

Detect if an academic monograph was made by an Artificial Intelligence system, using Machine Learning

Gloria T. Huamaní Huamaní, Dra.¹, Paul M. Tocto Inga, Dr.²

^{1,2}Universidad Nacional de Ingeniería, Perú, gloria.huamani.h@uni.edu.pe, ptocto@uni.edu.pe

Abstract– This research addresses the use of artificial intelligence (AI) in the educational context, specifically in the detection of academic works generated by AI systems in English. Due to the current development of AI, concerns arise about intellectual property and the quality of learning as long as AI systems generate the monographs that students should complete, this project seeks to propose a solution to this problem, for which the steps are detailed. to design and train a recurrent neural network using TensorFlow and the evaluation of the model using classification metrics such as precision, recall and F1-score, the results obtained show that the created model has good performance in the classification of monographic works generated by AI, with high precision, recall and F1-score values. It is concluded that the model is effective in identifying these jobs.

Keywords--AI, Tensor Flow, Recurrent Neural Network, Academic work.

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

Detectar si una monografía académica fue realizada por un sistema de Inteligencia Artificial, usando Machine Learning

Gloria T. Huamaní Huamaní, Dra.¹, Paul M. Tocto Inga, Dr.²

^{1,2}Universidad Nacional de Ingeniería, Perú, gloria.huamani.h@uni.edu.pe, ptocto@uni.edu.pe

Resumen.- Esta investigación aborda el uso de la inteligencia artificial (IA) en el contexto educativo, específicamente en la detección de trabajos académicos generados por sistemas de IA en Inglés. Por el desarrollo actual de la IA, surgen preocupaciones sobre la propiedad intelectual y la calidad del aprendizaje en tanto los sistemas IA generen las monografías que deberían realizar los estudiantes, este proyecto busca plantear una solución a esta problemática, para lo cual se detallan los pasos para diseñar y entrenar una red neuronal recurrente usando TensorFlow y la evaluación del modelo usando métricas de clasificación como precisión, recall y F1-score, los resultados obtenidos muestran que el modelo creado tiene un buen rendimiento en la clasificación de trabajos monográficos generados por IA, con altos valores de precisión, recall y F1-score. Se concluye que el modelo es efectivo para identificar estos trabajos.

Keywords—IA, Tensor Flow, Red Neuronal Recurrente, Trabajo académico.

I. INTRODUCCIÓN

La UNESCO tiene como uno de sus objetivos garantizar la utilización de las tecnologías de la Inteligencia artificial (IA) en el contexto educativo bajo los principios fundamentales de inclusión y equidad, asimismo “está decidida a ayudar a los Estados Miembros para que saquen provecho del potencial de las tecnologías de la IA con miras a la consecución de la Agenda de Educación 2030”. [1] con la aparición de herramientas de IA los estudiantes universitarios y docentes lo utilizan cada vez más para optimizar su tiempo y mejorar la productividad, sin embargo está en cuestionamiento acerca de los logros de aprendizaje, también el respeto por la propiedad intelectual, toda vez que se presenta un texto sin diferenciar lo que escribe un ser humano respecto a la copia del texto desarrollado por IA.

La inteligencia artificial es un nuevo paradigma, y las instituciones educativas están expuestas al riesgo de que los alumnos presenten sus trabajos académicos creados por sistemas de inteligencia artificial. Creando sistemas que detecten si el trabajo monográfico fue desarrollado por un sistema de inteligencia artificial, las instituciones educativas detectarán dicha práctica. Los beneficiarios serían todas las instituciones educativas, a nivel nacional e internacional, en particular la Universidad Nacional de Ingeniería. [2] [3] [4]

El objetivo de este trabajo es detectar si una investigación fue realizada por una Inteligencia Artificial (IA) utilizando el aprendizaje automático (Machine Learning) con una precisión mayor al 80%.

