Procesamiento de Lenguaje Natural aplicado en la detección de plagio

Yoselint Yovana Vázquez Pérez., Fernando Guzmán Mendoza, Dra. Dora María Calderón Nepamuceno.

[a] Universidad Autónoma del Estado de México CU Nezahualcóyotl, yvazquezp001@alumno.uaemex.mx, Nezahualcóyotl, Estado de México, México.
[b] Universidad Autónoma del Estado de México CU Nezahualcóyotl, fguzmanm002@alumno.uaemex.mx, Nezahualcóyotl, Estado de México, México.
[c] Universidad Autónoma del Estado de México CU Nezahualcóyotl, dmcalderonn@uaemex.mx, Ecatepec, Estado de México, México.

Resumen

La automatización de los procesos en una organización, estancia o dependencia el uso y aprovechamiento total de las capacidades de los recursos tecnológicos y computacionales para el mejoramiento y soporte de los procedimientos, son algunos de los enfoques de mayor auge en el campo de la Informática, debido a que incrementan el rendimiento de las actividades a realizar en dicho espacio. Es por esta razón, que se justifica y fortalece las bases del tema propuesto a desarrollar.

El proyecto se delimita a la implementación de un Sistema aplicado a la comparación de dos textos (por el momento solo con tipo de formato de texto plano **".txt"**), el cual tiene como objeto auxiliar a la revisión de trabajos para la detección de plagio, en este caso estará determinado por el porcentaje de similitud que existe entre ambos textos. Si bien es sabido que para determinar plagio hay que analizar donde ocurre la similitud de texto, ya que no es lo mismo copiar conceptos generales que tener similitud en todo el texto. Sin embargo, este programa representa una herramienta que apoya en la determinación de la similaridad. Opiniones diversas indican que si la similitud entre textos es superior al 15% o 20% podría tratarse de un caso de plagio.

Palabras clave— Comparación, Plagio, Sistemas, Textos.

Abstract

The automation of the processes in an organization, facility or

The automation of the processes in an organization, farm or dependency, the use and total use of the capacities of the technological and computational resources for the improvement and support of the procedures, are some of the approaches of greater boom in the field of the computer science, because they increase the yield of the activities to be carried out in this space. It is for this reason, that justifies and strengthens the bases of the proposed topic to develop.
to develop.

The project is limited to the implementation of a system applied to the comparison of two texts (for the moment only with plain text format type ".txt"), which aims to assist in the review of works for the detection of plagiarism, in this case will be determined by the percentage of similarity between the two texts. It is well known that to determine plagiarism it is necessary to analyze where the text similarity occurs, since it is not the same to copy general concepts than to have similarity in the whole text. However, this program represents a tool that supports the determination of similarity. Different opinions indicate that if the similarity between texts is higher than 15% or 20% it could be a case of plagiarism. of plagiarism.

Keywords— *Comparison, Plagiarism, Systems, Texts.*

1. INTRODUCCIÓN

La Real Academia de la Lengua (RAE) define plagio como la copia de obras ajenas haciéndolas pasar por propias. En el ámbito académico el plagio es usar palabras o ideas de otras personas como si fueran propias. Se considera una forma de engaño y una mala práctica que compromete la honestidad y la integridad académica.

En este proyecto se busca implementar el desarrollo de un comparador de texto para el uso del día a día. De igual manera podrá identificar si es que un trabajo fue plagiado por parte de uno o más personas. Al introducir los documentos este devolverla el porcentaje de similitud que tiene uno con el otro dando a conocer si uno de los textos es copia del otro.

