TEC | Tecnológico de Costa Rica

Maestría en Computación, énfasis en Ciencias de la Computación

Escuela de Ingeniería en Computación

Estimación de Incertidumbre para la Detección de Texto Complejo en Español

Autor:

Miguel Guillermo Abreu Cárdenas

Tutor:

Saúl Calderón Ramírez

San José, Costa Rica

16 de diciembre de 2024

Resumen

En la actualidad, la simplificación de textos, que implica la transformación de textos para mejorar su legibilidad y comprensibilidad para públicos específicos, es un área de creciente interés. Este proceso es crucial para aumentar la inclusividad, especialmente para personas con baja escolaridad o con discapacidades visuales/auditivas. Aunque los avances recientes en el campo, especialmente con los Modelos del Lenguaje de Gran Tamaño*(Large Lenguage Models)*, han mejorado las técnicas de simplificación de texto basadas en máquinas, estos modelos a menudo requieren un uso intensivo de recursos y la mayoría se encuentra en manos de empresas privadas por lo que su utilización en fase de inferencia puede llegar a ser bastante costoso. Por ello, un modelo que pueda clasificar eficientemente los segmentos de texto que necesitan ser simplificados puede optimizar el uso de recursos y evitar la sobrecarga innecesaria.

En este contexto, la categorización precisa de los textos en términos de su complejidad —simples o complejos— se vuelve esencial. Sin embargo, esta tarea no está exenta de desafíos, como los falsos positivos o negativos que pueden surgir de un modelo inadecuadamente ajustado. Una estrategia para manejar estos desafíos es la implementación de un puntaje de incertidumbre para cada predicción, permitiendo así tomar decisiones más informadas sobre qué textos requieren simplificación. Esta investigación se enfoca en la exploración de diversos enfoques de Estimación de Incertidumbre para la identificación de textos complejos en español, un área que no ha sido explorada hasta ahora. Nuestro objetivo es no solo definir y clasificar la complejidad del texto, sino también minimizar la incertidumbre asociada con estas clasificaciones, mejorando así la eficiencia y efectividad de los procesos de simplificación de texto.

Palabras Claves: Simplificación de Textos, Estimación de Incertidumbre, Modelos del Lenguaje de Gran Tamaño, Clasificación de Oraciones.

Abstract

Currently, text simplification, which involves transforming texts to improve their readability and comprehensibility for specific audiences, is an area of growing interest. This process is crucial for enhancing inclusivity, especially for individuals with low levels of education or visual/hearing impairments. Although recent advances in the field, particularly with Large Language Models, have improved machine-based text simplification techniques, these models often require intensive resources, and most are controlled by private companies, making their inference phase usage quite costly. Therefore, a model capable of efficiently classifying text segments that need simplification can optimize resource usage and prevent unnecessary overload.

In this context, the precise categorization of texts in terms of their complexity—simple or complex—becomes essential. However, this task is not without challenges, such as false positives or negatives that may arise from an inadequately tuned model. One strategy to address these challenges is the implementation of an uncertainty score for each prediction, allowing for more informed decisions about which texts require simplification. This research focuses on exploring various approaches to Uncertainty Estimation for identifying complex texts in Spanish, an area that has not yet been explored. Our goal is not only to define and classify text complexity but also to minimize the uncertainty associated with these classifications, thereby improving the efficiency and effectiveness of text simplification processes.

Keywords: Text Simplification, Uncertainty Estimation, Large Lenguage Models, Sentence Classification.

Acta de Aprobación de Tesis

Escuela de Ingeniería en Computación
Unidad de Posgrado

TEC | Tecnológico de Costa Rica

ACTA DE APROBACION DE TESIS

Estimación de Incertidumbre para la Detección de Texto Complejo en Español

Por: ABREU CARDENAS MIGUEL GUILLERMO

TRIBUNAL EXAMINADOR

Dr. Saúl Calderón Ramírez
Profesor Asesor

Dr. Luis Alexander Calvo Valverde
Profesor Lector

Dr. Angel Mario García Pedrero
Lector Externo

Dra.-Ing. Lilliana Sancho Chavarría
Presidente, Tribunal Evaluador Tesis
Programa Maestría en Computación

SINAES POST-GRADO ACREDITADO COSTA RICA

16 de diciembre, 2024

Lista de Publicaciones

Publicaciones en Conferencias con *Peer Review*

1. Abreu-Cardenas, M., Calderón-Ramírez, S., & Solís, M. (2023, November). Uncertainty Estimation for Complex Text Detection in Spanish. In 2023 IEEE 5th International Conference on BioInspired Processing (BIP) (pp. 1-6). IEEE.

Índice general

Índice de figuras

Índice de cuadros

1. Introducción

1.1. Antecedentes

En el presente trabajo, exploraremos diferentes métodos de Estimación de Incertidumbre(UE, en inglés) vinculados específicamente a la detección de textos complejos en español, con énfasis en su aplicación en la simplificación de textos. La Simplificación de Textos(TS, en inglés) es un campo crucial que busca mejorar la accesibilidad y comprensión de la información textual, especialmente para personas con dificultades de lectura o comprensión.

Con el avance vertiginoso de los modelos de lenguaje, como los Modelos Grandes del Lenguaje(LLM, en inglés), se ha incrementado la necesidad de automatizar el proceso de TS, aprovechando las capacidades excepcionales de estos modelos en tareas de Procesamiento del Lenguaje Natural(NLP, en inglés). Sin embargo, al enfocarnos en tareas específicas para idiomas como el español, nos enfrentamos a retos únicos. Entre estos, destaca la escasa disponibilidad de conjuntos de datos amplios y de alta calidad en español, lo cual limita significativamente la eficacia de los modelos diseñados para la simplificación de textos en este idioma. Este obstáculo se ve acentuado por los recursos de investigación más limitados en comparación con aquellos disponibles para el inglés, como se resalta en [9, 12]. Además, las fluctuaciones en cuanto a la calidad y uniformidad de los datos inciden negativamente en la capacidad de aprendizaje de los modelos [9, 12].

La UE en la TS adquiere una importancia especial en este contexto. Dado que los modelos de lenguaje automatizados, como los LLM´s, están cada vez más presentes en la generación y modificación de textos, es fundamental que estos sistemas puedan evaluar y comunicar el grado de confianza en sus propias predicciones. Esto no solo mejora la fiabilidad de los sistemas de NLP, sino que también proporciona a los usuarios finales una mejor comprensión de la precisión y los límites de los textos generados o modificados por la inteligencia artificial [13].

La implementación extensiva de LLM´s plantea cuestionamientos significativos

sobre su impacto ambiental y los costos asociados, destacando la urgencia de desarrollar enfoques más eficientes y sostenibles para la TS [11, 12]. En este contexto, la UE se presenta como una herramienta esencial para garantizar una implementación segura y eficaz de modelos, especialmente cuando se utiliza la tecnología LLM de manera selectiva. Al profundizar en este estudio, identificaremos las lagunas existentes en la investigación actual, con un enfoque particular en la falta de estudios específicos sobre la UE en la TS, subrayando la necesidad apremiante de abordar estas brechas de conocimiento [12].

1.1.1. Simplificación de Textos: Una Perspectiva General

La simplificación de textos TS es una técnica que busca reducir la complejidad de un texto, mejorando su legibilidad y comprensibilidad, mientras mantiene su significado léxico y sintáctico original [1]. En el contexto de la globalización y la diversidad lingüística, la capacidad de simplificar textos en múltiples idiomas, incluido el español, se convierte en una herramienta poderosa para superar las barreras del idioma y promover una mayor inclusión [14].

Existen diversas razones para simplificar un texto, desde brindar ayuda a personas con bajos niveles de alfabetización y a quienes están aprendiendo un nuevo idioma, hasta apoyar a personas que tienen dificultades para comprender el contexto de un texto determinado [1, 12].

A diferencia de otras técnicas de NLP, como el resumen de textos y la generación de paráfrasis, la TS va más allá de solo condensar o reestructurar la información; su objetivo principal es hacer que el contenido sea más accesible y fácil de entender. Esto implica un proceso cuidadoso de reescribir frases complejas en formas más simples, sustituir jergas o términos técnicos por palabras de uso común y desglosar estructuras sintácticas complejas en otras más claras y directas [1, 10]. Por ejemplo, en el Cuadro 1.1 se presenta una comparativa entre la TS y el resumen de textos con un ejemplo básico.

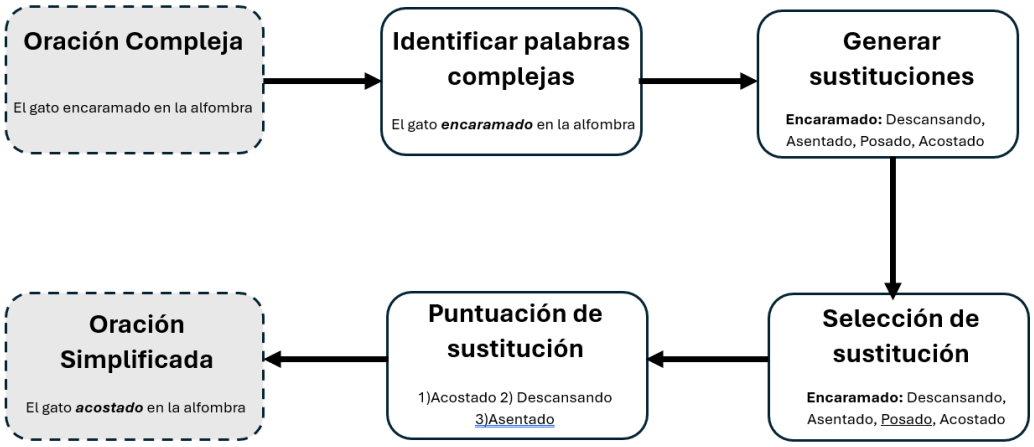

Figura 1.1: Proceso de detección de palabras complejas en un texto y la simplificación del mismo sustituyendo las palabras complejas por otras más simples. Tomado de [1].

Los avances recientes en la inteligencia artificial y los modelos de aprendizaje automático han facilitado el desarrollo de sistemas automatizados de simplificación de textos, aunque estos sistemas aún enfrentan desafíos en cuanto a la precisión, la naturalidad del lenguaje simplificado y la conservación del significado original. La investigación continua en este campo es vital para mejorar la eficacia de estas tecnologías y ampliar su aplicabilidad en diferentes contextos lingüísticos y culturales [15, 16].

En la literatura, se han desarrollado diversos enfoques para la simplificación automática de textos [17], incluyendo métodos basados en reglas, métodos estadísticos, redes neuronales y redes neuronales profundas [18]. Recientemente, los enfoques basados en redes neuronales profundas han incorporado LLM´s, que son arquitecturas tipo *transformer* preentrenadas con grandes conjuntos de datos. Por ejemplo, en [19] se detalla el proceso de ajuste de el Modelo Grande del Lenguaje de Meta AI(LLaMa, en inglés) para crear el Modelo Grande de Lenguaje de Meta AI para Simplificación Léxica(LSLLaMA, en inglés). Otro ejemplo se encuentra en [20].

Estos LLM´s han exhibido un rendimiento destacado en la simplificación de textos. No obstante, su uso extensivo plantea preocupaciones medioambientales debido a los considerables recursos necesarios tanto en el entrenamiento como en las pruebas. Además, algunos de estos LLM´s son de propiedad privada, como los Generative

Pre-trained Transformers (GPT) presentados por OpenAI en 2018 [21], lo que implica costos asociados a su utilización [11, 12, 22].

Texto Original	Texto Simplificado	Texto Resumido
"La fotosíntesis es un proceso complejo por el cual las plantas, algas y algunas bacterias transforman la luz solar en energía química, utilizando dióxido de carbono y agua."	"La fotosíntesis es cuando las plantas y algas usan la luz del sol para hacer su propia comida. Necesitan agua y CO_2 para esto."	"La fotosíntesis, realizada por plantas, algas y ciertas bacterias, convierte la luz solar, CO_2 y agua en energía."

Cuadro 1.1: Comparativa entre el resumen y la simplificación de un texto. La TS se distingue significativamente del resumen de textos, ya que este último tiene como objetivo principal reducir la longitud y el contenido sin perder el contexto. Durante el proceso de generación de resúmenes, el texto resultante tiende a ser más conciso, pero no necesariamente más claro o fácil de entender. Por otro lado, la técnica de TS se caracteriza por producir textos que, frecuentemente, resultan ser más largos. Esta característica se origina en la inclusión de explicaciones adicionales destinadas a facilitar la comprensión del contenido, según se detalla en el estudio de [10].

1.1.2. Desafíos y Necesidades en la Simplificación de Textos en Español

La simplificación de textos en español enfrenta desafíos únicos, que van desde la escasez de recursos específicos hasta los retos inherentes al uso de LLM´s [11, 12, 22]. En contraste con el inglés, se observa una notable carencia en español de investigaciones y herramientas enfocadas en la simplificación textual. Esta situación restringe significativamente las posibilidades de desarrollo de aplicaciones especializadas, tales como aquellas destinadas a brindar asistencia a individuos con discapacidades cognitivas. Según [1, 23], cerca del 10 % de la población mundial padece de dislexia, una condición neurológica que afecta la precisión en la lectura de palabras, así como la fluidez y la ortografía. La Asociación Internacional de Dislexia(IDA, en inglés) define la dislexia como:

"La dislexia es una discapacidad específica de aprendizaje de origen neurobiológico. Se caracteriza por dificultades en el reconocimiento preciso y/o fluido de palabras y por habilidades de ortografía y decodificación deficientes."

Investigaciones anteriores [1, 23, 24] han demostrado que el uso de palabras extensas y poco comunes (consideradas como complejas) puede perjudicar la facilidad de lectura y el entendimiento del texto en individuos con dislexia. Por consiguiente, la implementación de técnicas de simplificación léxica, que consisten en reemplazar términos complejos por sinónimos más breves y habituales, sería beneficiosa para mejorar la comprensión lectora en personas con esta condición. Por otro lado, individuos con Trastornos del Espectro Autista (TEA) afrontan retos al inferir información del contexto y al entender frases extensas que presentan estructuras sintácticas complejas [1, 25]. Para mitigar estas barreras, se podrían implementar técnicas de simplificación sintáctica, enfocadas en la reducción de la complejidad de las estructuras gramaticales en los textos.

No obstante, el estudio presentado en [17] sigue siendo la investigación más completa sobre TS hasta la fecha. Sin embargo, el campo de la TS ha experimentado grandes cambios en los últimos años con el desarrollo de nuevas técnicas de aprendizaje profundo. Un ejemplo de lo mencionado es el proyecto *ClearText*, en el cual se trabaja en la creación de recursos como el corpus *CLEARSIM* y la herramienta *Simple.Text*, destinados a la simplificación de textos en español para este grupo demográfico. A pesar de los desafíos iniciales, como las limitaciones de los modelos generativos, este proyecto busca abordar fenómenos lingüísticos complejos y mejorar la accesibilidad del lenguaje [26].

El uso de LLM´s para la simplificación de textos implica consideraciones medioambientales y económicas significativas. Los grandes modelos de lenguaje, aunque poderosos, requieren una gran cantidad de recursos computacionales, lo que conlleva un impacto ambiental y costos elevados [11, 12]. Esto plantea la necesidad de desarrollar métodos más eficientes y sostenibles que puedan ser accesibles para una amplia gama de usuarios, especialmente en el contexto de países de habla hispana [12, 27]. Por lo tanto, es importante implementar estrategias rentables para la simplificación de textos, haciendo un uso más eficiente de los LLM´s [12, 27]. Una forma de lograrlo es detectar la complejidad de un texto para determinar si es necesario solicitar al LLM que simplifique un segmento en particular.

La identificación de la complejidad textual no solo contribuye al desarrollo de métricas específicas para medir esta característica, sino que también resulta funda-

mental para la evaluación del desempeño de modelos enfocados en la simplificación de textos. Dichas métricas poseen el potencial de ser aplicadas en una variedad de contextos adicionales, enriqueciendo así el campo de estudio [12, 20, 27].

En el contexto de los modelos de NLP, la tarea de distinguir textos de alta complejidad frecuentemente resulta en una incidencia notable tanto de falsos positivos como de falsos negativos. Los primeros ocurren cuando el modelo clasifica erróneamente textos simples como complejos, usualmente por malinterpretar ciertas palabras o frases. En contraste, los falsos negativos se presentan cuando textos inherentemente complejos son incorrectamente identificados como simples, lo que suele deberse a la incapacidad del modelo para reconocer estructuras gramaticales complejas o terminología específica [12, 28–30]. Por consiguiente, resulta fundamental la estimación de la incertidumbre para garantizar la confiabilidad en la utilización de estos modelos de detección. Dicha estimación capacita a los usuarios para comprender de manera más precisa el desempeño del modelo en nuevos conjuntos de datos y para confiar en aquellas predicciones que presentan bajas estimaciones de incertidumbre, mientras que se desestiman aquellas con estimaciones elevadas [12].

1.1.3. Importancia de la Estimación de Incertidumbre en la Detección de Textos Complejos en Español

La UE en la detección de textos complejos en español es un campo de vital relevancia dentro del ámbito del NLP, pero que aún requiere una exploración más profunda en la investigación actual, ya que existen pocos estudios previos sobre este tema, como se indica en [12]. Esta importancia se deriva de la necesidad de comprender y cuantificar el grado de confianza en las predicciones de los modelos de lenguaje, especialmente en aplicaciones que involucran la simplificación de textos y la accesibilidad lingüística [19, 20].

