Universidad Politécnica de Madrid

Ciencia de Datos e Inteligencia Artificial

Proyecto de Ciencia de Datos

Inteligencia Artificial vs Humanos en la Generación de Textos

Realizado por: Raúl Andrino

2024/2025

Tabla de contenido

1. Abstract

Este trabajo explora la capacidad para distinguir entre textos generados por inteligencia artificial (IA) y aquellos escritos por humanos. Utilizando técnicas avanzadas de procesamiento del lenguaje natural (NLP), se analiza un conjunto textos para identificar patrones que permitan inferir la autoría del texto. El objetivo principal es desarrollar un enfoque robusto que facilite la clasificación entre contenido humano y generado por IA, evaluando la eficacia y limitaciones de los modelos actuales en la detección.

Para llevar a cabo esta tarea, se emplean distintos algoritmos de aprendizaje profundo, incluyendo redes neuronales y *Transformers* [1], con el fin de maximizar la precisión de los modelos. En primer lugar, se realiza un análisis exploratorio de las frecuencias de palabras y la longitud de los textos para observar las diferencias iniciales entre ambos tipos de contenido. Posteriormente, los textos son preprocesados y normalizados, y se aplican modelos de codificación como *TF-IDF* [2] y *Bag of Words* [3], además de técnicas avanzadas de vectorización, con el objetivo de mejorar la discriminación entre textos.

Los resultados obtenidos demuestran que existen diferencias significativas en términos de longitud, estilo y uso de vocabulario entre textos generados por IA y aquellos creados por humanos. Estos hallazgos permiten optimizar modelos y mejorar su capacidad predictiva, revelando fortalezas y desafíos en el campo de la detección de contenido generado por IA.

Keywords: aprendizaje profundo, *Transformers*, procesamiento del lenguaje natural, *Embedding*, detección de autoría.

2. Introducción

En los últimos años, los avances en inteligencia artificial y en procesamiento del lenguaje natural han permitido que modelos de generación de texto produzcan contenido coherente y fluido, comparable al escrito por humanos. Sin embargo, esta capacidad plantea retos importantes, como la detección de textos generados por IA, especialmente en áreas donde la autenticidad del contenido es crítica, como la educación, los medios y las redes sociales.

Este trabajo se enfoca en desarrollar y evaluar un método para clasificar textos según su autoría, ya sea humana o artificial, empleando un enfoque de aprendizaje supervisado. Para ello, se utilizan técnicas de preprocesamiento y vectorización de texto, junto con modelos de clasificación avanzados como redes neuronales y encoders basados en *Transformers*, para identificar patrones que diferencien textos humanos de aquellos generados por IA.

A través de varios experimentos, se compara el rendimiento de estos modelos en términos de precisión y eficacia. Este estudio no solo busca mejorar los métodos de detección de textos generados por IA, sino también proporcionar una perspectiva sobre los desafíos y oportunidades en esta área en crecimiento.

3. Metodología

Para abordar el problema de clasificación de textos según su fuente de generación se emplea una metodología basada en el aprendizaje supervisado, que incluye distintas etapas de preprocesamiento y pruebas con modelos avanzados de clasificación. La metodología se desarrolla en los siguientes pasos clave:

i. **Preprocesamiento de Datos**: Se obtienen características generales de los datos, como el tipo de dato de cada columna, la cantidad de valores nulos y la cantidad de valores duplicados. Además, para evitar sesgos en los modelos, el *dataset* [4] es balanceado seleccionando una cantidad igual de textos generados por humanos y por IA.

ii. **Análisis Exploratorio:** Se realiza un análisis exploratorio para identificar diferencias iniciales en los textos, como la distribución de la longitud y la frecuencia de palabras. Este análisis permite observar patrones que pueden indicar diferencias estilísticas y estructurales como también permite sacar alguna información valiosa que será utilizada posteriormente, como la máxima longitud por secuencia o la correlación de cada tipo de texto por número de palabras.

iii. **Vectorización y Representación de Texto**: Los textos se someten a un proceso de limpieza y normalización, que incluye la eliminación de caracteres especiales, la conversión a minúsculas y la eliminación de *stopwords.*
Para transformar los textos en representaciones numéricas, se utilizan diferentes métodos de vectorización, como el modelo de bolsa de palabras (*Bag of Words*), el modelo *TF-IDF* con bigramas o el modelo de números enteros. Todos ellos seleccionando también el tamaño del vocabulario.

