

Universidad Nacional del Altiplano
Alma mater de los puneños

Facultad de Ingeniería Mecánica eléctrica, Electrónica y Sistemas

Escuela Profesional Ingeniería de Sistemas



PRODUCTO II UNIDAD:

**DESARROLLO DE UN MODELO PARA DETECCIÓN DE PLAGIO EN
DOCUMENTOS MEDIANTE ALGORITMOS DE PROCESAMIENTO DE LENGUAJE
NATURAL**

ALUMNOS:

- ANTHONY JHONATAN PORTUGAL CHIPANA
- FRANKLIN JOSE ARIAS ESCARCENA
- DANY SALCCA LAGAR

INGENIERA: DONIA ALIZANDRA RUELAS ACERO

CURSO: TÓPICOS AVANZADOS EN INTELIGENCIA COMPUTACIONAL

Puno-Perú

2023

ÍNDICE

| | |
|--|----|
| I. INTRODUCCIÓN | 4 |
| II. OBJETIVOS | 5 |
| A. Objetivo General | 5 |
| B. Objetivos Específicos | 5 |
| III. MARCO TEÓRICO | 6 |
| 1. Interfaz | 6 |
| 2. Procesamiento de Lenguaje Natural (NLP) | 6 |
| 2.1. Stemming | 7 |
| 2.2. Lematización | 7 |
| 2.3 Stopwords | 7 |
| 2.4. Tokenización | 7 |
| 3. Vectorización | 8 |
| 4. Distancia euclidiana | 8 |
| 5. Algoritmo | 8 |
| 6. Coseno de similitud | 8 |
| 7. Modelo de detección | 9 |
| 8. Application Program Interface (API) | 9 |
| 9. Plagio | 10 |
| 10. Metodología SEMMA | 10 |
| 11. Umbral | 11 |
| 12. URL | 12 |
| IV. MATERIALES Y METODOS | 13 |
| A. Materiales empleados | 13 |
| 1. Recursos de Hardware | 13 |
| 2. Recursos de Software | 13 |

| | |
|---|----|
| B. Metodología y Procedimiento | 14 |
| V. RESULTADOS | 16 |
| A. Fase 1: Muestreo | 16 |
| B. Fase 2: Exploración | 17 |
| C. Fase 3: Modificación | 19 |
| D. Fase 4: Modelado | 22 |
| 1. Implementación de los Algoritmos NLP..... | 22 |
| 2. Variación de Threshold | 23 |
| 3. Uso del API | 24 |
| 4. Comparación entre Coseno de Similitud y Distancia Euclidiana | 24 |
| E. Fase 5: Evaluación | 24 |
| 1. Determinación del Performance del Modelo de Detección | 24 |
| Interfaz de Usuario | 26 |
| VI. CONCLUSIONES | 28 |
| VII. REFERENCIAS | 29 |
| VIII. ANEXOS..... | 32 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Ilustración 1: Fases de la Metodología SEMMA | 11 |
| Ilustración 2: Documentos de Prueba | 17 |
| Ilustración 3: Función Distancia Euclidiana | 18 |
| Ilustración 4: Función Coseno de Similitud..... | 18 |
| Ilustración 5: Función de Preprocesamiento | 19 |
| Ilustración 6: Vectorización | 19 |
| Ilustración 7: Función de Umbral Dinámico..... | 20 |
| Ilustración 8: Aumento de número de Documentos..... | 20 |
| Ilustración 9: Variabilidad de Formatos..... | 21 |
| Ilustración 10: Configuración de la clave del API | 21 |
| Ilustración 11: Búsqueda en Google | 22 |
| Ilustración 12: Obtener Contenido de Resultados..... | 22 |
| Ilustración 13: Obtención de Oraciones | 22 |
| Ilustración 14: Prueba con Lematización | 23 |
| Ilustración 15: Prueba sin Lematización | 23 |
| Ilustración 16: Error de API..... | 24 |
| Ilustración 17: Resultado cuando hay plagio | 25 |
| Ilustración 18: Resultado cuando no hay plagio | 25 |
| Ilustración 19: Resultado de similitud de 3 documentos a más | 25 |
| Ilustración 20: Resultado de detección de plagio con la API de Google | 26 |
| Ilustración 21: Botón de subir documentos..... | 26 |
| Ilustración 22: Interfaz de Inicio | 27 |

I. INTRODUCCIÓN

Hoy en día la mayoría de autores definen de forma general el plagio como una copia de ideas, pensamientos u obras y su correspondiente presentación o publicación como propias. Sin embargo, consideramos como más adecuada la definición que plantea el IEEE (Instituto de Ingenieros Eléctricos y Electrónicos, por sus siglas en inglés), que indica lo siguiente: “plagiar es reusar las ideas, procesos, resultados o palabras de alguien más, sin mencionar explícitamente la fuente y su autor” (IEEE, 2018).

En este contexto, el desarrollo de modelos de detección de plagio se ha convertido en una herramienta valiosa para prevenir el uso no autorizado de contenido. En este trabajo, se presenta un enfoque innovador basado en algoritmos de Procesamiento de Lenguaje Natural (NLP) para prevenir el problema de la detección de plagio en documentos.

En este estudio, se explora la utilización de técnicas avanzadas de NLP, como la representación vectorial de palabras y el cálculo de similitud de documentos mediante el coseno, para desarrollar un modelo de clasificación capaz de determinar la cantidad de plagio presente en un documento. Se empleará un conjunto de datos de prueba, que incluyen diferentes tipos de plagio y variaciones de texto, para evaluar el rendimiento del modelo propuesto.