1.1METODOLOGÍA

Procesamiento natural del lenguaje

El procesamiento del lenguaje natural (PLN) es un campo del conocimiento al que han contribuido a su desarrollo disciplinas como la lingüística, la informática, la ciencia cognitiva y la ingeniería electrónica, esta última más estrechamente relacionada con las tecnologías del habla. A lo largo de la historia del PLN los dos enfoques principales de investigación adoptados han sido los paradigmas simbólico y estadístico [1]:

- El enfoque simbólico se caracteriza por la construcción de sistemas que almacenan explícitamente los hechos lingüísticos (p.ej. fonológicos/fonéticos, morfológicos, sintácticos, semánticos, pragmáticos o discursivos) a través de esquemas de representación del conocimiento, desarrollados principalmente de forma manual.
- El enfoque estadístico se caracteriza por la construcción de sistemas que no almacenan explícitamente el conocimiento lingüístico o del mundo, sino que aplican técnicas matemáticas sobre extensos textos informatizad con el fin de inferir dicho conocimiento.

Comparadores de textos

Comparar textos o archivos manualmente es una tarea tediosa y prácticamente imposible. Hay muchas posibilidades de que se pierda algo. Lo más inteligente que puedes hacer aquí sería utilizar una herramienta de comparación de archivos para realizar el trabajo de manera efectiva, mientras ahorras tiempo.

Los comparadores de texto, como su nombre indica, son herramientas con las que pueden compararse varios textos para determinar su validez y autenticidad.

Puede ser útil en diversas situaciones, entre las que se incluyen:

- Detección de plagio: Un comparador de textos puede ayudar a identificar si un texto es similar o idéntico a otro existente en su totalidad o en parte. Esto es especialmente útil en el ámbito académico y en la detección de contenido duplicado en línea.

- Verificación de originalidad: Si tienes un texto y deseas verificar si es original o si existen similitudes con otros documentos, un comparador de textos puede ayudarte a identificar coincidencias parciales o totales con otros textos previamente escritos.

- Revisión y edición: Al comparar diferentes versiones de un texto, un comparador de textos puede resaltar las diferencias entre ellos, lo que facilita la revisión y la edición. Esto es útil en la escritura colaborativa, en la verificación de cambios realizados o en la corrección de errores.

- Análisis de similitud: En áreas como la lingüística, la traducción y la investigación, un comparador de textos puede ayudar a analizar la similitud entre diferentes textos para detectar patrones, identificar características comunes o realizar análisis comparativos.

- Control de calidad de traducciones: En la traducción automática o en la traducción realizada por humanos, un comparador de textos puede ayudar a evaluar la calidad de la traducción al compararla con el texto original o con una referencia de traducción.

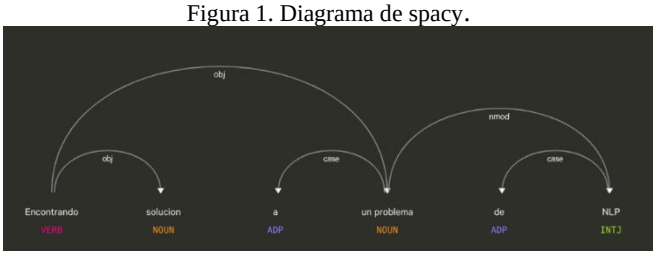
- Análisis forense y legal: En situaciones legales o forenses, un comparador de textos puede ser utilizado para verificar la autenticidad de documentos, analizar similitudes entre textos relacionados con un caso o identificar autorías.

Si bien las aplicaciones como Google Drive o Dropbox pueden ayudarte hasta cierto punto al ejecutar revisiones de archivos, necesitarías utilizar estos servicios. Para eso, también necesitarías una conexión a Internet. Además, debes utilizar estos servicios y cargar los archivos para realizar un seguimiento del servicio de forma regular.

Spacy

Para el desarrollo del proyecto plantado se trabajó con NLP básico con Spacy ya que este es una herramienta para trabajar problemas relacionados con lenguaje, con un amplio soporte de lenguajes provee una sintaxis intuitiva para operar en ámbitos lingüísticos.

Figura 1. Diagrama de spacy.



Fuente: Necronet.

SpaCy es una biblioteca de procesamiento de lenguaje natural Python diseñada específicamente con el objetivo de ser una biblioteca útil para implementar sistemas listos para producción. Es particularmente rápido e intuitivo, por lo que es un competidor superior para las tareas de procesamiento de lenguaje natural (PLN) figura 1. Si bien la compensación es menor flexibilidad en algunos aspectos de su canalización PLN, el resultado debería ser un mayor rendimiento [2].

