Método basado en aprendizaje automático para la calificación de ensayos cortos en inglés de una muestra de estudiantes de bachillerato

Joan Gabriel Bofill Barrera

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia
2024

Método basado en aprendizaje automático para la calificación de ensayos cortos en inglés de una muestra de estudiantes de bachillerato

Joan Gabriel Bofill Barrera

Trabajo final presentado como requisito parcial para optar al título de:
Magíster en Ingeniería de Sistemas y Computación

Director:
Luis Fernando Niño Vázquez, Ph.D.

Línea de Investigación:
Sistemas Inteligentes
Grupo de Investigación:
Laboratorio de Investigación en Sistemas Inteligentes - LISI

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia
2024

A mi madre Sulma, a mi novia Sara, a todas las personas que me han ayudado en el camino y a mi querida *alma mater*...

"Hace 20 años, todo esto de la IA era ciencia ficción. Hace 10 años, era un sueño. Hoy lo estamos viviendo".

Jensen Huang, CEO y cofundador NVIDIA.

Declaración de obra original

Yo declaro lo siguiente:

He leído el Acuerdo 035 de 2003 del Consejo Académico de la Universidad Nacional. "Reglamento sobre propiedad intelectual" y la Normatividad Nacional relacionada al respeto de los derechos de autor. Esta disertación representa mi trabajo original, excepto donde he reconocido las ideas, las palabras, o materiales de otros autores.

Cuando se han presentado ideas o palabras de otros autores en esta disertación, he realizado su respectivo reconocimiento aplicando correctamente los esquemas de citas y referencias bibliográficas en el estilo requerido.

He obtenido el permiso del autor o editor para incluir cualquier material con derechos de autor (por ejemplo, tablas, figuras, instrumentos de encuesta o grandes porciones de texto).

Por último, he sometido esta disertación a la herramienta de integridad académica, definida por la universidad.

Joan Gabriel Bofill Barrera
Fecha: 26/01/2024

Agradecimientos

Quisiera agradecer en primera instancia a mi alma mater, la Universidad Nacional de Colombia, por brindarme la oportunidad de aprender de filosofía, de política, de ciencia y sobre todo de la vida, permitiéndome también conocer personas increíbles mientras cursaba mis estudios de pregrado y posgrado.

Asimismo, quiero agradecer especialmente al profesor Luis Fernando Niño, el director del presente trabajo y profesor de varias asignaturas que cursé, por sus valiosos consejos y su diligencia, así como todos los conocimientos y habilidades prácticas que me brindaron sus clases a lo largo de estos años de maestría. Además, quiero agradecer a todos los miembros del grupo de investigación LISI cuyas observaciones en los seminarios y cuyas investigaciones también contribuyeron bastante a la ideación y elaboración de este trabajo.

También, quiero expresar mi gratitud hacia mi mamá Sulma Barrera y mi novia Sara Reina y a nuestras familias por su compañía y apoyo en momentos difíciles, así como su confianza en mí y su ayuda en todos los desafíos que se me han presentado de los cuales el presente trabajo final de maestría no es la excepción.

De la misma manera, extiendo mi agradecimiento a todos mis amigos, mis compañeros de trabajo y a todos aquellos que han aportado un granito de arena en mi vida y en particular en mi manera de ver el mundo, este es un trabajo o un logro de todos nosotros. Finalmente, no puedo evitar agradecer al cosmos por conspirar para que estemos aquí hoy, tratando de entenderlo poco a poco y de disfrutar el camino.

Resumen

Método basado en aprendizaje automático para la calificación de ensayos cortos en inglés de una muestra de estudiantes de bachillerato.

Este trabajo aborda el desafío de la calificación automática de ensayos argumentativos en inglés escritos por estudiantes de bachillerato que están aprendiendo el inglés como segunda lengua. El objetivo general es implementar un método automático basado en aprendizaje supervisado que permita resolver esta tarea para 6 indicadores en simultáneo: *Cohesión, Sintaxis, Vocabulario, Gramática, Fraseología y Convenciones* en escala de 1 a 5. Para lograrlo, se realiza un análisis descriptivo de los datos, se aplican procedimientos de preprocesamiento y se extraen características relevantes; se exploran diferentes estrategias, técnicas de representación y modelos desde algunos clásicos hasta aquellos con mejor desempeño en la actualidad, evaluando en cada iteración su rendimiento, contrastándola con las calificaciones humanas. Luego, se presenta el modelo con menor error que está basado principalmente en DeBERTa al cual se le aplican distintas técnicas para mejorar su desempeño y se combina con un modelo SVR que toma como características los *embeddings* de los textos concatenados en 10 modelos preentrenados sin *fine-tuning*. Con esta estrategia, el resultado se acerca bastante a las calificaciones humanas, presentando un RMSE de 0.45 sobre todos los indicadores.

Palabras clave: PLN, calificación automática de ensayos, aprendizaje supervisado, transformers, ensamble de modelos, SVR, concurso Kaggle.

Abstract

Machine learning based method for scoring short english essays from a high school student sample.

This work addresses the challenge of automatically grading argumentative essays in English written by high school students that learn English as a second language. The general objective is to implement an automatic method based on supervised learning that allows solving this task for 6 indicators simultaneously: *Cohesion, Syntax, Vocabulary, Grammar, Phraseology and Conventions* rated on a scale from 1 to 5. To achieve this, a descriptive analysis of the data is conducted, preprocessing procedures are applied and relevant features are extracted; different strategies, representation techniques and models are explored, from some classic ones to the currently best performing models. Their performance is evaluated in each iteration, contrasting it with human ratings with a chosen measure. Then, the method with the best performance is presented, it is based mainly on DeBERTa V3 Large, where different techniques are applied to improve its performance. Finally, and is combined with a regressor model SVR that takes as features the concatenated embeddings of the texts in 10 different pretrained models. With this strategy, the result is quite close to human ratings, presenting a root mean square error of 0.45 over all indicators.

Keywords: NLP, Automatic essay grading, Supervised learning, ensamble of models, SVR, Kaggle contest.

Este Trabajo Final de maestría fue calificado en abril de 2024 por la siguiente evaluadora:

Elizabeth León Guzmán, PhD.
Profesora Facultad de Ingeniería
Universidad Nacional de Colombia

Contenido

Lista de símbolos

Abreviaturas

Abreviatura	**Término en Español**
AES	Calificación Automática de Ensayos
AWP	Perturbaciones Adversarias en los Pesos
ANN	Red Neuronal Artificial
$BERT$	Representación de Codificador Bidireccional de Transformadores
BoW	Bolsa de palabras
CNN	Red Neuronal Convolucional
$DeBERTa$	BERT con codificador mejorado con atención separada.
$LSTM$	Redes neuronales recurrentes con memoria a corto-largo plazo.
QWK	Kappa cuadrático ponderado
ML	Aprendizaje Automático
NLP	Procesamiento del Lenguaje Natural
$RMSE$	Raíz cuadrada del error cuadrático medio
SVM	Support Vector Machine
SVR	Support Vector Regression

1. Introducción

Actualmente, la evaluación es un tema familiar para todos, la mayoría de las oportunidades de estudio o trabajo van de la mano con algún examen, sistema de clasificación o ranking [1]. Existen diversas maneras de evaluar al momento de realizar un examen, la más común es sin duda la pregunta de selección múltiple, aunque hay métodos más complejos y, en ocasiones, más dicientes como lo son los textos (ya sean ensayos, síntesis, cartas, etc.). Uno de los problemas de evaluar a través de textos es el costo que implica, pues se ha de conseguir uno o varios calificadores para poder determinar la habilidad del evaluado. Esto es un problema que enfrentan actualmente todas las instituciones educativas al evaluar las pruebas de escritura. Se puede ver a nivel local cuando un profesor de bachillerato quiere calificar varios ensayos a sus alumnos, pero esto le toma bastante tiempo y se expone a no ser equitativo, a un alto nivel de sesgo y a mayores tiempos de calificación. También se puede apreciar la complejidad de una calificación a nivel nacional como en la prueba de *Comunicación Escrita* en exámenes oficiales como el Saber Pro y Saber TyT que son obligatorios para todos los que aspiren a una formación técnica o profesional en Colombia y que por tanto incurren en costos altísimos para poder evaluar los textos.

Una de las herramientas más populares que han surgido del avance tecnológico son los métodos de *machine learning* o aprendizaje automático, por medio de los cuales se resuelven distintos tipos de problemas mediante algoritmos que se entrenan de manera iterativa basados en un conjunto de datos. Existen innumerables áreas de aplicación en las que dichos métodos resultan de utilidad, como la clasificación de imágenes, los pronósticos del clima, traducción o hasta creación de textos, imágenes y toda clase de contenido, una de estas áreas de trabajo en la actualidad es la clasificación de textos de la cual la calificación es un caso particular. En este contexto, una posible optimización del proceso de evaluación de escritos, el cual está presente en todos los niveles de formación académica convencional así como en los exámenes de formación técnica y profesional, sería la calificación automática de textos y es algo en lo cual se trabaja hoy en día con la ayuda del Procesamiento del lenguaje Natural (NLP, por sus siglas en inglés, *Natural Language Processing*) para la extracción de características y el entendimiento profundo del lenguaje, así como distintos tipos de técnicas de representación y con modelos supervisados que se entrenan con conjuntos de datos revisados por humanos. Un ejemplo de esto es el popular sistema ChatGPT que se ha vuelto uno de los temas más relevantes de la actualidad y que tiene mucho impacto en la escritura de ensayos [2]. Si bien es ambicioso y difícil imaginarse que en algún momento procedimientos automáticos reem-

placen totalmente a los calificadores, es probable que sí se logre parcialmente, de manera que dichos procedimientos contribuyan en el proceso de evaluación como sucede en la actualidad en algunos exámenes importantes en idioma inglés. Así, aplicar técnicas de procesamiento del lenguaje natural podría permitir determinar de antemano si un texto es impertinente o si no fue escrito en el formato argumentativo adecuado, así como clasificar textos según sus temas y registros, reduciendo el sesgo que un evaluador pueda tener. Además, este tema de investigación permitirá entender más el lenguaje escrito, sus criterios y la manera como se percibe, sin mencionar que es potencialmente aplicable a muchos otros ámbitos ajenos a la evaluación.

Entre los métodos computacionales presentes que podrían ayudar a mitigar las dificultades de calificar pruebas de pregunta abierta se destaca el área de calificación automatizada o automática de ensayos (AES, por sus siglas en inglés *Automatic Essay Scoring*), aunque el mismo concepto puede ser encontrado en la literatura de diversas maneras: *Automated Essay Grading* (AEG), *Automated Essay Evaluation* (AEE), *Automated Writing Evaluation* (AWE) or *Analytic Writing Assessment* (AWA). Estos términos se refieren al uso de algún modelo computacional o estadístico para asignar calificaciones a los ensayos en un entorno educativo. Estos desarrollos se utilizaron inicialmente para reducir el costo de la calificación manual de ensayos en los años 60 [3][4]. Aparte de la efectividad de los costos, AES se considera como un enfoque inherentemente más consistente y menos sesgado que la calificación por evaluadores humanos. Se pueden comparar los resultados de un método de calificación automática con aquellos proporcionados por los evaluadores humanos utilizando estadísticas de confiabilidad o concordancia entre evaluadores [5]. Recientemente, se desarrollaron modelos de calificación automática que obtuvieron una concordancia muy alta con las calificaciones de un conjunto de ensayos cortos en inglés, exhibiendo un rendimiento igual o hasta ligeramente superior al humano basado en un motor en el que los expertos determinaron cuidadosamente un conjunto de características que se debían extraer de los textos [6]. En general es un tema de investigación bastante mencionado últimamente en la literatura científica, como se puede apreciar en la Figura **1-1** proveniente de Scopus.

Considerando lo anterior, surge la idea de desarrollar y aplicar un método para la calificación automática de ensayos cortos que pueda ser empleado fácilmente para estimar distintos indicadores de los ensayos de una muestra de estudiantes de bachillerato, más específicamente estudiantes que se encuentran aprendiendo el inglés como segunda lengua y que tienen sus condiciones particulares que hace que sean poco comparables con el grupo de estudiantes de habla inglesa en todos los constructos relacionados con el idioma. Los puntajes obtenidos con dicho método deben ser similares a las calificaciones realizadas por evaluadores humanos, contribuyendo así a reducir el costo y sesgo de la calificación de estos ensayos como parte de la evaluación académica de un año lectivo o como parte de algún examen particular externo a la formación de bachillerato.

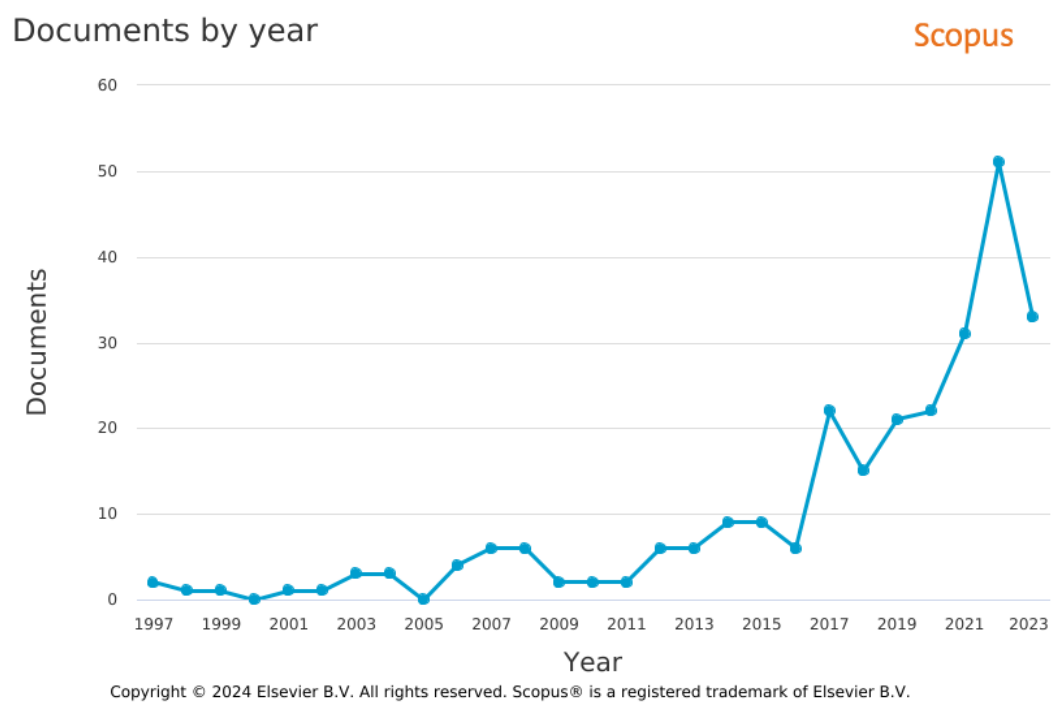

Figura 1-1.: Número de documentos por año para la ecuación de búsqueda "Automatic Essay Scoring" en Scopus.

En particular, en el presente trabajo se tomará como punto de partida el conjunto de datos de un concurso de Kaggle llamado *Feedback prize 3.0* [7], sobre el cual se llevarán a cabo las fases de preprocesamiento, extracción de características y se ajustarán distintos modelos, comparándolos con respecto a una métrica definida de desempeño. Se implementará la mejor estrategia ya sea un modelo o combinación de los mismos y se evaluará el desempeño de la estrategia seleccionada, reportando los hallazgos en todo este proceso de investigación y experimentación. Finalmente, se plantearán las conclusiones y recomendaciones, así como las posibles líneas de investigación a profundizar en trabajos futuros.

1.1. Planteamiento del Problema

Problema: En la actualidad no se utilizan ampliamente métodos para calificar de manera automática ensayos logrando predecir distintos indicadores de los textos de forma eficiente, controlando los costos de aplicación y el sesgo de los evaluadores.

Causas: No hay confianza en alternativas automáticas a la calificación tradicional de textos en exámenes oficiales [8]. Además, no se conoce tanto la estructura del lenguaje natural como para poder aplicar modelos automáticos que sean equivalentes al criterio de calificadores humanos [9]. No es de amplio conocimiento qué alternativas de modelos o metodologías existen

para esta tarea actualmente y cómo optimizarla según las circunstancias y las restricciones.

Efectos: Los profesores tienden a no solicitar tantos ensayos sobre todo en cursos de bastantes estudiantes pues resulta complicado calificarlos y más si se trata de varias notas en un mismo ensayo. Además, se dificulta la aplicación de exámenes que incluyan una prueba de producción de textos, pues suben los costos considerablemente [10]. Por tanto, normalmente se opta por otro tipo de preguntas como de selección múltiple y no se aprovecha el potencial psicométrico de la producción de textos por parte de los evaluados. Además, los tiempos de calificación aumentan considerablemente, debido a que es necesario que todos los calificadores lean todos los ensayos y los puntúen.

Pregunta de Investigación: ¿Cómo calificar un conjunto de ensayos de estudiantes de bachillerato usando métodos de aprendizaje automático supervisado?

1.2. Justificación

Actualmente es una necesidad en diversos contextos evaluar las capacidades de escribir un texto argumentativo sobre un tema determinado. Por ejemplo, cuando se cursa el bachillerato en varias asignaturas es una tarea recurrente tener que escribir ensayos sobre distintos temas. En estos casos, se evalúa el conocimiento del alumno en los temas de la asignatura y, además, se evalúan habilidades transversales de capacidad argumentativa, su cohesión, ortografía y estilo, entre otros aspectos. Existen además otras situaciones donde hay una necesidad de evaluar textos es en exámenes oficiales los cuales tienen una demanda constante pues son requisitos para ciertos trámites. Un ejemplo habitual de esto son los exámenes de conocimientos en lenguas extranjeras como el TOEFL en el caso de inglés, donde el componente de producción escrita (como una de las 4 habilidades de la lengua) tiene una sección separada, para su calificación la empresa ETS (Educational Testing Service) que ofrece el examen a nivel mundial está usando a día de hoy herramientas de calificación automática las cuales se ponderan con las calificaciones humanas [11], probando así el potencial de estos desarrollos tecnológicos. Otro ejemplo, de un contexto local, donde resultaría útil una metodología de calificación automática y donde se investiga activamente el tema es en el Instituto Colombiano para la Evaluación de la Educación (Icfes) más específicamente los exámenes de Estado de la educación Superior*Saber Pro* o *Saber TyT* donde se incluye una prueba de escritura y la evaluación de dichos textos implica contratar codificadores para que clasifiquen en 4 niveles cada ensayo y posteriormente usar esas codificaciones para calcular los puntajes de los evaluados con modelos de Teoría de Respuesta al Ítem, más específicamente el modelo de múltiples facetas de Rasch. Considerando la gran cantidad de textos argumentativos que se generan en las instituciones educativas de todos los niveles de formación, así como en cada examen oficial y el requerimiento de al menos dos codificadores humanos por cada texto para controlar los sesgos, los costos y tiempos debidos a las capacitaciones y la generación de

codificaciones se tornan bastante elevados. Además, existe la posibilidad de tener numerosas discrepancias o inconsistencias debidas al conocimiento variado de los calificadores o de sus propias subjetividades al evaluar textos.

Uno de los beneficios anticipados de aplicar una estrategia de calificación automática de ensayos es la eliminación de factores como la fatiga del evaluador, la experiencia del evaluador, la gravedad/indulgencia, la contracción de la escala, los estereotipos, el efecto halo, la deriva del evaluador, la diferencia de percepción y la inconsistencia [12]. Por lo anterior, la calificación automática facilita la evaluación de los ensayos a nivel económico y en ocasiones también logístico. Asimismo, los resultados de un examen podrían estar disponibles en un tiempo mucho más corto, pues solo dependen del tiempo de procesamiento del método de calificación que se seleccione. Por último, una calificación automática se podría ajustar para ciertas poblaciones que tengan características particulares beneficiando así la medición de dicha población, aunque se ha evidenciado que la mayoría de la investigación en calificación automática tiende a ser heterogénea en cuanto a ambientes del lenguaje, características geográficas o económicas de los autores o nivel formativo o del lenguaje [13] por lo que controlar uno de esos factores podría conducir a hallazgos valiosos en algún dominio particular.