iv. **Selección de Modelos de Clasificación**: Se implementan y comparan varios modelos de clasificación, incluyendo redes neuronales de capas densas y un encoder de tipo *Transformer* usando tanto un *Embedding* [5] previo en un caso, como un *Positional Embedding* en otro. Cada modelo se entrena utilizando una división de los datos en conjuntos de entrenamiento, validación y prueba, con el fin de optimizar su rendimiento y minimizar el sobreajuste.

v. **Evaluación y Validación con Nuevos Textos**: Los modelos se evalúan en términos de precisión, exactitud y sensibilidad utilizando el conjunto de prueba. Finalmente, se realiza una validación de los modelos utilizando nuevos textos de distintas fuentes, tanto de IA como de humanos. Esta etapa permite evaluar la robustez de los modelos frente a contenido no visto previamente, proporcionando una visión clara de su aplicabilidad en escenarios reales.

4. Experimentos y Resultados

Para evaluar la efectividad de los modelos se llevaron a cabo diversos experimentos utilizando los modelos propuestos. A continuación, se describen los experimentos realizados y los resultados obtenidos.

Primero, el *dataset* fue dividido en un conjunto de entrenamiento (70% de los datos), un conjunto de validación (15%) y un conjunto de prueba (15%). La etapa de entrenamiento permite ajustar los parámetros del modelo, mientras que la validación permite seleccionar los hiperparámetros óptimos, además de minimizar el riesgo de sobreajuste. Finalmente, el conjunto de prueba se utilizó para evaluar el rendimiento final de cada modelo.

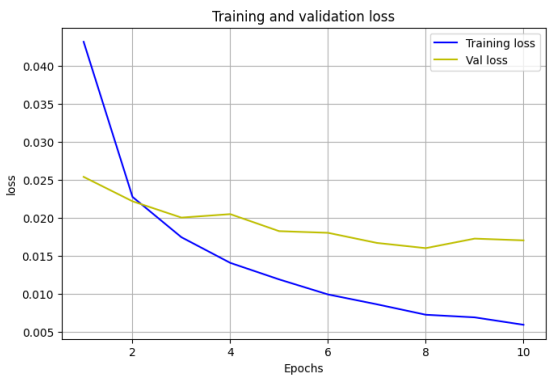
Posteriormente, se entrenaron y evaluaron cuatro modelos de clasificación principales. Para todos los modelos se ha elegido *rmsprop* como optimizador, *binary_crossentropy* como función de *loss* y *accuracy* como métrica principal.

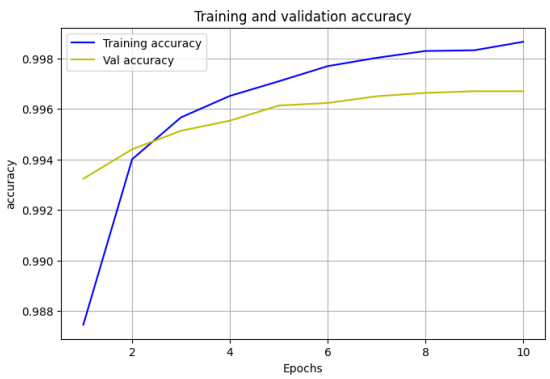
1) Red neuronal simple con codificación *Bags of Words*

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 30000)	0
dense (Dense)	(None, 32)	960,032
dropout (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 1)	33

Para este modelo se ha elegido una codificación de tipo bolsa de palabras con un tamaño de vocabulario de 30.000 *tokens*.
El modelo consta de una capa densa de 32 neuronas con activación *ReLU*, una capa *Dropout*, y por último, una capa densa como salida de 1 neurona con activación *Sigmoid* para realizar la clasificación binaria.

Historial de entrenamiento de este modelo con 10 *epochs:*





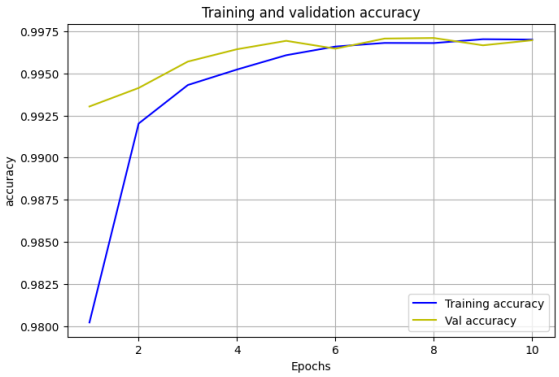
Este modelo logra resultados aceptables en términos de precisión y sensibilidad, alcanzando una precisión en *test* de 99,75%, con prácticamente nada de *overfitting*. Sin embargo, su capacidad para captar la semántica y el contexto es limitada, como se verá posteriormente.