Los modelos en spacy son módulos pre entrenados a través de Convolucional Neural Networks, se dividen en modelos bases y modelos iniciales. Los primeros son claves para inferir características lingüísticas de los datos, y se clasifican en general en dependencia del idioma. Los segundo son paquetes iniciales con valores ponderados para continuar entrenando otras arquitecturas que puedan resultar más convenientes de acuerdo con el contexto del problema.
Cargar los modelos que se utilizaran para procesar texto, es_core_news_*. SM se refiere a modelos reducidos, más rápidos sin embargo menos precisos. MD sin modelos medios entrenados con mayor datos y mayor precisión [3].

En general su funcionamiento es el siguiente:

1. **Tokenización:** Divide el texto en partes más pequeñas llamadas tokens. Utilizando reglas específicas del idioma para dividir el texto en dichos tokens.
2. **Análisis morfológico:** una vez realizado lo anterior, analiza la estructura morfológica de cada uno de los tokens que se generaron, y genera una reducción de las palabras a su forma base (ej. “escribiendo” a “escribir”)
3. **Análisis sintáctico:** Analiza la estructura sintáctica del texto para comprender como se relaciona las palabras en la oración.
4. **Reconocimiento de entidades nombradas:** identifica nombres, organizaciones, lugares, fechas, cantidades etc., mediante aprendizaje automático.
5. **Integración con modelos de aprendizaje automático:** Utiliza modelos previamente entrenados en diversos idiomas y tareas como se mencionó en el punto anterior, también el análisis de sentimientos y la clasificación de los textos. Estos modelos pueden adaptarse a las circunstancias.
6. **Personalización y extensibilidad:** Permite implementar modelos propios para actividades de PLN específicas.

2. CONTENIDO

Una tarea muy común al momento de realizar trabajo con texto es encontrar la similitud entre muchos documentos o entre oraciones dentro del mismo. Como se mencionó anteriormente Spacy facilita este trabajo al proveer dentro de sus modelos preentrenados funciones que calculan la similitud entre palabras. Con lo antes mencionado se elaboró el siguiente código.

Se importan las librerías tkinter para crear la ventana y sus elementos, y spacy para realizar el cálculo de la similaridad como se muestra en la figura 2.

Requerimientos de las librerías

Tkinter: El paquete tkinter («interfaz Tk») es la interfaz por defecto de Python para el kit de herramientas de GUI Tk. Tanto Tk como tkinter están disponibles en la mayoría de las plataformas Unix, así como en sistemas Windows.
Ejecutar python -m tkinter desde la línea de comandos debería abrir una ventana que demuestre una interfaz Tk simple para saber si tkinter está instalado correctamente en su sistema. También muestra qué versión de Tcl/Tk está instalada para que pueda leer la documentación de Tcl/Tk específica de esa versión.
Tkinter soporta un amplio rango de versiones TCL/TK, construidos con o sin soporte de hilo. La versión oficial del binario de python incluye Tcl/Tk 8.6 con subprocesos.
Tk: El objeto de la aplicación Tk creado al instanciar Tk. Esto proporciona acceso al intérprete de Tcl. Todos los widgets que se adjuntan a la misma instancia de Tk tienen el mismo valor para el atributo tk.
tkinter.filedialog: Cuadros de diálogo por defecto que permiten al usuario especificar un archivo para abrir o guardar.
tkinter.messagebox: Acceso a cuadros de diálogo estándar de Tk.