Una población estudiantil en rápido crecimiento son los estudiantes que aprenden inglés como segundo idioma (conocidos como ELL, por sus siglas en inglés *English Language Learners*). Esta población se ve especialmente afectada por la falta de herramientas para calificar adecuadamente sus tareas de escritura. Si bien las herramientas de retroalimentación automatizada facilitan que los maestros asignen más tareas de escritura, no están diseñadas teniendo en cuenta a los ELLs. Es decir, las herramientas existentes no pueden proporcionar retroalimentación supeditadas a la competencia lingüística del estudiante, lo que resulta en una evaluación final que puede estar sesgada en contra del aprendiz. Esta puede ser una problemática que enfrenten todos los jóvenes que están aprendiendo inglés como segunda lengua y que se pueden ver desfavorecidos por las herramientas de calificación automática actuales, por lo tanto, conviene entrenar modelos y aplicar técnicas de ML en este tipo de ensayos. La ciencia de datos, la inteligencia artificial y el procesamiento del lenguaje natural son áreas que sin duda pueden mejorar las herramientas de retroalimentación automatizada para satisfacer mejor las necesidades únicas de estos estudiantes.

Por lo anterior, sería adecuado optimizar recursos y concentrar esfuerzos para tener más metodologías disponibles para estas tareas según cada situación. Si no se empieza a investigar en el área de calificación automática de ensayos, se desconocerá el desempeño de los modelos implicados en esas metodologías aplicados en distintos tipos de ensayos, al predecir distintos indicadores de un texto simultáneamente. Considerando lo amplio que es el contexto de los ensayos, resulta razonable empezar por ensayos más sencillos, como los de nivel de formación básica (bachillerato), antes de considerar calificar de manera automática ensayos de niveles

de profesionales o posgrados que pueden ser mucho más elaborados y más complicados en su contenido, estructura, vocabulario y demás factores. Por otro lado, si la evaluación de los ensayos no tiene por lo menos una componente automática, la medición quedaría expuesta en su totalidad a los posibles sesgos antes mencionados de los codificadores además de estar proclives a posibles errores. Asimismo, el hecho de construir sistemas de calificación automática de ensayos que arrojan resultados similares a los calificadores humanos ayuda a entender mejor el criterio con el que se perciben los ensayos y el lenguaje en general, lo que permitirá entender mejor cómo elaborar cada vez mejores ensayos y más adelante se podrán realizar automáticamente tareas que sólo hoy pueden hacer los humanos [14].

1.3. Objetivo General

Implementar un método basado en aprendizaje supervisado para la calificación automática de un conjunto de ensayos argumentativos en inglés de estudiantes de bachillerato.

1.4. Objetivos Específicos

- Determinar las características relevantes del conjunto de ensayos depurado para su calificación según la literatura a partir de los indicadores evaluados.
- Determinar la metodología para calcular cada uno de los indicadores basados en la literatura incluyendo procedimientos estadísticos complementarios y múltiples estrategias de aplicación de técnicas de aprendizaje automático.
- Implementar la metodología de calificación automática incluyendo los modelos de aprendizaje automático supervisados de acuerdo con las características seleccionadas y cada indicador que se quiere predecir.
- Evaluar la metodología obtenida sobre el conjunto de ensayos de prueba, contrastando los resultados de su implementación con respecto a las calificaciones otorgadas por los calificadores humanos.

2. Antecedentes

Puede parecer que el tema de la calificación automática es algo de última tecnología o verdaderamente moderno, pero en el lejano año de 1966 Ellis Batten Page propuso el primer sistema automático para evaluar ensayos de estudiantes llamado PEG, *Project Essay Grading* [4]. A pesar de su capacidad para predecir las calificaciones de los profesores, esta versión tuvo una limitada aceptación en la comunidad. La disponibilidad de herramientas necesarias como computadores, internet, técnicas computacionales para la extracción automática de medidas de calidad de escritura, entre otras, era escasa y la sociedad criticó la idea de desplazar a los calificadores humanos.

En 2013, Mayfield y Rosé publicaron el software *LightSIDE*, una máquina para evaluación automática de textos en inglés con código fuente disponible. Este programa se diseñó como una herramienta de fácil uso para propósitos de análisis de textos, incluyendo la evaluación de ensayos [15]. Posteriormente, tres sistemas comerciales fueron predominantes en el campo de evaluación automática de ensayos en inglés: el *E-rater*, el *Intelligent Essay Assessor (IEA)* y el *IntelliMetric*. Desde entonces, muchas empresas e investigadores han propuesto diversas herramientas de software para calificación automática de ensayos, algunos trabajando el ensayo como un todo, de manera holística y otros extrayendo distintos tipos de características [16].

Burrows et al. (2015) revisaron los sistemas AES y los describieron en seis dimensiones: conjunto de datos, técnicas de PNL, construcción de modelos, modelos de calificación, evaluación y efectividad del modelo [17]. En 2017, Zupanc y Bosnic proponen una extensión de los sistemas de evaluación automática de ensayos al incorporar atributos adicionales de coherencia y consistencia semántica [18]. De manera similar, varios autores han probado con distintas combinaciones de características y modelos para entrenar con ellas. El tema ha sido de tanto interés que hay libros dedicados a calificación automática [19]. En consecuencia, esta área constituye una de las más populares aplicaciones del procesamiento del lenguaje natural.

Existen distintos enfoques para abordar el tema de la calificación automática de ensayos, entre estos se encuentran modelos de regresión, modelos de clasificación y modelos de redes neuronales. Los modelos de regresión no logran encontrar cohesión y coherencia en el ensayo porque se entrenan con características de Bolsa de Palabras (BoW por su sigla en inglés). El modelo bolsa de palabras se utiliza en el procesado del lenguaje para representar documen-

tos ignorando el orden de las palabras y basándose solamente en su frecuencia de aparición. Al procesar datos desde la entrada hasta la salida, los modelos de regresión clásicos son más simples que las redes neuronales debido a que estas últimas usan funciones no lineales y tienen muchos más parámetros. Sin embargo, con estos no se pueden encontrar muchos patrones intrincados en el ensayo y no se puede encontrar la conectividad de las oraciones. En consecuencia, si se entrena un modelo de red neuronal con características de BoW, este no podrá considerar la coherencia y cohesión del ensayo.

También, existe la opción de entrenar los modelos directamente con los ensayos y no con características extraídas de los mismos, para esto se utilizan técnicas de representación. En primer lugar, para entrenar un algoritmo de aprendizaje automático con ensayos, todos los ensayos se convierten a formato vectorial numérico, para esto se dividen en unidades de interés llamadas tokens, ya sean palabras, oraciones, párrafos, etc. Se han elaborado numerosas técnicas para llevar las palabras o tokens a esta representación vectorial, entre las más famosas se encuentran GloVe, BoW y Word2vec.

La mayoría de los modelos aplicados en este campo son supervisados donde la variable de interés o *target* es una calificación realizada por expertos, de esta manera los modelos entrenan con los datos con la intención de poder descifrar patrones escondidos en ellos y generalizar la calificación a otros textos que no estaban inicialmente en la base de datos de entrenamiento. Casi todos los sistemas de calificación automática de textos se basan en evaluaciones de evaluadores expertos, aunque algunos autores utilizan escritos de expertos para desarrollar los modelos de evaluación de dichos sistemas [20].

Por otro lado, es posible usar enfoques no supervisados para ver patrones ocultos en los textos y determinar qué tan similares son entre sí [21]. Esto permite agrupar los textos de acuerdo con diversos criterios, los cuales pueden estar asociados a su desempeño en el área o constructo que se desee medir y, por lo tanto, los grupos pueden representar una calificación. También, han habido enfoques para calificación automática de textos basados en aprendizaje por refuerzo como el reportado en [22]. Además, existen técnicas que no solo pretenden calificar el texto del estudiante sino también darle retroalimentación de sus escritos, como se puede ver en [23] o en [24] donde los modelos proporcionan realimentación de algunos aspectos como gramática u ortografía.

Existen dos grandes enfoques de la calificación automática de textos. El primero consiste de una estructura ligada a características diseñadas manualmente, por ejemplo: la coincidencia de patrones, la longitud del texto, la dicción, organización, errores gramaticales y ortográficos, entre otras características ligadas al lenguaje. Este enfoque tiene como ventaja que permite ser interpretado y explicado; sin embargo, exige un gran esfuerzo para ajustar las características evaluadas y así obtener una buena precisión [25]. Las metodologías que se

usan implican ajustar similitudes con modelos estadísticos gaussianos y logísticos, y ajustes a partir de métodos clásicos como *Support Vector Machines* (SVM), regresión logística o naïve Bayes para la clasificación después del proceso de selección de características. Estas metodologías son dependientes de una gran cantidad de datos, lo cual es una desventaja crucial en este campo, dado que la cantidad de textos usualmente no es suficiente para estos modelos.

El otro gran enfoque está basado en redes neuronales profundas, cuya metodología permite, mediante capas ocultas, omitir la necesidad de la extracción manual de características textuales y así enfocarse en el estilo o contenido de los textos. Entre las arquitecturas de redes neuronales más empleadas en calificación automática de ensayos se encuentran los *Transformers* [26] y las redes neuronales recurrentes, específicamente, *Gated Recurrent Units* (GRU) y *Long Short-term Memory* (LSTM) [27]. La GRU es muy eficiente para conjuntos de datos pequeños y textos cortos. Por el contrario, LSTM es más usada en textos largos. Otras metodologías que han tenido gran impacto son las basadas en *transformers* que en los últimos años han aportado en la investigación en múltiples campos, un ejemplo de estas es *Bidirectional Encoder Representations from Transformers* (BERT), originalmente propuesto en [28]. Estas metodologías tienen como desventaja que su entrenamiento es bastante lento [29], por lo cual aún se investigan métodos de pre-entrenamiento para mejorarlas y volverlas más efectivas. En [30] se sugiere que no se deben emplear estas arquitecturas tan profundas debido a que no ofrecen un rendimiento significativamente mejor que los métodos clásicos y sí se incrementa bastante el costo computacional y la complejidad.

2.1. Conceptos Claves

- **Modelo de aprendizaje supervisado:** es un modelo de aprendizaje automático que dispone de un conjunto de datos de entrenamiento y además sus etiquetas (*targets*) también están disponibles. A lo largo de la revisión de literatura realizada se encontró que casi la totalidad de desarrollos en calificación automática son supervisados, dado que resulta de vital importancia poder comparar con las calificaciones de personas para ver qué tanto emula el modelo el criterio humano. Estas calificaciones jugarían el rol de etiquetas o de valores posibles de la variable de interés.

- **Extracción de características:** La extracción de características (*features*) se trata de un proceso de reducción y codificación, donde un conjunto inicial de variables sin procesar (ej. texto en un documento) se reduce a características con un procesamiento más sencillo (ej. números), que describen con precisión el conjunto de datos original. En el contexto de la investigación, estas pueden ser: i) estadísticas, cuando se refieren a la longitud del ensayo o al número de palabras promedio por frase; ii) de forma, cuando describen aspectos del texto como la ortografía o la gramática; o iii) semánticas, cuando se tratan del poder expresivo de las frases o la coherencia del texto en sí. A continuación,

se presenta una tabla de ejemplo con algunas características de interés:

Características Estadísticas	**Características por Estilo**	**Características por Contenido**
Longitud del ensayo con respecto al número de palabras	Estructura de la oración	Cohesión entre oraciones en un documento
Longitud del ensayo con respecto a oraciones	POS	*Overlapping (prompt)*
Longitud de oración promedio	Puntuación	Relevancia de la información
Promedio de longitud de palabra	Gramatical	Rol semántico de la palabra
N-grama	Operadores lógicos	Exactitud
	Vocabulario	Consistencia
		Oraciones con conceptos claves

Tabla 2-1.: Clasificación de las características de interés en un ensayo, adaptada de [31].

- **Quadratic Weighted Cohen's Kappa:** Es una de las métricas más usadas para comparar la similitud o concordancia entre puntajes en el contexto de calificación automática de ensayos [32]. Se aplica cuando la calificación es de tipo nominal, es decir, es una categoría a la que no se le puede asignar un grupo (*cluster*). Esta medida se usa para determinar qué tan cercanas son las predicciones de los modelos a las notas establecidas por los calificadores humanos. Resulta ser una métrica robusta del error, ya que supone como línea base la posibilidad de concordancia por azar. Oscila entre 0 y 1, aunque en algunos casos puede ser menor a 0 cuando la concordancia es menor que la concordancia al azar. Está dada por:

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2} \qquad \kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}$$

  donde w es una matriz cuadrada de dimensión k (siendo k el número de categorías) de ponderaciones y cada elemento $w_{i,j}$ se calcula basado en la diferencia entre las puntuaciones de los evaluadores como sugiere la primera ecuación. La matriz O se calcula con las frecuencias que cada evaluador da a cada categoría de puntaje, mientras que la matriz E corresponde a una matriz de concordancia en la que los evaluadores ponen la nota al azar, similar a si se lanzara una moneda. Esta medida es de utilidad también para comparar un evaluador humano en dos escenarios distintos o para medir la concordancia entre un evaluador humano y un modelo de calificación automática, lo cual resulta de interés para el proyecto.

- **Corpus disponibles:** En la literatura, se suele emplear un conjunto de datos en inglés llamado ASAP (Automated Student Assessment Prize's, `https://www.kaggle.com/c/asap-aes`) para evaluar los métodos de calificación automática de ensayos, postulado originalmente por Kaggle en el 2013 en el contexto de una competencia de ciencia de datos. Este conjunto cuenta con aproximadamente 13000 ensayos divididos en 8 subconjuntos los cuales tienen distintos temas y contextos. Cada ensayo tiene una longitud

aproximada de entre 150 y 550 palabras. Múltiples artículos han empleado el conjunto ASAP para probar modelos de calificación automática, como se puede ver en [33]. A continuación, se presenta una tabla que menciona varios corpus utilizados para AES.

Corpus	System	Scoring Task	Approach	Features										Evaluation Results		
				L	X	E	C	P	R	S	A	M	D	QWK	PCC	MAE
CLC-FCE	Yannakoudakis and Briscoe [2012]	Holistic	Ranking	✓			✓			✓		✓	✓	-	0.749	-
ASAP	Cozma et al. [2018] (In-domain)	Holistic	Regression			✓								0.785	-	-
	Cozma et al. [2018] (Cross-domain)	Holistic	Regression			✓								$1 \rightarrow 2 : 0{,}661$ $3 \rightarrow 4 : 0{,}779$ $5 \rightarrow 6 : 0{,}788$ $7 \rightarrow 8 : 0{,}649$	-	-
TOEFL11	Vajjala [2018]	Holistic	Regression	✓	✓			✓		✓			✓	-	0.800	0.400
ICLE	Wachsmuth et al. [2016]	Organization	Regression		✓		✓	✓		✓	✓	✓		-	-	0.315
	Persing and Ng [2013]	Thesis Clarity	Regression		✓		✓			✓		✓		-	-	0.483
	Persing and Ng [2014]	Prompt Adherence	Regression		✓		✓			✓		✓		- -	0.360	0.348
	Wachsmuth et al. [2016]	Persuasiveness	Regression		✓		✓	✓		✓	✓	✓		- -	-	0.378
AAE	Ke et al. [2018]	Persuasiveness	Regression (Neural)	✓		✓	✓		✓					-	0.236	1.035

Tabla 2-2.: Rendimiento de sistemas AES de última generación en corpus de evaluación de uso común. Las características se dividen en diez categorías: basadas en la longitud (L), léxicas (X), incrustaciones de palabras (E), basadas en categorías (C), relevantes para el aviso (P), legibilidad (R), sintácticas (S), argumentación (A), semántica (M) y discurso (D). Adaptada de [34].

Como se puede apreciar, la mayoría de los desarrollos de calificación automática se hacen para textos en formato digital. Sin embargo, hay algunas excepciones como [35], donde se propone un sistema de calificación automática de ensayos a mano, lo cual supone otros problemas como el de la digitalización de los manuscritos y aplicación de técnicas de reconocimiento de textos. En cuanto al idioma, la mayoría de literatura en el área está centrada en textos en inglés, pero hay algunas excepciones como [36] donde se emplea un enfoque multilingüe y se aplica al español y al vasco.

- **Validación Cruzada:** La validación cruzada es una técnica utilizada en *Machine Learning* para evaluar el rendimiento de un modelo de manera robusta. En este método el conjunto de entrenamiento se divide en varios subconjuntos, denominados *folds* en inglés o pliegues en español. El modelo se entrena en diferentes combinaciones de estos *folds*, lo que permite realizar múltiples predicciones.

  La implementación más común es la validación cruzada *k-fold* o con k pliegues, en la cual el conjunto de entrenamiento se divide en k folds. Estos pueden ser obtenidos de manera aleatoria sobre todo el conjunto en caso de que las categorías de la variable de interés estén balanceadas, o de manera estratificada sobre una o varias variables de interés para que cada *fold* sea representativo del comportamiento de las variables. Así,

el modelo se entrena k veces, utilizando cada *fold* como conjunto de prueba en una iteración y el resto como conjunto de entrenamiento. Posteriormente, se aplican los k modelos entrenados al conjunto de prueba original y se promedian sus predicciones. Este enfoque proporciona una evaluación más completa del rendimiento del modelo, contribuyendo a evitar el sobreajuste del modelo a los datos.

- **Modelos preentrenados:** Los modelos preentrenados son modelos que se han entrenado previamente con vastos conjuntos de datos, lo que les permite extraer información valiosa a partir de los datos. Esto posibilita que no solo comprendan la estructura y semántica del lenguaje, sino que también capturen patrones complejos y contextos variados. Al contar con un conocimiento previo extenso, estos modelos son altamente versátiles, abordando diversas tareas en el procesamiento del lenguaje natural, tales como la comprensión de texto, generación de texto y traducción automática. Por esta razón, proporcionan una base sólida para los usuarios de *Machine Learning*, permitiendo adaptarse a tareas específicas con una menor necesidad de datos y tiempo de entrenamiento.

- ***Fine-tuning:*** El *fine-tuning* se refiere a un proceso en el cual se realiza un ajuste fino de algunos parámetros, particularmente en algunas capas, de una red neuronal de un modelo preentrenado, para adaptarlo a una tarea específica. Así, en lugar de comenzar el entrenamiento de un modelo desde cero, el *fine-tuning* permite aprovechar el conocimiento previo del modelo preentrenado y ajustarlo para abordar la tarea deseada de manera más efectiva. Este enfoque resulta especialmente valioso en situaciones donde los conjuntos de datos de entrenamiento para tareas específicas pueden ser limitados, ya que se aprovecha la información aprendida en tareas más generales. Al hecho de utilizar un modelo preentrenado sin hacerle fine-tunning se le conoce como *transfer learning*.

2.2. Métricas

Actualmente hay un consenso de que un sistema automatizado como una computadora tiene la capacidad para producir evaluaciones de escritos altamente fidedignas, a diferencia de las proporcionadas por los lectores humanos. Las computadoras no están sujetas a ninguno de los rasgos o condiciones (impaciencia o fatiga, por ejemplo) que hacen que los evaluadores humanos sean menos consistentes a lo largo del tiempo. Además, si diferentes modelos o computadoras se programan de la misma manera (seleccionando los mismos parámetros), prácticamente siempre estarán de acuerdo entre sí, a diferencia de sus contrapartes humanas.

