Tecnológico de Monterrey

CAMPUS ESTADO DE MÉXICO

TC3002B

Desarrollo de aplicaciones avanzadas de ciencias computacionales

Miguel González Mendoza

Raúl Monroy Borja

Ariel Ortíz Ramírez

Jorge Adolfo Ramírez Uresti

Evidencia

INTEGRANTES

Jorge Rojas Rivas	A01745334
Nadia Paola Ferro Gallegos	A01752013
Melissa Aurora Fadanelli Ordaz	A01749483

20 de mayo de 2024 (Versión 2.0)

Descripción

Para el diseño del modelo de detección de plagio, se utilizarán técnicas de Procesamiento de Lenguaje Natural (PLN) para identificar similitudes entre documentos y detectar posibles casos de plagio. El enfoque principal de nuestro modelo se basará en un pipeline de procesamiento que incluye la técnica de lematización, aplicándola con la librería NLTK de Python.

En el artículo *Lemmatization and Stemming* (Mccabe, 2020) y en el artículo de *Systematic Literature Review of Stemming and Lemmatization Performance for Sentence Similarity* (Gunawan, 2023) se explica en una manera detallada cómo funcionan los métodos de preprocesamiento de *stemming* y lematización. Debido a la información contenida en esos documentos, se ha decidido utilizar lematización sobre otras técnicas. La razón radica en que no solo simplifica las palabras, también las reduce a una raíz común entre las demás palabras relacionadas, teniendo en cuenta el contexto lingüístico. Esto disminuye la variabilidad en el vocabulario, permite una comparación entre palabras más adecuada y brinda una precisión adicional al asegurar que las palabras son comparadas en su forma base, a diferencia del *stemming*, que solo corta los sufijos de las palabras. Aunque, a diferencia del método del *stemming*, la lematización toma un poco más de tiempo en completarse debido a su complejidad, es justificado por la mejora significativa en la precisión.

Además, el pipeline va a contar con procesos de limpieza del texto, tokenización y comparación de textos utilizando la similitud coseno. Esta decisión se tomó basado en el trabajo de De Silva (2023), ya que el texto a menudo contiene mucho ruido, en forma de caracteres especiales, signos de puntuación, números y palabras que se repiten constantemente durante el texto (stopwords), las cuales no dan valor al momento de analizar los textos. Por otro lado, de acuerdo a Kant (2022) la tokenización es un proceso crucial al momento de trabajar con problemas de procesamiento de lenguaje natural, ya que sirve para descomponer el texto en unidades más pequeñas llamados tokens. Los tokens son utilizados en todo el procesamiento del texto ya que la estructura de la tokenización es la que entienden principalmente los modelos. Por último, de acuerdo Koirala, (2021) al momento de trabajar con la comparación de texto, la similitud de coseno es mas ventajosa ya que incluso si los textos están separados por la distancia euclidiana debido al tamaño, aun así pueden tener un ángulo menor, por lo que cuanto menor sea el ángulo mayor la similitud.

Listado de Módulos

Para implementar el pipeline, se dividirá en los siguientes módulos principales:

1. **Módulo de entrada de texto:**

   Este módulo se encargará de recibir un archivo de texto como entrada al sistema, por lo que se llevarán a cabo los siguientes pasos.

a. Se ingresa un archivo de texto al sistema.
b. Una vez que el texto está arriba, se leerá el texto del archivo ingresado.
c. Se tomará la información de identificación del documento, como título y autor, y se verificará que no exista registro dentro de la base de datos, si ya existe se utilizará el vector ya generado anteriormente.

2. Módulo de preprocesamiento:

En este módulo se llevará a cabo el preprocesamiento del texto para prepararlo para la detección de similitud, por lo que se llevarán a cabo los siguientes pasos.

a. Se hará la limpieza de los archivos, como eliminación de: *stopwords* (palabras de uso común) sacadas de la librería NLTK, los símbolos, como paréntesis y signos de puntuación.
b. Después, se generará una lista de *tokens* de las palabras identificadas de la entrada de texto recibida.
c. Una vez realizado este proceso se aplicará el método de lematización a los tokens obtenidos y limpios obteniendo una lista de las palabras de origen.
d. De los documentos de referencia se guardarán dentro de un archivo .csv los tokens resultantes, para usarlos más adelante y no tener que procesar los textos cada vez que se utilice el programa. Del documento que se quiere conocer si es plagio se hará la limpieza en el momento.

