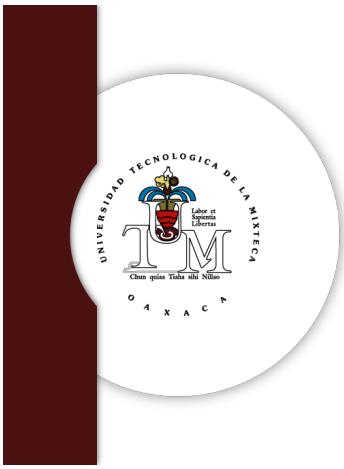

Universidad Tecnológica de la Mixteca

Detección de textos académicos generados en inglés por inteligencia artificial mediante redes convolucionales

Tesis

para obtener el título de:

Ingeniero en Computación

Presenta:

Omar Alfonso Cruz Ramírez

Director de tesis:

Dr. Christian Eduardo Millán Hernández

Huajuapan de León, Oaxaca, México Febrero de 2026

A mi maye, por el amor, la paciencia y la confianza con las que sostuviste cada paso de este camino, convertiste cada "no puedo" en un "inténtalo otra vez" y cada tropiezo en una lección. Esta carrera nació de tu fe, creció con tu apoyo y se sostiene sobre tu amor.

A mi hermanita, a quien también dedico este logro, gracias por mostrarme que, sin importar lo que uno atraviese sentimentalmente, lo importante es perseverar para alcanzar los objetivos. Tu ejemplo y tu aliento fueron una guía constante.

Índice

Índice de figuras

Índice de tablas

Resumen

Esta tesis aborda el problema de detección de textos académicos generados por inteligencia artificial en inglés, con una evaluación exploratoria en español, mediante un pipeline reproducible que combina rasgos estilométricos y representaciones semánticas. Se revisa el estado del arte, se define una metodología cuantitativa con curación del corpus, preprocesamiento mínimo y controles de *leakage*, y se construye un conjunto de 29,139 textos humanos y de inteligencia artificial (IA). Sobre este marco se comparan modelos clásicos como máquinas de soporte vectorial (SVM), Random Forest y Extreme Gradient Boosting (XGBoost), y modelos profundos como Bidirectional Encoder Representations from Transformers (BERT), redes neuronales convolucionales (CNN) y una red neuronal convolucional para texto (TextCNN), incorporando calibración probabilística, selección conservadora de umbral y pruebas de robustez.

Los resultados muestran que la CNN propuesta logra el mejor equilibrio global (Accuracy 0.9939 y F1-score 0.9924), con TextCNN muy cercano y XGBoost como la mejor línea base clásica. La comparación con detectores comerciales resalta ventajas en transparencia, trazabilidad y control de umbral cuando se dispone de un modelo propio. En conjunto, la tesis concluye que es viable superar el desempeño de los modelos del estado del arte presentados así como los de la competencia publicada en Kaggle llamada *LLM Detect AI Generated Text* (la cual llamaremos *TextTuring*) y los modelos de las herramientas comerciales (GPTZero, ZeroGPT, Justdone, etc.) para la detección de texto generado

por IA en el dominio académico, aportar un esquema de evaluación robusto y establecer lineamientos para un uso responsable en contextos educativos.

1. Introducción

Contenidos del Capítulo

La generación de texto usando modelos de inteligencia artificial se ha expandido rápidamente en entornos académicos. Modelos capaces de producir ensayos, correos y reportes con coherencia y estilo formal han reducido la barrera de producción de contenido, pero también han incrementado el riesgo de usos no declarados. En educación, esto plantea un desafío de integridad académica y de evaluación justa, ya que los textos generados pueden ser plausibles y difíciles de distinguir de los escritos por humanos.

Ante este escenario, se requiere el desarrollo de métodos de detección que sean reproducibles, transparentes y con métricas de evaluación robustas. La motivación del presente trabajo es contribuir con un enfoque metodológico que combine señales estilométricas y representaciones semánticas modernas, con especial atención a textos académicos en inglés y una evaluación exploratoria en español.

La adopción acelerada de modelos generativos de lenguaje a gran escala (LLM) en educación, como ChatGPT y servicios afines, ha modificado procesos de enseñanza, aprendizaje y evaluación, al tiempo que introduce tensiones sobre la autoría y la integridad académica (UNESCO, 2023; Cotton, 2023). En paralelo, evaluaciones independientes advierten que los detectores automáticos presentan limitaciones sustantivas (p. ej., falsos positivos y sensibilidad al dominio o idioma) que impiden su uso como única base para decisiones de alto impacto (Weber-Wulff et al., 2023).

La evidencia reciente sugiere que el uso estudiantil de herramientas de IA generativa (por ejemplo, ChatGPT y servicios afines) es significativo y variable según edad, país y contexto institucional; por ejemplo, entre adolescentes en Estados Unidos se observa un aumento en el empleo de ChatGPT para tareas escolares entre 2023 y 2024 (Pew, 2025). En consecuencia, las instituciones enfrentan el doble reto de reconocer usos legítimos (ideación, borradores, retroalimentación) y, simultáneamente, preservar la evaluación auténtica del aprendizaje (Cotton, 2023).

Desde una perspectiva técnica y lingüística, la detección se complica fuera del inglés: estudios comparativos a gran escala reportan variaciones notables de desempeño en escenarios multilingües (incluido el español) y degradación ante paráfrasis, traducción u ofuscación (Macko et al., 2023). Además, resultados empíricos muestran una amplia dispersión de precisión al comparar detectores sobre conjuntos multi-dominio (p. ej., rangos cercanos a 55–97 % de exactitud según la herramienta), lo que refuerza la necesidad de validaciones transparentes

y específicas al contexto educativo (Akram et al., 2023).

Organismos internacionales recomiendan alfabetización en Inteligencia Artificial (IA), formación docente y políticas claras sobre usos permitidos, con procedimientos de revisión humana para casos sensibles (UNESCO, 2023). Además, evidencia reciente sugiere posibles costos cognitivos asociados al uso intensivo de asistentes de IA en escritura académica (Kosmyna et al., 2025). La escritura académica sigue siendo un medio privilegiado para evidenciar pensamiento crítico, síntesis y argumentación; la disponibilidad de LLM's reconfigura estos procesos y obliga a distinguir entre apoyo legítimo y sustitución del esfuerzo intelectual (UNESCO, 2023). Asimismo, estudios sobre integridad subrayan que políticas ambiguas y ausencia de criterios operativos favorecen usos inapropiados y brechas de equidad evaluativa (Cotton, 2023).

Aunque el idioma principal de este trabajo es el inglés y la evaluación en español es exploratoria, la identificación confiable de autoría artificial enfrenta tres limitaciones clave al considerar el ámbito hispanohablante: (i) **sesgo lingüístico** y caída de desempeño al cambiar del inglés a otro idioma o cambiar de variedad dialectal; (ii) **fragilidad ante transformaciones del texto** (paráfrasis, traducción u ofuscación); y (iii) **dependencia institucional** de detectores generalistas en ausencia de lineamientos explícitos y revisión humana (Macko et al., 2023; Weber-Wulff et al., 2023; UNESCO, 2023). Este panorama establece un punto de partida, desde la perspectiva computacional, para el problema que se plantea en la siguiente sección.

En los trabajos del estado del arte de esta tesis, se muestran los avances en la detección de texto generado por IA.

1.1 Estado del arte: resultados reportados

Para contextualizar el rango de desempeño de textos generados por humanos vs IA, se resume el estado del arte con estudios que reportan métricas en detección humano vs. IA. La Tabla 1.1 sintetiza modelos y métricas reportadas para contextualizar el rango de desempeño y la diversidad de enfoques. Estos valores se presentan tal como aparecen en los estudios y no son estrictamente comparables entre sí debido a diferencias de corpus y protocolos de evaluación. Más allá de los enfoques clásicos, trabajos como *DetectGPT* (Mitchell et al., 2023;DetectGPT) y *Fast-DetectGPT* (Bao et al., 2024; Fast-DetectGPT) proponen estrategias de detección basadas en la geometría de la probabilidad del texto, que no requieren entrenamiento supervisado y ofrecen alternativas eficientes para la verificación en entornos multilingües y educativos. La detección de texto generado por LLM's ha transitado desde enfoques puramente estilométricos hacia métodos basados en representaciones distribuidas y, más recientemente, combinaciones híbridas. En términos generales, la literatura distingue tres familias con señales complementarias (Weber-Wulff et al., 2023; Macko et al., 2023):

Tabla 1.1: Modelos, métricas reportadas e investigaciones relacionadas.

Estudio (año)	**Algoritmos evaluados**	**Corpus/Tarea**	**Principales métricas reportadas**
Najjar et al. (2025, arXiv)	XGBoost, Random Forest (tradicionales) + baseline.	Conjunto CyberHumanAI (500 humano / 500 LLM; clasificación binaria).	Accuracy: XGBoost 83 %, Random Forest 81 %; en tarea de tres clases, el modelo propuesto ≈ 77.5 % vs. GPTZero ≈ 48.5 %.
Prova (2024, arXiv)	BERT, XGBoost, SVM.	Detección binaria humano vs. IA.	Accuracy: BERT 93 %, XGB 84 %, SVM 81 %; se muestran también matrices de confusión y análisis por clase.
Yadgiri et al. (2024, ACL-ICON)	RoBERTa (*fine-tuned*) vs. SVM, Random Forest, XGBoost (con rasgos estilométricos).	Ensayos en inglés (comparan detectores por generador).	Hallazgo: RoBERTa reduce falsos negativos frente a SVM.

- **Estilometría clásica**: rasgos léxicos (riqueza, *type–token ratio*, palabras

función), sintácticos (longitud de oraciones, distribución de categorías gramaticales (POS), puntuación) y discursivos (coherencia local/global, conectores). Estos marcadores capturan regularidades micro y mesoestructurales asociadas a la generación automática en ciertos contextos.

- **Modelos supervisados con representaciones semánticas**: clasificadores entrenados sobre *embeddings* documentales derivados de arquitecturas *Transformer*, que modelan dependencias de largo alcance mediante autoatención (Vaswani et al., 2017). Suelen integrarse con calibración probabilística y selección de umbral orientada a costes de error.
- **Métodos híbridos**: combinan señales estilométricas y representaciones distribuidas y, en ocasiones, estimadores auxiliares (p. ej., medidas de fluidez) para mejorar sensibilidad y control de falsos positivos en dominios específicos (Weber-Wulff et al.,2023).

En síntesis, el estado del arte muestra avances, pero también una diversidad de enfoques y niveles de transparencia que dificultan comparar herramientas de forma homogénea. En línea con el primer párrafo del planteamiento del problema, la heterogeneidad de las herramientas comerciales, sus umbrales de decisión y su reproducibilidad desigual limitan su adopción institucional, lo que justifica definir criterios de evaluación claros y replicables.

1.2 Planteamiento del problema

Las herramientas comerciales disponibles para detectar texto generado por IA son heterogéneas en cuanto a su transparencia, umbrales de decisión y reproducibilidad. Su desempeño puede variar según dominio, longitud del texto o idioma, y no siempre se reporta con métricas comparables. Esto dificulta su adopción institucional y limita su valor como apoyo a la evaluación académica.

El problema central consiste en determinar si es posible construir un modelo de detección que supere la precisión de las herramientas actuales (GPTZero,

ZeroGPT, Justdone y WinstonAI) y de los resultados de la competencia de Kaggle[1] (*TextTuring*), y al mismo tiempo ofrezca criterios de evaluación claros y replicables, en el contexto de textos académicos.

La pregunta que guía este trabajo es: *¿Es posible desarrollar un modelo de detección de texto generado por inteligencia artificial que supere la precisión de las herramientas actuales, con enfoque en textos académicos en inglés y evaluación exploratoria en español?*

1.3 Objetivos

En esta sección se presentan los objetivos que orientan el estudio y delimitan su alcance operativo, distinguiendo entre el propósito general y los objetivos específicos.

1.3.1 Objetivo general

Desarrollar un modelo para la detección de texto generado por inteligencia artificial que supere la precisión de las herramientas actuales, con enfoque primario en textos académicos en inglés y evaluación exploratoria en español, alineando su evaluación con métricas y procedimientos robustos.

1.3.2 Objetivos específicos

1. Realizar una revisión documental y del estado del arte sobre técnicas y herramientas para la detección de texto generado por IA en contextos académicos.

[1] `https://www.kaggle.com/competitions/llm-detect-ai-generated-text`

2. Diseñar un método híbrido que combine rasgos estilométricos y representaciones basadas en Transformers para identificar texto generado por IA.

3. Implementar el método propuesto y construir un pipeline reproducible de extremo a extremo (datos → rasgos → modelos → métricas).

4. Evaluar el desempeño con foco principal en inglés e incluir una evaluación exploratoria en español para estimar transferencia.

1.4 Hipótesis

Con base en el problema planteado, los objetivos y el diseño metodológico, se formulan las siguientes hipótesis para guiar la validación empírica del detector propuesto:

Hipótesis general. Un detector híbrido que combina (i) rasgos estilométricos multicapas (léxicos, sintácticos y discursivos) y (ii) representaciones semánticas derivadas de modelos *Transformer*, con calibración probabilística y selección conservadora de umbral, **superará el desempeño de detectores generalistas** al discriminar textos académicos en inglés generados por IA, manteniendo una tasa de falsos positivos acotada para su uso en contextos educativos (Vaswani et al., 2017; WeberWulff et al., 2023; UNESCO, 2023).

1.5 Justificación

La presente investigación se justifica por sus beneficios directos para la integridad académica, la mejora de la evaluación educativa y el avance de la disciplina del Procesamiento del Lenguaje Natural (PLN) en contextos hispanohablantes. El enfoque principal recae en textos académicos en inglés; la

evaluación en español se aborda de manera exploratoria para estimar transferencia y robustez multilingüe, dadas las caídas reportadas al pasar del inglés a otras lenguas (Macko et al., 2023). En un entorno donde los LLM se integran rápidamente a prácticas de estudio y producción científica, se requieren herramientas y pautas que permitan distinguir con mayor certeza entre autoría humana y generación automática, sin sustituir la revisión humana ni incurrir en decisiones injustas (UNESCO, 2023).

Desde la perspectiva académica y educativa, el proyecto aporta: (i) evidencias y métricas confiables para docentes y comités académicos al tomar decisiones informadas, (ii) reducción de falsos positivos mediante umbrales conservadores y calibración probabilística, lo que protege a estudiantes ante acusaciones infundadas, y (iii) materiales y protocolos que fortalecen la alfabetización en IA y promueven usos responsables en el aula (UNESCO, 2023; Weber-Wulff et al., 2023). Para las instituciones, el pipeline reproducible y las guías de reporte aumentan la transparencia y la trazabilidad, facilitando lineamientos institucionales y auditorías técnicas.

Para la disciplina del PLN, la contribución es doble: (a) técnica, al evaluar un enfoque híbrido que combina el análisis de estilo de escritura con representaciones Transformer bajo protocolos robustos (*Area bajo la curva de precisión-recuperación (AUPRC)* como métrica primaria, validación por grupos, pruebas de robustez por paráfrasis y traducción), con énfasis en el dominio en inglés y análisis exploratorio en español; y (b) de ciencia abierta, al priorizar documentación, calibración y estimación de incertidumbre que favorecen la comparabilidad entre estudios (Macko et al., 2023; Weber-Wulff et al., 2023). Este énfasis en prácticas sólidas puede transferirse a tareas afines de verificación y control de calidad textual.

En términos de beneficiarios, el resultado de la tesis busca sentar bases para construir herramientas que favorezcan a: estudiantes (evaluaciones más justas y formativas), profesorado y cuerpos colegiados (mejores insumos para

dictámenes), áreas de integridad académica y autoridades educativas (marcos de decisión con evidencia y límites de uso), y a la comunidad de investigación en PLN (mejores bases metodológicas y resultados reproducibles). Asimismo, al enfocarse en inglés con extensión exploratoria al español, el trabajo aspira a contribuir a reducir brechas de desempeño multilingüe y a mejorar la equidad en herramientas de detección (Macko et al., 2023).

El crecimiento acelerado del uso de modelos generativos en educación y ciencia (Maslej et al., 2023) refuerza la oportunidad y pertinencia de una solución contextualizada que eleve la calidad de la evaluación, proteja a la comunidad académica y avance el estado del arte del PLN aplicado a integridad académica.