Al finalizar, se espera obtener un modelo que pueda ser implementado en sistemas de detección de plagio existentes o incorporado en futuras soluciones tecnológicas para combatir este fenómeno en constante evolución. A través de este trabajo, se busca impulsar el avance en el campo de la detección de plagio mediante el aprovechamiento de los avances en Procesamiento de Lenguaje Natural y así contribuir al fomento de la integridad académica y científica en la sociedad actual.

II. OBJETIVOS

A. Objetivo General

Desarrollar un modelo para detección de plagio en documentos mediante algoritmos de procesamiento de lenguaje natural.

B. Objetivos Específicos

- Elaborar documentos que incluyen diferentes tipos de plagio y variaciones de texto.
- Desarrollar una interfaz, empleando el modelo para detección de plagio de procesamiento de lenguaje natural.
- Evaluar el rendimiento del modelo propuesto en termino de porcentaje de plagio.

III. MARCO TEÓRICO

1. Interfaz

La interfaz es el programa que el usuario utiliza para comunicarse e interactuar. Una interfaz es un dispositivo que permite comunicar dos sistemas que no hablan el mismo lenguaje. Restringido a aspectos técnicos, se emplea el termino interfaz para definir el juego de conexiones y dispositivos que hacen posible la comunicación entre sistemas.

Es un concepto que abarca arquitectura de información, patrones y diferentes elementos visuales que nos permiten interactuar de forma eficaz con sistemas operativos y softwares de diversos dispositivos, también se define como el medio a través del cual el usuario interactúa con un dispositivo tecnológico. Esto abarca todos los puntos de contacto entre la persona y el equipo. (Corrales, 2019).

2. Procesamiento de Lenguaje Natural (NLP)

Es una disciplina que abarca la interacción entre seres humanos y computadoras utilizando el lenguaje humano natural. Combina la lingüística computacional y la inteligencia artificial en un intento de comprender cómo los humanos generan y comprenden el lenguaje, y de hacer que las computadoras realicen tareas útiles con el lenguaje (Jurafsky & Martin, 2009).

También se le define como s una rama de la inteligencia artificial y la lingüística que se centra en la interacción entre las computadoras y el lenguaje humano. Su objetivo es permitir que las computadoras entiendan y utilicen el lenguaje humano de manera significativa y práctica (Jurafsky D. , 2020)

2.1. Stemming

Es una técnica utilizada en NLP para reducir las palabras a su forma base o raíz, eliminando sufijos y prefijos con el fin de agrupar palabras relacionadas que comparten una raíz común. El objetivo del stemming es simplificar el análisis del texto y mejorar la eficiencia en tareas como recuperación de información, clasificación de texto, análisis de sentimientos y otras aplicaciones de procesamiento de lenguaje.

El stemming es el proceso de eliminar sufijos y prefijos de las palabras para reducirlas a su forma base, o 'stem', con el objetivo de mejorar el procesamiento de texto y facilitar la recuperación de información (Manning, Raghavan, & Schütze, 2008)

2.2. Lematización

Es una técnica de normalización de textos que busca reducir las palabras a su raíz (lema). Es muy utilizada para reducir la cardinalidad del vocabulario asociado para diferentes formas flexionadas con un único token ('entreno', 'entrenarás', 'entrenaría' → 'entrenar'). También se usan mucho en motores de búsqueda (KeepCoding, 2023)

2.3 Stopwords

Los stopwords son palabras como 'un', 'el', y 'en' que se eliminan del texto antes de ciertas tareas de procesamiento del lenguaje, ya que son comunes y aportan poco valor semántico para el análisis (Jurafsky & Martin, Speech and Language Processing, 2019).

2.4. Tokenización

La tokenización es el proceso de dividir un texto en unidades más pequeñas llamadas tokens. Un token puede ser una palabra, un carácter, una frase o cualquier otra unidad lingüística, según

el nivel de granularidad requerido para el análisis (Manning, Raghavan, & Schütze, Introducción a la lingüística computacional, 2008)

3. Vectorización

La vectorización en Procesamiento del Lenguaje Natural (NLP) es el proceso de convertir el texto o las palabras en un formato numérico, representando el texto como vectores matemáticos. Estos vectores numéricos permiten a los algoritmos de aprendizaje automático y otros modelos estadísticos trabajar con datos de texto, ya que los algoritmos generalmente requieren datos numéricos como entrada.

4. Distancia euclidiana

La distancia euclidiana es la distancia más corta entre dos puntos en un espacio euclidiano, y se calcula mediante el teorema de Pitágoras, sumando la diferencia de cada coordenada elevada al cuadrado y luego tomando la raíz cuadrada del resultado (Bishop, Pattern Recognition and Machine Learning, 2006)

5. Algoritmo

Un algoritmo es un conjunto ordenado de instrucciones o pasos precisos que se siguen para resolver un problema específico o llevar a cabo una tarea en un número finito de pasos (Cormen, Leiserson, Rivest, & Stein, 2009)

6. Coseno de similitud

El coseno de similitud es una medida numérica utilizada para evaluar la similitud entre dos vectores en un espacio multidimensional. Es comúnmente aplicado en campos como la recuperación de información, el procesamiento del lenguaje natural y la minería de datos, entre otros.

En términos simples, el coseno de similitud mide el ángulo entre dos vectores, lo que implica que valores cercanos a 1 indican una alta similitud entre los vectores (ángulo cercano a 0 grados), mientras que valores cercanos a 0 indican una baja similitud (ángulo cercano a 90 grados) y valores negativos indican similitud en dirección opuesta (D. Manning, Raghavan, & Schütze, 2009).

7. Modelo de detección

Es un enfoque o sistema que se utiliza para identificar o detectar la presencia o ausencia de ciertas características, patrones o eventos en un conjunto de datos. Estos modelos son comúnmente empleados en diversos campos como el procesamiento de señales, la inteligencia artificial, la ciberseguridad, la detección de fraudes, entre otros.