3. Módulo de vectorización:

En este módulo se realiza la vectorización del texto procesado junto con un texto de referencia.

a. Se utilizará la librería NLTK para realizar la vectorización a partir de los tokens resultantes del módulo anterior. El vector se hará a partir de la cuenta de la aparición de cada token único identificado dentro de los textos.
b. Se regresarán los vectores resultantes para utilizarlos posteriormente en la medición de similitud. Estos están formados por una matriz de números que representa el uso de cada palabra dentro de los textos.
c. En total se formarán dos matrices, cada una correspondiente a cada uno de los textos para comparar.

4. Módulo de comparación de texto

En este módulo se utiliza la información de los vectores obtenidos con la librería NLTK, y se aplicará el método de similitud elegido.

a. De acuerdo a nuestra investigación, determinamos que la técnica de similitud de coseno será la más apropiada.
b. Para la función de similitud de coseno se envía las matrices que se regresan en el módulo anterior, cada una representando un texto. Esta función nos regresará un número que corresponde a la similitud entre ambos vectores.
c. El número obtenido de la similitud se guardará en una lista de tuplas conformadas por el resultado y el nombre del archivo con el cuál se midió.

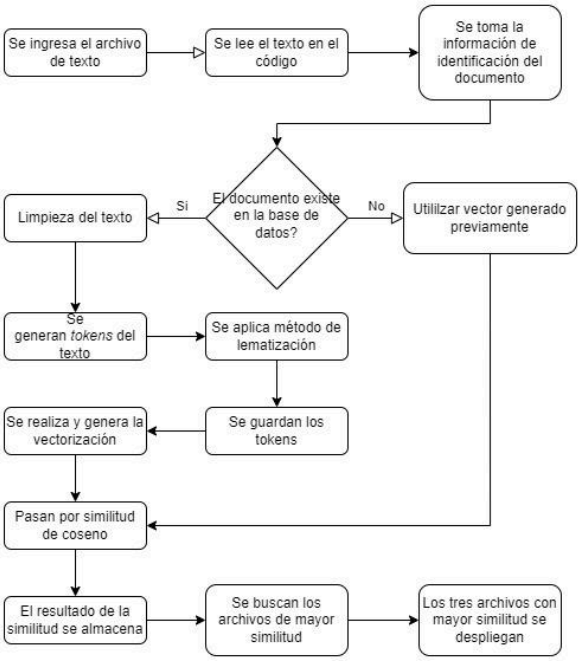
5. **Módulo de visualización de resultados**

En este módulo se visualizarán los resultados obtenidos de la detección de similitud entre documentos.

a. Después de que se termine de llenar la lista de resultados con las similitudes de coseno se elegirán los números con mayor valor, los cuáles son indicadores de mayor similitud.
b. Los tres valores mayores serán guardados en una variable.
c. Finalmente para tomar una decisión se elegirá un umbral de porcentaje para evitar falsos resultados y tener mayor precisión. En este caso se hizo a base de prueba y error, probando 5 umbrales diferentes y viendo cuál de todos se acercaba más a los resultados reales de las pruebas. Determinando finalmente usar un 40% de similitud como umbral.
d. Una vez elegido este umbral se presenta el documento comparado, si se determinó plagio, el porcentaje de similitud y el nombre del documento plagiado.

Descripción de relaciones y dependencias

Los módulos que se encuentran a continuación representan datos y procesos necesarios para la estructura del pipeline antes expuesto. Cada uno de estos módulos depende del proceso realizado que se trabajó por módulos anteriores. Esto debido a la manipulación de datos que se realiza a lo largo del pipeline.