2) Red neuronal simple con codificación *TF-IDF* con bigramas

Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 30000)	0
dense_2 (Dense)	(None, 32)	960,032
dropout_1 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 1)	33

Este modelo es prácticamente igual que el anterior, con las mismas capas y mismo tamaño del vocabulario. Únicamente se ha cambiado la codificación de las palabras a una *TF-IDF* con bigramas.

Historial de entrenamiento de este modelo con 10 *epochs:*



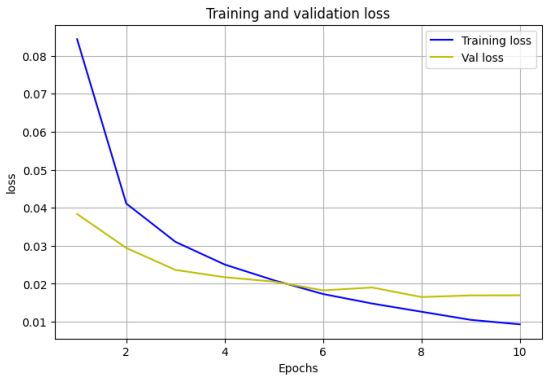
Con este modelo se logra unos resultados bastante parecidos al anterior, alcanzando una precisión de 99,69% en *test*, reduciendo todavía más el *overfitting*.

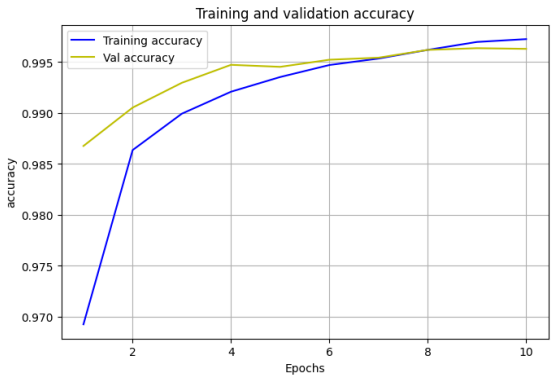
3) *Transformer* encoder con codificación de números enteros

Layer (type)	Output Shape	Param #
input_layer_4 (InputLayer)	(None, None)	0
embedding (Embedding)	(None, None, 128)	3,840,000
transformer_encoder (TransformerEncoder)	(None, None, 128)	140,832
global_max_pooling1d (GlobalMaxPooling1D)	(None, 128)	0
dropout_3 (Dropout)	(None, 128)	0
dense_6 (Dense)	(None, 1)	129

Para este modelo se ha utilizado una codificación de números enteros con 30.000 *tokens* de vocabulario, al igual que las anteriores. Además, se ha elegido 500 *tokens* como máxima longitud de secuencia.
El modelo consta de una capa *Embedding* con 128 dimensiones, seguida de un *TransformerEncoder* con función de activación *ReLU*, 32 neuronas para la capa densa y 2 *heads.* Posteriormente, se añade un *GlobalMaxPooling1D*, una capa *Dropout*, y por último, una capa densa de 1 neurona como salida con activación *Sigmoid* para realizar la clasificación binaria.

Historial de entrenamiento de este modelo con 10 *epochs:*





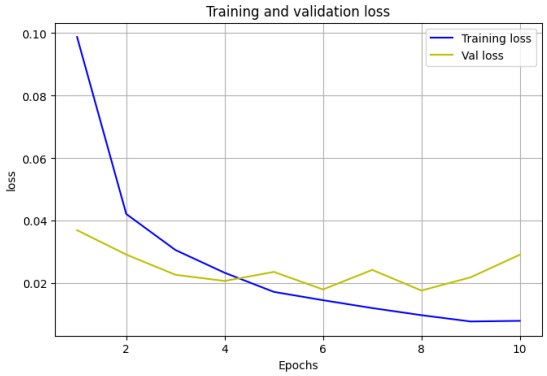
Con este modelo se mantienen los resultados bastante buenos de los modelos anteriores, alcanzando una precisión de 99,66% en *test*.