Varios esfuerzos de investigación han demostrado una fuerte relación entre puntuaciones humanas y las automatizadas, lo que implica la validez de las puntuaciones generadas por

computador. Esta implicación, sin embargo, se basa en el supuesto de que las puntuaciones asignadas por los lectores humanos son válidas *per se* y que pueden por lo tanto servir como el "estándar de oro" (*gold standard*) contra el cual validar la puntuación automatizada [37]. En caso de que la evaluación sea una categoría, se han utilizado diversas métricas para realizar este tipo de comparaciones, unas de estas son comúnmente empleadas en estadística o en aprendizaje automático, como lo son exactitud (*accuracy*), *precision*, recordación o exhaustividad (*recall*), así como el error cuadrático medio, y distintas funciones de pérdida. Por ejemplo, en [36] para las puntuaciones exactas, se calcula la antes mencionada estadística QWK - Quadratic Weighted Kappa , lo que permite comparaciones entre las puntuaciones humanas y las puntuaciones del modelo para cada conjunto de ensayos. En algunas ocasiones se reportan medidas de similitud (coincidencias exactas o cercanas) en las notas dadas por evaluadores humanos y las notas producidas de manera automática. La mayoría de los métodos de calificación automática de textos utiliza la métrica de evaluación QWK y otros usan el error absoluto medio MAE, la raíz del error cuadrático medio RMSE o el coeficiente de correlación de Pearson en caso de puntajes continuos. Estas métricas permiten conocer el grado de precisión de la calificación realizada por el modelo frente a las calificaciones reales. QWK varía usualmente entre 0 y 1, donde 1 se interpreta como un acuerdo completo entre evaluadores y 0 desacuerdo total. El error absoluto medio toma valores en los números reales positivos, donde entre más cercano a 0 sea, existe más acuerdo entre evaluadores. El coeficiente de correlación de Pearson corresponde a valores entre -1 y 1, donde 0 es que no existe relación entre las calificaciones y 1 o -1 existe relación perfecta directa e inversa, respectivamente. En la Tabla **2-3**, se presenta un ejemplo de cómo se usa la métrica QWK para comparar distintos modelos en el conjunto de ensayos ASAP mencionado anteriormente. Cabe resaltar que ninguna arquitectura es sistemáticamente mejor en los 8 subconjuntos distintos de ensayos.

Item	1 qwk (%)	2 qwk (%)	3 qwk (%)	4 qwk (%)	5 qwk (%)	6 qwk (%)	7 qwk (%)	8 qwk (%)	Avg qwk(%)
BERT	79.20	67.99	71.52	80.08	80.59	80.53	78.51	59.58	74.75
XLNet	77.69	68.06	69.29	80.62	78.33	79.37	78.67	62.68	74.34
LSTM [13]	77.50	68.70	68.30	79.50	81.80	81.30	80.50	59.40	74.63
BERT Ensemble	80.21	67.21	70.82	81.56	80.63	81.47	80.42	59.74	75.26
XLNet Ensemble	80.49	68.59	70.09	79.56	79.94	80.54	80.02	59.76	74.87
BERT + XLNet Ensemble	80.78	69.67	70.31	81.90	80.82	81.45	80.67	60.46	75.76
LSTM (+CNN) Ensemble [13]	82.10	68.80	69.40	80.50	80.70	81.90	80.80	64.40	76.08
EASE (Bag of Words) [13]	78.10	62.10	63.00	74.90	78.20	77.10	72.70	53.40	69.90
H1-H2 Agreement	72.08	81.23	76.90	85.10	75.27	77.59	72.09	62.03	75.29

Tabla 2-3.: Medidas QWK para distintos métodos de aprendizaje automático aplicados en los diferentes subconjuntos de ensayos de ASAP, adaptado de [37].

2.3. Retos

Las problemáticas principales de la calificación automática de textos argumentativos están relacionadas directamente con el entendimiento del texto. Las metodologías implementadas

no evalúan los ensayos basados en el conocimiento, sino en su relación gramatical. Por ejemplo, en un texto la palabra "apuntar" puede adquirir dos significados diferentes, la acción de escribir y la acción de señalar. En estas metodologías existe el problema de que al transformar las palabras en vectores, sus algoritmos asignan el mismo vector para la palabra sin tener en cuenta el significado. Asimismo, las metodologías que usan similitud en cuanto a bolsas de palabras, por ejemplo, las metodologías estadísticas tienen la desventaja que comparan oraciones por palabras y no por la intención, es decir, las frases con las mismas palabras, pero con diferente significado tendrían la misma relevancia. Por ejemplo, "Ricardo refutó el ensayo de Lina" y "Lina refutó el ensayo de Ricardo". Por otro lado, la mayoría de los sistemas de calificación automática de ensayos no son capaces de atender aspectos esenciales de la escritura como la efectividad retórica, la argumentación, el propósito o la audiencia. Además, la efectividad de los modelos AES generalmente se ha limitado a tipos de ensayos más cortos, como los que se encuentran en las pruebas estandarizadas, y han sido menos efectivos para calificar evaluaciones de escritura más auténticas [3].

Existen diversos desafíos en este campo, como llegar a aplicar textos de evaluación automática para exámenes oficiales o cursos de alto nivel, aunque aún se está lejos de este objetivo. Un desafío en el cual ya se está trabajando es todo lo relacionado con el campo a *Automatic writing evaluation systems* (AWE), cuyas técnicas están basadas generalmente en sistemas de calificación automática y pretenden ir más lejos proporcionando además retroalimentación al estudiante sobre sus habilidades escritas [3]. Por otro lado, un reto importante en la calificación automática de textos está relacionado con evaluar la creatividad. En ese caso, la medida de similitud no sería confiable, es decir, si los textos son similares se puede llegar a la conclusión de que no existe creatividad en alguno de ellos. Entonces, el problema surge al identificar textos con baja similitud, pues no se puede diferenciar si el texto evaluado es creativo o por el contrario carece de creatividad. Por último, existe el reto constante de seguir experimentando con estos métodos en corpus de otros idiomas o de poblaciones con características especiales, donde se puedan medir distintos tipos de indicadores en el mismo corpus de textos, campos en los que se sigue avanzando actualmente.

2.4. Técnicas de Representación

También conocidos popularmente como *word embeddings* estos métodos permiten interpretar los tokens o palabras como vectores numéricos con los que se puede ajustar un modelo; normalmente se construyen a base de redes neuronales, entre los más populares se encuentran: word2vec, Bag of Words, Glove, TfidfVectorizer [38]. Existen otras técnicas personalizadas para los modelos basados en transformers como BERT y DeBERTa, donde los parámetros de extracción de características también se aprenden en el entrenamiento [39].

- BOW: La técnica *Bag of Words* (BoW) representa un enfoque fundamental en el pro-

cesamiento de texto dentro del ámbito del Procesamiento del Lenguaje Natural y la minería de texto. Su esencia radica en la simplificación de documentos de texto al tratarlos como conjuntos de palabras, obviando el orden y la estructura gramatical [40]. En este modelo, se construye un vocabulario que abarca todas las palabras únicas presentes en un conjunto de documentos, y cada documento se convierte en un vector cuyo tamaño coincide con el del vocabulario. Cada posición en este vector corresponde a una palabra en el vocabulario, y el valor en cada posición indica la frecuencia de esa palabra en el documento. Esta representación, por ejemplo, asignaría (2,3,0) a un documento con las frecuencias mencionadas para las palabras "gato", "perro" y "juguete". La técnica BoW encuentra aplicación en la clasificación y agrupación de documentos basados en su contenido textual, siendo ampliamente empleada en diversas aplicaciones como análisis de sentimientos, clasificación de texto, etiquetado automático y recomendación de contenidos.

- Word2vec: Word2Vec una técnica implementada dentro de un modelo de aprendizaje no supervisado, utiliza una red neuronal de tres capas para aprender asociaciones entre palabras a partir de extensos corpus de texto [41]. Su esencia reside en la habilidad para capturar conexiones semánticas y contextuales entre palabras, representándolas como puntos cercanos en un espacio vectorial tridimensional. Esta técnica presenta dos variantes distintivas: Skip-gram y *Continuous Bag of Words* (CBOW).

  En la arquitectura de la Bolsa Continua de Palabras, el modelo predice la palabra actual a partir de una ventana de palabras de contexto circundante. El orden de las palabras en el contexto no afecta la predicción, siguiendo un enfoque de "bolsa de suposición de palabras". Por otro lado, en la arquitectura Skip-gram, también conocida como N-gramas con saltos continuos, el modelo utiliza la palabra actual para anticipar la ventana de palabras de contexto que la rodea. Esta variante asigna un mayor peso a las palabras de contexto cercano en comparación con las palabras de contexto más distante, lo que destaca su importancia relativa.

  Al recibir un corpus extenso como entrada, Word2Vec genera un espacio vectorial de múltiples dimensiones, asignando cada palabra única en el corpus a un vector correspondiente. La disposición de estos *embeddings* en el espacio vectorial se realiza de tal manera que las palabras que comparten contextos comunes en el corpus se encuentran cercanas entre sí. Esta organización facilita la interpretación y manipulación de las representaciones semánticas, permitiendo a Word2Vec destacar no solo las similitudes léxicas, sino también las relaciones intrínsecas entre palabras que comparten contextos en el lenguaje.

- Glove: Es una técnica de representación de palabras propuesta por científicos de Standford, constituye un modelo destacado para la representación distribuida de palabras[42]. Este algoritmo de aprendizaje no supervisado se dedica a obtener representaciones vec-

toriales de palabras mediante la asignación de estas a un espacio de significado, donde la distancia entre palabras refleja su similitud semántica. El proceso de entrenamiento se lleva a cabo utilizando estadísticas de co-ocurrencia global de palabra a palabra en un corpus, y las representaciones resultantes exhiben estructuras lineales en el espacio vectorial de palabras. GloVe se distingue por su capacidad para capturar relaciones semánticas complejas y ofrecer representaciones vectoriales que reflejan de manera efectiva las interacciones y significados subyacentes entre palabras en el lenguaje.

- Tfidfvectorizer: (Term Frequency-Inverse Document Frequency) es un método ampliamente utilizado en el procesamiento de lenguaje natural [43] y la recuperación de información que se basa en evaluar la importancia de las palabras en un corpus de documentos. El cálculo de TF-IDF tiene en cuenta tanto la frecuencia de una palabra en un documento específico como su frecuencia en todo el corpus.

  Las palabras que aparecen con alta frecuencia en un documento y también en todo el corpus (a menudo conocidas como "stopwords.º palabras comunes como .[el]", "la", "de", etc.) recibirán un valor bajo de TF-IDF. Esto se debe a que estas palabras no aportan información distintiva y, por lo tanto, no son útiles para distinguir o clasificar documentos.

  En contraste, las palabras que aparecen en solo algunos documentos del corpus tendrán un valor de IDF (*Inverse Document Frequency*) mayor que aquellas que aparecen en un gran número de documentos. Esto significa que las palabras menos comunes, pero específicas para ciertos documentos, tienen un mayor peso en la representación TF-IDF.

  El cálculo del TF-IDF se realiza mediante la fórmula:

$$tf\,\mathrm{idf}(t, d, D) = tf(t, d) \times \mathrm{idf}(t, D)$$

  donde
  $tf_{(t,d)}$ es la frecuencia con la que aparece el término t en el documento d,
  $\mathrm{idf}(t, D)$ es la frecuencia inversa con la que aparece el término t en el corpus de documentos D y se calcula mediante la siguiente fórmula:

$$idf(t, D) = \log\left(\frac{N}{\text{count } (d \in D : t \in d)}\right)$$

  donde N es el número total de documentos y el denominador se refiere al número de documentos que contienen el término t.

2.5. Modelos de Calificación

2.5.1. Regresión Lineal

La regresión lineal es un método estadístico utilizado para modelar la relación entre una variable dependiente (también llamada variable de respuesta) y una o más variables independientes (también conocidas como variables predictoras o regresoras). El objetivo principal de la regresión lineal es encontrar una ecuación lineal que describa la relación entre estas variables. La ecuación lineal general se expresa de la siguiente manera:

$$Y = aX + b$$

donde:

- Y es la variable dependiente que queremos predecir,
- X es la variable independiente que utilizamos para hacer la predicción,
- a es la pendiente de la línea (coeficiente de regresión) que representa cómo cambia Y en respuesta a cambios en X, y
- b es la intersección en Y cuando $X = 0$ (también conocida como el término de intercepto).

La regresión lineal busca encontrar los valores de b que minimicen la diferencia entre las predicciones del modelo y los valores reales observados. Esto se hace ajustando la línea de regresión de tal manera que la suma de los cuadrados de las diferencias entre las predicciones y los valores reales (conocida como la suma de los residuos cuadrados) sea la más pequeña posible.

Para modelos de regresión lineal múltiple, donde hay más de una variable independiente, la ecuación se extiende de la siguiente manera:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

donde:

- Y es la variable dependiente,
- β_0 es el término del intercepto y
- $\beta_1, \beta_2, \ldots, .., \beta_p$ son los coeficientes de regresión correspondientes a las variables predictoras $X_1, X_2, \ldots, X_p$.

El estimador de los coeficientes de regresión ($\hat{\beta}$) se calcula mediante la siguiente fórmula en forma vectorial:

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T Y$$

donde:

- X^T es la matriz transpuesta de X,
- $\left(X^T X\right)^{-1}$ es la matriz inversa de la matriz de productos cruzados $X^T X$ y
- Y es el vector de la variable dependiente.

2.5.2. Regresión LightGBM

LightGBM (*Light Gradient Boosting Machine*) es un método de mejora de gradiente que utiliza algoritmos de aprendizaje basados en árboles de decisión [44]. Está diseñado para trabajar con conjuntos de datos grandes y de alta dimensionalidad, siendo especialmente eficiente en problemas de clasificación y regresión.

El algoritmo incorpora dos técnicas innovadoras conocidas como Muestreo Unilateral Basado en Gradientes (GOSS) y Agrupación Exclusiva de Características (EFB):

- GOSS es una técnica que mejora la eficiencia del entrenamiento al reducir el número de instancias utilizadas durante cada iteración del proceso. En lugar de muestrear aleatoriamente instancias de manera uniforme, GOSS prioriza aquellas instancias con gradientes más altos, es decir, las que contribuyen más a la pérdida total, realizando muestreo aleatorio en las instancias de menor gradiente [45].

- EFB es una técnica que permite reducir el número de características, acelerando así el proceso de entrenamiento, especialmente en conjuntos con características dispersas. Esta técnica busca agrupar óptimamente las características, reduciendo este problema a uno de grafos, donde las características se toman como vértices y se agregan aristas para cada par de características que no son mutuamente excluyentes [45].

2.5.3. XGBoost

El *Extreme Gradient Boosting* (XGBoost) es un algoritmo de aprendizaje automático supervisado, se destaca por su eficacia en problemas de regresión y clasificación. Similar al modelo LightGBM, XGBoost se basa en árboles de decisión y sobresale por su velocidad y escalabilidad en diversos escenarios. Hace uso de técnicas como el procesamiento en paralelo, poda de árboles y regularización para prevenir el sobreajuste [46].

El algoritmo emplea el método del descenso del gradiente mediante un ensamblado secuencial de árboles de decisión. Cada árbol aprende de los anteriores y corrige los errores acumulados hasta que ya no es posible mejorar más la predicción.
El proceso inicia con la generación de un árbol F_0 para predecir una variable objetivo Y asociando un residual al resultado $Y - F_0$. Luego, se construye un nuevo árbol h_1 que ajusta el error del paso anterior. Los resultados de F_0 y h_1 se combinan para obtener el árbol F_1, reduciendo el error cuadrático medio. Este proceso continúa hasta minimizar el error de la siguiente manera [47]:

$$F_m(x) < -F_{m-1}(x) + h_m(x)$$

XGBoost optimiza la función objetivo utilizando el gradiente y la hessiana de la función de pérdida. Esto mejora la convergencia y la velocidad de entrenamiento.
Dentro de sus principales parámetros se encuentran:

- *Objective:* La tarea que se quiere optimizar con XGBoost. En este trabajo se utiliza regresión con pérdida al cuadrado.
- *N_estimators:* Número de árboles que llevan a cabo el *boosting*, en este caso se usaron 1000 iteraciones.

2.5.4. Support Vector Regression (SVR)

Es un método basado en las *Support Vector Machines* [48] que se enfoca en identificar un hiperplano que se adapte de manera óptima a los datos, permitiendo una tolerancia específica a los errores. Básicamente, SVR es una extensión de la regresión lineal con restricciones, buscando encontrar un equilibrio entre el ajuste preciso de los datos y la minimización de la complejidad del modelo. En el marco de la SVR, el propósito es descubrir una función $f(x)$ que tenga, como máximo, una desviación ϵ con respecto a los objetivos obtenidos y_i para todos los datos de entrenamiento, al mismo tiempo que sea lo más plana posible. En ocasiones se permite que algunos puntos queden por fuera del margen siempre y cuando sean menores a ξ_i. El objetivo es minimizar la margen.

En el caso de un problema lineal la función de la regresión es así:

$$y = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \cdot \langle x_i, x \rangle + b$$

Mientras que para la mayoría de casos, el problema no es líneal y se representan los datos en una dimensión más alta donde sí son separables mediante los métodos de kernel.

$$y = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \cdot \langle \varphi(x_i), \varphi(x) \rangle + b = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \cdot K(x_i, x) + b$$

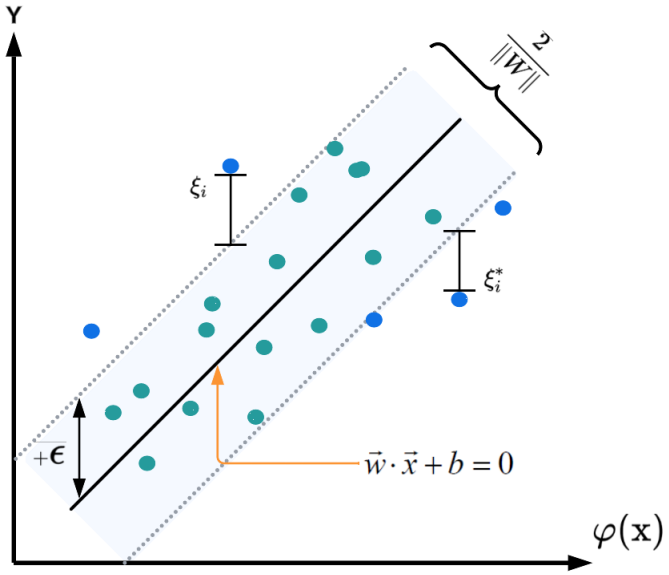

Figura 2-1.: Ejemplo de ajuste de SVR con X linealmente separable.

En la Figura **2-1** se aprecia los puntos a ajustar y el margen, equivalente a $\frac{2}{\|W\|}$, que se quiere minimizar en este caso, a diferencia del SVM original. En el eje X ya no están los datos originales sino en el espacio de características asociado al kernel, donde el kernel entre 2 observaciones corresponde al producto punto en dicho espacio. Específicamente, la función objetivo a minimizar es la siguiente, donde C es un parámetro que indica qué tanto se permiten valores fuera del margen. Los puntos extremos en color azul son los candidatos a ser vectores de soporte para la regresión[49].

