

An Approach to Initial Tokens of Attention Layers for Detecting Author Changes in Multi-authored Texts.

César Espin-Riofrio, MSc.¹, Verónica Mendoza Morán, MSc.¹, Lilia Santos Díaz, MSc.¹, Jenniffer Tenempaguay-Borja, Ing.¹, Jhonn Montenegro-Arellano, Ing.¹, Arturo Montejo-Ráez, PhD.²

¹Universidad de Guayaquil, Ecuador, cesar.espinr@ug.edu.ec, veronica.mendozam@ug.edu.ec, lilia.santosd@ug.edu.ec, jennifer.tenempaguayb@ug.edu.ec, jhonn.montenegroa@ug.edu.ec

²Universidad de Jaén, España, amontejo@ujaen.es

Abstract– Author change detection is crucial in an environment where multiple people have contributed to the same content, being essential to ensure the transparency and originality of a document, benefiting multiple areas such as academic and scientific. The objective of this research is to detect where author change occurs in multi-authored documents, where a model based on the Transformers architecture is proposed using the pre-trained DeBERTa and mDeBERTa models. In the experimental process, we extract embeddings of the initial tokens from the model layers and apply transfer learning to adjust them. We validate our approach using an English text dataset taken from PAN CLEF 2023, evaluating its efficiency and performance. The results show F1-scores of 0.9721 and 0.9647 with DeBERTa and mDeBERTa, respectively, validating that both have high accuracy in detecting author changes in multi-author texts. DeBERTa slightly outperforms mDeBERTa. The proposal demonstrates that embedding extraction of initial tokens from the attention layers and, subsequent fine-tuning in both models, are highly effective for accurate author change detection in multi-authored documents.

Keywords-- Author changes, Natural Language Processing, Transformers, Embeddings of initial tokens.

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).
DO NOT REMOVE

Un Enfoque a los Tokens Iniciales de las Capas de Atención para Detectar Cambios de Autor en Textos Multi-autor

César Espin-Riofrio, MSc.¹, Verónica Mendoza Morán, MSc.¹, Lilia Santos Díaz, MSc.¹, Jenniffer Tenempaguay-Borja, Ing.¹, Jhonn Montenegro-Arellano, Ing.¹, Arturo Montejo-Ráez, PhD.²

¹Universidad de Guayaquil, Ecuador, cesar.espinr@ug.edu.ec, veronica.mendozam@ug.edu.ec, lilia.santosd@ug.edu.ec, jennifer.tenempaguayb@ug.edu.ec, jhonn.montenegroa@ug.edu.ec

²Universidad de Jaén, España, amontejo@ujaen.es

Resumen— La detección de cambios de autor es crucial en un entorno donde múltiples personas han aportado al mismo contenido, siendo esencial para garantizar la transparencia y originalidad de un documento, beneficiando a múltiples áreas como la académica y científica. El objetivo de esta investigación es detectar dónde se produce el cambio de autor en documentos multi-autor, donde se propone un modelo basado en la arquitectura Transformers utilizando los modelos pre-entrenados DeBERTa y mDeBERTa. En el proceso experimental, extraemos los embeddings de los tokens iniciales de las capas del modelo y aplicamos aprendizaje por transferencia para ajustarlos. Validamos nuestro enfoque utilizando un dataset de textos en inglés tomado de PAN CLEF 2023, evaluando su eficacia y rendimiento. Los resultados muestran F1-scores de 0.9721 y 0.9647 para DeBERTa y mDeBERTa, respectivamente, validando que ambos tienen una alta precisión en la detección de cambios de autor en textos multi-autor. DeBERTa se destaca ligeramente por encima de mDeBERTa. La propuesta demuestra que la extracción de embeddings y el posterior fine-tuning en ambos modelos, son altamente efectivos para la detección precisa de cambios de autor en documentos multi-autor.

Palabras claves— Cambios de Autor, Procesamiento de Lenguaje Natural, Transformers, Embeddings de tokens iniciales.

I. INTRODUCCIÓN

Hoy en día, con la sociedad altamente conectada e informatizada, la colaboración en documentos ha aumentado, generando mayor complejidad por las intervenciones de diferentes autores. Por tanto, es crucial detectar y distinguir cambios de autoría en documentos multi-autor para asegurar transparencia y atribución correcta, especialmente en artículos científicos y académicos. Esto facilita el seguimiento de ideas y responsabilidades, así como la verificación de la veracidad y fiabilidad del trabajo. Para lograr una predicción precisa de la ubicación de estos cambios, se requiere analizar y comprender el contenido textual en detalle, empleando herramientas avanzadas del Procesamiento del Lenguaje Natural (PLN).