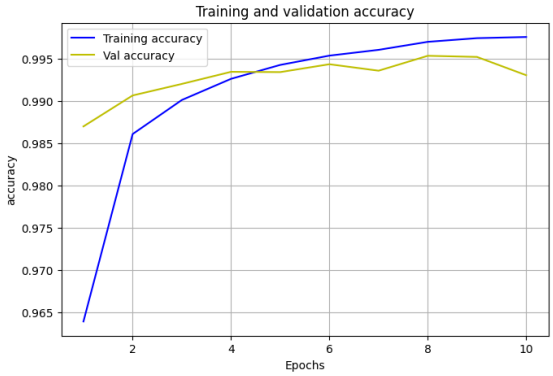
4) *Transformer* encoder y *PositionalEmbedding* con codificación de números enteros

Layer (type)	Output Shape	Param #
input_layer_8 (InputLayer)	(None, None)	0
positional_embedding (PositionalEmbedding)	(None, None, 256)	10,368,000
transformer_encoder_1 (TransformerEncoder)	(None, None, 256)	543,776
global_max_pooling1d_1 (GlobalMaxPooling1D)	(None, 256)	0
dropout_6 (Dropout)	(None, 256)	0
dense_11 (Dense)	(None, 1)	257

Para este modelo se ha utilizado una codificación de números enteros con 40.000 *tokens* de vocabulario, aumentando en 10.000 *tokens* con respecto a los modelos anteriores. Además, se ha elegido 500 *tokens* como máxima longitud de secuencia.
El modelo consta de una capa *PositionalEmbedding* con 256 dimensiones, introduciendo la información posicional de los textos en la red, algo que debería mejorar el rendimiento del modelo. A parte de este cambio, todas las capas posteriores de la red se mantienen inalteradas respecto al modelo anterior.

Historial de entrenamiento de este modelo con 10 *epochs:*





Los resultados siguen siendo buenos, aunque la precisión es ligeramente inferior a modelos anteriores, alcanzando un 99,3% en *test*.

Todos los modelos han demostrado tener muy buen rendimiento evaluándose con los datos de prueba, alcanzando precisiones por encima de 99%, algo muy positivo. Sin embargo, cuando se intenta predecir la autoría de un texto que no está en el *dataset* inicial los resultados son muy diferentes dependiendo del modelo.

Para realizar estas pruebas se utilizarán textos de toda índole, además de una versión alternativa a los mismos escritos por una inteligencia artificial online. Estos son los textos elegidos:

a. Texto escrito a mano por mí.
b. *The Lord of the Rings* [6], Book 5 Chapter 3, Primeros 3 párrafos.
c. Fragmento de la *Wikipedia* de *Albert Einstein* [7].
d. *Automatic lateral control for unmanned vehicles via genetic algorithms* [8], 5. *Conclusions and future works* (2011).
e. Fragmento de la noticia *How Donald Trump took the Republican Party by storm* de la CNN (2015) [9].

A continuación, se presenta el rendimiento de cada modelo con estos textos. Las tablas muestran la probabilidad predicha por cada modelo de que el texto haya sido generado por IA. De esta manera, en la tabla superior valores cercanos a 0% representan una buena clasificación. Por otro lado, en la tabla inferior una buena clasificación implica valores cercanos a 100%, pues han sido generados por inteligencia artificial.

Predicción de cada modelo sobre textos escritos por humanos

	TEXTO A	TEXTO B	TEXTO C	TEXTO D	TEXTO E
MODELO 1	17,03 %	7,04 %	92,68 %	-	-
MODELO 2	100 %	100 %	100 %	-	-
MODELO 3	0,12 %	0,01 %	0 %	0,04 %	0 %
MODELO 4	0,02 %	0,03 %	13,61 %	31,62 %	0,1 %

Predicción de cada modelo sobre textos reescritos por IA

	TEXTO A	TEXTO B	TEXTO C	TEXTO D	TEXTO E
MODELO 1	100 %	100 %	100 %	-	-
MODELO 2	100 %	100 %	100 %	-	-
MODELO 3	99,98 %	7,54 %	0 %	53,09 %	19,41 %
MODELO 4	99,72 %	99,92 %	93,17 %	100 %	99,65 %

El Modelo 1) presenta un rendimiento inconsistente con textos humanos, con valores de predicción que oscilan entre 17,03% y 92,68%, lo que indica que tiene dificultades para identificar textos escritos por humanos. Por otro lado, predice correctamente los textos generados por IA con un 100% en todos los casos, mostrando eficacia en esta categoría.

El Modelo 2) clasifica incorrectamente todos los textos humanos, con predicciones constantes del 100%. Esto refleja un sesgo hacia la clasificación de IA. Por otro lado, alcanza un 100% de precisión en los textos de IA, pero su incapacidad para clasificar textos humanos lo hace poco confiable.