Un modelo de detección se entrena utilizando datos etiquetados, es decir, datos en los que ya se conoce la presencia o ausencia de la característica o evento de interés. El modelo utiliza esta información para aprender a distinguir entre casos positivos (presencia del evento) y casos negativos (ausencia del evento). Una vez entrenado, el modelo puede utilizarse para realizar predicciones sobre nuevos datos y determinar si la característica o evento está presente o no (Bishop, 2006).

8. Application Program Interface (API)

Es un conjunto de reglas y protocolos que permiten que distintos programas o aplicaciones se comuniquen entre sí para compartan datos o funcionalidades de manera estandarizada y segura. Las API actúan como intermediarios que permiten que una aplicación acceda a ciertas funciones o datos de otra sin necesidad de conocer los detalles internos de cómo están implementadas.

Las API se utilizan ampliamente en el desarrollo de software para facilitar la integración entre diferentes sistemas y componentes. Por ejemplo, en el desarrollo web, las API permiten que aplicaciones y sitios web interactúen con servicios externos como redes sociales, servicios de pago, proveedores de mapas, entre otros. También se utilizan en aplicaciones de escritorio y móviles para acceder a funciones del sistema operativo y compartir datos entre aplicaciones (Jacobson, Brail, & Woods, 2011).

9. Plagio

El plagio es la acción de presentar, copiar el trabajo, ideas, palabras o creaciones originales de otra persona, ya sea de manera total o parcial, sin dar el crédito adecuado o sin obtener la autorización correspondiente. Es una forma de violación de los derechos de autor y de la propiedad intelectual, y se considera un acto deshonesto y poco ético en el ámbito académico, profesional y creativo.

El plagio puede tomar diversas formas, como copiar y pegar texto de fuentes sin citarlas adecuadamente, parafrasear o reescribir ideas sin otorgar la atribución correspondiente, utilizar imágenes, gráficos o cualquier otro tipo de material sin el permiso adecuado, entre otros (Rodríguez, 2012).

Es importante destacar que el plagio es una práctica inaceptable en cualquier contexto y puede tener consecuencias graves, como la pérdida de reputación, sanciones académicas o legales y daños a la carrera profesional.

10. Metodología SEMMA

La metodología SEMMA es un enfoque comúnmente utilizado en el campo del análisis de datos y la minería de datos. Es un acrónimo que representa las etapas clave del proceso de análisis de

datos, y fue popularizado por la empresa SAS Institute, una compañía de software especializada en análisis de datos y estadísticas (Manuel, 2018).

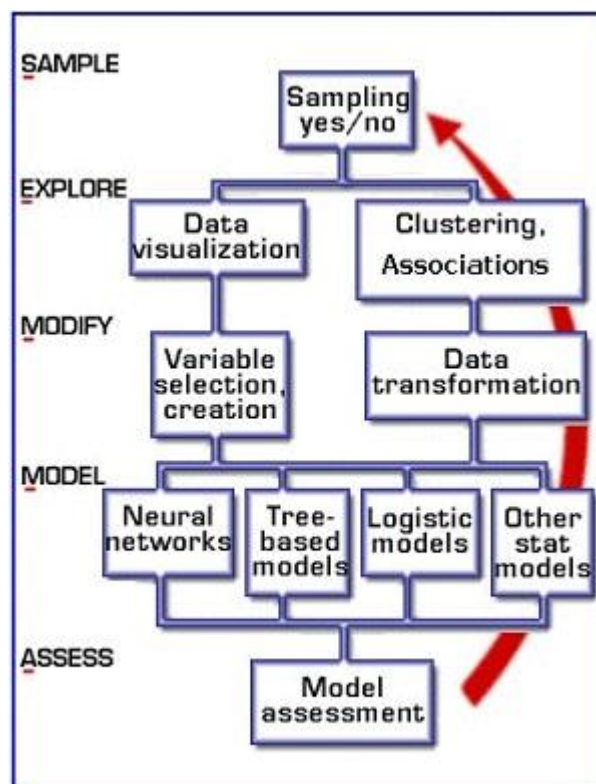


Ilustración 1: Fases de la Metodología SEMMA

11. Umbral

En el procesamiento de datos, un "umbral" es un valor límite o punto de corte que se utiliza para tomar decisiones o clasificar datos. Es una técnica comúnmente utilizada para separar o filtrar datos en dos categorías distintas, como "positivo" y "negativo", "verdadero" y "falso", o "aprobado" y "rechazado". Los datos que superan el umbral se clasifican en una categoría, mientras que los datos que no lo superan se clasifican en la otra (Bishop, 2006).

En nuestro caso práctico, se establece un umbral para clasificar a los plagios como "Posible plagio" si califican con una puntuación mayor o igual a 0.8 y como "No hay plagio" si califican con una puntuación menor a 0.8.

12. URL

Un localizador uniforme de recursos (URL) es una dirección web completa que apunta a un archivo específico en Internet (Diana, 2023).

En cuanto a su estructura, una URL consta de varios elementos:

- HTTP o HTTPS. Es un protocolo de comunicación en red que conecta los servidores web y los navegadores. Este último es más seguro que el primero.
- www. También llamado subdominio, es la parte que precede al primer punto de una URL. Los propietarios de los sitios también pueden utilizar cualquier palabra o frase para organizar su sitio web.
- Nombre de dominio. También conocida como dirección del sitio, es lo que los usuarios escriben en sus navegadores para llegar a un sitio web.
- Extensión de dominio. Es la parte que sigue a un nombre de dominio, por ejemplo, .com y .org.
- Ruta del recurso. Separado por el signo de la barra diagonal (/), este elemento da información adicional a la dirección de un sitio web.
- Parámetros. Suelen llamarse cadenas de consulta o variables de URL. Un signo de interrogación indica un parámetro.