1.6 Alcances y limitaciones

Para clarificar el alcance del estudio, se distinguen explícitamente los alcances y las limitaciones:

1.6.1 Alcances

- Clasificación binaria (humano vs. IA) en textos académicos en inglés, con una exploración adicional en español para observar la transferencia del modelo.
- Comparabilidad y reproducibilidad del pipeline, con procedimientos, particiones y métricas consistentes.
- Evaluación centrada en umbrales conservadores y control de falsos positivos para contextos educativos.
- Análisis de robustez ante paráfrasis, traducción y cambios de dominio en las pruebas reportadas.
- Reporte de criterios de calibración y trazabilidad de artefactos para facilitar auditoría.

1.6.2 Limitaciones

- No se pretende cubrir todos los dominios de escritura ni todas las variantes de modelos generativos.
- La exploración en español se basa en traducción automática, lo que puede introducir sesgos de estilo.
- Los resultados no se interpretan como decisión automática, sino como apoyo a la evaluación humana.
- La validez del modelo puede variar ante nuevas versiones de LLM o cambios de dominio no evaluados.

1.7 Estructura de la tesis

La tesis se organiza en cinco capítulos. El Capítulo 1 presenta la introducción, el planteamiento del problema, los objetivos, la justificación, el alcance y el estado del arte. El Capítulo 2 desarrolla el marco teórico sobre modelos generativos, enfoques de detección y fundamentos de evaluación. El Capítulo 3 describe la metodología propuesta, el diseño experimental, el pipeline y los criterios de validación. El Capítulo 4 reporta los resultados y comparativas, junto con el análisis de métricas y visualizaciones. Finalmente, el Capítulo 5 expone las conclusiones, aportaciones y líneas de investigación futura.

2. Marco Teórico

Contenidos del Capítulo

Este capítulo establece el marco teórico para la detección de autoría en textos académicos. Primero se definen los conceptos de autoría y se diferencia entre clasificar y detectar; luego se revisan los modelos de aprendizaje automático y profundo, la extracción de características (*Term Frequency–Inverse Document Frequency*, por sus siglas en inglés TF–IDF, n-gramas y *embeddings*), la evaluación de rendimiento y los trabajos relacionados, en línea con el índice del capítulo.

El alcance se centra en clasificación a nivel de documento, que permite evaluaciones controladas y comparables; la detección localizada se considera una extensión futura que requiere anotaciones más finas. Con ello, el marco conecta la problemática educativa con la implementación computacional y explicita los supuestos del análisis.

2.1 Clasificación de autoría

Antes del auge del Procesamiento de Lenguaje Natural (PLN) moderno, la atribución de autoría se abordaba desde la estilometría y la lingüística computacional, buscando patrones de estilo que distinguieran autores o tipos de texto. Este antecedente conceptualiza la autoría como un conjunto de señales medibles en el lenguaje: léxico, sintaxis, puntuación y coherencia discursiva. En el contexto de la Inteligencia Artificial (IA) generativa, se entiende por autoría artificial la producción total o parcial del texto por modelos automáticos.

El PLN define la tarea de *clasificación de autoría humano vs. IA* como clasificación supervisada, donde cada documento recibe una etiqueta global (humano o IA). Este encuadre exige un corpus representativo del dominio y un esquema de anotación consistente, pues los modelos aprenden las regularidades presentes en esos datos y pueden fallar cuando el dominio cambia o el estilo del texto se transforma.

En este trabajo se distingue entre **clasificar** y **detectar**:

- **Clasificar:** decidir si un texto completo fue escrito por una persona o generado por IA (problema binario humano vs. IA).
- **Detectar:** localizar en qué partes del texto se utilizó IA, lo que implica un análisis más fino por segmentos o unidades discursivas.

Como antecedente relevante, se considera la competencia *LLM Detect AI Generated Text* de Kaggle, orientada a la detección de autoría humano vs. IA.[1] La clasificación es el foco principal del estudio, mientras que la detección segmentada se reconoce como una extensión más compleja que requiere anotaciones y modelos a nivel local.

La diferencia entre ambos enfoques implica requisitos distintos de datos y de evaluación. En clasificación se emplean etiquetas por documento y métricas

[1] `https://www.kaggle.com/competitions/llm-detect-ai-generated-text`

agregadas; en detección se requieren anotaciones por fragmento y modelos capaces de producir predicciones locales, lo que incrementa el costo de anotación y la complejidad de validación.

En el ámbito educativo, la clasificación de autoría se vincula con la integridad académica y la necesidad de lineamientos claros. Organismos internacionales recomiendan alfabetización en IA, políticas explícitas y revisión humana en decisiones sensibles (UNESCO, 2023; Cotton, 2023). A ello se suman limitaciones tecnológicas que condicionan la confiabilidad de los detectores y su aplicación en contextos multilingües (Weber-Wulff et al., 2023; Macko et al., 2023).

Además, la disponibilidad de LLM's con alta fluidez dificulta separar el estilo humano del generado por IA, especialmente cuando el texto es revisado o mezclado. Por ello, la conceptualización de autoría debe contemplar escenarios híbridos, donde una parte del contenido es humana y otra asistida, lo que refuerza la necesidad de enfoques complementarios y revisión humana.

En particular, aunque el foco principal es el inglés, se identifican tres restricciones recurrentes:

- **Sesgo lingüístico:** caída de desempeño al salir del inglés o cambiar de variedad dialectal.
- **Fragilidad ante transformaciones:** degradación bajo paráfrasis, traducción u ofuscación.
- **Dependencia institucional:** uso de detectores generalistas sin criterios operativos ni revisión humana.

Estas restricciones motivan el uso de modelos y evaluaciones robustas, así como la comparación entre familias algorítmicas. En particular, se busca equilibrar rendimiento con explicabilidad para evitar decisiones opacas en contextos educativos. Desde el punto de vista computacional, estas tareas se abordan con dos familias de métodos: algoritmos clásicos basados en rasgos explícitos y enfoques

de *deep learning* que aprenden representaciones distribuidas. La siguiente sección formaliza estos modelos.

Un punto clave es la definición del criterio de decisión. La clasificación puede operar con umbrales fijos o adaptativos según el contexto, y los puntajes deben interpretarse como probabilidades aproximadas, no como certezas absolutas. Esto es relevante cuando el texto combina edición humana y salida de IA, pues los puntajes pueden reflejar grados intermedios de intervención.

La conceptualización también considera la estabilidad temporal: los LLM evolucionan rápidamente y los patrones que hoy distinguen texto generado pueden cambiar con nuevas versiones. Por ello, se plantea la necesidad de evaluar y analizar periódicamente los modelos para actualizar criterios y construir detectores cada vez más robustos, documentando el contexto de entrenamiento para sostener la validez de los resultados.

Finalmente, la clasificación de autoría en educación debe equilibrar exactitud con equidad. Un modelo muy sensible puede elevar falsos positivos y afectar a grupos específicos; un modelo demasiado conservador puede dejar pasar textos generados. De ahí la necesidad de protocolos claros y revisión humana.

2.2 Modelos de aprendizaje automático

El aprendizaje automático (ML) se define como el conjunto de técnicas que aprenden una función de predicción a partir de datos y características observables. En clasificación de texto, estos modelos suelen operar sobre representaciones vectoriales (p. ej., frecuencia de término–frecuencia inversa de documento *"TF–IDF"* o n-gramas) y permiten establecer comparaciones basadas en métricas objetivas.

En términos prácticos, el flujo de ML incluye vectorización, partición de datos, ajuste de hiperparámetros y validación cruzada. El objetivo es minimizar

el error de generalización y controlar el sobreajuste mediante regularización, selección de variables o combinación de modelos.

Aprendizaje automático clásico

En esta sección se describen tres algoritmos clásicos empleados como líneas base: Random Forest, SVM y XGBoost. Se definen, se formalizan y se acompañan de esquemas conceptuales que sintetizan su flujo de aprendizaje.

Random Forest (RF). RF es un ensamble de T árboles de decisión entrenados con muestreo *bootstrap* y selección aleatoria de atributos, lo que reduce la correlación entre árboles y la varianza del modelo (Breiman, 2001). Para clasificación, la predicción se obtiene por voto mayoritario:

$$\hat{y} = \text{mode}_{t=1}^{T} h_t(x), \quad \hat{p}(y = 1 \mid x) = \frac{1}{T} \sum_{t=1}^{T} p_t(y = 1 \mid x),$$

donde h_t es el clasificador del árbol t y p_t su probabilidad estimada. En clasificación de textos, RF captura interacciones no lineales entre rasgos (p. ej., n-gramas y señales estilométricas) y ofrece medidas de importancia de variables.

RF: muestreo *bootstrap* → entrenar árboles → combinar por voto/promedio → predicción estable.

Figura 2.1: Esquema conceptual de Random Forest.

Máquina de soporte vectorial (SVM). SVM busca un hiperplano de máximo margen que separe las clases en el espacio de atributos (Cortes & Vapnik,

1995). La formulación lineal con margen suave se expresa como:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \quad \text{s.a. } y_i(w^\top x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0,$$

y la regla de decisión es $f(x) = \text{sign}(w^\top x + b)$. El kernel lineal es eficiente en espacios dispersos y los pesos w permiten interpretar qué rasgos favorecen cada clase.

SVM: representar textos → encontrar hiperplano de máximo margen → clasificar por el lado del margen.

Figura 2.2: Esquema conceptual de una SVM lineal.

Extreme Gradient Boosting (XGBoost). XGBoost implementa *gradient boosting* de árboles con regularización explícita, construyendo un modelo aditivo de la forma (Chen & Guestrin, 2016):

$$\hat{y}_i = F(x_i) = \sum_{m=1}^{M} f_m(x_i), \quad f_m \in \mathcal{F},$$

con objetivo

$$\mathcal{L} = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{m=1}^{M} \Omega(f_m), \quad \Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2,$$

donde Ω penaliza la complejidad de cada árbol. El aprendizaje es secuencial y corrige errores previos mediante gradientes; hiperparámetros como profundidad, tasa de aprendizaje y número de árboles controlan el sesgo–varianza.

XGBoost: entrenar árbol → ajustar residuales → sumar modelos → regularizar complejidad.

Figura 2.3: Esquema conceptual de XGBoost.

En conjunto, estos tres algoritmos ofrecen un balance entre interpretabilidad y potencia predictiva, y sirven como referencia para contrastar el aporte de arquitecturas profundas.

Aprendizaje profundo

El aprendizaje profundo (DL) utiliza redes neuronales con múltiples capas para aprender representaciones jerárquicas del texto y capturar patrones no lineales (Goodfellow et al., 2016). En clasificación de texto, las redes convolucionales han mostrado un buen equilibrio entre costo computacional y capacidad para modelar patrones locales de estilo y contenido (Kim, 2014; Zhang et al., 2015).

Capas convolucionales. Una capa convolucional aplica filtros deslizantes sobre una secuencia de vectores (p. ej., *embeddings*), generando mapas de activación que detectan patrones locales. Para una convolución 1D con kernel de tamaño k, el cálculo básico es:

$$s_t = \sigma\left(\sum_{i=0}^{k-1} w_i \cdot x_{t+i} + b\right),$$

donde x_{t+i} es el vector de entrada en la posición $t+i$, w_i son los pesos del filtro y σ es una función de activación (Goodfellow et al., 2016).

Convolución 1D: ventana de tamaño k se desliza sobre tokens y produce un mapa de activación.
Convolución 2D: kernel $k \times k$ se desliza sobre una matriz (p. ej., imagen) y produce mapas en dos dimensiones.

Figura 2.4: Diferencia conceptual entre convolución 1D y 2D.

CNN (Conv1D). La convolución 1D recorre una sola dimensión (la secuencia de tokens) y es adecuada cuando el orden lineal es la estructura principal del dato. En texto, esto permite capturar patrones tipo n-grama con menos parámetros y menor costo que una convolución 2D, que está pensada para datos en rejilla (imágenes) (Kim, 2014; Zhang et al., 2015). Además, la 1D se adapta bien a longitudes variables y preserva la interpretabilidad local de los filtros.

Una CNN de texto típica usa *embeddings* como entrada, aplica Conv1D y luego un *pooling* global que conserva las activaciones más informativas. El **padding** (*same* o *valid*) controla si se preserva o reduce la longitud, y el *GlobalMaxPooling* introduce invariancia local al seleccionar el máximo de cada filtro (Goodfellow et al., 2016; Kim, 2014).

Tokens → *embeddings* → Conv1D → *GlobalMaxPooling* → capa densa → salida sigmoidal.

Figura 2.5: Esquema de CNN para clasificación de texto.

Una variante regularizada usa 128 filtros, regularización L2 (1×10^{-4}) y *dropout* 0.3, junto con *early stopping* (paciencia 2) y *checkpointing*. Esta configuración mejora la estabilidad y reduce el sobreajuste (Srivastava et al., 2014; Prechelt, 1997).

TextCNN. TextCNN agrega varias convoluciones en paralelo con distintos tamaños de kernel y concatena sus salidas antes del *pooling*. Este diseño captura patrones locales de distinta longitud y suele mejorar la sensibilidad a estructuras diversas en el texto (Kim, 2014).

embeddings → Conv1D (k=3) Conv1D (k=4) Conv1D (k=5) → concat → *GlobalMaxPooling* → densa → salida.

Figura 2.6: Esquema conceptual de TextCNN con múltiples kernels.

La etapa de entrenamiento requiere controlar aspectos como inicialización, balance de clases y longitud máxima de secuencia. En CNN, el tamaño del vocabulario y el truncamiento influyen en la cobertura de señales estilométricas, por lo que se busca un equilibrio entre representatividad y costo computacional.

En términos de interpretabilidad, los modelos clásicos permiten analizar pesos o importancias de variables, mientras que las CNN requieren técnicas indirectas (p. ej., análisis de activaciones o perturbaciones). Esto no invalida su uso, pero sí obliga a complementar el rendimiento con análisis cualitativos y métricas consistentes.

La selección de arquitectura no es universal: en dominios con textos muy largos, las CNN pueden perder dependencias de largo alcance; en dominios con señales locales fuertes, suelen rendir bien. Por ello, el marco teórico justifica el uso de CNN como un punto intermedio entre modelos lineales y Transformers.

2.3 Extracción de características

La extracción de características transforma el texto en representaciones numéricas que puedan procesarse por los modelos. En este trabajo se consideran dos familias principales: representaciones dispersas basadas en frecuencia y representaciones densas basadas en *embeddings*.

Entre las representaciones dispersas destacan TF–IDF a nivel de palabra y de caracter, n-gramas y conteos normalizados. Estas vistas capturan frecuencias y regularidades de estilo y son especialmente adecuadas para SVM, RF y XGBoost por su interpretabilidad y eficiencia en espacios de alta dimensión.

Embeddings y GloVe

Un *embedding* es una representación numérica que asigna cada token a un vector denso en $\mathbb{R}^d$, de modo que la cercanía geométrica refleje similitud semántica y sintáctica. A diferencia de las representaciones dispersas, los *embeddings* codifican relaciones continuas entre palabras y permiten capturar regularidades de lenguaje en un espacio de baja dimensión (Goodfellow et al., 2016).

En Word2Vec, los vectores se aprenden al predecir contexto a partir de una palabra (skip-gram) o la palabra a partir del contexto (CBOW), lo que induce que palabras con contextos similares queden cercanas en el espacio vectorial (Mikolov et al., 2013). GloVe, por su parte, factoriza una matriz de coocurrencias globales con una pérdida ponderada, combinando información global y local para obtener representaciones estables y reutilizables (Pennington et al., 2014). Estas representaciones capturan analogías y regularidades léxicas, y son útiles como inicialización de modelos profundos.

Los embeddings pueden ser **estáticos** (GloVe, Word2Vec), con un vector fijo por palabra, o **contextuales** (BERT, DistilBERT), donde la representación depende del contexto y permite distinguir sentidos de una misma palabra. En tareas de clasificación, se puede usar un vector agregado del documento (promedio, máximo o [CLS]) o alimentar la secuencia de embeddings a una CNN para capturar patrones locales.

En comparación con TF–IDF y n-gramas, los embeddings ofrecen varias ventajas: reducen la esparsidad, mantienen una dimensionalidad controlada, capturan sinonimia y similitud semántica, y generalizan mejor cuando el vocabulario o el dominio cambia. Su limitación principal es la menor interpretabilidad y la dependencia de preentrenamiento, lo que puede introducir sesgos si el corpus de origen no es representativo.