El estudio es importante por diversas razones: a) capacidades y limitaciones. Los humanos producen textos con pensamiento crítico, a partir de su experiencia de vida. Mientras ChatGTP opera a partir de patrones en datos textuales, genera respuestas a partir de la base de datos de textos y algoritmos complejos. b) ética y responsabilidad. Los humanos responden de sus actos y decisiones. Mientras que ChatGPT4 no tiene responsabilidad moral o legal. c) Respecto a la interacción y comunicación, el ChatGPT4 responde a la actualización de sus bases de datos, frente a situaciones novedosas responde que hasta actualizado hasta tal periodo. La interacción en seminarios, foros podrían no estar sistematizados y la conversación queda allí. d) respecto al aprendizaje y adaptación los seres humanos se retroalimentan constantemente se adaptan a través de experiencias, en tanto ChatGTP aprende lentamente, sus actualizaciones y ajustes dependen de sus algoritmos alimentados por seres humanos. e) Si nos referimos a la creatividad y empatía, los seres humanos procesan información además lo contextualizan, generan nuevas ideas, en tanto ChaGTP puede simular.[5]

“En resumen, comprender las diferencias entre las capacidades humanas y las de una IA como ChatGPT4 es esencial para utilizar estas herramientas de manera efectiva y ética, maximizando sus beneficios mientras se minimizan los riesgos y malentendidos.”

OPENAI ha desarrollado herramientas para detectar AI Text classifier [6] “El proceso implica Machine Learning (aprendizaje automático), que es una rama de la IA, en el que se entrena a un modelo con una gran cantidad de datos. Este modelo identifica patrones de contenido escrito por humanos de patrones de contenido generado por IA. Una vez entrenado, el modelo podrá analizar nuevos textos y predecir si fueron escrito por humanos o por máquinas.” Puede identificar: repeticiones de contenido, patrones de uso de palabras, estructuras de las oraciones y tokens generados.

Otro detector es Copyleaks AI Content Detector, también Eleuther IA ha desarrollado GPTZero Classic. [7]

A. La inteligencia artificial y el aprendizaje automático

La inteligencia artificial y el aprendizaje automático han avanzado exponencialmente, hay diversas aplicaciones. Tocto, Huamaní [2] [3] [4] han desarrollado modelos de predicción, de clasificación a partir de los algoritmos de machine learning y los datos de los alumnos. Un modelo de clasificación es importante, en tanto permite identificar el objeto de estudio, el estilo de redacción, una aplicación importante es clasificar si el texto ha sido desarrollado por humanos o por IA. Salah et.al [8] afirma: “las interacciones generadas por IA es su versatilidad y escalabilidad. ChatGPT puede generar diálogos entre personajes controlados por IA que exhiben un amplio espectro de opiniones, creencias y actitudes. “ esto permitiría “a los investigadores crear escenarios controlados que podrían ser difíciles, si no imposibles, de construir en entornos del mundo real. Además, estas interacciones pueden replicarse y modificarse a escala, permitiendo la ejecución de estudios a gran escala sin comprometer la consistencia de los parámetros de interacción.” Por otra parte Odri y Yun [9] indican que la distinción entre textos generados por humanos y textos generados por IA se vuelven cada vez más difíciles, sin embargo, Deziel, citado por Ruopen et al [10] afirma “Dadas las infinitas variaciones en el contenido y los estilos del texto, usar solo pasajes podría ser insuficiente para identificar de forma fiable texto generado por máquina” Hay diversos estudios para Optimizar los modelos, detectar si ha sido elaborado por máquina o por humanos. Los chatbots de IA como ChatGPT están revolucionando nuestras capacidades de IA, especialmente en la generación de texto, para ayudar a acelerar muchas tareas, pero introducen nuevos dilemas la detección de texto generado por IA se ha convertido en un tema de gran debate. [11],[12],[13]

Frente a ello considerando las limitaciones conocidas e inesperadas del detector de texto AI [11] evalúan el escenario opuesto, medir la capacidad de la herramienta de detección para discriminar texto generado por humanos.

Por otra parte, en el diseño y optimización de los modelos de aprendizaje automático se debe considerar la etapa de la selección adecuada de los hiperparámetros, porque es imprescindible para el éxito de los proyectos de Machine Learning. Son una parte integral del diseño y optimización de modelos en el aprendizaje automático, y su selección adecuada es esencial para el éxito de los proyectos de Machine Learning.