El Modelo 3) logra buenos resultados en textos humanos, con valores bajos cercanos al 0% en la mayoría de los casos. En la clasificación de IA, el rendimiento es inconsistente, con valores que oscilan entre 99,98% y 7,54%, lo que sugiere problemas para identificar algunos textos generados por IA.

El Modelo 4) clasifica correctamente la mayoría de los textos humanos con valores bajos, aunque tiene dificultades con algunos textos específicos (13,61% y 31,62%). Por otro lado, muestra un rendimiento sólido en la detección de textos de IA, con valores cercanos o iguales al 100% en casi todos los textos.

5. Conclusiones

Los resultados obtenidos en este estudio destacan la efectividad y confiabilidad del Modelo *Transformer* Enconder con *Positional Embedding* (Modelo 4)) como la mejor opción para la detección de autoría en textos generados por humanos y por IA. A lo largo de las pruebas con datos nuevos, este modelo demostró ser el más equilibrado, alcanzando una alta precisión tanto en la clasificación de textos humanos como en aquellos generados por IA. Esta capacidad para mantener un rendimiento consistente frente a datos no vistos previamente lo posiciona como el modelo más robusto para aplicaciones en entornos reales, donde la detección precisa y confiable es esencial.

En comparación, los otros modelos presentaron limitaciones significativas. En particular, el Modelo 2) mostró un rendimiento deficiente en la clasificación de textos humanos, ya que los etiquetó incorrectamente en su totalidad, lo que indica una falta de generalización y una posible dependencia de características específicas que no son representativas de los textos humanos en general. El presente trabajo evidencia que los modelos basados en arquitecturas avanzadas son superiores en la detección de la fuente de generación de textos, al capturar de manera efectiva las diferencias estilísticas y contextuales entre los textos de IA y los textos humanos.

Aunque el Modelo 4) emerge como la opción más confiable para aplicaciones de clasificación de autoría, sigue mostrando ciertas dificultades dependiendo del texto a predecir. Así, hay varias mejoras que se pueden realizar para mejorar todavía más su precisión. Entre estas, destacan el aumento del tamaño del vocabulario, el mantenimiento de las *stopwords* a la hora de entrenar el modelo, la adición de capas a la red (densas o *MultiHeadAttention*) o hacer pruebas con todos los textos del *dataset*, generando sintéticamente más textos de IA para mantenerlo balanceado.

Para finalizar, aunque el Modelo 4) ha demostrado ser el más adecuado y equilibrado, estas mejoras tienen el potencial de fortalecer aún más su rendimiento y confiabilidad. A medida que los modelos avanzan en su capacidad para identificar diferencias estilísticas y semánticas entre textos humanos e IA, se abren nuevas posibilidades para aplicaciones prácticas en autenticación de contenido y detección de textos generados artificialmente.

6. Referencias

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2023). *Attention is all you need*. arXiv.
2. Robertson, Stephen. (2004). *Understanding Inverse Document Frequency: On Theoretical Arguments for IDF*. Journal of Documentation - J DOC. 60. 503-520.
3. Qader, Wisam & M. Ameen, Musa & Ahmed, Bilal. (2019). *An Overview of Bag of Words; Importance, Implementation, Applications, and Challenges*. 200-204.
4. Gerami, S. (2024). *AI vs Human Text Dataset.* Kaggle. https://www.kaggle.com/datasets/shanegerami/ai-vs-human-text/data
5. Almeida, F., & Xexéo, G. (2023). *Word embeddings: A survey*. arXiv.
6. Tolkien, J. R. R. (2005). *The Lord of the Rings*. HarperCollins. https://gosafir.com/mag/wp-content/uploads/2019/12/Tolkien-J.-The-lord-of-the-rings-HarperCollins-ebooks-2010.pdf
7. Wikipedia. (n.d.). *Albert Einstein*. https://en.wikipedia.org/wiki/Albert_Einstein
8. Onieva, E., Naranjo, J. E., Milanés, V., Alonso, J., García, R., & Pérez, J. (2011). *Automatic lateral control for unmanned vehicles via genetic algorithms*. Centro de Automática y Robótica (UPM-CSIC).
9. Collinson S. (2015, December 15). *How Donald Trump took the Republican Party by storm*. CNN. https://edition.cnn.com/2015/12/14/politics/donald-trump-republican-party-history/index.html