Spacy : La versión spacy 3.0 se destaca por la implementación de familias de modelos reentrenadas para 18 idiomas y 59 pipelines entrenados en total, incluidos 5 nuevos pipelines basados en transformadores. El modelo se ofrece en tres versiones (16 MB, 41 MB – 20 mil vectores y 491 MB – 500 mil vectores) y está optimizado para trabajar bajo la carga de CPU e incluye los componentes tok2vec, morphologizer, parser, senter, ner, attribute_ruler y lemmatizer. spaCy es compatible con CPython 3.6+ de 64 bits y se ejecuta enUnix/Linux, macOS/OS XyWindows. Los últimos lanzamientos de spaCy están disponibles en pip y conda.
Usando pip, las versiones de spaCy están disponibles como paquetes fuente y ruedas binarias. Antes de instalar spaCy y sus dependencias, asegúrese de que su pipy setuptools estén actualizados.

Figura. 2. Librerías utilizadas.

```
import tkinter as tk
from tkinter.filedialog import askopenfilename
from tkinter import messagebox as MessageBox
import spacy
```

Fuente: elaboración propia.

Se crea la clase SimilaridadApp en la que estarán las funciones a utilizar. Se crea el constructor donde se llama a la función inicializar_gui.

En la función inicializar_gui se crean e inicializan los elementos de la ventana, como los botones y las áreas de texto donde se mostrarán los textos a comparar. También se realizarán las funciones abrir_archivo1 y abrir_archivo2.

En la 1 se abre el primer archivo haciendo uso de askopenfilename, al abrir el archivo se lee, se inicializa el text area en vacío, y se guarda el texto en la lista textos, se bloquea el text area y el botón de abrir archivo.
En la otra función es igual solo que se hace uso del textarea2 en lugar del 1.

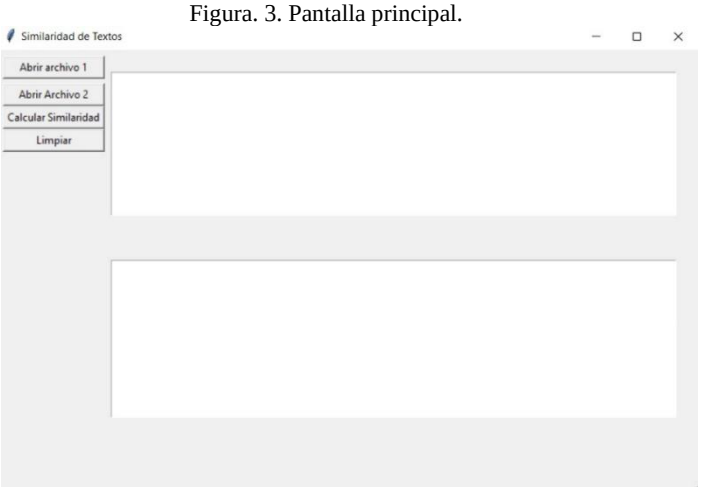
La función limpia se reinicia a vacío los text áreas y la lista textos y se vuelven a activar los botones de abrir y abrir2, en verificaSimilaridad se hace uso de spacy y se carga el idioma español que es el idioma de los textos a comparar.

Solo lo hará mientras la lista Textos tenga 2 textos para comparar la similitud se calcula con la función similarity de spacy y el resultado lo muestra en un MessageBox
En la función main se crean los objetos de Tkinter y el de la aplicación.

2.1 RESULTADOS

La ventana inicial del programa se muestra en la figura 3:

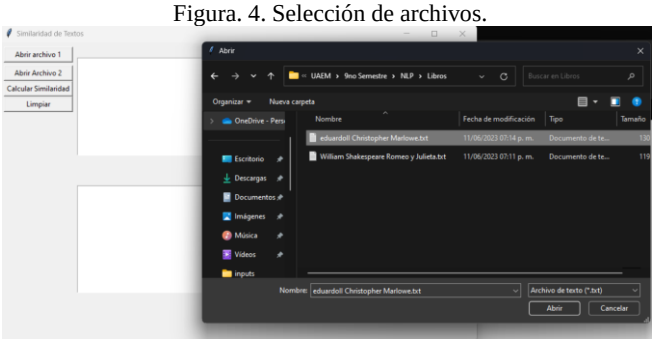
Figura. 3. Pantalla principal.



