Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

Desarrollo de aplicaciones avanzadas de ciencias computacionales
TC3002B, Grupo 301

Raúl Monroy
Jorge Adolfo Ramírez Uresti
Ariel Ortiz Ramírez
Miguel González Mendoza

Evidencia 1 - Fase 2 - Parte B: Implementación usando NLP

Equipo #2 | Integrantes:

Germán Guzmán López	A01752165
Isabel Vieyra Enríquez	A01745860
Marco Barbosa Maruri	A01746163

Mayo 2024

Índice

Descripción del Modelo de Solución Propuesto

Los detectores de plagio se han vuelto esenciales para garantizar la originalidad y la integridad en diversos campos. Estas herramientas promueven la creatividad y el pensamiento crítico, al asegurar que los trabajos sean auténticos y que cualquier forma de copia sea identificada. En el ámbito educativo, son fundamentales para evaluar de manera justa y equitativa, manteniendo la integridad académica. Además, protegen los derechos de autor de los creadores, evitando el uso no autorizado de sus obras y fomentando el respeto por la propiedad intelectual.

Para este reto se propone desarrollar una herramienta de detección de plagio utilizando técnicas de procesamiento de lenguaje natural. Esta herramienta comparará textos para determinar la similitud entre ellos. El sistema seguirá la serie de pasos mencionados a continuación para procesar y realizar la comparación de manera efectiva.

1. Preprocesamiento de Texto
2. Generación de Representación Textual
3. Cálculo de Similitud
4. Comparación y Detección de Plagio

El preprocesamiento de texto, que incluye la limpieza de palabras repetidas, puntuación y stopwords, mejora la precisión al eliminar el ruido y centrarse en las palabras significativas. La generación de representación textual mediante la lematización reduce las palabras a su forma base, facilitando la comparación entre términos similares. El cálculo de similitud, utilizando la distancia del coseno, mide la similitud entre textos de manera eficiente. Finalmente, el modelo da como resultado un reporte con la comparación y porcentaje de plagio detectado, proporcionando una evaluación clara de la originalidad del contenido sospechoso.

Medidas de Desempeño del Algoritmo Propuesto

Para evaluar el desempeño del algoritmo de detección de plagio, se ha definido un conjunto claro y específico de medidas de desempeño que se encuentran sujetas a optimización. Nuestro objetivo principal es generar un reporte detallado con los resultados del análisis de similitud, documentando los hallazgos obtenidos durante el proceso de comparación de textos.

El informe final se diseñó para incluir los siguientes elementos:

Documento sospechoso	**Copia (Sí/No)**	**Documento Plagiado**	**Tipo de Plagio**
El archivo que se está evaluando.	Indicador de si se ha detectado plagio	El archivo original del cual se ha plagiado.	La categoría o naturaleza del plagio identificado.

Inicialmente, sin embargo, para realizar las pruebas y tener una vision sobre el comportamiento del proyecto, generamos un reporte prototipo que mostrará:

Archivo	**Copia (Sí/No)**	**Porcentaje de Plagio**	**Archivo Fuente**
El archivo que se está evaluando.	Indicador de si se ha detectado plagio	El porcentaje de similitud encontrado entre el archivo sospechoso y los archivos en el corpus.	El archivo original con el que se presenta mayor similitud con el archivo sospechoso.

Este prototipo permitirá una interpretación preliminar de los resultados del análisis de similitud para cada par de textos comparados. Se destacarán los textos que superen el umbral de similitud establecido, indicando una posible copia. La medida de desempeño clave es la precisión del porcentaje de plagio, que muestra la similitud entre textos. Esta medida se optimiza mediante ajustes en el algoritmo y preprocesamiento del texto, asegurando que los resultados sean precisos y confiables para la identificación de plagio. Posteriormente se buscará implementar un análisis más profundo con las herramientas correctas que nos permitan evaluar el tipo de plagio y de esta manera llegar al informe final que está establecido.