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

Sujeto a:

$$y_i - w^T x_i \leq \varepsilon + \xi_i^* \quad i = 1 \ldots N$$
$$w^T x_i - y_i \leq \varepsilon + \xi_i \quad i = 1 \ldots N$$
$$\xi_i, \xi_i^* \geq 0 \quad i = 1 \ldots N$$

Existen distintas funciones de kernel, entre las más comunes encontramos:

- Kernel polinomial

$$k\left(\mathrm{x}_i, \mathrm{x}_j\right) = \left(\mathrm{x}_i \cdot \mathrm{x}_j\right)^d$$

- Kernel de función de base radial.

$$k\left(\mathrm{x}_i, \mathrm{x}_j\right) = \exp\left(-\frac{\left\|\mathrm{x}_i - \mathrm{x}_j\right\|^2}{2\sigma^2}\right)$$

2.5.5. Red Neuronal Perceptrón Multicapa

El perceptron multicapa es una de las redes neuronales más conocidas y utilizadas, se caracteriza por tener tres componentes [50]:

- La capa de entrada, que representa las características o variables de entrada del modelo.
- Las capas ocultas, que realizan transformaciones y cálculos intermedios. En cada nodo de estas capas se realiza una combinación lineal de las entradas ponderadas por sus respectivos pesos, luego esta se pasa a través de una función de activación que distorsiona el valor de salida añadiendo deformaciones no lineales. Las tres funciones de activación más conocidas son:
  1. Sigmoide:

     $$f(x) = \frac{1}{1 + e^{-x}}$$

  2. Tangente Hiperbólica:

     $$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

  3. ReLu:

     $$f(x) = \begin{cases} 0 \text{ para } x < 0 \\ x \text{ para } x \geq 0 \end{cases}$$

- La capa de salida, que produce las salidas del modelo y suele tener una función de activación SoftMax.

Para el entrenamiento de estás redes es necesario el uso de la técnica de retropropagación para minimizar una función de perdida que mida las diferencias entre las predicciones del modelo y las etiquetas reales. Los perceptrones multicapa son versátiles y pueden aplicarse a una variedad de tareas, como clasificación y regresión. Sin embargo, la elección del tamaño de la red, la arquitectura y otros hiperparámetros es crucial para evitar el sobreajuste y garantizar un rendimiento óptimo.

2.5.6. Red Neuronal Convolucional

Este tipo de redes neuronales tienen una arquitectura similar a la de un perceptrón multicapa usada frecuentemente para procesar y analizar imágenes o datos con estructura de cuadrícula. La arquitectura de una red neuronal convolucional se fundamenta en capas convolucionales

que aplican filtros para extraer características significativas de la entrada. Durante el entrenamiento, estos filtros son aprendidos, permitiendo la identificación de patrones complejos en los datos, usualmente patrones visuales. Además de las capas convolucionales, una red neuronal convolucional típica también incorpora capas de agrupación (*pooling*) para reducir la dimensionalidad y capas completamente conectadas para realizar la clasificación o regresión final.

2.5.7. Transformers

El *transformer* es un modelo de aprendizaje profundo introducido por Vaswani et al. en 2017 en su artículo "Attention Is All You Need". Constituye la base de los famosos modelos actuales como Dall·e, ChatGpt [51] o Gemini. Este presenta una arquitectura basada en un mecanismo de atención, evitando la recurrencia y convoluciones que presentaban modelos de redes neuronales anteriores; permitiendo con esto una mayor paralelización, una mejora en la eficiencia computacional y requiriendo un menor tiempo de entrenamiento [26].

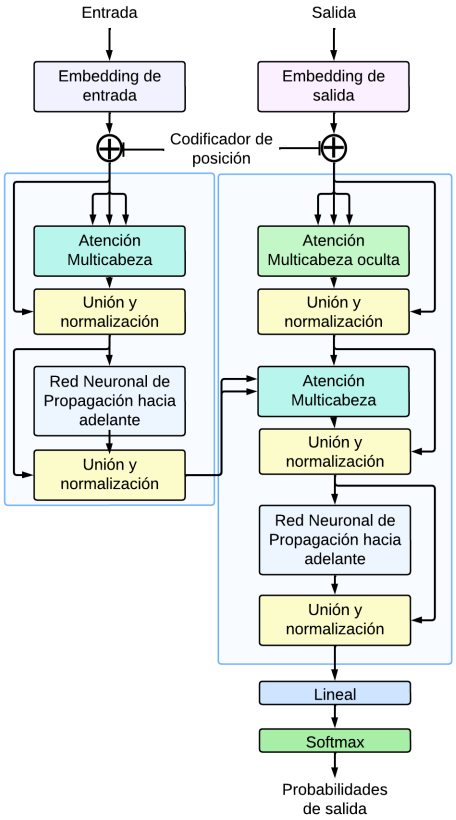

Figura 2-2.: Arquitectura del *transformer*, adaptada de [26].

En la Figura **2-2** se ilustra la arquitectura del modelo *Transformer*, incluyendo:

- Codificador: La parte izquierda de la figura muestra el codificador, compuesto por una pila de capas idénticas. Cada capa consta de dos subcapas: una capa de atención multi-cabeza y una red completamente conectada con propagación hacia adelante.

- Decodificador: A la derecha de la figura, está el decodificador, también compuesto por una pila de capas idénticas, cada una con tres subcapas: atención multi-cabeza, atención de encoder-decoder y una red completamente conectada.

- Conexiones Residuales y Normalización: En toda la arquitectura, se utilizan conexiones residuales y capas de normalización para facilitar el flujo de la información y mitigar problemas como el desvanecimiento del gradiente.

- Máscaras y Posicionamiento: La figura también destaca posterior al *embedding* y la codificación posicional para manejar la información secuencial en la entrada, lo cual permite al modelo saber dónde están los tokens en el texto.

En 2018, Devlin et al. introdujeron un modelo de representación del lenguaje llamado BERT (*Bidirectional Encoder Representations from Transformers*), marcando un hito significativo en el aprendizaje profundo para el Procesamiento del Lenguaje Natural. BERT se distingue por ser un modelo multicapa que se basa en la arquitectura *Transformer* y que es preentrenado de manera no supervisada en un extenso corpus de texto. Esta metodología de preentrenamiento le permite al modelo capturar representaciones contextuales de las palabras, lo que significa que es capaz de entender el significado de una palabra en relación con el contexto que la rodea. La capacidad bidireccional del modelo para procesar información tanto hacia adelante como hacia atrás en una secuencia de texto contribuye a su eficaz comprensión contextual y a su versatilidad en una variedad de tareas relacionadas con el lenguaje natural [35].

Desde entonces, se han desarrollado numerosas variantes y modelos preentrenados basados en *Transformers*, como RoBERTa [29], GPT-3 (*Generative Pre-trained Transformer* 3) [51] de OpenAI, entre otros. Creados para abordar diferentes aplicaciones de NLP como comprensión, generación, inferencia rápida o aplicaciones plurilingües. Asimismo, la investigación y el desarrollo de modelos *Transformer*, como BERT y sus variantes, han tenido un impacto significativo en la calificación automática de textos. Mejorando la precisión y el rendimiento en tareas de clasificación de texto, como análisis de sentimientos, categorización de documentos y calificación de contenido, entre otros [52].

En 2021, surgió una innovadora evolución de los modelos BERT y RoBERTa bajo la denominación de DeBERTa. Este modelo introduce una arquitectura avanzada que incorpora un mecanismo de atención desentrelazada, un decodificador de máscara mejorado y un método

de entrenamiento adversario virtual para el ajuste fino. Estas técnicas mejoran significativamente la eficiencia del preentrenamiento del modelo y su rendimiento en tareas de comprensión y generación del lenguaje natural.

El mecanismo de atención desentrelazada, característico de DeBERTa, permite una captura más efectiva de las relaciones semánticas complejas entre las palabras. A diferencia de BERT, que utiliza un solo vector para representar cada palabra en la capa de entrada, DeBERTa emplea dos vectores distintos para codificar tanto el contenido como la posición de cada palabra. Además, la asignación de pesos entre las palabras se realiza mediante matrices desenredadas basadas en su contenido y posiciones relativas, respectivamente.

Aunque DeBERTa comparte con BERT la fase de preentrenamiento a través del modelado de lenguaje enmascarado (MLM), presenta una mejora adicional al incorporar incrustaciones de posición absoluta de palabras justo antes de la capa softmax. Este ajuste permite que el modelo de DeBERTa decodifique las palabras enmascaradas basándose en las incrustaciones contextuales derivadas tanto de los contenidos como de las posiciones de las palabras. Este enfoque refinado fortalece la capacidad del modelo para entender y generar texto de manera más precisa y contextualizada [53].

3. Metodología

3.1. Selección del conjunto de datos

Para poder aplicar los métodos de aprendizaje automático para la tarea de calificación de textos, se buscaba un conjunto calificado de ensayos reciente y sobre el que no se hubiera trabajado tanto en la literatura y se optó por tomar un conjunto de un concurso de la página Kaggle al final del 2022 llamado *Feedback prize 3.0*, el cual es la tercera y última competición dentro de esa serie de desafíos [7]. Es un conjunto adecuado para probar una metodología de calificación automática ya que no cuenta con demasiados datos (menor a 4000), lo cual suele ser el escenario real a la hora de encontrar conjuntos de ensayos anotados para probar calificarlos automáticamente. Además, la base de datos tiene 6 notas distintas sobre cada ensayo y estos eran de un nivel de formación escolar en el que el lenguaje no suele representar una complejidad adicional.

	text_id	full_text	cohesion	syntax	vocabulary	phraseology	grammar	conventions
0	0016926B079C	I think that students would benefit from learn...	3.5	3.5	3.0	3.0	4.0	3.0
1	0022683E9EA5	When a problem is a change you have to let it ...	2.5	2.5	3.0	2.0	2.0	2.5
2	00299B378633	Dear, Principal\n\nIf u change the school poli...	3.0	3.5	3.0	3.0	3.0	2.5
3	003885A45F42	The best time in life is when you become yours...	4.5	4.5	4.5	4.5	4.0	5.0
4	0049B1DF5CCC	Small act of kindness can impact in other peop...	2.5	3.0	3.0	3.0	2.5	2.5

Figura 3-1.: Fragmento del conjunto de datos original.

El conjunto de datos utilizado en el presente trabajo e ilustrado en la Figura **3-1** proviene del corpus ELLIPSE (*English Language Learners Insight, Proficiency and Skills Evaluation*) y contiene ensayos argumentativos escritos por estudiantes estadounidenses que aprenden inglés en los grados 8 a 12. Los ensayos fueron anotados por evaluadores expertos produciendo unos elementos o indicadores comúnmente encontrados en la escritura argumentativa. Fue obtenido como material disponible (solamente el conjunto de entrenamiento: train.csv) en un concurso llamado Feedback prize 3 publicado en 2022 en `https://www.kaggle.com/competitions/feedback-prize-english-language-learning/`. En efecto, a la fecha no hay artículos publicados acerca de este conjunto de datos, se encontró un proyecto de final de curso de la asignatura *Natural Language Processing* en la Universidad de Standford que

lo aborda [54], por esto se considera de interés usarlos en el presente trabajo para proponer la metodología de calificación automática que responda a la pregunta de investigación planteada .

La Universidad de Vanderbilt, anfitriona de la competencia, es una universidad privada de investigación en Nashville, Tennessee. Vanderbilt fomenta la investigación interdisciplinaria en aras de lograr descubrimientos con un impacto global. Vanderbilt y el coanfitrión *The Learning Agency Lab*[1] se enfocan en desarrollar herramientas y programas basados en la ciencia del aprendizaje para el bien social. El objeto del concurso es mejorar las herramientas de retroalimentación automatizada para los estudiantes ELL al sensibilizarlos sobre el dominio del idioma mediante aprendizaje automático, procesamiento del lenguaje natural y el análisis de datos educativos.

El conjunto de entrenamiento cuenta exactamente con 3911 observaciones y se tienen los siguientes indicadores de interés anotados por revisores expertos: Cohesión, Sintaxis, Vocabulario, Fraseología, Gramática y Convenciones. La nota de cada indicador va de 1 a 5 donde notas enteras o con .5 en cada unidad son posibles. Cada ensayo fue calificado por al menos dos calificadores en valores enteros de 1 a 5 y, en caso de diferencia de más de una unidad, se buscaba llegar a un acuerdo o se llamaba otro calificador y, posteriormente, se aplicaba un modelo de Múltiples Facetas de Rasch. La malla de criterios utilizados por los calificadores, la cual fue publicada en los foros de Kaggle por uno de los anfitriones de la competición, está disponible en el anexo A del presente documento y en la siguiente dirección web `https://docs.google.com/document/d/1PBNshCCbjIF7Hw4L-dwWHKosNVAHS8P3vHYM_EkpCvA/edit`

3.2. Análisis Exploratorio de los datos

En un primer momento se realizan las verificaciones de datos faltantes o ensayos duplicados en el corpus de datos antes mencionado y no se encuentra ningún inconveniente. Producto del análisis inicial se observa que, en general, son ensayos cortos con una media de 237 palabras por ensayo siendo bastante más frecuentes ensayos más cortos que la media como se puede apreciar en la Figura **3-2** (distribución sesgada a la derecha). Además, se realiza un análisis de frecuencia de las palabras en los ensayos del corpus, donde se evidencia la aparición de varias palabras comunes del contexto estudiantil, como lo ilustra la Figura **3-3**.

[1] Organización independiente sin fines de lucro con sede en Arizona, Estados Unidos.

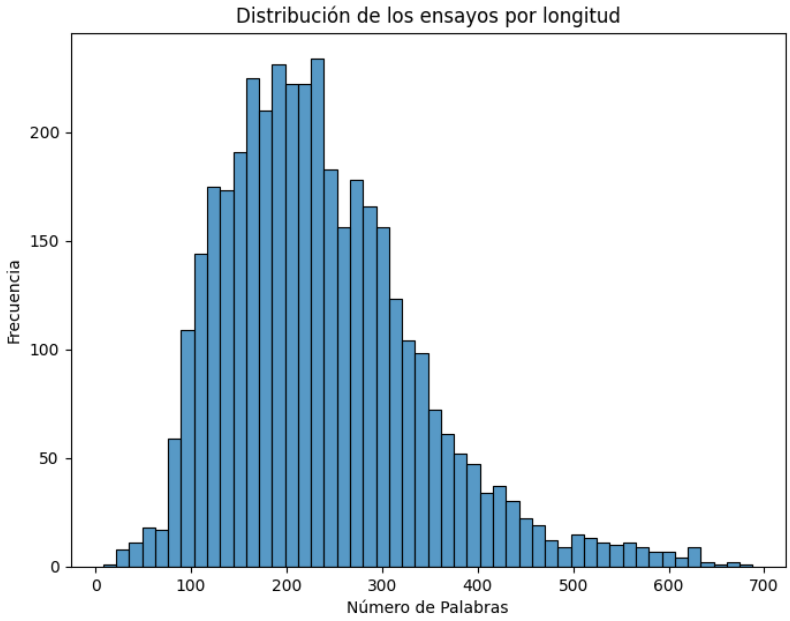

Figura 3-2.: Frecuencia de ensayos por número de palabras.

Como se evidencia en **3-3**, las palabras más frecuentes son las asociadas al contexto escolar y son palabras de uso o de un lenguaje común lo cual se explica además de por la edad de los estudiantes, por el hecho de que son estudiantes que están aprendiendo inglés como lengua extranjera. Esto puede simplificar la tarea de predecir la calificación de los ensayos comparado con otros conjuntos de textos más especializados que se pueden encontrar en la literatura.

Figura 3-3.: Nube de las palabras más usadas en los ensayos.

También, se realizaron algunos análisis de las notas en los 6 indicadores del conjunto, las cuales exhiben un comportamiento normal con la salvedad de que son discretas y se aprecia visualmente que se encuentran positivamente correlacionadas. En la Figura **3-4** se observa la comparación de las notas entre todos los indicadores, por ende las escalas de los ejes en cada pequeña cuadrícula del gráfico es de 1 a 5.

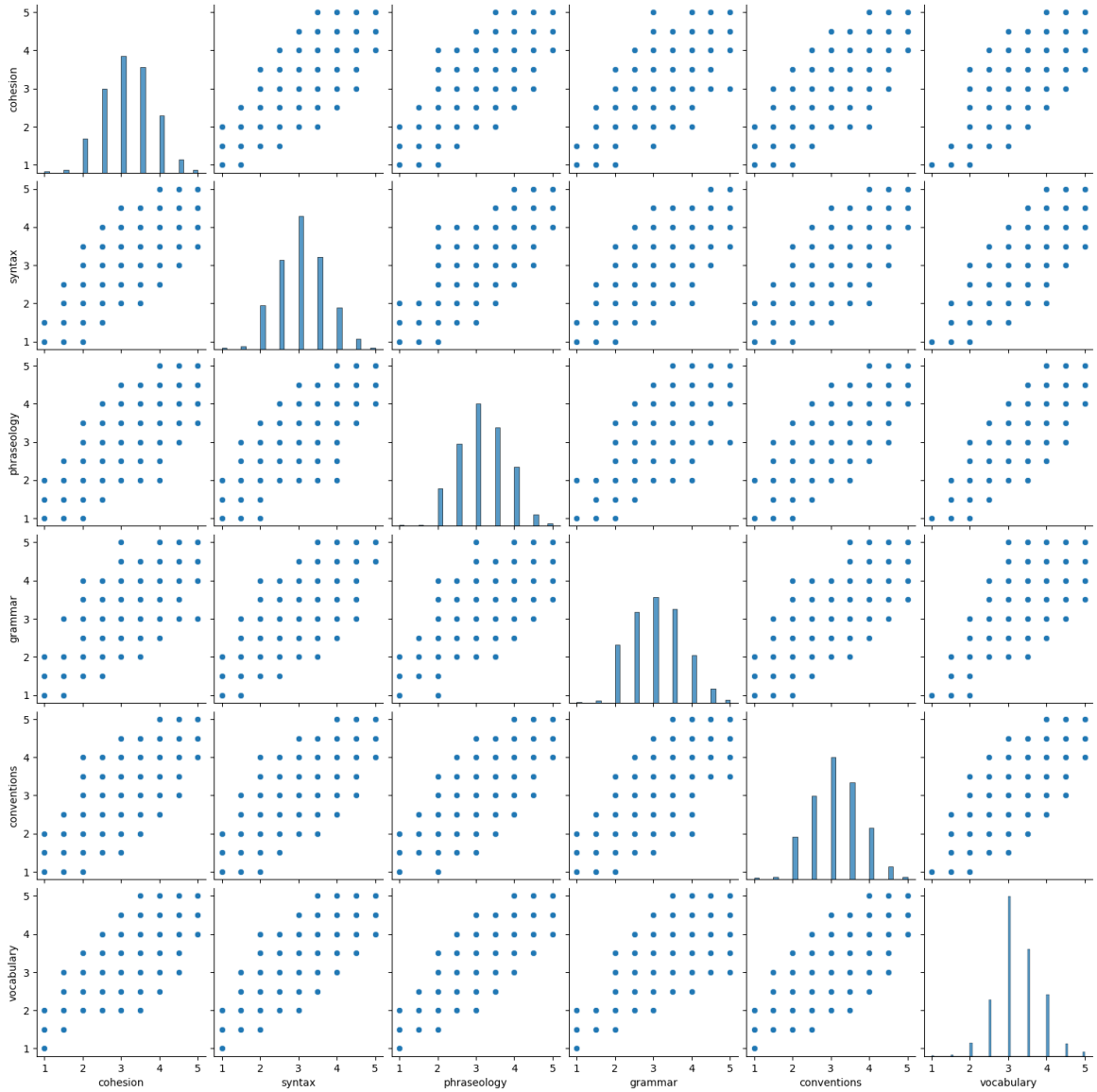

Figura 3-4.: Densidades y diagramas de puntos entre las notas de los 6 indicadores.

En el diagrama de boxplot representado en la Figura **3-5**, se aprecia que, en general, existe una variabilidad similar en todos los indicadores, a excepción del vocabulario, donde la variabilidad es menor. Esta diferencia podría facilitar la predicción del indicador de vocabulario en comparación con los demás.