El PLN es una rama de la Informática e Inteligencia Artificial que se centra en la capacidad computacional del

lenguaje humano que tiene como objetivo el hacer que las máquinas puedan ser capaces de entender y generar lenguajes humanos [1]. El PLN permite extraer características lingüísticas y analizar el contexto de un texto. De modo que, para la detección de cambios de autor en textos multi-autor, es necesario la comparación entre textos y determinar si se produce un cambio o no. Esto requiere el uso de herramientas del PLN, como la arquitectura Transformers mediante los modelos pre-entrenados basados en BERT: DeBERTa y mDeBERTa, en este estudio.

La arquitectura Transformer fue presentada en el año 2017 con el artículo Attention Is All You Need [2]; principalmente fue creado para el PLN, pero fácilmente fue adoptado a otros campos del Machine Learning (ML). Esta es una red neuronal de aprendizaje profundo, en la que su característica clave, es el uso de los mecanismos de atención, que permite el procesamiento de los datos de forma paralela haciendo el modelo más rápido y eficiente [3]. Los modelos Transformers son usados en [4] para determinar la afinidad política de usuarios de Twitter en Ecuador, o para descubrir perfiles explorando la frecuencia relativa de las palabras [5].

Tras la publicación de la arquitectura Transformers, surgen muchos modelos basados en este mismo, entre uno de ellos: BERT.

El modelo BERT (Bidirectional Encoder Representations from Transformers) [6], es presentado por Google AI en 2018, representando una revolución dentro del PLN. BERT es un modelo de código abierto pre-entrenado de forma bidireccional a gran escala [7], que permite abarcar una amplia variedad de tareas. Presentando el modelo con dos tamaños: BERT-base y BERT-large. El primero, cuenta con 12 capas y el segundo, 24. Otra diferencia es su tamaño oculto (hidden size), las capas de atención y su cantidad de parámetros. Haciendo que BERT-base sea más ligero al requerir una menor cantidad de memoria a comparación de BERT-large. Esta revolución representó la creación de múltiples modelos para la comprensión y generación del lenguaje natural. DeBERTa (Decoding-enhanced BERT with disentangled attention), en la que utiliza el mecanismo de atención desenredada y emplea un decodificador de máscara mejorado [8] donde su preentrenamiento es realizado con datos en inglés. mDeBERTa es una versión multilingüe de DeBERTa que utiliza la misma

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).
DO NOT REMOVE

estructura del modelo original y ha sido entrenada con datos de varios idiomas.

En trabajos anteriores, [9] tuvieron la tarea de detección de cambio de estilo, en la que dentro de sus objetivos fueron determinar si el documento tiene varios autores, averiguar dónde se han producido los cambios de estilos, y etiquetar el identificador de autor para cada párrafo del documento. [10] ofrecen una solución a las tareas compartidas de detección de cambios de estilo, que presenta un desafiante problema de clasificación multi etiqueta y multi salida. [11] menciona que la detección de cambios de estilo tiene como objetivo identificar violaciones en el estilo de escritura, es decir, los puntos en los que se producen cambios en el estilo de escritura y los autores se alternan dentro de un documento con múltiples autores. [12] presentaron un método basado en el modelo de preentrenamiento de Bert y convolución unidimensional para abordar la tarea de detección de cambios de estilo en documentos con múltiples autores. [13] muestran un método BertAA que se basa en el ajuste fino (fine-tuning) de un modelo de lenguaje BERT pre entrenado, en lo que añadieron una capa densa y una activación softmax para la clasificación de la autoría, entrenado durante unas pocas épocas.

[14] explican que los embeddings de los tokens iniciales en el modelo BERT representan una amplia gama de características del texto, que incluyen la ortografía de las palabras, su significado, el contexto sintáctico y semántico, así como el tema general del texto. Estos embeddings son una parte esencial de BERT y son responsables de su éxito en diversas tareas de procesamiento del lenguaje natural.

