ANÁLISIS DE SIMILITUD EMPLEANDO INTELIGENCIA ARTIFICIAL

Carlos Adrián García Estrada A01707503
José Sebastián Pedrero Jiménez A01703331
Leslie Sánchez Reyes A01708987

CONTENIDOS

01.

EL RETO

EL DERECHO DE AUTOR

Otorgado al creador de una obra original de autoría, siendo esta un libro, artículo, imagen, juegos, entre otras.

El dueño cuenta con el derecho de reproducir la obra y derivaciones de la misma.

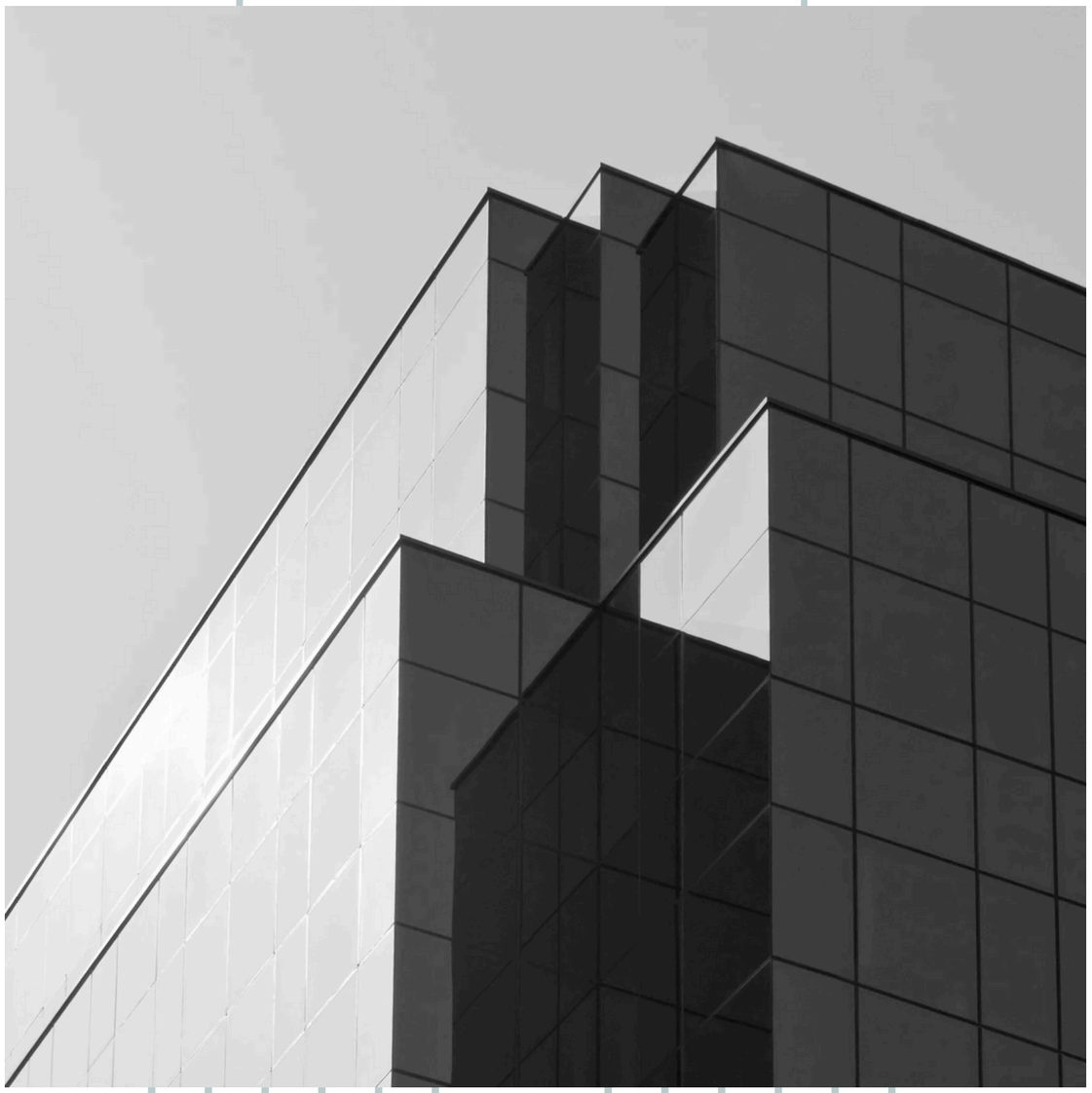
Hoy en día, la distribución de información permite acceso a todo tipo de obras de manera rápida y económica, incrementando la importancia del derecho de autor.