IV. MATERIALES Y METODOS

A. Materiales empleados

1. Recursos de Hardware

Tabla 1. Hardware Utilizado en el proyecto

| N° | NOMBRE | CARACTERÍSTICAS |
|----|----------------------------|--|
| 1 | PC TUF GAMING FX504 SERIES | Procesador: AMD Ryzen 5 3600 6-Core 3.59 GHz RAM: 16.0 GB |
| 2 | DESKTOP-MTBFAS2 HP | Procesador: Intel(R) Core (TM) i5-7200U CPU @ 2.50GHz 2.70 GHz RAM: 4.0 GB |
| 3 | DESKTOP-IK8G9KN LENOVO | Procesador: Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz 2.30 GHz RAM: 8.00 GB |

2. Recursos de Software

Tabla 2. Software Utilizado en el proyecto

| N° | NOMBRE | VERSIÓN |
|----|---------------------|---|
| 1 | Google Colaboratory | Python actualizado de 3.10.11 a 3.10.12 |
| 2 | Google Chrome | Google Chrome 114.0.5735.201 |
| 3 | Visual Studio Code | Visual Studio 1.80.1 |
| 4 | Python | Python 3.11.4 |

Tabla 3. Librerías Utilizadas

| N° | NOMBRE | FUNCIÓN |
|----|----------|--|
| 1 | nlTK | Para obtener la parte de preprocesamiento. |
| 2 | string | Para poder utilizar las funcionalidades de la manipulación de cadenas. |
| 3 | PyPDF2 | Para trabajar con documentos de extensión pdf. |
| 4 | docx | Para trabajar con documentos de extensión docx. |
| 5 | requests | Para obtener el contenido de los URL. |
| 6 | sklearn | Para vectorizar los textos. |

| | | |
|---|------------------------|---|
| 7 | numpy | Para implementar la función de Coseno de Similitud. |
| 8 | re (Expresión Regular) | Para realizar el procesamiento de texto. |
| 9 | os | Para mostrar los nombres de los documentos. |

Tabla 4. API Utilizada

| N° | NOMBRE | DESCRIPCIÓN |
|----|----------------------|---|
| 1 | APIs de Google Cloud | Nos permitirá tener nuestro propio buscador Json. |

B. Metodología y Procedimiento

La metodología seleccionada es SEMMA es un acrónimo (Stands for Sample, Explore, Modify, Model, and Assess) y se adecua para este proyecto ya que se basa en el campo de análisis de datos y la minería de datos. Sus fases son las siguientes:

- **Sample (Muestreo):** En Esta fase se inicia con la extracción de una muestra representativa de la población sobre la que se va a aplicar el análisis. La muestra debe ser representativa para garantizar la validez de todo el modelo y los resultados. La forma más común de obtener una muestra es mediante el muestreo aleatorio simple, donde cada individuo tiene la misma probabilidad de ser seleccionado.
- **Explore (Exploración):** Una vez que se ha determinado una muestra representativa, se procede a la exploración de la información disponible. En esta fase, se utilizan herramientas de visualización y técnicas estadísticas para simplificar el problema y identificar relaciones entre variables. El objetivo es determinar qué variables explicativas se utilizarán como entradas para el modelo.

- **Modify (Modificación):** En base a la exploración realizada, se manipulan los datos para definir y dar el formato adecuado a los datos que serán introducidos en el modelo. Esta fase implica la preparación y limpieza de los datos para asegurar que estén en la forma adecuada para el análisis.
- **Model (Modelado):** Una vez que se han definido las entradas del modelo con el formato adecuado, se procede al análisis y modelado de los datos. El objetivo es establecer una relación entre las variables explicativas y las variables objetivo del estudio para inferir el valor de estas últimas con un nivel de confianza determinado. Se utilizan diversas técnicas de modelado, tanto estadísticas tradicionales como técnicas basadas en datos, como redes neuronales, árboles de decisión, entre otras.
- **Assess (Evaluación):** La última fase del proceso consiste en la valoración de los resultados mediante el análisis de bondad del modelo. Se contrastan los resultados con otros métodos estadísticos o con nuevas poblaciones muestrales para evaluar su eficacia y precisión.

V. RESULTADOS

A. Fase 1: Muestreo

En la primera fase, se elaboro una muestra representativa de nuestros datos sobre la que se aplicara el modelo. Esta muestra se inició con documentos de prueba y entrenamiento, en documentos de entrenamiento no se utilizo ninguno porque el modelo no requiere un entrenamiento especial, pero en el caso de documentos de prueba se utilizaron 20 documentos de los cuales 10 en formato Word y 10 en formato de PDF.

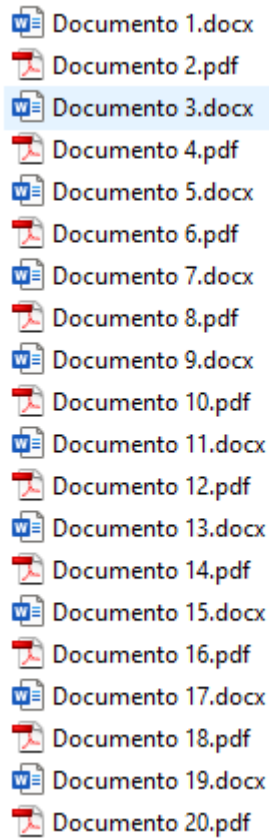


Ilustración 2: Documentos de Prueba

Fuente: Elaboración Propia

B. Fase 2: Exploración

Utilizamos técnicas de procesamiento de lenguaje natural (distancia euclidiana, coseno de similitud, Preprocesamiento y vectorización) para extraer y estructurar los textos de los documentos de ambas muestras.