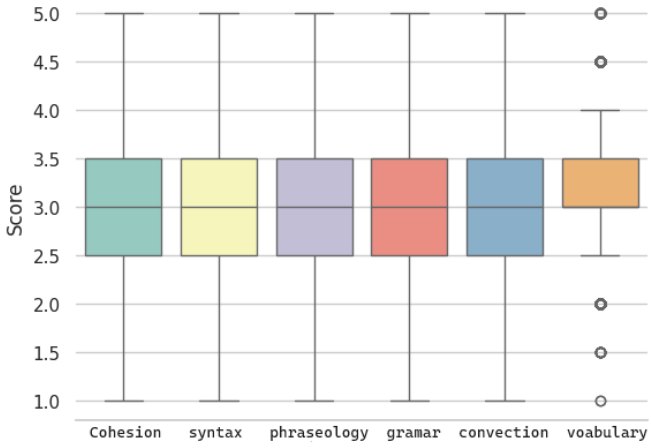

Figura 3-5.: Diagrama de boxplot de los indicadores.

Posteriormente, se realiza un análisis de sentimiento y de subjetividad provenientes de la librería TextBlob, en los cuales se muestra que los estudiantes escriben textos ligeramente positivos y que, en general, tienden a escribir más de manera subjetiva. Estas variables también podrían explicar parte de la variabilidad de las notas en los ensayos. Por otro lado, los conteos de las distintas partes del discurso (pronombres, verbos, conjunciones, preposiciones, etc.) son obtenidos con ayuda de la librería spacy; estos conteos suelen tener distribuciones similares a la distribución Gamma, lo que indica que es menos probable encontrar ensayos con longitudes muy altas, ya sea de palabras, de oraciones o de verbos, entre otros conteos.

En la Tabla **3-1** se reporta el número de ensayos por nota para cada uno de los indicadores. En general, se observa que la mayoría de ensayos tiene nota de 3.0, indicando que los profesores no ponen notas tan altas, ni tan bajas, pero que, en general, tienden a calificar ligeramente más frecuente con notas altas que con notas bajas.

Nota	Sintaxis	Fraseología	Gramática	Convenciones	Vocabulario
1.0	11	10	8	15	2
1.5	29	11	20	20	14
2.0	410	350	544	402	124
2.5	839	772	855	784	528
3.0	1250	1153	994	1151	1503
3.5	867	929	880	908	1007
4.0	388	553	447	484	577
4.5	100	108	134	122	115
5.0	17	25	29	25	41

Tabla 3-1.: Cantidad de ensayos por nota para cada indicador.

Otro aspecto importante a señalar es la intensidad de la correlación entre los indicadores a calificar, ya que como se observa en la Figura **3-4** es positiva y alta, ya que si fueran totalmente independientes, quizás se podría considerar buscar enfoques diferentes para predecirlos por separado. Para este fin, se construye una matriz de correlación basada en la correlación de Pearson debido al carácter numérico de estas variables. En dicha matriz, se aprecia se observa que la menor correlación entre cualquier par de indicadores es de 0.65. Con esto se concluye que a los estudiantes que obtienen un alto puntaje en algún indicador suelen obtener un alto desempeño en los demás y viceversa, esto es un fenómeno que se observa en exámenes como Saber 11 donde las distintas pruebas que los componen también están correlacionadas.

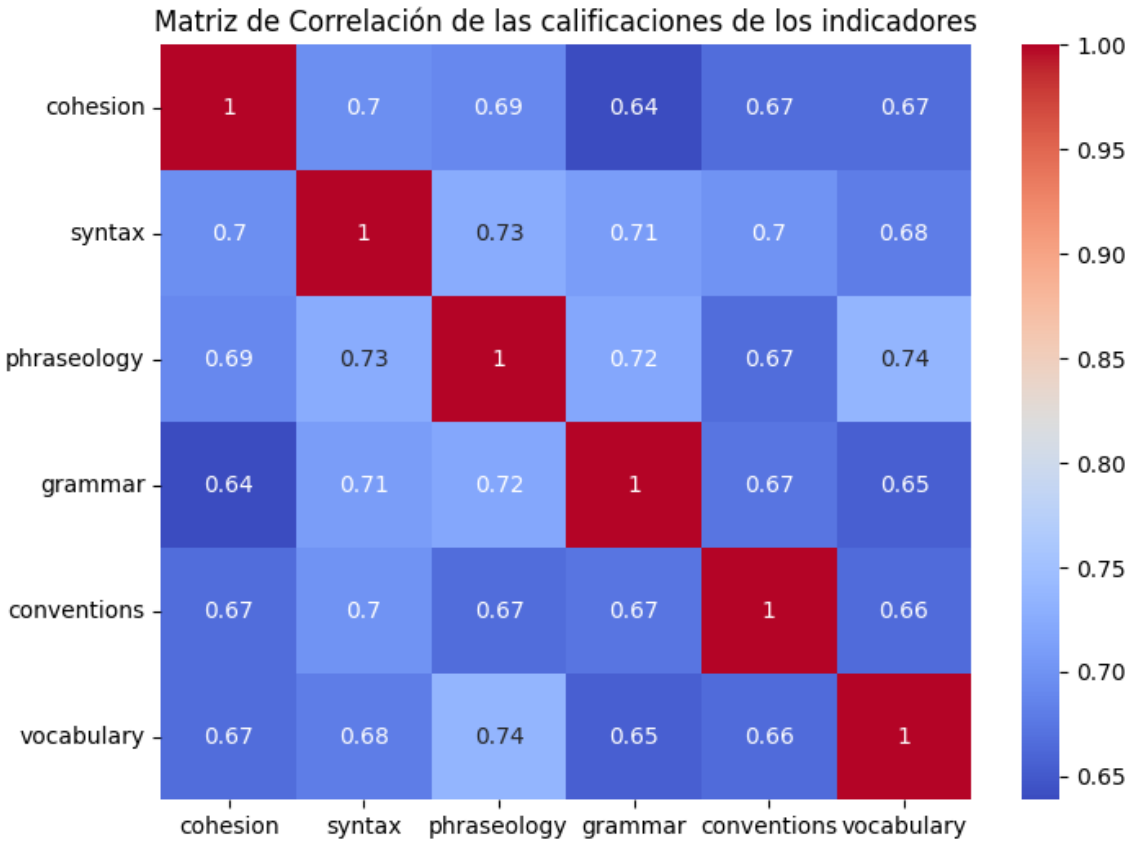

Figura 3-6.: Matriz de Correlación entre los indicadores.

Finalmente, para reforzar esta idea, en la Figura **3-7** se observa que la diferencia entre la máxima nota y la mínima nota obtenida de un ensayo suele ser menor o igual a 1 (una unidad de un total de cinco en la nota). Por lo tanto, también confirma el hecho de que para un texto determinado están relativamente cerca los indicadores a evaluar y de que es razonable predecirlos todos al tiempo.

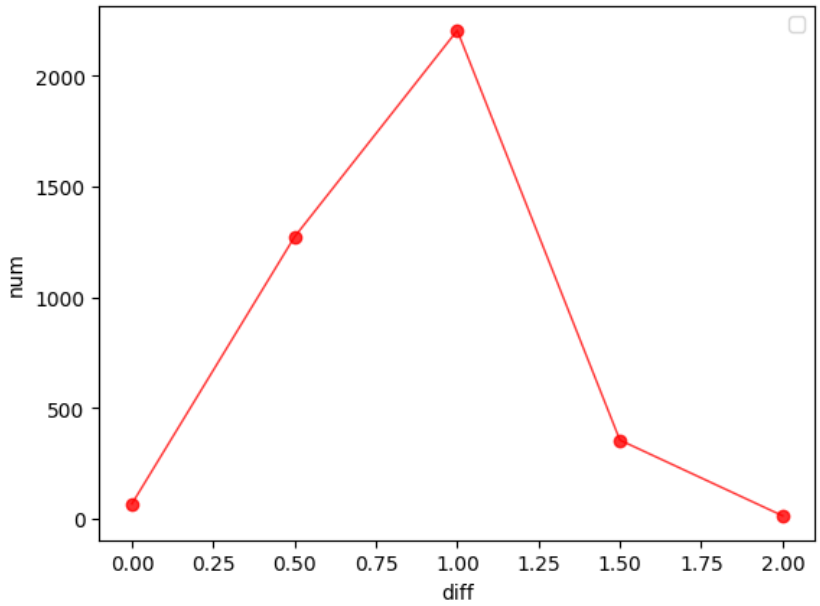

Figura 3-7.: Distribución de ensayos por su diferencia máxima entre notas.

3.3. Métrica de desempeño

Actualmente, existen múltiples maneras para medir la calificación automática de ensayos, tales como correlaciones, QWK, o incluso métricas clásicas en modelos de aprendizaje automático para clasificación como lo son: *accuracy*, *precision*, *f1-score*, entre otras. En este caso, se tomará como referencia la métrica oficial del concurso para tener algo de comparabilidad con los resultados reportados. Esta métrica se basa en la raíz del error cuadrático medio para cada uno de los 6 indicadores. Sin embargo, se debe tener en cuenta que en el presente documento se trabajará con los datos de entrenamiento oficiales del concurso, debido a que los datos de prueba no son públicos, es decir, se entrenará con menos datos y, por tanto, no se podrían alcanzar los mismos resultados.

En primer lugar, se calcula la raíz del error cuadrático medio (RMSE) para cada indicador, esta medida tiene la ventaja de tener las mismas unidades que la variable original a diferencia del error cuadrático medio (MSE).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Posteriormente, se calcula la media entre los 6 RMSE calculados para los indicadores, obte-

niendo así la raíz cuadrada del error cuadrático medio - MCRMSE

$$MCRMSE = \frac{1}{m}\sum_{j=1}^{m}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{ij} - \hat{y}_{ij})^2}$$

o

$$MCRMSE = \frac{1}{m}\sum_{j=1}^{m} RMSE_j$$

donde:
m - número de variables predichas, en este caso 6,
n - número de muestras de prueba,
y_{ij} - valor real i-ésimo de la j-ésima variable ,
$\hat{y}_{ij}$ - valor predicho i-ésimo de la j-ésima variable

En cuanto a la interpretación, dado que la métrica empleada es la raíz del error cuadrático medio está en las mismas unidades de la variable original, en este caso números entre 1 y 5, se podría interpretar como el error promedio en la nota que tiene el modelo al calificar los 6 indicadores de manera simultánea. El mejor desempeño reportado en el concurso fue de 0.433356 en el *LeaderBoard*(LB)[2] y de 0.44073 (CV) en el cálculo del indicador usando validación cruzada[3], la cual es con la que se debería comparar debido al desconocimiento del conjunto de datos de prueba utilizado en la competición.

3.4. Estrategia general

Para poder llevar a cabo la tarea AES existen distintos métodos ampliamente utilizados en la literatura, dicha tarea puede ser realizada con modelos clásicos como el de regresión lineal, con redes neuronales o incluso con modelos del estado del arte como los *transformers*, y modelos basados en ellos en sus más recientes versiones [53]. Todos estos desarrollos comparten componentes teóricos han avanzado mucho el rendimiento en las distintas tareas de NLP los últimos años, y por tanto se considera de interés dejar la trazabilidad del rendimiento de múltiples modelos en cuanto a la calificación automática. En consecuencia, se presentará la experimentación de los modelos empleados y sus variaciones y se reportarán los resultados obtenidos, proponiendo además un método que tenga un excelente desempeño.

Como se describió en el capítulo anterior, el conjunto de datos cuenta con 6 notas diferentes por cada ensayo, las cuales están correlacionadas y tienen forma distribucional similar, aparentemente normal pero discretizada, por tanto se intentará predecirlas todas de manera

[2]Es el puntaje calculado por Kaggle con los datos de prueba privados con los que se calculan los puntajes de la tabla de posiciones del concurso.
[3]Se refiere al *score* reportado por los autores del código mediante validación cruzada.

simultánea con un mismo modelo en lugar de intentar predecirlas por separado. Esto permitirá obtener estrategias que sean buenas para las 6 notas a la vez y que generalicen mejor el proceso de calificación.

Otra decisión importante que se toma temprano es que se va a resolver el problema con predicciones continuas, es decir, será un enfoque de regresión y no de clasificación ya que el poder poner resultados continuos da un espacio de posibilidades más amplio a los modelos para que se ajusten, imitando la lógica de los calificadores humanos pero sin estar restringidos a dar notas con precisión máxima de 0.5, ya que los constructos que se intentan calificar se pueden modelar de manera continua, como se hace en psicometría. Por otro lado, debido al desbalance en las notas de los ensayos un enfoque de clasificación conduciría a muchas predicciones a estar en torno a 3, ya que es lo más frecuente y desconocería la naturaleza continua de cada uno de los constructos a evaluar.

Considerando que no se tiene información de los calificadores o de los enunciados o tareas de los ensayos o alguna información descriptiva de los estudiantes, se descarta usar algún modelo de teoría de respuesta al ítem (TRI), que tendría la ventaja de poder estimar además de la habilidad de cada evaluado incluso la severidad del evaluador o la dificultad de cada enunciado usando por ejemplo un modelo de múltiples facetas de Rasch [55]. Por otro lado, a pesar que se ha trabajado en calificación automática de ensayos mediante aplicaciones de grandes modelos del lenguaje como *chatGPT* a base de prompts [56], ese camino no se investiga tampoco debido a que solo es aplicable de manera manual a unos ensayos determinados que también se le deben pasar al modelo de lenguaje. Es decir, poder crear un método que efectivamente use el modelo de lenguaje que ofrecen GPT 3.5 para leer un conjunto de ensayos y calificarlos con respecto a unas rúbricas incurriría en costos elevados ya que tocaría generar una API que consuma el servicio que ofrece *openAI* a base de créditos que dependen de los tokens que se le pasan.

En este estudio, se aplicarán diversas técnicas, desde las más simples hasta las más complejas, teniendo en cuenta una gran variedad de escenarios y utilizando una combinación de modelos con el objetivo de predecir de manera precisa los puntajes de calificación. Estas estrategias se presentarán en orden, comenzando por las que demostraron un rendimiento menor en el conjunto de entrenamiento. La última estrategia, que se recomendará como base de la metodología de calificación debido a su sobresaliente desempeño, será objeto de ajustes continuos con el propósito de mejorar la calidad de la calificación automática. Se proporcionará una descripción detallada de cada paso y ajuste realizado en esta fase, resaltando los cambios que condujeron a mejoras significativas en la precisión y eficacia de la calificación automática.

Para ajustar los scripts se utilizará el lenguaje python versión 3.10.12 en el entorno Google Colab donde se ha adquirido el plan *pro+* para la facilidad de usar GPU, TPU y experimentar

diversos escenarios. Se emplearon distintas librerías como *scikit-learn*, *TensorFlow*, *pytorch* y *transformers*, entre otras. Se encuentra disponible un repositorio del proyecto donde se incluirá toda la información pertinente relacionada como códigos, modificaciones a la base original, etc. para que el que esté interesado pueda reproducir los modelos. En la Figura **3-8** se presentan las distintas fases de la metodología empleada para cumplir con el propósito planteado.

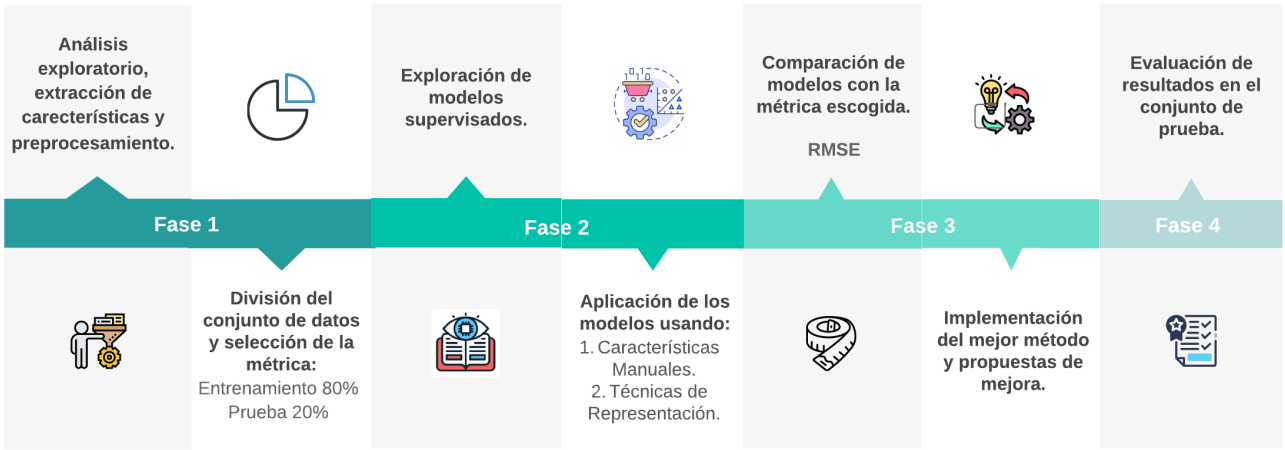

Figura 3-8.: Fases de la metodología aplicada.

3.5. Preprocesamiento

Esta es una tarea común a todos los enfoques y es importante porque puede ayudar a mejorar el rendimiento de distintas tareas de procesamiento de lenguaje natural [57]. Sin embargo, existe discusión sobre si es conveniente realizar el preprocesamiento en el caso de los modelos más modernos, ya que en general los resultados no cambian demasiado aunque esto depende del modelo y del paso puntual dentro del preprocesamiento [58]. Sin embargo, en este mismo artículo se concluye que los mejores desempeños alcanzados en distintas tareas de procesamiento del lenguaje natural suelen tener algunos pasos de preprocesamiento.

En este caso, en el preprocesamiento de los ensayos se llevaron a cabo las siguientes instrucciones:

- Transformar a minúsculas el texto.
- Remover espacios al inicio y final del texto.
- Remover signos de puntuación y en general símbolos.

- Eliminar contracciones (ejemplo: won't sería will not)
- Remover las *stopwords* con ayuda del paquete *nltk*.

3.6. Extracción de Características

Debido a que la base de datos no cuenta con información adicional aparte del texto y las notas que son el objeto de predicción se deben extraer variables o características del texto y estas deben poder ser representadas de manera numérica en una determinada dimensionalidad. Hay distintos tipos de características que se extrajeron de los textos para lograr este resultado numérico, se dividen en las siguientes dos:

3.6.1. Características manuales

Estas son las características extraídas a partir del texto (*handcrafted features*) y tienen la ventaja de ser interpretables; serán usadas para ajustar modelos de regresión y tratar de predecir cada uno de los puntajes en el conjunto de ensayos. Se busca tener un conjunto de variables que puedan influenciar el desempeño de un texto en los indicadores calificados de la base de datos. Las características son las siguientes:

1. n_punct: Número de signos de puntuación.
2. spelling_errors: Número de errores
3. n_unique :: número de palabras únicas
4. n_unique_n_stop: número de palabras únicas, no *stopwords*
5. n_n_word: número de no palabras (grupos de letras que no son palabras).
6. noun_phrase_count : número de frases.
7. PRON : Conteo de partes del discurso: pronombres.
8. VERB : Conteo de partes del discurso: verbos.
9. SCONJ : Conteo de partes del discurso: conjunciones subordinadas.
10. NOUN : Conteo de partes del discurso: Sustantivos.
11. AUX : Conteo de partes del discurso: Verbos auxiliares.
12. ADP : Conteo de partes del discurso: Adposiciones (preposiciones y postposiciones).
13. PUNCT : Conteo de partes del discurso: Signos de puntuación.
14. PART : Conteo de partes del discurso: Partículas.
15. CCONJ : Conteo de partes del discurso: Conjunciones coordinadas.