Flujo de trabajo

El flujo general de extracción de características se resume en:

- Normalizar y limpiar el texto con procesamiento mínimo.
- Tokenizar y construir vocabulario.
- Generar representaciones (TF–IDF, n-gramas o *embeddings* preentrenados).
- Alimentar los modelos de ML/DL con las matrices resultantes.

Texto crudo → limpieza mínima → tokenización/vocabulario → vectorización (TF–IDF/n-gramas/*embeddings*) → modelo ML/DL → predicción.

Figura 2.7: Diagrama general del flujo de representación de texto y modelado (adaptado de Goodfellow et al., 2016).

En estos pasos se definen parámetros como tamaño de vocabulario (p. ej., 10,000), longitud máxima (p. ej., 128) y estrategia de *padding*/truncamiento, que afectan directamente el costo computacional y el rendimiento. El manejo de tokens desconocidos, la normalización de mayúsculas y la preservación de puntuación también influyen en la calidad de la representación.

Además de TF–IDF y embeddings, pueden considerarse rasgos estilométricos como diversidad léxica, promedio de longitud de oraciones, frecuencia de conectores y densidad de puntuación. Estos rasgos enriquecen la interpretabilidad y permiten contrastar patrones de escritura humana frente a generada.

Para textos académicos, la presencia de citas, estructura de secciones y estilo formal puede introducir sesgos en la representación. Por ello, el flujo de extracción debe preservar señales relevantes sin eliminar información que pueda ser discriminante (p. ej., comillas, paréntesis o numerales).

El uso de embeddings contextuales también habilita estrategias híbridas: combinar un vector semántico global con rasgos locales. Esta combinación me-

jora la sensibilidad a diferencias sutiles de estilo y contenido, particularmente en textos largos.

2.4 Evaluación de rendimiento

La evaluación sigue un flujo controlado: (i) obtener datos crudos, (ii) realizar procesamiento mínimo de limpieza, y (iii) entrenar y validar los modelos con particiones estables. El uso de la validación cruzada (5-*folds*) divide el conjunto en k particiones, entrena en $k-1$ y evalúa en la restante de forma rotativa, lo que permite estimar el rendimiento promedio y su variabilidad, reduciendo el riesgo de conclusiones optimistas por particiones fortuitas (Kohavi, 1995).

Para evitar sesgos, se privilegia la estratificación de clases, la comparación bajo el mismo protocolo y el reporte de promedios y desviaciones. Además, la selección de umbral puede ajustarse según el costo relativo de falsos positivos y falsos negativos, criterio relevante en entornos educativos.

Se emplean métricas clásicas de clasificación binaria y se complementan con curvas *Receiver Operating Characteristic* (ROC) y *Precision–Recall* (PR) cuando corresponde:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{ROC-AUC} = \int_0^1 TPR(FPR)\, dFPR$$

Cuando existe desbalance, las curvas Precision–Recall ofrecen una lectura más informativa del trade-off entre aciertos y falsos positivos (Davis & Goadrich,

2006; Saito & Rehmsmeier, 2015).

Adicionalmente, se reportan curvas de ROC y PR promediadas por partición, y se analiza la matriz de confusión para identificar patrones de error. Este enfoque permite evaluar no sólo el rendimiento global, sino también el equilibrio entre precisión y sensibilidad bajo distintos umbrales.

La evaluación también incluye análisis de estabilidad: se observa la varianza entre particiones y se reportan promedios con desviación estándar. Esto permite identificar si un modelo es consistente o si depende excesivamente de la partición de datos.

En tareas sensibles, se recomienda acompañar las métricas con análisis cualitativo. Revisar ejemplos de falsos positivos y falsos negativos ayuda a detectar patrones sistemáticos de error, como textos muy editados o con lenguaje técnico altamente repetitivo.

Finalmente, el rendimiento debe interpretarse en función del uso esperado. Un modelo con alta precisión pero baja cobertura puede ser útil para revisión inicial, mientras que un modelo con mayor recall puede servir como filtro amplio, siempre con supervisión humana.

2.5 Trabajos relacionados

Los estudios publicados entre 2020 y 2024 reportan comparaciones entre modelos clásicos y Transformers, con resultados que dependen del corpus y del protocolo. La Tabla 1.1 resume los trabajos más cercanos al problema planteado. A continuación se describen con mayor detalle sus procedimientos y hallazgos principales.

Najjar et al. (2025, arXiv) se enfocan en detección humano vs. LLM con el conjunto CyberHumanAI (500 humanos / 500 LLM). Su diseño compara líneas base tradicionales (XGBoost, Random Forest) bajo el mismo esquema de

clasificación binaria y reporta *accuracy* de 83 % y 81 %. Además, en una tarea de tres clases reportan que el modelo propuesto alcanza ≈ 77.5 %, mientras que GPTZero queda en ≈ 48.5 %. Este contraste evidencia que el rendimiento varía sustancialmente al cambiar la tarea y el protocolo.

Prova (2024, arXiv) plantea una evaluación binaria humano vs. IA comparando BERT, XGBoost y SVM. La metodología se centra en evaluar modelos de distinta familia con una misma etiqueta global de documento y reporta *accuracy* de 93 %, 84 % y 81 % respectivamente. Además de las métricas globales, presenta matrices de confusión y análisis por clase, lo que permite observar patrones de error y no sólo promedios agregados.

Yadgiri et al. (2024, ACL-ICON) comparan detectores basados en RoBERTa ajustado frente a SVM, Random Forest y XGBoost con rasgos estilométricos, usando ensayos en inglés y comparaciones por generador. El resultado clave es la reducción de falsos negativos con RoBERTa respecto a SVM, sugiriendo que las representaciones contextuales son particularmente útiles cuando el texto es fluido o está bien editado.

En conjunto, estos trabajos muestran que los mejores resultados dependen del tipo de texto, del diseño experimental y del objetivo (binario vs. multiclase), por lo que las comparaciones deben interpretarse con cautela.

Un antecedente relevante es la competencia *LLM – Detect AI Generated Text* de Kaggle, orientada a la detección humano vs. IA en ensayos educativos. El desafío empleó ROC-AUC como métrica y reunió 4,358 equipos; los mejores resultados del *leaderboard* privado alcanzaron aproximadamente 0.96 de AUC, con diferencias de milésimas entre los primeros lugares. El diseño incluyó un conjunto de entrenamiento con fuerte desbalance y un *test* oculto por *prompt*, lo que incentivó la generación de datos sintéticos y el uso de *ensembles* de Transformers. Los detalles de organización, protocolo y técnicas ganadoras se sintetizan en el Apéndice A.[2]

[2] `https://www.kaggle.com/competitions/llm-detect-ai-generated-text`

En herramientas comerciales, la evidencia es heterogénea y a menudo proviene de fuentes no académicas, lo que limita la comparabilidad. La Tabla 2.1 sintetiza métricas reportadas para GPTZero, ZeroGPT y Winston AI. En el caso de GPTZero, los resultados disponibles reportan tasas de falsos positivos y falsos negativos bajo un estudio clínico, lo que sugiere utilidad en detección humano → IA, pero menor capacidad en la dirección inversa. Las comparativas en blogs muestran diferencias sustantivas entre GPTZero y ZeroGPT, aunque sin protocolos controlados. Para Winston AI, los valores publicados son de tipo comercial y no incluyen validaciones externas. En conjunto, estas fuentes aportan una referencia práctica, pero no reemplazan evaluaciones reproducibles.

Tabla 2.1: Herramientas comerciales: métricas reportadas y contexto.

Herramienta	**Métrica reportada**	**Fuente/contexto**	**Comentarios relevantes**
GPTZero	Falsos positivos ≈ 10 %, falsos negativos ≈ 35 %.	Estudio clínico (PMC).	Buena para detectar humano → IA, peor para IA → humano.
GPTZero	Comparativo contra ZeroGPT; en 6 pruebas detectó AI-text 85 %, 99 %, 87 % vs. ZeroGPT 75 %, 65 % y 45 %.	Blog Twixify.	No son académicos, pero son comparativos.
ZeroGPT	Detección de "humano" con errores altos: textos *human-written* fueron marcados ≈ 49 %, 62 %, 86 % como IA.	Mismo blog.	Alto riesgo de falsos positivos con ZeroGPT.
Winston AI	Promoción de "99.98 % accuracy".	Marketing Winston AI.	Valor autoreclamado, sin estudio externo verificado.

En documentos académicos de tesis, se reportan enfoques alternativos y notas metodológicas que aportan contexto sobre comparaciones y sesgos. La Tabla 2.2 resume dos trabajos recientes que combinan clasificación multiclase, comparaciones con detectores comerciales y observaciones sobre caída fuera de dominio (OOD).

Tabla 2.2: Documentos académicos y tesis: resumen de resultados.

Documento	**Modelo/Enfoque**	**Corpus/Tarea**	**Resultados y comparación**
Golchoubian (2024), MSc, UNBC.	Multinomial Naive Bayes con pares de palabras.	Clasificación 4 clases: Humano, ChatGPT, Gemini, OtherAI.	*Accuracy* 92.91 %; reporta precision/recall/F1 por experimento; comparación con NB "original" y herramientas comerciales (GPTZero/QuillBot).
Al Ali (2025), MSc, Charles University.	TF–IDF + Naive Bayes; RobCzech (BERT-like); logits de Llama.	Varios dominios en checo (nativos y no nativos).	Sin cifras globales; alto rendimiento en dominio de entrenamiento y caída OOD; comparación con detector comercial sin números; no evidencia de sesgo sistemático contra no nativos.

En síntesis, los trabajos relacionados muestran que los resultados dependen tanto del modelo como del diseño experimental, y que las fuentes no revisadas por pares deben interpretarse con cautela al comparar herramientas.

2.6 Resumen del capítulo

Este capítulo estableció el marco teórico para la clasificación y detección de autoría, delimitando conceptos y el nivel de análisis. Se explicaron las diferencias entre clasificar y detectar, así como los retos de dominio, estabilidad temporal y uso responsable en contextos educativos. Además, se describieron los modelos clásicos (RF, SVM, XGBoost) y los enfoques de aprendizaje profundo (CNN/TextCNN), junto con consideraciones de arquitectura, regularización y entrenamiento. Se detallaron estrategias de extracción de características, el uso de *embeddings* y el flujo de trabajo para construir representaciones. Finalmente, se presentaron criterios de evaluación con validación cruzada y métricas

clave, y se contextualizó el trabajo con estudios relacionados y sus implicaciones metodológicas.

3. Metodología

Contenidos del Capítulo

3.1 Método propuesto

En este capítulo se presenta el método propuesto basado en un modelo hibrído mediante una CNN, así como los pasos desde la obtención de los datos y la evaculación. Como punto de partida, se recuerda la pregunta de investigación: "¿Es posible desarrollar un modelo de detección de texto generado por inteligencia artificial que supere la precisión de las herramientas existentes,

adaptado específicamente al contexto académico de la educación media superior y superior en México?". A continuación se presenta la idea general del método propuesto para la detección de textos generados por inteligencia artificial (IA) en contextos académicos, formulado como un problema de clasificación binaria (humano vs. modelos de lenguaje a gran escala (LLM)) y desplegado como un pipeline reproducible de extremo a extremo. En términos metodológicos, el diseño evita tratar la detección como un "modelo aislado" y, en cambio, la concibe como una cadena de transformaciones verificables (datos → representaciones → modelos → decisiones), con trazabilidad de artefactos, control de fuentes de variación y criterios de decisión explícitos.

El pipeline prioriza dos objetivos de ingeniería de modelos que suelen estar en tensión: (i) rendimiento discriminativo (separar clases) y (ii) interpretabilidad operacional (producir probabilidades calibradas y umbrales defendibles). Este segundo objetivo es particularmente relevante en educación, donde el costo social de un falso positivo es alto; de hecho, evaluaciones sistemáticas de herramientas comerciales y públicas concluyen que, en condiciones realistas, los detectores pueden exhibir sesgos y desempeño insuficiente, por lo que su uso requiere prudencia y supervisión humana (Weber-Wulff et al., 2023). Por consiguiente, la metodología incorpora calibración probabilística, selección conservadora de umbral y pruebas de robustez, como mecanismos de control y de rendición de cuentas.

3.1.1 Diseño de investigación

Se adopta un diseño cuantitativo, no experimental, explicativo y de corte transversal. El objetivo es estimar la probabilidad de que un texto haya sido generado por un LLM frente a autoría humana, usando evidencia del propio texto (rasgos estilométricos y representaciones semánticas), sin intervenir ni manipular variables en un entorno controlado. En términos de validez interna, se reduce el riesgo de sesgos mediante particiones controladas, controles de

leakage y protocolos de evaluación reproducibles.

Se priorizan protocolos conservadores para minimizar falsos positivos en contextos educativos, debido a que el uso de detectores como mecanismo punitivo ha sido cuestionado por su confiabilidad limitada y por la posibilidad de errores sistemáticos. En particular, estudios comparativos concluyen que los detectores disponibles pueden no ser "*accurate nor reliable*" bajo condiciones realistas, por lo que se recomienda un enfoque de apoyo a la revisión humana y la transparencia sobre incertidumbre (Weber-Wulff et al., 2023). Por consiguiente, el método se orienta a reportar probabilidades calibradas, umbrales con restricciones explícitas de tasa de falsos positivos (FPR) y análisis de errores como evidencia para decisiones institucionales informadas (Guo et al., 2017; Tong et al., 2018).

3.1.2 Diseño de alto nivel del método propuesto

Como preámbulo, se muestra el flujo global del método propuesto: adquisición y preparación de datos, representación del texto, entrenamiento de modelos y reporte reproducible. Esta vista resume los componentes principales y sus transiciones, resaltando que la confiabilidad del resultado depende tanto del modelo como de la higiene experimental (particiones, controles de fuga de información, y consistencia de preprocesamiento) (Arlot & Celisse, 2010; Kaufman et al., 2012).

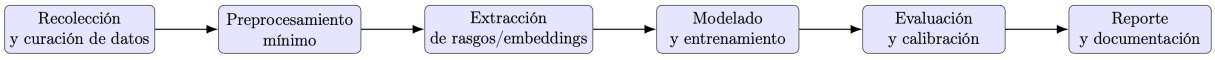

Figura 3.1: Diagrama de alto nivel de la metodología.

En la Figura 3.1 se muestra un resumen del flujo completo de la metodología, desde la preparación del corpus hasta la entrega de resultados reproducibles. En particular, la secuencia enfatiza que el proceso no es exclusivamente de entrenamiento, sino un pipeline integral que asegura (i) calidad de datos, (ii) coherencia en la representación del texto y (iii) evaluación responsable.

Desglose por etapas:

- **Recolección y curación de datos:** Integra textos humanos y textos generados, verificando consistencia de etiquetas, corrigiendo defectos de estructura (p. ej., separadores, saltos de línea embebidos) y eliminando duplicados exactos y cuasi-duplicados. Además, se definen criterios de inclusión (idioma, rangos de longitud, completitud de campos) y se conservan metadatos (fuente, prompt, partición, hash de contenido) para habilitar particiones controladas y auditoría posterior. En adicion se realiza un análisis exploratorio del corpus mediante estadísticos básicos descriptivos.
- **Preprocesamiento mínimo:** Normaliza el texto sin eliminar señales

estilísticas relevantes: estandariza Unicode, espacios, comillas y codificaciones; preserva puntuación y mayúsculas cuando aportan información discriminante; y evita transformaciones agresivas (stopwords, *stemming*) por defecto.

- **Extracción de rasgos/*embeddings*:** Construye dos vistas complementarias del texto: rasgos estilométricos (lexicales, sintácticos y discursivos) y representaciones semánticas (*embeddings*). En la práctica, esta etapa convierte el texto crudo en matrices numéricas comparables, cuidando que cada transformación sea determinista (mismas versiones de tokenizador, mismas reglas de normalización, mismas semillas) para que los resultados sean replicables. La idea de usar representaciones profundas se apoya en arquitecturas Transformer, cuya formulación "dispensa recurrencia y convoluciones" al basarse en mecanismos de atención.

- **Modelado y entrenamiento:** Aplica modelos clásicos y neuronales sobre las representaciones generadas, manteniendo una configuración estable para comparaciones justas: mismo protocolo de partición, mismos criterios de búsqueda de hiperparámetros y mismos reportes de métricas. Además, se incorporan estrategias para estabilizar el aprendizaje (regularización, control de desbalance, *early stopping*) y para evitar conclusiones optimistas por exploración excesiva del conjunto de prueba.