- **Mejora de la Accesibilidad y Comprensión:** La simplificación de textos es crucial para hacer la información más accesible a personas con dificultades de lectura, como individuos con dislexia o Trastornos del Espectro Autista (TEA). La UE ayuda a identificar qué partes de un texto pueden ser más difíciles de entender, permitiendo una adaptación más precisa a las necesidades de estos

usuarios [14, 17].

- **Optimización de Recursos en LLM´s:** Los LLM´s son herramientas poderosas para la simplificación de textos, pero su uso implica consideraciones de costos computacionales y ambientales. La UE permite un uso más eficiente de estos modelos, al identificar cuándo es realmente necesario aplicar la simplificación, reduciendo así el consumo de recursos [11, 12].

Modelos	Hardware	Potencia(W)	Horas	kWh	CO2e	Costo de computación en la nube
Transformer(base)	P100x8	1415.78	12	27	26	41−140
Transformer(big)	P100x8	1515.43	84	201	192	289−981
ELMo	P100x3	517.66	336	275	262	433−1472
BERT(base)	V100x64	12,041.51	79	1507	1438	3751−12,571
BERT(base)	TPUv2x16	—	96	—	—	2074−6912
NAS	P100x8	1515.43	274,120	656,347	626,155	942,973−3,201,722
NAS	TPUv2x1	—	32,623	—	—	44,055−146,848
GPT-2	TPUv3x32	—	168	—	—	12,902−43,008

Cuadro 1.2: El costo estimado de entrenar un modelo en términos de emisiones de CO_2 (libras) y costo de cómputo en la nube (USD) refleja tanto el impacto ambiental como el económico del entrenamiento de modelos de inteligencia artificial. Estas estimaciones son cruciales para comprender la huella de carbono y los costos financieros asociados con el desarrollo de tecnologías de aprendizaje automático avanzadas. Tomado de [11]

- **Mejora de Sistemas Educativos y de Traducción:** En contextos educativos y de traducción, la comprensión precisa de textos complejos es fundamental. La UE en la detección de textos complejos puede mejorar la calidad de los materiales educativos y las traducciones, asegurando que sean adecuados para su público objetivo [31].

- **Contribución a la Investigación Lingüística:** La UE también aporta a la investigación lingüística, proporcionando perspectivas sobre la complejidad del lenguaje y cómo diferentes estructuras y vocabularios afectan la comprensión del texto. Esto puede tener implicaciones significativas en el estudio de la lingüística del español [32].

1.2. Definición del Problema

El avance en los LLM´s ha impulsado significativamente el rendimiento en diversas tareas de NLP, incluyendo la TS. Sin embargo, estos modelos enfrentan desafíos relacionados con el alto costo computacional y el considerable consumo de energía, lo que se traduce en una huella de carbono significativa y costos elevados, especialmente cuando son operados por empresas privadas.

Una dimensión crítica, frecuentemente subestimada, es la UE en la identificación de la complejidad textual [12]. Una UE efectiva permite determinar con qué grado de confianza se predice la complejidad de un texto, lo que puede conducir a una mayor eficiencia en el uso de recursos computacionales al seleccionar solo los textos que realmente requieren simplificación. Además, un método avanzado de UE puede optimizar los procesos de etiquetado de datos y mejorar el rendimiento del modelo a lo largo del tiempo.

En este contexto, proponemos la implementación, además de los métodos Monte Carlo Dropout (MCD) y Deep Ensemble Based (DEB) como líneas base, de un novedoso método de UE basado en la Estimación de Densidades de Características(FDE, en inglés) en las representaciones latentes de los modelos de lenguaje [33]. Este enfoque, inspirado en investigaciones recientes [33–35], busca superar las limitaciones de los métodos convencionales, como MCD y DEB, proporcionando una medición más precisa y detallada de la incertidumbre.

MCD y DEB son métodos populares en el campo del aprendizaje automático, especialmente en aplicaciones que requieren una estimación precisa de la incertidumbre. Su popularidad se debe a su eficacia en una amplia gama de problemas y a su capacidad para mejorar la fiabilidad de los modelos predictivos. Sin embargo, ambos enfoques presentan desventajas que limitan su eficacia en ciertos contextos. El MCD, por ejemplo, puede requerir múltiples pasadas a través del modelo para obtener estimaciones robustas de la incertidumbre, lo que incrementa el costo computacional, especialmente en modelos grandes o en aplicaciones en tiempo real [12, 36]. Por otro lado, DEB implica entrenar múltiples modelos desde cero, lo que no solo es computacionalmente costoso sino que también puede no ser práctico en escenarios con recursos limitados [12, 36].

En contraste, el enfoque basado en FDE en las representaciones latentes ofrece una alternativa prometedora. Este método supera algunas de las limitaciones mencionadas proporcionando una evaluación de la incertidumbre que es computacionalmente más eficiente y puede aplicarse post-entrenamiento, lo que lo hace adaptable a cualquier arquitectura de modelo existente [33–36].

Para evaluar la fiabilidad de la UE de los métodos propuestos, implementaremos la Distancia Jensen-Shannon(JSD, en inglés) como métrica, permitiendo una comparación objetiva entre los métodos de UE en términos de su capacidad para manejar correctamente las puntuaciones de incertidumbre de las predicciones clasificadas tanto correcta como incorrectamente. Este estudio se centra en el uso del modelo transfomer Bidirectional Encoder Representations from Transformers (BERT), adaptado al español como Spanish BERT (BETO), y evalúa los métodos de UE utilizando un conjunto de datos propio de textos simples y complejos en el ámbito de la educación financiera en español.

1.3. Objetivos

Objetivo general: Proponer al menos un método de UE en el proceso de detección de textos complejos en español, con la finalidad de identificar modos de fallo en el modelo, permitiendo así detectar cuándo es necesario reentrenarlo y controlar su comportamiento en dichos escenarios.

Objetivos específicos:

1. Implementar un modelo de aprendizaje profundo para la detección de textos complejos en español.

2. Proponer al menos un método novedoso de UE en la detección de texto complejo en español basado en FDE.

3. Proponer una métrica para cuantificar la confiabilidad de las estimaciones de incertidumbre en la clasificación de texto complejo en español.

1.3.1. Hipótesis de Investigación

Esta investigación propone un enfoque novedoso para la UE en el modelo BETO, mediante la aplicación de un método basado en la estimación de densidades de características en el espacio latente. Tradicionalmente, técnicas como MCD y DEB han sido empleadas para esta tarea, pero enfrentan limitaciones significativas debido a su demanda de alta capacidad computacional, al requerir múltiples ejecuciones o la construcción de conjuntos de modelos. Inspirado en el trabajo de [33–35], FDE emerge como un enfoque superior para evaluar la incertidumbre en redes neuronales, principalmente por su eficiencia y escalabilidad, al ofrecer una metodología aplicable post-entrenamiento y adaptable a cualquier arquitectura. Este método no solo facilita una estimación integral de incertidumbres aleatoria y epistémica, sino que también permite una correlación más precisa con el rendimiento del modelo, destacando su utilidad en detectar datos fuera de distribución y en evaluar la capacidad de generalización del modelo de manera más efectiva y con menos recursos computacionales. Para la evaluación de esta hipótesis se propondrá una métrica que compare la confiabilidad de las estimaciones de incertidumbre, midiendo la divergencia entre las distribuciones de probabilidad de las clasificaciones correctas e incorrectas. Se espera que la metodología basada en densidades de características demuestre una superioridad estadísticamente significativa sobre MCD y DEB, proporcionando una medida de incertidumbre más informativa y confiable.

Hipótesis

El método FDE mejora significativamente la estimación de incertidumbre en el modelo BETO en comparación con las técnicas tradicionales de MCD y DEB. Se espera que FDE proporcione una medida de incertidumbre más informativa y confiable, demostrando superioridad estadísticamente significativa al correlacionarse de manera más precisa con el rendimiento del modelo y detectar datos fuera de distribución de forma más efectiva y con menos recursos computacionales.

1.4. Contribuciones

Como se introdujo anteriormente, la UE en la clasificación de textos complejos en español representa un desafío significativo dentro del paradigma del NLP utilizando LLM´s. Este desafío se hace evidente en contextos donde se requiere una alta precisión en la clasificación textual, como en el dominio de la educación financiera [9]. Por lo tanto, esta tesis se basa particularmente en la aplicación de BETO para la identificación de textos complejos en español, abordando específicamente los desafíos asociados con la precisión y confiabilidad en la clasificación textual. Además, se explora el uso de métodos avanzados de UE, como MCD y DEB, en arquitecturas de aprendizaje profundo para problemas de NLP en entornos del mundo real. Dentro del contexto de esta tesis y su publicación resultante [12], se enumeran las contribuciones al estado del arte de la siguiente manera:

- Desarrollo y entrenamiento del modelo BETO, específicamente entrenado para clasificar textos en español en las categorías de simple o complejo, utilizando un conjunto de datos del dominio específico de la educación financiera propuesto en [9]. Este enfoque no solo mejora la precisión en la identificación de la complejidad textual, sino que también proporciona un modelo de referencia para futuras investigaciones en dominios similares.

- Implementación de métodos de UE utilizando los métodos MCD, FDE y DEB para el clasificador de textos mencionado anteriormente. Esta investigación representa uno de los primeros esfuerzos en aplicar la UE de forma específica para la detección de la complejidad textual en español, llenando un vacío significativo en la literatura existente y abriendo nuevas vías para la exploración de técnicas de UE en NLP.

- Comparación exhaustiva de los métodos de UE implementados, utilizando métricas cuantitativas como la JSD) o Expected Calibration Error (ECE) para evaluar la confiabilidad de las predicciones correctas e incorrectas del modelo. Estas métricas, tal como se describen en [37–40], no solo validan la efectividad de los métodos implementados, sino que también establecen un marco para futuras evaluaciones de la confiabilidad en modelos de NLP.

1.5. Estructura de la Tesis

La presente tesis se organiza en seis capítulos que abordan de manera sistemática el problema de la estimación de incertidumbre en la clasificación de la complejidad textual en español, utilizando modelos de aprendizaje profundo y técnicas avanzadas de procesamiento del lenguaje natural. A continuación, se describe brevemente el contenido de cada capítulo:

- **Capítulo 1: Introducción.** Se presentan los antecedentes y motivaciones que justifican la realización de esta investigación. Se discute la importancia de la simplificación de textos, los desafíos específicos en el idioma español y la relevancia de la estimación de incertidumbre en la detección de textos complejos. Además, se define el problema a abordar, se establecen los objetivos generales y específicos, se formula la hipótesis de investigación y se enumeran las contribuciones principales del trabajo.

- **Capítulo 2: Estado del Arte.** Se realiza una revisión exhaustiva de la literatura relacionada con la predicción de texto complejo y la estimación de incertidumbre en el procesamiento del lenguaje natural. Se exploran las aplicaciones existentes en educación y accesibilidad, los retos específicos del español, y la evolución de los modelos para la predicción de la complejidad textual. Asimismo, se analizan los estudios previos sobre estimación de incertidumbre y los desafíos asociados en tareas de NLP.

- **Capítulo 3: Marco Teórico.** Se proporcionan los fundamentos teóricos que sustentan la investigación. Se introducen las redes neuronales artificiales, incluyendo el perceptrón y sus limitaciones, y los algoritmos de optimización como el descenso de gradiente. Se describe la arquitectura Transformer y su impacto en el NLP, detallando las limitaciones de las redes neuronales recurrentes y la introducción del Transformer. También se aborda la evolución de los modelos de lenguaje preentrenados, sus aplicaciones en tareas de NLP, y se profundiza en la estimación de incertidumbre en aprendizaje automático, los tipos de incertidumbre y los métodos de estimación. Finalmente, se presentan las métricas de fiabilidad en las estimaciones de incertidumbre.

- **Capítulo 4: Metodología.** Se detalla el método propuesto para la estimación de incertidumbre en la clasificación de textos simples y complejos en español. Se describe el uso del modelo BETO para la clasificación, la implementación de las técnicas de estimación de incertidumbre, incluyendo MCD, DEB y el método propuesto de FDE. Además, se presenta el conjunto de datos utilizado, se define la complejidad textual y se especifican los recursos de hardware y software empleados.

- **Capítulo 5: Evaluación de Incertidumbre en Clasificación de Complejidad Textual.** Se presenta el diseño experimental, incluyendo la configuración del conjunto de datos, el entrenamiento y ajuste fino de BETO, y la implementación de los métodos de estimación de incertidumbre. Se muestran los resultados obtenidos para cada método, se realiza un análisis estadístico de los mismos y se comparan los desempeños en términos de fiabilidad de las estimaciones de incertidumbre y costo computacional. Se discuten las implicaciones prácticas y las limitaciones identificadas.

- **Capítulo 6: Conclusiones y Trabajo Futuro.** Se sintetizan las conclusiones generales de la investigación, evaluando el cumplimiento de los objetivos y la validación de la hipótesis. Se destacan las contribuciones significativas del trabajo y se señalan las limitaciones del estudio. Finalmente, se proponen líneas de trabajo futuro y se ofrecen reflexiones finales sobre el impacto y la relevancia de la investigación realizada.

2. Estado del Arte

La evolución del NLP ha permitido avances significativos en tareas como la predicción de la complejidad textual y la UE en modelos de Aprendizaje Profundo(DL, en inglés). Estas áreas son fundamentales para mejorar la accesibilidad y adaptabilidad de los contenidos escritos, especialmente en idiomas como el español, que presentan desafíos particulares debido a su diversidad y complejidad lingüística.

En este capítulo se revisa el estado del arte en la predicción de textos complejos y en las técnicas de estimación de incertidumbre aplicadas al NLP. A pesar de los avances logrados, evidenciados por un incremento en las publicaciones recientes, persisten retos significativos, especialmente en el contexto del español, donde la investigación es menos abundante en comparación con otros idiomas como el inglés.

Primero, se explorará la importancia de la predicción de la complejidad textual, sus aplicaciones en educación y accesibilidad, y los desafíos específicos que plantea el idioma español. Se analizará la transición desde métodos tradicionales basados en características lingüísticas y modelos superficiales, hacia enfoques modernos que emplean redes neuronales profundas y modelos de lenguaje preentrenados.

Posteriormente, se examinarán las aplicaciones y desafíos de la estimación de incertidumbre en el NLP. Se revisarán estudios previos que han aplicado técnicas de estimación de incertidumbre en diversas tareas, destacando su relevancia para mejorar la confiabilidad y robustez de los modelos. Además, se discutirán los desafíos específicos que enfrenta la estimación de incertidumbre en el NLP, como la alta dimensionalidad de los datos, la naturaleza secuencial del lenguaje y la escasez de investigaciones en español.

2.1. Predicción de Texto Complejo

La predicción de la complejidad textual es una tarea fundamental en el NLP, con implicaciones significativas en educación y accesibilidad [41]. Consiste en determinar el nivel de dificultad de un texto o fragmento, lo cual es esencial para adaptar

contenidos a diferentes públicos, como estudiantes de distintas edades, personas con dificultades de lectura o hablantes no nativos [23].

2.1.1. Aplicaciones en Educación y Accesibilidad:

En el ámbito educativo, la predicción de la complejidad textual permite personalizar materiales de enseñanza acorde al nivel de comprensión de los estudiantes [42]. Facilita la selección y adaptación de textos que sean adecuados para diferentes niveles de alfabetización, promoviendo un aprendizaje más efectivo [43].

En términos de accesibilidad, es crucial para desarrollar herramientas que ayuden a personas con discapacidades cognitivas o lingüísticas [17]. Por ejemplo, los sistemas de simplificación automática pueden transformar textos complejos en versiones más sencillas, mejorando la comprensión para individuos con dislexia, autismo o para quienes están aprendiendo el idioma [23].

2.1.2. Retos Específicos en Español:

Aunque ha habido avances significativos en la predicción de complejidad textual en inglés, el idioma español presenta desafíos particulares [44]. La diversidad dialectal, la riqueza morfológica y la complejidad sintáctica del español complican la tarea de modelar la complejidad [32]. Además, existe una escasez de recursos y corpus de alta calidad anotados específicamente para esta tarea en español, lo que limita el desarrollo y evaluación de modelos precisos [12].

2.1.3. Predicción de la complejidad textual, de lo antiguo a lo moderno

En los últimos años, han surgido modelos para la simplificación automática de textos complejos, también conocida como predicción o estimación de la complejidad textual. Estos modelos incluyen desde enfoques basados en modelos superficiales como Máquina de Soporte Vectorial(SVM, en inglés), Random Forest (RF), *Naive Bayes*, *XGBoost* y otros. Estas técnicas generalmente utilizan características lingüísticas, atributos de *bag-of-word* [45], características de legibilidad [46] y *n-grams* [47] como entrada. Más recientemente, se han aplicado modelos basados en Long Short-Term

Memory (LSTM). Por ejemplo, en [48], se desarrolló una arquitectura con dos capas de unidades neuronales LSTM y una capa completamente conectada para clasificar oraciones en italiano e inglés como fáciles o difíciles de entender. Los autores señalan que las capas LSTM analizan peculiaridades léxicas y sintácticas aprovechando su habilidad para recordar la secuencia de entrada. En [49], se generó una red LSTM de dos capas para clasificar oraciones en tres niveles de dificultad, donde la primera capa consta de 512 unidades LSTM y la segunda es una capa totalmente conectada con tres salidas activadas por la función softmax.