También al crear los modelos es importante diseñar y optimizar modelos de aprendizaje automático mediante hiperparámetros óptimos para lograr modelos altamente

eficientes. Para esto se requiere algunos pasos esenciales: a) comprender el problema, permite elegir el modelo adecuado y determinar los hiperparámetros más relevantes b) Seleccionar el modelo adecuado para resolver el problema. Por ejemplo un modelo de red neuronal requiere hiperparámetros relacionado con la arquitectura de red, mientras que un modelo de máquina de soporte vectorial tendrá hiperparámetros relacionados con el kernel y la regularización c) Identificar los hiperparámetros clave, conocer el que tiene más influencia en el modelo d) Ajuste inicial de hiperparámetros. Determinar un conjunto inicial de hiperparámetros basado en el conocimiento, experiencia tomando en cuenta el rendimiento del modelo y los recursos computacionales e) Optimizar el modelo utilizando técnicas como la búsqueda en cuadrícula o búsqueda aleatoria o se puede utilizar el método bayesiano con la finalidad de explorar el espacio de hiperparámetros y encontrar la combinación óptima f) validación cruzada g) Evaluación h) mejora continua e i) automatización j) documentación y registro. [5]. El ajuste de un hiperparámetro requiere comprensión del modelo, intuición y experimentación sistemática. En tanto el objetivo es encontrar un equilibrio entre rendimiento, complejidad del modelo y la eficiencia computacional

Además de esta sección presentamos la colección de datos y definición de variables, metodología, experimento, análisis de resultados, conclusiones y recomendaciones.

II. COLECCIÓN DE DATOS Y DEFINICIÓN DE VARIABLES

Los datos usados en el presente Proyecto fueron publicados en Kaggle [14], creados para Realizar el entrenamiento de un modelo que detecte si un trabajo monográfico en inglés ha sido creado por un Sistema de Inteligencia Artificial, tiene 73,573, de los cuales 46,203 han sido generados por un Sistema de inteligencia artificial y 27,370 creados por humanos. Algunas de las fuentes consideradas en los datos se muestran en [15,16,17,18,19,20]. Kaggle plataforma fundada por Goldbloom y Ben Hammer en 2016, adquirida por Google sirve como punto de encuentro para científicos de datos y profesionales del aprendizaje automático. Por otra parte los usuarios pueden encontrar recursos como Kernels(scripts o cuadernos Jupyter). Asimismo esta disponible en Kaggle el “Persuade Corpus” que contiene anotaciones para el reconocimiento de actos de habla persuasivos en texto. Corpus está diseñado para ser utilizado en procesamiento de lenguaje natural (PLN).

:

La estructura de los datos obtenidos para el entrenamiento es la indicada en la Tabla I, para el entrenamiento se consideró solo los atributos: text y label.

TABLA I
ESTRUCTURA DE LOS DATOS

Nro	Atributo	Descripción
1	text	Texto de la monografía
2	label	(1 Generado por IA, 0 No Generado por IA)
3	prompt_name	Tema
4	source	Origen del dato

III. METODOLOGÍA

La preparación de los datos, entrenamiento y validación del modelo se realizó usando el lenguaje Python, en la plataforma Colaboratory de Google, considerando la siguiente metodología, ver la fig. 1:



Fig. 1 Metodología desarrollada

El detalle de la metodología seguida lo tenemos en los siguientes pasos:

Paso 1. Subir los datos indicados en [11] a la plataforma de Colaboratory de Google en la nube

Paso 2.- Separar el 20% del total de los datos para la validación del modelo.

Paso 3.- Preparar los datos convirtiendo los datos de texto contenidos de un formato numérico a cadenas de caracteres utilizando la codificación UTF-8, usando librería TensorFlow, según (1) y (2).

```
train_datapr = [str(text.numpy(), 'utf-8') for text, _ in train_data2] (1)
```

```
test_datapr = [str(text.numpy(), 'utf-8') for text, _ in test_data2] (2)
```

Paso 4.- Crear un tokenizer (vectorizador de texto) que convierte texto en secuencias de números enteros, limitando el vocabulario a las 10,000 palabras más frecuentes y asegurándose de que todas las secuencias tengan una longitud máxima de 200 enteros. Luego, se adaptó este tokenizer al conjunto de datos de entrenamiento, construyendo un vocabulario interno basado en esos datos, según (3) y (4).

```
tokenizer = tf.keras.layers.TextVectorization(max_tokens=10000, output_mode='int', output_sequence_length=200) (3)
```

```
tokenizer.adapt(train_datapr) (4)
```