- **Distancia Euclidiana:** Se utilizó para la realización de la comparación de puntos y así determinar si la información de los documentos contiene algún grado de similitud.

```
def euclidean_distance(vectors1, vectors2):
    return np.linalg.norm(vectors1 - vectors2)
```

Ilustración 3: Función Distancia Euclidiana

Fuente: Elaboración Propia

- **Coseno de Similitud:** De igual manera se utilizo para la comparación de vectores (los vectores representan la información que tiene el texto) para obtener la similitud entre documentos.

```
def coseno_similitud(vector1, vector2):
    dot_product = np.dot(vector1, vector2)
    norm_vector1 = np.linalg.norm(vector1)
    norm_vector2 = np.linalg.norm(vector2)

    if norm_vector1 != 0 and norm_vector2 != 0:
        similarity = dot_product / (norm_vector1 * norm_vector2)
    else:
        similarity = 0.0

    return similarity
```

Ilustración 4: Función Coseno de Similitud

Fuente: Elaboración Propia

- **Preprocesamiento:** Se utilizo para limpiar los textos con los siguientes métodos limpieza general, stopwords, tokenización, eliminación de signos de puntuación y lematización.

```
[ ] def process_text(text):

    tokenizer = TweetTokenizer(preserve_case=False, strip_handles=True, reduce_len=True)
    text_tokens = tokenizer.tokenize(text)

    stopwords_english = stopwords.words('english')
    text_clean = []

    for word in text_tokens:
        if (word not in stopwords_english and
            word not in string.punctuation):
            text_clean.append(word)

    return ' '.join(text_clean)
```

Ilustración 5: Función de Preprocesamiento

Fuente: Elaboración Propia

- **Vectorización:** Primero se utilizó el método TF- IDF para convertir el texto en matrices, luego la matriz se convirtió en un vector mediante la función toarray.

```
[ ] tfidf_vectorizer = TfidfVectorizer()

    tfidf_matrix = tfidf_vectorizer.fit_transform([text1_preprocessed, text2_preprocessed])

[ ] print(tfidf_matrix)

[ ] vector1 = tfidf_matrix[0].toarray()[0]
    vector2 = tfidf_matrix[1].toarray()[0]
```

Ilustración 6: Vectorización

Fuente: Elaboración Propia

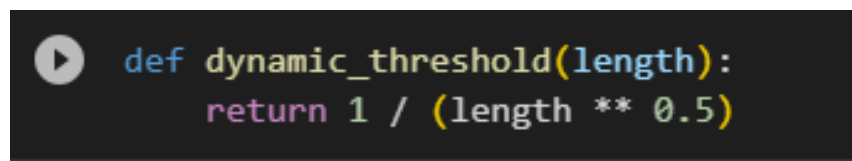
C. Fase 3: Modificación

Después de la fase de exploración se hacen las mejoras en el código dándole las especificaciones necesarias para que el modelo funcione de la mejor manera.

Se tuvo 4 modificaciones las cuales son:

- **Umbral Dinámico:** Se utilizó para determinar si los documentos son detectados como plagio o no, este método se debe a que no se puede hacer comparaciones de textos

extensos con textos cortos, donde se tiene como entrada la longitud de los textos. La función ajusta el umbral directamente proporcional a la longitud de los textos.

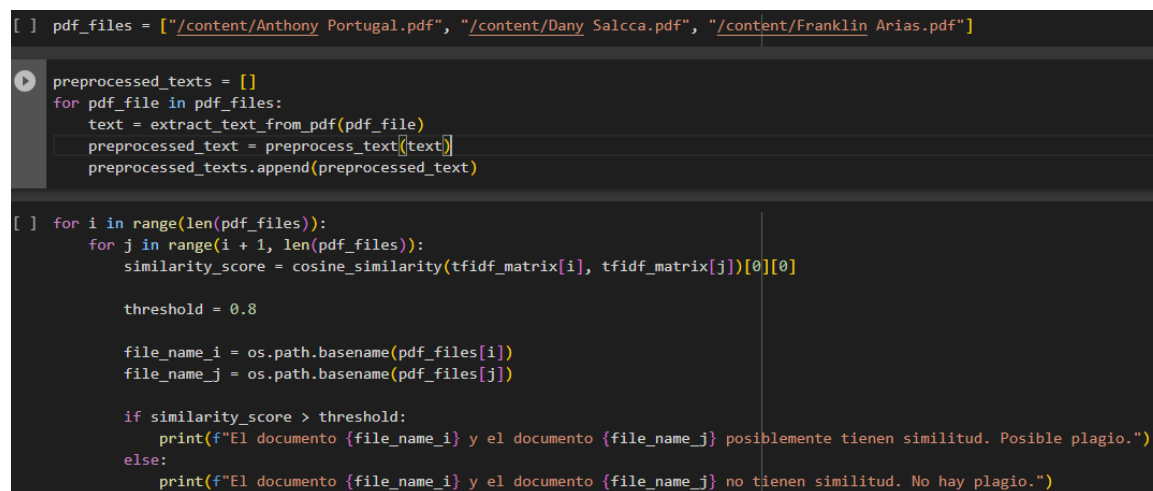


```
def dynamic_threshold(length):
    return 1 / (length ** 0.5)
```