Técnicas de Optimización

Para optimizar el desempeño de la herramienta de detección de plagio, se implementaron varias técnicas específicas que mejoraron significativamente el rendimiento del sistema. Estas técnicas incluyen:

1. **Mejora en el Proceso de Limpieza y Lematización:** Se optimizó el preprocesamiento del texto mediante la inclusión de nuevas expresiones regulares y condiciones que permiten una limpieza más exhaustiva. Ahora, el sistema identifica y procesa contracciones (por ejemplo, "can't" se convierte en "can not"), detecta una mayor variedad de signos de puntuación y elimina una lista más amplia de stopwords. Esta limpieza avanzada asegura que los textos sean más consistentes y homogéneos, lo que mejora la precisión de la vectorización y posterior análisis.
2. **Cambio en la Vectorización de Word2Vec a TF-IDF:** Inicialmente, se utilizó Word2Vec para la vectorización de los textos, que aunque eficaz para ciertos contextos, no proporcionaba resultados satisfactorios para nuestro caso específico. Se decidió cambiar a TF-IDF (Term Frequency-Inverse Document Frequency), una técnica que evalúa la importancia de cada palabra en un documento en relación con un corpus. TF-IDF demostró ser más preciso y relevante para la detección de plagio, capturando mejor las particularidades de cada documento.
3. **Cálculo de un Umbral Ideal Basado en Pruebas:** Se implementó un proceso de evaluación de umbrales para determinar el punto de corte más efectivo para detectar plagio. Se probaron umbrales desde 0 a 1 con incrementos de 0.01, y se evaluaron los resultados mediante métricas de desempeño como precisión, recall y F1. El

umbral que ofreció el mejor balance entre estas métricas fue seleccionado como el umbral óptimo.

El código mejorado ofrece las siguientes ventajas en comparación con la aproximación original:

1. **Mayor Precisión y Eficiencia en la Limpieza del Texto:** Las mejoras en la limpieza y lematización del texto garantizan que los datos sean más consistentes y manejables. Esto reduce el ruido en los datos y mejora la precisión de los modelos de detección de similitud. Al procesar adecuadamente las contracciones y eliminar palabras irrelevantes, se mejora la calidad del texto para su análisis posterior.
2. **Vectorización Más Relevante y Eficaz:** El cambio de Word2Vec a TF-IDF ha demostrado ser más adecuado para la tarea de detección de plagio. TF-IDF ofrece una representación más precisa de la importancia de las palabras dentro de un documento en relación con el corpus, lo que mejora la detección de similitudes relevantes y reduce los falsos positivos.
3. **Umbral de Decisión Basado en Evaluación Sistemática:** El proceso de determinar el umbral óptimo a través de pruebas exhaustivas y métricas de desempeño permite que el sistema sea más robusto y confiable. Este enfoque basado en datos asegura que el umbral seleccionado maximice la precisión y el recall, equilibrando la detección efectiva de plagio con la minimización de errores.

La validación de las mejoras se realizó mediante un análisis exhaustivo de los resultados obtenidos con el código optimizado. Se registraron y analizaron métricas clave de desempeño, y se encontró que el umbral ideal resultante fue 0.19 (o 19%). Los resultados fueron los siguientes:

- **Umbral:** 0.19
- **Verdaderos Positivos:** 9
- **Verdaderos Negativos:** 8
- **Falsos Positivos:** 2
- **Falsos Negativos:** 2

Con este umbral, el sistema logró una precisión de 0.81, un recall de 0.9 y un F1 score de 0.85. Estos resultados validan que las técnicas implementadas mejoraron significativamente la capacidad del sistema para detectar plagio de manera precisa y eficiente. La limpieza avanzada, la vectorización con TF-IDF y la evaluación sistemática del umbral contribuyeron a un sistema más robusto y confiable, superando el rendimiento del código original.

Actividades de Optimización

Conforme se desarrolló el programa se fueron tomando medidas de corrección para obtener mejores resultados y así optimizar nuestra herramienta.

