ISBN: 978-628-95207-8-1. ISSN: 2414-6390. Digital Object Identifier: 10.18687/LACCEI2024.1.1.108

Embeddings of Initial Tokens from BERT-Based Models to Identify Human-Written or Automatically Generated Text

César Espin-Riofrio, MSc.[1], Jorge L. Charco, PhD.[1], Débora K. Preciado-Maila, MGs.[1], Luis Ramos-Ramírez, Ing.[1], Holger Camacho-Villalva, Ing.[1], Arturo Montejo-Ráez, PhD.[2]
[1]Universidad de Guayaquil, Ecuador, cesar.espinr@ug.edu.ec, jorge.charcoa@ug.edu.ec, debora.preciadom@ug.edu.ec, luis.ramosra@ug.edu.ec, holger.camachovi@ug.edu.ec
[2]Universidad de Jaén, España, amontejo@ujaen.es

Abstract– ***Remarkable advances in text generation models have significantly expanded their applicability in a wide variety of fields. It is difficult to identify whether a text has been written by human or automatically generated, due to the ability of these models to mimic human style, coherence and expression. In this research, a Deep Learning method focused on Natural Language Processing (NLP) is proposed to identify the origin of a text. It is based on the extraction of the embeddings of the initial tokens of the twelve hidden layers of BERT-based Transformers models. The dataset provided in the IberLEF 2023 AuTexTification task was used, with texts extracted from different domains, in English and Spanish language. The DeBERTa model was used for the English texts and mDeBERTa for the Spanish texts. Optuna was used to automate the search for the optimal hyperparameters for the final training, performing fine-tuning of each model for its subsequent prediction and evaluation. The evaluation results of the proposed model were excellent, while the prediction results were not so good, being an interesting point for discussion and analysis of the proposal.***

Keywords-- ***Natural Language Processing, Transformers, Embeddings of initial tokens, human or machine.***

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

Embeddings de Tokens Iniciales de Modelos Basados en BERT para Identificar Texto Escrito por Humano o Generado Automáticamente.

César Espin-Riofrio, MSc.[1], Jorge L. Charco, PhD.[1], Débora K. Preciado-Maila, MGs.[1], Luis Ramos-Ramírez, Ing.[1], Holger Camacho-Villalva, Ing.[1], Arturo Montejo-Ráez, PhD.[2]
[1]Universidad de Guayaquil, Ecuador, cesar.espinr@ug.edu.ec, jorge.charcoa@ug.edu.ec, debora.preciadom@ug.edu.ec, luis.ramosra@ug.edu.ec, holger.camachovi@ug.edu.ec
[2]Universidad de Jaén, España, amontejo@ujaen.es

Resumen– ***Los notables avances en los modelos de generación de texto han expandido significativamente su aplicabilidad en una amplia variedad de campos. Resulta difícil identificar si un texto ha sido escrito por humano o generado automáticamente, debido a la capacidad de estos modelos para imitar el estilo, la coherencia y la expresión humana. En esta investigación, se propone un método de Deep Learning enfocado al Procesamiento de Lenguaje Natural (PLN) para identificar el origen de un texto. Se basa en la extracción de los embeddings de los tokens iniciales de las doce capas ocultas de modelos Transformers basados en BERT. Se utilizó el dataset proporcionado en la tarea AuTexTification de IberLEF 2023, con textos extraídos de diferentes dominios, en idioma inglés y español. El modelo DeBERTa se utilizó para los textos en inglés y mDeBERTa para los textos en español. Con Optuna se automatizó la búsqueda de los hiperparámetros óptimos para el entrenamiento final, realizando fine-tuning de cada modelo para su posterior predicción y evaluación. Los resultados de evaluación del modelo propuesto fueron excelentes, mientras que los de predicción no lo fueron tanto, siendo un punto interesante para la discusión y análisis de la propuesta.***

Palabras clave—Procesamiento de Lenguaje Natural, Transformers, Embeddings de tokens iniciales, humano o máquina.

I. INTRODUCCIÓN

Durante los últimos años, los modelos de generación de textos (text generation models, TGM) han experimentado un crecimiento exponencial en cuanto a uso y popularidad, debido a que estos sistemas pueden generar textos con características similares al ser humano. Algunos sistemas como Writer, Jasper o ChatGPT son de libre acceso y utilizados en múltiples aplicaciones como asistente de escritura, generación de resúmenes de textos, traducción automática, generación de textos periodísticos, generación de guiones, entre otros. La ventaja que ofrece a sus múltiples usuarios es que puede generar texto de forma automática mediante una simple instrucción de entrada, esta funcionalidad beneficia el ahorro de tiempo y aumenta significativamente la productividad de quienes los utilizan.

