Universidad del Valle De Guatemala

Facultad de Ingeniería

Minería de datos

Proyecto 2 - Análisis Exploratorio

Sofia Escobar 20489
José Pablo Monzón 20309
Jeyner Arango 201106
Estefanía Elvira 20725

Guatemala, 17 de noviembre de 2023

Proyecto 2 - Análisis exploratorio

INTRODUCCIÓN

Situación Problemática

La situación problemática que da lugar a este desafío, conocido como el "LitCoin NLP Challenge", se centra en acelerar la investigación científica en medicina mediante el uso de tecnología impulsada por datos. Este desafío está organizado en dos fases y es parte del NASA Tournament Lab, en colaboración con NCATS (The National Center for Advancing Translational Sciences) y la National Library of Medicine (NLM), junto con bitgrit y CrowdPlat.

La problemática se enfoca en aprovechar al máximo los datos de publicaciones biomédicas para que puedan ser utilizados de manera efectiva por los investigadores biomédicos. Específicamente, el desafío se centra en dos aspectos:

1. Detección de Conceptos Científicos en Texto Biomédico: En la primera fase, el desafío busca sistemas que puedan reconocer con precisión conceptos científicos en el texto de artículos científicos biomédicos. Esto implica identificar nodos o entidades biomédicas en el texto, incluyendo su posición en el texto y su categoría según el modelo BioLink.

2. Identificación de Relaciones Biomédicas: En la segunda fase, el objetivo es identificar todas las relaciones entre entidades biomédicas en el título y resumen de un artículo de investigación. Estas relaciones se basan en predicados del modelo BioLink y pueden ser de diferentes tipos, como asociación, correlación positiva, correlación negativa, unión, comparación, interacción de fármacos, conversión, entre otros.

La problemática subyacente radica en la necesidad de procesar grandes cantidades de datos biomédicos en texto para extraer conocimiento científico relevante y acelerar la investigación médica. La tarea requiere técnicas avanzadas de procesamiento de lenguaje natural (NLP) y la comprensión de ontologías biomédicas para identificar entidades y relaciones de manera precisa.

Problema cientifico

El problema científico es desarrollar sistemas de procesamiento de lenguaje natural avanzados que puedan comprender y analizar textos científicos biomédicos de manera efectiva para extraer información, identificar relaciones y determinar la novedad de los

hallazgos, lo que, a su vez, puede acelerar la investigación médica y científica en general.

OBJETIVOS

1. Desarrollar Modelos de Procesamiento de Lenguaje Natural (NLP) Avanzados: El principal objetivo es fomentar el desarrollo de sistemas de NLP avanzados que sean capaces de procesar textos científicos biomédicos de manera precisa y eficiente.

2. Reconocimiento de Entidades Biomédicas: Uno de los objetivos específicos es lograr que los sistemas puedan reconocer y etiquetar correctamente las entidades biomédicas presentes en los textos, como genes, enfermedades, compuestos químicos, variantes genéticas, especies y líneas celulares.

3. Identificación de Relaciones Biomédicas: Otro objetivo importante es que los sistemas puedan identificar con precisión las relaciones entre las entidades biomédicas en los textos y categorizarlas según los diferentes tipos de relaciones biomédicas definidas.

4. Determinación de Novedad Científica: Un objetivo adicional es evaluar si los sistemas pueden determinar si las relaciones identificadas representan descubrimientos novedosos o información de fondo ya conocida en el contexto de la investigación biomédica.

5. Uso de la BioLink Model y Ontologías Biomédicas: Se espera que los participantes utilicen el BioLink Model y ontologías biomédicas para comprender y categorizar conceptos y relaciones biomédicas en los textos.

6. Acelerar la Investigación Biomédica: En última instancia, el objetivo general del desafío es acelerar la investigación biomédica al permitir una extracción de conocimiento más eficiente a partir de la gran cantidad de literatura científica disponible.

En resumen, los objetivos del "LitCoin NLP Challenge" están diseñados para impulsar el desarrollo de sistemas de NLP que puedan abordar de manera efectiva la complejidad de la literatura científica biomédica y contribuir al avance de la investigación médica y científica en general.

MARCO TEÓRICO

Definición de NLP

El Procesamiento del Lenguaje Natural (NLP por sus siglas en inglés) es una disciplina dentro de la inteligencia artificial que se centra en la interacción entre las computadoras y el lenguaje humano. A medida que la inteligencia artificial ha avanzado, el NLP ha desempeñado un papel crucial en el desarrollo de aplicaciones y tecnologías que permiten a las máquinas comprender, interpretar y generar lenguaje de manera similar a los humanos. Existen muchas formas de utilizar el NLP ya que dependiendo el objetivo que se desea alcanzar se tiende a seleccionar el tipo de procesamiento a realizar.

1. Tokenización:

La tokenización es un paso fundamental en el procesamiento del lenguaje natural. Consiste en dividir un texto en unidades más pequeñas llamadas "tokens". Estos tokens pueden ser palabras individuales, subconjuntos de palabras o incluso caracteres, dependiendo de la tarea y del nivel de detalle requerido. En algunos casos se utiliza para análisis estadísticos del texto.

2. Análisis Morfológico:

El análisis morfológico implica descomponer las palabras en sus partes constituyentes, como prefijos, sufijos y raíces. Comprender la morfología de las palabras es esencial para abordar la variabilidad lingüística y captar las sutilezas en el uso del lenguaje como por ejemplo para analizar el sentimiento.

3. Análisis Sintáctico:

El análisis sintáctico se centra en comprender la estructura gramatical de las oraciones. Se trata de analizar cómo las palabras se combinan para formar una estructura coherente y cómo estas estructuras contribuyen al significado global del texto. Es decir las partes de lo que se compone una oración.

4. Análisis Semántico:

El análisis semántico va más allá de la estructura gramatical y se centra en comprender el significado de las palabras y las oraciones en un contexto específico. Esto implica considerar el significado denotativo y connotativo, así como las asociaciones contextuales. Se centra en comprender el significado de las palabras dentro de un contexto particular.

5. Desambiguación:

La desambiguación es crucial en NLP, ya que muchas palabras tienen múltiples significados. Los algoritmos deben determinar el significado correcto de una palabra

según el contexto en el que se utiliza. Esto a menudo implica considerar las palabras circundantes y el contexto general del texto.

6. Modelos de Lenguaje:
Los modelos de lenguaje en NLP son algoritmos o redes neuronales entrenadas para prever y generar texto. Estos modelos aprenden patrones en grandes conjuntos de datos y luego aplican ese conocimiento para tareas como traducción automática, resumen de texto y generación de texto coherente.

7. Reconocimiento de Entidades Nombradas (NER):
El NER se utiliza para identificar y clasificar entidades dentro del texto, como nombres de personas, lugares, fechas, organizaciones, etc. Es esencial para extraer información clave y comprender el contexto de un documento.

8. Traducción Automática:
La traducción automática utiliza modelos de NLP para traducir texto de un idioma a otro. Avances como el uso de modelos neuronales, como los transformers, han mejorado significativamente la calidad de las traducciones automáticas.