EL PLAGIO

Una infracción en los derechos de autor, donde se reproduce algo como si fuera propio sin acreditar al autor apropiadamente (incluso si eres tú).

Los materiales generados con IA también caen bajo esta definición.

Se desarrolló un modelo de IA que evalua si existe plagio entre dos códigos de java, además usando un método para comparar los dos con cosenos de similaridad como forma de verificación adicional.



02.

METODOLOGÍA

¿PORQUÉ EL MODELO SELECCIONADO?

Criterio:

- Solución con Machine Learning
- Coseno para doble verificación
- Estado del arte (Embeddings)
- Dentro de nuestras habilidades con el tiempo asignado

DATASET USADO

- **ConPlag_Version_2, proporcionado por el profesor Pérez.**
  - Plagio: 251 Pares
  - No Plagio: 660 Pares
  - Código de Java
  - Desbalanceado con sesgo a No Plagio

WORD 2 VEC

Transformar:

```
public class HelloWorld {
    public static void
main(String[] args) {

System.out.println("Hello,
world!");
    }
}
```

A esto:

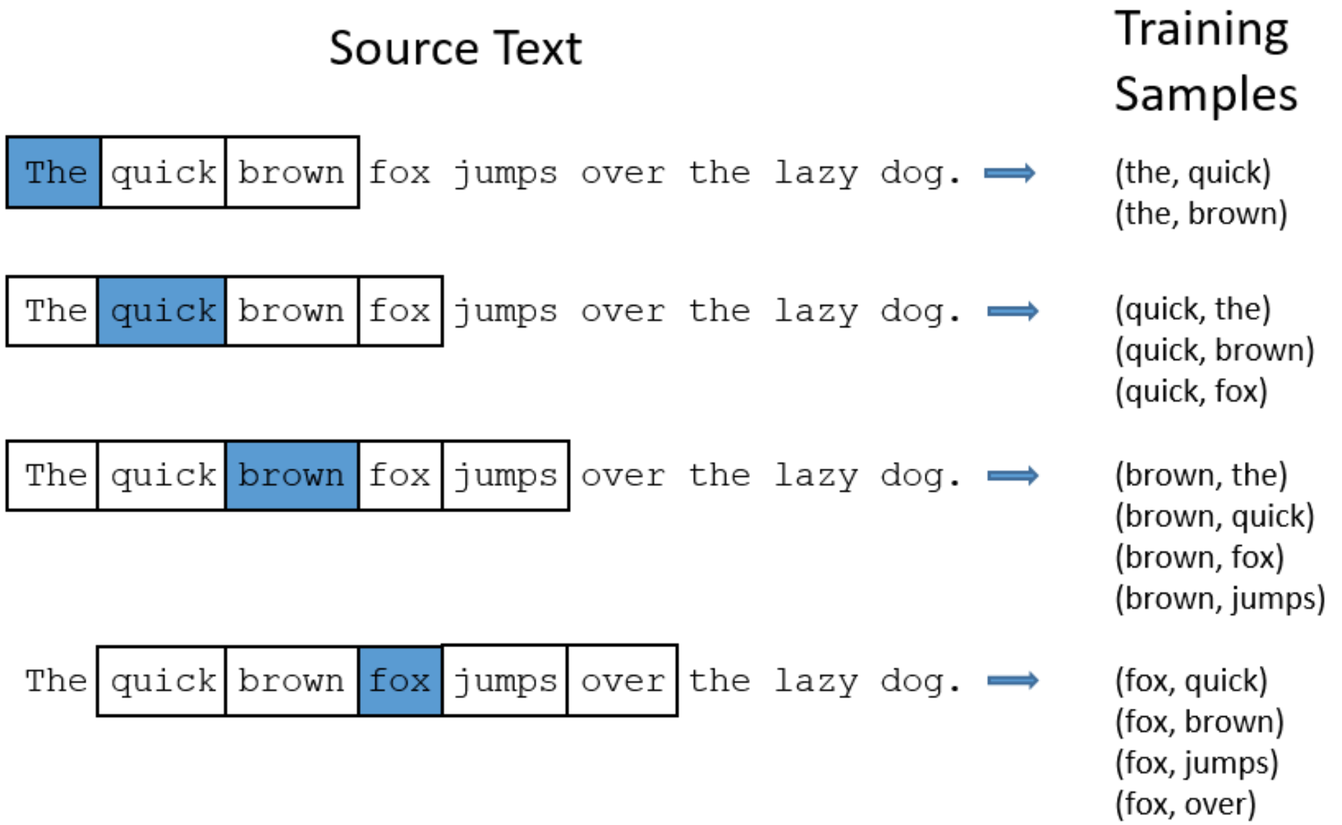
```
[
  [-0.023, 0.085, ..., 0.012],   #
'public'
  [ 0.014, -0.034, ..., 0.078],  #
'class'
  [ 0.061, 0.009, ..., -0.045],  #
'VAR_1'
  ...
]
```