Sin embargo, el uso incorrecto y desconocimiento puede generar desafíos que deben ser analizados, como el plagio, generación y propagación de información falsa, desconocimiento del autor al que se le atribuye un texto, también los textos pueden carecer de originalidad. Por lo tanto, se propone desarrollar un método de aprendizaje automático para determinar si un texto fue escrito por humano o generado automáticamente, mediante la extracción de embeddings de tokens iniciales de las capas de modelos Transformers basados en BERT.

La relevancia de realizar esta investigación surge a partir de la necesidad de identificar el origen o autor de un texto, actualmente estamos expuestos a factores como la desinformación presente en el entorno digital, donde cada vez es frecuente encontrar con información falsa o engañosa, la falta de transparencia sobre los textos que leemos, el uso con fines malicioso de estos sistemas puede afectar la seguridad de los usuarios, la posible atribución de un texto generado automáticamente por estos sistemas como una elaboración propia.

En este sentido, [1] señalan que el impacto de los TGM en el mercado global reflejó una recaudación de $423.8 millones en el 2022. Se proyecta que para el 2032 esta cifra aumente considerablemente a $2 mil millones, con un crecimiento anual del 18,2% estimado para esos años.

En la actualidad, contamos con una variedad de modelos y técnicas de generación de texto, por lo que resulta fundamental explorar investigaciones recientes en este campo. Usando modelos Transformers, [2] presentaron el modelo de generación de texto XLNet para capturar dependencias bidireccionales en el texto, demostrando superioridad del modelo con otros modelos de distintos campos del PLN. [3] exponen el modelo CTRL con la particularidad de que permite al usuario especificar "instrucciones" o "influir" en la generación de texto, además ofrece la característica que puede generar textos de múltiples dominios, ya sean informes, blogs, reseñas, entre otros. Con GPT-3, [4] tienen el objetivo de investigar la capacidad que tiene el modelo para realizar tarea de PLN con pocas indicaciones, demostrando un rendimiento aceptable en tareas como traducción, preguntas y respuestas, generación de texto. Un área en constante investigación y que busca mejorar la calidad en la redacción es la generación de texto controlada, [5] plantea utilizar future discriminators, para predecir el texto futuro, el objetivo principal es otorgar una orientación para mantener la coherencia del tiempo, cumplir

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

con la estructura especificada y focalizar algunos aspectos del contenido durante el proceso de generación del texto.

El Test de Turing [6], fue la primera prueba que se realizó para identificar la capacidad de una máquina para simular un comportamiento inteligente difícil de distinguir por el ser humano. Existen diferentes alternativas para detectar textos escritos por humano o generado por una máquina, dentro de las investigaciones realizadas tenemos, [7] en colaboración entre el MIT-IBM Watson AI laboratorio y HarvardNLP, presentaron Giant Language Model Test Room (GLTR), modelo capaz de identificar texto escrito por humano o generado por GPT-2, emplearon como parte de su metodología detectar la entropía, la longitud del texto, la frecuencia de palabras, la capacidad del modelo para detectar los textos es de 72% de precisión. [8] desarrollaron un algoritmo de machine Learning con el fin de detectar desinformación basada en IA, aplicando un modelo Transformers SmallBERT y una red neuronal, obteniendo una precisión del 97% dependiendo de la función de pérdida empleada. La propuesta de [9] obtuvo una precisión alta para detectar textos generados por modelos GPT, "DetectGPT" como sus autores lo nombraron, detecta textos sin la necesidad de entrenamiento o Zero-Shot. Dentro de las investigaciones de modelos de detección de textos, un tema de importancia es la vulnerabilidad que puede presentar, [10] evidenciaron que los sistemas pueden ser vulnerables si se parafrasea el contenido generado por IA, mediante la técnica del parafraseo se logró engañar a los sistemas generando así la necesidad de crear técnicas más efectivas. Para detectar texto generado por ChatGPT, [11] desarrolló "GPTZero" utilizando la técnica de regresión lineal como versión inicial y regresión logística en la versión mejorada, la versión mejorada del modelo fue puesto a prueba con artículos de la BBC y artículos generados por IA, dando como resultado una tasa de falsos positivos menos del 2% evidenciando su gran aporte para la detección de textos.