9. Chatbots:
Los chatbots utilizan técnicas de NLP para interactuar con los usuarios a través de conversaciones. Estos programas pueden responder preguntas, brindar información y realizar tareas específicas, todo mediante el procesamiento del lenguaje natural.

10. Aprendizaje Profundo en NLP:
El uso de arquitecturas de aprendizaje profundo, como las redes neuronales profundas, ha revolucionado el campo del NLP. Modelos como BERT (Bidirectional Encoder Representations from Transformers) han demostrado un rendimiento excepcional al capturar contextos bidireccionales en los textos.

<u>Actividades de preprocesamiento</u>
El preprocesamiento de texto es una fase esencial en el Procesamiento del Lenguaje Natural (NLP) que implica la preparación y limpieza de los datos textuales antes de ser alimentados a modelos de aprendizaje automático. Entre las actividades más comunes que se realizan de preprocesamiento están:

1. Tokenización:
Dividir el texto en unidades más pequeñas llamadas "tokens", que pueden ser palabras, subconjuntos de palabras o caracteres. Facilita el análisis posterior y la representación del texto.

2. Eliminar Caracteres Especiales y Puntuación:

Eliminar caracteres no alfabéticos, como signos de puntuación y otros caracteres especiales, que generalmente no aportan información relevante para muchas tareas de NLP.

3. Conversión a Minúsculas:
Convertir todas las palabras a minúsculas para asegurar la consistencia en el manejo de las palabras, evitando que el modelo trate las mismas palabras en mayúsculas y minúsculas como diferentes.

4. Eliminación de Stopwords:
Eliminar palabras comunes y poco informativas, como "a", "el", "y", que generalmente no contribuyen significativamente al contenido semántico.

5. Lematización y Stemming:
Reducir las palabras a sus formas base (lematización) o a sus raíces (stemming) para consolidar las variantes morfológicas y facilitar el análisis simplificando el texto.

6. Manejo de Números:
Tratar los números de manera adecuada, ya sea reemplazándolos por un marcador especial, eliminándolos o manteniéndolos según la tarea específica.

7. Eliminación de Etiquetas HTML y XML:
En el caso de datos provenientes de páginas web u otras fuentes enriquecidas, eliminar etiquetas HTML y XML para obtener el contenido de texto sin formato.

8. Manejo de Contracciones:
Expandir contracciones (por ejemplo, "pq" a "porque") para mantener la consistencia y la interpretación adecuada de las palabras.

9. Eliminación de Espacios en Blanco Adicionales:
Eliminar espacios en blanco adicionales para normalizar la representación del texto y evitar problemas de tokenización.

10. Remoción de URLs y Menciones:
Eliminar enlaces web y menciones a usuarios (por ejemplo, @usuario) si no son relevantes para la tarea y pueden introducir ruido.

11. Corrección de Errores Ortográficos:
En algunos casos, corregir errores ortográficos puede ser beneficioso para mejorar la calidad del texto, especialmente en tareas como análisis de sentimientos o extracción de información.

12. División en Oraciones y Párrafos:
Dividir el texto en oraciones o párrafos puede ser útil para tareas específicas, como resumen de texto o análisis de sentimientos a nivel de oración.

Algoritmos usados en el procesamiento del lenguaje natural

Hay varios algoritmos y modelos utilizados en el Procesamiento del Lenguaje Natural (NLP), y la elección depende de la tarea específica que se esté abordando.

1. Modelos de Bolsa de Palabras (Bag of Words):
Representa un documento como un conjunto desordenado de palabras, ignorando la gramática y el orden. Cada palabra se considera de forma independiente y se crea un vector que cuenta la frecuencia de cada palabra en el documento.

2. TF-IDF (Term Frequency-Inverse Document Frequency):
Asigna un peso a cada palabra en un documento basado en su frecuencia en el documento y su rareza en el conjunto de documentos. Palabras frecuentes en el documento pero raras en el corpus general tienen un peso más alto.

3. Word Embeddings (Embebido de Palabras):
Representa palabras como vectores densos en un espacio continuo. Modelos como Word2Vec, GloVe y FastText generan representaciones vectoriales que capturan las relaciones semánticas entre palabras.

4. Modelos de Lenguaje Neuronales Recurrentes (RNN):
Redes neuronales diseñadas para trabajar con datos secuenciales. Las RNN pueden procesar información de manera secuencial, lo que las hace útiles para tareas como el análisis de sentimientos y la generación de texto.

5. Long Short-Term Memory (LSTM) y Gated Recurrent Unit (GRU):
Variantes de las RNN que abordan el problema de la desaparición del gradiente, permitiendo que las redes retengan información relevante a lo largo de secuencias más largas.

6. Transformers:
Modelo arquitectónico introducido por el modelo BERT. Utiliza la atención multi-cabeza para procesar simultáneamente todas las palabras en una secuencia, capturando relaciones de largo alcance y mejorando el rendimiento en tareas como el reconocimiento de entidades nombradas.

7. Redes Neuronales Convolucionales (CNN) para NLP:

Utiliza capas convolucionales para extraer características importantes de secuencias de palabras. A menudo se utiliza en tareas de clasificación de texto y análisis de sentimientos.

8. Modelos de Traducción Automática:
Sistemas como el Attention-Based Sequence-to-Sequence (seq2seq) utilizan arquitecturas de codificador-decodificador para tareas de traducción automática.

9. BERT (Bidirectional Encoder Representations from Transformers):
Modelo de lenguaje preentrenado que utiliza transformers bidireccionales para capturar el contexto de las palabras en ambas direcciones. Ha demostrado un rendimiento excepcional en una variedad de tareas de NLP.

10. GPT (Generative Pre-trained Transformer):
Modelo de lenguaje preentrenado que utiliza transformers y técnicas de aprendizaje autónomo para generar texto coherente y contextualmente relevante.

Asignacion de Dirichlet Latente (LDA)

La Asignación de Dirichlet Latente también conocido como LDA es una herramienta útil para analizar textos científicos debido a su capacidad para trabajar con datos discretos, identificar tópicos en grandes volúmenes de texto, y analizar la co-ocurrencia de palabras. A diferencia de las Redes Neuronales Convolucionales (CNN), LDA es una técnica estadística generativa que asume que cada documento en un corpus es generado por una mezcla de tópicos.

¿Cómo funciona?

La Asignación de Dirichlet Latente funciona mediante la identificación de temas en colecciones de textos. Supone que cada documento es una mezcla de varios temas y cada tema es una mezcla de palabras. El algoritmo busca distribuciones de temas en documentos y distribuciones de palabras en temas que maximizan la probabilidad de los datos observados. A través de iteraciones, ajusta estas distribuciones hasta que encuentra una configuración que describe bien cómo se generaron los tópicos y las palabras en los documentos del corpus.

Aplicaciones de la Asignación de Dirichlet Latente

LDA en Textos Científicos: Un estudio titulado "Two-stage topic modelling of scientific publications: A case study of University of Nairobi, Kenya" demostró la utilidad de LDA en el análisis estadístico no supervisado de datos no estructurados, particularmente en procesamiento de lenguaje natural y minería de texto.