En estudios que compararon la implementación de modelos preentrenados con otros enfoques, los resultados sugirieron que el ajuste fino de modelos como BERT mejora el rendimiento. Por ejemplo, la adaptación de BERT para clasificar oraciones en complejas y simples en el idioma amárico mostró un mejor desempeño que los modelos LSTM y Bidirectional Long Short-Term Memory según [50]. [51] demostró que el ajuste fino de RuBERT supera ligeramente al entrenamiento de una red neuronal gráfica para la clasificación de la complejidad textual en ruso. Además, [47] encontró que BERT con ajuste fino alcanzó la mayor precisión (81.45 %) para clasificar oraciones en complejas y simples usando el conjunto de datos Wikilarge, y el mejor clasificador para el conjunto de datos Newsela fue el ajuste fino de ULMFit (80.8 %). Estos modelos se compararon con modelos superficiales no preentrenados como RF y SVM, así como con otros modelos de aprendizaje profundo basados en LSTM.

Uso de Características Lingüísticas y de Legibilidad:

Las características utilizadas incluyen medidas de legibilidad clásicas, como las fórmulas de *Flesch-Kincaid* adaptadas al español [52], así como características léxicas (longitud de palabras, frecuencia de uso), sintácticas (profundidad de árbol sintáctico, tipos de frases) y discursivas [53].

Estas características permiten capturar aspectos que influyen en la comprensión del texto. Sin embargo, su efectividad depende de la calidad de las características seleccionadas y puede no generalizar bien a diferentes dominios o tipos de texto [54].

2.2. Aplicaciones y desafíos de la estimación de incertidumbre en el Procesamiento del Lenguaje Natural

La UE en el NLP ha cobrado relevancia en los últimos años, ya que proporciona información sobre la confiabilidad de las predicciones realizadas por modelos de DL. Esto es especialmente importante en aplicaciones críticas donde las decisiones erróneas pueden tener consecuencias significativas [55].

2.2.1. Algunos estudios previos sobre la estimación de incertidumbre

- **UE en modelos de traducción automática:** En el estudio de [56], los investigadores abordaron la problemática de la evaluación de calidad en traducción automática, en particular cómo los modelos de traducción pueden generar resultados de baja confianza en datos ruidosos y con sesgo. Utilizando técnicas como MCD y DEB, estos métodos generaron puntuaciones de calidad junto con intervalos de confianza, que se correlacionaron con el nivel de incertidumbre en la traducción. La combinación del marco COMET con estos métodos de UE les permitió observar que este enfoque es capaz de identificar posibles errores críticos en las traducciones, lo cual es fundamental para mejorar la confiabilidad de las evaluaciones en múltiples pares de lenguas.

- **UE en clasificación de texto con técnicas bayesianas:**

  1. En [57] investigaron varias estrategias de escalabilidad para la UE en clasificación de texto en contextos de multi-clase, empleando métodos como MCD y DEB, además de extensiones como Concrete Dropout. Uno de los hallazgos clave fue que los ensambles profundos combinados con Concrete Dropout demostraron un mejor rendimiento en la calibración dentro del dominio y en la clasificación ante cambios de dominio. Esto permite una detección de clase novedosa (robustez frente a datos fuera de distribución) superior, un aspecto útil en aplicaciones que requieren una predicción confiable, como la detección de spam y el análisis de sentimientos.

2. Por otro lado, en [58] presentaron un tutorial sobre cuantificación de incertidumbre aplicada específicamente en clasificación de texto. Explicaron la importancia de estimar dos tipos de incertidumbre: la aleatoria (incertidumbre de los datos) y la epistémica (incertidumbre del modelo),(mas información respecto a esto en la sección 3.4). Los investigadores analizaron técnicas avanzadas, como el uso de redes bayesianas y ensambles profundos, y evaluaron su rendimiento en conjuntos de datos grandes, destacando cómo estas técnicas pueden aplicarse en escenarios prácticos como la detección de spam, donde es fundamental identificar con precisión ejemplos donde el modelo no está seguro.

- **Redes de Prior para UE en clasificación y detección de datos fuera de distribución:**

1. Las Redes de Prior introducidas por [59] representan una aproximación novedosa para modelar explícitamente la incertidumbre de distribución en tareas de clasificación. Este método estima una distribución Dirichlet sobre las distribuciones predictivas, permitiendo identificar incertidumbres asociadas con Datos fuera de la Distribución(OOD, en inglés) de entrenamiento. Los experimentos realizados en conjuntos de datos sintéticos y en los conjuntos MNIST y CIFAR-10 mostraron que las Redes de Prior superaron a métodos basados en MCD en la identificación de muestras fuera de distribución. Este enfoque es especialmente útil para identificar con mayor precisión cuándo un modelo se enfrenta a datos significativamente diferentes de los de su entrenamiento, mejorando así la seguridad y robustez del sistema.

2. En un trabajo posterior, en [60] ampliaron este enfoque al introducir una nueva técnica de entrenamiento basada en la divergencia Kullback-Leibler inversa entre distribuciones Dirichlet, lo que permitió a las Redes de Prior escalarse a conjuntos de datos más complejos y con más clases. Esto también mejoró la capacidad de las redes para detectar ataques adversarios, aumentando la cantidad de esfuerzo computacional necesario para que los ataques sean efectivos. Este estudio demuestra que las Redes de Prior

no solo mejoran la UE, sino que también robustecen el sistema frente a manipulaciones adversas en el contexto de seguridad de Inteligencia Artificial(AI, en inglés).

Estos estudios ejemplifican la importancia de las técnicas de UE para mejorar la precisión, seguridad y robustez en tareas de NLP, donde errores de clasificación o predicciones de baja confianza pueden tener implicaciones significativas en aplicaciones prácticas.

2.2.2. Desafíos en la Estimación de Incertidumbre para Tareas de Procesamiento del Lenguaje Natural

A pesar de los avances, la UE en NLP enfrenta desafíos particulares que dificultan su implementación y generalización.

- **Alta Dimensionalidad de los Datos:** Los datos en NLP suelen ser de alta dimensionalidad debido al gran vocabulario y la complejidad lingüística [61, 62]. Esto complica la modelización de la incertidumbre, ya que los espacios de entrada son vastos y las muestras disponibles pueden no cubrir adecuadamente todas las regiones relevantes.

- **Naturaleza Secuencial y Contextual del Lenguaje:** El lenguaje es intrínsecamente secuencial y contextual, lo que significa que las palabras y frases dependen de su contexto [62, 63]. Capturar la incertidumbre en este contexto es complejo, ya que requiere modelos que comprendan dependencias a largo plazo y variabilidad en el uso del lenguaje.

- **Escasez de Estudios en Español:** La mayoría de los estudios sobre UE en NLP se han centrado en el inglés, existiendo una carencia de investigaciones y recursos en español [12, 64]. Esto limita el desarrollo de modelos precisos para este idioma y resalta la necesidad de adaptar y evaluar métodos de UE en contextos multilingües.

2.3. Conclusiones

Este capítulo se ha revisado el estado del arte en la predicción de la complejidad textual y la UE en el NLP, con un enfoque en el idioma español. Se ha destacado la importancia de adaptar textos a diferentes niveles de comprensión para mejorar la educación y la accesibilidad, identificando retos específicos en español debido a su diversidad y complejidad lingüística.

La evolución de los modelos ha pasado de enfoques superficiales, como SVM y RF, a modelos de DL y modelos de lenguaje preentrenados como BERT, que han demostrado mejoras significativas en precisión y generalización. Sin embargo, la estimación de incertidumbre en NLP presenta desafíos particulares, especialmente en español, por la escasez de estudios y recursos disponibles.

Métodos como MCD, DEB y Redes de Prior han sido efectivos en otros idiomas para cuantificar la incertidumbre y mejorar la confiabilidad de las predicciones. No obstante, se requiere investigación adicional para adaptar estos enfoques al español, considerando sus particularidades lingüísticas.

3. Marco Teórico

En la actualidad el Aprendizaje Automático(ML, en inglés) se encuentra presente en todo tipo de áreas y aplicaciones, ejemplo de ello son las aplicaciones de traducción y subtitulado automático como realiza youtube [5], también en aplicaciones vinculadas a la medicina [65], bioinformática [66, 67] y astronomía [68], entre otros. Existen otras áreas de aplicación donde el nivel de rendimiento requerido por los modelos debe ser alto, como la conducción autónoma y los sistemas automátizados que realizan apoyo al diagnóstico de pacientes. Los fallos en este tipo de sistemas puede llevar a grandes pérdidas económicas y en un extremo mucho peor a la pérdida de vidas humanas, de ahí la necesidad de controlar la automatización de las decisiones que se toman con este tipo de aplicaciones. Esto es especialmente pertinente cuando se aplican sistemas de **DL**.

El DL basa su arquitectura en **Redes Neuronales Artificiales** [69, 70]. Las diferencias principales con sistemas tradicionales de ML es que pueden capturar características significativas de los datos de entradas y adaptarlas a las diferentes tareas de aprendizaje a realizar (**Ingeniería de Características**) [71–73]. Lo anteriormente descrito, unido al gran número de parámetros del modelo de las arquitecturas modernas de **DL**, hacen que estos sistemas sean difíciles de interpretar. Esta falta de interpretabilidad puede mitigarse si acompañamos **una medida de la incertidumbre vinculada a las predicciones** que realizan los distintos modelos, con ello podemos controlar el riesgo en la toma de decisiones [74].

3.1. Redes Neuronales Artificiales

A lo largo de la evolución del ser humano, el cerebro ha adquirido características deseables que no se encuentran presentes en la arquitectura de Von Neumann o en la computación paralela actual [75, 76]. Dado lo anterior, nacen las Redes Neuronales(NN, en inglés), también conocidas como *Artificial Neural Networks*. Las NN están inspiradas, como su nombre lo indica, en las redes neuronales biológicas; son

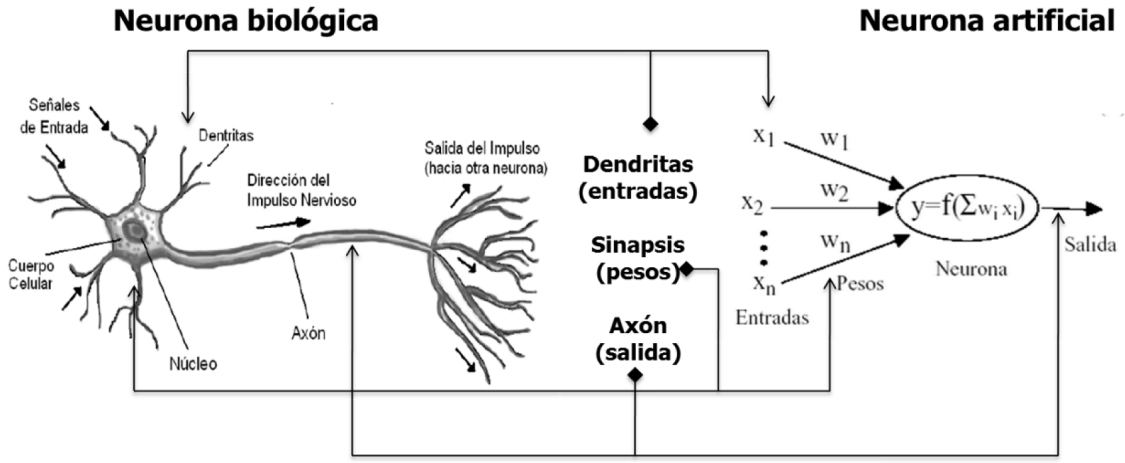

Figura 3.1: En el **lado izquierdo** se encuentra la representación de la **Neurona Biológica**, fuente de inspiración para el desarrollo de las NN representada de forma simple en la **imagen de la derecha**. Tomado de [2].

Neurona Biológica	Neurona Artificial
Núcleo Celular (Soma)	Nodo
Dendritas	Entradas
Sinapsis	Pesos o Interconexiones
Axón	Salidas

Cuadro 3.1: Comparativa entre los componentes presentes en la Neurona Biológica y la Neurona Artificial.

sistemas informáticos masivamente paralelos, conformados por un gran número de procesadores simples con múltiples interconexiones [75].

Las neuronas biológicas son células especializadas que transmiten información a través de señales eléctricas y químicas en el sistema nervioso [77]. De manera análoga, las neuronas artificiales procesan señales de entrada y producen una salida, replicando en cierto grado la función básica de sus contrapartes biológicas.

En la Tabla 3.1 se presenta una comparación entre los componentes de una neurona biológica y una neurona artificial.

3.1.1. El Perceptrón y sus Limitaciones

La red neuronal más antigua y básica conocida es el Perceptrón, descrito por Rosenblatt en 1958 [78]. Este sencillo modelo fue creado inicialmente para tareas de clasificación binaria y sentó las bases de las redes neuronales modernas y el desarrollo del DL tal como se conoce actualmente [78].

El Perceptrón consiste en una sola capa de neuronas que toman las entradas, las ponderan y producen una salida. Cada una de las neuronas del Perceptrón recibe múltiples entradas y devuelve una única salida. Las entradas son multiplicadas por una matriz de pesos, proceso que es semejante a un producto punto. Estos pesos son parámetros ajustables que el modelo utiliza para aprender patrones en los datos.

Después de realizar este proceso, el resultado se pasa a través de una función de activación. En el diseño original, esta función era del tipo escalón, produciendo una salida binaria (0 ó 1) dependiendo de si el resultado supera cierto umbral:

$$f(x) = \begin{cases} 1 & \text{si } x \geq \text{umbral} \\ 0 & \text{si } x < \text{umbral} \end{cases} \tag{3.1}$$

Función de Activación $f(\mathbf{x})$:

La función de activación **f** funciona como una puerta entre la matriz de entrada **x** y la salida de la neurona [3].

A pesar de su importancia histórica, el Perceptrón tiene limitaciones significativas. Una de ellas es su incapacidad para resolver problemas donde los datos no son linealmente separables, como el famoso problema del XOR [79,80]. Estas limitaciones llevaron al desarrollo de redes neuronales multicapa y algoritmos de aprendizaje más avanzados.

3.1.2. Descenso de Gradiente y Optimización en Redes Neuronales

El entrenamiento de redes neuronales implica ajustar los pesos sinápticos para minimizar una función de pérdida que mide el error entre las predicciones de la red y los valores reales. El método más común para realizar este ajuste es el Descenso del Gradiente(GD, en inglés) [73].

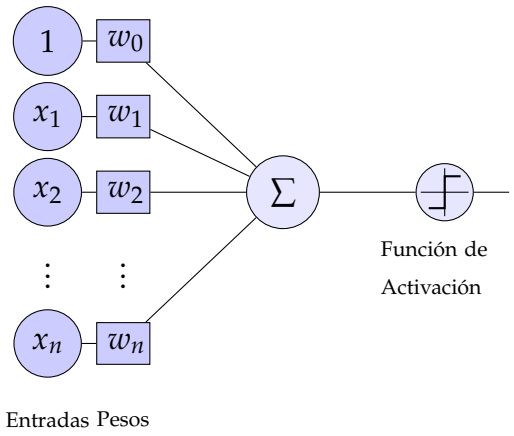

Figura 3.2: Diagrama que representa al Perceptrón. Tomado de [3].

Descenso de Gradiente:

El GD es un algoritmo de optimización que busca encontrar los mínimos de una función diferenciable. Se basa en el cálculo del gradiente de la función de pérdida con respecto a los pesos, y actualiza los pesos en la dirección opuesta al gradiente para reducir el error [81].

La actualización de los pesos se realiza mediante la ecuación:

$$w_{t+1} = w_t - \eta \nabla_w L(w_t) \tag{3.2}$$

donde w_t son los pesos en la iteración t, η es la tasa de aprendizaje y $\nabla_w L(w_t)$ es el gradiente de la función de pérdida L con respecto a los pesos.

Algoritmos de Optimización Comunes:

Existen diversas variantes del GD que mejoran la convergencia y estabilidad del entrenamiento:

- Descenso del Gradiente Estocástico(SGD, en inglés): Actualiza los pesos usando muestras individuales o pequeños lotes (*mini-batches*), lo que introduce ruido en el cálculo del gradiente pero permite escapar de mínimos locales [82–84].
- Estimación Adaptativa de Momentos(Adam, en inglés): Combina ideas de Momentum y Propagación de Raíz Cuadrada Media(RMSProp, en inglés) [83, 84], adaptando la tasa de aprendizaje para cada peso y acelerando la convergencia [85].

- RMSProp: Ajusta la tasa de aprendizaje para cada peso en función de las medias cuadráticas de gradientes pasados [83, 84, 86].

- Algoritmo de Gradiente Adaptativo(AdaGrad, en inglés): Adapta la tasa de aprendizaje basándose en el historial de gradientes, permitiendo mayores actualizaciones para parámetros infrecuentes [83, 84, 87].

La elección del algoritmo de optimización y sus hiperparámetros es crucial para el rendimiento de las NN. Un entrenamiento eficaz permite que la red aprenda representaciones adecuadas de los datos y generalice bien a nuevos ejemplos [73].

El GD y sus variantes son herramientas fundamentales que permiten entrenar redes profundas, haciendo posible avances significativos en áreas como visión por computadora, procesamiento del lenguaje natural y otras aplicaciones de inteligencia artificial [73].

3.2. Arquitectura Transformer y su impacto en el Procesamiento del Lenguaje Natural

En los últimos años, el campo del NLP ha experimentado avances significativos gracias al desarrollo de arquitecturas de redes neuronales más eficientes y potentes. Una de las innovaciones más destacadas es la introducción de la arquitectura Transformer, propuesta por [4] en 2017. Esta arquitectura ha revolucionado la manera en que se abordan las tareas de NLP, superando limitaciones inherentes a modelos anteriores como las Redes Neuronales Recurrentes(RNN, en inglés) y las LSTM.