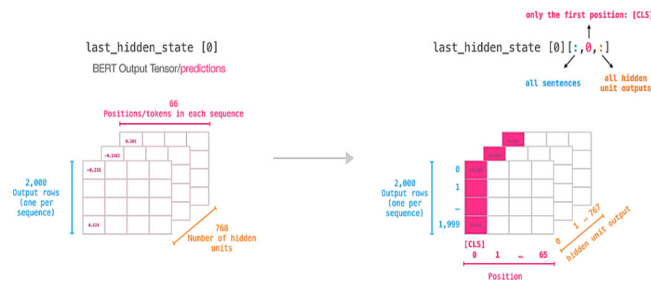


Fig. 1 Salida del último estado oculto

[15] indica que, para extraer capas intermedias de BERT, es necesario considerar dos elementos: la salida agrupada (pooled output) y los últimos estados ocultos (last hidden states). La salida agrupada es el último estado oculto del primer token en la secuencia procesada por la capa lineal y la función de activación. Los últimos estados ocultos son la salida de la secuencia de estados ocultos en las capas del modelo.

En esta investigación, proponemos un enfoque para detectar cambios de autor en textos multi-autor utilizando diferentes modelos donde se realizará fine-tuning para explorar la información de los embeddings extraídos de los tokens iniciales de las capas de atención. Nuestro objetivo principal es analizar la viabilidad y eficacia de este modelo en la detección de cambios de autoría en documentos. Los resultados serán

útiles para evaluar la eficiencia de nuestro enfoque. Además, este estudio podría impactar en la creación de herramientas para detectar plagio en documentos científicos y académicos, preservando la integridad y originalidad en la producción académica y científica. Esperamos contribuir al avance de las técnicas de PLN y aportar nuevos conocimientos aplicables en el análisis de textos multi autores y la identificación precisa de cambios de autoría en documentos.

II. METODOLOGÍA

En el presente trabajo, se ha realizado una revisión bibliográfica documental exhaustiva que abarca una selección de artículos científicos destacados. Esta revisión nos ha permitido obtener una comprensión profunda del estado actual de la investigación en el campo y conocer los métodos utilizados en estudios similares. También, la metodología utilizada en esta investigación es de índole experimental, en el cual se realizaron varias pruebas con los modelos pre-entrenados DeBERTa y mDeBERTa. Así como un enfoque cuantitativo, debido a que, en base a métricas de evaluación se podrá comparar y determinar cuáles de los modelos tiene un mejor desempeño.

A. Modelo propuesto

El modelo propuesto sigue un proceso en varias etapas. En primer lugar, se realiza el procesamiento de datos para prepararlos adecuadamente. Una vez concluida esta operación, se aplica la tokenización de los textos, la cual es esencial para el entrenamiento del modelo. En esta etapa, los datos se transforman en tokens, permitiendo que el modelo comprenda mejor. Después se extraen los tokens iniciales de las capas del modelo. Esta extracción es crucial para aplicar el fine-tuning, un proceso en el cual el modelo ajusta sus parámetros específicos para mejorar el rendimiento en la tarea específica. Una vez finalizado, el modelo se encuentra listo para realizar predicciones en nuevos datos y evaluar su desempeño. Esta secuencia de operaciones, desde el procesamiento de datos hasta la predicción, permite que nuestro modelo alcance su máximo potencial y demuestre su eficacia en la detección de cambios de autor.

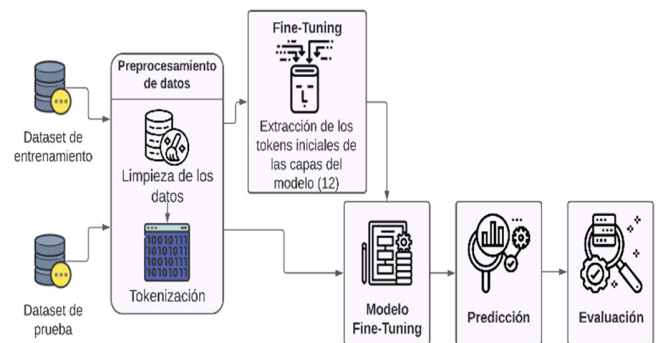


Fig. 2 Esquema del modelo propuesto.

B. Dataset utilizado

Los datos [16] fueron obtenidos de las tareas compartidas de Multi-Author Writing Style Analysis de PAN CLEF 2023¹, el cual, contiene textos en el idioma inglés, y sobre cada texto, múltiples párrafos cortos que varían del tema que serán el valor característico para la identificación del cambio de autoría. El conjunto de dato se basa en las publicaciones de usuarios de varios subreddits de la plataforma Reddit. Se hará uso de 5100 registros, el cual se encuentran dividido en 3 partes, como se presenta en la TABLA 1.