TOKENIZACIÓN

- Generar un diccionario de palabras (similar a BOW)
- Asignar un ID a cada palabra

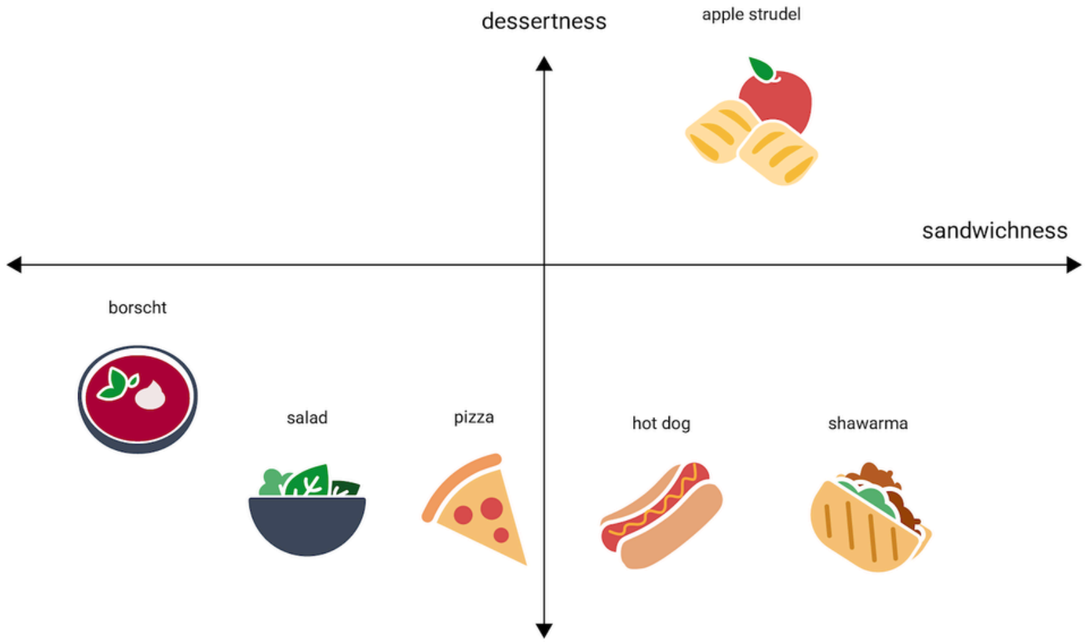
SKIPGRAMS

- Para cada token, predecimos que palabra seguirá o esta detrás de ella
- Generamos una regla probabilística
- Para nuestro modelo utilizamos un skipgram = 5