16. ADV : Conteo de partes del discurso: Adverbios.
17. DET : Conteo de partes del discurso: Determinantes
18. ADJ : Conteo de partes del discurso: Adjetivos
19. SPACE : Conteo de partes del discurso: Espacios
20. PROPN : Conteo de partes del discurso: Pronombres personales
21. NUM : Conteo de partes del discurso: Números.
22. INTJ : Conteo de partes del discurso: Interjecciones
23. SYM : Conteo de partes del discurso: Símbolos
24. X : Conteo de partes del discurso: Otras palabras que no entren en las variables anteriores.
25. polarity : Puntaje de sentimiento (centrado en 0, positivo ¿0, negativo ¡0)
26. subjectivity : Puntaje de sentimiento (puntaje , positivo ¿0, negativo ¡0)
27. spell_score : Puntaje de ortografía basado en lo parecidas (producto punto) que son las palabras a sus correcciones.
28. av_sent_len : estadísticas de las oraciones: longitud promedio.
29. max_sent_len : estadísticas de las oraciones: longitud máxima.
30. min_sent_len : estadísticas de las oraciones: longitud mínima.
31. med_sent_len : estadísticas de las oraciones: longitud mediana.
32. std_sent_len : estadísticas de las oraciones: desviación estándar de longitud.
33. compound : puntaje de polaridad compuesta (entre -1 y 1).
34. negative : puntaje de sentimiento negativo de 0 a 1.
35. positive : puntaje de sentimiento positivo de 0 a 1.
36. neutral : puntaje de sentimiento neutro de 0 a 1.
37. char_len : número de caracteres.

Aunque se consideró ajustar variables adicionales como los distintos índices de legibilidad [59] o de gramática, no se incluyeron características adicionales ni se optimizó la escogencia fina de estas características debido a que, producto de la investigación y experimentación se encontró que proporciona mejores resultados usar técnicas de representación que buscan automáticamente qué extraer de los textos para poder predecir mejor las notas. Por lo anterior, en el método de calificación automática final se optará por usar una técnica de representación debido a su mayor desempeño en todas las situaciones testeadas.

3.6.2. Técnicas de representación

Se consideran 3 técnicas de representación *Word2vec*, *Glove* y *TfidfVectorizer* para los métodos de regresión ajustados en primer lugar. En el caso de los modelos de redes neuronales se usa una función predeterminada *TextVectorization()* basada en la estrategia BOW utilizando n-gramas como características. Finalmente, en los modelos basados en *transformers* se usa un tokenizador especializado para el modelo DeBERTa y posteriormente se realiza el *embedding* de los textos. En consecuencia, se usan los pesos ya disponibles en los modelos preentrenados, para que con la gran cantidad de parámetros fijos y las relaciones del lenguaje aprendidas por el modelo entrenando en millones de textos, pueda representar mejor los ensayos con el objetivo de calificarlos.

3.7. División del conjunto en entrenamiento y prueba

Como los datos tomados de Kaggle son los de entrenamiento del concurso y no se tiene acceso aún a los datos de test oficiales o al conjunto original completo[4], se tuvo que realizar una división para generar un subconjunto de test que no tenga participación ni directa ni indirecta en el entrenamiento y que permita probar la capacidad de generalizar del modelo. Para esto se utiliza la librería *Scikit-learn* y más específicamente la función *train_test_split()* asignando el 80 % de los datos al conjunto de entrenamiento, correspondiente a 3192 ensayos y el 20 % al conjunto de prueba que equivale a 783 ensayos.

Cabe resaltar la importancia de eliminar el factor estocástico y fijar la semilla para que se realice siempre la misma partición y los resultados sean reproducibles y comparables entre modelos, se usa el número 42 como semilla. Esto también condiciona el resultado del entrenamiento pues los valores iniciales de los parámetros se ven afectados por el componente estocástico.

En los enfoques donde se usa validación cruzada, que como se mencionó consiste en dividir en k partes los datos de entrenamiento y entrenar k modelos con k-1 partes para predecir la sección faltante cada vez, se utiliza la función *StratifiedKFold* en donde también se fija la semilla. Para la validación cruzada se escogieron 5 *folds* o pliegues y la idea es predecir en 5 iteraciones usando el 80 % de los datos el 20 % restante y entrenar cada modelo y luego usarlo para predecir el conjunto de prueba ajeno a todos los 5 modelos. Posteriormente, se promedian todas las predicciones para lograr una predicción más acertada.

Por ultimo, un detalle que es importante mencionar es que la escogencia de la partición en pliegues se hace de manera estratificada teniendo en cuenta los 6 indicadores a predecir,

[4]Dicen que lo van a publicar pronto para investigación pero a la fecha de enero de 2024 aún no se encuentra publicado, fuente: `https://the-learning-agency-lab.com/the-feedback-prize-overview/`

esto se realiza para que cada partición sea lo más completa y homogénea posible y contenga ensayos con todas las calificaciones para cada indicador.

3.8. Modelos de regresión usados

Como se determinó que se iba a entrenar y a evaluar sobre los 6 indicadores a la vez, resulta necesario utilizar un sistema que permita transformar un método de regresión de variable objetivo escalar a uno de variable objetivo vectorial; en este caso se usa el método *MultiOutputRegressor()* también de la librería scikit-learn, el cual recibe como argumento un modelo de regresión que sirve para un solo indicador a la vez.

Es crucial establecer una línea base, por lo que en esta sección se comienza ajustando modelos simples, como el modelo de regresión lineal, para evaluar su rendimiento en el conjunto de datos. Posteriormente, se implementan los métodos lightGBM, XGBoost y finalmente el método de regresión SVR. En todos estos modelos, se llevan a cabo experimentos utilizando las características manuales extraídas y los ensayos representados con las diversas técnicas descritas anteriormente, para ver cuál es la forma más óptima de realizar la calificación.

En el caso del SVR, se exploran diferentes valores para el parámetro de kernel, aunque no se observa una variación significativa. Como resultado, se opta por mantener el kernel de función de base radial, que es el valor por defecto y ha demostrado un buen desempeño. En cuanto al parámetro de penalización (C), se exploran varias opciones y se opta por dejarlo en 2.

3.9. Redes neuronales

3.9.1. Perceptrón multicapa

Se sabe que las redes neuronales son arquitecturas potentes capaces de representar cualquier función continua solo con dos capas (teorema de aproximación universal) [60]. En consecuencia, y considerando el reducido número de ensayos se construye una arquitectura simple para ver qué tal se desempeña en la tarea.

Se escoge una capa oculta de 16 neuronas para procesar la entrada para evitar crecer el número de parámetros, dado que la entrada es la representación de frecuencia de cada palabra (Similar a *Bag of Words*, mediante la función *TextVectorization()*, que se considera una capa inicial de la red neuronal en *keras* y *tensorflow*) que en este caso se le fija una dimensión máxima de número de tokens en el vocabulario de 50.000. El vocabulario original de todos los ensayos era de aproximadamente 200.000. Para la capa oculta, se usa una función de

activación de tipo ReLu. Cabe resaltar que se programó la función de pérdida personalizada en *TensorFlow*, correspondiente al MCRMSE descrito anteriormente. Por otro lado, para la capa de salida se seleccionan 6 neuronas con una función de activación *Softmax* que corresponden a los distintos indicadores que se buscan predecir de manera simultánea, donde la intensidad de la salida de la neurona será la que determinará la nota. Esta capa de salida es similar en los modelos usados de aquí en adelante.

3.9.2. Red Neuronal Convolucional

Al igual que en la red neuronal anterior, la red neuronal convolucional implementada utiliza una representación de frecuencia de cada palabra con una dimensión de 50,000, correspondiente al número de tokens diferentes para representar el corpus. Se incorpora una capa convolucional 1D, primero con 128 filtros y luego con 256, y una ventana de convolución de tamaño variable que busca patrones locales en la entrada. Después de cada capa de convolución, se aplica un *pooling 1D* para reducir la dimensionalidad con base en el promedio y capturar características relevantes.

La arquitectura incluye capas de *dropout* además capas densas para reducir el sobreajuste y adaptarse eficientemente al conjunto de datos. Las capas densas se utilizan con activación ReLU, donde la salida de la capa anterior se conecta a cada unidad de la capa actual. El modelo tiene capas densas con 128, 16 y 6 neuronas, respectivamente.

3.10. Transformers: DeBERTa

A continuación se llega al modelo que tiene actualmente métricas del nivel de estado del arte en varias aplicaciones dentro del procesamiento del lenguaje natural, originalmente propuesto en [53] y mejorado con un mecanismo que cifra la posición de los tokens y su significado de manera separada [61]. Para poder implementarlo se usó la librería pytorch y se utilizaron recursos gráficos para mejorar su entrenamiento. Se pusieron 2 capas completamente conectadas adicionales, la primera con 64 neuronas y luego la capa de salida con 6 neuronas al igual que las arquitecturas de red neuronal y red neuronal convolucional.

Se debe tener en cuenta la recomendación de HugginFace[5] de que al usar un modelo previamente entrenado, es importante usar el tokenizador asociado, ya que este dividirá en tokens al texto que se le proporcione, de la misma manera en que lo hizo para el corpus de preentrenamiento y usará el mismo token de correspondencia para indexar (que normalmente se

[5]Empresa de tecnología que se dedica al desarrollo de herramientas y plataformas de procesamiento de lenguaje natural o NLP basadas en inteligencia artificial. Crearon una plataforma de código abierto llamada *Transformers*, la cual le permite a los desarrolladores de todo el mundo acceder y utilizar modelos de NLP preentrenados de última generación.

llama un vocablo).

3.10.1. Mejoras metodológicas en su implementación

En primer lugar, se le realizó *fine-tunning* al modelo DeBERTa v3 base, el cual cuenta con 86 millones de parámetros y fue entrenado con un vocabulario de 128 mil tokens distintos. En segundo lugar, se ajustó el modelo DeBERTa-v3-large que tiene 304 millones de parámetros. En la siguiente fase, se realizó validación cruzada con 5 pliegues para obtener una mejor generalización. En seguida se realizó un proceso de búsqueda en artículos de métodos para poder mejorar aún más el rendimiento de un modelo como el ajustado. La estrategia aplicada es ir implementando de manera experimental cada una de las propuestas o técnicas encontradas y solamente incorporar una de ellas si disminuía el MCRMSE. Entre las técnicas que fueron implementadas e incorporadas siguiendo este principio se encuentran las siguientes:

- Explorar distintos métodos de *pooling*: implica que el modelo condensa de maneras diferentes las representaciones de los tokens de manera única para expresar el texto completo. Al combinar diferentes enfoques de *pooling*, se logra una representación mejorada del texto y su contexto.
- Aumentar la longitud máxima de tokens: esto aumenta directamente el tiempo de entrenamiento, pero también permite explorar soluciones en un espacio de dimensión mayor. Por tanto, si no se tiene una restricción muy estricta en cuanto al coste computacional o al tiempo de procesamiento, podría explorarse variar este hiperparámetro.
- Incorporar *Adversarial Weights Perturbations*: es una técnica donde, durante el entrenamiento de una red neuronal, se hacen pequeñas y deliberadas alteraciones (perturbaciones) a los pesos de la red de una manera que imita un ataque adversario. El objetivo de esto es hacer que la red sea más resistente: al "experimentar" estos pequeños cambios o ataques durante el entrenamiento, la red aprende a manejar mejor situaciones inesperadas o difíciles cuando se encuentra con datos nuevos, lo que mejora su capacidad de generalización robusta [62].
- Incorporar *Dropout* multi-muestra: sirve para hacer distintos *dropouts* en capas diferentes y luego pasarlas por una capa densa con los mismos pesos para finalmente promediar los valores. En general, este método hace que el modelo apague distintas secciones y se fuerce a aprender en paralelo de esta forma, logrando una mejor generalización [63].
- Ensamblaje de modelos: es una estrategia que se presenta en [64] para resolver el problema planteado en la versión anterior del concurso, y ha sido muy utilizada y comentada por los participantes del concurso actual. En este caso, se usa para ensamblar un modelo entrenado con un modelo de regresión basado en SVR que se entrena con

las representaciones de los ensayos en un total de 10 modelos diferentes. Los modelos preentrenados fueron obtenidos de *HuggingFace*.

3.11. Repositorio del proyecto

Cabe resaltar que todos los scripts utilizados en el presente trabajo, así como los conjuntos de ensayos, se encuentran disponibles en el siguiente repositorio público de *GitHub*: `https://github.com/joangb2020/Trabajo-Final-Maestria` En el repositorio mencionado se encuentran referenciados todos los códigos que se tuvieron en cuenta para construir los códigos. Además, estos pueden ser encontrados en el Anexo 2 B. Se invita a cualquier lector a quien le interese que haga uso de los códigos, ya sea para desarrollos de calificación automática u otro tipo de tareas del procesamiento del lenguaje natural. Cualquier sugerencia o comentario será bienvenido.

4. Resultados

A continuación, se exponen los resultados obtenidos mediante la implementación de los diversos modelos y técnicas de aprendizaje automático descritos en el capítulo *Metodología*, precisando los detalles en cada caso en las respectivas secciones. Adicionalmente, al final del presente capítulo se describe el método elaborado e implementado para calificación automática, el cual se considera el principal resultado del presente trabajo. También se presenta una comparación de todas las estrategias exploradas. En cada caso se señala el mejor rendimiento de los modelos ofreciendo ideas valiosas sobre su eficacia y capacidad de predecir la nota, o calificar. No se precisan detalles del tiempo de procesamiento o la complejidad computacional debido a que una comparación rigurosa que incluya estos aspectos excede el alcance del presente trabajo y tiene altas exigencias de hardware. Adicionalmente, es de suma importancia señalar que estos resultados se obtuvieron entrenado los modelos con el 80 % de los datos de entrenamiento de la competición original y por eso posiblemente se obtendrá un error un poco mayor que en los reportados en la tabla de posiciones del concurso, no obstante los valores obtenidos fueron bastante cercanos (a menos de dos centésimas en la escala de las notas del mejor desempeño reportado en la tabla final de posiciones).

4.1. Regresores

Considerando los métodos de regresión lineal y los regresores basados en árboles de decisión, como XGBoost y LightGBM, se observa que logran un MCRMSE (promedio por columna de la raíz del error cuadrático medio) entre 0.6 y 0.7, equivalente a 6 o 7 décimas en la escala original de las notas. Estos métodos son sencillos de implementar y presentan tiempos de entrenamiento reducidos. La única técnica que supera su rendimiento al utilizar características extraídas manualmente es la regresión lineal. Por lo tanto, se recomienda planificar cuidadosamente el conjunto de características a extraer al aplicar este método.

En contraste, tanto para LGBM como para XGBoost, se observa una mejora sustancial al utilizar técnicas de representación, como TfidfVectorizer. En el caso de LGBM, el método empleado permitía la opción directa de validación cruzada, cuando se activó esta opción, se registraron mejoras tanto en el caso de las características manuales como en el de la técnica de representación TfidfVectorizer, alcanzando un valor de 0.546 en la escala de unidades de la nota.

4.2. SVR

Este método también presenta un ajuste muy rápido gracias a su implementación eficiente para aprovechar la GPU [1]. Incluso permite considerar una gran cantidad de características provenientes del ensamblaje de varias técnicas de representación. Se exploraron explícitamente las variables extraídas manualmente y varias técnicas de representación como Word2Vec, Glove y TfidfVectorizer. De manera análoga a los otros modelos de regresión previamente mencionados, este método presenta un rendimiento superior al utilizar técnicas de representación vectorial en comparación con las características manuales. Dentro de estas técnicas de representación, *TfidfVectorizer* demostró un rendimiento especialmente destacado sin usar validación cruzada, alcanzando un valor de 0.547. Debido a su buen desempeño, este modelo se incluirá como parte de un ensamblaje en el método propuesto de calificación que se describirá posteriormente.

4.3. Redes Neuronales

El uso de esta arquitectura permite mejorar un poco los resultados alcanzados llegando a 0.543 con un modelo de 2 capas y 16 neuronas en la capa oculta. Se dejó correr durante 20 épocas y se obtuvieron los resultados representados en la Figura **4-1** en cuanto a su función de pérdida:

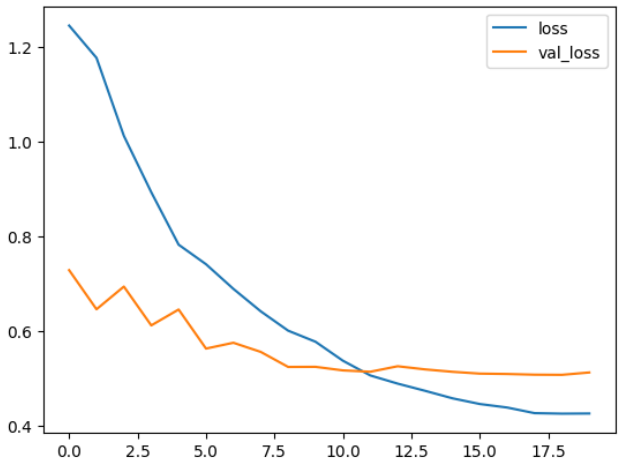

Figura 4-1.: Función de pérdida de la red neuronal durante el entrenamiento.

También, se ajustó una red convolucional para ver si podía mejorar los resultados obtenidos por la red neuronal simple, pero en realidad obtuvo peores resultados y se evidencia que se sobreajusta en menos épocas, aproximadamente en 5.

[1] Librería Cuml, detalles técnicos en: `https://docs.rapids.ai/api/cuml/stable/api/#support-vector-machines`

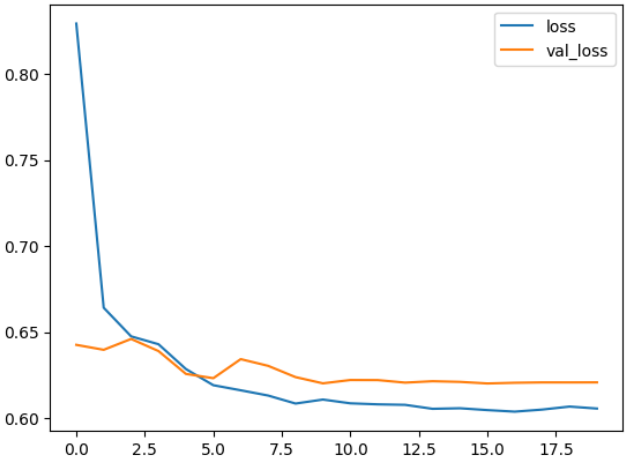

Figura 4-2.: Función de pérdida de la red neuronal convolucional durante el entrenamiento.

4.4. Transformers: DeBERTa-v3 y variaciones

Para obtener resultados más próximos a las evaluaciones humanas, se ha observado que simplemente incrementar la complejidad de las arquitecturas no es suficiente, en cambio, se ha optado por utilizar modelos preentrenados en volúmenes de datos gigantescos y que tienen arquitecturas más avanzadas que permiten obtener mejores resultados. De esta manera, dicho aprendizaje y el entendimiento del lenguaje adquirido se puede usar para la tarea de calificación automática. Esto se llevó a cabo específicamente descargando los modelos proporcionados por HuggingFace y que son de libre uso. Inicialmente, se seleccionó el modelo DeBERTa-v3-base como punto de partida, siendo el primero en ser ajustado y elegido por su rendimiento superior en comparación con otras arquitecturas basadas en transformers [61].

El ajuste del modelo se llevó a cabo primero en su versión base y luego en su versión *large*. Se realizó un cambio en el parámetro *max_length*, aumentándolo de 512 a 2048, lo cual resultó en mejoras observables en el rendimiento. Luego, se procedió a ajustar el modelo utilizando la metodología de validación cruzada con 5 pliegues o *folds* y así la predicción del conjunto de test es el promedio de las predicciones de los 5 *folds* en el conjunto.

4.4.1. Versión mejorada

Posteriormente, se exploraron diversas estrategias para mejorar el rendimiento del modelo deBERta. En primer lugar, se aplicaron múltiples técnicas de *pooling*, incluyendo *maxpooling*, *average pooling* y *attention pooling* [65], y luego se concatenaron con el objetivo de incorporar más información de los tokens en el modelo. En segundo lugar, se implementó la técnica de *dropout* multimuestra [63], definiendo 5 capas con *dropouts* de 0.1 a 0.5, donde se procesaba en paralelo y luego se promediaba para mejorar la generalización del modelo. Además, la

técnica de *Adversarial Weight Perturbations* también se incorporó para ayudar a prevenir el sobreajuste, siendo entrenada en las cinco últimas capas para mejorar la capacidad de generalización considerando el conjunto de datos reducido [66].