- **Evaluación y calibración:** Mide el rendimiento con métricas robustas basadas en curvas *Receiver Operating Characteristic* (ROC) y *Precision–Recall* (PR), y evalúa la calibración probabilística para que los puntajes sean interpretables como probabilidades. Esta distinción es clave: un modelo puede separar clases y aun así estar mal calibrado; por ello se utilizan procedimientos de calibración post-hoc reportados como efectivos en redes modernas (Guo et al., 2017) y se adoptan decisiones por umbral bajo restricciones explícitas de falsos positivos.

- **Reporte y documentación:** Finalmente, consolida los resultados con

tablas, figuras y descripción metodológica para asegurar reproducibilidad. Incluye el registro de parámetros, semillas, versiones de librerías, hashes de datos y artefactos (features, modelos, calibradores), habilitando replicación por terceros y auditorías de error centradas en falsos positivos y sesgos.

3.1.3 Descripción a detalle del método propuesto

En esta subsección se detallan los componentes operativos y su interacción: análisis exploratorio de datos (p. ej., cuantificación de palabras comunes), fusión de características, líneas base (SVM/Random Forest/XGBoost), ajuste fino de un Transformer, ensamble y calibración, seguidos por validación, selección de umbral y pruebas de robustez. Este nivel de detalle es importante porque buena parte de los fallos en detección no provienen del "tipo de modelo", sino de decisiones de partición, de representación y de evaluación que introducen sesgos sistemáticos.

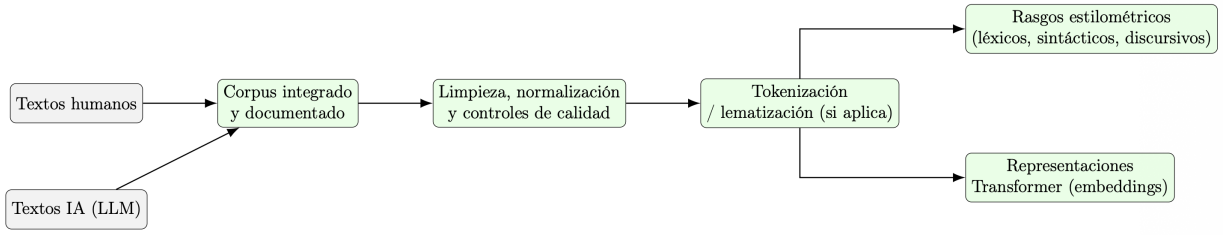

Figura 3.2: Diagrama de bajo nivel (I): adquisición, limpieza y representación del texto.

Descripción (bajo nivel I). En la Figura 3.2 se detalla la fase de adquisición, limpieza y representación del texto. Se enfatiza la construcción de un corpus integrado y la obtención de dos representaciones complementarias: rasgos estilométricos y *embeddings*. La finalidad es capturar tanto patrones de estilo como señales semánticas, conservando la mayor cantidad de información relevante del texto original, dado que la literatura en atribución/estilometría muestra que la señal puede residir en marcadores distribucionales finos.

Desglose por etapas:

- **Integrar y documentar el corpus:** Se unifican los textos en una sola estructura con metadatos de origen y etiquetas claras, incorporando identificadores estables (hash del texto) para rastrear deduplicación y evitar colisiones entre splits. Esta práctica habilita auditorías posteriores

y reduce la probabilidad de que variaciones triviales del mismo contenido aparezcan en entrenamiento y prueba, lo cual inflaría artificialmente las métricas. El análisis exploratorio del corpus se realiza mediante estadísticos como conteo de ejemplos por clase, longitud del texto y conteo de palabras.

- **Limpiar, normalizar y controlar calidad:** Se corrigen errores de codificación y separadores, se eliminan duplicados y se validan longitudes para evitar extremos que distorsionen el entrenamiento. En la práctica, se implementan chequeos automáticos (estadísticas de longitud, proporción de caracteres no alfabéticos, tasas de repetición) y revisiones manuales por muestreo para detectar casos patológicos (texto truncado, etiquetas erróneas, campos vacíos).

- **Aplicar tokenización/lematización:** Se usa cuando mejora la consistencia léxica para ciertos *baselines* (p. ej., frecuencia de término–frecuencia inversa de documento (TF–IDF)), pero se evita si puede eliminar señales estilísticas importantes. La decisión es conservadora: en estilometría se preservan frecuencias de palabras funcionales, puntuación y patrones de segmentación; y en Transformers se opera sobre texto cercano al crudo, alineado con su preentrenamiento.

- **Extraer rasgos estilométricos:** Se extraen indicadores léxicos, sintácticos y discursivos como longitudes (media y dispersión) de oración/palabra, riqueza léxica, distribución de palabras funcionales, n-gramas de caracteres/palabras, patrones de puntuación, densidad de conectores y perfiles de categorías gramaticales (part-of-speech, POS). Esta familia de rasgos está respaldada por la literatura de atribución de autoría, donde el estilo se modela como un conjunto de marcadores distribucionales relativamente estables.

- **Obtener *embeddings* Transformer:** Finalmente, se obtienen representaciones semánticas que sintetizan el contenido global del texto y captu-

ran dependencias de largo alcance mediante atención. En términos operativos, se define una estrategia de agregación (p. ej., pooling del token [CLS] o promedio de estados ocultos) y se asegura consistencia del tokenizador/longitud máxima, para que las comparaciones entre modelos sean controladas.

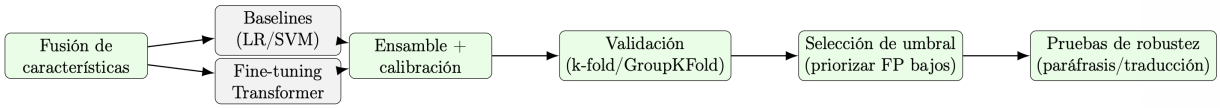

Figura 3.3: Diagrama de bajo nivel (II): modelado, evaluación y pruebas de robustez.

La Figura 3.3 muestra una profundización de la fase de modelado, evaluación y robustez, así como, una ruta de entrenamiento para modelos clásicos y *Transformer*, la combinación mediante ensambles y la calibración probabilística. La validación y la selección de umbral se presentan como mecanismos de control para minimizar falsos positivos, con pruebas de robustez que evalúan generalización ante cambios del texto. Esta última dimensión es crítica porque ataques de parafraseo pueden reducir tasas de detección sin degradar demasiado la calidad superficial del texto, evidenciando vulnerabilidades prácticas.

Desglose por etapas:

- **Fusionar características:** Integra rasgos estilométricos y *embeddings* para aprovechar señales complementarias (forma + contenido). En implementación, se estandarizan escalas (normalización/estandarización) y se evalúan esquemas de concatenación o combinación tardía, con análisis de ablación para cuantificar la contribución de cada vista (Wu et al., 2025).
- **Establecer *baselines* (SVM/Random Forest/XGBoost):** Ofrecen una referencia estable y explicable, particularmente sobre TF–IDF y rasgos estilométricos. Además de servir como comparación, estos modelos facilitan interpretación (vectores de soporte en SVM, importancia de va-

riables en Random Forest/XGBoost) y permiten evaluar cuánta ganancia real aporta el ajuste fino de Transformers bajo el mismo protocolo experimental (Davis & Goadrich, 2006).

- **Ajustar un Transformer (*fine-tuning*):** Ajusta un modelo preentrenado a la tarea binaria, capturando contexto profundo. En práctica, se controla el sobreajuste con validación, regularización y *early stopping*, y se reportan configuraciones esenciales (*batch size*, *learning rate*, máxima longitud) para reproducibilidad.
- **Combinar y calibrar:** Combina predicciones para mejorar estabilidad (p. ej., promedio de probabilidades) y aplica calibración para producir probabilidades interpretables. La calibración se justifica porque redes modernas pueden estar "*miscalibrated*", y métodos simples como *temperature scaling* o variantes de Platt scaling pueden corregirlo de forma efectiva (Guo et al., 2017; Lin et al., 2007).
- **Validar con k-fold/GroupKFold (k=5):** Asegura evaluaciones justas y controla *leakage* cuando existen metadatos compartidos (fuente/prompt). El uso de particiones por grupos se alinea con la recomendación de impedir que instancias correlacionadas aparezcan en entrenamiento y prueba, lo cual inflaría el desempeño.
- **Seleccionar umbral (FPR bajo):** Define el punto operativo considerando el impacto de falsos positivos en contextos educativos. Formalmente, se plantea como un problema de control de error tipo I (FPR) bajo un nivel α prefijado, coherente con el paradigma Neyman–Pearson, que busca minimizar error tipo II sujeto a una cota en el error tipo I (Scott & Nowak, 2005; Tong et al., 2018).

Recolección y curación de datos

Esta etapa cubre la recolección y curación del corpus, los criterios de inclusión y limpieza, y la partición con controles de *leakage*. La motivación central es que una evaluación creíble requiere garantizar que el conjunto de prueba sea efectivamente "no visto" y no comparta trazas ilegítimas con entrenamiento (Kaufman et al., 2012).

- **Obtener el corpus:** LLM — Detect AI Generated Text Dataset (Kaggle), documentando de forma explícita el origen del dataset (dataset card), su licencia y cualquier metadato disponible (fuente/prompt) que permita prevenir correlaciones falsas entre divisiones.
- **Definir inclusión y limpieza:** idioma (inglés alto, con exploración al español), longitud 200–5,000 tokens, deduplicación (hash/MinHash) y preservación de metadatos para particionado por grupos. La deduplicación se asume como una medida preventiva contra inflación de métricas por repetición parcial, un problema frecuente cuando se mezclan fuentes o se reusan prompts.
- **Particionar datos:** train/test 80/20 estratificado. Selección de hiperparámetros por validación cruzada; GroupKFold si hay metadatos de fuente/prompt para mitigar *leakage*. Semilla global fija y test ciego, reduciendo el riesgo de *"tuning"* indirecto sobre el conjunto de prueba.

Preprocesamiento mínimo

Se aplica el preprocesamiento mínimo necesario para alimentar los modelos posteriores, preservando señales de estilo relevantes.

- **Aplicar preprocesamiento mínimo:** normalización Unicode; estandarización de comillas/espacios; conservar mayúsculas y puntuación; sin *stemming* por defecto; tokenización/lematización sólo si es necesario. Esta cautela se justifica porque los marcadores de estilo son sensibles a transformaciones agresivas, mientras que los Transformers operan sobre texto cercano al crudo conforme a su preentrenamiento (Stamatatos, 2009; Vaswani et al., 2017).

3.1.4 Extracción de rasgos/*embeddings*

Se combinan vistas complementarias del texto para enriquecer la señal del clasificador y reducir dependencia de una sola familia de rasgos. Esta estrategia es coherente con la evidencia en atribución/estilometría: algunos rasgos capturan "huellas" de forma, mientras que otros capturan contenido y coherencia semántica (Stamatatos, 2009; Koppel et al., 2009).

- **Representar la semántica:** TF–IDF de n-gramas (1–3) y *embeddings* documentales derivados de Transformers. En la práctica, TF–IDF captura patrones locales y repetición; los *embeddings*capturan relaciones de largo alcance mediante atención, apoyados en la formulación Transformer (Vaswani et al., 2017).
- **Extraer estilometría:** longitud media y dispersión de oración/palabra, type–token ratio, distribución POS, densidad de palabras funcionales, repetición de n-gramas, entropía de caracteres y patrones de puntuación. Estos rasgos siguen prácticas clásicas en atribución de autoría, donde el estilo se operacionaliza como distribuciones de marcadores relativamente estables (Stamatatos, 2009; Koppel et al., 2009).
- **Calcular fluidez (auxiliar):** se estima la *pseudo-perplejidad* con un modelo de lenguaje (LM) fijo; no se usa como criterio único de decisión. Esto evita reducir el problema a una sola señal, potencialmente frágil ante OOD o ataques (Wu et al., 2025; Sadasivan et al., 2023).

3.1.5 Modelado y entrenamiento

En esta etapa se comparan líneas base clásicas, una CNN como modelo principal y un modelo Transformer ajustado, además de explorar un ensamble simple; se aplican técnicas de calibración probabilística para obtener puntajes interpretables y adecuados para decisión por umbral. La calibración es central porque existe evidencia de que modelos modernos pueden estar mal calibrados y que procedimientos post-hoc simples pueden mejorar sustancialmente la confiabilidad probabilística (Guo et al., 2017; Niculescu-Mizil & Caruana, 2005).

- **Entrenar el *baseline*:** Regresión Logística y Linear SVM sobre TF–IDF + rasgos. Se controlan hiperparámetros (C, regularización) y se reportan configuraciones, evitando comparaciones injustas por tuning desigual (Arlot & Celisse, 2010).
- **Entrenar la CNN:** arquitectura convolucional 1D para texto (embeddings → Conv1D → *pooling* global → densa), con regularización (dropout/L2) y *early stopping*. Se reportan hiperparámetros clave (filtros, kernel, longitud máxima) por su impacto directo en el rendimiento (Kim, 2014; Zhang et al., 2015).
- **Ajustar el modelo *Transformer*:** ajuste fino para clasificación binaria, con control de sobreajuste mediante validación y *early stopping*. Se reportan decisiones de tokenización y longitud máxima por su impacto directo en el desempeño (Vaswani et al., 2017).
- **Ensamblar y calibrar:** combinación (promedio de probabilidades) entre el mejor *baseline* y el mejor *Transformer*; calibración (Platt/Isotónica/ *Temperature scaling*) para probabilidades confiables. Platt scaling y sus mejoras son estándar en SVM (Lin et al., 2007) y la calibración isotónica es una alternativa clásica para mapear scores a probabilidades (Zadrozny & Elkan, 2002).

3.1.6 Evaluación y calibración

Se definen las métricas principales, se fija un umbral operativo con restricciones de falsos positivos y se evalúa la robustez ante paráfrasis, traducción y desplazamientos de dominio, incluyendo análisis de calibración y sesgos. La justificación es que un buen valor de área bajo la curva ROC (AUROC) no garantiza decisiones seguras cuando se exige FPR muy bajo; por ello se adopta una selección de umbral compatible con un control explícito del error tipo I (Scott & Nowak, 2005; Tong et al., 2018).

- **Definir métrica primaria:** área bajo la curva de precisión-recuperación (AUPRC); secundarias: AUROC, F1/F2, coeficiente de correlación de Matthews (MCC), con intervalos por bootstrap. Se privilegia PR en escenarios potencialmente desbalanceados porque ofrece interpretaciń más directa del trade-off entre aciertos y falsos positivos (Saito & Rehmsmeier, 2015; Davis & Goadrich, 2006).

- **Fijar umbral operativo:** seleccionar en validación con restricción de FPR $\leq 1\,\%$ y tasa de verdaderos positivos (TPR) competitiva; comparación justa (misma partición/
transformaciones, semillas) y GroupKFold cuando aplique. Esta formulación es coherente con el paradigma Neyman–Pearson, que prioriza controlar el error tipo I por debajo de un nivel fijado (Scott & Nowak, 2005; Tong et al., 2018).

- **Probar robustez:** paráfrasis y traducción ida–vuelta (inglés $\leftrightarrow$ español), domain shift (fuentes no vistas), ablación de grupos de variables y curvas de calibración (Brier/fiabilidad). Esto se alinea con evidencia de que ofuscación (p. ej., reescritura o traducción) puede degradar detectores, y con ataques de paraphraseo que reducen tasas de detección (Weber-Wulff et al., 2023; Sadasivan et al., 2023).

- **Analizar sesgos y desbalance:** reporte de distribución de clases/longitudes;

class weights u oversampling conservador si procede; análisis por longitud/tema y por fuente/prompt. Este análisis es clave para evitar conclusiones globales que oculten fallos sistemáticos en subpoblaciones (Wu et al., 2025; Weber-Wulff et al., 2023).