Listado de Datos de entrada y resultado

Para la implementación del pipeline, se encontraron los siguientes datos:

- Datos de entrada:
    - Archivos de texto: Se recibirá el documento que será analizado por el sistema. El archivo tendrá un formato de texto .txt. El contenido del documento va a ser el resumen de un artículo científico el cual va a ser tokenizado.
    - Entrada de teclado: Se recibirá el nombre de un archivo que se encuentre dentro de la carpeta de ejemplos que se encuentra junto con los archivos de referencia subidos en la nube.
    - Carpetas:
        - Carpeta comprimida: Una carpeta en .zip que incluye a su vez dos carpetas, donde se encuentran los archivos de prueba y donde se encuentran los archivos de referencia. Estas llamadas test y train respectivamente.
        - Archivo separado por comas: Un archivo de tipo .csv que está formado por 3 tipos de datos: Índice, Nombre y Lematización, con el objetivo de guardar los tokens lematizados de cada archivo de referencia.
- Datos resultantes:
    - Calculados:
        - Tokens: Refiere al resultado del proceso de tokenización por el que pasan los archivos, son calculados y se utilizan para el proceso de vectorización.

        - Vectores de Texto: Esta es la representación del texto ya preprocesado y lematizado, el cual se vectoriza. Estos vectores nos ayudarán a aplicar técnicas de medición de distancia de similitud, como la distancia de Manhattan. Podremos con esto encontrar los textos que tengan una similitud alta con el documento de entrada.
    - Desplegados:
        - Nombre del texto comparado: Se muestra también el nombre que tiene el archivo que concuerda en la distancia medida. De esta forma podemos identificar los archivos que se parecen en mayor medida al texto que ingresamos.
        - Porcentaje de similitud: Se refiere al número resultante de aplicar el método de medición de similitud de coseno, este número representa qué tan parecidos son los vectores de ambos textos, durante más cercano a 100% sea el valor mayor simil tienen los archivos.
        - Nombre del archivo plagiado: El nombre de los archivos que entran dentro del umbral elegido de similitud. Con el objetivo de identificar el origen de la información del documento.

Conclusión

Se evaluó la herramienta desarrollada con los documentos provistos por los profesores junto con la tabla de resultados esperados. El objeto a evaluar fue la salida de resultados que fueron generados comparados con los resultados esperados. De ella, se identificaron 10 resultados verdaderos negativos (TN), 7 verdaderos positivos (TP), 1 falso negativo (FN) y 2 falsos positivos (FP). Estos números resultan en una tasa de verdadero positivo (TPR) de 87.5% y una tasa de falso positivo (FPR) de 16.6%. Finalmente, utilizando el área bajo la curva operacional receptora (AUC) se puede concluir que la AUC de nuestra herramienta es del 85.41%.

Cabe destacar algunas observaciones registradas durante la evaluación de la herramienta. La primera observación es que, en uno de los resultados (el texto FID-010), se detectó un texto plagiado adicional a pesar de que este no debía ser detectado. Esto se debe a la tasa de plagio que fue declarada para declarar un documento como plagio, sin embargo, esta tasa también fue beneficiosa para identificar un caso de plagio (el texto FID-005). La segunda observación es que se identificó un caso de plagio de un texto completamente diferente. Se detectó plagio en el texto FID-020 del texto org-066, sin embargo se tenía que detectar el org-014. Nunca se identificó parte del texto org-014 y el 066 alcanzó una tasa alta de plagio.

Referencias bibliográficas

1. Mccabe, A. (2020, 14 octubre). Lemmatization and Stemming. Medium. https://medium.com/@alec.mccabe93/lemmatization-and-stemming-5b6b3718b49
2. Kant, U. (2022, 28 enero). Tokenization - A complete guide. Medium. https://medium.com/@utkarsh.kant/tokenization-a-complete-guide-3f2dd56c0682
3. De Silva, M. (2023, 30 abril). Preprocessing Steps for Natural Language Processing (NLP): A Beginner's Guide. Medium. https://medium.com/@maleeshadesilva21/preprocessing-steps-for-natural-language-processing-nlp-a-beginners-guide-d6d9bf7689c9
4. Agung Santoso Gunawan, A. et al. (2023, 20 diciembre). Systematic Literature Review of Stemming and Lemmatization Performance for Sentence Similarity.
5. Koirala, A. (2021, 14 enero). Does text similarity metrics help in text classification problems? Medium. https://medium.com/analytics-vidhya/text-similarity-approach-for-text-classification-f25e284d3e92