TABLA 1
CANTIDAD DE MUESTRAS DE LOS DATASETS

Dataset	Cantidad de muestras
Entrenamiento	3360
Validación	900
Prueba	840

El dataset fue proporcionado de forma no estructurada, en el que se contó con dos tipos de archivos: tipo texto y tipo json, siendo la primera, los párrafos de un documento y el segundo, un json que corresponde a la cantidad de autores que han intervenido en un documento y dónde se produce esos cambios. De modo que se procedió a etiquetarla, estructurarla y guardarla en un documento json.

id	textos	same	authors	totalParrafo
0	3233 The reason why unions are so important is in t...	[1, 1, 1]	4	4
1	3923 While conducting research in Ireland, 1981-198...	[0, 1]	2	3
2	4155 The difference between a revolt and a revoluti...	[1, 1, 1, 1, 0, 1]	4	7
3	1936 In general, be courteous to others. Debate/dis...	[1, 1, 1, 1, 0, 1]	4	7
4	3041 Most of western Europe operated within a cultu...	[1, 1, 0]	2	4
...
3355	3568 I have a fam full of nurses and wayyyy to much...	[1, 1]	3	3
3356	1131 While the political aspects of the French Revo...	[0, 1, 1]	2	4
3357	343 The nurse on duty should have verified the bra...	[1, 1, 1]	4	4
3358	3075 My librarian friend loves to point out that it...	[1, 1]	3	3
3359	2386 So I was watching a YouTube video about traffi...	[1, 1, 1]	4	4

Fig. 3 Dataset estructurado.

C. Preprocesamiento de datos

Inicialmente, se realiza un agrupamiento de pares contiguos de cada párrafo para cada texto, en el cual se procede a realizar la limpieza de datos y posteriormente su tokenización. Para cada grupo de par, se define la etiqueta “same” de valor binario, donde indicará si existe o no un cambio de autor entre análisis de los párrafos consecutivos. Tal como se puede apreciar en la Fig. 4.

id	pair	same	text_vec
0	3408	[“For clarity, you retain all of your ownershi...	1 [1, 286, 10498, 6, 47, 7615, 70, 9, 110, 4902...
1	3408	[Other people have quoted the ToS so I’m not g...	1 [1, 24989, 82, 33, 5304, 5, 598, 104, 98, 38, ...
2	2796	[So you should pull up a record of how many go...	1 [1, 2847, 47, 197, 2999, 62, 10, 638, 9, 141, ...
3	2796	[If the humidity is not something the plants r...	0 [1, 1106, 5, 19849, 16, 45, 402, 5, 3451, 2703...
4	3483	[But the most incredible part of the story....	1 [1, 1708, 5, 144, 3997, 233, 9, 5, 527, 734, 2...
...
10396	2088	[Dictatorship? Warcrimes? Fuck off outta here ...	1 [1, 495, 11726, 3629, 4128, 116, 1771, 8344, 9...
10397	2088	[We actually are, the a massive solar plant be...	1 [1, 170, 888, 32, 6, 5, 10, 2232, 4118, 2195, ...
10398	3260	[That’s what they’ve been trying to do, with l...	1 [1, 1711, 579, 99, 51, 5030, 57, 667, 7, 109, ...
10399	3260	[Trump’s Assistant Attorney General, Jeffrey...	1 [1, 140, 579, 6267, 2745, 1292, 6, 9011, 4433...
10400	3260	[That’s the key to it right there. Those who d...	1 [1, 1711, 579, 5, 762, 7, 24, 235, 89, 4, 2246...

Fig. 4 Dataset pre-procesado.

D. Tokenización

La tokenización es una parte esencial del proceso de preparación de los datos a entrenar de los modelos basados en BERT. De manera que, es necesario transformar los textos a valores numéricos para que el modelo a entrenar pueda ser capaz de entender y, aprender los patrones y características del texto. Para ello, en la TABLA 2, se mencionan los tokenizadores utilizados respectivos para cada modelo.

TABLA 2
TOKENIZADORES USADOS

Modelo	Tokenizador
microsoft/mdeberta-v3-base	AutoTokenizer
microsoft/deberta-base	DebertaTokenizer

E. Extracción de los embeddings

Los modelos empleados tienen una arquitectura que consta de 12 capas ocultas. Los embeddings de los tokens iniciales necesarios en este estudio se localizan en la posición outputs[1] del estado oculto. De la misma posición de salida se obtienen los tokens [CLS] para cada modelo. Se itera sobre las 12 capas ocultas para luego agrupar cada token extraído y guardarlos en una variable.

```
[outputs[1][n][:, 0, :] for n in range(1, 13)]
```

Fig. 5 Extracción de los embeddings con los modelos DeBERTa y mDeBERTa.