Con el nacimiento de los modelos Transformers [12], revolucionaron los estudios en el campo del PLN, generando un aporte no solo en la clasificación de texto sino en múltiples tareas, las principales propuesta del modelo eran los mecanismos de self-attention para conservar el contexto de las palabras generadas a largo plazo y las representaciones de posición para cada palabra ingresada al modelo, permitiendo realizar un procesamiento del texto en paralelo sin perjudicar el orden de la oración.

BERT es un modelo basado en la arquitectura de los Transformers usando solo el bloque codificador. Presentado por [13], detallan a BERT como un modelo pre-entrenado con grandes volúmenes de textos sin etiquetas con el objetivo de aprender el conocimiento general del lenguaje. A diferencia de otros modelos similares, BERT usa un enfoque bidireccional para captar el contexto de una palabra según las palabras que tengan antes y después en la oración.

Las salidas de codificación de BERT determinan que son estructuras de datos que contienen información que el modelo devuelve, dichos datos se pueden usar en tuplas o diccionarios, [14]. Hay tres tipos de salidas importantes dentro de cada modelo Transformers, para el ejemplo de BERT podemos resaltar las siguientes:

- Output_0 *Last hidden state* o último estado oculto, es la última representación contextualizada de cada token, se refiere al estado oculto final asociado con cada token de entrada después de pasar por las doce capas del modelo BERT.
- Output_1 *Pooler output,* es la salida de la capa "Pooling" la cual es considerada una representación incorporada del texto completo, es usada comúnmente para tareas de clasificación de texto o para tener representaciones fijas del texto.
- Output_2 *Hidden state* o estado oculto, son las doce capas de atención que se encuentra en el modelo BERT, dependiendo del modelo el número de capas varia, cada una de las capas presentes en el modelo tienen su propio estado oculto el cual contiene información sobre la representación contextual de los textos ingresados.

Según las observaciones de [15] las doce capas se pueden dividir en dos partes fundamentales, en la primera parte detalla que las capas inferiores del modelo se puede apreciar la información sintáctica del texto, mientras que en la segunda parte las capas superiores capturan información semántica más elaborada.

Una versión mejorada del modelo BERT es el modelo utilizado para nuestra tarea, DeBERTa [16] es un modelo que ha sido entrenado en grandes cantidades de texto para aprender representaciones contextualizadas del lenguaje, propone técnicas para mejorar el rendimiento como el mecanismo de atención desacoplada y la decodificación, presenta una alta capacidad para aprender representaciones contextuales más precisas y capturar relaciones complejas en el lenguaje. mDeBERTa [17], versión mejorada del modelo original DeBERTa, utiliza la estructura de su predecesor, entrenado con datos multilingües de CC100, con un total de 86 millones de parámetros en el backbone y un vocabulario de 250 mil tokens, el modelo comprende mejor las relaciones entre palabras y obtiene información contextual relevante. Para llevar a cabo la clasificación de texto utilizamos modelos Transformers, estos modelos pre-entrenados han demostrado un rendimiento adecuado. [18] utilizaron el modelo Transformers Selectra-Medium para clasificar textos cortos en diferentes categorías como deportes, salud, economía, etc. [19] usaron los modelos Transformers para inferir la orientación política de textos extraídos de Twitter de usuarios en Ecuador.

II. METODOLOGÍA

Para llevar a cabo la investigación, se utilizó la metodología bibliográfica-documental. Consistió en revisar el estado del arte y analizar artículos científicos relevantes para obtener conocimiento sobre el tema. En el contexto de la investigación, se empleó un algoritmo de Machine Learning, aplicando el enfoque de aprendizaje no supervisado y la

clasificación de texto. Se usó la investigación experimental para probar métodos de extracción de los embeddings de codificación de los tokens iniciales de las capas de atención de los modelos Transformers basados en BERT, y con diversos modelos pre-entrenados para clasificar textos en inglés y en español en la tarea de determinar si fue escrito por un humano o generado por una máquina. El método cuantitativo es utilizado para evaluar la eficacia del modelo propuesto mediante métricas que dan idea del rendimiento en las fases de entrenamiento y predicción.