Ilustración 7: Función de Umbral Dinámico

Fuente: Elaboración Propia

- **Cantidad de Documentos:** Se modifico el código para que el modelo pueda detectar el plagio en más de 20 documentos optimizando lo del inicio que solo eran 2 documentos.



```
[ ] pdf_files = ["/content/Anthony Portugal.pdf", "/content/Dany Salcca.pdf", "/content/Franklin Arias.pdf"]

preprocessed_texts = []
for pdf_file in pdf_files:
    text = extract_text_from_pdf(pdf_file)
    preprocessed_text = preprocess_text(text)
    preprocessed_texts.append(preprocessed_text)

[ ] for i in range(len(pdf_files)):
    for j in range(i + 1, len(pdf_files)):
        similarity_score = cosine_similarity(tfidf_matrix[i], tfidf_matrix[j])[0][0]

        threshold = 0.8

        file_name_i = os.path.basename(pdf_files[i])
        file_name_j = os.path.basename(pdf_files[j])

        if similarity_score > threshold:
            print(f"El documento {file_name_i} y el documento {file_name_j} posiblemente tienen similitud. Posible plagio.")
        else:
            print(f"El documento {file_name_i} y el documento {file_name_j} no tienen similitud. No hay plagio.")
```

Ilustración 8: Aumento de número de Documentos

Fuente: Elaboración Propia

- **Variabilidad de Documentos:** Se realizo la modificación para que también se acepten documentos en formato docx.

```

def extract_text_from_docx(docx_path):
    doc = docx.Document(docx_path)
    full_text = []
    for para in doc.paragraphs:
        full_text.append(para.text)
    return '\n'.join(full_text)

[ ] preprocessed_texts = []
for doc in documents:
    if doc.endswith('.pdf'):
        text = extract_text_from_pdf(doc)
    elif doc.endswith('.docx'):
        text = extract_text_from_docx(doc)

    preprocessed_text = preprocess_text(text)
    preprocessed_texts.append(preprocessed_text)

```

Ilustración 9: Variabilidad de Formatos

Fuente: Elaboración Propia

- **Implementación Web:** Se realizó la implementación necesaria para poder realizar las comparaciones entre documentos e información de Google.
 - **Configuración del API:** Se estableció las credenciales necesarias para utilizar el API de Google.

```

API_KEY = "AIzaSyA4gkoX1KN4wfEAWyTdfx7shN5pdCT9HqE"
SEARCH_ENGINE_ID = "b3efb1284abf04927"

```

Ilustración 10: Configuración de la clave del API

Fuente: Elaboración Propia

- **Función para obtener los resultados:** Se estableció la función para poder obtener los resultados encontrados mediante la búsqueda del texto en Google.

```
def search_google(query):
    service = build("customsearch", "v1", developerKey=API_KEY)
    res = service.cse().list(q=query, cx=SEARCH_ENGINE_ID).execute()
    return res.get("items", [])
```

Ilustración 11: Búsqueda en Google

Fuente: Elaboración Propia

- **Función para obtener el contenido de los resultados:** Esta función se utilizo para obtener el texto de los resultados de la búsqueda realizada en Google.

```
def get_page_content(url):
    response = requests.get(url)
    return response.text
```

Ilustración 12: Obtener Contenido de Resultados

Fuente: Elaboración Propia

- **Obtención de Oraciones:** Este proceso se utilizo para mejorar el límite de la extensión de un texto.

```
page_sentences = [sentence.strip() for sentence in page_text.split('. ')]
```

Ilustración 13: Obtención de Oraciones

Fuente: Elaboración Propia

D. Fase 4: Modelado

1. Implementación de los Algoritmos NLP

Estos algoritmos están diseñados para trabajar con datos grandes, pero no hizo uso de las etapas ya que no eran compatibles con el modelo pensado, esto considerando que el modelo planteado no

utiliza un entrenamiento específico, es por eso que se realizó algunas modificaciones al algoritmo para trabajar con documentos de prueba.

El modelo descarta partes como lematización ya que esta técnica consiste en volver la palabra en su forma base afectando al análisis y comparación de la similitud de documentos. La limpieza general no se realizó porque se eliminaba los signos de puntuación y números que también afectaban a la comparación de documentos.

```

Texto procesado con lematización

[ ] text2_preprocessed_1

'qué e la ciencia se denomina ciencia todo el conocimiento saber constituido mediante la observación el estudio sistemático r
azonado de la naturaleza la sociedad el pensamiento el objetivo de la ciencia e descubrir la leyes que rigen los fenómenos d
a realid ad comprenderlos explicarlos de allí se deriva que la función de la ciencia e describir explicar predecir tale fenó
nos fin de mejorar la vida humana la ciencia produce conocimiento científico este se define como todo saber que ha sido obten
mediante el método científico e decir través de la observación el análisis sistemáticos en consecuencia el conocimiento cientí
fico ofrece conclusiones razonadas válidas que pueden ser probadas en este sentido la ciencia comprende todos los campos de c
nocimiento estudio incluyendo ciencias formales naturales sociales humanas que conllevan al desarrol lo de teorías métodos pa
ticulares para cada área la ciencia también está íntimamente relacionada con la tecnología sobre todo desde la segunda...'

```

Ilustración 14: Prueba con Lematización

Fuente: Elaboración Propia

```

Texto procesado sin lematización

[ ] text1_preprocessed

'qué es la ciencia se denomina ciencia todo el conocimiento saber constituido mediante la observación el estudio sistemático r
azonado de la naturaleza la sociedad el pensamiento el objetivo de la ciencia es descubrir las leyes que rigen los fenómenos d
e la realid ad comprenderlos explicarlos de allí se deriva que la función de la ciencia es describir explicar predecir tales f
enómenos fin de mejorar la vida humana la ciencia produce conocimiento científico este se define como todo saber que ha sido o
bteni mediante el método científico es decir través de la observación el análisis sistemáticos en consecuencia el conocimiento
científico ofrece conclusiones razonadas válidas que pueden ser probadas en este sentido la ciencia comprende todos los campos
de conocimiento estudio incluyendo ciencias formales naturales sociales humanas que conllevan al desarrol lo de teorías método
s particulares para cada área la ciencia también está íntimamente relacionada con la tecnología sobre todo desde la s...'