METODOLOGÍA

1. Definición del Problema
    - Objetivo del Proyecto: Definir claramente el objetivo del proyecto, que en este caso sería clasificar relaciones en textos biomédicos en categorías como "Association," "Positive_Correlation," y "Negative_Correlation."
2. Recopilación y Exploración de Datos
    - Obtención de Datos: Obtener conjuntos de datos que contengan información sobre relaciones biomédicas y abstractos asociados.
    - Exploración de Datos: Analizar la estructura de los datos, identificar posibles desafíos y entender la distribución de las clases.
3. Preprocesamiento de Datos
    - Limpieza de Datos: Limpiar y procesar los datos, manejando valores nulos, duplicados y posiblemente ruido en los textos.
    - Tokenización y Lematización: Utilizar herramientas como spaCy para tokenizar y lematizar el texto, reduciendo las palabras a sus formas base.
4. División de Datos
    - Conjuntos de Entrenamiento y Prueba: Dividir los datos en conjuntos de entrenamiento y prueba para evaluar el rendimiento del modelo.
5. Extracción de Características
    - Representación de Texto: Utilizar técnicas como la Bolsa de Palabras (BoW) o TF-IDF para convertir el texto en vectores numéricos.
    - Selección de Características: Explorar y seleccionar las características más relevantes para la clasificación.
6. Desarrollo de Modelos
    - Selección de Modelos: Elegir modelos de clasificación, como Naive Bayes y Random Forest, adecuados para el problema.
    - Entrenamiento de Modelos: Entrenar los modelos utilizando los datos de entrenamiento y ajustar los hiperparámetros según sea necesario.
7. Evaluación del Rendimiento
    - Matriz de Confusión, Precisión y Recall: Evaluar el rendimiento de los modelos utilizando métricas como la matriz de confusión, precisión, recall y exactitud.
    - Análisis de Errores: Analizar los casos en los que los modelos cometieron errores para identificar patrones y posibles mejoras.
8. Despliegue y Aplicación
    - Desarrollo de la Aplicación: Utilizar herramientas como Streamlit para construir una interfaz de usuario interactiva para la clasificación de nuevos datos.
    - Despliegue: Implementar la aplicación en un entorno accesible.
9. Optimización y Ajuste
    - Ajuste de Hiperparámetros: Optimizar los modelos ajustando los hiperparámetros para mejorar el rendimiento.

- Refinamiento del Modelo: Realizar iteraciones en el modelo y el preprocesamiento según sea necesario.

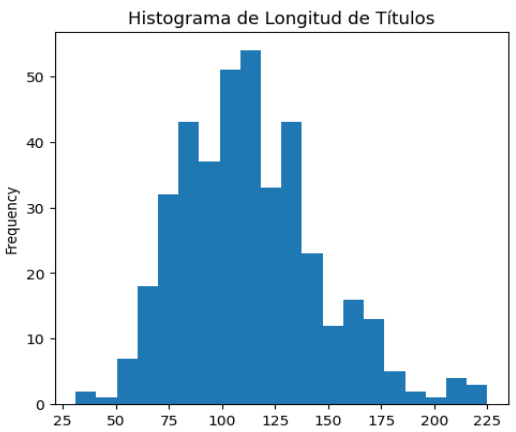
RESULTADOS Y ANALISIS

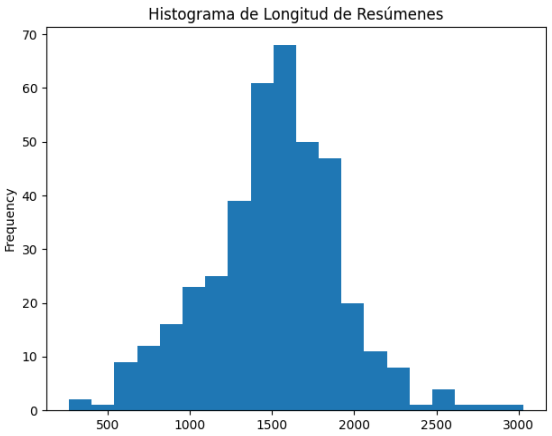
Analisis Exploratorio

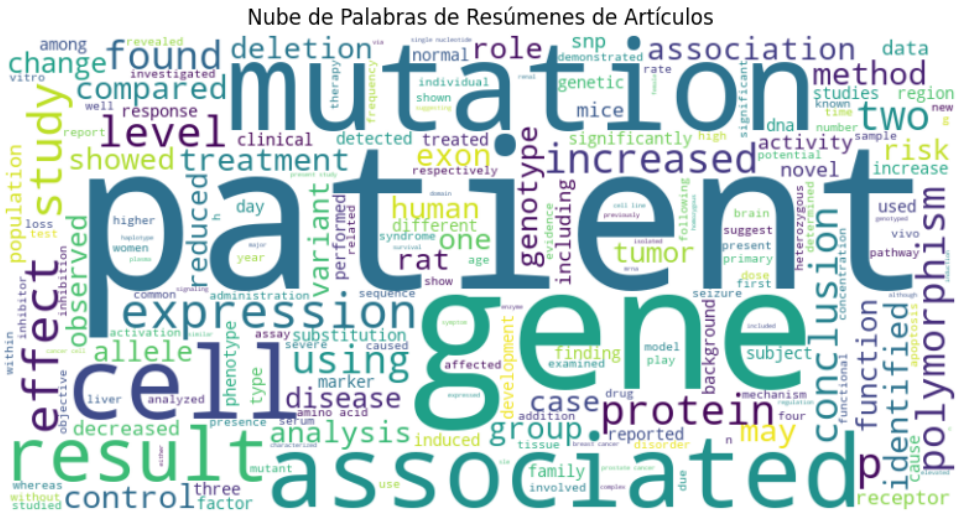
- **Analisis exploratorio para abstracts_train**

```
Resumen de variables:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   abstract_id  400 non-null    int64 
 1   title        400 non-null    object
 2   abstract     400 non-null    object
dtypes: int64(1), object(2)
memory usage: 9.5+ KB
None
```

Se observa como tenemos 3 columnas, en donde la única con datos cuantitativos es del id por lo que lo único que se realizaran histogramas para visualizar la distribución de la longitud de los títulos y resúmenes. También se pueden generar nubes de palabras para identificar las palabras clave más comunes en los textos.