3.1.7 Análisis comparativo con los resultados publicados en Kaggle y de las herramientas comerciales

La comparación con el estado del arte se plantea como un análisis interpretativo más que como una competencia directa, debido a diferencias en corpus, idioma, dominio y protocolo de evaluación. El contraste se realiza usando métricas reportadas de manera consistente (Accuracy, F1, AUPRC/AUROC cuando estén disponibles), y se discuten las discrepancias bajo las condiciones de cada estudio (tarea binaria o multiclase, tamaño del conjunto y fuente de datos). Esta práctica es necesaria porque comparativas previas muestran dispersión sustantiva de resultados entre herramientas y protocolos (Macko et al., 2023; Weber-Wulff et al., 2023; Akram et al., 2023).

3.2 Resumen del capítulo

Este capítulo presentó la metodología como un pipeline reproducible de extremo a extremo y justificó su enfoque en la calidad experimental. Se describieron la adquisición y curación del corpus, la definición de criterios de inclusión, la partición con controles de *leakage* y el preprocesamiento mínimo orientado a preservar señales estilísticas. Se detalló la extracción de rasgos y *embeddings*, y el entrenamiento de modelos clásicos y de un Transformer con configuraciones comparables. De manera central, se explicitó la arquitectura y el rol de la CNN como modelo principal del estudio, junto con su integración en el esquema de ensamble y calibración. La evaluación incorporó métricas robustas, selección de umbral con restricción de falsos positivos, calibración probabilística y pruebas de robustez ante desplazamientos de dominio. Por último, se establecieron lineamientos de reporte, reproducibilidad y consideraciones éticas para sostener conclusiones confiables y transparentes.

4. Resultados

Contenidos del Capítulo

Como recordatorio, la hipótesis plantea que un detector híbrido que combina (i) rasgos estilométricos multicapas (léxicos, sintácticos y discursivos) y (ii) representaciones semánticas derivadas de modelos Transformer, con calibración probabilística y selección conservadora de umbral, superará el desempeño de detectores generalistas al discriminar textos académicos en inglés generados por

inteligencia artificial (IA), manteniendo una tasa de falsos positivos acotada para su uso en contextos educativos. Este capítulo presenta los resultados a partir de una base de datos de 29,139 textos diferentes entre hechos por humanos y hechos por IA. Para asegurar comparaciones consistentes, se reportan promedios y desviaciones estándar cuando aplica validación cruzada. La interpretación se centra en el comportamiento de cada modelo frente a un mismo problema binario, con especial atención al equilibrio entre falsos positivos y falsos negativos, así como a la estabilidad del rendimiento. Como exploración adicional, se realizó una prueba en español mediante un script que tradujo 4,378 textos de IA y 5,468 textos de humanos del mismo corpus (total 9,846). Esta etapa permite observar la transferencia, pero introduce un posible sesgo: al ser textos traducidos de forma automática, pueden parecer más homogéneos y ser clasificados con mayor frecuencia como IA. Por ello, los resultados deben interpretarse con cautela.

Durante una fase preliminar fue necesario reestructurar el corpus, ya que el archivo CSV original venía mal separado, generando aproximadamente 290,000 líneas y un tamaño cercano a 10 veces el real. En términos prácticos, este tipo de fallas de formateo puede inducir errores silenciosos (por ejemplo, cortes de texto, corrimientos de columnas o etiquetas desalineadas) que contaminan tanto la representación como la validación. Por ello, se desarrolló un script en Python orientado a (i) normalizar separadores y comillas, (ii) unificar filas fragmentadas preservando el texto completo, (iii) verificar invariantes de consistencia (conteos esperados, longitudes válidas, ausencia de nulos críticos) y (iv) registrar versiones/hash del corpus curado para auditoría. Después de este proceso de curación, el corpus quedó consolidado en 29,139 textos reales, que son los utilizados en el resto del pipeline, reduciendo el riesgo de *data leakage* y de inconsistencias de etiquetado que suelen pasar inadvertidas en competiciones o datasets grandes (Kaufman et al., 2012).

4.1 Entorno de desarrollo y experimentación

Para el desarrollo y realización de los experimentos se utilizó un equipo Mac con chip Apple M2, 8 GB de RAM y almacenamiento de 256 GB. Además, se empleó Google Colab como entorno complementario para ejecuciones en la nube y una máquina virtual para pruebas controladas. La Tabla 4.1 sintetiza

Tabla 4.1: Especificaciones del entorno de desarrollo utilizado.

Componente	**Especificación**
Equipo	Mac con chip Apple M2
Sistema operativo	macOS 26
Memoria RAM	8 GB
Almacenamiento	256 GB
Plataformas adicionales	Google Colab; máquina virtual

los recursos disponibles durante la ejecución de los experimentos. El uso de un equipo local con chip M2 permitió iteraciones rápidas de desarrollo, mientras que Google Colab se utilizó para ejecuciones con mayor demanda de cómputo. Este esquema mixto facilitó pruebas repetibles bajo el mismo pipeline de datos y código, reduciendo tiempos de espera y manteniendo consistencia metodológica.

La máquina virtual se utilizó para ejecuciones controladas y reproducibles en un entorno Linux. Sus especificaciones se resumen en la Tabla 4.2.

Tabla 4.2: Especificaciones de la máquina virtual utilizada.

Componente	**Especificación**
Arquitectura	x86_64
CPU	Intel(R) Xeon(R) CPU @ 2.20GHz
CPU(s)	2
Threads por core	2
Cores por socket	1
Sockets	1
Memoria RAM	12 GB
Almacenamiento	108 GB (87 GB disponibles)

La implementación de los modelos se realizó en Python (3.12.12) utilizando las bibliotecas listadas en la Tabla 4.3. La Tabla 4.3 muestra las dependen-

Tabla 4.3: Bibliotecas de Python utilizadas para la implementación.

Biblioteca	**Versión**	**Uso principal**
NumPy	2.0.2	Operaciones numéricas y arreglos
Pandas	2.2.2	Manipulación de datos tabulares
Matplotlib/Seaborn	3.10.0	Visualización de resultados
Scikit-learn	1.6.1	Modelos clásicos, métricas y validación
TensorFlow/Keras	2.19.0	Modelos neuronales y entrenamiento
NLTK	3.9.1	Tokenización, stopwords y lematización
WordCloud	1.9.4	Nubes de palabras
Librerías estándar	—	Manejo de archivos y utilidades (os, json, etc.)
Google Colab	—	Integración con Drive

cias principales y su función dentro del flujo de trabajo. En conjunto, estas bibliotecas cubren la preparación del corpus, la representación del texto, el entrenamiento de modelos clásicos y neuronales, así como la evaluación y la visualización de resultados. Esta combinación permitió sostener un proceso completo y reproducible, desde la limpieza de datos hasta la interpretación final.

4.2 Experimentos

En todos los experimentos se utilizó el conjunto *"TextTuring"* obtenido de la competencia de Kaggle `LLM Detect AI Generated Text Dataset` disponible en Kaggle.[1] El objetivo del conjunto es entrenar modelos que distingan si un ensayo fue escrito por un estudiante o por un LLM. El corpus integra ensayos humanos y ensayos generados por distintos modelos, y contiene más de 29,139 textos. En el corpus utilizado se cuenta con 17,508 textos humanos y 11,631 textos de IA. Las variables principales son **text** (contenido del ensayo) y **generated** (etiqueta objetivo: 0 = humano, 1 = IA). Todos los modelos reportados en esta sección (XGBoost, Random Forest, SVM, BERT, CNN y red neuronal convolucional para texto (TextCNN)) fueron desarrollados para este corpus en particular. A continuación se agrupan los resultados en dos experimentos: modelos clásicos y modelos neuronales.

4.2.1 Análisis Exploratorio del conjunto de datos TextTuring

Para complementar el análisis cuantitativo, se realizaron visualizaciones léxicas con la frecuencia relativa de términos, así como nubes de palabras con los vocablos más frecuentes en textos generados por IA y en textos humanos. Estas figuras permiten identificar patrones de estilo y frecuencia léxica que no siempre se reflejan en las métricas globales. La comparación cualitativa se apoya en estas visualizaciones para contrastar distribuciones de vocabulario y posibles sesgos de generación. Como referencia descriptiva del corpus, la Tabla 4.4 resume los textos "más cortoz "más largo", junto con sus conteos básicos. La Figura 4.1 compara la distribución relativa de términos en ambos grupos. Esta representación ayuda a detectar palabras o construcciones que aparecen

[1] `https://www.kaggle.com/datasets/sunilthite/llm-detect-ai-generated-text-dataset`

Tabla 4.4: Ejemplos de textos extremos del corpus.

Tipo	Caract.	Pal.	Tokens	Idx	Fragmento
Más corto	238	48	49	16414	*Dear TEACHER_NAME, I WRITE THIS LETTER TO SAID THAT IS NOT FAIR THAT Only the people that have good grades can play in one of the sport. All the people with A, or B can play because the C is the baldest grade on the school. Because people ...*
Más largo	9151	1650	1649	3367	*Phones & driving I strongly agree with cell phones being banned while driving. I can honestly say that I have almost run into someone's back on numerous occasions because I was focused on texting on my phone. I think driving while using cell phones should be banned because of the many car accidents ...*

Tabla 4.5: Percentiles de longitud en caracteres (char_len).

Percentil	Caracteres
P1	612
P5	901
P50	2152
P95	4058
P99	5225

de forma desproporcionada en textos de IA o humanos, lo que sugiere rasgos estilísticos distintivos. Además, permite identificar si hay concentraciones semánticas o vocablos dominantes que pudieran explicar la separabilidad entre clases. Este tipo de información respalda la efectividad de enfoques basados en n-gramas y convoluciones, que capturan patrones de frecuencia y coocurrencia con sensibilidad a diferencias locales.

La Figura 4.2 muestra las palabras de mayor presencia en textos generados por IA. La predominancia de términos recurrentes indica una tendencia a reutilizar vocabulario y estructuras similares, lo que produce patrones detectables por modelos de clasificación. En particular, la recurrencia de ciertos términos sugiere un estilo más homogéneo y menos diverso. Esta concentración léxica ayuda a explicar el alto desempeño de las arquitecturas convolucionales, sensibles a patrones locales y repeticiones de n-gramas.

La Figura 4.3 evidencia mayor diversidad léxica y variación de conectores

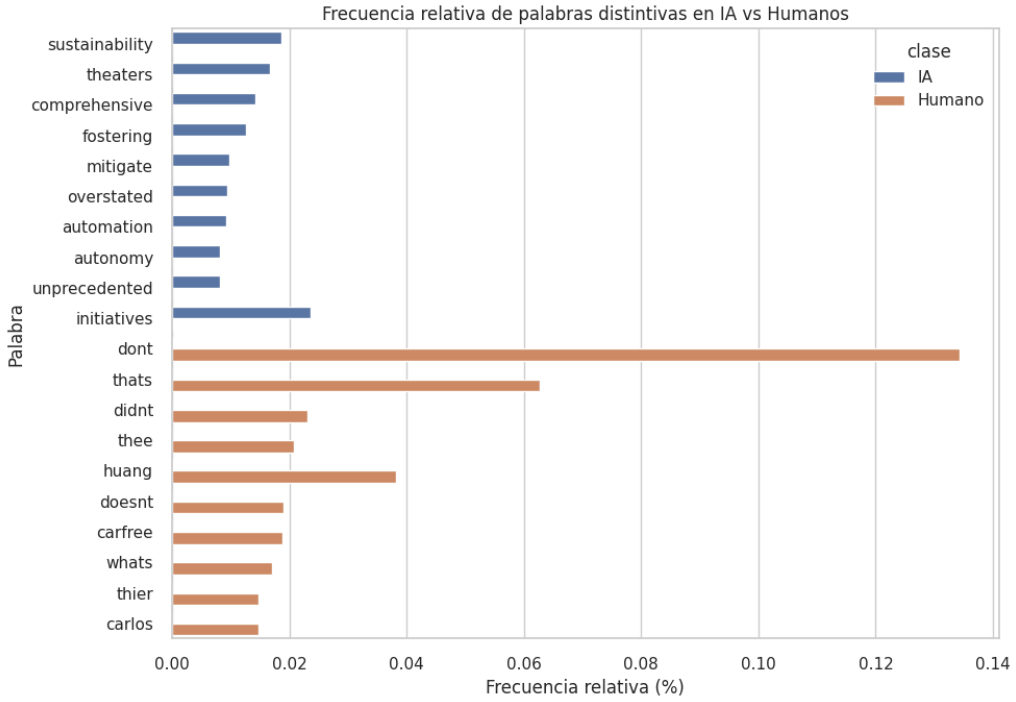

Figura 4.1: Frecuencia relativa de términos en textos de IA y humanos.

discursivos en los textos humanos. La heterogeneidad observada sugiere un estilo menos repetitivo y más flexible, lo que incrementa la complejidad de la separación de clases y eleva la exigencia para los modelos. Aún así, los modelos entrenados logran distinguir estas diferencias con alta precisión, lo que indica que existen señales discriminantes suficientes incluso en estilos humanos más variables. Estas figuras se interpretan como apoyo cualitativo a los modelos: los enfoques basados en TF–IDF y convoluciones capturan precisamente patrones léxicos y n-gramas que suelen manifestarse en estas distribuciones de frecuencia, y complementan la evidencia cuantitativa de las métricas.

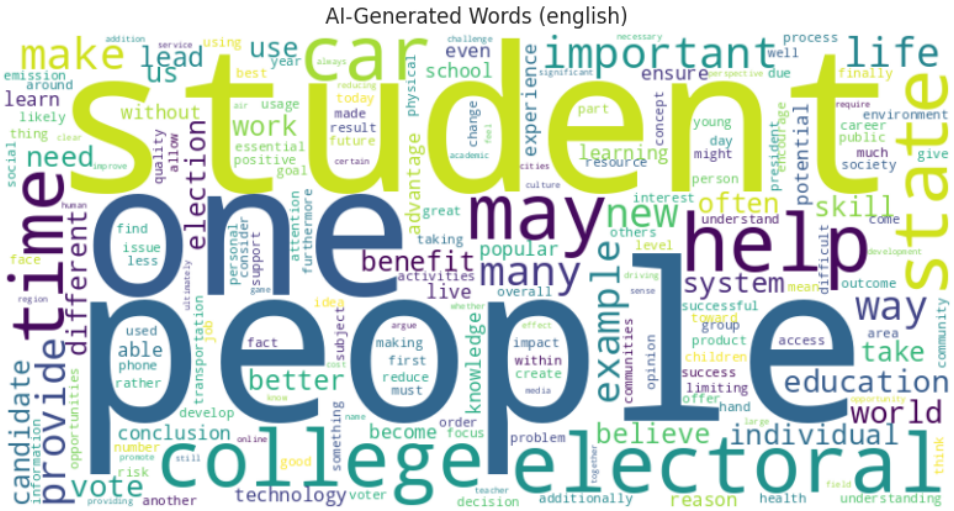

Figura 4.2: Palabras más comunes en textos generados por IA.

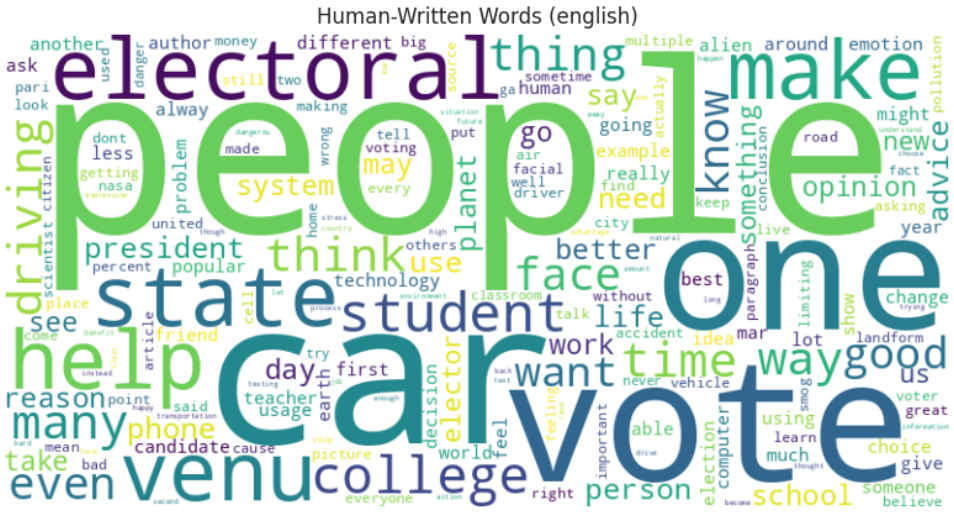

Figura 4.3: Palabras más comunes en textos generados por humanos.