Comenzamos por mejorar y optimizar de manera significativa la limpieza del texto al tomar nuevas condiciones o expresiones regulares que pudiéramos usar a nuestro favor para depurar los textos a evaluar y así tener datos más fáciles de manipular y analizar. Por ejemplo ahora se identifican las contracciones, se encuentran más signos de puntuación y

una variedad más amplia de stopwords. Estas mejoras resultaron en datos más consistentes y coherentes, al eliminar inconsistencias y normalizar el texto, se mejoró la precisión de los modelos de detección de similitud, permitiendo al algoritmo detectar con mayor exactitud los casos de plagio y mejorando la fiabilidad de los resultados.

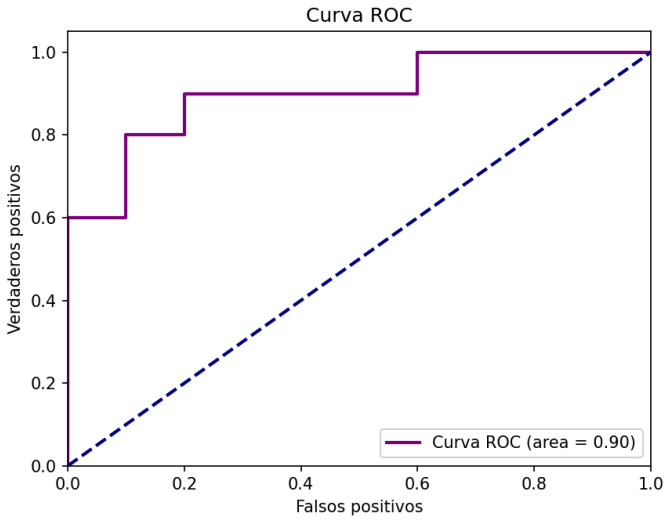
Ya teniendo un mejor preprocesamiento, proseguimos a trabajar en el módulo de generación de representación textual. En este módulo buscamos que el texto se transforme en una representación numérica utilizando técnicas como embeddings.
Inicialmente, utilizamos Word2Vec para medir la similitud entre textos. Word2Vec es una técnica de modelado de lenguaje que transforma las palabras en vectores de alta dimensionalidad, capturando las relaciones semánticas entre ellas. Sin embargo, aunque Word2Vec es eficaz en representar el significado de las palabras en contextos similares, no logró proporcionar resultados satisfactorios en nuestro caso. Los resultados fueron consistentes pero no lograron capturar de manera adecuada la similitud entre los textos base y el texto con plagio. Nos encontrábamos con que el porcentaje de similitud rondaba entre los 90's y el 100 por ciento con todos los textos sin importar el contenido.
Posteriormente, probamos con SpaCy, una librería de NLP que incluye funcionalidades como el etiquetado de partes del discurso, lematización (que fue de donde partió nuestro uso de la librería) y análisis de dependencias. Originalmente, probamos SpaCy por su algoritmo de lematización pero al adentrarnos en la documentación encontramos su potencial para utilizarla en el proyecto. SpaCy ofrece capacidades de similitud semántica utilizando modelos pre-entrenados, los cuales planeamos utilizar para no tener que entrenar un modelo por nuestra parte. Aunque SpaCy mejoró la precisión en comparación con Word2Vec, seguíamos encontrando resultados que no eran coherentes con las pruebas a las que sometemos el código.
Luego, exploramos el uso de n-gramas como lo vimos en clase, basándonos en que ya los sabemos utilizar y su implementación iba a ser relativamente sencilla con lo que ya teníamos como avance previo y en su debido momento parecía ser la mejor opción para aproximarnos a la solución. Esta técnica es útil para capturar patrones en el texto, pero resultó ser insuficiente para nuestro programa. Aunque los n-gramas mejoraron la detección de plagio en ciertos contextos, su efectividad disminuye con textos demasiados cortos como los que utilizamos para las pruebas, además de no ser tan eficiente como las soluciones previas y la solución final a la que llegamos.
Finalmente, implementamos la librería TF-IDF. TF-IDF es una técnica que evalúa la importancia de una palabra en un documento en relación con un corpus de documentos. Esta importancia se calcula multiplicando la frecuencia de término (TF) de una palabra por la frecuencia inversa de documentos (IDF). TF-IDF demostró ser la herramienta más efectiva para nuestra tarea de detección de plagio. A diferencia de las técnicas anteriores, TF-IDF pudo capturar las particularidades y diferencias en la frecuencia de términos de manera más precisa y relevante. La implementación de TF-IDF es relativamente sencilla y computacionalmente eficiente, permitiendo un procesamiento rápido y eficaz de grandes volúmenes de texto.