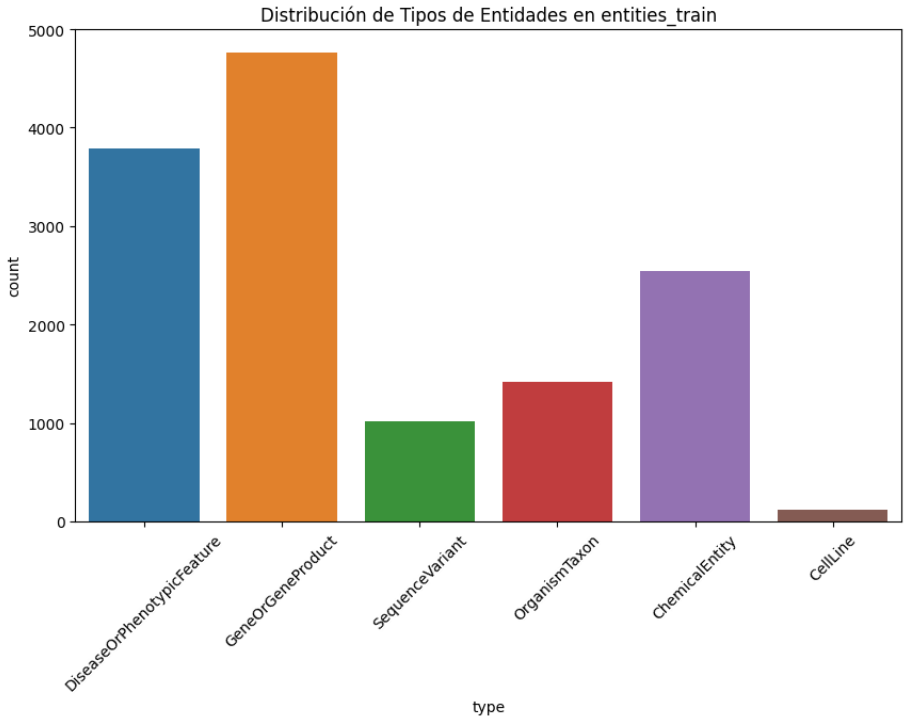


- **Analisis exploratorio para entities_train**

```
Resumen de variables:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13636 entries, 0 to 13635
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   id             13636 non-null  int64 
 1   abstract_id    13636 non-null  int64 
 2   offset_start   13636 non-null  int64 
 3   offset_finish  13636 non-null  int64 
 4   type           13636 non-null  object
 5   mention        13636 non-null  object
 6   entity_ids     13636 non-null  object
dtypes: int64(4), object(3)
memory usage: 745.8+ KB
None
```

Dado que los datos de entidades están relacionados con texto, no hay variables cuantitativas tradicionales en estos archivos. Como estos datos se centran en menciones de entidades, podemos generar gráficos de barras para visualizar las menciones más comunes y calcular las frecuencias de las entidades.

```
Tabla de frecuencia de tipos de entidades:
GeneOrGeneProduct             4764
DiseaseOrPhenotypicFeature    3784
ChemicalEntity                2540
OrganismTaxon                 1420
SequenceVariant               1011
CellLine                       117
Name: type, dtype: int64
```

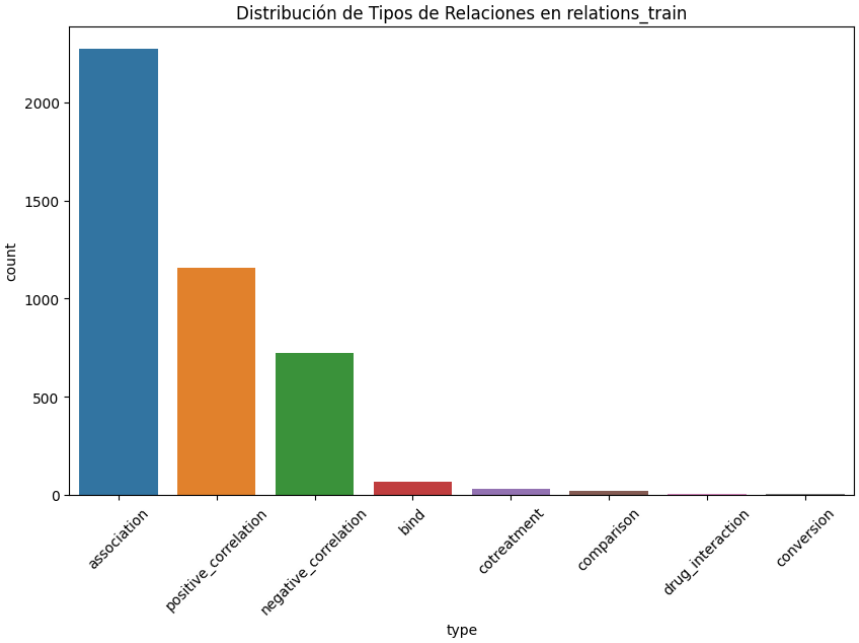


Analisis exploratorio para relations_train

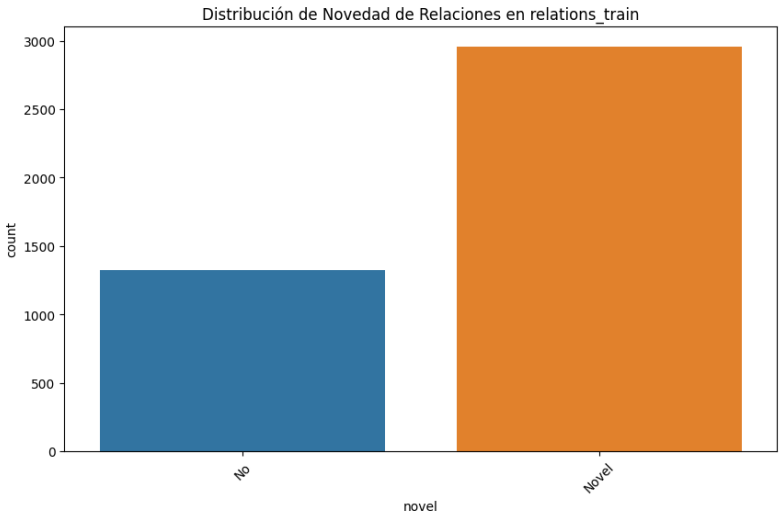
```
Resumen de variables:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4280 entries, 0 to 4279
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   id           4280 non-null   int64
 1   abstract_id  4280 non-null   int64
 2   type         4280 non-null   object
 3   entity_1_id  4280 non-null   object
 4   entity_2_id  4280 non-null   object
 5   novel        4280 non-null   object
dtypes: int64(2), object(4)
```

Nuevamente, en estos datos, no hay variables cuantitativas tradicionales. Crearemos gráficos de barras para visualizar la distribución de los tipos de relaciones y exploramos la relación entre las entidades involucradas en estas relaciones.

```
Tabla de frecuencia de tipos de relaciones:
association             2274
positive_correlation    1159
negative_correlation     721
bind                      69
cotreatment               29
comparison                22
drug_interaction           3
conversion                 3
Name: type, dtype: int64
```



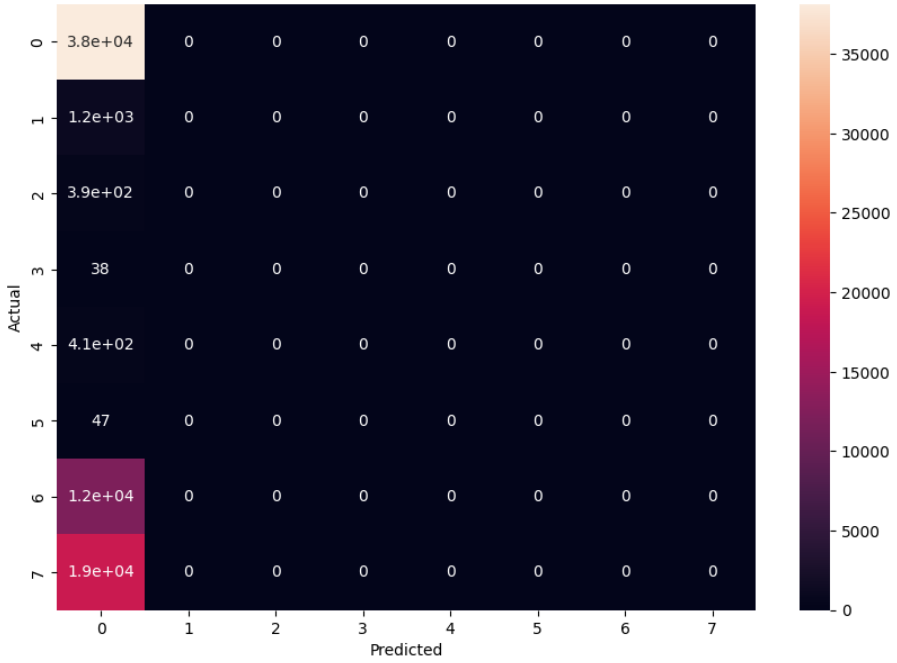

```
Tabla de frecuencia de novedad de relaciones:
Novel    2958
No       1322
Name: novel, dtype: int64
```