4.2.2 Experimento 1: Modelos clásicos en el conjunto de datos TextTuring

Este experimento reúne los métodos tradicionales de clasificación basados en representaciones de frecuencia de término–frecuencia inversa de documento (TF–IDF) y modelos lineales o de ensamble. Se consideran SVM, Random Forest y XGBoost como líneas base.

SVM

Se construye un pipeline con TF–IDF (unigramas y bigramas, 5,000 características, *stopwords* en inglés) y un clasificador LinearSVC. La evaluación se realiza con validación cruzada estratificada de 5 *folds*. SVM alcanza un **ROC-AUC de 0.9034 ± 0.0007**, lo que refleja un rendimiento inferior frente a los modelos de ensamble en este corpus.

Tabla 4.6: Métricas de SVM.

Métrica	**Valor**
Accuracy	0.8751 ± 0.0006
Precision	0.9000 ± 0.0011
Recall	0.8700 ± 0.0015
F1-score	0.8700 ± 0.0007
ROC-AUC	**0.9034 ± 0.0007**

Random Forest

Se incluyó Random Forest como *baseline* clásico dentro del experimento para contrastar el efecto de modelos de ensamble sobre rasgos TF–IDF (ver Tabla 4.7). La evaluación se realiza con validación cruzada estratificada de 5 *folds*. Random Forest alcanza un **ROC-AUC de 0.9849 ± 0.0024**, mostrando un desempeño competitivo dentro de las líneas base clásicas.

Tabla 4.7: Métricas de Random Forest.

Métrica	**Valor**
Accuracy	0.9880 ± 0.0019
Precision	0.9913 ± 0.0022
Recall	0.9785 ± 0.0030
F1-score	0.9849 ± 0.0024
ROC-AUC	**0.9849 ± 0.0024**

XGBoost

Se utiliza preprocesamiento léxico con normalización y lematización, seguido de un esquema TF–IDF combinado de palabras y caracteres. Para 29,139 documentos se emplea un límite de 12,000 características de palabras y 4,000 de caracteres, con $min\_df \approx 0{,}2\,\%$ y $max\_df = 0{,}95$. Los n-gramas considerados son unigramas y bigramas de palabras, y trigramas a pentagramas de caracteres. La evaluación se realiza con validación cruzada estratificada de 5 *folds* (ver Tabla 4.8). XGBoost alcanza un **ROC-AUC de 0.9993 ± 0.0002**,

Tabla 4.8: Métricas de XGBoost.

Métrica	**Valor**
Accuracy	0.9906 ± 0.0013
Precision	0.9901 ± 0.0017
Recall	0.9862 ± 0.0042
F1-score	0.9882 ± 0.0016
ROC-AUC	**0.9993 ± 0.0002**

indicando separación casi perfecta entre textos humanos e IA en este corpus.

4.2.3 Experimento 2: Modelos neuronales en el conjunto de datos TextTuring

Este experimento agrupa los modelos basados en representaciones profundas, con BERT como *baseline* contextual y CNN/TextCNN como arquitecturas principales para la comparación en redes convolucionales.

BERT

En términos generales, BERT modela el contexto bidireccional mediante capas Transformer y se ajusta finamente para clasificación binaria. La evaluación se realiza con validación cruzada estratificada de 5 *folds* (ver Tabla 4.9). BERT

Tabla 4.9: Métricas de BERT.

Métrica	**Valor**
Accuracy	0.9716 ± 0.0025
Precision	0.9746 ± 0.0041
Recall	0.9557 ± 0.0038
F1-score	0.9651 ± 0.0031
ROC-AUC	$\mathbf{0.9929 \pm 0.0008}$

alcanza un **ROC-AUC de $\mathbf{0.9929 \pm 0.0008}$**, lo que refleja alta capacidad discriminativa en la tarea, aunque con margen frente a los modelos convolucionales evaluados en este corpus.

CNN

La CNN base emplea vocabulario de 10,000 términos y *embeddings* de 64 dimensiones, una capa Conv1D con 64 filtros y kernel 5, *GlobalMaxPooling1D*, y dos capas densas antes de la salida sigmoidal. Se entrena con entropía cruzada binaria y Adam, con *batch size* 64 y 5 épocas, usando validación cruzada estratificada de 5 *folds*, pesos de clase balanceados y *early stopping* (Kim, 2014).

Este experimento es el eje central de la investigación. Ver Tabla 4.10. CNN

Tabla 4.10: Métricas de CNN.

Métrica	**Valor**
Accuracy	0.9939 ± 0.0013
Precision	0.9935 ± 0.0023
Recall	0.9912 ± 0.0023
F1-score	0.9924 ± 0.0016
ROC-AUC	$\mathbf{0.9997 \pm 0.0002}$

alcanza un **ROC-AUC de 0.9997 ± 0.0002**, indicando una separación casi perfecta entre textos humanos e IA.

TextCNN

El modelo TextCNN utiliza múltiples kernels de convolución para capturar patrones locales de distinta longitud. Esta configuración en paralelo enriquece la representación sin perder estabilidad. La evaluación se realiza con validación cruzada estratificada de 5 *folds*. Ver Tabla 4.11. TextCNN alcanza un **ROC-**

Tabla 4.11: Métricas de TextCNN.

Métrica	**Valor**
Accuracy	0.9935 ± 0.0013
Precision	0.9922 ± 0.0037
Recall	0.9914 ± 0.0034
F1-score	0.9918 ± 0.0016
ROC-AUC	$\mathbf{0.9997 \pm 0.0002}$

AUC de 0.9997 ± 0.0002, con desempeño equivalente al de la CNN en términos de discriminación.

La Tabla 4.12 evidencia que ambos modelos convolucionales mantienen un desempeño muy cercano, con diferencias pequeñas en *Accuracy* y *F1-score*. La CNN obtiene una ventaja ligera en precisiones globales, mientras que TextCNN logra un recall marginalmente mayor, lo cual puede ser relevante si se prioriza

Tabla 4.12: Comparativa de métricas entre CNN y TextCNN.

Modelo	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**
CNN	**0.9939**	**0.9935**	0.9912	**0.9924**	**0.9997**
TextCNN	0.9935	0.9922	**0.9914**	0.9918	**0.9997**

sensibilidad. Esta comparación refuerza que el diseño base de CNN es suficiente para capturar patrones locales efectivos en el corpus.

Visualizaciones del CNN

Las siguientes figuras sintetizan el desempeño promedio y la calidad de las predicciones del modelo CNN, incluyendo curvas de discriminación y calibración. Se agrega una explicación amplia para que las gráficas no queden aisladas y puedan interpretarse como parte del análisis. La Figura 4.4 resume los promedios y la variabilidad de las métricas clave en la validación cruzada. La concentración de valores altos y con dispersión acotada sugiere que el rendimiento del modelo es estable entre folds. Además, la cercanía entre *Accuracy, Precision* y *F1-score* indica un balance saludable entre errores de tipo I y tipo II. Esta estabilidad es coherente con el uso de pesos de clase balanceados y con el control del largo máximo por percentil, lo que reduce el impacto de secuencias atípicamente largas.

En la Figura 4.5 la curva ROC se aproxima al vértice superior izquierdo, lo que refleja una alta tasa de verdaderos positivos con una baja tasa de falsos positivos. Esto concuerda con el ROC-AUC de 0.9997 reportado en el experimento, indicando una separación casi perfecta entre textos humanos e IA. La forma de la curva confirma que el modelo conserva capacidad discriminativa a lo largo de distintos umbrales de decisión, lo que es valioso para ajustar umbrales conservadores sin sacrificar rendimiento global.

La Figura 4.6 muestra la relación entre precisión y recall para distintos umbrales. La permanencia de una precisión alta en un rango amplio de recall es

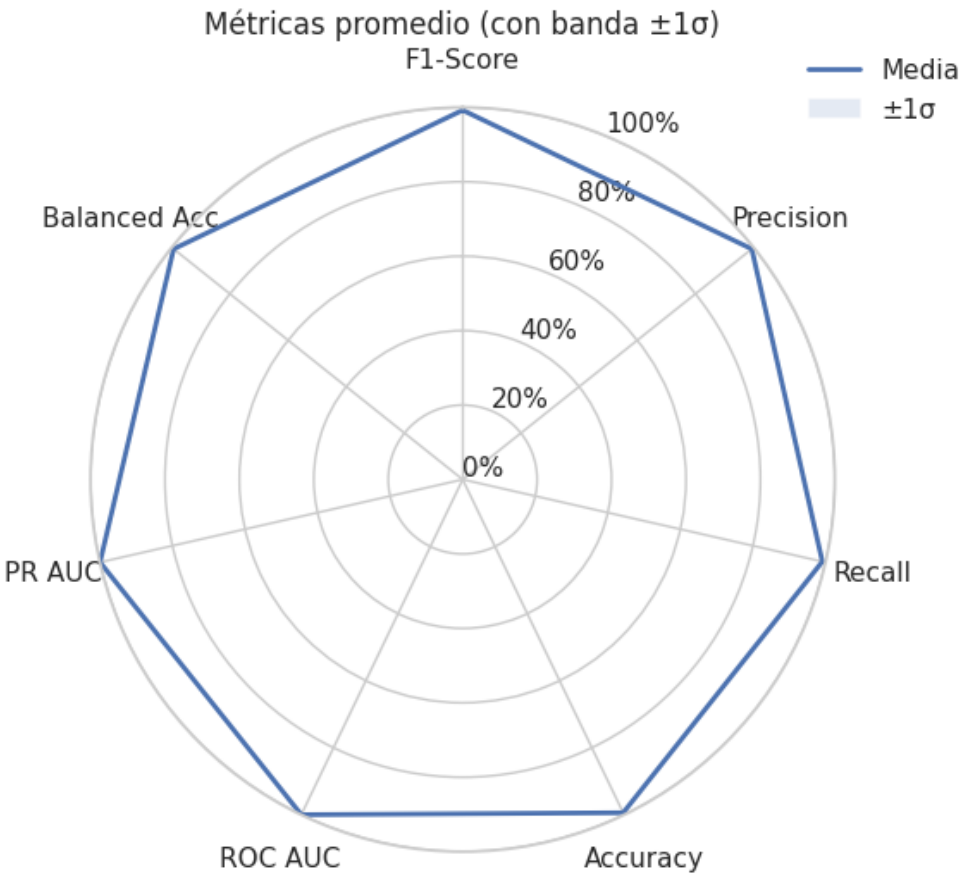

Figura 4.4: Métricas promedio del modelo CNN en validación cruzada.

coherente con el F1-score obtenido y sugiere un buen equilibrio entre errores de tipo I y tipo II. Este comportamiento es especialmente relevante en contextos educativos, donde es deseable minimizar falsos positivos sin sacrificar detecciones reales de textos generados por IA.

La Figura 4.7 evalúa la coherencia entre probabilidades predichas y frecuencias observadas. La cercanía a la línea ideal, junto con un Brier bajo, indica que las probabilidades emitidas por el modelo son informativas y confiables para la toma de decisiones. Esta propiedad es clave cuando se utilizan umbrales operativos, ya que permite interpretar el puntaje de salida como una medida de confianza y facilita la definición de políticas de revisión manual.

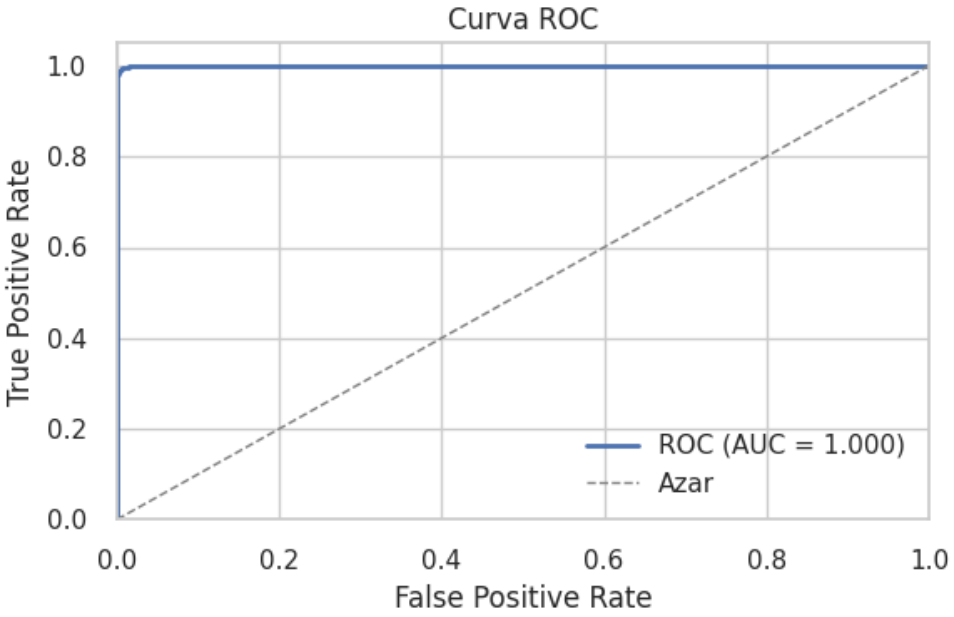

Figura 4.5: Curva ROC del modelo CNN.

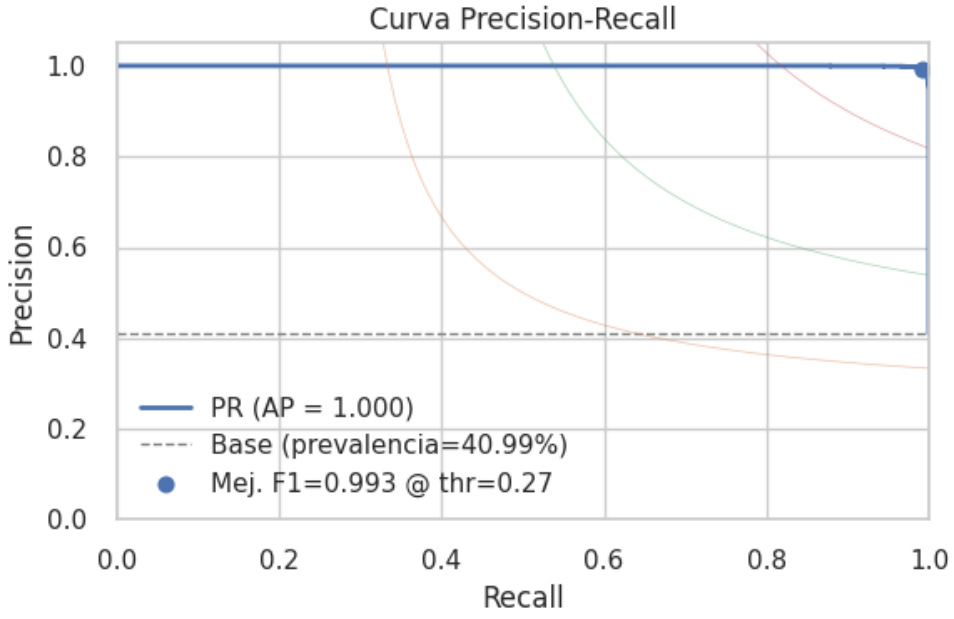

Figura 4.6: Curva Precision-Recall del modelo CNN.

