Tecnológico de Monterrey

Desarrollo de aplicaciones avanzadas de ciencias computacionales

Grupo 201

Pedro Oscar Pérez Murueta
Manuel Iván Casillas del Llano
Benjamin Arteta Obrajero

José Armando Rosas Balderas | A01704132
Ramona Najera Fuentes | A01423596
Ian Joab Padrón Corona | A01708940

Santiago de Querétaro, Querétaro
Mayo 05, 2025

¿Por qué se ha seleccionado el modelo empleado?

Se eligió el modelo descrito en el artículo "SimilaR: R Code Clone and Plagiarism Detection" (Bartoszuk et al., 2020), debido a que su enfoque se alinea con los contenidos vistos en clase y con los conocimientos técnicos del equipo. Aunque el método original utiliza un grafo de dependencias, se adaptó a un grafo de control de flujo por ser más viable dentro del alcance del proyecto. Además, el modelo propuesto es comprensible e intuitivo, lo que facilitó su implementación.

¿Cuáles fueron las variables que se tomaron en cuenta para esa decisión?

Las principales variables consideradas fueron:

- Compatibilidad conceptual con el curso: El enfoque del artículo se relaciona estrechamente con los temas estudiados.
- Dominio del lenguaje Java: Java se eligió por ser ampliamente utilizado, tener una sintaxis clara y facilitar el análisis de patrones estructurales.
- Viabilidad técnica: Se optó por el grafo de control de flujo en lugar del grafo de dependencias debido a su menor complejidad de implementación.
- Capacidad del modelo para detectar similitud estructural: Se priorizo la robustez en la detección de plagio, más allá de simples coincidencias textuales o sintácticas.

¿Cuáles fueron los resultados obtenidos?

El sistema fue evaluado utilizando dos métodos de comparación:

- Cadenas de Markov: Alcanzaron resultados bajos en precisión cuando el orden del código cambiaba, mostrando vulnerabilidad a alteraciones superficiales.
- Distancia de edición de grafos (Graph Edit Distance, GED): Mostró una mayor capacidad para detectar plagio estructural, inclusive en casos donde los códigos tenían la misma lógica pero distinta estructura.

El modelo basado en GED superó el objetivo propuesto del proyecto, que era alcanzar al menos un 70% de precisión, y se comportó de forma consistente en otras pruebas.

Ventajas de modelo propuesto

A comparación del estado del arte, la herramienta que propusimos es más fácil de adaptar para procesar una mayor variedad de lenguajes, porque solo necesitamos proporcionar la gramática del lenguaje para generar el AST y adaptar la lista de palabras reservadas para incluir las alternativas de los identificadores de las estructuras de control, manteniendo el código que existe para procesar cada una.

Además, nuestra propuesta de solución es más sencilla de comprender a comparación de muchos de los proyectos que encontramos al realizar el estado del arte, permitiendo que más

personas puedan tomar como base nuestro proyecto para seguir desarrollando herramientas más especializadas en la detección de plagio en código fuente.