Durante esta etapa, se utiliza los modelos pre-entrenados DeBERTa o mDeBERTa como punto de partida. Extrayendo los embeddings correspondientes a los tokens iniciales ([CLS]) de las 12 capas del modelo. Estos tokens iniciales son especialmente valiosos porque nos permiten capturar información contextual y resumir la representación del texto completo.

¹ <https://pan.webis.de/clef23/pan23-web/style-change-detection.html>

F. Determinación de hiperparámetros

Los hiperparámetros se ajustaron y definieron antes del inicio del entrenamiento, los cuales afectan directamente en el desempeño del modelo. Los hiperparámetros utilizados, son: la función de activación, drop out, learning rate, la cantidad de epochs y el tamaño de aprendizaje (batch size).

Los valores de los hiperparámetros utilizados fueron determinados utilizando la biblioteca Optuna², la cual se encarga de buscar y determinar los hiperparámetros que mejoren el rendimiento del modelo a entrenar, mediante la maximización del F1. Trabajando con los siguientes rangos y alternativas de hiperparámetros:

TABLA 3
HIPERPARÁMETROS A EXPERIMENTAR

Hiper parámetro	Valor
Función de activación	Tanh, ReLU, GELU
Drop out	0.2, 0.5
Learning rate (Tasa de aprendizaje)	3,00E-05, 5,00E-05
Batch size	8, 16, 64
Epoch	5

Adicional, se utilizó la técnica de parada temprana (Early Stopping) al estar calculando los hiperparámetros óptimos con Optuna. El cual comparará los últimos 4 registros, y si no se observa una mejora significativa en las métricas, se detendrá la ejecución. Retornando para ambos modelos los hiperparámetros óptimos de modo que, para cada modelo se aplicó diferentes hiperparámetros con la finalidad de maximizar sus resultados en la predicción.

TABLA 4
HIPERPARÁMETROS ÓPTIMOS

Modelo	Drop out	Función activación	Learning rate	Epochs	Batch size
DeBERTa	0.2479	Relu	3.02E-05	4	8
mDeBERTa	0.2991	Gelu	4.28E-05	4	16

G. Ajuste del modelo (Fine-Tuning)

Al entrenar el modelo, se realiza el fine-tuning para adaptarlo a la tarea específica de detección de cambios de autor en textos multi-autor. Donde se le agrega dos capas lineales, la función de activación y de pérdida. Este nos ayudará a adaptar el modelo obtenido, previamente entrenado, aprendiendo las características en la identificación de la sección de cambio de autoría.

H. Entrenamiento del modelo

Para iniciar el entrenamiento, se inicia con los hiperparámetros óptimos establecidos por Optuna. El entrenamiento se ejecuta con una función `trainer.train()`, el cual contiene el modelo, que extraerá los embeddings y realizará el proceso de Fine-Tuning, los parámetros para personalizar el

modelo; dataset de entrenamiento y validación (previamente pre-procesados); y la función `compute_metrics`, el cual calculará las métricas de evaluación del modelo entrenado. Con ello, se procede a guardar el modelo para su posterior uso.

I. Predicción

Se carga el modelo entrenado, se toma el dataset de prueba previamente procesado (tokenizado), y en conjunto, se ingresan los parámetros necesarios para proceder a realizar la predicción, del cual obtendremos las métricas de evaluación para evaluar la eficacia y precisión del modelo en la identificación de cambio de autor.

	predict	labels
0	[1, 1]	[1, 0]
1	[1, 1, 1]	[1, 1, 1]
2	[1, 1]	[1, 1]
3	[1, 1]	[1, 1]
4	[1]	[1]
...
835	[1, 1, 1, 1]	[1, 1, 1, 1]
836	[1, 1, 1, 1]	[1, 1, 1, 1]
837	[1, 1]	[1, 1]
838	[1, 1, 1]	[1, 1, 1]
839	[1, 1, 1, 1, 1, 1, 1, 1, 1]	[1, 0, 1, 1, 1, 1, 1, 0]

840 rows × 2 columns

Fig. 6 Muestra de la predicción del modelo.

La predicción arroja dos listas por cada texto: 'predict' y 'labels'. En 'labels' se encuentran los cambios de autoría reales del documento, mientras que en 'predict' se registran las predicciones del modelo. En ambas listas, el valor 1 representa un cambio de autor detectado en la comparación de textos, mientras que el valor 0 indica la ausencia de cambio.