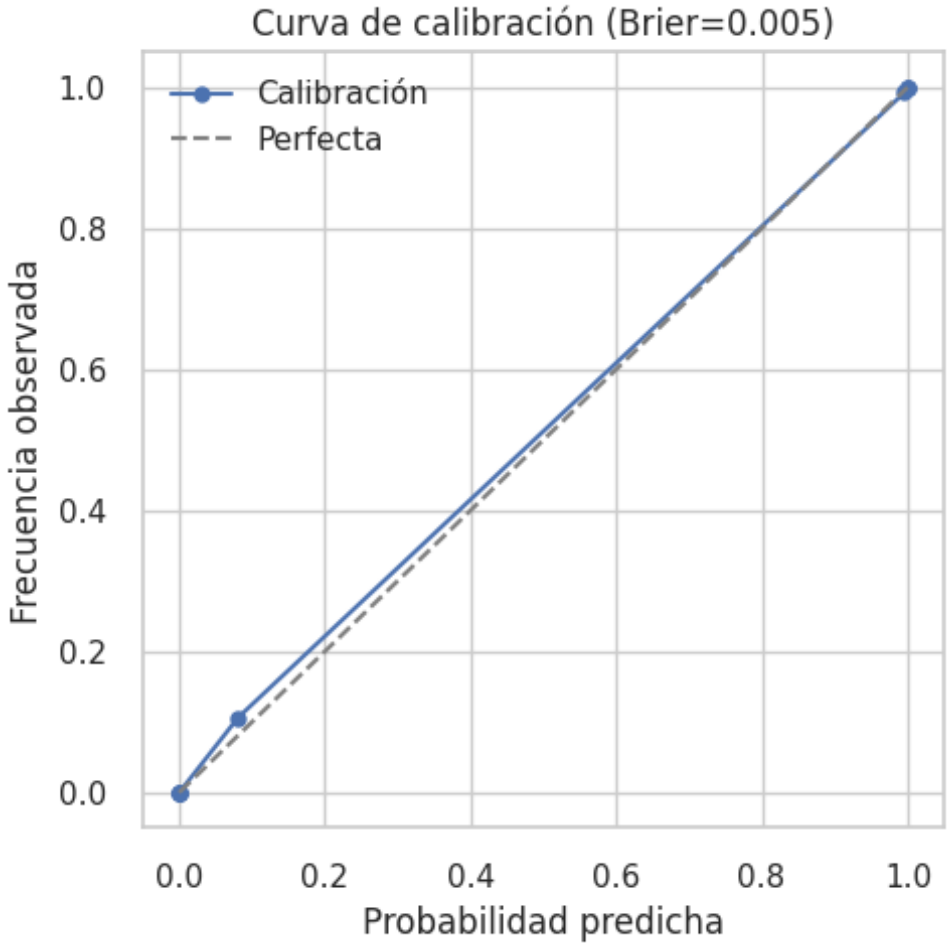

Figura 4.7: Curva de calibración del modelo CNN.

4.3 Comparativa: mejor método clásico vs CNN

Dentro de los modelos clásicos, XGBoost presenta el mejor balance global (*Accuracy* 0.9906, *F1-score* 0.9882). Al comparar con los modelos neuronales, la CNN obtiene los mejores valores en *Accuracy*, *Precision* y *F1-score*, mientras que TextCNN alcanza el mejor *Recall* y empata en *ROC-AUC*. En conjunto, la CNN mantiene un rendimiento general superior y un comportamiento probabilístico consistente.

Tabla 4.13: Comparativa de métricas entre XGBoost, CNN y TextCNN.

Modelo	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC-AUC**
XGBoost	0.9906	0.9901	0.9862	0.9882	0.9993
CNN	**0.9939**	**0.9935**	0.9912	**0.9924**	**0.9997**
TextCNN	0.9935	0.9922	**0.9914**	0.9918	**0.9997**

La Tabla 4.13 permite visualizar de forma compacta la superioridad de la CNN en métricas clave y la cercanía de TextCNN en recall y ROC-AUC. En conjunto, se observa que la CNN ofrece el mejor equilibrio global, mientras que TextCNN actúa como una alternativa competitiva cuando se prioriza sensibilidad. Esta comparativa evidencia que la CNN aporta ventajas en discriminación e interpretabilidad de probabilidades, justificando su papel como modelo estrella del trabajo.

4.4 Comparativa con Kaggle y herramientas comerciales de detección de textos generados por Inteligencia Artificial vs. generada por Humanos

Esta sección organiza la discusión comparativa en dos etapas: (i) una referencia externa con la competencia de Kaggle (*TextTuring*) frente a los modelos propuestos, y (ii) una comparación final del mejor modelo (CNN) con herramientas comerciales.

4.4.1 Referencia con la competencia Kaggle vs modelos propios

La competencia *LLM – Detect AI Generated Text* de Kaggle utiliza **ROC-AUC** como métrica oficial. En los experimentos realizados, los modelos propuestos (XGBoost, CNN y TextCNN) alcanzan valores de **ROC-AUC** entre **0.9993** y **0.9997** (Tabla 4.13). Estos resultados superan el rango reportado en el *leaderboard* privado de Kaggle (alrededor de **0.96**), aunque la comparación es referencial y no estrictamente directa, dado que el protocolo y los *splits* del *test* oculto no son equivalentes.

Para contextualizar ese rango, la Tabla 4.14 resume el top 10 del *leaderboard* privado según la captura disponible e incorpora los modelos propios para estimar su posición por ROC-AUC. En la tabla se presentan los valores redondeados a cuatro decimales; el rango de Kaggle va de **0.9878** a **0.9474**, lo que refuerza la distancia frente a los puntajes de los modelos propios.

Tabla 4.14: Top 10 del *leaderboard* privado de Kaggle vs modelos propios (posición estimada por ROC-AUC).

#	**Equipo/Modelo**	**Score (ROC-AUC)**
1	**CNN (propio)**	**0.9997**
2	**TextCNN (propio)**	**0.9997**
3	**XGBoost (propio)**	**0.9993**
4	**BERT (propio)**	**0.9929**
5	(Equipo con emojis)	0.9878
6	Guanshuo Xu	0.9834
7	nlp team	0.9749
8	Ertugrul & Chase	0.9739
9	Linguistic Ninjas	0.9721
10	Davide Cozzolino	0.9691
11	Hao Mei	0.9651
12	Abdullah Meda	0.9565
13	LLMLab	0.9477
14	IC2	0.9473

4.4.2 Comparativa del mejor modelo (CNN) con herramientas comerciales

Con base en los resultados anteriores, el CNN es el modelo con mejor equilibrio global; por ello, la comparativa con herramientas comerciales se realiza exclusivamente con este modelo. Se consideran servicios ampliamente usados en contextos educativos y profesionales: ZeroGPT, GPTZero, Justdone y WinstonAI. El objetivo es ubicar el desempeño del CNN dentro del ecosistema real de detectores disponibles y contrastar, además de la capacidad de clasificación, criterios operativos de transparencia, control y reproducibilidad.

La comparativa se organiza en dos planos complementarios:

- **Plano metodológico:** nivel de transparencia del modelo, posibilidad de reproducir el proceso, acceso a configuración/umbral, y trazabilidad de la decisión.
- **Plano empírico:** evaluación con el mismo subconjunto de prueba del corpus, registrando la etiqueta o puntaje devuelto por cada herramienta y calculando métricas equivalentes (Accuracy, Precision, Recall, F1 y ROC-AUC) para una comparación justa.

Como insumo adicional para esta comparativa, se incorporaron cinco correos electrónicos formales generados por diferentes IAs (ChatGPT, Gemini, Claude, Copilot y DeepSeek). Estos correos se integraron porque representan el tipo de documentos formales presentes en el corpus y permiten verificar si las herramientas comerciales y el CNN reaccionan de forma consistente ante textos con estructura discursiva similar.

Con esta estructura, la comparación integra el plano metodológico y el plano empírico. El CNN destaca por su control y trazabilidad, mientras que las herramientas comerciales aportan una referencia práctica de uso cotidiano. Al aplicar las mismas métricas y el mismo corpus de evaluación, la comparativa permite

Tabla 4.15: Herramientas comerciales consideradas para la comparativa.

Herramienta	**Uso en la comparativa**
ZeroGPT	Servicio externo; se registra la etiqueta/puntaje devuelto sobre el mismo conjunto de prueba.
GPTZero	Servicio externo; se registra la etiqueta/puntaje devuelto sobre el mismo conjunto de prueba.
Justdone	Servicio externo; se registra la etiqueta/puntaje devuelto sobre el mismo conjunto de prueba.
WinstonAI	Servicio externo; se registra la etiqueta/puntaje devuelto sobre el mismo conjunto de prueba.

estimar cuánto se gana en transparencia y ajuste fino cuando se dispone de un modelo propio, y si los servicios comerciales ofrecen o no un desempeño competitivo frente a la CNN, como se muestra en la tabla 4.15.

Tabla 4.16: Criterios de comparación entre el modelo CNN y herramientas comerciales.

Criterio	**Relevancia para la comparativa**
Reproducibilidad	El CNN permite replicar el pipeline completo (datos, preprocesamiento y modelo), mientras que las herramientas comerciales se evalúan como cajas negras.
Control de umbral	En el CNN se define un umbral operativo explícito; en servicios comerciales se usa el umbral incorporado en la herramienta.
Calibración	El CNN reporta curvas de calibración y Brier; en herramientas comerciales se analiza la coherencia de su puntaje con las frecuencias observadas.
Transparencia	El CNN ofrece parámetros y decisiones auditables; los servicios externos se contrastan por su salida observable.
Privacidad y trazabilidad	El CNN puede ejecutarse localmente; los servicios externos implican envío de texto y operación en línea.

4.4.3 Caso práctico (E-mails) y ensayo corto

Para dejar un referente visible en el documento, se consideran cartas en español y en inglés del tipo académico-formal presentes en el corpus, lo que permite contrastar el comportamiento del CNN y de los detectores comerciales sobre un mismo estilo discursivo. El prompt de la Figura 4.8 se definió para estandarizar longitud, tono y estructura del correo, y para controlar variables como el rol del remitente y el contexto del destinatario. Esto permite que las muestras generadas sean comparables entre sí y representen escenarios realistas de comunicación académica.

Prompt utilizado: "Escribe un correo electrónico formal en inglés (convenciones de EE. UU.) para {{objective}}; escenario: {{job_application | academic_outreach | clarification_request}}; destinatario: {{name, role, org, city/country}}; remitente: {{name, title/role, phone, email, city/country}}. Incluye una línea de asunto; estructura = saludo ->propósito en 1 oración ->1-2 párrafos breves y específicos ->llamado a la acción claro con el siguiente paso/plazo propuesto ->cierre cortés; tono profesional y conciso; 120-200 palabras; completa los detalles faltantes de forma plausible (sin datos sensibles ni afirmaciones no verificables); gramática correcta; termina con firma completa; elimina todos los placeholders del texto final.".

Figura 4.8: Prompt utilizado para la generación de correos.

Se probó con 1 carta en español y 5 en inglés. En el cuerpo se incluye la carta en español y un ejemplo en inglés; las otras cuatro cartas en inglés (ChatGPT, Claude, Copilot y DeepSeek) se presentan en el Apéndice B.

Exploración en español

```
Prácticas profesionales
Buen día, mi nombre es Juan Pérez Mariscal, soy estudiante de octavo
↪ semestre de la carrera de ciencias empresariales en la universidad
↪ tecnológica de la Mixteca. He investigado sobre la empresa huevos San
↪ Juan y me llamó bastante la atención para realizar ahí mis prácticas
↪ profesionales.
Ayudaba a mi papá en su negocio de compra y venta de huevos así que tengo un
↪ poco de experiencia en eso, por lo cual me gustaría aplicar para las
↪ áreas de mercadotecnia o ventas.
Las fechas de las prácticas son del 1ro de julio al 22 de agosto.
Esperando contar con su respuesta les deseo un buen día.
Atentamente,
Juan Pérez Mariscal
```

Comparativa de detectores (email escrito por humano)

La Tabla 4.17 presenta la salida de los detectores para un email humano, con el fin de contrastar el comportamiento de cada herramienta frente a un texto auténtico.

Tabla 4.17: Resultados de detección para la email humana.

Herramienta	**Resultado**	**Observaciones**
CNN (modelo propuesto)	Probable humano	Probabilidad de IA **25.47 %** con umbral 0.709 (config. 0.70).
GPTZero	Mixto	"Lightly edited by AI": 2 % IA, 98 % mixto, 0 % humano.
ZeroGPT	78.24 % IA/GPT	Reporta señales mixtas con parte del texto generado.
Justdone	73 % IA	Indica 73 % de contenido generado por IA.
WinstonAI	2 % humano	Human Score 2 %, clasificado como texto de IA.

Email en inglés (Gemini)

IA generadora: Gemini. Correo académico breve dirigido a un profesor, con solicitud de aclaración sobre requisitos de proyecto.

```
Subject: Question about the Final Project - CS101 Introduction to
↪ Programming

Dear Professor Smith,

My name is Alejandro Reyes, and I am a student in your Introduction to
↪ Programming class (Section B) on Tuesdays and Thursdays at 10:00 AM.

I am writing to respectfully ask for a clarification regarding the final
↪ project requirements. I have reviewed the syllabus and the project
↪ description document on the university portal, and I had a specific
↪ question about the scope of the project. The description mentions that
↪ we should create a "simple inventory management application." I would
↪ like to know if an application that tracks up to 50 items with basic
↪ add, remove, and search functionalities would be considered sufficient,
↪ or if a more complex system with features like sales tracking is
↪ expected.

I am very excited to start working on this project and want to ensure I am
↪ on the right track from the beginning.

Thank you for your time and guidance. I look forward to your response.

Sincerely,

Alejandro Reyes

Student ID: 12345678

Computer Science Major
```

Comparativa de detectores (Gemini)

La Tabla 4.18 resume el análisis del email generado por Gemini en cuatro detectores comerciales. WinstonAI requiere suscripción y otorga 2000 créditos por plan; en un solo intento se descontaron 171 créditos.

Tabla 4.18: Resultados de detección para la email de Gemini.

Herramienta	**Resultado**	**Observaciones**
CNN (modelo propuesto)	Probable IA	Probabilidad de IA **99.91 %** con umbral 0.709 (config. 0.70).
GPTZero	100 % IA	Alta confianza: texto generado por IA (0 % mixto, 0 % humano).
ZeroGPT	100 % IA/GPT	Clasificado como texto generado por IA/GPT.
Justdone	84 % IA	Indica 84 % de contenido generado por IA.
WinstonAI	1 % humano	Human Score 1 %; un intento consumió 171 de 2000 créditos.

Ensayo corto (humano)

Autoría: humana (2010). Texto narrativo breve tomado de `https://journals.openedition.org/jsse/1083`. Se incluye para contrastar el comportamiento de los detectores con un texto humano no epistolar.

```
It is interesting to observe that, quite recently, Anne Tyler, another
↪ famous contemporary writer who was launched at the start of her career
↪ as the epitome of the scribbling housewife (even though she resented
↪ being referred to as such), used more or less the same words when
↪ talking about her attitude towards writing. After considering that she
↪ had ‘‘no secret hobbies or extra-curricular activities at all,’’ she
↪ concluded that this happened because she was ‘‘too busy daydreaming’’
↪ (quoted in Allardice). Much earlier, when she was still ‘‘an author 8:05
↪ to 3:30,’’ she had stressed the same point using Munro’s very same image:

It seems to me often that I am sort of looking from a window at something at
↪ a great distance and wondering what it is. But I’m not willing to
↪ actually go into it. I would rather sit behind the windowsill and write
↪ about it. So all my curiosity has to be answered within myself instead
↪ of by crossing the street and asking what’s going on. (quoted in
↪ Michaels)
```

Comparativa de detectores (ensayo corto humano)

La Tabla 4.19 resume el análisis del ensayo corto humano en los detectores comerciales y en el modelo CNN, para contrastar su respuesta ante un texto narrativo que no es un correo.

Tabla 4.19: Resultados de detección para el ensayo corto humano.

Herramienta	**Resultado**	**Observaciones**
CNN (modelo propuesto)	Probable humano	Probabilidad de IA **1.89 %** con umbral 0.709 (config. 0.70).
GPTZero	100 % humano	0 % IA generado, 0 % mixto, 100 % humano.
ZeroGPT	0 % IA/GPT	Clasificado como texto escrito por un humano.
Justdone	70 % IA	Reporta 70 % de contenido generado por IA.
WinstonAI	98 % humano	Human Score 98 %.

5. Conclusiones

Contenidos del Capítulo

La detección de textos académicos generados por intelencia artificial mediante modelos de predicción puede ser mejorada si se realiza una adaptación específica al contexto de acuerdo con los resultados mostrados en el capítulo anterior. La evidencia presentada confirma que un diseño metodológico orientado a reproducibilidad, control de sesgos y evaluación robusta permite alcanzar un desempeño competitivo y consistente en el dominio objetivo.

El objetivo general del estudio fue desarrollar un modelo para la detección de texto generado por inteligencia artificial que supere la precisión de las herramientas actuales, con enfoque primario en textos académicos en inglés y evaluación exploratoria en español, alineando su evaluación con métricas y procedimientos robustos e integrando señales estilométricas con representaciones semánticas basadas en Transformers. Este objetivo se cumplió al finalizar los experimentos y reportar las métricas obtenidas (ROC-AUC, Accuracy y F1-score) en las comparativas de los modelos clásicos y neuronales, evidenciando un desempeño superior frente a líneas base y detectores comerciales, con resultados estables y criterios operativos claros para su uso en contextos educativos.