EMBEDDINGS

- Representación vectorial del significado semántico de las palabras en *n* dimensiones
- Palabras similares tendrán embedding similares
- Detectamos similitud semántica



MODELO DE IA

Arquitectura: Source Code Similarity Indicators Based on Machine Learning

- Capa Densa 8 neuronas activación ReLU
- Capa Densa 10 neuronas activación ReLU
- Capa Densa 10 neuronas activación ReLU
- Capa Densa 10 neuronas activación ReLU
- Capa Densa 1 neuronas activación Sigmoid

- **50 Epochs**

Accuracy teorico: 93.22%

SIMILITUD EN COSENOS

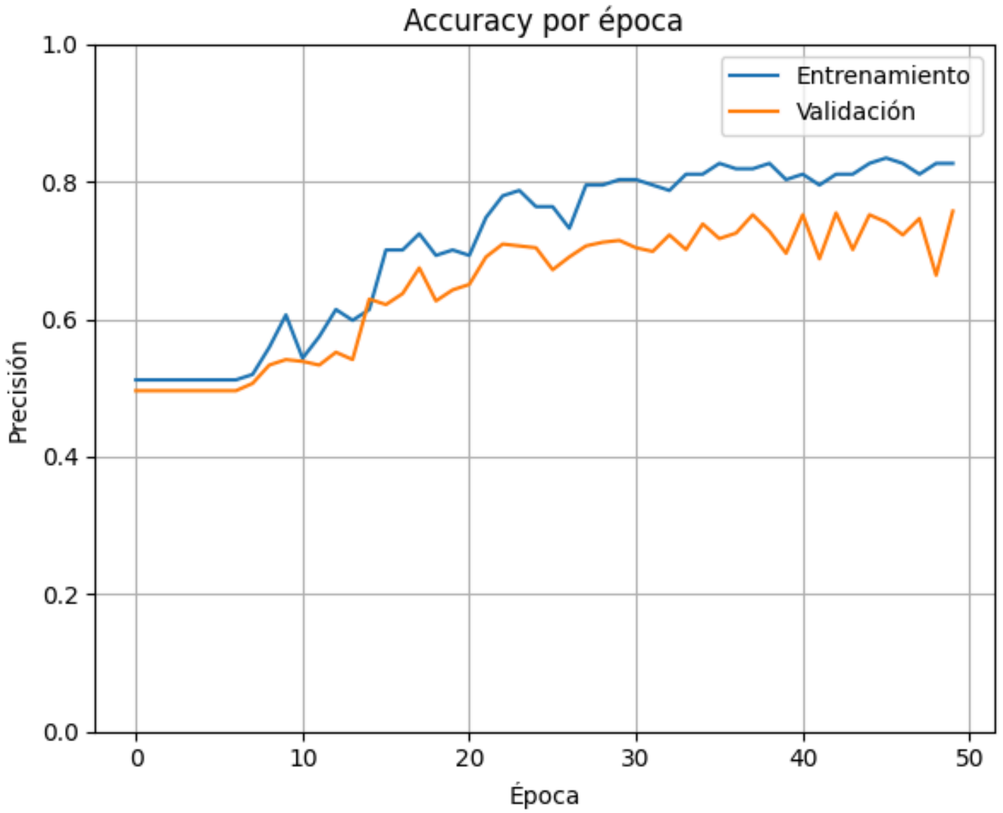
```python
def prepare_pair_embedding(code1, code2, w2v_model):
    tokens1 = preprocess_code(code1)
    tokens2 = preprocess_code(code2)

    vec1 = embed_code(tokens1, w2v_model)
    vec2 = embed_code(tokens2, w2v_model)

    similarity = cosine_similarity([vec1], [vec2])[0][0]

    combined = np.concatenate([vec1, vec2, np.abs(vec1 - vec2)])
    features = combined.reshape(1, -1)

    prediction = model.predict(features)[0][0]

    print(f"\n Similitud coseno: {similarity:.4f}")
    print(f" Probabilidad (modelo) de plagio: {prediction:.4f}")

    if similarity < 0.366:
        print("Resultado: No hay plagio (baja similitud)")
    elif similarity < 0.566:
        print("Resultado: Posible similitud, pero no concluyente")
    else:
        print("Resultado: ¡Plagio detectado!")

    return combined.reshape(1, -1)
```