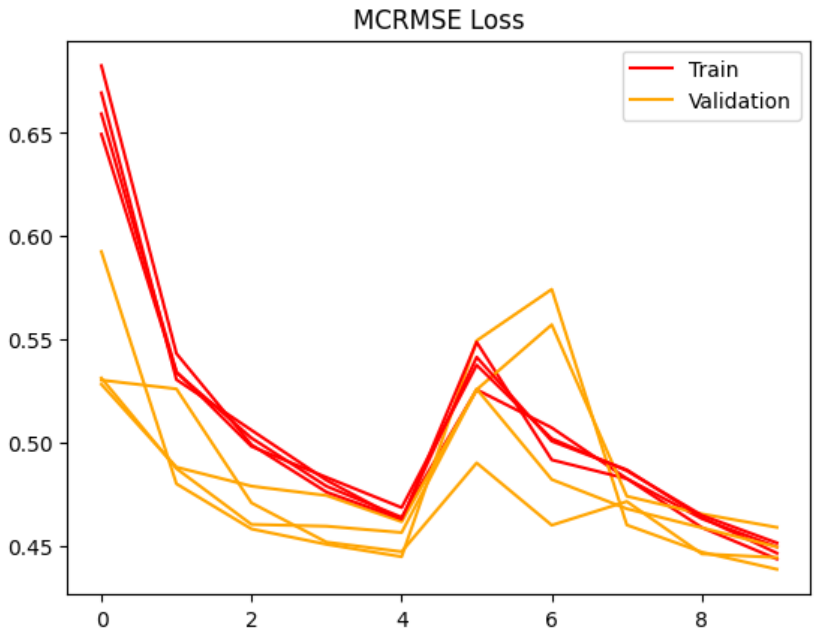

Figura 4-3.: Entrenamiento de los cinco modelos de DeBERTa-V3-Large mejorados.

En la Figura **4-3** se observan los resultados de la función de pérdida en el conjunto de entrenamiento y en el de validación de los 5 modelos DeBERTa-v3-large, que incorporan las mejoras mencionadas anteriormente. Estos fueron entrenados durante 10 épocas en cada uno de los conjuntos de entrenamiento de la partición en 5 pliegues. En ninguno de ellos se aprecia un sobreajuste significativo. Por el contrario, se observa un fenómeno en la época número 5 que deteriora los resultados, generando el pico que se aprecia en la gráfica. Este fenómeno se atribuye inicialmente a la técnica *Adversarial Weight Perturbation*, aunque se desconoce el origen exacto. Sin embargo, después de un par de épocas, la función de pérdida vuelve a disminuir tanto en el conjunto de entrenamiento como en el de validación.

Las predicciones de estos 5 modelos para el conjunto de prueba promediadas es lo que constituye la predicción final, que logró el mejor rendimiento de modelo individual presente en **4-1**. El detalle de cómo se mejoró aún más el rendimiento es presentado en la siguiente sección.

4.4.2. Propuesta final del método de calificación

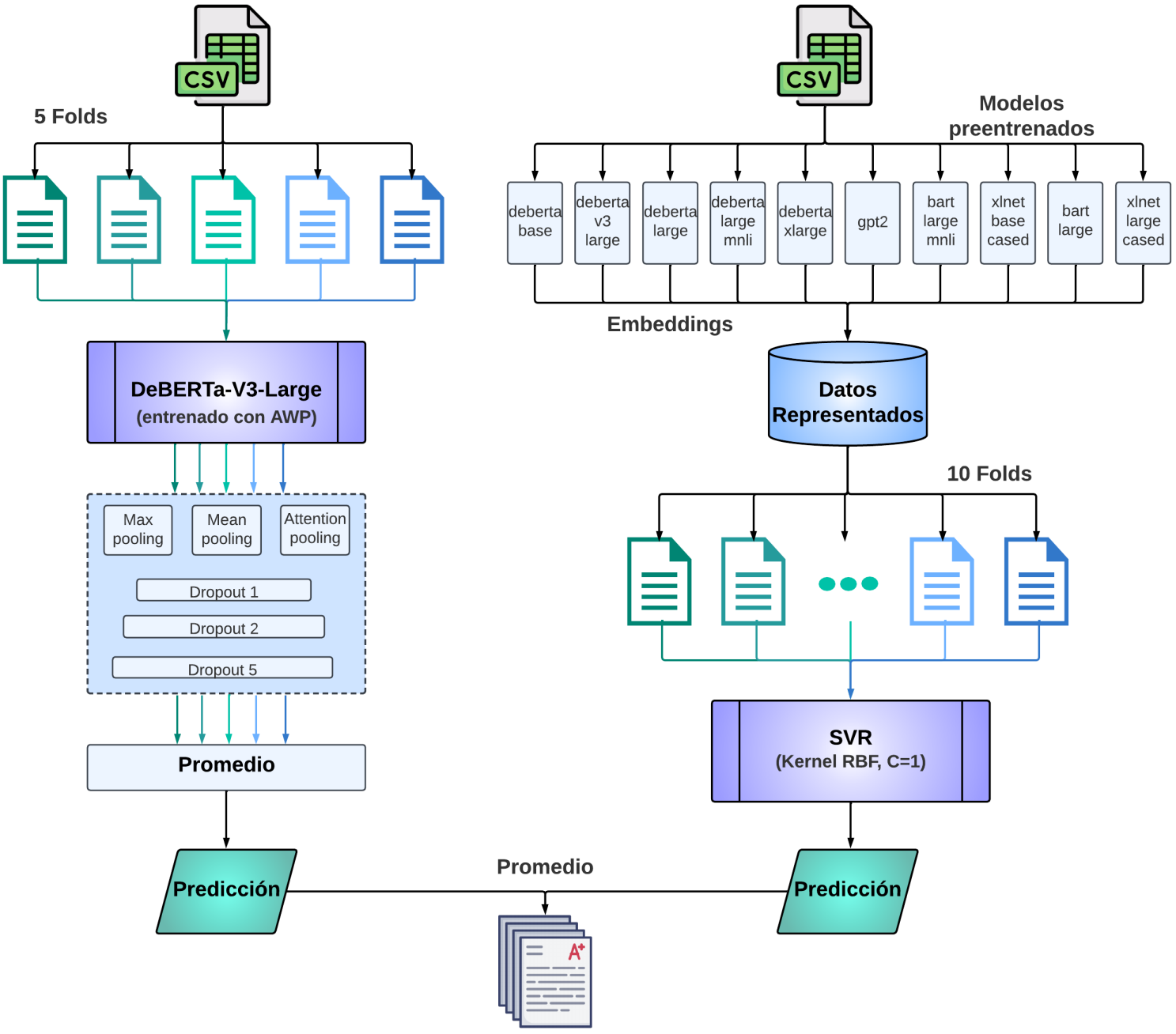

Figura 4-4.: Diagrama general del método de calificación propuesto.

La propuesta del método final de este trabajo, que logró el mejor rendimiento en todos los experimentos realizados y en todas los indicadores de los ensayos con un MCRMSE de 0.451, se compone de dos métodos fundamentales:

1. El modelo individual mejorado basado en DeBERTa-v3-large, descrito en detalle en la sección anterior, que fue entrenado en los 5 pliegues del conjunto de entrenamiento y las predicciones provienen del promedio de la predicción de cada uno de esos modelos.

2. Un modelo adicional, desarrollado en paralelo, en el que se tomó el conjunto de datos y se representó utilizando los *embeddings* de 10 modelos preentrenados (xlnet-large-cased, xlnet-base-cased, Deberta-Large, DeBERTa-large-mnli, Bart-large, Bart large mnli, deBERTa-xLarge, GPT2, deBERTa-v3-base, deBERTa-v3-large). Vale anotar que ninguno de estos modelos fue entrenado con los ensayos del conjunto de datos, en cambio, se aplicó *transfer learning*, obteniendo así sus representaciones vectoriales usando los parámetros fijos de estos *transformers*. Estas representaciones se concatenaron por columnas en un dataframe resultante, que contiene un amplio conjunto de características. Posteriormente, se entrenó un modelo de regresión SVR con todas estas características a lo largo de 12 pliegues, y se realizaron predicciones sobre las notas del conjunto de prueba, obteniendo un destacado resultado de 0.457, superando, por ejemplo, las primeras implementaciones de deBERTa presentadas.

Finalmente, para mejorar aún más la métrica, se promediaron las predicciones obtenidas de las dos partes descritas anteriormente. Este proceso integral constituye el método sugerido para la calificación automática de ensayos y representa el resultado más concreto y valioso de este trabajo. Este método se puede apreciar visualmente en la Figura **4-4**, donde la parte izquierda representa el mejor modelo individual entrenado y la derecha representa el modelo SVR que usa el ensamble de los *embeddings* de modelos preentrenados.

4.5. Comparativa general

Modelo	Características	RMSE						MCRMSE
		Cohesión	Sintaxis	Vocabulario	Fraseología	Gramática	Convenciones	
Reg Lineal	TfidfVectorizer	0,753	0,709	0,627	0,707	0,807	0,749	0,725
Reg Lineal	Custom	0,763	0,791	0,551	0,642	0,670	0,776	0,699
XGBoost	Custom	0,673	0,639	0,599	0,690	0,738	0,713	0,675
SVR	Custom	0,654	0,644	0,557	0,645	0,740	0,751	0,665
LGBM	Custom	0,647	0,635	0,569	0,670	0,702	0,685	0,651
Red Convolucional (CNN)	Bag of words	0,606	0,614	0,537	0,616	0,677	0,652	0,617
SVR	Glove	0,605	0,600	0,530	0,604	0,650	0,621	0,602
XGBoost	TfidfVectorizer	0,580	0,590	0,506	0,564	0,629	0,618	0,581
SVR	Word2Vec	0,584	0,576	0,508	0,564	0,605	0,601	0,573
LGBM	Custom CV	0,566	0,556	0,503	0,583	0,645	0,582	0,572
LGBM	TfidfVectorizer	0,551	0,552	0,492	0,547	0,601	0,578	0,553
SVR	TfidfVectorizer	0,554	0,546	0,491	0,533	0,589	0,568	0,547
LGBM	TfidfVectorizer CV	0,545	0,546	0,484	0,539	0,584	0,575	0,546
Red Neuronal (ANN)	Bag of words	0,550	0,533	0,482	0,534	0,585	0,573	0,543
DeBERTa-V3-Base	autotokenizer	0,518	0,501	0,458	0,496	0,472	0,469	0,486
DeBERTa-V3-Large	autotokenizer	0,487	0,463	0,444	0,470	0,465	0,453	0,463
DeBERTa-V3-Large	autotokenizer CV	0,481	0,463	0,436	0,470	0,466	0,454	0,462
DeBERTa-V3-L.Mejorado	autotokenizer	0,491	0,476	0,431	0,464	0,479	0,472	0,457
SVR	10 Embeddings CV	0,478	0,453	0,426	0,458	0,468	0,454	0,456
DeBERTa-V3-L.Mejorado	**autotokenizer CV**	**0,479**	**0,451**	**0,427**	**0,457**	**0,466**	**0,445**	**0,454**
Método Propuesto	**10 embdd.+autotok. CV**	**0,473**	**0,448**	**0,422**	**0,453**	**0,462**	**0,445**	**0,451**

Tabla 4-1.: Estadísticas de desempeño para los modelos ajustados.

En la Tabla **4-1**, se proporciona un resumen de los errores obtenidos en la calificación de los seis indicadores de forma individual y el promedio de errores para cada uno de los modelos implementados con sus respectivas variaciones y características empleadas en cada caso. La tabla se encuentra ordenada de mayor a menor MCRMSE. Cabe aclarar que las características *custom* son las extraídas manualmente en la sección 5.2.1 y que la abreviatura CV corresponde a validación cruzada.

El modelo individual que mejor desempeño obtuvo en todos los indicadores, así como en la métrica general, fue DeBERTa-v3-large entrenado en 5 pliegues y mejorado con las técnicas mencionadas anteriormente. El método propuesto, descrito en la sección 6.4.2 e ilustrado en la Figura **4-4**, obtuvo el mejor desempeño en la predicción de todos los indicadores y solo está empatado en el caso de *convenciones* con el modelo individual mejorado basado en DeBERTa v3-large.

Por otra parte, en todos los casos se evidencia un rendimiento superior en la predicción de la calificación de *vocabulario*, mientras que los modelos enfrentan mayores desafíos al prever los indicadores de *cohesión* y *gramática*. En particular, se destaca que los modelos con mejor rendimiento presentan un mayor error en el indicador de cohesión, mientras que aquellos con menor rendimiento se ven más afectados por los indicadores de *gramática* y *convenciones*.

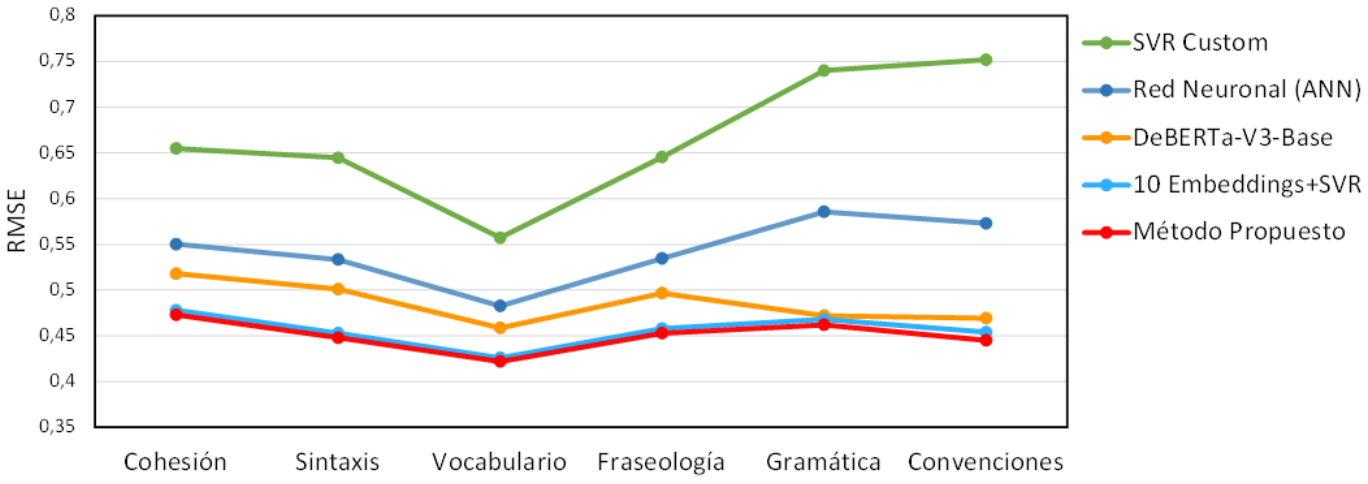

Figura 4-5.: Resultados gráficos de 5 modelos presentados en los 6 indicadores.

Después de realizar pruebas exhaustivas, siguiendo los enfoques sugeridos en la literatura, se observan mejoras en el rendimiento, aunque estas se vuelven cada vez más sutiles. Un ejemplo de esto se puede apreciar en la Figura **4-5**, donde se comparan los errores de predicción en los 6 indicadores para 5 de los modelos implementados. Se nota que los últimos modelos exhiben valores cercanos entre sí. Este patrón sugiere la presencia de una cota de error mínimo,

posiblemente atribuible a cierto grado de aleatoriedad en las evaluaciones. Utilizando el 80 % de los datos de entrenamiento oficiales esta cota puede estar un poco por debajo del 0.45, y según los resultados del concurso quizás con el 100 % de los datos de entrenamiento se puede llegar a 0.43. La implicación clave de este hallazgo es que, para lograr mejoras sustanciales adicionales, podría ser necesario recurrir a conjuntos de textos más extensos para entrenar los modelos. Esto permitiría que los modelos profundos capturen relaciones más complejas entre el lenguaje utilizado y puedan acercarse más las evaluaciones realizadas por humanos.

5. Conclusiones y recomendaciones

5.1. Conclusiones

La calificación automática es una tarea que se puede hacer razonablemente bien con las tecnologías disponibles actualmente. Los mejores desempeños obtenidos llegan a predecir las 6 notas calificadas con un error de solo 4.5 décimas en promedio sobre todas las notas de manera simultánea, lo cuál es un error pequeño considerando la escala de calificación de 5 unidades, ya que un valor de esa magnitud incluso podría estar entre dos calificaciones de evaluadores humanos al mismo ensayo o incluso de un mismo evaluador a un mismo ensayo en momentos o circunstancias diferentes.

Se observa un mejor desempeño para medir *vocabulario* en general, esto puede deberse a que la manera en la que percibimos el vocabulario de un ensayo está más relacionada con características del texto que se pueden extraer e interpretar más fácilmente como conteos de palabras distintas, número de errores, etc. o porque el vocabulario es más identificable o calificable gracias a los gigantes corpus de entrenamiento de los modelos preentrenados empleados. Por el contrario, el indicador de *gramática* fue el más difícil de predecir para los modelos de regresión junto con *convenciones*, mientras que la *cohesión* fue uno de los que resultó más difícil de predecir para los grandes modelos preentrenados además de *gramática* que también resultó difícil en ese caso; esto puede deberse en parte a que son aspectos complicados de la lengua y que probablemente estudiantes que estén aprendiendo inglés tengan más variabilidad en sus resultados y se necesite una muestra más grande para poder entender patrones que expliquen.

Por otro lado, se puede apreciar que las relaciones entre las distintas características que se pueden extraer manualmente del texto y las notas no son evidentes y, por tanto, los modelos lineales convencionales no las pueden capturar y se desempeñan peor en la tarea de calificar. Por consiguiente, es conveniente usar una técnica de representación vectorial. Sin embargo, estos modelos con características manuales tienen la ventaja de ser interpretables y más simples en términos de la cantidad de parámetros, lo cual puede ser útil en una amplia gama de contextos. En el caso de los modelos de redes neuronales construidos desde cero, presentan un rendimiento razonable, pero se corre el riesgo de sobreajuste en solo unas pocas épocas, debido principalmente al pequeño tamaño del conjunto de datos.

La recomendación para calificación automática de cualquier conjunto es utilizar modelos preentrenados en corpus enormes, como es el caso de *DeBERTa-v3*, ya que estos modelos han demostrado tener un rendimiento superior en la mayoría de tareas de PLN en la actualidad. Pese a que estos grandes modelos de lenguaje han sido entrenados pensando en otras tareas y con un corpus de datos enorme que no es sobre ensayos específicamente, se puede observar que incluso sin entrenar con los datos particulares, sino solo utilizando los *embeddings* de distintos modelos como características y luego ajustando un modelo de regresión (SVR) se pueden obtener resultados muy buenos. Sin embargo, para lograr los mejores resultados con un modelo individual, se debe hacer *fine-tunning*, es decir, debe entrenar parámetros de las últimas capas para adaptarse mejor a los datos de interés. Esto se evidencia con el hecho de la versión mejorada de DeBERTa-v3-large fue el mejor que todos los demás modelos individuales en cada uno de los 6 indicadores. Su desempeño solo pudo ser superado por el ensamble propuesto como método final del cual DeBERTa también forma parte.

Además, es importante señalar que estrategias como las muestras múltiples en el *dropout* o como usar distintas técnicas de *pooling* o *adversarial Weight Perturbations* parecen surtir efecto en mejorar el desempeño de deBERTa v3 en su versión Base y en su versión Large, pero estas mejoras cada vez son más pequeñas. Se tiene como hipótesis que hay una asíntota de la que no se puede bajar sin sobreajustar los datos o sin agrandar el conjunto de datos de entrenamiento debido a que hay un error inherente en las calificaciones que incluso se podría modelar de manera probabilística.