A continuación, en Fig.1, se presentan los detalles del modelo propuesto.

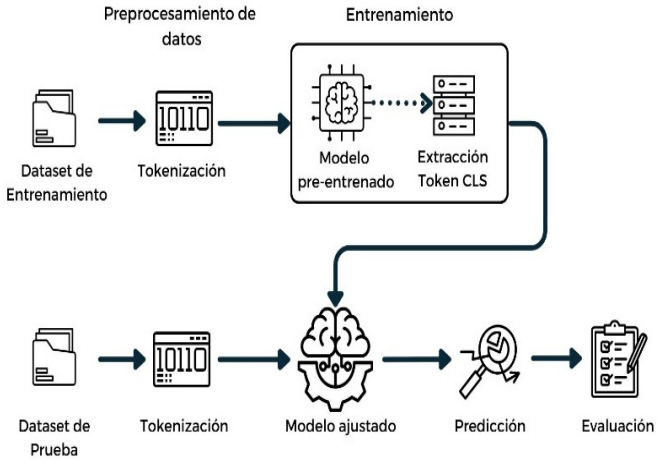

Fig. 1 Esquema del modelo propuesto.

Las etapas del modelo propuesto inician con la selección del dataset a utilizarse, al cual se le realiza el preprocesamiento de los datos, en esta fase se agrega los tokens especiales al inicio y final de cada secuencia de texto. Antes del entrenamiento se extraen los embeddings de los tokens iniciales de las doce capas ocultas de los modelos DeBERTa y mDeBERTa, para proceder a entrenar el modelo propuesto. A partir de aquí, se realizan distintas pruebas utilizando Optuna para obtener los mejores hiperparámetros de entrenamiento, adicional se aplica la técnica de early stopping para detener el aprendizaje cuando no encuentre mejoras, de esta manera las predicciones alcanzan una mayor efectividad, por último, se evalúa el modelo con las métricas establecidas.

A. Dataset

El dataset [20], es el conjunto de datos del desafío académico colaborativo AuTexTification[1] de IberLEF 2023, la propuesta ofrece un dataset para el entrenamiento y otro para las pruebas, cada dataset se encuentran separados en idioma inglés y español. El dataset de entrenamiento contiene textos extraídos de diferentes dominios, tenemos tweets provenientes de red social, wikis obtenidos de artículos de Wikipedia, y dominio legal que corresponde a textos de documentos legales. En cambio, el dataset de prueba contiene textos extraídos de noticias (news) y reseñas (reviews), dominios diferentes al dataset de entrenamiento. La cantidad de textos en los datasets se aprecian en tabla 1.

TABLA 1
TAMAÑO DEL DATASET DE ENTRENAMIENTO Y PRUEBA

Muestras por dataset		
idioma	Entrenamiento	Prueba
Inglés	33.845	21.832
Español	32.062	20.129

Tanto el dataset de entrenamiento y prueba están etiquetados de la siguiente manera; "id" representación numérica única para identificar cada fila, "text" contiene el texto extraído de los diferentes dominios, "label" establece si el contenido de la columna "Text" fue escrito por un humano o generado por una máquina (human o generated), "domain" indica el dominio u origen de donde proviene cada texto. En Figura 2 se aprecia una muestra del dataset de entrenamiento.

	id	text	label	domain
0	5464	Entrada en vigor. La presente Directiva entrar...	human	legal
1	30129	Preguntas: 1. ¿Cuáles son los principales argu...	generated	wiki
2	19553	¿Desea algo? Póngame una caja de madera. ¿Qué ...	generated	tweets
3	13005	@victor28088 1665 Tweets no originales, que as...	human	tweets
4	16919	De pequeño Dios me dio a elegir entre tener un...	human	tweets
...	...	...	...	...
32057	16850	Mamá, ¿por qué no me despertaste? Te hable 5 v...	human	tweets
32058	6265	. Artículo 2. Los Estados miembros aplicarán l...	human	legal
32059	11284	Mi memoria es: □ 5% de los médicos tienen una ...	generated	tweets
32060	860	APROBAR el proyecto de resolución que se adjun...	generated	legal
32061	15795	Siento que todo el mundo se alejó de mí por se...	human	tweets

32062 rows × 4 columns

Fig. 2 Contenido del dataset de entrenamiento en español.