```

Ilustración 15: Prueba sin Lematización

Fuente: Elaboración Propia

2. Variación de Threshold

Al modelar se vio que el umbral threshold varía entre 0.6 y 0.8 ya que esto depende la longitud del texto porque no se puede realizar comparaciones entre textos grandes y pequeños ya que no

nos daría valores reales de similitud, como solución a ello se utilizo el umbral dinámico para controlar la similitud con más precisión.

3. Uso del API

La API proporciona una funcionalidad para buscar similitud entre textos, pero con una restricción de no permitir búsquedas que contengan más de 12 palabras.

```

ConnectionError                                Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/requests/adapters.py in send(self, request, stream, timeout, verify, cert, proxies)
    499
    500     except (ProtocolError, socket.error) as err:
--> 501         raise ConnectionError(err, request=request)
    502
    503     except MaxRetryError as e:

ConnectionError: ('Connection aborted.', RemoteDisconnected('Remote end closed connection without response'))

```

Ilustración 16: Error de API

Fuente: Elaboración Propia

La longitud de texto no permite realizar comparaciones entre textos extensos y pequeños por lo que se optó por una técnica de comparación por oraciones para poder trabajar entre diferentes longitudes de texto, garantizando una comparación equitativa y precisa.

4. Comparación entre Coseno de Similitud y Distancia Euclidiana

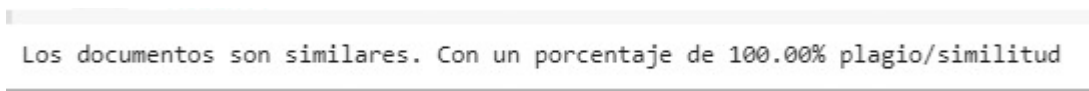
En el proceso de lenguaje natural ambas técnicas se utilizan para medir la similitud o distancia entre vectores sin embargo se optó por la métrica del coseno de similitud por tener características más optimas para el desarrollo de la comparación de plagios.

E. Fase 5: Evaluación

1. Determinación del Performance del Modelo de Detección

Teniendo en cuenta que se trabajó con documentos de prueba se empleó una técnica de porcentaje de plagio. El modelo alcanzo una precisión de más del 90% para documentos que tienen plagio con threshold dinámico que varia según la longitud de los textos a comparar.

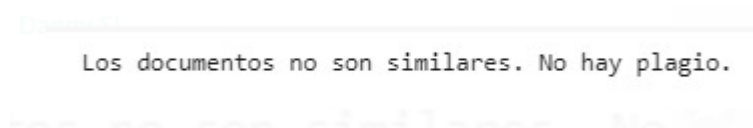
El primer resultado es de 2 documentos subidos donde el modelo compara la similitud entre ellos y su porcentaje.



Los documentos son similares. Con un porcentaje de 100.00% plagio/similitud

Ilustración 17: Resultado cuando hay plagio

Fuente: Elaboración Propia

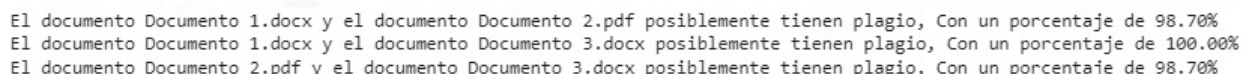


Los documentos no son similares. No hay plagio.

Ilustración 18: Resultado cuando no hay plagio

Fuente: Elaboración Propia

Ahora el modelo compara todos los documentos subidos y ve la similitud entre todos de la misma forma.



El documento Documento 1.docx y el documento Documento 2.pdf posiblemente tienen plagio, Con un porcentaje de 98.70%
 El documento Documento 1.docx y el documento Documento 3.docx posiblemente tienen plagio, Con un porcentaje de 100.00%
 El documento Documento 2.pdf y el documento Documento 3.docx posiblemente tienen plagio, Con un porcentaje de 98.70%

Ilustración 19: Resultado de similitud de 3 documentos a más

Fuente: Elaboración Propia

Finalmente se tiene la modificación utilizando la API de Google que nos ayudará a detectar el plagio de un texto con todos los documentos y páginas de Google, y a la vez nos dará las URL de estas páginas que contienen el plagio.

Plagio detectado en <https://www.significados.com/ciencia/> (Similitud de coseno media: 0.29)
 Plagio detectado en <https://www.gocongr.com/en/flowchart/19128982/se-denomina-ciencia-a-todo-el-conocimiento-a-saber-constituido-por-una-serie>
 Plagio detectado en <https://www.maimonides.edu/pandemia-de-la-ciencia-y-el-sentido-comun-al-cientificismo/> (Similitud de coseno media: 0.34)
 Plagio detectado en <https://www.espaciologopedico.com/revista/articulo/3688/ciencia-hacer-ciencia.html> (Similitud de coseno media: 0.23)
 Plagio detectado en <https://www.argentina.gob.ar/noticias/el-metodo-cientifico> (Similitud de coseno media: 0.27)
 Plagio detectado en http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1025-55832009000300011 (Similitud de coseno media: 0.25)
 Plagio detectado en <https://www.um.es/docencia/barzana/DIVULGACION/CIENCIA/Ciencia-y-metodo-cientifico.html> (Similitud de coseno media: 0.22)

Ilustración 20: Resultado de detección de plagio con la API de Google

Fuente: Elaboración Propia

Interfaz de Usuario

Las interfaces que forman parte del modelo fueron desarrolladas con los lenguajes HTML, CSS, Python y JavaScript, las cuales nos permitirán una conexión con las técnicas desarrollado para la detección del plagio.