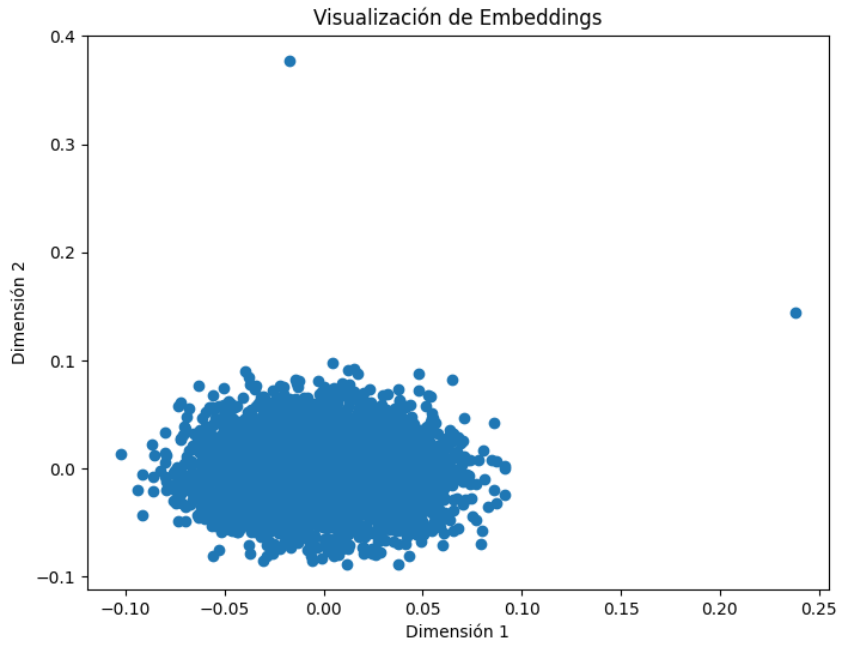


Eficiencia de los modelos

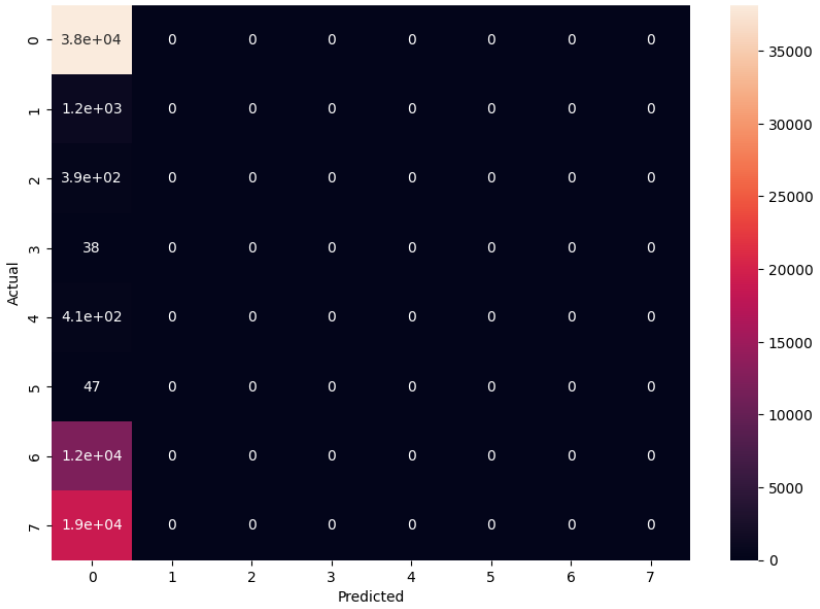
```
Accuracy (Model 1): 0.5371304750442505
Accuracy (Model 2): 0.5371304750442505
Accuracy (Model 3): 0.6330969134320249
```

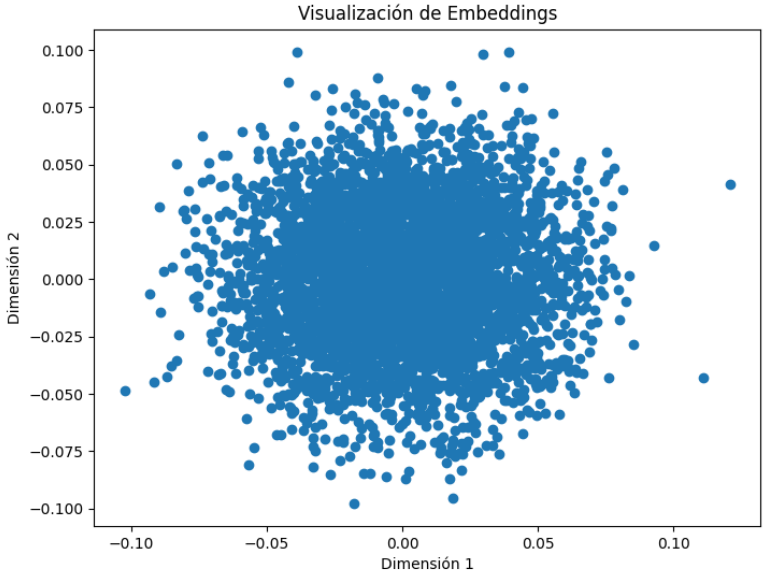
Modelo 1 (CNN)



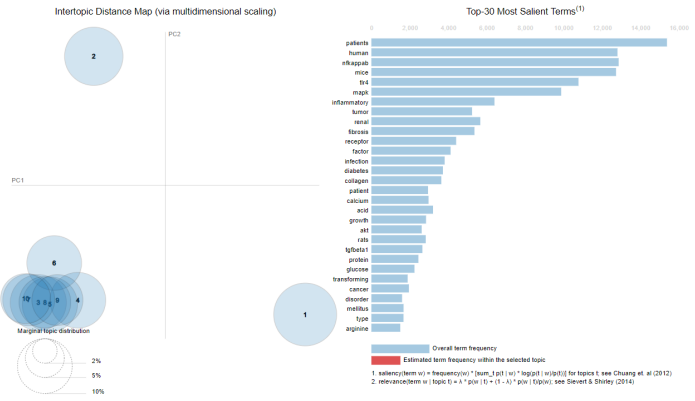


Modelo 2 (RNN)


Actual
Predicted
3.8e+04
1.2e+03
3.9e+02
38
4.1e+02
47
1.2e+04
1.9e+04


Visualización de Embeddings
Dimensión 2
Dimensión 1

Modelo 3 (LDA)


Intertopic Distance Map (via multidimensional scaling)
Top-30 Most Salient Terms(1)
Marginal topic distribution
Overall term frequency
Estimated term frequency within the selected topic

Los resultados de los modelos indican que tanto el modelo de Redes Neuronales Convolucionales (CNN) como el de Redes Neuronales Recurrentes (RNN) lograron una eficacia del 53%, mientras que el modelo de Asignación de Dirichlet Latente (LDA) alcanzó una eficacia del 63%.