Paso 5.- Usar el tokenizer creado previamente en el paso anterior, para convertir el texto en secuencias de números enteros tanto en el conjunto de datos de entrenamiento como en el de prueba. Esto es necesario para preparar los datos de texto para su alimentación a modelos de aprendizaje automático que requieren datos numéricos en lugar de texto, según (5) y (6).

```
train_datapt = tokenizer(train_datapr) (5)
```

```
test_datapt = tokenizer(test_datapr) (6)
```

Paso 6.- Definir un modelo de red neuronal recurrente (RNN) con una capa de incrustación, una capa bidireccional de LSTM, seguida de dos capas densas para procesar y clasificar datos de secuencia, y finalmente, producir una salida binaria para el problema de clasificación binaria, de identificar el texto creado por un sistema de inteligencia artificial, según (7).

```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(input_dim=len(tokenizer.get_vocabulary()), output_dim=64, mask_zero=True),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid') # Salida binaria para clasificación
]) (7)
```

Paso 7.- Compilar el modelo de red neuronal, creado previamente en el paso anterior, utilizando el optimizador Adam, la función de pérdida de entropía cruzada binaria y la métrica de precisión, lo que configura el modelo para el entrenamiento y la evaluación en un problema de clasificación binaria, según (8).

```
model.compile(optimizer='adam',
loss='binary_crossentropy', metrics=['accuracy']) (8)
```

Paso 8.- Entrenar el modelo de red neuronal definido anteriormente, utilizando el conjunto de datos de entrenamiento y las etiquetas correspondientes durante 5 épocas, utilizando lotes de 64 muestras en cada paso de entrenamiento. Además, que el modelo evalúe en el conjunto de datos de prueba después de cada época de entrenamiento para monitorear su desempeño durante el entrenamiento, según (9).

```
model.fit(train_datapt2, train_labels, epochs=5,
batch_size=64, validation_data=(test_datapt2, test_labels)) (9)
```

Paso 9.- Después de entrenar según los pasos indicados anteriormente se validó el modelo con las métricas de precisión, recall, puntaje F1 y soporte para cada clase (en este caso, "Negativo" y "Positivo"), así como la precisión global, el recall y el puntaje F1 del modelo en general, según (10).

```
classification_rep = classification_report(y_true,
y_pred_binary, target_names=["Negativo", "Positivo"]) (10)
```

Descripción de los indicadores

Precisión (Precisión):

La precisión se calcula como el número de verdaderos positivos dividido por la suma de verdaderos positivos y falsos positivos, esta proporción de predicciones positivas que fueron correctas indican una medida de la exactitud del modelo cuando predice una clase específica. Un valor alto de precisión (cerca de 1) indica que el modelo hace pocas predicciones incorrectas de falsos positivos.

Recall (Recuperación o Sensibilidad):

El recall se calcula como el número de verdaderos positivos dividido por la suma de verdaderos positivos y falsos negativos, indicando la proporción de instancias positivas reales que fueron correctamente clasificadas por el modelo .Es una medida de la capacidad del modelo para encontrar todas las instancias de una clase. Un valor alto de recall (cerca de 1) indica que el modelo identifica la mayoría de las instancias positivas reales.

F1-score (Puntaje F1):

El puntaje F1 es la media armónica de precisión y recall. Proporciona un equilibrio entre precisión y recall, proporciona

una métrica balanceada entre ambas. Un puntaje F1 alto (cerca de 1) indica un buen equilibrio entre precisión y recall

Support (Soporte):

El soporte es el número de instancias reales de cada clase en el conjunto de datos de prueba.

Accuracy (Precisión Global):

La precisión global es la proporción de instancias correctamente clasificadas sobre el total de instancias en el conjunto de datos de prueba, indicando la proporción de predicciones correctas en general sobre todas las clases. Un alto valor de precisión global (cerca de 1) indica un buen rendimiento general del modelo en la clasificación de todas las clases.

Macro avg (Media Macro):

La media macro es el promedio no ponderado de precisión, recall y F1-score para todas las clases.

Weighted avg (Media Ponderada):

La media ponderada es similar a la media macro, pero tiene en cuenta el soporte de cada clase.