Fuente: Elaboración propia.

Para verificar la correcta funcionalidad del programa se realizó la comparación entre **"Eduardo II"** del autor Christopher Marlowe y **"Romeo y Julieta"** William Shakespeare. Ya que existe una teoría que menciona que el verdadero autor de algunas obras de William es en realidad Christopher, dada la similaridad que existe en la manera en que se redactan dichos textos

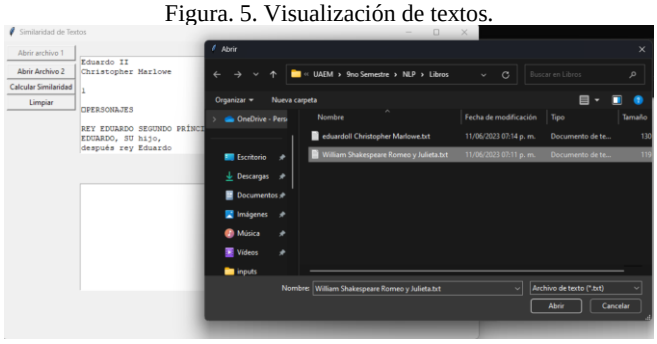
Figura. 4. Selección de archivos.



Fuente: Elaboración propia.

Los archivos abiertos se muestran en los textarea en la figura 4 y 5.

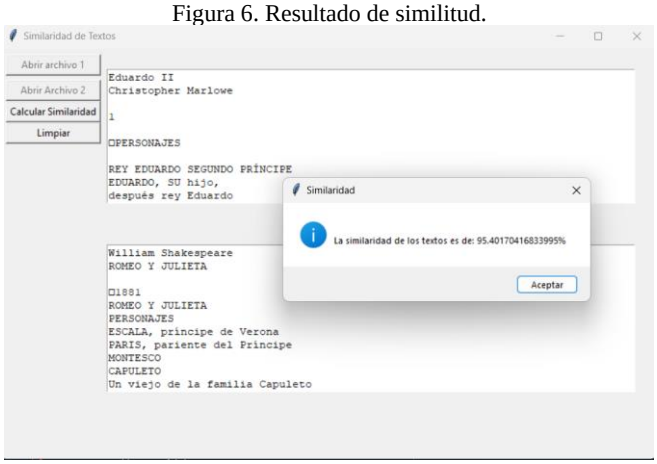
Figura. 5. Visualización de textos.



Fuente: Elaboración propia.

En la figura 6 se muestra el porcentaje de similitud que existen entre los archivos.

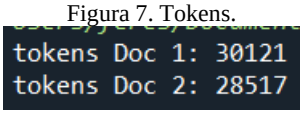
Figura 6. Resultado de similitud.



Fuente: Elaboración propia.

La cantidad de tokens generados por dicho proceso son los que se muestran en la figura 7.

Figura 7. Tokens.



Fuente: elaboración propia.

3. CONCLUSIONES Y RECOMENDACIONES

Se logró elaborar y plantear el objetivo propuesto de la creación de un programa para la comparación de los textos. En él se abordaron los principales aspectos para su elaboración e implementación. Y aunque esta delimitado a textos planos, este programa representa una herramienta que apoya en la determinación de la similaridad. Como trabajo futuro se pretender que no esté limitado a textos planos.

4. REFERENCIAS

[1] Lectures on government and binding, Dordrecht: Foris. christiansen, morten H. y nick chater, 1999: "connectionist natural language processing: the state of the art", Cognitive Science 23, 417-437.

[2] Mayo, Matthew. "Comenzando Con Spacy Para Procesamiento de Lenguaje Natural." Medium, Ciencia y Datos,9 May 2018, medium.com/datos-y-ciencia/comenzando-con-spacy-para-procesamiento-de-lenguaje-natural-e8cf24a18a5a.

[3] Necronet. "Trabajar NLP Basico Con Spacy." Necronet, 16 July 2020, necronet.github.io/Spacy-getting-started-in-spanish/.