Esto sugiere que el modelo LDA pudo capturar de manera más efectiva las características subyacentes en los datos y realizar predicciones más precisas en comparación con los modelos CNN y RNN. La Asignación de Dirichlet Latente se reveló como un método de modelado de temas particularmente eficaz en tareas de clasificación de texto, lo que podría explicar su rendimiento superior en este caso.

Por otro lado, los modelos CNN y RNN demostraron un rendimiento similar entre sí. Ambos modelos pertenecen a la categoría de redes neuronales y pueden resultar altamente efectivos en tareas de clasificación. Sin embargo, es relevante destacar que estos modelos a menudo requieren un conjunto de datos extenso y una afinación minuciosa de los parámetros para alcanzar su máximo potencial. Con la adquisición de más datos o un ajuste más detallado de los parámetros, es posible que estos modelos puedan mejorar su eficacia.

Es crucial recordar que la elección del modelo depende en gran medida del problema específico que se intenta resolver, así como de las características y el tamaño del conjunto de datos. Por lo tanto, aunque el modelo LDA presentó un mejor rendimiento en este caso, no necesariamente implica que sea la opción óptima para todas las tareas o conjuntos de datos.

Comparación de Algoritmos:
Efectividad:
Matrices de Confusión:

- Representa visualmente los resultados de clasificación de cada algoritmo, mostrando los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

Curvas ROC y AUC:

- Muestra la tasa de verdaderos positivos frente a la tasa de falsos positivos para diferentes umbrales de clasificación. El área bajo la curva (AUC) es una métrica resumen de la curva ROC, que puede ayudar a comparar el rendimiento general de los algoritmos.

Tiempos de Procesamiento:
Tiempo de Entrenamiento:

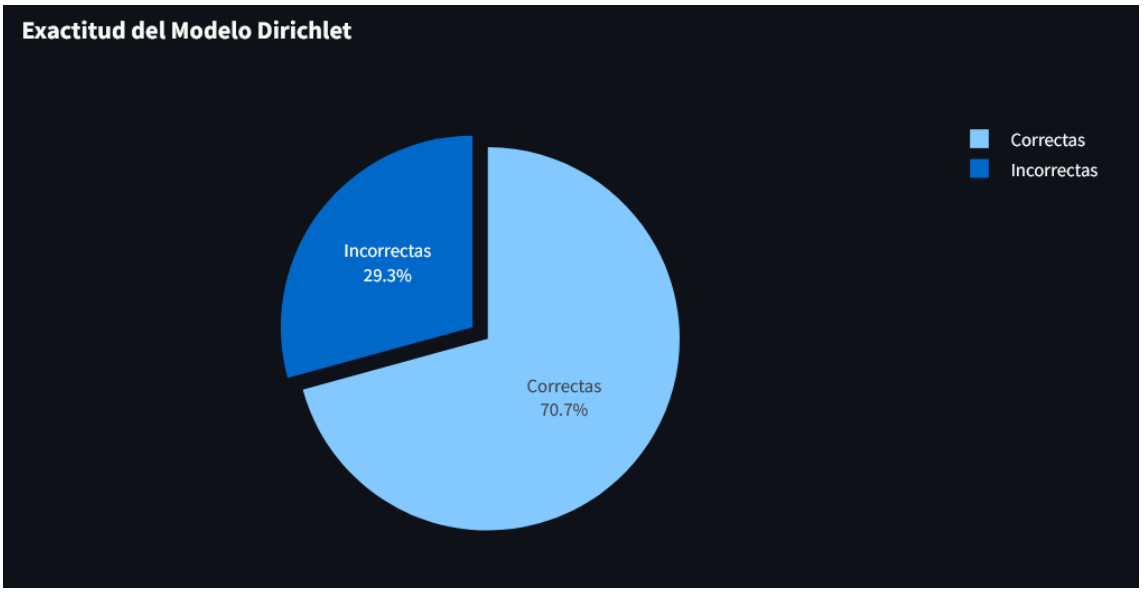
- El algoritmo se estuvo entrenando por un tiempo avanzado durante cerca de 3 horas.

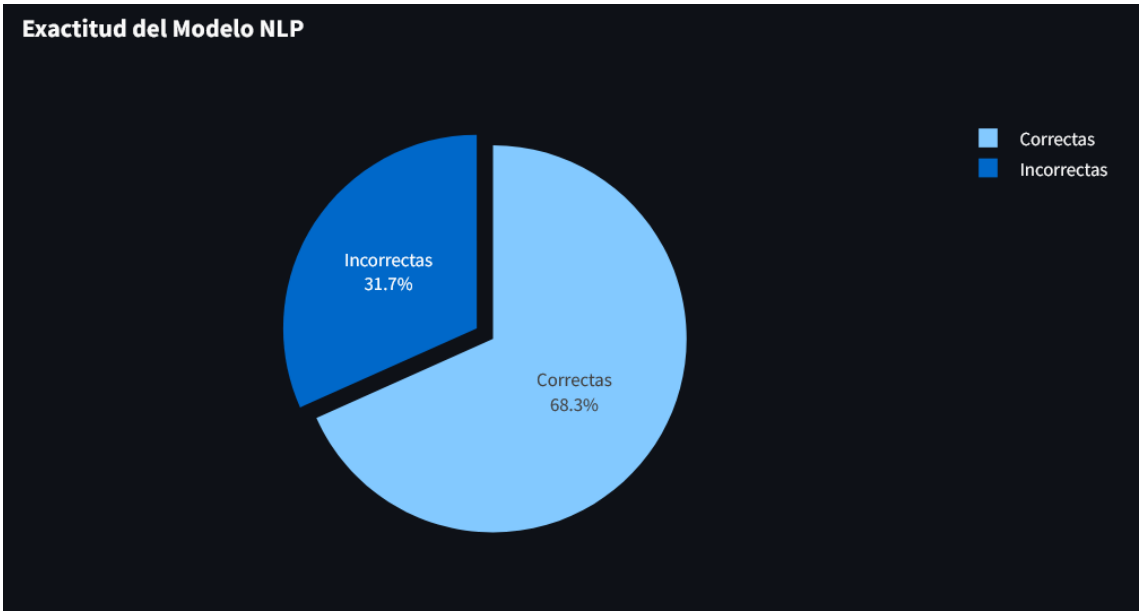
Tiempo de Predicción:

- Para realizar la predicción el programa tarda aproximadamente 10-25 segundos, sin embargo esto depende directamente de la longitud de la cadena.