IV. RESULTADOS

Después de seguir todos los pasos de la metodología, se obtuvieron los siguiente resultados de validación, ver Tabla II, en las siguientes métricas de clasificación que fueron generadas por la función classification_report de la biblioteca scikit-learn, después de evaluar al modelo de clasificación en un conjunto de datos de prueba previamente separadas en el paso 2. Las métricas obtenidas son las siguientes:

TABLA II
INDICADORES DE VALIDACIÓN DEL MODELO

Métricas de Clasificación:				
	precision	recall	f1-score	support
Negativo	0.96	0.93	0.94	5394
Positivo	0.96	0.97	0.97	9321
accuracy			0.96	14715
macro avg	0.96	0.95	0.96	14715
weighted avg	0.96	0.96	0.96	14715

Precisión (Precisión):

En este caso, para la clase "Negativo", la precisión es 0.96, lo que significa que el 96% de las instancias que el modelo predijo como "Negativo" fueron realmente "Negativo". Para la clase "Positivo", la precisión es también 0.96, lo que indica que el 96% de las instancias que el modelo predijo como "Positivo" fueron realmente "Positivo".

Recall (Recuperación o Sensibilidad):

En este caso, para la clase "Negativo", el recall es 0.93, lo que significa que el 93% de todas las instancias "Negativo" en el conjunto de datos de prueba fueron correctamente identificadas por el modelo. Para la clase "Positivo", el recall es 0.97, lo que indica que el 97% de todas las instancias "Positivo" fueron correctamente identificadas por el modelo.

F1-score (Puntaje F1):

En este caso, para la clase "Negativo", el puntaje F1 es 0.94, mientras que para la clase "Positivo" es 0.97.

Support (Soporte):

En este caso, hay 5394 instancias de la clase "Negativo" y 9321 instancias de la clase "Positivo".

Accuracy (Precisión Global):

En este caso, la precisión global del modelo es 0.96, lo que significa que el 96% de todas las instancias en el conjunto de datos de prueba fueron correctamente clasificadas por el modelo.

Macro avg (Media Macro):

En este caso, la precisión, recall y F1-score macro promedio son 0.96, 0.95 y 0.96 respectivamente.

Weighted avg (Media Ponderada):

En este caso, la precisión, recall y F1-score ponderados promedio son 0.96, 0.96 y 0.96 respectivamente.

En resumen, estas métricas de clasificación proporcionan una evaluación detallada del rendimiento del modelo en la clasificación de dos clases ("Negativo" y "Positivo") en el conjunto de datos de prueba

V. ANÁLISIS DE RESULTADOS

Basándonos en las métricas proporcionadas por la función `classification_report`, podemos extraer varias conclusiones sobre el rendimiento del modelo de clasificación en el conjunto de datos de prueba. Aquí están algunas conclusiones comunes que se pueden derivar:

Precisión (Precisión):

En este caso, tanto para la clase "Negativo" como para la clase "Positivo", la precisión es alta (0.96), lo que sugiere que el modelo tiene una alta precisión en la predicción de ambas clases.

Recall (Recuperación o Sensibilidad):

En este caso, el recall es alto para ambas clases: 0.93 para "Negativo" y 0.97 para "Positivo", lo que indica que el modelo identifica la gran mayoría de las instancias reales de ambas clases.

F1-score (Puntaje F1):

En este caso, el puntaje F1 es alto tanto para "Negativo" (0.94) como para "Positivo" (0.97), lo que indica un buen equilibrio entre precisión y recall para ambas clases.

Accuracy (Precisión Global):

En este caso, la precisión global es 0.96, lo que indica que el modelo tiene un alto rendimiento general en la clasificación.

En general, las conclusiones basadas en estas métricas sugieren que el modelo tiene un buen rendimiento en la clasificación de las clases "Negativo" y "Positivo" en el conjunto de datos de prueba, con altos valores de precisión, recall y puntaje F1, así como una alta precisión global.

VI. CONCLUSIONES Y RECOMENDACIONES

A. Conclusiones

- Se creó un modelo de una red neuronal, que permite clasificar si un trabajo monográfico en inglés fue creado por un Sistema de inteligencia artificial.
- Las métricas indican que el modelo tiene un buen rendimiento en la clasificación de las clases "Negativo" y "Positivo" en el conjunto de datos de prueba, con altos valores de precisión, recall y puntaje F1, así como una alta precisión global.
- Se plantea una metodología para crear modelos que permitan detectar si un trabajo monográfico ha sido creado por un sistema de inteligencia artificial.
- Este modelo permitiría elevar la calidad del trabajo de los estudiantes, incrementado la productividad del docente universitario.