Por último, en aras de lograr un desempeño aún mayor, se puede realizar *fine-tuning* a varios modelos y ensamblarlos. En general, en la literatura y en los repositorios explorados, se encuentran muchos ensambles de modelos que alcanzan buenos resultados como [64]. Se observa que, en general, ensamblar modelos mejora el desempeño, pero incrementa bastante el coste computacional y no es algo que cualquiera pueda reproducir fácilmente [67] o que necesariamente vaya a mejorar el desempeño.

5.2. Recomendaciones

Para el lector interesado en reproducir los códigos disponibles en el repositorio o para realizar otro tipo de problemas con grandes modelos de lenguaje preentrenados, se sugiere utilizar métodos que aprovechen la capacidad de GPU y TPU de las que disponen los entornos de ejecución (por ejemplo, los de Google Colab) o la plataforma que se esté usando para una mayor eficiencia en términos computacionales. En el caso de Google Colab Pro, la opción de un entorno de ejecución de tipo A100 con GPU y RAM aceleradas es muy buena para la mayoría de necesidades computacionales del presente trabajo.

De igual manera, se debe tener en cuenta que incluso un modelo simple como el perceptrón

multicapa ajustado en la metodología puede sobreajustar fácilmente los datos como se aprecia en las figuras **4-1** y **4-2**, esto debido al gran número de parámetros de los modelos y a que la base de datos con la que se está trabajando no es muy grande. Por tanto, eventualmente termina por aprendérselos y no generalizar bien. Entonces hay que buscar activamente estrategias contra el sobreajuste, sobre todo en el caso de querer entrenar una arquitectura de cero utilizando conjuntos de datos pequeños. No obstante, si se tiene la posibilidad de acceder a un modelos preentrenados, posiblemente los resultados encontrados van a ser mejores debido a toda la información de base que tienen estos modelos para cumplir cualquier tarea.

Resultaría de gran interés aplicar esta metodología a un conjunto de ensayos en español u otros idiomas y ver qué resultados se obtienen, ya que no se ha trabajado tanto en este campo y existen grandes modelos preentrenados en corpus de texto multilingües. De la misma manera, se podría probar cómo es el desempeño de un método como el propuesto en el presente trabajo en un conjunto de ensayos más especializado en un tema específico o de otro nivel de formación.

Finalmente, en el caso de querer mejorar los resultados presentados en el presente documento, se sugiere enfáticamente revisar la idea de ensamblar distintos modelos entrenados adicionales y luego promediar las predicciones o incluso utilizar un metamodelo para que decida cómo ponderar las predicciones de los distintos modelos en el ensamble. Dicho metamodelo también podría efectuar un análisis riguroso de un gran número de características extraídas manualmente a ver cuáles ser podrían incluir además de los embeddings para contribuir a mejorar el desempeño en la calificación. Además, estrategias como el aumento sintético de datos, como predecir *pseudolabels* para entrenamiento a los conjuntos de ensayos de las competencias anteriores (lo cual estaba permitido en el concurso *Feedback3.0*) o usar las predicciones *Out of fold (OOF)* como características adicionales, así como tener en cuenta los descriptores de las notas A, podrían conducir a una calificación automática aún más cercana a la otorgada por humanos.

A. Anexo: Criterios que siguen los calificadores humanos

	HOLISTIC	ANALYTIC					
	Overall	**Cohesion**	**Syntax**	**Vocabulary**	**Phraseology**	**Grammar**	**Conventions**
5	Native-like facility in the use of language with syntactic variety, Appropriate word choice and phrases; well-controlled text organization; precise use of grammar and conventions; rare language inaccuracies that do not impede communication.	Text organization consistently well controlled using a variety of effective linguistic features such as reference and transitional words and phrases to connect ideas across sentences and paragraphs; appropriate overlap of ideas.	Flexible and effective use of a full range of syntactic structures including simple, compound, and complex sentences; There may be rare minor and negligible errors in sentence formation.	Wide range of vocabulary flexibly and effectively used to convey precise meanings; skillful use of topic-related terms and less common words; rare negligible inaccuracies in word use.	Flexible and effective use of a variety of phrases, such as idioms, collocations, and lexical bundles, to convey precise and subtle meanings; rare minor inaccuracies that are negligible.	Command of grammar and usage with few or no errors	Consistent use of appropriate conventions to convey meaning; spelling, capitalization, and punctuation errors nonexistent or negligible.
4	Facility in the use of language with syntactic variety and range of words and phrases; controlled organization; accuracy in grammar and conventions; occasional language inaccuracies that rarely impede communication.	Organization generally well controlled; a range of cohesive devices used appropriately such as reference and transitional words and phrases to connect ideas; generally appropriate overlap of ideas.	Appropriate use of a variety of syntactic structures, such as simple, compound, and complex sentences; occasional errors or inappropriateness in sentence formation.	Sufficient range of vocabulary to allow flexibility and precision; appropriate use of topic-related terms and less common lexical items	Appropriate use of a variety of phrases, such as idioms, collocations, and lexical bundles; occasional inaccuracies and colloquialisms.	Minimal errors in grammar and usage	Generally consistent use of appropriate conventions to convey meaning; spelling, capitalization, and punctuation errors few and not distracting.
3	Facility limited to the use of common structures and generic vocabulary; organization generally controlled although connection sometimes absent or unsuccessful; errors in grammar and syntax and usage. Communication is impeded by language inaccuracies in some cases.	Organization generally controlled; cohesive devices used but limited in type; Some repetitive, mechanical, or faulty use of cohesion use within and/or between sentences and paragraphs.	Simple, compound, and complex syntactic structures present although the range may be limited; some apparent errors in sentence formation, especially in more complex sentences.	Minimally adequate range of vocabulary for the topic; no precise use of subtle word meanings; topic related terms only used occasionally; attempts to use less common vocabulary but with some inaccuracy	Evident use of phrases such as idioms, collocations, and lexical bundles but without much variety; some noticeable repetitions and misuses.	Some errors in grammar and usage	Developing use of conventions to convey meaning; errors in spelling, capitalization, and punctuation that are sometimes distracting.
2	Inconsistent facility in sentence formation, word choice, and mechanics; organization partially developed but may be missing or unsuccessful. Communication impeded in many instances by language inaccuracies.	Organization only partially developed with a lack of logical sequencing of ideas; some basic cohesive devices used but with inaccuracy or repetition.	Some sentence variation used; many sentence structure problems.	Narrow range of vocabulary to convey basic and elementary meaning; topic related terms used inappropriately; errors in word formation and word choice that may distort meanings	Narrow range of phrases, such as collocations and lexical bundles, used to convey basic and elementary meaning; many repetitions and /or misuses of phrases.	Many errors in grammar and usage.	Variable use of conventions; spelling, capitalization, and punctuation errors frequent and distracting.
1	A limited range of familiar words or phrases loosely strung together; frequent errors in grammar (including syntax) and usage. Communication impeded in most cases by language inaccuracies.	No clear control of organization; cohesive devices not present or unsuccessfully used; presentation of ideas unclear.	Pervasive and basic errors in sentence structure and word order that cause confusion; basic sentences errors common.	Limited vocabulary often inappropriately used; limited control of word choice and word forms; little attempt to use topic-related terms	Memorized chunks of language, or simple phrasal patterns predominate; many repetitions and misuses of phrases.	Errors in grammar and usage throughout.	Minimal use of conventions; spelling, capitalizatio n, and punctuation errors throughout.

Figura A-1.: Rúbrica de puntuación del dominio del inglés para estudiantes del idioma inglés - (English Proficiency Scoring Rubric for English Language Learners).

B. Repositorios Consultados

- `https://www.kaggle.com/code/jonbown/feedback-prize-feature-engineering`

  Este cuaderno efectúa ingeniería de características y es útil en el caso de querer ajustar un modelo simple y en caso de querer interpretabilidad.

- `https://www.kaggle.com/code/jonbown/feedback-prize-candidate-model-exploration/notebook`

  Una exhaustiva exploración de métodos de regresión, desde regresión lineal hasta métodos basados en árboles como XGBoost.

- `https://www.kaggle.com/code/shreydan/deberta-v3-base-accelerate-finetuning`

  Implementación base de deBERTa v3 usando la librería pytorch de python y aprovechando un acelerador por GPU.

- `https://github.com/rohitsingh02/kaggle-feedback-english-language-learning-1st-place-so`

  Los ganadores del concurso Feedback 3.0, el equipo realiza un ensamble de cerca de 20 modelos. Proporcionan múltiples ideas que sirvieron para mejorar el desempeño de la versión base de deberta-V3-Large.

- `https://www.kaggle.com/code/cdeotte/rapids-svr-cv-0-450-lb-0-44x/notebook`

  Este cuaderno muestra cómo lograr desempeños excelentes sin entrenar ningún parámetro sino con los modelos preentrenados disponibles en HuggingFace para calcular distintos embeddings de los mismos textos y luego pegar esas variables para aplicar un SVR optimizado para GPU. Fundamental para la estrategia final propuesta.

Bibliografía

[1] P. Kline, *The New Psychometrics: Science, Psychology and Measurement.* Routledge, 1 ed., 1999.

[2] T. N. Fitria, "Artificial intelligence (AI) technology in OpenAI ChatGPT application: A review of ChatGPT in writing English essay.," *ELT Forum: Journal of English Language Teaching,*, vol. 12, no. 1, pp. 44–58, 2023.

[3] E. B. Page, "Grading Essays by Computer: Progress Report. Proceedings of the 1966 Invitational Conference on Testing Problems.," *Princeton, N.J. Educational Testing Service*, pp. 87–100, 1967.

[4] E. Page, "The use of the computer in analyzing student essays," *Int Rev Educ*, pp. 210–225, 1968.

[5] K. L. Gwet, "Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters," *Advanced Analytics, LLC*, 2014.

[6] Mark D. S., "Contrasting State-of-the-Art in the Machine Scoring of Short-Form Constructed Responses," *Educational Assessment*, vol. 20, no. 1, pp. 46–65, 2015.

[7] Alex Franklin, Natalie Rambis, Maggie Meg Benner, Perpetual Baffour, Ryan Holbrook, and u. Scott Crossley, "Feedback Prize - English Language Learning. Kaggle. ," 2022.

[8] S. A. Crossley, K. Kyle, and D. S. Mcnamara, "To Aggregate or Not? Linguistic Features in Automatic Essay Scoring and Feedback Systems," *Grantee Submission*, vol. 8, no. 1, 2015.

[9] C. Ramineni and D. M. Williamson, "Automated essay scoring: Psychometric guidelines and practices," *Assessing Writing*, pp. 25–39, 2013.

[10] S. P. Balfour, "Assessing Writing in MOOCs: Automated Essay Scoring and Calibrated PeerReview™," *Research & Practice in Assessment*, pp. 40–48, 2013.

[11] S. Cushing Weigle, "Validation of automated scores of TOEFL iBT tasks against non-test indicators of writing ability," *Language Testing*, vol. 27, no. 3, pp. 335–353, 2010.

[12] K. Taghipour, *Robust trait-specific essay scoring using neural networks and density estimators.* PhD thesis, National University of Singapore, Singapore, 2017.

[13] H. Shi and V. Aryadoust, "Correction to: A systematic review of automated writing evaluation systems," *Education and Information Technologies*, vol. 28, pp. 6189–6190, 5 2023.

[14] P. C. Jackson, *Toward human-level artificial intelligence: Representation and computation of meaning in natural language.* Dover Publications, 11 2019.

[15] E. Mayfield and C. P. Rosé, "LightSIDE," in *Handbook of Automated Essay Evaluation*, Routledge, 1 ed., 2013.

[16] M. Shermis and J. Burstein, *Handbook of Automated Essay Evaluation: Current Applications and New Directions.* Routledge, 1 ed., 2013.

[17] S. Burrows, I. Gurevych, and B. Stein, "The Eras and Trends of Automatic Short Answer Grading," *Int J Artif Intell Educ*, vol. 25, pp. 60–117, 2015.

[18] K. Zupanc and Z. Bosni´c, "Automated essay evaluation with semantic analysis," *Elsevier*, vol. 120, pp. 118–132, 2017.

[19] D. Yan, A. A. Rupp, and P. W. Foltz, *Handbook of Automated Scoring; Theory into Practice.* Chapman and Hall/CRC., 1 ed., 2020.

[20] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review," *Information and Software Technology*, vol. 51, no. 1, pp. 7–15, 2009.

[21] Y.-Y. Chen, C.-L. Liu, C.-H. Lee, and T.-H. Chang, "An unsupervised automated essay-scoring system," *IEEE Intelligent Systems*, vol. 25, no. 5, pp. 61–67, 2010.

[22] Y. Wang, Z. Wei, Y. Zhou, and X. Huang, "Automatic essay scoring incorporating rating schema via reinforcement learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pp. 791–797, 2018.

[23] C. Lu and M. Cutumisu, "Integrating Deep Learning into An Automated Feedback Generation System for Automated Essay Scoring," *International Educational Data Mining Society*, 2021.

[24] K. S. McCarthy, R. D. Roscoe, L. K. Allen, A. D. Likens, and D. S. McNamara, "Automated writing evaluation: Does spelling and grammar feedback support high-quality writing and revision?," *Assessing Writing*, vol. 52, 4 2022.

[25] A. Sharma and D. B. Jayagopi, "Automated grading of handwritten essays," in *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*, vol. 2018-August, pp. 279–284, 2018.

[26] A. Vaswani, G. Brain, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Neural Information Processing Systems*, 2017.

[27] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet::Similarity-Measuring the Relatedness of Concepts," *AAAI*, vol. 4, pp. 25–29, 7 2004.

[28] F. Dong and Y. Zhang, "Automatic Features for Essay Scoring – An Empirical Study. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing," *Association for Computational Linguistics.*, pp. 1072–1077, 11 2016.

[29] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA*, 2019.

[30] E. Mayfield and A. W. Black, "Should You Fine-Tune BERT for Automated Essay Scoring?," *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications. Association for Computational Linguistics*, pp. 151–162, 7 2020.

[31] D. Ramesh and S. K. Sanampudi, "An automated essay scoring systems: a systematic literature review," *Artificial Intelligence Review*, vol. 55, pp. 2495–2527, 3 2022.

[32] H. Yannakoudakis and R. Cummins, "Evaluating the performance of automated text scoring systems," *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 213–223, 2015.

[33] R. Bhatt, M. Patel, G. Srivastava, and V. Mago, "A Graph Based Approach to Automate Essay Evaluation," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 2020-Octob, pp. 4379–4385, 2020.

[34] Z. Ke and V. Ng, "Automated essay scoring: A survey of the state of the art," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2019-Augus, pp. 6300–6308, 2019.

[35] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[36] D. Castro-Castro, R. Lannes-Losada, M. Maritxalar, I. Niebla, C. Pérez-Marqués, N. Álamo-Suárez, and A. Pons-Porrata, "A multilingual application for automated essay scoring," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5290 LNAI, pp. 243–251, 2008.

[37] P. U. Rodriguez, A. Jafari, and C. M. Ormerod, "Language models and Automated Essay Scoring," *ArXiv*, 9 2019.

[38] S. Ghannay, B. Favre, Y. Estève, and N. Camelin, "Word Embeddings Evaluation and Combination," *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pp. 300–305, 5 2016.

[39] M. Mars, "From Word Embeddings to Pre-Trained Language Models: A State-of-the-Art Walkthrough," *Applied Sciences (Switzerland)*, vol. 12, 9 2022.

[40] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International journal of machine learning and cybernetics*, vol. 1, pp. 43–52, 12 2010.

[41] K. W. CHURCH, "Word2Vec," *Natural Language Engineering*, vol. 23, pp. 155–162, 1 2017.

[42] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1535–1543, 2014.

[43] V. Kumar and B. Subba, "A TfidfVectorizer and SVM based sentiment analysis framework for text data corpus," in *2020 National Conference on Communications (NCC)*, pp. 1–6, IEEE, 2 2020.

[44] Microsoft, "GitHub - Microsoft/LightGBM:Light Gradient Boosting Machine."

[45] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," *31st Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 3149–3157, 2017.

[46] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 785–794, Association for Computing Machinery, 8 2016.

[47] J.J.Espinosa-Zúñiga, "Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito," *Ingeniería Investigación y Tecnología*, vol. 21, no. 3, pp. 1–16, 2020.

[48] C. Cortes, V. Vapnik, and L. Saitta, "Support-Vector Networks," *Machine Leaming*, vol. 20, pp. 273–297, 1995.

[49] M. Awad, R. Khanna, M. Awad, and R. Khanna, *Support vector regression. Efficient learning machines: Theories, concepts, and applications for engineers and system designers.* Apress, 2015.

[50] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer Perceptron and Neural Networks," *WSEAS Transactions on Circuits and Systems*, vol. 8, no. 7, pp. 579–588, 2009.

[51] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," *ArXiv*, 5 2020.

[52] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. Von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-Art Natural Language Processing," *Association for Computational Linguistics*, pp. 38–45, 2020.

[53] P. He, X. Liu, J. Gao, W. Chen, and M. Dynamics, "DEBERTA: DECODING-ENHANCED BERT WITH DIS-ENTANGLED ATTENTION," *Conference paper at ICLR*, 2021.

[54] P. Zhang, "Longformer-based Automated Writing Assessment for English Language Learners Stanford CS224N Custom Project," tech. rep., Standford, 2023.

[55] K. K. Y. Chan, T. Bond, and Z. Yan, "Application of an Automated Essay Scoring engine to English writing assessment using Many-Facet Rasch Measurement," *Language Testing*, vol. 40, pp. 61–85, 1 2023.

[56] A. Mizumoto and M. Eguchi, "Exploring the potential of using an AI language model for automated essay scoring," *Research Methods in Applied Linguistics*, vol. 2, 8 2023.

[57] V. Mohan, M. J. Ilamathi, and M. Nithya, "Preprocessing Techniques for Text Mining-An Overview," *International Journal of Computer Science & Communication Networks,* vol. 5, no. 1, pp. 7–16, 2015.

[58] M. Siino, I. Tinnirello, and M. La Cascia, "Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers," *Information Systems*, vol. 121, p. 102342, 3 2024.

[59] S. A. Crossley, D. B. Allen, and J. S. Danielle McNamara, "Text readability and intuitive simplification: A comparison of readability formulas," *Reading in a foreign language*, vol. 23, no. 1, pp. 84–101, 2011.

[60] F. Scarselli and A. C. Tsoi, "Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results," *Neural Networks*, vol. 11, no. 1, pp. 15–37, 1998.

[61] P. He, J. Gao, and W. Chen, "DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing," *ArXiv*, 11 2021.

[62] D. Wu, S.-t. Xia, and Y. Wang, "Adversarial Weight Perturbation Helps Robust Generalization," *ArXiv*, 4 2020.

[63] H. Inoue, "Multi-Sample Dropout for Accelerated Training and Better Generalization," *ArXiv*, 5 2019.

[64] A. Stanciu, I. Cristescu, E. M. Ciuperca, and C. E. Cîrnu, "USING AN ENSEMBLE OF TRANSFORMER-BASED MODELS FOR AUTOMATED WRITING EVALUATION OF ESSAYS," in *14th International Conference on Education and New Learning Technologies*, (Palma, Spain), pp. 5276–5282, IATED, 7 2022.

[65] H. Zhang, Y. Gong, Y. Shen, W. Li, J. Lv, N. Duan, and W. Chen, "Poolingformer: Long Document Modeling with Pooling Attention," *ArXiv*, 2021.

[66] A. Aziz, M. Akram Hossain, and A. Nowshed Chy, "CSECU-DSG at SemEval-2023 Task 4: Fine-tuning DeBERTa Transformer Model with Cross-fold Training and Multi-sample Dropout for Human Values Identification," tech. rep., Department of Computer Science and Engineering University of Chittagong, Chattogram, Bangladesh, 2023.

[67] E. Mayfield and A. W. Black, "Should You Fine-Tune BERT for Automated Essay Scoring?," *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications. Association for Computational Linguistics*, pp. 151–162, 7 2020.