En esta sección, se explorarán las limitaciones de las arquitecturas previas y se presentará en detalle la arquitectura Transformer, destacando sus componentes clave y su impacto en el NLP. Se analizará cómo esta innovación ha transformado la manera de abordar problemas lingüísticos complejos y ha sentado las bases para el desarrollo de LLM´s.

3.2.1. Limitaciones de las Redes Neuronales Recurrentes y las Long Short-Term Memory

Las RNN y sus variantes como las LSTM han sido fundamentales en el procesamiento de secuencias y en el campo del NLP [88, 89]. Sin embargo, presentan ciertas limitaciones que dificultan su capacidad para modelar dependencias a largo plazo y para aprovechar eficientemente los recursos computacionales.

Problemas de Dependencia a Largo Plazo:

Aunque las RNN y LSTM están diseñadas para manejar secuencias de datos, en la práctica tienen dificultades para capturar dependencias a largo plazo debido al problema del desvanecimiento del gradiente [90]. Esto significa que, a medida que la distancia temporal entre las dependencias aumenta, la capacidad del modelo para aprender relaciones significativas disminuye.

Las LSTM, por otro lado, mitigan parcialmente este problema mediante puertas de entrada, olvido y salida que regulan el flujo de información [88], pero aún enfrentan desafíos cuando se trata de secuencias muy largas o de capturar relaciones complejas en el texto [4, 88].

Paralelización Limitada:

Otro inconveniente de las RNN y LSTM es su naturaleza secuencial en el procesamiento de datos. Cada paso en la secuencia depende del estado anterior, lo que impide la paralelización de los cálculos y resulta en tiempos de entrenamiento más prolongados [4, 88, 89]. Esto limita la eficiencia y escalabilidad de los modelos al trabajar con grandes conjuntos de datos y secuencias largas.

3.2.2. Introducción al Transformer

Para superar las limitaciones de las arquitecturas recurrentes, en [4] presentaron el modelo Transformer, una arquitectura que se basa completamente en mecanismos de atención y elimina la recurrencia. El Transformer ha revolucionado el campo del NLP al permitir una mayor paralelización y al manejar eficientemente dependencias a largo plazo [63].

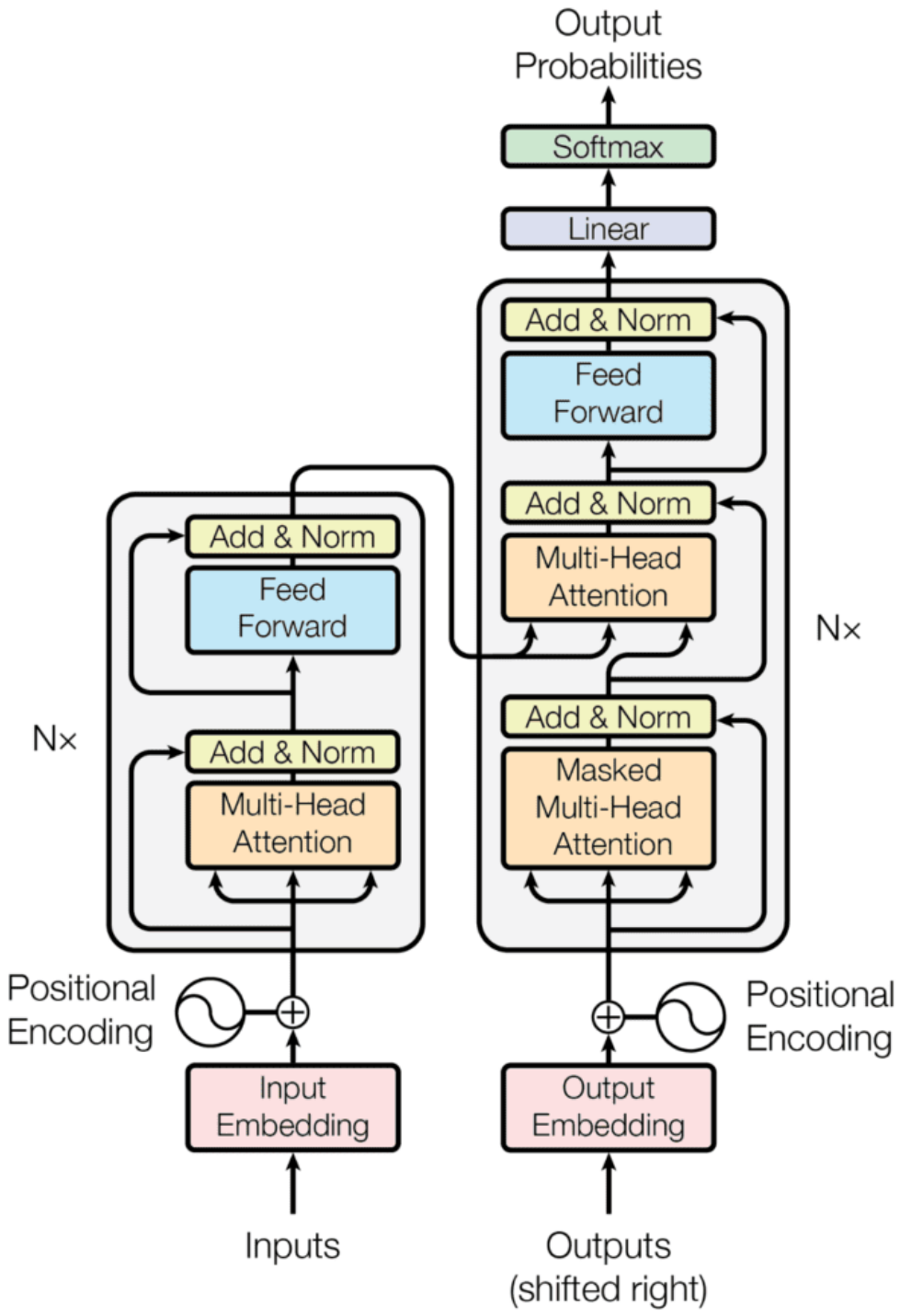

Figura 3.3: Arquitectura del Transformer, mostrando el codificador y el decodificador con sus componentes principales. Tomado de [4].

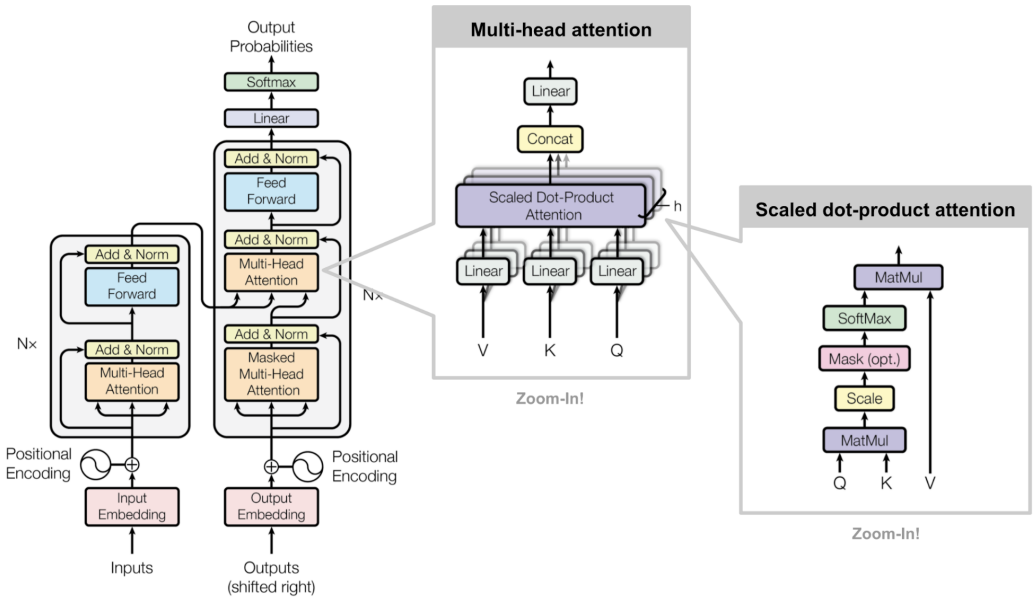

Figura 3.4: Mecanismo de Atención de la arquitectura del Transformer, desglosado para una mejor visualización. Adaptado de [4].

Mecanismo de Atención:

El mecanismo de atención es el componente central del Transformer [4]. Permite que el modelo evalúe la relevancia de diferentes partes de la secuencia de entrada al procesar cada elemento [4, 91]. En lugar de procesar la secuencia de manera secuencial, la atención permite acceder directamente a cualquier parte de la secuencia independientemente de su posición relativa, facilitando la captura de dependencias a largo plazo. [4, 92]

El mecanismo de atención se define mediante las matrices de *queries* (Q), *keys* (K) y *values* (V), y se calcula como:

$$\text{Atención}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{3.3}$$

donde d_k es la dimensión de las claves y el factor de escala $\sqrt{d_k}$ ayuda a estabilizar los gradientes.

Componentes Clave - Codificador y Decodificador:

El Transformer consta de dos partes principales: el codificador (*encoder*) y el decodificador (*decoder*).

- **Codificador:** Compuesto por una pila de bloques idénticos, cada uno con dos subcapas: una capa de multi-atención auto-regresiva (*self-attention*) y una capa de red neuronal (*feed-forward*) totalmente conectada. Cada subcapa está seguida por un mecanismo de normalización y conexiones residuales [4].

- **Decodificador:** Similar al codificador, pero se realizó una adición y modificación de una subcapa de autoatención para evitar que las posiciones atiendan a posiciones posteriores. Este enmascaramiento, combinado con el hecho de que las incrustaciones de salida están desplazadas una posición, garantiza que las predicciones para la posición i sólo pueden depender de las salidas conocidas en posiciones inferiores a i [4]. Resumiendo, tomamos la salida del codificador y nos enfocamos en partes mas importantes de la entrada para generar cada *token* de salida.

Ventajas sobre Arquitecturas Previas:

El Transformer ofrece varias ventajas significativas sobre las RNN y LSTM:

- **Captura de Dependencias a Largo Plazo:** Gracias al mecanismo de atención, el Transformer puede relacionar directamente cualquier par de posiciones en la secuencia, independientemente de su distancia [4].

- **Paralelización Eficiente:** Al eliminar la necesidad de procesar secuencias de manera secuencial, el Transformer permite la paralelización durante el entrenamiento, aprovechando mejor los recursos computacionales y reduciendo los tiempos de entrenamiento [4].

- **Mejor Rendimiento en Tareas de NLP:** Los modelos basados en Transformers han alcanzado el estado del arte en diversas tareas de NLP, como traducción automática, resumen de texto y respuesta a preguntas [63, 93].

- **Flexibilidad Arquitectónica:** La naturaleza modular del Transformer facilita su adaptación y expansión, permitiendo el desarrollo de modelos más grandes y complejos, como los LLM´s [93, 94].

3.3. Modelos de Lenguaje Preentrenados

En los últimos años, los modelos de lenguaje preentrenados han revolucionado el campo del NLP, proporcionando avances significativos en una amplia gama de tareas. Estos modelos, entrenados en enormes volúmenes de datos textuales, han demostrado una capacidad sin precedentes para capturar las complejas estructuras sintácticas y semánticas del lenguaje humano. Esto ha permitido mejoras sustanciales en aplicaciones como la traducción automática, el análisis de sentimiento, la generación de texto y la clasificación de documentos.

3.3.1. Evolución de los Modelos de Lenguaje

El modelado del lenguaje es una tarea fundamental en el NLP, que implica predecir la probabilidad de una secuencia de palabras o tokens [95]. A lo largo de las últimas décadas, los modelos de lenguaje han evolucionado significativamente, pasando de enfoques estadísticos básicos a modelos neuronales avanzados y, más recientemente, a grandes modelos de lenguaje LLM´s.

Modelos Estadísticos del Lenguaje(SML, en inglés):

Los SML fueron los primeros en utilizarse para estimar la probabilidad de una secuencia de palabras en un lenguaje natural. Estos modelos se basan en el supuesto de la cadena de Markov, donde la probabilidad de una palabra depende de un número fijo de palabras anteriores [96]. Los modelos n-gramas son un ejemplo común de SML, donde se utilizan frecuencias de n-gramas en un corpus de texto para estimar probabilidades [97].

Sin embargo, los SML presentan limitaciones significativas, como la explosión combinatoria de posibles n-gramas y la incapacidad para capturar relaciones semánticas profundas entre palabras [98].

Modelos del Lenguaje(LM, en inglés):

Para superar las limitaciones de los SML, se introdujeron los LM, que utilizan las NN para aprender representaciones continuas de palabras y capturar dependencias

a largo plazo en el lenguaje [95]. Los LM permiten representar palabras en vectores densos (embeddings), capturando similitudes semánticas y sintácticas [99].

Modelos como Word2Vec [99] y GloVe [100] aprendieron representaciones de palabras a gran escala, mejorando el rendimiento en diversas tareas de NLP. Además, las RNN y LSTM se utilizaron para modelar secuencias de palabras, permitiendo capturar dependencias contextuales [101].

Embeddings:

Los embeddings son representaciones matemáticas en un espacio vectorial de baja dimensión que encapsulan información semántica y contextual de palabras, frases u otras entidades. En el NLP, los embeddings transforman datos textuales, originalmente categóricos o simbólicos, en vectores continuos que las máquinas pueden procesar eficientemente. Estas representaciones capturan relaciones semánticas y sintácticas entre entidades, de modo que palabras similares (según el contexto) tienden a estar cerca unas de otras en el espacio vectorial [102, 103].

Modelos de Lenguaje Preentrenados(PLM, en inglés):

En los últimos años, los PLM han revolucionado el campo del NLP. Estos modelos, como BERT [63] y GPT [94], son entrenados en grandes cantidades de texto no supervisado para aprender representaciones contextuales profundas del lenguaje.

Los PLM utilizan arquitecturas basadas en Transformers [4], aprovechando mecanismos de autoatención para capturar dependencias a largo plazo sin las limitaciones de las RNN. Después del preentrenamiento, estos modelos pueden ser ajustados (*fine-tuning*) para tareas específicas con cantidades relativamente pequeñas de datos etiquetados, obteniendo resultados prometedores en una amplia gama de aplicaciones de NLP [63].

Aunque la mayoría de los PLM iniciales se desarrollaron para el idioma inglés, en los últimos años ha habido esfuerzos significativos para crear modelos preentrenados para otros idiomas, incluido el español [64].

BETO (Bidirectional Encoder Representations from Transformers (BERT) para Español):

BETO es un modelo de lenguaje preentrenado específico para el idioma español, presentado por [64]. BETO se basa en la arquitectura BERT [63] y fue entrenado utilizando el corpus de Wikipedia en español y otras fuentes de texto, sumando aproximadamente 3 mil millones de palabras.

Detalles del Entrenamiento y Arquitectura:

BETO utiliza la arquitectura BERT-base, con 12 capas Transformer, 768 dimensiones ocultas, 12 cabezas de atención y un total de 110 millones de parámetros [64]. El modelo fue preentrenado utilizando las mismas tareas que BERT original: *Masked Language Modeling* (MLM) y *Next Sentence Prediction* (NSP).

Comparación con BERT:

Mientras que BERT fue entrenado principalmente en textos en inglés, BETO fue entrenado exclusivamente en español, lo que le permite capturar mejor las particularidades sintácticas y semánticas del idioma [64]. En evaluaciones realizadas en diversas tareas de NLP en español, BETO ha demostrado un rendimiento superior en comparación con versiones multilingües de BERT y otros modelos basados en Transformer [64].

Otros Modelos en Español:

Además de BETO, existen otros modelos preentrenados en español, como:

- **Spanish RoBERTa** [104]: Basado en la arquitectura (Un Enfoque de Preentrenamiento de BERT Optimizado de Manera Robusta(RoBERTa, en inglés)) [105], entrenado en un corpus amplio de texto en español.
- **MarIA** [106]: Un modelo basado en RoBERTa entrenado con el corpus en español más grande hasta la fecha (570GB).

Estos modelos amplían las opciones para aplicaciones en español y han contribuido al avance del procesamiento del lenguaje en este idioma.

3.3.2. Aplicaciones de los Modelos del Lenguaje Preentrenados en tareas de Procesamiento del Lenguaje Natural

Los PLM han sido aplicados con éxito en diversas tareas de NLP, mejorando significativamente el estado del arte.

Clasificación de Texto:

Los PLM se han utilizado para tareas de clasificación de texto, como análisis de sentimiento, detección de spam y categorización temática [107]. Ajustando los modelos preentrenados con conjuntos de datos etiquetados del dominio de investigación, se logra una comprensión profunda del contexto y una mejora en la precisión de las clasificaciones [12].

Resumen Automático:

En tareas de resumen automático, los PLM han permitido generar resúmenes coherentes y relevantes de documentos extensos [108]. Modelos como BART [109] y T5 [110] han demostrado capacidades sobresalientes en resumen de textos mediante enfoques de modelado secuencia a secuencia.

Traducción Automática:

La traducción automática se ha beneficiado enormemente de los PLM. Modelos como mBART [111] y M2M-100 [112] han sido entrenados en múltiples idiomas, permitiendo traducciones de alta calidad sin necesidad de grandes cantidades de datos paralelos para cada par de idiomas.

Estos avances en los PLM han impulsado el desarrollo de aplicaciones más precisas y eficientes en el NLP, ampliando las posibilidades en investigación y en soluciones prácticas para diversos idiomas, incluido el español.