J. Evaluación

Se evalúa el modelo de detección de cambio de autor en textos multi-autor, midiendo el rendimiento del modelo mediante las métricas de evaluación. Las métricas nos proporcionarán información del desempeño del modelo, de modo que, las métricas utilizadas son: F1 binary y Accuracy. Adicional, también se tomó la matriz de confusión, el cual permitió visualizar las predicciones del modelo y el desempeño que ha tenido.

III. RESULTADOS

Tras finalizar el entrenamiento y realizar la predicción del modelo, junto con la obtención de sus métricas en textos que se encuentran en inglés, se pudo notar que DeBERTa presenta un tiempo de entrenamiento considerablemente más eficiente en

² Optuna, <https://optuna.org/>

comparación con el modelo mDeBERTa, al ser un tiempo de entrenamiento menor.

TABLA 5
TIEMPO DE ENTRENAMIENTO DE AMBOS MODELOS

Modelo	DeBERTa	mDeBERTa
Tiempo	5:45:33	6:45:35

Durante el entrenamiento, se evaluó su rendimiento dando los siguientes resultados:

TABLA 6
MÉTRICAS DE EVALUACIÓN EN ENTRENAMIENTO DE AMBOS MODELOS

Modelo	Eval loss	F1 binary	Accuracy	Precision Weighted	Epoch
DeBERTa	0,2782	0,9908	0,9840	0,9841	4
mDeBERTa	0,3178	0,9901	0,9827	0,9826	4

Se compararon los modelos usando la métrica F1 para identificar cambios de autor. El modelo DeBERTa obtuvo un F1 de 0.9908, acercándose al límite ideal de 1, demostrando alta precisión en la detección de cambios de autoría. El modelo mDeBERTa obtuvo un F1 ligeramente inferior de 0.9901. La exactitud evaluada por el accuracy, fue más alta en DeBERTa (0.9840) que en mDeBERTa (0.9827). Estos resultados indican que DeBERTa superó a mDeBERTa en la identificación de cambios de autoría.

TABLA 7
MÉTRICAS DE EVALUACIÓN DE LA PREDICCIÓN DE AMBOS MODELOS

Modelo	F1 macro	F1 binary	Accuracy	Precision Weighted
DeBERTa	0,9721	0,9935	0,9886	0,9885
mDeBERTa	0,9647	0,9912	0,9846	0,9846

En la predicción, mediante el análisis basado en la métrica F1-binary, se evidenció que el modelo DeBERTa superó al modelo mDeBERTa con 0.9721 en la tarea de detectar cambios de autor en un texto, frente a 0.9647 obtenidos por mDeBERTa. Asimismo, DeBERTa destacó por su mayor precisión en la métrica F1, así como en la métrica accuracy, demostrando un mejor rendimiento general en comparación con mDeBERTa. Estos resultados clarifican que DeBERTa sobresale y muestra excelencia en la detección de cambios de autoría, superando al modelo mDeBERTa.

La matriz de confusión nos permite visualizar el rendimiento de los modelos al predecir sobre el conjunto de datos de prueba, como se muestra en la Fig. 7 y Fig. 8.

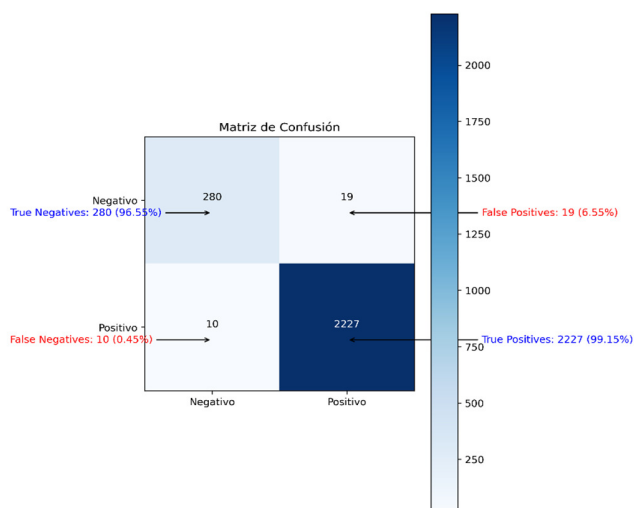


Fig. 7 Matriz de confusión de la predicción con modelo DeBERTa.