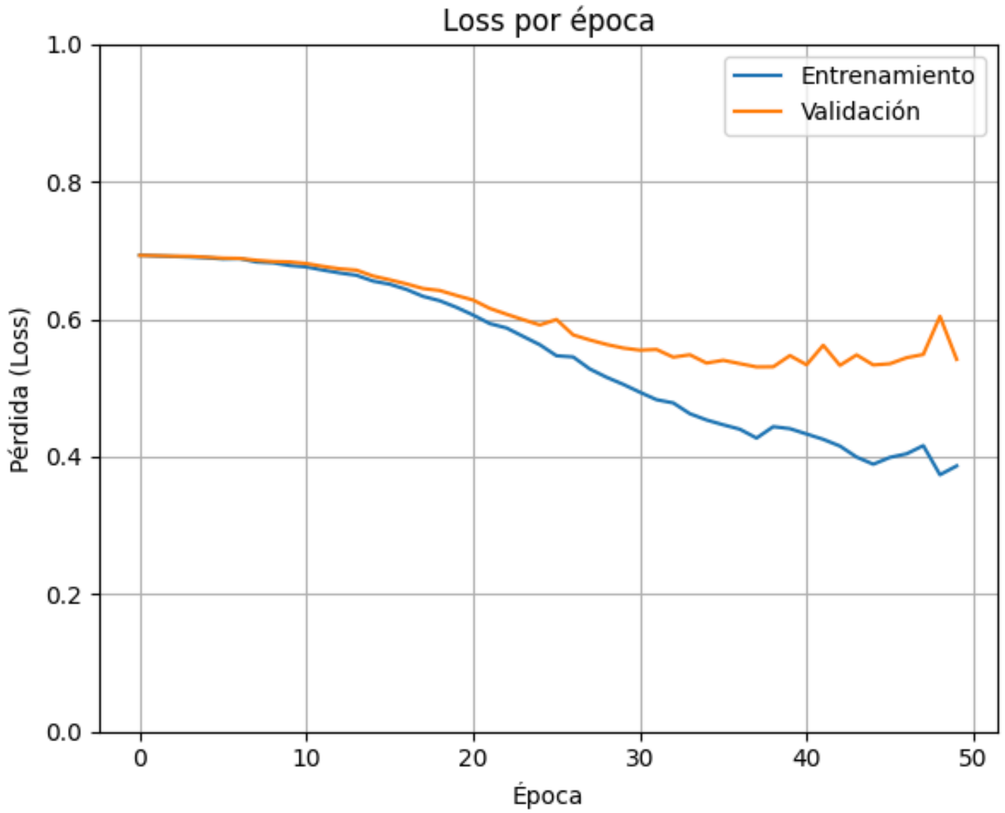
- Una vez procesados los datos, se aplica la formula de cosenos.
- Los porcentajes de plagio fueron basados en investigaciones previas y moda.

03.

