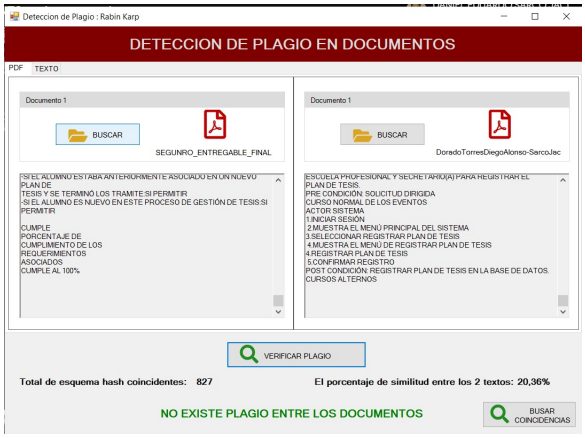

Fig. 7. Figura final del Aplicativo

IV. Conclucion

El algoritmo de Rabin-Karp está muy bien hecho para calcular el porcentaje de similitud de documentos. Además del proceso rápido, este algoritmo tiene parámetros ajustables para ajustar la precisión delevaluación. El cálculo del valor hash afecta en gran medida el resultado de este algoritmo. Los ajustes debenaún debe hacerse al seleccionar el valor de K-Gram que se utilizará. Cada analista puede determinar la viabilidad tolerancia para cada documento, pertenezca o no a la categoría de plagio. La desventajade este algoritmo es que el sistema nunca puede saber qué documentos llegaron primero. El algoritmo puedeSolo determine que existen similitudes que ocurren en los documentos comparables.

References

[1] D. Leman, M. Rahman, F. Ikorasaki, B. S. Riza and M. B. Akbbar, "Rabin Karp And Winnowing Algorithm For Statistics Of Text Document Plagiarism Detection",*2019 7th International Conference on Cyber and IT Service Management (CITSM)*, Jakarta, Indonesia, 2019, pp. 1-5, doi: 10.1109/CITSM47753.2019.8965422.

[2] D. Baitu, M. Luthfi, R. Br Purba, I Ranggadara, "Text Mining To Detect Plagiarism In E-Learning System Using Rabin Karp Algorithm",*Iconic Research And Engineering Journals*, vol. 3, no. 8, pp. 183-191, Feb. 2020, Disponible en: https://www.irejournals.com/paper-details/1701953

[3] A. D. Hartanto, A. Syaputra and Y. Pristyanto, "Best Parameter Selection Of Rabin-Karp Algorithm In Detecting Document Similarity",*2019 International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia, 2019, pp. 457-461, doi: 10.1109/ICOIACT46704.2019.8938458.

[4] B. Leonardo, S. Hansun, "Text Documents Plagiarism Detection using Rabin-Karp and Jaro-Winkler Distance Algorithms", *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 5, no. 2, pp. 462-471, Feb. 2017, doi :10.11591/ijeecs.v5.i2.pp462-471.

[5] S. Shivaji, S. Prabhudeva, "Plagiarism Detection by using Karp-Rabin and String Matching Algorithm Together",*International Journal of Computer Applications*, vol. 116, no. 23, pp. 37-41, Abr 2015, doi : 10.5120/20294-2734.

[6] Cormen Leiserson TH, Rivest CE, Stein RL, Clifford. Introduction to Algorithm. USA: Massachusetts Institute of Technology. 2003.