3.4. Estimación de Incertidumbre en Aprendizaje Automático

Incertidumbre:

La incertidumbre implica un estado de indecisión, sugiriendo una ausencia de certeza, firmeza, control y previsión. Comúnmente asociada con la sensación de duda en el momento actual, la incertidumbre también abarca inseguridades relacionadas con el pasado y el futuro, así como cuestionamientos sobre otras personas, las cualidades del entorno y la dinámica de interacción entre uno mismo, los demás y el entorno [113, 114].

3.4.1. Incertidumbre en Aprendizaje Automático

Incertidumbre en ML:

Se refiere a la falta de confianza o certeza que un modelo tiene en sus predicciones [115].

Comprender y cuantificar esta incertidumbre es crucial, especialmente en aplicaciones donde las decisiones basadas en predicciones incorrectas pueden tener consecuencias significativas, como en medicina, conducción autónoma o finanzas [55].

La estimación de la incertidumbre permite identificar cuándo un modelo es probable que falle, proporcionando información valiosa para la toma de decisiones. Esto es especialmente importante en modelos de aprendizaje profundo, que a menudo se consideran como cajas negras y pueden ser excesivamente confiados en sus predicciones [116].

Impacto en la Toma de Decisiones:

Integrar la UE en los sistemas de aprendizaje automático mejora la robustez y confiabilidad de las decisiones automatizadas [117]. Al cuantificar la incertidumbre, es posible:

- Detectar casos donde el modelo puede estar equivocado y requerir intervención humana.

- Mejorar la interpretación y explicabilidad de los modelos.
- Informar políticas de riesgo y confianza en sistemas críticos.
- Facilitar el aprendizaje activo al identificar ejemplos donde el modelo necesita más datos para mejorar.

3.4.2. Tipos de Incertidumbre

La incertidumbre en el aprendizaje automático generalmente se categoriza en dos tipos principales: incertidumbre epistémica e incertidumbre aleatoria [115]. Además, se reconoce la incertidumbre distributiva, que se relaciona con cambios en la distribución de los datos.

Incertidumbre Epistémica [5, 118]:

Este tipo de Incertidumbre se origina debido a la falta de conocimiento que presentan los datos:

- **Falta de evidencia:** Debido a esto el modelo no tiene suficientes casos para tener un conocimiento adecuado y por ende las predicciones que realizará no serán confiables.
- **Ignorancia:** El modelo se adapta muy bien a un problema determinado por lo que cuando queremos realizar predicciones en un contexto o problema diferente **(Generalizar)**, estas no son confiables porque está adaptado a ver un tipo de datos diferentes a los que se le están ingresando

Incertidumbre Aleatoria [5, 118]:

Cuando se habla de Incertidumbre aleatoria, se hace referencia a la parte de la variabilidad en los datos que no se puede atribuir a causas específicas o predecibles, sino que proviene de la naturaleza misma del proceso de recopilación de datos. Esta incertidumbre se asocia comúnmente con el azar y la probabilidad, y no se puede eliminar por completo.

La incertidumbre aleatoria puede clasificarse en dos tipos, **homocedástica** y **heterocedástica**, a continuación se hablará un poco mas sobre el tema [5, 118, 119]:

- **Incertidumbre aleatoria Homocedástica:** Este tipo de incertidumbre mide el nivel de ruido presente en el proceso de medición, además permanece **invariante** a las diferentes entradas.

- **Incertidumbre aleatoria Heterocedástica:** Este tipo de incertidumbre aleatoria mide el ruido intrínseco presente en las observaciones de una manera que este depende de las entradas del modelo.

¿Qué es ruido?

Cuando hablamos de **ruido** en el contexto de las mediciones, nos referimos a las fluctuaciones o variaciones no deseadas en los datos que se recopilan durante un experimento o proceso de medición. Estas fluctuaciones pueden deberse a diversas fuentes y factores, y su presencia puede afectar la precisión y la confiabilidad de las mediciones.

Incertidumbre Distributiva [5]:

Este tipo de incertidumbre corresponde a cambios en la distribución de los datos en tiempo de predicción. Estos cambios en la distribución de los datos pueden deberse a la posible falta de generalización del modelo, ya que se requiere que el modelo prediga ciertos puntos de datos que son diferentes a los observados durante el entrenamiento.

La **Figura** 3.5 muestra los diferentes tipos de incertidumbre descritos utilizando un modelo lineal simple para ilustrar cómo [5]:

- **a)** La incertidumbre sobre los parámetros del modelo está relacionada con el conjunto de diferentes modelos que resuelven el problema

- **b)** Los puntos cercanos al límite de decisión muestran una alta incertidumbre aleatoria

- **c)** Los puntos de datos no incluidos en el conjunto de entrenamiento pueden inducir una alta incertidumbre en el tiempo de predicción.

La **Figura** 3.6 representa un proceso lineal real **(y=x)** que se muestreó alrededor de **x= -2,5** y **x=2,5**.

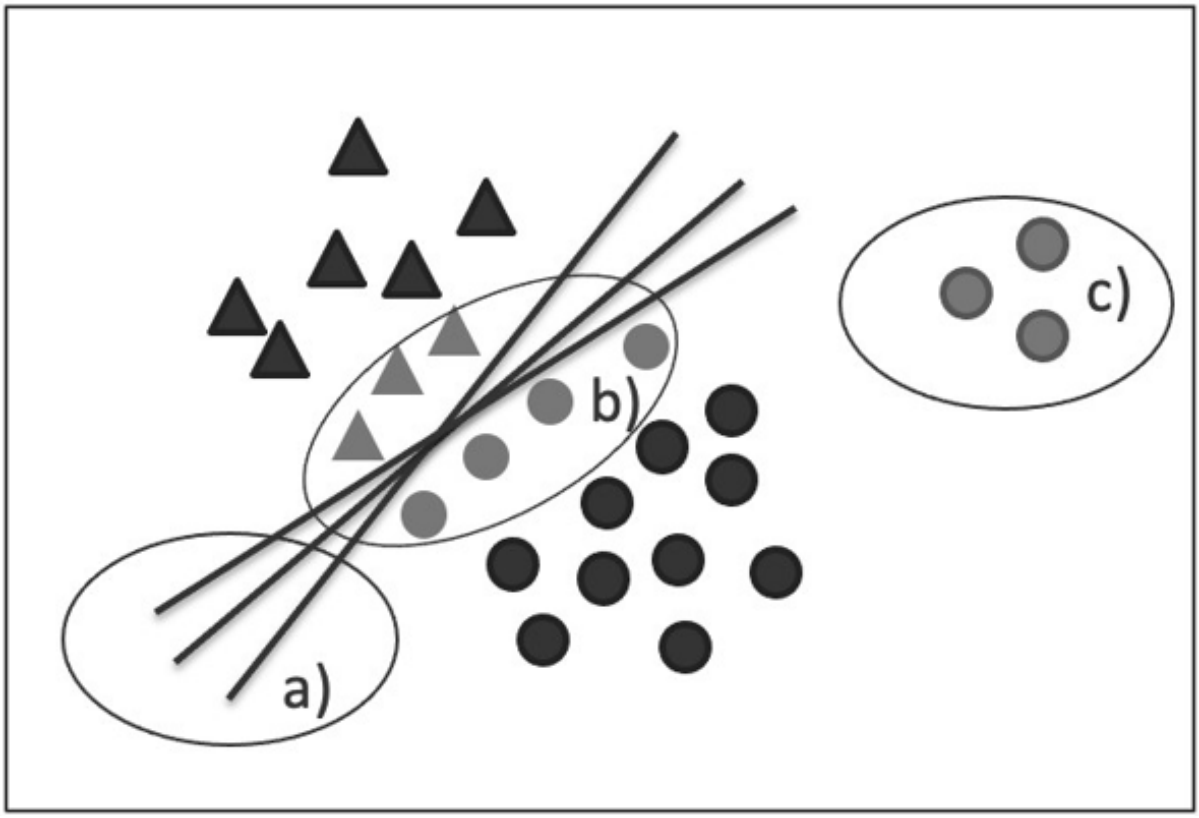

Figura 3.5: Modelo lineal que permite ilustrar las diferentes fuentes de incertidumbres vistas anteriormente **a)** Epistémica, **b)** Aleatoria, **c)** Distributiva. Tomado de [5]

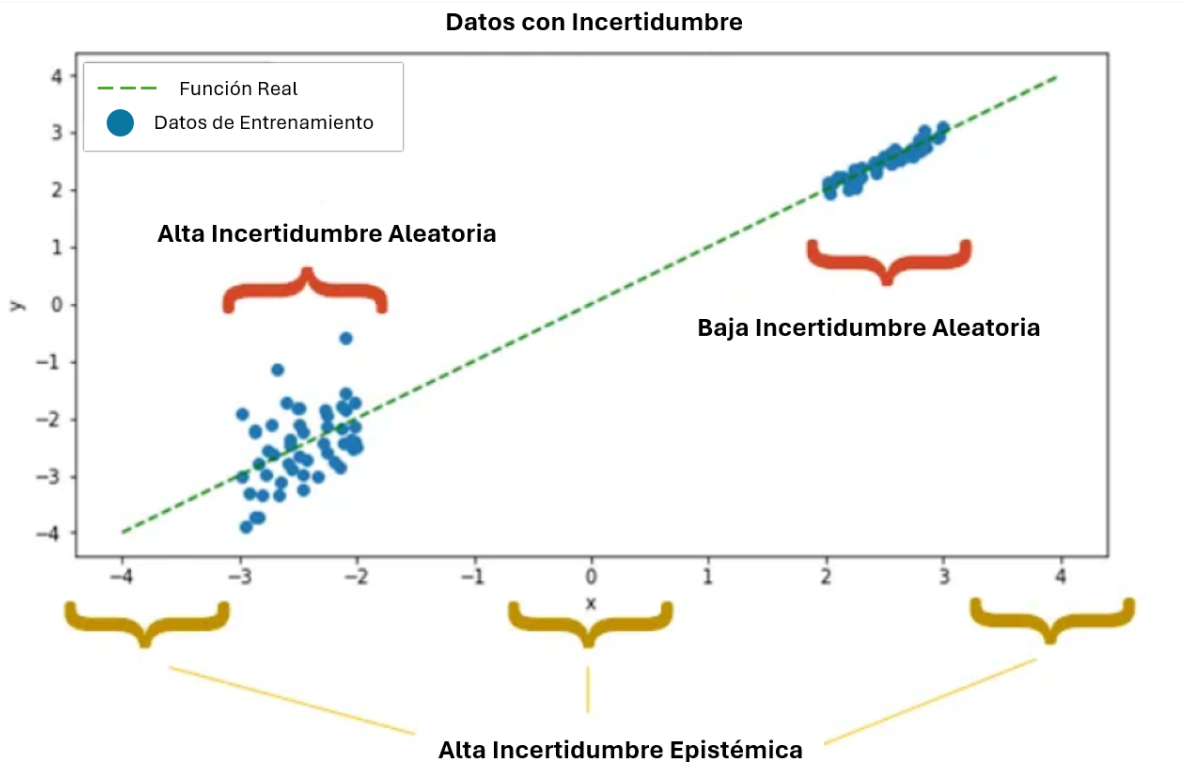

Figura 3.6: Exhibición de los diferentes tipos de **Incertidumbre(Aleatoria y Epistémica)** en un contexto de **regresión lineal**. Adaptado de [6]

Podemos observar que alrededor de $x = -2.5$, existe una **alta Incertidumbre Aleatoria**, esto se debe a que los datos de entrenamiento contienen mucho ruido y se diferencian mucho a los reales. Dicha incertidumbre es inherente y persiste a pesar de realizar mediciones adicionales, debido a errores constantes del sensor en torno a $x = -2.5$, inherentes a su diseño.

En este contexto, la **alta Incertidumbre Epistémica** se manifiesta en áreas con escasez o ausencia total de observaciones. El modelo no sabría que responder con certeza, o mas bien alucinaría en estos puntos ya que no tiene datos de los que aprender. La disponibilidad de más datos en estas regiones contribuiría a reducir tal incertidumbre.

En un aspecto mas generalizado se suele clasificar la incertidumbre en *Reducible* e *Irreducible*, y las anteriormente descritas dentro de estas clasificaciones. A continuación se dará información sobre esto:

Incertidumbre Reducible [5, 120]:

Este tipo de incertidumbre, como su nombre indica, puede ser reducida a medida que vamos agregando más ejemplos a nuestro conjunto de entrenamiento, o sea aumentamos el conocimiento para que el modelo pueda aprender mejor y realizar predicciones más precisas. Lo mismo pasa con las configuraciones de los diferentes elementos de los modelos, una vez ajustada la arquitectura, los hiperparámetros y la complejidad del mismo vamos a poder disminuir la incertidumbre debido a que el modelo va a realizar su trabajo de una forma óptima.

Incertidumbre Irreducible [5, 120]:

Como su nombre indica, este tipo de incertidumbre no puede ser reducida bajo ninguna circunstancia y está estrechamente vinculada a los datos. Podemos realizar las acciones mencionadas en el caso de la **Incertidumbre Reducible** , pero sería en vano. La suposición de esto es que este tipo de incertidumbre es causada por los diferentes procesos que son inherentemente aleatorios, tanto en la adquisición como en la propia generación de los datos.

En la mayoría de las investigaciones vinculan la **incertidumbre epistémica** con la **reducible** y la **incertidumbre aleatoria** con la **irreducible**. Estas interpretaciones pueden estar sujetas a diferentes interpretaciones, por ejemplo en [120] muestran situaciones en que la incertidumbre aleatoria se convierte en incertidumbre epistémica a medida que el modelo va evolucionando.

3.4.3. Métodos de Estimación de Incertidumbre

Existen varios métodos para estimar la incertidumbre en modelos de aprendizaje profundo. A continuación, se describen algunos de los más utilizados y relevantes para esta tesis.

Monte Carlo Dropout (MCD):

El método de MCD utiliza el *dropout* durante la fase de inferencia para aproximar distribuciones posteriores y estimar la incertidumbre [116]. Al realizar múltiples pa-

sadas con *dropout* activado, se obtienen diferentes predicciones, cuya variabilidad se utiliza para cuantificar la incertidumbre.

Funcionamiento y Aplicaciones

MCD interpreta el *dropout* como una técnica de aproximación variacional a inferencia bayesiana. Es sencillo de implementar y no requiere cambios significativos en la arquitectura del modelo [116]. Se ha aplicado en diversas áreas, incluyendo visión por computadora y procesamiento del lenguaje natural, para mejorar la UE [115].

Deep Ensembles (DEB):

Los DEB consisten en entrenar múltiples modelos independientes con inicializaciones aleatorias y posiblemente diferentes subconjuntos de datos [121]. La combinación de sus predicciones proporciona una estimación de la incertidumbre basada en la diversidad entre los modelos.

Descripción y Ventajas

DEB es un método no bayesiano que ha demostrado un rendimiento sólido en la UE y detección de datos fuera de distribución [117]. Aunque requiere mayor costo computacional debido al entrenamiento de múltiples modelos, ofrece ventajas en robustez y precisión en comparación con métodos individuales.

Feature Densities Estimation (FDE):

La FDE es un enfoque que estima la incertidumbre modelando la densidad de las representaciones latentes en las capas intermedias de la red [33]. Al evaluar cuán probable es una representación interna dada en relación con las distribuciones aprendidas durante el entrenamiento, es posible estimar la incertidumbre asociada a una predicción.

Introducción y Relevancia para la Tesis

FDE es particularmente relevante para esta tesis, ya que ofrece una UE eficiente y adaptable a diferentes arquitecturas [33, 122]. Al centrarse en las representaciones internas, puede detectar situaciones donde el modelo está menos seguro debido a entradas que se alejan de la distribución aprendida. Esto es esencial en la detección de textos complejos en español, donde la variabilidad lingüística puede afectar la confianza del modelo.

3.5. Métricas de Fiabilidad en las Estimaciones de Incertidumbre

La evaluación de la fiabilidad en las estimaciones de incertidumbre es crucial para garantizar que los modelos de DL proporcionen predicciones confiables y útiles [39]. Existen diversas métricas que permiten cuantificar qué tan bien calibradas están las UE de un modelo. A continuación, se describen algunas de las métricas más relevantes.

3.5.1. Error de Calibración Esperado (ECE)

El ECE es una métrica que cuantifica la discrepancia entre la confianza predicha por un modelo y la precisión real observada [40]. Es ampliamente utilizada para evaluar la calibración de modelos de clasificación.

Definición y Fórmula:

El ECE se calcula dividiendo las predicciones en M intervalos (o *bins*) basados en los niveles de confianza. Para cada intervalo m, se calcula la confianza promedio y la precisión empírica. El ECE es entonces la suma ponderada de las diferencias absolutas entre estas dos cantidades:

$$\text{ECE} = \sum_{m=1}^{M} \frac{n_m}{n} \left| \text{acc}(B_m) - \text{conf}(B_m) \right| \tag{3.4}$$

donde:

- n es el número total de muestras.
- n_m es el número de muestras en el intervalo B_m.
- $\text{acc}(B_m)$ es la precisión promedio en el intervalo B_m.
- $\text{conf}(B_m)$ es la confianza promedio de las predicciones en el intervalo B_m.

Uso en la Evaluación de Modelos:

Un ECE bajo indica que el modelo está bien calibrado, es decir, su confianza predicha se alinea con la precisión real [39]. Esta métrica es útil para comparar modelos y para diagnosticar problemas de sobreconfianza o subconfianza en las predicciones. La calibración es especialmente importante en aplicaciones críticas donde la confiabilidad de las predicciones es esencial [117].