1. Botón



Ilustración 21: Botón de subir documentos

Fuente: Elaboración Propia

2. Interfaz de Inicio

The image shows a web application interface for plagiarism detection. The title 'Deteccion de Plagio' is at the top left. Below it is a text input area labeled 'Ingresa el texto:'. To the right of the input area are two buttons: 'Verificar Similitud' (green) and 'Verificar Plagio' (blue). Below the input area is a section labeled 'Selecciona los documentos:' with a red button 'Elegir archivos' and a red box containing the text 'Ninguno archivo selec.'. On the right side of the interface, there is a section labeled 'Resultados' with a large empty space below it.

Ilustración 22: Interfaz de Inicio

Fuente: Elaboración Propia

VI. CONCLUSIONES

- Durante el desarrollo del tema, se pudo comprobar que el coseno de similitud fue una elección acertada para comparar la similitud semántica entre textos. Esta métrica permitió una comparación equitativa y precisa entre documentos, independientemente de su longitud o magnitud, lo que mejoró la calidad de los resultados en la detección de similitudes y plagio.
- La implementación del umbral dinámico fue una solución clave para evitar problemas al comparar textos extensos con textos cortos. Al ajustar el umbral proporcionalmente a la longitud de los textos, se logró un mayor control y precisión en la detección de plagio, permitiendo resultados más realistas y confiables.
- Durante el desarrollo del modelo, se realizaron modificaciones y adaptaciones a los algoritmos de procesamiento del lenguaje natural para trabajar con documentos de prueba. La eliminación de ciertas técnicas, como lematización y limpieza general, demostró ser beneficioso para el análisis y comparación de similitudes entre textos.
- La elección de una técnica de comparación por oraciones resultó esencial para abordar la diferencia de longitud entre textos. Esta aproximación equitativa garantizó una comparación justa y precisa entre documentos largos y cortos, mejorando la calidad de los resultados en tareas de detección de plagio.

VII. REFERENCIAS

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Inglaterra, Reino Unido.

Recuperado el 19 de Julio de 2023, de <http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf>

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*.

Carrión, J., & Serrano, V. (07 de Julio de 2021). evisión sistemática de literatura: características y funcionamiento respecto alos modelos BERT y SQuAD. 08. Recuperado el 20 de Julio de 2023, de <https://revistas.unl.edu.ec/index.php/cedamaz/article/view/1041/793>

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*.

Corrales, J. A. (19 de Agosto de 2019). *Rockcontent*. Obtenido de Interfaz de Usuario: <https://rockcontent.com/es/blog/interfaz-de-usuario/#:~:text=Es%20aquella%20que%20brinda%20informaci%C3%B3n,la%20pantalla%20de%20su%20dispositivo.>

D. Manning, C., Raghavan, P., & Schütze, H. (2009). *Introduction to Information Retrieval*. Inglaterra, Reino Unido. Recuperado el 20 de Julio de 2023, de <https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf>

Diana. (09 de Mayo de 2023). *hostinger*. Recuperado el 19 de Julio de 2023, de Qué es una URL: Ejemplos, estructura y más: <https://www.hostinger.es/tutoriales/que-es-una-url>

IEEE. (2018). *IEEE*. Recuperado el 20 de 07 de 2023, de <https://www.ieee.org/publications/rights/plagiarism/plagiarism.html>

Jacobson, D., Brail, G., & Woods, D. (2011). *APIs: A Strategy Guide*. E.E.U.U. Recuperado el 20 de Julio de 2023, de <https://books.google.cl/books?id=om5tNwKW4xkC&printsec=frontcover#v=onepage&q&f=false>

Jurafsky, D. (29 de Septiembre de 2020). *Cornell University*. Obtenido de Utility is in the Eye of the User: A Critique of NLP Leaderboards: <https://arxiv.org/abs/2009.13888>

Jurafsky, D., & Martin, J. (2009). *Speech and Language Processing (second edition). Book Review*, 4.

Jurafsky, D., & Martin, J. (2019). *Speech and Language Processing*.

Jurafsky, D., & Martin, J. H. (2019). *Speech and Language Processing*.

KeepCoding. (8 de Febrero de 2023). *KeepCodingTechSchool*. Obtenido de Lematización en python: <https://keepcoding.io/blog/que-es-la-lematizacion-en-python/>

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introducción a la lingüística computacional*.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introducción a la lingüística computacional*.

Manuel, J. &. (2018). Metodologías para la realización de proyectos de data mining. Recuperado el 19 de Julio de 2023, de <https://ebb5c31b-a-e1e09691-s-sites.googlegroups.com/a/unicesar.edu.co/alvaroonate/pagina-1/parcelacion/METDOLOGIAS%20MINERIA%20DE%20DATOS.pdf?attachauth=ANo>

Y7cqlqr8DtGzyiXwbmvNmCsdp91zgOFrwXaH2iPHY2S_0RPJfZVYbBx8Cv8vS9VVli
E586hD2XiItpWZ-AbYzoI5awlqmXzTpq

Rodríguez, A. S. (2012). *El plagio y su impacto a nivel académico y profesional*. Costa Rica.

Recuperado el 20 de Julio de 2023, de

<https://revistas.ucr.ac.cr/index.php/eciencias/article/view/1213/1276>

VIII. ANEXOS

Código subido a GitHub: https://github.com/DaNySLagar/Deteccion_de_plagio.git

Código en Colab: <https://colab.research.google.com/drive/11bOJVhpEW54dF8-Derunlvb0MHBBy-GPd?usp=sharing>