Es necesario codificar los valores de la etiqueta "label", se establece valores de 0 para textos generados por máquina y 1 para los generados por humano, se normalizan los datos para obtener un formato uniforme. El dataset de entrenamiento es dividido en 80% para entrenamiento y 20% para validación.

B. Tokenización

Para trabajar con el dataset de entrenamiento con textos en inglés se utilizó DeBeRTa con su respectivo tokenizador

[1] https://sites.google.com/view/autextification

DeBERTaTokenizer, mientras que para los textos en español se empleó mDeBERTa con el tokenizador AutoTokenizer, los cuales agregan los tokens CLS inicial y SEP de separación para formar los vectores respectivos de los pares de textos. Con los vectores correspondientes se genera el input_ids y attention_mask necesarios para realizar el ingreso de los modelos pre entrenados. En Figura 3 se aprecia el resultado del texto tokenizado.

	id	text	label	same	text_vec
0	5464	Entrada en vigor. La presente Directiva entrar...	human	1	[1, 30495, 338, 2095, 1177, 28131, 368, 4, 158...
1	30129	Preguntas: 1. ¿Cuáles son los principales argu...	generated	0	[1, 510, 4950, 5973, 281, 35, 479, 10016, 1526...
2	19553	¿Desea algo? Póngame una caja de madera. ¿Qué ...	generated	0	[1, 211, 4468, 102, 1076, 2977, 116, 221, 1479...
3	13005	@victor28088 1665 Tweets no originales, que as...	human	1	[1, 25005, 368, 18868, 2580, 117, 1461, 293, 6...
4	16919	De pequeño Dios me dio a elegir entre tener un...	human	1	[1, 13365, 3723, 3407, 1021, 211, 4544, 162, 3...
5	25630	Si lo haces después, es posible que muevas las...	human	1	[1, 35684, 4600, 1368, 11667, 18690, 257, 5739...
6	17644	¿Me puede vender un preservativo? Disculpe, aq...	human	1	[1, 1464, 181, 6796, 242, 748, 7232, 542, 9383...
7	17134	Cosas que hago mientras estudio: 1% estudiar 2...	human	1	[1, 37294, 281, 1192, 1368, 6643, 475, 4843, 5...

Fig. 3 Texto tokenizado.

C. *Extracción de embeddings*

Los modelos utilizados cuentan con 12 capas ocultas dentro de su estructura, los embeddings de los tokens iniciales requeridos en esta investigación, se encuentran en la posición outputs[1] del hidden state, del mismo índice de salida se obtiene los tokens [CLS] para cada modelo. Se recorre cada una de las 12 capas ocultas para posteriormente apilar cada token extraído y almacenarlo en una variable.

```
outputs[1][n][:,0,:]
```

Fig. 4 Extracción de los tokens CLS.

D. *Hiperparámetros*

Mediante la biblioteca Optuna se generan múltiples pruebas combinando los hiperparámetros de la Tabla 2, de esta manera se busca el óptimo entrenamiento del modelo mediante el objetivo de maximizar el f1 y accuracy durante la búsqueda.

TABLA 2
VARIABLES DE HIPERPARÁMETROS A OPTIMIZAR

Hiperparámetros	**Valores**
Función de activación	"tanh", "relu", "gelu"
Tasa de aprendizaje (Learning rate)	3e-5 hasta 5e-5
Dropout	0.2 hasta 0.5
Batch size	8, 16
Epoch	1 a 5

Para garantizar la optimización de recursos y tiempo, se realizó una parada temprana (early stop) durante la optimización de Optuna, el proceso de optimización se realiza para cada modelo DeBERTa y mDeBERTa con su respectivo dataset de entrenamiento. Los mejores hiperparámetros obtenidos por Optuna para proceder con el fine-tuning de cada modelo se aprecian en Tabla 3.

TABLA 3
HIPERPARÁMETROS ÓPTIMOS

Hiperparámetros	**DeBERTa**	**mDeBERTa**
Función de activación	Relu	Tanh
Tasa de aprendizaje (Learning rate)	4.24e-5	3.74e-5
Dropout	0.3227	0.2223
Batch size	8	16
Epoch	5	3

E. *Ajuste del modelo*

El fine-tuning de los modelos base DeBERTa y mDeBERTa se realiza agregando dos funciones lineales, el dropout, la función de activación, y la función de pérdida CrossEntropyLoss correspondiente para evaluar el rendimiento del modelo, de esta manera se obtiene el ajuste necesario de acuerdo con lo propuesto, concluido los ajustes se procede al entrenamiento de los modelos. Ha sido importante el uso de los recursos de la super computadora de la Universidad de Jaén para la ejecución de este proyecto.