3.5.2. Distancia Earth Mover's (EMD)

La Distancia Earth Mover's (EMD, en inglés), también conocida como Distancia de Wasserstein de primer orden, es una medida que cuantifica la diferencia entre dos distribuciones de probabilidad considerando el costo mínimo necesario para transformar una distribución en otra [123]. Se interpreta metafóricamente como el mínimo "trabajo"necesario para mover y reorganizar "tierra"(probabilidad) para convertir una distribución en otra.

Concepto y Cálculo:

Matemáticamente, la EMD entre dos distribuciones P y Q definidas en un espacio métrico (Ω, d) se define como:

$$\text{EMD}(P, Q) = \inf_{\gamma \in \Gamma(P,Q)} \int_{\Omega} d(x, y)\, \mathrm{d}\gamma(x, y) \tag{3.5}$$

donde $\Gamma(P, Q)$ es el conjunto de todas las medidas de probabilidad en $\Omega \times \Omega$ con marginales P y Q, y $d(x, y)$ es la distancia entre los puntos x y y en el espacio Ω.

En casos discretos, si P y Q son histogramas normalizados con n bins, la EMD se puede calcular resolviendo un problema de programación lineal que minimiza el costo total de transformar P en Q [123].

Aplicación en la comparación de Distribuciones de Incertidumbre:

En el contexto de UE, la EMD se utiliza para medir la discrepancia entre las distribuciones de incertidumbre asociadas a predicciones correctas e incorrectas [124]. A diferencia de medidas como la divergencia de Kullback-Leibler, la EMD considera

la estructura y las relaciones entre los valores de las distribuciones, siendo sensible a la forma y la ubicación de las mismas.

Una EMD mayor indica que el modelo asigna distribuciones de incertidumbre significativamente diferentes a las predicciones correctas e incorrectas, lo que es deseable para distinguir cuándo el modelo es confiable y cuándo no [123, 124]. Esto permite una mejor discriminación y comprensión de las áreas donde el modelo puede ser propenso a errores.

3.5.3. Otras Métricas Relevantes

Además del ECE y la EMD, existen otras métricas que son útiles para evaluar la calibración y la calidad de las estimaciones de incertidumbre.

- **Entropía:** La entropía mide la incertidumbre promedio en una distribución de probabilidad [125]. En modelos de clasificación, la entropía de la distribución de probabilidad sobre las clases puede utilizarse como una estimación de la incertidumbre predictiva [126]:

$$H(p) = -\sum_{i=1}^{C} p_i \log p_i \tag{3.6}$$

  donde C es el número de clases y p_i es la probabilidad predicha para la clase i.

- **Log-Likelihood Negativo:** El Log-Likelihood Negativo(NLL, en inglés) es una métrica que evalúa la calidad de las probabilidades predichas por un modelo [127, 128]. Se define como:

$$\text{NLL} = -\sum_{i=1}^{n} \log p_\theta(y_i|x_i) \tag{3.7}$$

  donde $p_\theta(y_i|x_i)$ es la probabilidad asignada por el modelo a la etiqueta verdadera y_i dado el ejemplo x_i.

  Un NLL más bajo indica que el modelo asigna mayores probabilidades a las etiquetas correctas, lo que sugiere una mejor calibración y confianza en las predicciones [117, 129].

3.6. Conclusiones

En este capítulo se ha presentado un marco teórico que fundamenta la investigación desarrollada en esta tesis, abarcando desde los conceptos básicos de las redes neuronales artificiales hasta las técnicas avanzadas de estimación de incertidumbre en modelos de aprendizaje profundo. A continuación, se resumen los puntos clave tratados:

- **Importancia del Aprendizaje Automático y el Aprendizaje Profundo:** Se destacó la presencia omnipresente del ML y el DL en diversas áreas y aplicaciones, desde la traducción automática hasta la medicina y la conducción autónoma. Se enfatizó la necesidad de controlar la automatización de decisiones en sistemas críticos donde los errores pueden tener consecuencias graves.

- **Fundamentos de las Redes Neuronales Artificiales:** Se exploró la analogía entre las neuronas biológicas y las neuronas artificiales, describiendo la estructura y funcionamiento básico de las NN. Se abordó el Perceptrón como la unidad fundamental de las redes neuronales y se discutieron sus limitaciones, especialmente en problemas no linealmente separables.

- **Optimización y Descenso de Gradiente:** Se describieron los métodos de optimización utilizados en el entrenamiento de redes neuronales, resaltando el papel del GD y sus variantes como SGD y Adam. Estos algoritmos son cruciales para ajustar los pesos sinápticos y minimizar la función de pérdida, permitiendo que las redes aprendan patrones complejos en los datos.

- **Arquitectura Transformer y su Impacto en el NLP:** Se introdujo la arquitectura Transformer como una solución a las limitaciones de las RNN y LSTM, especialmente en la captura de dependencias a largo plazo y la paralelización eficiente. Se explicó el mecanismo de atención y se detallaron los componentes clave del codificador y decodificador, destacando las ventajas significativas sobre arquitecturas previas.

- **Modelos de Lenguaje Preentrenados y BETO:** Se revisó la evolución de los modelos de lenguaje desde los SML hasta los PLM. Se presentó BETO, un mo-

delo de lenguaje preentrenado específico para el español, basado en BERT. Se discutió su arquitectura, entrenamiento y ventajas en comparación con modelos multilingües, subrayando su relevancia para tareas de NLP en español.

- **Estimación de Incertidumbre en Aprendizaje Automático:** Se destacó la importancia de cuantificar la incertidumbre en los modelos de aprendizaje automático, especialmente en aplicaciones críticas. Se describieron los tipos de incertidumbre (epistémica, aleatoria y distributiva) y su impacto en la toma de decisiones. Comprender y estimar la incertidumbre es esencial para mejorar la confiabilidad y robustez de los modelos.

- **Métodos de Estimación de Incertidumbre:** Se exploraron varios métodos para estimar la incertidumbre en modelos de aprendizaje profundo, incluyendo MCD, DEB y FDE. Se discutieron sus fundamentos, ventajas y limitaciones, preparando el terreno para la selección del método más adecuado para esta investigación.

- **Métricas de Fiabilidad en las Estimaciones de Incertidumbre:** Se presentaron métricas como el ECE y la EMD para evaluar la calibración y fiabilidad de las estimaciones de incertidumbre. Estas métricas permiten cuantificar qué tan bien un modelo refleja su confianza en las predicciones, lo cual es crucial para aplicaciones donde la confiabilidad es esencial.

Este marco teórico proporciona una comprensión sólida de los conceptos y técnicas fundamentales que sustentan esta tesis. Al explorar las arquitecturas de redes neuronales y los modelos de lenguaje preentrenados, se establece la base para el desarrollo e implementación de métodos avanzados de estimación de incertidumbre en el contexto de la clasificación de textos en español.

La importancia de la estimación de incertidumbre y su impacto en la toma de decisiones automatizadas se ha enfatizado como un componente esencial para mejorar la confianza y seguridad en los sistemas de aprendizaje automático. Los métodos y métricas discutidos en este capítulo serán fundamentales para los experimentos y análisis presentados en los capítulos siguientes.

4. Metodología

En este trabajo se propone el uso de una arquitectura transformer como base para poder realizar una correcta discriminación del texto en simple o complejo. Específicamente para este contexto se utilizará BETO [130] el cual es una implementación en español de BERT. Este modelo será utilizado, haciendo uso del ajuste fino(**fine-tuning**) con un *dataset* propio elaborado con textos de educación financiera en español, con el objetivo de probar que tan bien puede diferenciar entre segmentos de textos simples y complejos. Los métodos fundamentados en BERT han evidenciado una ventaja significativa sobre los enfoques más tradicionales y superficiales en la tarea de simplificación de textos complejos. Adicionalmente, se ha observado que los modelos previamente entrenados en el idioma objetivo incrementan notablemente el rendimiento, tal como se destaca en la investigación [12].

Además de las técnicas de UE tradicionales, como el MCD y DEB, se propone el desarrollo de un método de UE basado en la estimación de densidades de características de las representaciones latentes en el modelo BETO, siguiendo el enfoque discutido por [33]. Este enfoque avanzado implica la extracción de representaciones latentes del modelo y la aplicación de técnicas avanzadas de aprendizaje profundo para modelar su distribución. Además nos permite cuantificar la incertidumbre asociada a cada predicción. Entre las ventajas de este método se pueden encontrar que son computacionalmente livianos y dejan el procedimiento de entrenamiento sin cambios [33, 34].

4.1. Descripción del Método Propuesto

En esta sección se presenta el **método propuesto** para la estimación de incertidumbre en la clasificación de textos en español. El enfoque se basa en la FDE, que aprovecha las representaciones latentes generadas por modelos de lenguaje pre-entrenados como BETO. El método busca estimar la densidad de probabilidad de las características latentes para cuantificar la incertidumbre asociada a cada predicción.

El método propuesto ofrece una alternativa eficiente y efectiva a las técnicas tradicionales como MCD y DEB, al reducir significativamente el costo computacional durante la fase de evaluación sin comprometer el desempeño en la estimación de incertidumbre.

A lo largo de esta sección, se proporcionan detalles técnicos y consideraciones prácticas para la implementación del método, así como discusiones sobre sus ventajas y posibles limitaciones.

4.1.1. Uso de BETO para Clasificación de Textos Simples y Complejos

En este estudio, se emplea BETO [64], un modelo basado en la arquitectura BERT [63] preentrenado específicamente para el idioma español. BETO ha demostrado un rendimiento superior en diversas tareas de procesamiento del lenguaje natural en español, debido a que captura de manera efectiva las particularidades sintácticas y semánticas del idioma [64]. Este modelo presenta una ligera modificación para poder convertir su salida tradicional en una correspondiante a clasificación binaria, tarea que nos compete en esta investigación. Tenemos la misma arquitectura que el modelo base BERT, solo que a esta se le adiciona:

1. **Capa de Dropout:** Se encuentra justamente después de la capa de salida del modelo. Es un método de regularización por lo que evita el sobreajuste del modelo.

2. **Capa lineal:** Permite concentrar la información obtenida del modelo en 768 dimensiones a solo 2 dimensiones. Cada una de estas dimensiones posee un valor probabilístico, predicha por el modelo, de cada etiqueta por cada observación.

En la imagen 4.1 podemos visualizar detalladamente la arquitectura de BERT que se utilizó para la clasificación de oraciones en simple y complejas. Explicación detallada por George Mihaila en el siguiente link

La elección de BETO se justifica por las siguientes razones:

- **Representación Contextual Profunda:** BETO utiliza mecanismos de atención bidireccionales que permiten modelar el contexto, en ambos sentidos, tanto a la

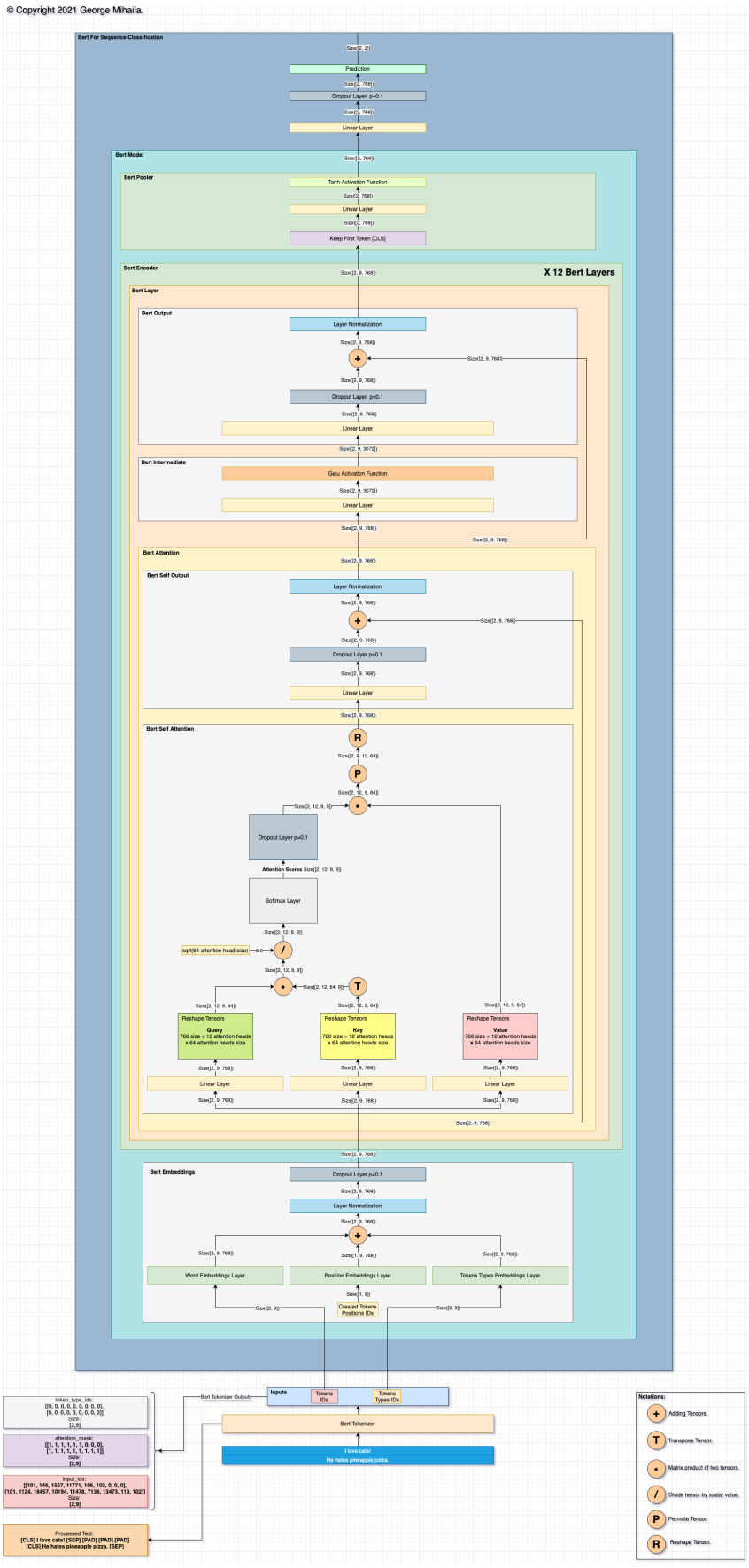

Figura 4.1: Arquitectura de BERT para clasificación de oraciones.

izquierda como a la derecha de una palabra, lo que es esencial para comprender la complejidad textual en el idioma español [63].

- **Preentrenamiento en Corpus Amplio:** BETO fue entrenado en grandes cantidades de texto en español, utilizando el corpus de Wikipedia en español y otras fuentes de texto, sumando aproximadamente 3 mil millones de palabras, lo que le permite tener un amplio conocimiento del lenguaje [64].

- **Rendimiento en Tareas de Clasificación:** Estudios previos han demostrado que BETO supera a otros modelos en tareas de clasificación de texto en español, lo que sugiere que es adecuado para distinguir entre textos simples y complejos [64].

- **Disponibilidad y Comunidad de Soporte:** BETO es de código abierto y está disponible en la plataforma Hugging Face, facilitando su implementación y ajuste fino para tareas específicas [131].

La complejidad textual en español presenta desafíos particulares debido a su rica morfología y sintaxis. Por lo tanto, utilizar un modelo que esté preentrenado en español y que capture estas características es esencial para el éxito de esta investigación.

Ver la sección 3.3.1 para mas información técnica respecto al modelo BETO

4.1.2. Estimación de Incertidumbre

La estimación de la incertidumbre en modelos de aprendizaje profundo es crucial para comprender la confianza del modelo en sus predicciones, lo que es especialmente importante en aplicaciones sensibles donde errores pueden tener consecuencias graves. Existen diversos métodos para estimar la incertidumbre en las predicciones de un modelo, entre los cuales destacan los métodos tradicionales como MCD y DEB, así como metodologías más recientes y avanzadas como la FDE. En esta sección, se describen estos enfoques en detalle y se justifica la elección del método propuesto para este trabajo.

Monte Carlo Dropout para la Estimación de Incertidumbre

El método MCD ha ganado popularidad en el ámbito de la UE en redes neuronales, como se destaca en la investigación [116]. Originalmente, el dropout se ha utilizado como una técnica de regularización, tal como sugieren [132, 133]. La funcionalidad de MCD como herramienta de regularización se basa en la desactivación aleatoria de un conjunto de neuronas durante el entrenamiento de la red, lo que promueve el aprendizaje de características más robustas por parte de las neuronas restantes.

El método MCD fue introducido por [116] como una técnica para estimar la incertidumbre en redes neuronales profundas. Es uno de los métodos mas utilizados en la actualidad para dicha tarea. Este método es una extensión de la técnica de regularización dropout al proceso de inferencia. Mientras que el dropout se utiliza habitualmente durante el entrenamiento para reducir el sobreajuste [132, 133], en MCD se mantiene el dropout activado también durante la fase de inferencia, lo que permite obtener una distribución de predicciones en lugar de una única predicción determinista [116], promoviendo el aprendizaje de características más robustas por parte de las neuronas restantes.