Resultados de la Optimización

Primero se optimizó la limpieza de los textos, esto resultó en datos más consistentes y coherentes, mejorando la precisión del modelo en detección de similitud. Posteriormente, al

intentar implementar Word2Vec, el cual es bastante eficaz en representar el significado de las palabras; sin embargo, en nuestro modelo no logró sacar resultados satisfactorios, ya que clasificaba casi todos los textos con un 99% de similitud. Después, al probar la misma librería que se utilizó para la lematización, SpaCy, se pudo observar una mejora en la precisión, pero los resultados aún no eran coherentes no se podía detectar el plagio de un documento exactamente igual a uno de los originales, marcaba aproximadamente 43% de plagio cuando debía ser 100%. Se intentó con n-gramas, que aunque útiles en ciertos contextos, resultaron insuficientes para detectar el plagio. Finalmente, al implementar TF-IDF para la vectorización, se logró la mejor optimización hasta el momento. Este algoritmo al ser diseñado para calcular el peso de cada token en un documento en relación con su frecuencia en un corpus, logró capturar las particularidades y diferencias en la frecuencia de tokens de manera más precisa y relevante, siendo la técnica más efectiva para la detección de plagio.

Evaluación del Software Desarrollado



	Falsos Positivos (FPR)	Verdaderos Positivos (TPR)
0	0.0	0.0
1	0.0	0.1
2	0.0	0.6
3	0.1	0.6
4	0.1	0.8
5	0.2	0.8
6	0.2	0.9
7	0.6	0.9
8	0.6	1.0
9	1.0	1.0

Para la evaluación de la discriminación del modelo se utilizó una Curva ROC para medir su rendimiento. Esta curva es una herramienta que permite visualizar la habilidad del modelo para distinguir entre casos positivos y negativos a través de diferentes umbrales de decisión. En esta curva, el eje Y representa la tasa de verdaderos positivos (TPR) y el eje X representa la tasa de falsos positivos (FPR).

Además, calculando el área bajo la curva (AUC), podemos evidenciar que el modelo tiene un muy buen desempeño, ya que el valor del AUC es de 0.9. Esto indica una alta capacidad para separar ambos tipos de documento (plagiados y originales), tomando en cuenta que un AUC de 1.0 representa un modelo perfecto y un AUC de 0.5 representa un modelo que selecciona al azar. Como se puede observar la curva ROC se encuentra por encima de la línea diagonal que representa una clasificación aleatoria (50 y 50), lo cual confirma que el modelo es eficaz para la detección de plagio.

Resultado de protocolo de evaluación:

	Archivo	Plagio	Porcentaje de Plagio	Fuentes
17	FID-018.txt	Si	100.000000	org-050.txt: %26.0, org-057.txt: %100.0, org-0...
14	FID-015.txt	Si	99.389202	org-034.txt: %99.0, org-035.txt: %32.0, org-03...
15	FID-016.txt	Si	99.043423	org-046.txt: %99.0
9	FID-010.txt	Si	97.706183	org-091.txt: %98.0, org-097.txt: %22.0
4	FID-005.txt	Si	78.536803	org-023.txt: %79.0, org-059.txt: %32.0
13	FID-014.txt	Si	75.915648	org-011.txt: %49.0, org-019.txt: %76.0
11	FID-012.txt	Si	24.926287	org-078.txt: %25.0
19	FID-020.txt	Si	23.196553	org-003.txt: %23.0, org-007.txt: %19.0, org-01...
18	FID-019.txt	Si	23.196553	org-003.txt: %23.0, org-007.txt: %19.0, org-01...
8	FID-009.txt	Si	20.166577	org-017.txt: %20.0, org-053.txt: %19.0
12	FID-013.txt	Si	19.669468	org-017.txt: %20.0
3	FID-004.txt	No	18.064831	
6	FID-007.txt	No	17.704278	
5	FID-006.txt	No	15.407134	
0	FID-001.txt	No	14.882428	
16	FID-017.txt	No	14.880516	
7	FID-008.txt	No	13.956592	
1	FID-002.txt	No	12.370367	
10	FID-011.txt	No	11.561193	
2	FID-003.txt	No	10.727671	

Relación de Similitud de Texto

Para justificar la relación de similitud entre los textos, se utilizó la distancia coseno. Esta métrica se eligió debido a su eficacia en medir la similitud entre dos documentos, independientemente de su tamaño. La distancia coseno mide el ángulo entre dos vectores en un espacio multidimensional, proporcionando una medida de cuán similares son en términos de contenido. Esta métrica es especialmente útil en el procesamiento de lenguaje natural y la detección de plagio porque se enfoca en la orientación de los vectores (que representan los textos) y no en su magnitud, lo que significa que dos textos con palabras similares en diferentes cantidades pueden ser identificados como similares.

La relación de similitud fue elegida por medio de un análisis exhaustivo de la precisión y eficiencia de diferentes métricas de similitud. La distancia coseno se aplica vectorizando los documentos, lo cual implica convertir cada documento en un vector en un espacio de alta dimensión donde cada dimensión representa una palabra del vocabulario total. Luego, se compara el archivo sospechoso con cada una de las fuentes originales calculando el coseno del ángulo entre sus respectivos vectores.

Los contenidos de estos vectores son los pesos de las palabras, típicamente calculados mediante técnicas de TF-IDF, que ponderan la frecuencia de una palabra en un documento en relación con su frecuencia en el corpus entero, reduciendo el impacto de palabras comunes y destacando las más significativas.

El umbral de decisión de plagio fue determinado mediante un proceso de validación sistemática. Para establecer el umbral óptimo, se realizó un análisis utilizando un archivo CSV que contenía la relación de los archivos sospechosos, indicando si tenían plagio y, en caso afirmativo, especificando los archivos fuente. Se probaron umbrales que variaban de 0 a 1 con incrementos de 0.01 en cada prueba para identificar el umbral más adecuado.

Durante el proceso de validación, se calcularon las siguientes métricas:

- **Verdaderos Positivos (**VP**):** Casos donde el algoritmo identificó correctamente plagio.
- **Verdaderos Negativos (**VN**):** Casos donde el algoritmo identificó correctamente que no había plagio.
- **Falsos Positivos (**FP**):** Casos donde el algoritmo indicó plagio incorrectamente.
- **Falsos Negativos (**FN**):** Casos donde el algoritmo no identificó plagio correctamente.

A partir de estos valores, se calcularon la Precisión (proporción de verdaderos positivos sobre el total de positivos identificados), el Recall (proporción de verdaderos positivos sobre el total de casos positivos reales) y el F1 Score (media armónica entre la precisión y el recall). El umbral que proporcionó el mejor equilibrio entre estas métricas fue seleccionado como el umbral óptimo para la detección de plagio. En este caso, el umbral ideal resultante fue 0.19 (o 19%), con los siguientes resultados:

	Real Positivos	Real Negativos
Predicción Positivos	9	2
Predicción Negativos	2	8

El umbral óptimo basado en la métrica F1 es 0.19 con una precisión de 0.81, un recall de 0.9 y un F1 score de 0.85.

Liga del repositorio de GItHub

https://github.com/GermanGuzmanLopez/HerramientaDeteccionPlagio.git