Los resultados respaldan la viabilidad del modelo propuesto y su aporte a la evaluación académica responsable. La evidencia sugiere que con un diseño metodológico cuidadoso, es posible ofrecer herramientas de apoyo a la decisión con criterios de transparencia, trazabilidad y control de riesgos. En este sentido, el trabajo contribuye a fortalecer la integridad académica y a orientar el uso de Inteligencia Artificial (IA) desde una perspectiva técnica y ética.

En consecuencia, **no se rechaza la hipótesis**. Esta conclusión se sustenta en el comportamiento consistente del modelo propuesto, en la comparativa con herramientas externas y en la consistencia de las métricas reportadas bajo los protocolos definidos.

5.1 Aportaciones

Las aportaciones del trabajo se articulan tanto en el plano técnico como en el plano metodológico y educativo. En conjunto, fortalecen la evidencia de que es posible desarrollar detectores más confiables considerando a la vez criterios de robustez, transparencia y trazabilidad.

- Un pipeline reproducible de extremo a extremo (datos → rasgos → modelos → métricas), con trazabilidad de decisiones y artefactos, que facilita auditorías técnicas y comparabilidad entre estudios.
- Un enfoque híbrido que combina estilometría y representaciones semánticas basadas en Transformers, permitiendo capturar señales complementarias de forma y contenido para mejorar la separabilidad entre clases.
- Un esquema de evaluación robusto, con *área bajo la curva de precisión-recuperación* (AUPRC) como métrica primaria, calibración probabilística y pruebas de robustez ante paráfrasis y traducción, alineado con el control de falsos positivos en contextos educativos.
- Una evaluación exploratoria en español y una comparativa con herramien-

tas comerciales, que aporta evidencia contextualizada sobre transferencia multilingüe, desempeño operativo y limitaciones de los detectores externos.

5.2 Investigación futura

Como línea inmediata se propone ampliar la validación multilingüe y fuera de dominio, incorporando nuevos conjuntos con diversidad lingüística, variación de géneros y modelos generativos más recientes. También se recomienda profundizar en la calibración y explicabilidad del modelo, con el fin de interpretar mejor sus decisiones y reducir el riesgo de errores en escenarios sensibles. Estas extensiones permitirán presentar resultados más robustos, evaluar estabilidad ante deriva del generador y favorecer una adopción responsable en contextos educativos.

Bibliografía

[Abdelnabi and Fritz, 2021] Abdelnabi, S. and Fritz, M. (2021). Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP)*, pages 121–140.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Brier, 1950] Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3.

[Chang, 2025] Chang, L.-j. A. (2025). Detecting ai-generated text: A comparative study of machine learning algorithms.

[Chen and Guestrin, 2016] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

[Clark et al., 2021] Clark, E., August, T., Serrano, S., Haduong, N., Gururangan, S., and Smith, N. A. (2021). All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 7282–7296.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

[Davis and Goadrich, 2006] Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240.

[Dugan et al., 2024] Dugan, L., Hwang, A., Trhlík, F., Zhu, A., Ludan, J. M., Xu, H., Ippolito, D., and Callison-Burch, C. (2024). RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

[Dugan et al., 2020] Dugan, L., Ippolito, D., Kirubarajan, A., and Callison-Burch, C. (2020). RoFT: A tool for evaluating human detection of machine-generated text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 189–196. Association for Computational Linguistics.

[Dugan et al., 2023] Dugan, L., Ippolito, D., Kirubarajan, A., Shi, S., and Callison-Burch, C. (2023). Real or fake text? investigating human ability to detect boundaries between human-written and machine-generated text. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 12763–12771.

[Gehrmann et al., 2019] Gehrmann, S., Strobelt, H., and Rush, A. (2019). GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[Guo et al., 2017] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1321–1330.

[Ippolito et al., 2020] Ippolito, D., Duckworth, D., Callison-Burch, C., and Eck, D. (2020). Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

[Jakesch et al., 2023] Jakesch, M., Hancock, J. T., and Naaman, M. (2023). Human heuristics for ai-generated language are flawed. *Proceedings of the National Academy of Sciences*, 120(11):e2208839120.

[Kehkashan et al., 2025] Kehkashan, T., Riaz, R. A., Al-Shamayleh, A. S., Akhunzada, A., Ali, N., Hamza, M., and Akbar, F. (2025). Ai-generated text detection: A comprehensive review of methods, datasets, and applications. *Computer Science Review*, 58:100793.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

[Kirchenbauer et al., 2023] Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. (2023). A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

[Kohavi, 1995] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1137–1143.

[Koppel et al., 2009] Koppel, M., Schler, J., and Argamon, S. (2009). Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1):9–26.

[Lin et al., 2007] Lin, H.-T., Lin, C.-J., and Weng, R. C. (2007). A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

[Miralles-González et al., 2026] Miralles-González, P., Huertas-Tato, J., Martín, A., and Camacho, D. (2026). Not all tokens are created equal: Perplexity attention weighted networks for ai-generated text detection. *Information Fusion*, 125:103465.

[Mitchell et al., 2023] Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., and Finn, C. (2023). DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

[Niculescu-Mizil and Caruana, 2005] Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 625–632.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

[Prechelt, 1997] Prechelt, L. (1997). Early stopping – but when? In *Neural Networks: Tricks of the Trade*, pages 55–69. Springer.

[Rao et al., 2025] Rao, V. S., Kumar, A., Lakkaraju, H., and Shah, N. B. (2025). Detecting llm-generated peer reviews. *PLoS One*, 20(9):e0331871.

[Rodriguez et al., 2022] Rodriguez, J. D., Hay, T., Gros, D., Shamsi, Z., and Srinivasan, R. (2022). Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1213–1233, Seattle, United States. Association for Computational Linguistics.

[Sadasivan et al., 2023] Sadasivan, V. et al. (2023). Can ai-generated text be reliably detected?

[Saito and Rehmsmeier, 2015] Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432.

[Scott and Nowak, 2005] Scott, C. and Nowak, R. (2005). A neyman-pearson approach to statistical learning. *IEEE Transactions on Information Theory*, 51(11):3806–3819.

[Simonyan et al., 2013] Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

[Stamatatos, 2009] Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

[Su et al., 2023] Su, J., Zhuo, T., Wang, D., and Nakov, P. (2023). DetectLLM:

Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12395–12412, Singapore. Association for Computational Linguistics.

[Sun and Lv, 2025] Sun, J. and Lv, Z. (2025). Zero-shot detection of llm-generated text via text reorder. *Neurocomputing*, 631:129829.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

[Weber-Wulff et al., 2023] Weber-Wulff, D. et al. (2023). Testing of detection tools for ai-generated text. Reporte tecnico.

[Wu et al., 2025] Wu, J., Yang, S., Zhan, R., Yuan, Y., Wong, D. F., and Chao, L. S. (2025). A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

[Yang et al., 2024] Yang, X., Pan, L., Zhao, X., Chen, H., Petzold, L. R., Wang, W. Y., and Cheng, W. (2024). A survey on detection of llms-generated content. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9786–9805, Miami, Florida, USA. Association for Computational Linguistics.

[Zellers et al., 2019] Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., and Choi, Y. (2019). Defending against neural fake news. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pages 9051–9062.

[Zhang et al., 2015] Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.

[Zhao et al., 2024] Zhao, X., Ananth, P. V., Li, L., and Wang, Y.-X. (2024). Provable robust watermarking for ai-generated text. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria.

A. Competencia Kaggle LLM – Detect AI Generated Text

La competencia *LLM – Detect AI Generated Text* fue una *code competition* en Kaggle, activa del 31 de octubre de 2023 al 22 de enero de 2024. Participaron 4,358 equipos y la métrica oficial de evaluación fue ROC-AUC, apropiada para el problema binario de autoría humano vs. IA.[1]

El desafío fue organizado por la Universidad Vanderbilt y The Learning Agency Lab, en colaboración con Kaggle, como parte del proyecto AIDE (AI Detection for Essays). El conjunto de datos (AIDE) contuvo alrededor de 10,000 ensayos educativos distribuidos en siete *prompts*. El entrenamiento se construyó con ensayos de sólo dos *prompts*, mientras que los otros cinco se reservaron para el *test* oculto, lo que introdujo un *domain shift* por *prompt*. Además, la clase IA en entrenamiento fue deliberadamente escasa, por lo que se alentó a los equipos a generar ejemplos sintéticos con distintos LLM para balancear la clase positiva. El dataset y sus descripciones pueden consultarse en Kaggle.[2]

En la tabla privada final, el primer lugar alcanzó aproximadamente 0.961 de AUC, el segundo alrededor de 0.953 y el tercero en el rango de 0.945–0.95. Varios equipos superaron 0.95 de AUC, con diferencias muy pequeñas entre los primeros puestos. Estos resultados sugieren alta separabilidad bajo el protocolo

[1] `https://www.kaggle.com/competitions/llm-detect-ai-generated-text`

[2] `https://www.kaggle.com/datasets/sunilthite/llm-detect-ai-generated-text-dataset`

del reto, aunque no garantizan el mismo rendimiento en escenarios adversarios o con textos ofuscados.

Las soluciones ganadoras convergieron en el ajuste fino de Transformers y en *ensembles*. El primer lugar combinó modelos ajustados con QLoRA, un clasificador DeBERTa-v3 con *ranking loss* y un módulo de *embeddings* con kNN para recuperar vecinos similares. El segundo lugar priorizó *ensembles* de modelos de lenguaje y generación amplia de datos sintéticos, complementando con rasgos estilísticos. El tercer lugar reportó un enfoque híbrido similar con Transformers preentrenados y datos generados. La comunidad destacó que las señales explotadas podrían debilitarse si los textos de IA se reescriben para evadir detección.

B. Apéndice B: E-mails en inglés

A continuación se presentan los correos en inglés generados por distintas IAs, utilizados como ejemplos para la comparativa de detectores.

Email 1: ChatGPT

IA generadora: ChatGPT. Correo formal de postulación profesional con estructura completa (encabezado, cuerpo, cierre y firma), usado como ejemplo de texto formal extenso.

```
123 Oakwood Lane
Monterrey, NL 64000
Mexico

November 8, 2025

Hiring Manager
Innovative Tech Solutions, Inc.
457 Market Street, Suite 900
San Francisco, CA 94103
United States

Subject: Application for Software Engineer Position
```

```
Dear Hiring Manager,

I am writing to express my interest in the Software Engineer position at
↪ Innovative Tech Solutions, Inc., as advertised on your careers page.
↪ With a strong background in computer science, experience building
↪ scalable web applications, and a sustained interest in solving complex
↪ real-world problems through clean and maintainable code, I am confident
↪ that I can contribute meaningfully to your development team.

Over the past several years, I have worked on full-stack projects using
↪ technologies such as TypeScript, Node.js, React, and relational as well
↪ as NoSQL databases. In my most recent role, I participated in the design
↪ and implementation of a microservices-based architecture that improved
↪ system reliability and reduced average response times by more than 30%.
↪ I am particularly focused on writing testable code, maintaining clear
↪ documentation, and following best practices in version control and
↪ continuous integration.

Beyond my technical skills, I value collaboration, clear communication, and
↪ continuous learning. I have contributed to code reviews, mentored junior
↪ developers, and participated in architecture discussions to ensure
↪ alignment between technical decisions and business objectives. I enjoy
↪ working in multidisciplinary teams where product managers, designers and
↪ engineers collaborate closely to deliver features that genuinely improve
↪ the user experience.

I am particularly drawn to Innovative Tech Solutions, Inc. because of your
↪ emphasis on innovation, ethical use of technology, and long-term product
↪ quality. Your recent initiatives in data-driven decision-making and
↪ cloud-native development align closely with my professional interests
↪ and the direction in which I wish to grow. I am enthusiastic about the
↪ opportunity to bring my experience in backend development, API design,
↪ and performance optimization to your organization.
```

```
Thank you for considering my application. I would welcome the opportunity to
↪ discuss in more detail how my skills and experience can support your
↪ current and future projects. Please feel free to contact me at +52 (81)
↪ 5555-1234 or via email at
↪ [oms.candidate@example.com](mailto:oms.candidate@example.com).

Sincerely,

Oms Candidate
Software Engineer
[oms.candidate@example.com](mailto:oms.candidate@example.com)
+52 (81) 5555-1234
```

Email 2: Claude

IA generadora: Claude. Email de intención para un grupo de investigación, con tono formal y referencias a trayectoria académica.

```
Sarah Mitchell
123 Campus Drive, Apt 4B
Boston, MA 02115
sarah.mitchell@email.edu
(555) 123-4567
October 23, 2025
Dr. Robert Chen
Professor of Computer Science
Department of Computer Science
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
Dear Professor Chen,
I hope this letter finds you well. My name is Sarah Mitchell, and I am a
 ↪ junior in the Computer Science program at MIT. I am writing to express
 ↪ my strong interest in joining your research group that focuses on
 ↪ artificial intelligence and machine learning applications in healthcare.
I have been following your recent publications on neural networks for
 ↪ medical image analysis, particularly your paper on early detection of
 ↪ cardiovascular diseases. Your work aligns perfectly with my academic
 ↪ interests and career goals. Last semester, I completed your course CS
 ↪ 6.867 Machine Learning with a grade of A, and I found the material both
 ↪ challenging and deeply engaging.
I would greatly appreciate the opportunity to discuss potential research
 ↪ opportunities in your lab for the upcoming spring semester. I have
 ↪ attached my resume and transcript for your review. I am particularly
 ↪ interested in contributing to your current project on predictive models
 ↪ for patient outcomes.
Would you be available for a brief meeting during your office hours, or at
 ↪ another time that works better for your schedule? I am flexible and
 ↪ happy to work around your availability.
```

```
Thank you very much for considering my request. I look forward to hearing
↪ from you.
Sincerely,
Sarah Mitchell
```

Email 3: Copilot

IA generadora: Copilot. Correo formal con solicitud académica, utilizado como ejemplo adicional de estilo profesional.

```
Subject: Application for Research Analyst Position

Dear Dr. Anderson,

I am writing to formally apply for the Research Analyst position at the
↪ Global Policy Institute in Washington, D.C.

With a Master's degree in Economics from Georgetown University and three
↪ years of experience conducting quantitative and qualitative research, I
↪ have developed strong analytical and communication skills that align
↪ well with the requirements of this role. My recent work at the Center
↪ for Urban Studies involved preparing policy briefs and statistical
↪ reports that informed decision-making at both municipal and state
↪ levels. I am confident that my background in data analysis and policy
↪ evaluation would allow me to contribute meaningfully to your team's
↪ ongoing projects.

I would appreciate the opportunity to discuss how my skills and experience
↪ can support the Institute's mission. Could we schedule a conversation
↪ within the next two weeks at your convenience? I am available for an
↪ interview either virtually or in person.

Thank you very much for your time and consideration.

Sincerely,
Michael Torres
Research Associate
Phone: +1 (202) 555-0147
Email: michael.torres@email.com
Washington, D.C., USA
```

Email 4: DeepSeek

IA generadora: DeepSeek. Correo formal en inglés con enfoque académico, incluido para ampliar la comparativa.

```
Subject: Clarification Request Regarding Upcoming Research Symposium
↪ Submission

Dear Dr. Elena Rodriguez,

I am writing to request clarification on the submission guidelines for the
↪ 2024 Global Linguistics Research Symposium. My name is David Chen, a
↪ postdoctoral researcher at the University of Toronto, Canada, and I am
↪ finalizing a paper for the conference's deadline on June 15.

While reviewing the submission portal, I noticed that the template for
↪ extended abstracts specifies a 500-word limit, but the general
↪ guidelines mention a 750-word maximum. To ensure my submission adheres
↪ to the correct format, could you please confirm which word count is
↪ applicable for the linguistics track? Additionally, I would appreciate
↪ confirmation on whether references are included in the word count.

I have attached my draft abstract for reference. Could you kindly provide
↪ clarification by Friday, May 24, so I can make any necessary adjustments
↪ ahead of the deadline? Please let me know if you require any further
↪ information from me.

Thank you for your time and assistance.

Best regards,
David Chen, Ph.D.
Postdoctoral Researcher, Department of Linguistics
University of Toronto
Toronto, Canada
Phone: +1 (416) 555-0198
Email: dchen@utoronto.ca
```