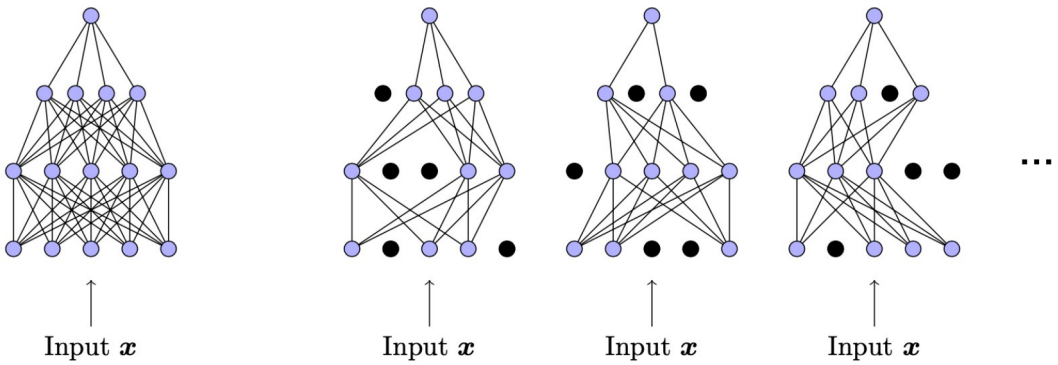

Figura 4.2: Representación del Método MCD. A la izquierda se muestra una NN sin habérsele aplicado MCD, a la derecha se puede observar diferentes instancias de la aplicación del método MCD a dicha NN, se muestra perfectamente la desactivación aleatoria de x neuronas en la red dada una probabilidad P. Tomado de [7]

Funcionamiento y aplicación en este estudio:

En esta aplicación, se habilita el dropout en la fase de inferencia del modelo

$\widetilde{ti} = f(xi)$, donde xi es la entrada y $\widetilde{ti}$ la predicción correspondiente del modelo con parámetros $\boldsymbol{\theta}$. Al generar N predicciones para la misma entrada $\mathbf{x}_a$ con dropout habilitado, la dispersión o entropía de estas predicciones (consideradas como la salida de un ensemble de tamaño N) se utiliza como medida de incertidumbre. Matemáticamente, esto se representa como:

$$t^{(j)} = f_{\boldsymbol{\theta}}(\boldsymbol{x}_a, d) \quad \text{for } j = 1, \ldots, N, \tag{4.1}$$

donde d es la tasa de dropout de inferencia para el modelo f. La densidad $\boldsymbol{p}_{\text{MCD}}$ de la salida del ensemble $\mathbf{t}$ se utiliza para calcular la entropía de salida:

$$H(\boldsymbol{p}_{\text{MCD}}) = -\sum_{i=1}^{N} \left(p_{\text{MCD}}^{(i)} \cdot \log_2 \left(p_{\text{MCD}}^{(i)} \right) \right) \tag{4.2}$$

La cual a su vez se emplea como medida de incertidumbre:

$$\mathcal{U}_{\text{MCD}}(\boldsymbol{x}_a, \boldsymbol{\theta}) = H(\boldsymbol{p}_{\text{MCD}}) \tag{4.3}$$

Deep Ensembles Base para Estimación de Incertidumbre

Conforme a lo descrito por [121], el enfoque de DEB implica la combinación de N modelos de redes neuronales profundas para predecir una observación. Esta técnica, al igual que la UE basada en MCD, utiliza la dispersión o entropía de las predicciones para una entrada dada $\mathbf{x}$ como una forma de UE. Según [121], una estrategia típica para crear un *ensemble* es emplear el método de *bagging*, seleccionando una muestra con reemplazo de la muestra de entrenamiento original. Además, una fuente importante de variabilidad entre los modelos es la inicialización aleatoria de sus pesos, lo que permite, según [55], que la UE de DEB capture tanto la incertidumbre aleatoria como la epistémica.

Funcionamiento y aplicación en este estudio:

Se establece una semilla que modifica los valores iniciales de los parámetros entrenables de cada modelo. En la fase de inferencia, aplicamos la técnica DEB para el modelo $\widetilde{t}_i = f(\boldsymbol{x}_i)$, considerando la entrada $\boldsymbol{x}_i$ y su correspondiente predicción $\widetilde{t}_i$ con parámetros $\boldsymbol{\theta}$. Utilizando DEB, entrenamos N modelos distintos usando N particiones aleatorias diferentes, lo que resulta en un conjunto de pesos $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N$. Al evaluar una entrada $\boldsymbol{x}_a$, la dispersión o entropía de estas N predicciones (similar

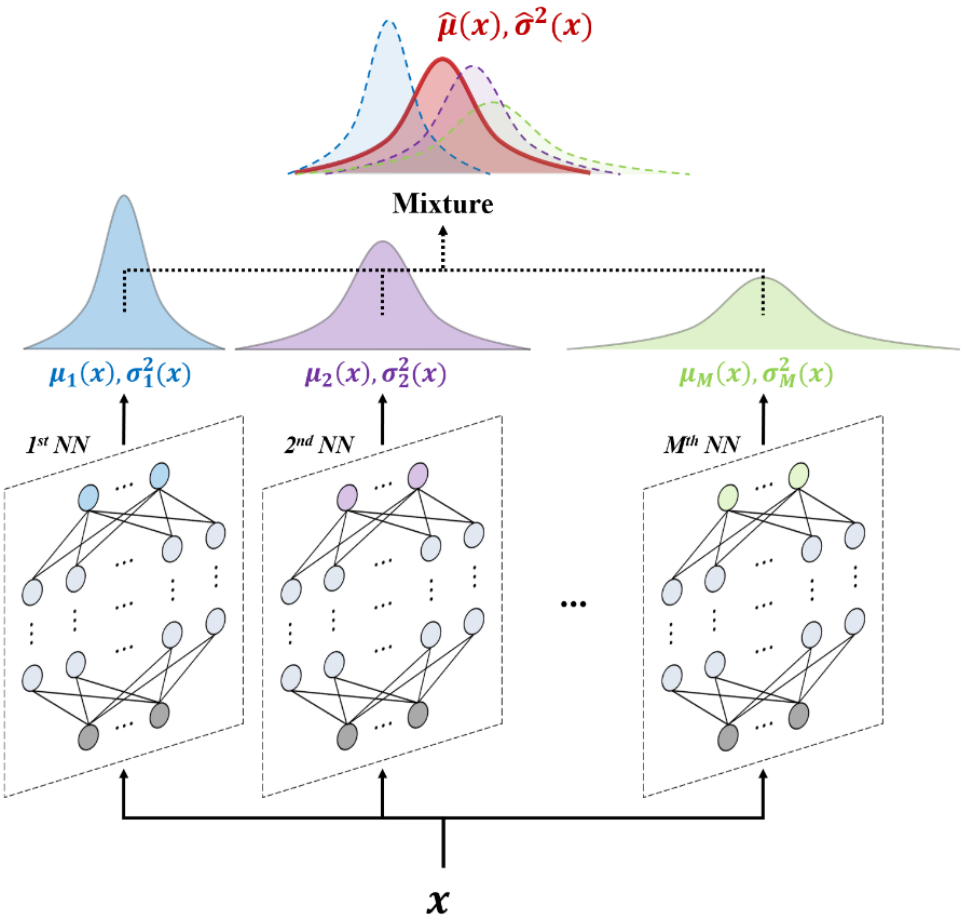

Figura 4.3: Representación del Método DEB. De izquierda a derecha podemos observar la representación de las diferentes instancias del modelo utilizado. Se muestra además, que las distribuciones resultantes son diferentes entre si debido a la inicialización aleatoria de los pesos del modelo. Tomado de [8]

a MCD, interpretada como la salida de un ensamble de tamaño N) se utiliza como medida de incertidumbre [121], descrita formalmente como:

$$t^{(j)} = f_{\theta_j}(\boldsymbol{x}_a) \quad \text{para } j = 1, \ldots, N \tag{4.4}$$

Calculamos la entropía de la salida del ensamble $\mathbf{t}$ a partir de la densidad $\boldsymbol{p}_{\text{DEB}}$:

$$H\left(\boldsymbol{p}_{\text{DEB}}\right) = -\sum_{i=1}^{N}\left(p_{\text{DEB}}^{(i)} \cdot \log_2\left(p_{\text{DEB}}^{(i)}\right)\right) \tag{4.5}$$

lo cual se interpreta como el puntaje de incertidumbre:

$$\mathcal{U}_{\text{DEB}}\left(\boldsymbol{x}_a, \boldsymbol{\theta}\right) = H\left(\boldsymbol{p}_{\text{DEB}}\right) \tag{4.6}$$

Método Propuesto: Estimación de Densidades de Características (FDE) en la clasificación de la complejidad textual

Aunque métodos como MCD y DEB han sido exitosos en estimar la incertidumbre, también tienen limitaciones. MCD puede incrementar significativamente el tiempo de inferencia debido a la necesidad de múltiples pasadas, mientras que entrenar múltiples modelos en DEB puede ser costoso computacionalmente. Para abordar estas limitaciones, [33] introdujeron la **Estimación de Densidades de Características (FDE)** como un método eficiente para estimar la incertidumbre.

Definición Detallada y Justificación:

La FDE se basa en la idea de que las representaciones latentes producidas por una red neuronal para sus datos de entrenamiento representan la distribución interna de características del modelo. Si podemos modelar la densidad de estas características internas para los datos de entrenamiento, entonces al evaluar nuevos datos podemos inferir cuán similares son estos nuevos datos a los datos de entrenamiento en el espacio latente. Cuanto menos similar sea un nuevo dato a los datos de entrenamiento, mayor será la incertidumbre en la predicción del modelo.

La incertidumbre es especialmente relevante en modelos complejos como BETO, donde la calidad de las representaciones internas de los datos influye de manera crucial en la eficacia de la modelación [33, 34]. Con el fin de cuantificar la incertidumbre

en este contexto, se propone un método basado en la estimación de densidades de características de las representaciones latentes generadas por BETO. Este método, FDE, aprovecha las distribuciones aprendidas internamente por el modelo para evaluar la confianza en sus predicciones.

Representación del Conjunto de Datos y el Objetivo del Modelo: Dado un conjunto de datos $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$, el modelo BETO se entrena para modelar la distribución condicional $p(\mathbf{y}|\mathbf{x})$, donde cada entrada $\mathbf{x} \in \mathbf{X}$ se asocia con una etiqueta $\mathbf{y} \in \mathbf{Y}$. La tarea del modelo implica capturar la complejidad y la incertidumbre inherentes a la predicción de $\mathbf{y}$ dado un $\mathbf{x}$ específico. Para lograr esto, el modelo genera representaciones latentes $\mathbf{z}_i$ en cada capa, las cuales encapsulan características relevantes de la entrada.

Análisis de las Representaciones Latentes: Se parte de la premisa de que la incertidumbre del modelo puede inferirse a partir de las representaciones latentes generadas por sus capas intermedias. Supongamos que una red neuronal profunda está compuesta por L capas. El modelo produce, entonces, $L-1$ conjuntos de representaciones latentes: $\{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{L-2}\}$, donde cada $\mathbf{z}_i$ es la salida de la capa i.

Procedimiento detallado del método:

1. **Obtención de las Representaciones Latentes del Conjunto de Entrenamiento:** Para los datos de entrenamiento, se realiza la inferencia con el modelo entrenado para obtener un conjunto de representaciones latentes:

$$\mathbf{Z}_{\text{train}} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}.$$

   Aquí, $\mathbf{z}_i \in \mathbb{R}^d$ representa la representación latente para la i-ésima instancia del conjunto de entrenamiento, y d es la dimensionalidad del espacio latente. Este paso permite capturar cómo el modelo representa internamente la información de los datos en sus dimensiones latentes.

2. **Construcción de Densidades de Probabilidad en las Dimensiones Latentes:** Para modelar la distribución de las representaciones latentes en el espacio del conjunto de entrenamiento, se lleva a cabo el siguiente procedimiento:

a) **Dividir cada Dimensión Latente en Bins:** Cada representación latente $\mathbf{z}_i$ tiene una dimensionalidad d. Para cada dimensión $j \in \{1, 2, \ldots, d\}$, se divide el rango de valores observado en la dimensión j en K *bins* de igual tamaño:

$$\{b_{j,1}, b_{j,2}, \ldots, b_{j,K}\}.$$

b) **Construir Histogramas de Frecuencias:** Se cuenta cuántos valores de las representaciones latentes $\mathbf{Z}_{\text{train}}$ caen en cada bin k para cada dimensión j. Estas frecuencias se normalizan para obtener densidades de probabilidad, resultando en:

$$p(z_{j,k}) = \frac{n_{j,k}}{N},$$

donde $n_{j,k}$ es el número de valores en la dimensión j que caen en el bin k, y N es el número total de observaciones.

Esto nos da una aproximación a la distribución de las representaciones latentes en cada dimensión para el conjunto de entrenamiento.

3. **Evaluación de Nuevos Datos Utilizando las Densidades de Características:** Para un nuevo conjunto de datos de prueba, denotado como $\mathbf{Z}_{\text{test}} = \{\mathbf{z}_1^*, \mathbf{z}_2^*, \ldots, \mathbf{z}_{N^*}^*\}$, se aplican los siguientes pasos para estimar la incertidumbre:

a) **Obtener las Representaciones Latentes:** Se pasa cada instancia nueva a través del modelo para obtener sus representaciones latentes $\mathbf{z}_i^*$ en el mismo espacio latente de dimensionalidad d que las del conjunto de entrenamiento.

b) **Buscar el Bin Más Cercano:** Para cada valor de característica z_j^* en la representación latente $\mathbf{z}_i^*$, se identifica el bin más cercano calculando la diferencia absoluta entre z_j^* y los bordes de los bins $b_{j,k}$:

$$d_{j,k} = |z_j^* - b_{j,k}|.$$

El índice del bin más cercano es aquel que minimiza esta diferencia:

$$\mathrm{B}_j = \arg\min_k d_{j,k}.$$

c) **Obtener la Densidad de Probabilidad para Cada Dimensión:** Para cada dimensión j, la densidad de probabilidad asociada al bin más cercano se obtiene de las densidades calculadas en el entrenamiento:

$$\lambda_j = p(z_{\mathrm{B}_j}) + \epsilon,$$

donde ϵ es una pequeña constante (por ejemplo, $\epsilon = 1e^{-8}$) para evitar problemas con el logaritmo de cero en pasos posteriores.

d) **Cálculo de la Probabilidad Conjunta:** Asumiendo independencia entre las dimensiones latentes (lo cual es una simplificación), se calcula la sumatoria logarítmica acumulada de las densidades en todas las dimensiones j:

$$\mathcal{L}_i = \sum_{j=1}^{d} \log(\lambda_j).$$

Aquí, $\mathcal{L}_i$ es el logaritmo de la probabilidad conjunta de que la representación latente para la instancia i del conjunto de prueba provenga de la distribución de las representaciones del conjunto de entrenamiento.

e) **Determinación del Puntaje de Incertidumbre:** El puntaje de incertidumbre S se define como el negativo de la suma de las log-probabilidades acumuladas para todas las observaciones i en $\mathbf{Z}_{\text{test}}$:

$$S = -\sum_{i=1}^{N^*} \mathcal{L}_i.$$

Este puntaje refleja cuán fuera de distribución están las nuevas observaciones en comparación con el conjunto de entrenamiento. Un puntaje más alto indica que las observaciones se alejan más del comportamiento latente esperado por el modelo, lo que implica mayor incertidumbre.

Evaluación del Método y Aplicación a la Detección de Textos Complejos

La eficacia del método FDE para la UE se evaluará comparando las distribuciones de probabilidad de las clasificaciones correctas e incorrectas mediante la métrica JSD [3, 123](mas información ver la sección 4.1.3). Al examinar cómo difieren las

distribuciones de incertidumbre entre predicciones correctas e incorrectas, se puede determinar cuán bien el método distingue los casos en que el modelo está seguro de aquellos en que no lo está.

Esta aproximación busca optimizar la detección de textos que requieren simplificación, minimizando el uso de recursos en aquellos suficientemente claros. Al mejorar la eficiencia en la detección de textos complejos, también abre camino para futuras investigaciones en la simplificación automática de textos en español, centrándose en la mejora continua tanto de la calidad de los datos como del rendimiento del modelo.

Ventajas del método FDE

1. **Eficiencia Computacional**
   - A diferencia de MCD y DEB, la FDE no requiere múltiples pasadas durante la inferencia ni entrenar múltiples modelos. Sólo necesita una evaluación del modelo principal y un cálculo de la densidad en el espacio latente.
   - Esto hace que la UE sea más rápida y menos costosa, especialmente en aplicaciones en tiempo real.
2. **Versatilidad**
   - FDE puede aplicarse a cualquier modelo que proporcione representaciones latentes. En este caso, se aplica a las representaciones generadas por BETO.
   - Permite la detección de datos fuera de distribución y casos que podrían ser más complejos de lo que el modelo ha visto durante el entrenamiento.
3. **Relevancia para la Detección de Textos Complejos**
   - Este método es particularmente relevante en el contexto de esta tesis. Dado que el modelo se entrenará principalmente en textos de educación financiera, los cuales pueden ser complejos, las representaciones latentes para textos fuera de este dominio podrían resultar inusuales.
   - La FDE permitirá detectar estos casos de alta complejidad o datos de dominios desconocidos, haciendo más confiable la clasificación y simplificación de textos.

4. **Adaptabilidad Post-Entrenamiento**

    - La estimación de densidades de características se puede refinar incluso después de que el modelo ha sido entrenado, ya que la distribución de características se puede actualizar con nuevos datos. Esto es útil si el modelo enfrenta datos que cambian con el tiempo, un escenario común en aplicaciones reales.

La metodología FDE combina interpretabilidad y eficiencia, alineándose con los objetivos de esta tesis al proporcionar una estimación de la incertidumbre robusta y eficiente. Con su ayuda, se espera mejorar la fiabilidad del sistema de clasificación de textos simples y complejos en español, optimizando así la detección de textos que requieren simplificación.