B. Recomendaciones

- Considerar la creación de modelos en otros idiomas.
- Tomar este estudio como punto de inicio para futuras investigaciones relacionados a la identificación de trabajos monográficos creados por sistemas inteligentes.

REFERENCIAS

- [1] UNESCO. Inteligencia artificial en la educación (s.f.). <https://es.unesco.org/themes/ticeduacion/inteligencia-artificial>
- [2] P. Tocto, G. Huamaní y L. Zuloaga. Construcción de un modelo basado en redes neuronales para determinar la duración de los estudios de ingeniería en una universidad pública en el Perú. 19th LACCEI International Multi-conference for engineering, education and technology, 2021.
- [3] P. Tocto, G. Huamaní y L. Villacorta. 20 Application of artificial intelligence in the management of a public university in Peru: A case of supervised machine learning using neural networks to classify if an engineering student would graduate in 5 years LACCEI International Multi-conference for engineering, education and technology. 2022.

- [4] P. Tocto, G. Huamaní. Machine learning application in university management: Classification model Dropping out of engineering students in Peru LACCEI 2023 [5]OpenAI. ChatGTP4.
- [5] Dolo. AI Text Classifier: Distinguir texto humano y generado por IA
- [6] <https://www.globalcobots.com/ai-text-classifier-distinguir-texto-humano-y-generado-por-ia/> 2023
- [7] Bing.5 herramientas para detectar texto AI y texto humanos en <https://copilot.microsoft.com/?culture=en-us&country=us>;
- [8] Salah, M., Halbusi, H y Abdelfattah, F. (2023) May the force of text data analysis be with you: Unleashing the power of generative AI for social psychology research. Computers in Human Behavior: Artificial Humans 1 (2023) 100006
- [9] An, Ruopen, Yang, Y, Yang, F y Wang, S. Use prompt to differentiate text generated by ChatGPT and humans Machine Learning with Applications 14 (2023) 100497
- [10] Odri, G., Yun D. Detecting generative artificial intelligence in scientific articles: Evasion techniques and implications for scientific integrity Orthopaedics & Traumatology: Surgery & Research 109 (2023) 103706
- [11] H. Rashidi, B. Fennell, S. Alhabra, B. Hu, y T. Gorbett. The ChatGPT conundrum: Human-generated scientific manuscripts misidentified as AI creations by AI text detection tool. Journal of pathology informatics 14 (2023) 100342. www.elsevier.com/locate/jpi
- [12] Singh, D. Sharma, A. Nandy, y V. Kumar. Towards a large sized curated and annotated corpus for discriminating between human human written and AI generated texts: A case study of text sourced from Wikipedia and ChatGPT. Vol 6 2024. Natural Language processing Journal en <https://www.sciencedirect.com/journal/natural-language-processing-journal>
- [13] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding and Y. Wu. How Close is ChatGPT to Human Experts? Comparison Corpus evaluation and detection.
- [14] DAIGT-V4-TRAIN-DATASET". Kaggle: Your Machine Learning and Data Science Community. Accedido el 2 de febrero de 2024. [En línea]. Disponible: https://www.kaggle.com/datasets/thedrcat/daigt-v4-train-dataset/data?select=train_v4_drcat_01.csv
- [15] ChatGPT by MOTH (<https://www.kaggle.com/datasets/alejopaulier/daigt-external-dataset>)
- [16] Persuade corpus contributed by Nicholas Broad (<https://www.kaggle.com/datasets/nbroad/persuade-corpus-2/>)
- [17] Text generated with Llama-70b and Falcon180b by Nicholas Broad (<https://www.kaggle.com/datasets/nbroad/daigt-data-llama-70b-and-falcon180b>)
- [18] Text generated with ChatGPT and GPT4 by Radek (<https://www.kaggle.com/datasets/radek1/llm-generated-essays>)
- [19] 2000 Claude essays generated by @darraghdog (<https://www.kaggle.com/datasets/darraghdog/hello-claude-1000-essays-from-anthropic>)
- [20] LLM-generated essay using PaLM from Google Gen-AI by @kingki19 (<https://www.kaggle.com/datasets/kingki19/llm-generated-essay-using-palm-from-google-gen-ai>)