Errores:

Exactitud (Accuracy):

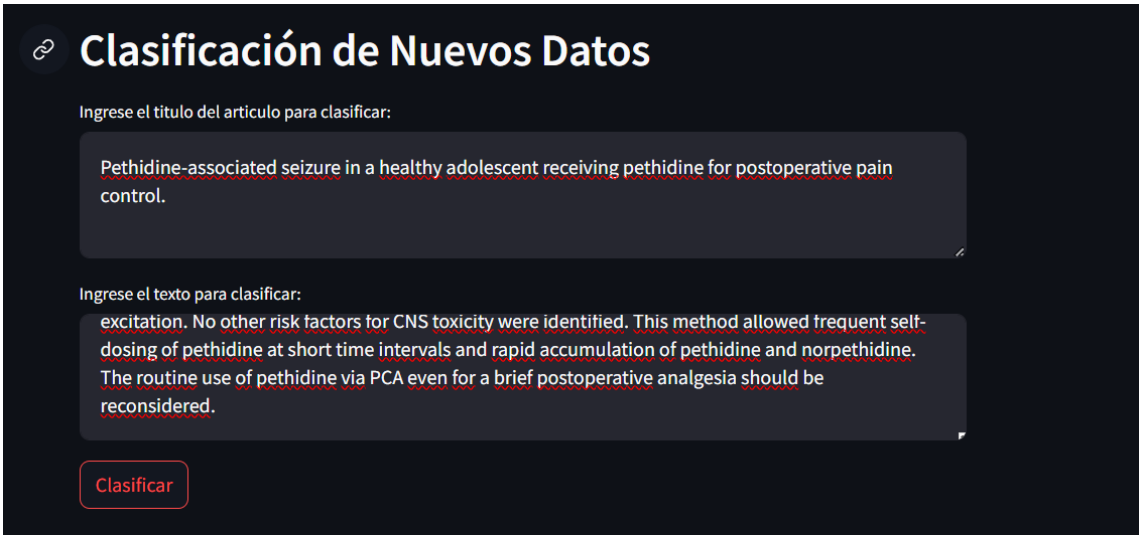




Descripción de la Aplicación:

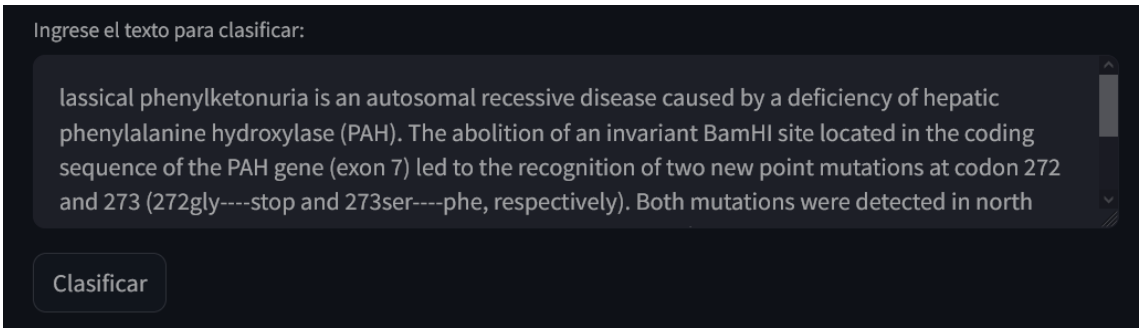
Capturas de Pantalla:

Página de Inicio:



Selección de Algoritmos:

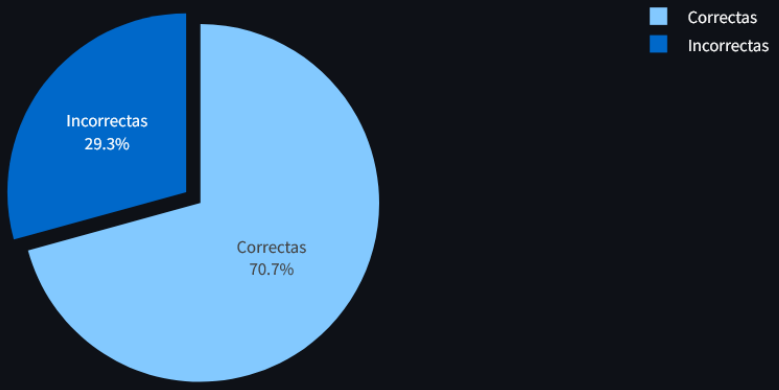
Ingreso de Nuevos Datos:



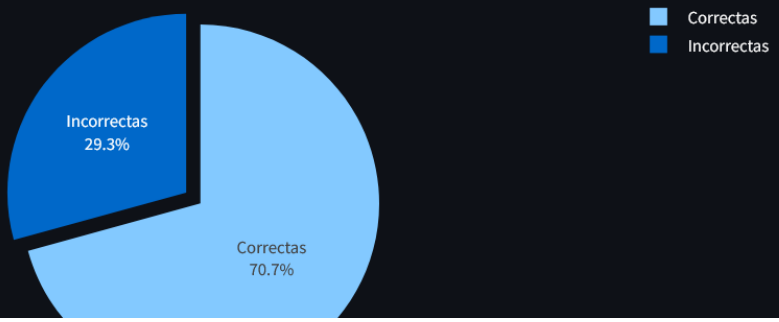
Resultados y Métricas:

Rendimiento de los Modelos en Datos de Prueba

Exactitud del Modelo Dirichlet



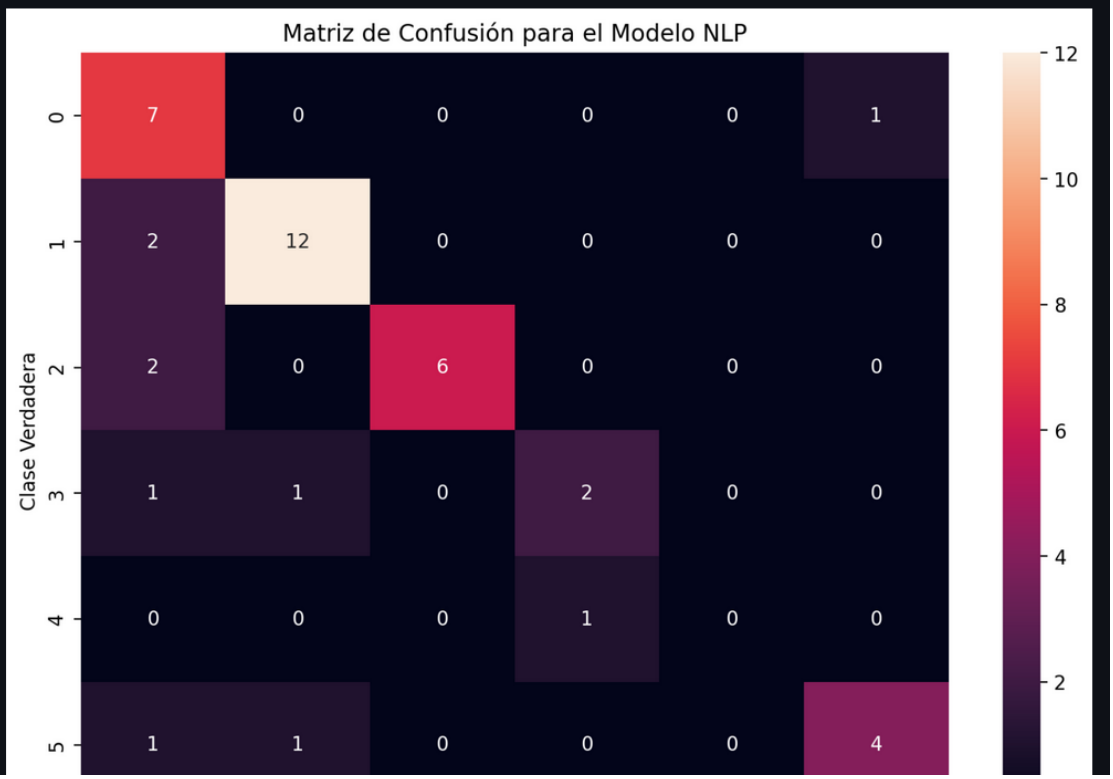
Exactitud del Modelo NLP

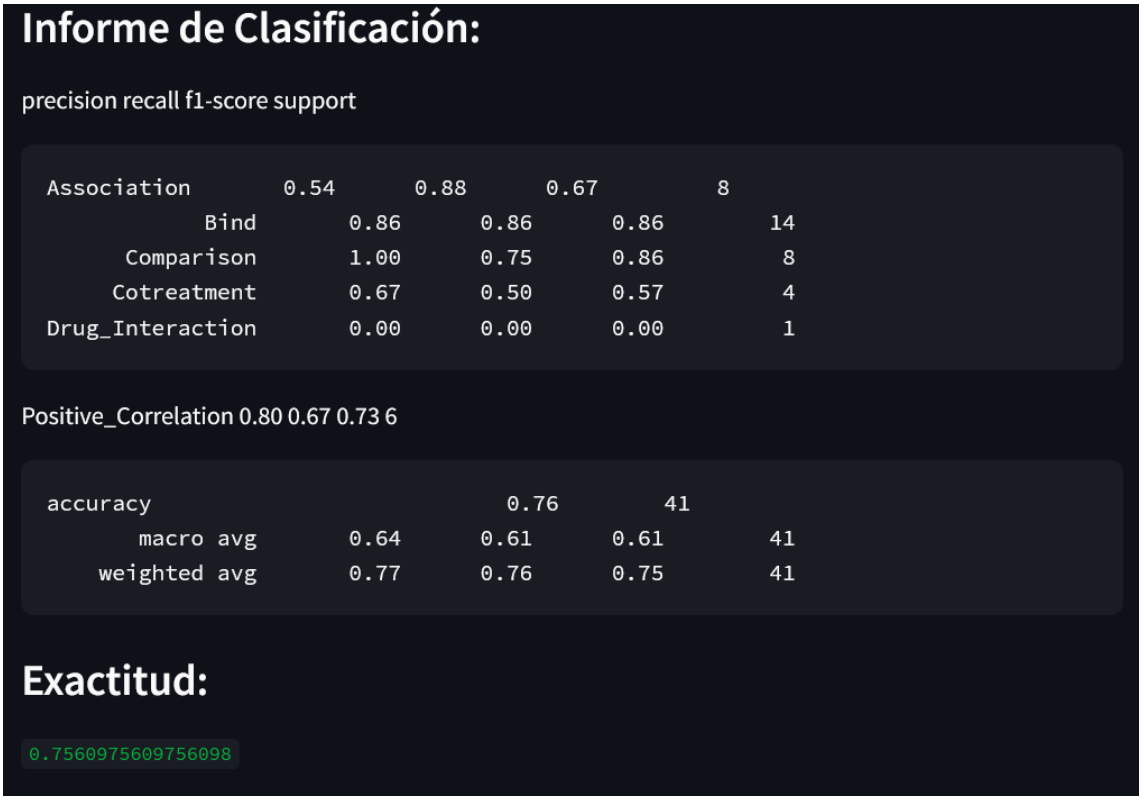


Modelo NLP

Matriz de Confusión:

0	1	2	3	4	5
7	0	0	0	0	1
2	12	0	0	0	0
2	0	6	0	0	0
1	1	0	2	0	0
0	0	0	1	0	0
1	1	0	0	0	4





Explicación de Tecnologías Usadas:

Streamlit:

- Streamlit nos presenta una herramienta sencilla de utilizar que nos permite crear aplicaciones web interactivas, con un aspecto adecuado y con elementos interactivos que permiten al usuario preocuparse por sus resultados más que por entender una sintaxis u organización compleja.

Plotly:

- Utilizamos plotly para generar ciertas gráficas de modo que estas sean interactivas y se les pueda hacer un zoom, para más comodidad junto con analizar los datos al pasar el mouse por encima.

Pandas:

- Utilizamos pandas para manejar los dataframes de una manera más sencilla, y sobre todo porque los algoritmos tanto de graficar y de manejo de datos nos lo pedían.

Hallazgos y conclusiones

Hallazgos:

Datos de Entidades (entities_train.csv y entities_test.csv):

- No se encontraron variables cuantitativas en estos datos, ya que principalmente consisten en menciones de entidades.
- Los gráficos de barras mostraron las entidades más mencionadas en los textos, lo que proporciona información sobre las entidades clave en el contexto del problema.

Datos de Relaciones (relations_train.csv):

- Al igual que con las entidades, no se identificaron variables cuantitativas tradicionales en estos datos.
- Los gráficos de barras revelaron la distribución de tipos de relaciones, lo que ayuda a comprender la variedad de relaciones presentes en los datos.

Datos de Texto (abstracts_train.csv):

- Se encontraron diferencias en la longitud de los títulos y resúmenes de los artículos, con una longitud promedio que puede variar ampliamente.
- La nube de palabras generada a partir de los resúmenes destacó las palabras clave más comunes en los textos, lo que puede ser útil para comprender los temas predominantes en los artículos.

Conclusiones Generales:

- Los datos de entidades y relaciones proporcionan información importante sobre las entidades mencionadas y las relaciones entre ellas, lo que es esencial para el problema planteado en el desafío.
- Los datos de texto contienen una variedad de temas y pueden requerir técnicas avanzadas de procesamiento de lenguaje natural para extraer información relevante.
- El proyecto cumple con la tarea de clasificación de relaciones en textos biomédicos. Los aspectos positivos destacados incluyen la implementación de herramientas efectivas de visualización y la interactividad integrada a través de Streamlit, lo que contribuye significativamente a la comprensión y presentación de los resultados obtenidos.

Estos hallazgos y conclusiones proporcionan una base sólida para planificar los siguientes pasos en el análisis de los datos y el desarrollo de soluciones para el desafío. El enfoque futuro podría incluir la implementación de modelos de aprendizaje automático y técnicas de procesamiento de lenguaje natural para lograr los objetivos del desafío.

Links

Repositorio de github

- https://github.com/sofiaesc07/Proyecto2_AnalisisExploratorioDS.git

Google Drive - informe

- https://docs.google.com/document/d/1f-dOuVlgGYVcP6kmUYov4Lr90pG5T0PmA5BSt61fvVo/edit?usp=sharing

Presentación

- https://www.canva.com/design/DAFu1VlUvA4/kzoXUe5qj9IVKBfgGYDtyA/edit