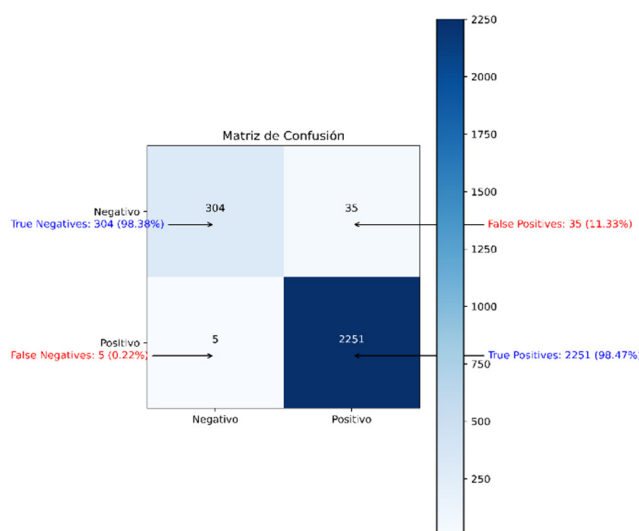


Fig. 8 Matriz de confusión de la predicción con modelo mDeBERTa.

Al evaluar los resultados mediante las matrices de confusión de los modelos DeBERTa y mDeBERTa, se observa un buen rendimiento en clasificar verdaderos positivos y verdaderos negativos para ambos modelos. Sin embargo, se destacan diferencias importantes entre ellos.

DeBERTa muestra mayor precisión al clasificar instancias negativas, con solo 19 falsos positivos en comparación con los 35 de mDeBERTa. Esto sugiere una ventaja de DeBERTa en la clasificación de instancias negativas. Por otro lado, mDeBERTa muestra mayor precisión en la métrica de falsos negativos, con solo 5 casos clasificados incorrectamente como negativos, mientras que DeBERTa tiene 10 casos. Esto indica una mayor precisión de mDeBERTa al clasificar instancias positivas como negativas, lo cual es relevante en contextos donde identificar con precisión los cambios de autor es esencial.

En general, ambos modelos han demostrado ser efectivos en la detección de cambios de autor, cada uno con sus fortalezas en distintos aspectos de la clasificación. Estos resultados son valiosos para comprender las capacidades y limitaciones de cada modelo en esta tarea específica.

IV. DISCUSIÓN

Ambos modelos de entrenamiento, DeBERTa y mDeBERTa, demostraron excelentes resultados, destacando su efectividad en la clasificación de verdaderos positivos y verdaderos negativos. DeBERTa mostró una mayor precisión al clasificar instancias negativas con solo 19 falsos positivos, mientras que mDeBERTa se destacó por su precisión en clasificar instancias positivas, con solo 5 falsos negativos.

Al comparar las métricas F1 y accuracy, ambos modelos obtuvieron resultados sobresalientes, con DeBERTa superando ligeramente a mDeBERTa en la identificación de cambios de autor, obteniendo un F1 de 0.9908 frente a 0.9901 de mDeBERTa.

Hay información semántica y sintáctica contenida en los tokens iniciales de las capas de atención, que sirvió para clasificar los textos y determinar si se producen cambios de autor entre diversos párrafos.

Estos resultados resaltan la efectividad y buen rendimiento de ambos modelos en la detección de cambios de autor en textos multi-autor en inglés. Tanto DeBERTa como mDeBERTa ofrecen soluciones sólidas para esta tarea y son opciones recomendables para mejorar la eficacia en la identificación precisa de cambios de autor en documentos multi-autor.

V. CONCLUSIONES

La investigación ha demostrado que ambos modelos han obtenido resultados excelentes en la detección de cambios de autoría en textos en inglés, mostrando un buen rendimiento y precisión general. Junto con la librería Optuna se experimentó con los diferentes modelos y permitió obtener los mejores hiperparámetros que permitieron maximizar el entrenamiento del modelo. De modo que, los resultados brindan una sólida base para futuras investigaciones y resaltan la importancia de la detección de cambios de autoría en documentos multi-autor, asegurando la transparencia y atribución correcta de la autoría en entornos colaborativos.

Los tokens iniciales de las capas de atención contienen información sintáctica y semántica de los textos, que fue usada para la clasificación de estos en la tarea. Utilizando los modelos DeBERTa y mDeBERTa se obtuvo excelentes resultados con el modelo propuesto para clasificación de textos en inglés. Sería interesante experimentar con otros modelos para el análisis de textos largos, así como para la clasificación de textos en idioma español.