RESULTADOS

PRECISIÓN Y PÉRDIDA





Recall = 64.52%

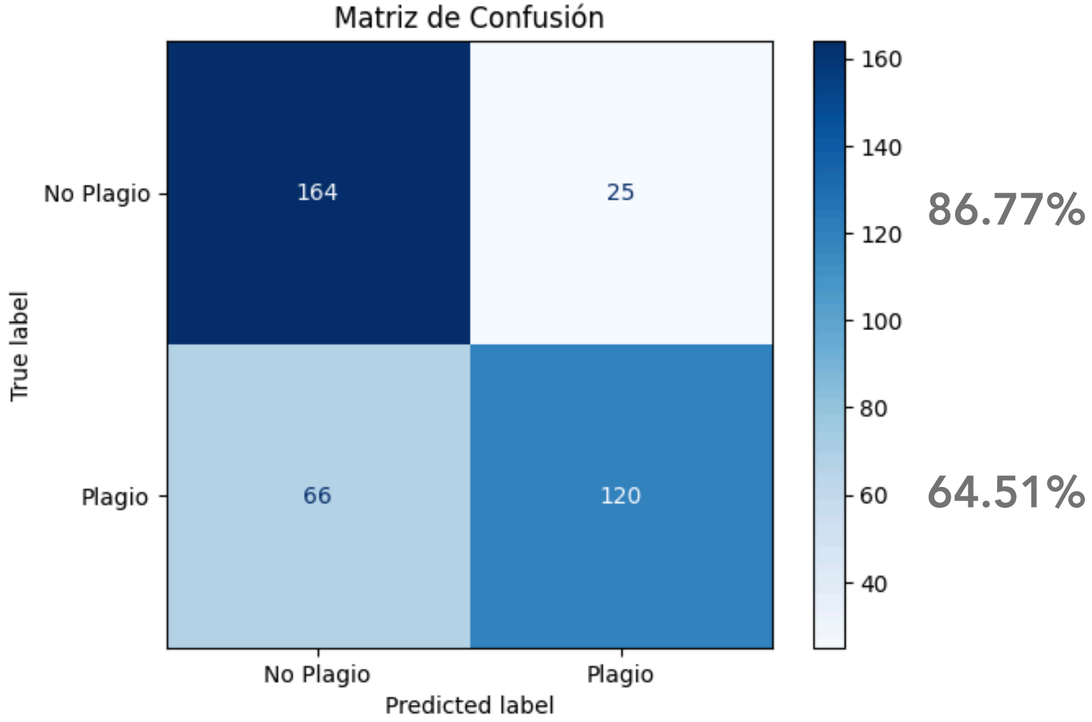
Precisión = 82.76%

F1 = 78.28%

EVALUACIÓN DE MODELO

(Dataset Balanceado)

Precisión: 75.73%



Coseno siempre marcó plagio debido a nuestro preprocesamiento de funciones y variables.

PRUEBAS REALES

No Plagio IA ✓ Cos ✗

```
------ Archivo 1: 464a03b8.java
------ Archivo 2: ff1fc018.java
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step

 Similitud coseno: 0.9928
 Probabilidad (modelo) de plagio: 0.4991
Resultado: ¡Plagio detectado!
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step

Probabilidad de plagio: 0.4991
Resultado: NO HAY PLAGIO
\Certeza del modelo: 0.1182
```

Plagio IA ✓ Cos ✓

```
------ Archivo 1: 548ffb07.java
------ Archivo 2: 3e6def38.java
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step

 Similitud coseno: 0.9967
 Probabilidad (modelo) de plagio: 0.7133
Resultado: ¡Plagio detectado!
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step

Probabilidad de plagio: 0.7133
Resultado: UPS! PLAGIO
\Certeza del modelo: 0.3393
```

04.

ÁREAS DE MEJORA

ÁREAS DE MEJORA

- Cantidad de ejemplos por clase (>1000)
- Modelos clasificatorios más poderosos
  - BERT modificado
- Mejora en la normalización de código
- Implementar detección de llamadas de código

GRACIAS