F. *Predicción*

Se dispone a emplear el dataset de prueba para generar las predicciones con cada modelo previamente entrenado, dicho dataset es preprocesado realizando la tokenización y extracción de embeddings. En Figura 5 y 6 se aprecian muestras de las predicciones generadas.

mDeBERTa

	id	text	same	predict
0	17414	Buscábamos tranquilidad y la encontramos. Me t...	1	1
1	16938	Nos sorprendió la cena, si vas con media pensi...	1	1
2	17379	Servicio atento y magníficas vistas al río.	1	1
3	5391	La Oficina Nacional de Estadísticas de China d...	1	0
4	17310	Pero no puedes tener a una sola persona sirvie...	1	1
...	...	...	...	...
20124	11284	Pero no fue un problema para mí en absoluto! E...	0	0
20125	11964	Para mí, no hay nada más importante en la vida...	0	0
20126	5390	El sindicato Futbolistas Argentinos Agremiados...	1	0
20127	860	Incluso ahora se siente joven. Fidel Castro es...	0	0
20128	15795	Pelusillas y pelos en el suelo habitación. Col...	1	0

20129 rows × 4 columns

Fig. 5 Predicciones generadas con mDeBERTa

DeBERTa

	id	text	same	predict
0	15725	It has remained one of my favorite country/swi...	1	1
1	17108	Even with very light use (hard to get motivate...	1	1
2	383	She died in 2015 at age 93. She is survived by...	0	0
3	7809	Londonderry Crown Court heard how Heaney false...	1	0
4	6215	Will Genia, Lachie Turner and Berrick Barnes e...	1	0
...	...	...	...	...
21827	11964	However, it still has some great songs that yo...	0	0
21828	21575	I bought the product because I wanted to see i...	0	0
21829	5390	This will make it easier for us to plan adequa...	1	0
21830	860	The plans include new emergency departments, a...	0	0
21831	15795	Quite frankly I found this volume of old essay...	1	1

21832 rows × 4 columns

Fig. 6 Predicciones generadas con DeBERTa.

III. RESULTADOS

Para medir y analizar el rendimiento de los modelos se optó por utilizar las métricas de evaluación f1, precision, accuracy y recall. Durante la fase de entrenamiento Optuna genera un resumen de los resultados óptimos en las métricas de evaluación de cada modelo, su resultado se aprecia en la Tabla 4.

TABLA 4
RESULTADOS DE LAS MÉTRICAS DE EVALUACIÓN EN ENTRENAMIENTO

Modelo	Train loss	F1-bynary	Accuracy	Precision Weighted	Recall Weighted
DeBERTa	0.0252	0.8897	0.8935	0.8952	0.8935
mDeBERTa	0.0394	0.8827	0.8935	0.9021	0.8935

Tanto el modelo DeBERTa para textos en inglés como el modelo mDeBERTa para textos en español demostraron haber sido entrenados de manera efectiva. Esto se refleja en el "Train loss", el cual alcanzó valores mínimos, indicando que ambos modelos lograron aprender y evaluarse muy bien utilizando datos de validación equivalentes al 20% del conjunto de entrenamiento. Esto se refleja en las métricas f1, precisión, exactitud y recuperación, que exhiben valores altos y cercanos alrededor del 89%.

Se evalúan las predicciones generadas para los textos del dataset de prueba utilizando las mismas métricas, los resultados de esta evaluación se presentan en Tabla 5.

TABLA 5
RESULTADOS DE LAS MÉTRICAS DE EVALUACIÓN EN LAS PREDICCIONES

Modelo	F1-bynary	Accuracy	Precision Weighted	Recall Weighted
DeBERTa	0.4651	0.6526	0.7610	0.6526
mDeBERTa	0.3544	0.6496	0.7702	0.6496

En cuanto a las predicciones, se nota un rendimiento apenas satisfactorio. La métrica accuracy muestra que ambos modelos, DeBERTa para textos en inglés y mDeBERTa para textos en español, están clasificando correctamente el 65% de todas las muestras totales. Sin embargo, hay una diferencia notable en la métrica f1, con un promedio del 41% para ambos modelos. Esto sugiere que los modelos pueden tener dificultades para clasificar los textos en sus etiquetas correctas.