DeBERTa y mDeBERTa obtuvieron resultados similares, considerando que mDeBERTa es la versión multilingüe de DeBERTa. mDeBERTa podría ser utilizado en futuras

investigaciones relacionadas que utilicen datasets de textos en español

Es importante mencionar que, a la fecha de la presente investigación y artículo, no se hallaron investigaciones que aborden el enfoque de extracción de embeddings de tokens iniciales para detectar cambios de autor en textos multi-autor, por lo que no hemos presentado comparaciones con el estado del arte al respecto.

El método propuesto sienta una base de investigaciones futuras y, por los excelentes resultados obtenidos, contribuye a estado del arte de la investigación sobre el tema. Con enfoque en el Procesamiento del Lenguaje Natural, esta investigación contribuye al avance de la verificación de autoría y la protección contra el plagio y atribuciones erróneas en trabajos académicos y científicos.

REFERENCIAS

- [1] F. A. Acheampong, H. Nunoo-Mensah, and W. Chen, "Transformer models for text-based emotion detection: a review of BERT-based approaches," *Artif Intell Rev*, vol. 54, no. 8, pp. 5789–5829, Dec. 2021, doi: 10.1007/s10462-021-09958-2.
- [2] A. Vaswani *et al.*, "Attention Is All You Need," 2017.
- [3] C. M. Ormerod, A. Malhotra, and A. Jafari, "Automated essay scoring using efficient transformer-based language models," Feb. 2021, Accessed: May 31, 2023. [Online]. Available: <http://arxiv.org/abs/2102.13136>
- [4] C. Espin-Riofrio *et al.*, "Determination of Political Affinity of Ecuadorian Twitter Users Using Machine Learning Techniques for Authorship Attribution Determinación de Afinidad Política de Usuarios de Twitter de Ecuador Utilizando Técnicas de Machine Learning para Atribución de Autoría," no. 2, doi: 10.18687/LACCEI2022.1.1.535.
- [5] C. Espin-Riofrio, J. Ortiz-Zambrano, and A. Montejó-Ráez, "SINAI at PoliticEs 2022: Exploring Relative Frequency of Words in Stylometrics for Profile Discovery," 2022, Accessed: Aug. 20, 2023. [Online]. Available: <http://ceur-ws.org>
- [6] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", Accessed: Jun. 14, 2023. [Online]. Available: <https://github.com/tensorflow/tensor2tensor>
- [7] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does BERT look at? An Analysis of BERT's Attention," pp. 276–286, 2019, Accessed: Jun. 13, 2023. [Online]. Available: <https://github.com/>
- [8] F. Pedregosa FABIANPEDREGOSA *et al.*, "Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011, Accessed:

- Jun. 21, 2023. [Online]. Available: <http://scikit-learn.sourceforge.net>.
- [9] Z. Zhang *et al.*, “Style Change Detection Based On Writing Style Similarity Notebook for PAN at CLEF 2021,” 2021.
- [10] J. Zi, L. Zhou, and Z. Liu, “Detección de cambios de estilo Basado en Bi-LSTM y Bert,” 2022, Accessed: May 31, 2023. [Online]. Available: <http://ceur-ws.org/Vol-3180/paper-234.pdf>
- [11] S. Alshamasi and M. Menai, “Agrupación en clústeres basada en conjuntos para la detección de cambios de estilo de escritura en documentos textuales de varios autores,” 2022, Accessed: May 31, 2023. [Online]. Available: <https://ceur-ws.org/Vol-3180/paper-187.pdf>
- [12] Q. Lao *et al.*, “Detección de cambios de estilo Basado en Bert y Conv1d,” 2022, Accessed: May 31, 2023. [Online]. Available: https://pan.webis.de/downloads/publications/papers/lao_2022.pdf
- [13] M. Fabien, E. Villatoro-Tello, P. Motlicek, and S. Parida, “BertAA: BERT fine-tuning for Authorship Attribution,” pp. 127–137.
- [14] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, no. Mlm, pp. 4171–4186, 2019.
- [15] “Utilizing Transformer Representations Efficiently.” Accessed: May 04, 2024. [Online]. Available: <https://www.kaggle.com/code/rhtsingh/utilizing-transformer-representations-efficiently/notebook>
- [16] E. Zangerle, M. Mayerl, M. Potthast, and B. Stein, “PAN23 Multi-Author Writing Style Analysis,” Mar. 2023, doi: 10.5281/ZENODO.7729178.