La Figura 7 muestra la matriz de confusión que permite conocer el desempeño de los modelos al comparar las predicciones realizadas por cada modelo utilizado con las etiquetas reales del dataset de prueba.

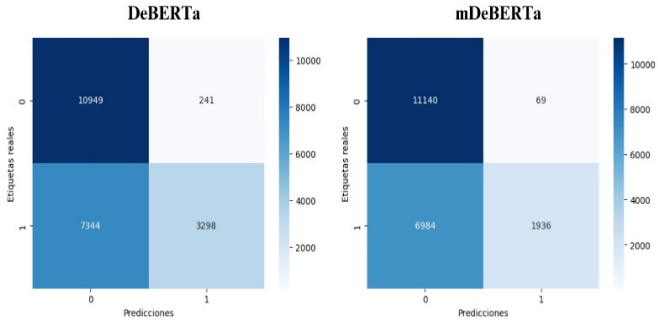

Fig. 7 Matriz de confusión.

IV. DISCUSIÓN

Los resultados obtenidos respaldan la afirmación que se puede clasificar texto escrito por humano o generado por máquina por medio del modelo propuesto. Es evidente señalar que el desempeño de cada modelo durante el entrenamiento fue excelente, no obstante, al enfrentarse a un conjunto de datos nuevos presentaron limitaciones. En cuanto a realizar la predicciones cada modelo obtuvo desaciertos para clasificar correctamente los textos, esto puede ser ocasionado por diferente factores, como los dominios que contienen los dataset, es clave tener en cuenta que los dominios utilizados para entrenamiento son distintos a los dominios usados para las pruebas, puesto que existe una diferencia en la información sintáctica y semántica de los textos que el modelo evalúa durante la predicción, también la variedad de dominios de entrenamiento es baja lo cual podría no estar ayudando al modelo a capturar características en diferentes contextos, factores que de ser analizados podrían mejorar los rendimientos de cada modelo.

Durante la evaluación de entrenamiento se han obtenido resultados satisfactorios alrededor del 90% de puntaje entre todas las métricas, generalmente el accuracy es catalogado como un resultado engañoso, sin embargo, para este caso se puede respaldar como dato confiable debido al alcance que tuvo el f1 del 89%, minimizando falsos negativos y positivos a un 10% de los datos procesados. Por otro lado, es evidente la caída abrupta que presentan las métricas en las predicciones, con el 65% de accuracy el cual no es un valor bajo, pero para este escenario el f1 no respalda dicho valor, la caída al 41% del f1 confirma que el accuracy presenta un puntaje engañoso para las predicciones.

V. CONCLUSIONES

Se evaluaron los modelos y los resultados de las predicciones mostraron que con DeBERTa se obtuvo un rendimiento del 65% para el dataset en inglés, mientras que mDeBERTa presentó el 64% para la predicción con el dataset en español, resultado similares tomando en cuenta que mDeBERTa es la versión extendida de DeBERTa para otros idiomas distintos del inglés.

El enfoque propuesto, en el uso de los embeddings de los tokens iniciales, ha sido importante para esta investigación, ya que los modelos han obtenido un desempeño aceptable para clasificar los textos inglés y español. Dado que el dataset de entrenamiento es de un dominio distinto al dataset de prueba, es una de las razones por la que influye en el rendimiento para clasificar textos con dominios nuevos o no vistos previamente.

Se propone experimentar con la aplicación de otros modelos Transformers basados o no en BERT para clasificar textos escritos por humano o generado por máquina. Así también agregar características de estilo como longitud de párrafo y frecuencia de palabras. Es importante en específico experimentar con textos en idioma español. Continuar con esta línea de investigación permitirá una comprensión más profunda de la capacidad de los modelos frente a las particularidades gramaticales, semánticas y sintácticas presentes en este idioma.

VI. REFERENCES

[1] S. Tanmay, M. Akhila, y K. Vineet, «AI Text Generator Market Research, 2032», *Allied Market Research*, junio de 2023. https://www.alliedmarketresearch.com/ai-text-generator-market-A84406 (accedido 26 de junio de 2023).

[2] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, y Q. V Le, «XLNet: Generalized Autoregressive Pretraining for Language Understanding», 2019. [En línea]. Disponible en: https://github.com/zihangdai/xlnet

[3] N. Shirish Keskar, B. Mccann, L. R. Varshney, C. Xiong, R. Socher, y S. Research, «CTRL: A CONDITIONAL TRANSFORMER LANGUAGE MODEL FOR CONTROLLABLE GENERATION», sep. 2019, Accedido: 23 de julio de 2023. [En línea]. Disponible en: https://github.com/salesforce/ctrl.

[4] T. B. Brown *et al.*, «Language Models are Few-Shot Learners», *Adv Neural Inf Process Syst*, vol. 33, pp. 1877-1901, 2020, Accedido: 29 de mayo de 2023. [En línea]. Disponible en: https://commoncrawl.org/the-data/

[5] K. Yang y D. Klein, «FUDGE: Controlled Text Generation With Future Discriminators», *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pp. 3511-3535, abr. 2021, doi: 10.18653/v1/2021.naacl-main.276.

[6] A. M. Turing, «M I N D A QUARTERLY REVIEW OF PSYCHOLOGY AND PHILOSOPHY I.-COMPUTING MACHINERY AND INTELLIGENCE», 1950, Accedido: 4 de junio de 2023. [En línea]. Disponible en: https://academic.oup.com/mind/article/LIX/236/433/986238

[7] S. Gehrmann, H. Strobelt, y A. M. Rush, «GLTR: Statistical Detection and Visualization of Generated Text», 2019, Accedido: 5 de junio de 2023. [En línea]. Disponible en: http://gltr.io.

[8] A. Najee-Ullah, L. Landeros, Y. Balytskyi, y S. Y. Chang, «Towards Detection of AI-Generated Texts and Misinformation», en *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2022, pp. 194-205. doi: 10.1007/978-3-031-10183-0_10.

[9] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, y C. Finn, «DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature», ene. 2023.

[10] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, y S. Feizi, «Can AI-Generated Text be Reliably Detected?», mar. 2023.

[11] Tian Edward, «gptzero update v1 | GPTZero», 5 de enero de 2023. https://gptzero.me/blogs/gptzero-update-v1 (accedido 28 de mayo de 2023).

[12] A. Vaswani *et al.*, «Attention Is All You Need».

[13] J. Devlin, M. W. Chang, K. Lee, y 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, pp. 4171-4186, oct. 2018, Accedido: 12 de junio de 2023. [En línea]. Disponible en: https://arxiv.org/abs/1810.04805v2

[14] Hugging Face, «Model outputs», *Hugging Face*. https://huggingface.co/docs/transformers/main_classes/output (accedido 25 de junio de 2023).

[15] M. E. Peters, M. Neumann, L. Zettlemoyer, y W. T. Yih, «Dissecting Contextual Word Embeddings: Architecture and Representation», *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pp. 1499-1509, ago. 2018, doi: 10.18653/v1/d18-1179.

[16] P. He, X. Liu, J. Gao, W. Chen, y M. Dynamics, «DeBERTa: Decoding-enhanced BERT with Disentangled Attention», jun. 2020, Accedido: 12 de junio de 2023. [En línea]. Disponible en: https://arxiv.org/abs/2006.03654v6

[17] P. He, J. Gao, y W. Chen, «DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing», nov. 2021, Accedido: 12 de junio de 2023. [En línea]. Disponible en: https://arxiv.org/abs/2111.09543v4

[18] C. I. Espin-Riofrio, K. I. Vera-Guamán, y R. Yela-García III, «Clasificación y etiquetado de tweets de Ecuador para determinar qué tema tratan, utilizando un modelo Transformer», vol. 7, pp. 1282-1295, 2022, doi: 10.23857/pc.v7i3.3791.

[19] C. I. Espín-Riofrio, W. I. Ferruzola-Sánchez, y A. Aspiazu-Torres III, «Identificación de ideología política mediante un modelo Transformer para estilometría y Clasificación por votos en Machine Learning», vol. 70, n.º 9, pp. 1457-1474, 2022, doi: 10.23857/pc.v7i8.

[20] A. Sarvazyan, J. Á. González, M. Franco, F. M. Rangel, M. A. Chulvi, y P. Rosso, «AuTexTification Dataset (Full data)», may 2023, doi: 10.5281/ZENODO.7956207.