4.1.3. Evaluación de la fiabilidad de las estimaciones de incertidumbre

En la literatura existen diversos enfoques para comparar la fiabilidad de las estimaciones de incertidumbre, como se revisa en la sección 3.4. Para un clasificador binario, es posible medir la distancia entre las distribuciones de las estimaciones de incertidumbre UE para las predicciones correctas e incorrectas. Como se propone en [37], la JSD se emplea como una métrica efectiva para esta comparación.

El procedimiento se establece de la siguiente manera: se consideran los conjuntos de UE para las estimaciones correctas e incorrectas, denotados por $U_{\text{incorrect}} = \{\mathcal{U}_1, \ldots, \mathcal{U}_M\}$ y $U_{\text{correct}} = \{\mathcal{U}_1, \ldots, \mathcal{U}_M\}$ respectivamente. Aquí, M representa el número total de estimaciones en ambos conjuntos de UE, garantizado mediante un muestreo equilibrado. Con estos conjuntos, estimamos las densidades $\boldsymbol{p}_{\text{correct}}$ y $\boldsymbol{p}_{\text{incorrect}}$ mediante el cálculo de histogramas normalizados. La métrica de fiabilidad para el método de UE se define entonces como la JSD entre las densidades $\boldsymbol{p}_{\text{correct}}$ y $\boldsymbol{p}_{\text{incorrect}}$:

$$d_{\text{JS}}\left(\boldsymbol{p}_{\text{correct}}, \boldsymbol{p}_{\text{incorrect}}\right) \tag{4.7}$$

Se utiliza esta métrica para comparar los métodos implementados de UE. La JSD entre dos distribuciones de probabilidad, $\boldsymbol{p}_{\text{correct}}$ y $\boldsymbol{p}_{\text{incorrect}}$, se define de la siguiente manera:

$$d_{\mathrm{JS}}\left(\boldsymbol{p}_{\text{correct}}, \boldsymbol{p}_{\text{incorrect}}\right) = \frac{\mathrm{KL}(\boldsymbol{p}_{\text{correct}}||m) + \mathrm{KL}(\boldsymbol{p}_{\text{incorrect}}||m)}{2} \tag{4.8}$$

Donde m es la densidad promedio:

$$m = \frac{\boldsymbol{p}_{\text{correct}} + \boldsymbol{p}_{\text{incorrect}}}{2}. \tag{4.9}$$

La divergencia Kullback-Leibler entre $\boldsymbol{p}_{\text{correct}}$ y $\boldsymbol{p}_{\text{incorrect}}$, $\mathrm{KL}(\boldsymbol{p}_{\text{correct}}||\boldsymbol{p}_{\text{incorrect}})$, se define como:

$$\mathrm{KL}(\boldsymbol{p}_{\text{correct}}||\boldsymbol{p}_{\text{incorrect}}) = \sum \boldsymbol{p}^{(i)}_{\text{correct}} \cdot \log\left(\frac{\boldsymbol{p}^{(i)}_{\text{correct}}}{\boldsymbol{p}^{(i)}_{\text{incorrect}}}\right). \tag{4.10}$$

El objetivo radica en maximizar la JSD para las densidades de UE entre las predicciones correctas e incorrectas. Al maximizar esta distancia, se asegura que las dos distribuciones comparadas sean distintas entre sí, lo cual es útil en tareas donde se desea una clara diferenciación entre dos distribuciones [134].

4.2. Conjunto de Datos

En español, existen pocos conjuntos de datos disponibles para la simplificación de textos. Destacan principalmente el conjunto de datos SIMPLEXT, incluye 200 artículos periodísticos simplificados de forma manual por expertos, con el objetivo de facilitar la comprensión para personas con problemas de aprendizaje [135]. Además tenemos el corpus de datos Aligned Newsela, contiene alrededor de 1.221 documentos, que luego de la limpieza y alineación de las oraciones se obtuvieron 55.890 pares de oraciones como referencia en español, el mismo no se encuentra disponible para su utilización en la investigación [9, 136]. En la tabla 4.1 se puede observar otros corpus de datos en español que se han desarrollado.

Dataset	N^o**. Instancias**	**Alineado**	**Dominio Específico**	**Población Específica**
ALEXSIS	381	Si	No	No
EASIER	5130	Si	No	No
EASIER-500	500	Si	No	No
Newsela	20000	Si	No	No
Med-EASi	1979	Si	Textos Médicos	No

Cuadro 4.1: Ejemplos de conjuntos de datos desarrollados para el idioma español. Tomado de [9]

La elaboración de conjuntos de datos destinados principalmente a la simplificación de textos en idiomas particularmente diferentes al inglés, enfrenta duros retos [9]. El primer aspecto a destacar es la escasez de conjuntos de datos amplios y de alta calidad en el idioma de interés, una situación que limita considerablemente la eficiencia de los modelos diseñados para la simplificación de textos en dicho idioma [9]. Por otra parte, las fluctuaciones en cuanto a la calidad y uniformidad presente en los datos incide negativamente en la capacidad de aprendizaje de los modelos [9]. Además, centrándonos en la comparación que se muestra en la tabla 4.1, se expone la limitada disponibilidad de conjuntos de datos específicos de dominio y centrados en una población objetivo, específicamente para la TS en español.

En [9], se plantea un conjunto de datos novedoso, el mismo va a ser utilizado en esta investigación. Este conjunto de datos consta de 5314 segmentos de texto en español, extraídos de 4 libros de educación financiera [137–140]. Un segmento de texto es definido como un fragmento de texto que esta separado del texto del siguiente segmento por los siguientes signos de puntuación: 1) punto, 2) punto y coma, 3) signo de interrogación de cierre y 4) signo de exclamación de cierre [9]. Cada uno de estos segmentos cuenta con una versión simplificada correspondiente. La simplificación de cada fragmento de texto se realizó manualmente por estudiantes avanzados de filología, siguiendo un conjunto de reglas establecidas para orientarlos en el proceso de simplificación. Por tanto, nuestro conjunto de datos final incluye un total de 5314 pares de segmentos de texto, cada par compuesto por una versión compleja y su equivalente simplificado [9, 12].

En las subsección 4.2.1 se describe las principales diferencias entre segmentos de texto simple y complejo, con el objetivo de brindar una mayor claridad sobre este ***"dataset"*** que será utilizado en los diferentes experimentos propuestos en esta investigación.

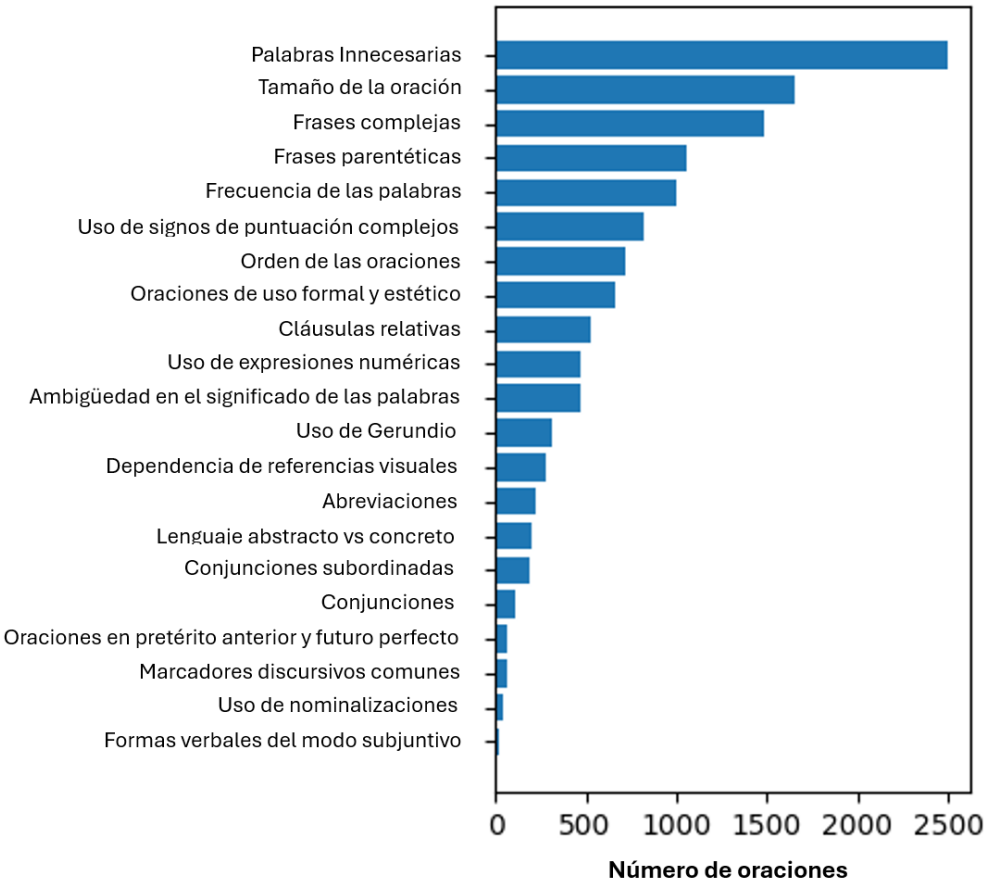

Figura 4.4: Histograma con las reglas de simplificación utilizadas para generar el conjunto de datos que se utilizará en esta investigación. Tomado de [9]

4.2.1. Definición de la Complejidad Textual

La complejidad textual emerge de la densidad de ideas, la sofisticación del lenguaje, o la estructura confusa de los argumentos presentados. Un texto complejo desafía al lector, invitándolo a una exploración más profunda y a menudo requiere de una reflexión y análisis más detallados para comprender más a fondo su significado. Esta se define por múltiples factores que afectan la legibilidad y comprensión del contenido. Textos con oraciones largas y estructuras sintácticas complejas, alto número de palabras técnicas, uso de formas verbales complejas y omisión de elementos que

facilitan la comprensión del texto, son algunos de los elementos que califican a un texto con una complejidad alta [9]. A continuación, se presentan los aspectos más frecuentes, ver figura 4.4, que contribuyen a esta complejidad y como tenerlos en cuenta para lograr una simplificación efectiva. Se presentan ejemplos ilustrativos en versiones simples y complejas para mayor entendimiento:

1. **Palabras innecesarias:** Eliminar palabras y frases innecesarias en un texto ayuda a clarificar el mensaje y facilitar su comprensión. Este proceso implica identificar y descartar elementos que no agregan valor significativo al contenido, redundancias, o detalles excesivos que pueden desviar la atención del lector del punto central. Al hacerlo, el texto se vuelve más conciso, directo y accesible, mejorando así la eficiencia de la comunicación [9].

    - Ejemplos:
        - **Complejo:** "**Es evidente** que existen muchas posibilidades de inversión que podrían considerarse beneficiosas."
        - **Simplificado:** "Existen inversiones beneficiosas."

2. **Longitud de la oración:** Usar segmentos de 20 palabras o menos y dividir el segmento en varios segmentos es una técnica recomendada para simplificar la información y mejorar la comprensión del lector. Esto permite presentar ideas de manera más clara y concisa, facilitando la absorción de información, especialmente en contextos complejos o técnicos. Al desglosar la información en unidades más pequeñas, se incrementa la legibilidad y se hace más accesible al público, permitiendo una mejor retención de los datos presentados [9].

    - Ejemplos:
        - **Complejo:** "La empresa, después de una larga deliberación y teniendo en cuenta varias variables económicas, decidió incrementar su capital mediante la emisión de nuevas acciones."
        - **Simplificado:** "La empresa decidió aumentar su capital emitiendo nuevas acciones."

3. **Frases complejas:** Reemplazar expresiones léxicas complejas por sus equivalentes de una sola palabra implica simplificar el texto para hacerlo más accesible.

Este proceso consiste en identificar términos o frases largas que pueden ser condensadas en una sola palabra sin perder su significado original. Por ejemplo, la frase **“en el momento de"** puede ser reemplazada por **“cuando"**, o **“debido al hecho de que"** por **“porque"** [9].

- Ejemplos:
  - **Complejo:** “A pesar de la volatilidad del mercado, los inversores mantienen una actitud optimista, confiando en la recuperación económica."
  - **Simplificado:** “Los inversores son optimistas sobre la recuperación económica, a pesar de la volatilidad del mercado."

4. **Frases parentéticas:** Son aquellas que se insertan en medio de otras frases, ofreciendo información adicional, aclaraciones o ejemplos, sin alterar el significado principal de la oración. Su uso se marca típicamente por comas, paréntesis o guiones. Aunque pueden enriquecer el texto con detalles o contextos adicionales, es recomendable reservar su uso para cuando sean esenciales para la comprensión del mensaje. Esto se debe a que un exceso de frases parentéticas puede hacer que el texto sea difícil de seguir, distraer al lector de los puntos principales o sobrecargar la oración con información no crucial [9].

   - Ejemplos:
     - **Complejo:** “El rendimiento de los bonos, **que es un indicador clave de la salud económica**, ha empezado a estabilizarse."
     - **Simplificado:** “El rendimiento de los bonos ha empezado a estabilizarse."

5. **Frecuencia de las palabras:** Se refiere a la cantidad de veces que aparece una palabra en un texto específico o en un corpus de textos. Reemplazar palabras de baja frecuencia por palabras de uso común es una técnica utilizada para simplificar textos y mejorar su comprensibilidad. Las palabras que aparecen raramente pueden ser desconocidas o difíciles de entender para muchos lectores. Al sustituirlas por sinónimos o términos equivalentes más comunes, se facilita

la lectura y comprensión del texto, haciendo la información más accesible para un público más amplio [9].

- Ejemplos:
  - **Complejo:** "La liquidez, un término frecuentemente asociado con la facilidad para convertir activos en efectivo, es crucial."
  - **Simplificado:** "La liquidez, la facilidad de convertir activos en efectivo, es crucial."

6. **Uso de signos de puntuación complejos:** Se refiere a la inclusión de elementos como punto y coma, corchetes, guiones, comillas simples y comillas angulares en los textos. Estos signos pueden aumentar la complejidad textual, dificultando la lectura y comprensión del contenido. La recomendación para simplificar el texto incluye evitar el uso de estos signos complejos, siempre y cuando sea posible. Sin embargo, es importante no reemplazar indiscriminadamente el punto y coma por un punto, especialmente cuando se utiliza para separar palabras o frases en una enumeración de elementos léxicos, manteniendo así la claridad y precisión del significado sin aumentar innecesariamente la complejidad [9].

   - Ejemplos:
     - **Complejo:** "La inversión, especialmente en oro y plata; criptomonedas, como Bitcoin y Ethereum; y bienes raíces, sigue siendo popular."
     - **Simplificado:** "La inversión en oro, plata, criptomonedas y bienes raíces es popular."

Teniendo en cuenta la información presentada anteriormente podemos llegar a la conclusión que la complejidad de un texto se refiere a cuán denso y sofisticado es el contenido que se encuentra en él, la cual afecta directamente su legibilidad y comprensión. Surge de factores múltiples como el uso de palabras innecesarias, la longitud y estructura de las oraciones, el empleo de frases complejas o parentéticas, entre otras (ver figura 4.4).

La complejidad textual no solo impide la accesibilidad para personas con discapacidades visuales, cognitivas o que estén aprendiendo un nuevo idioma, sino que

también puede afectar a lectores con diferentes niveles de habilidad lectora. Identificar y ajustar estos factores es crucial para mejorar la legibilidad y comprensión de los textos [9, 12].

Estas métricas ofrecen una base cuantitativa para diferenciar entre textos simples y complejos, permitiendo adaptaciones más accesibles para diversos públicos [9].

4.3. Recursos Utilizados (Software y Hardware)

La elección cuidadosa de los recursos de hardware y software es una piedra angular en la realización de experimentos científicos y técnicos. La descripción detallada de estos recursos no es meramente administrativa; es fundamental para la integridad y la replicabilidad de la investigación.

Para la realización de los diversos experimentos presentados en esta investigación se utilizó **Google Colaboratory** en su versión **Pro**.

4.3.1. Recursos de Hardware

Para llevar a cabo los experimentos detallados en esta tesis, se utilizó un conjunto específico de recursos de hardware. La configuración de hardware no solo permitió el procesamiento eficiente de los datos sino que también aseguró la capacidad para ejecutar modelos de aprendizaje profundo complejos. A continuación, se describen los componentes de hardware clave:

- **GPU V100 con 16 GB de VRAM**: Este componente fue esencial para el entrenamiento de los modelos de inteligencia artificial, ofreciendo una capacidad de cómputo significativa y acelerando el proceso de aprendizaje automático.
- **12.7 GB de RAM**: La memoria RAM permitió manejar grandes conjuntos de datos en memoria durante el procesamiento y análisis, facilitando operaciones eficientes y reduciendo el tiempo de ejecución.
- **80 GB de almacenamiento**: Este espacio fue usado para almacenar los conjuntos de datos utilizados, así como los modelos generados durante el entrenamiento. Esta capacidad de almacenamiento garantizó que los recursos estuvieran disponibles localmente, mejorando el acceso y la manipulación de los datos.

4.3.2. Recursos de Software

En el desarrollo de los experimentos presentados en esta tesis, se ha hecho uso de una serie de herramientas de software específicas, fundamentales para la ejecución y análisis de los datos obtenidos. A continuación, se detallan los paquetes de software y bibliotecas empleadas, así como su propósito dentro del marco experimental.

Paquetes Utilizados

Para asegurar la configuración adecuada del entorno de desarrollo y ejecución, se realizaron las siguientes instalaciones mediante el gestor de paquetes *pip*:

- **PyTorch:** Una biblioteca de aprendizaje automático que facilita tanto la construcción de redes neuronales como su entrenamiento, aprovechando la potencia de las GPUs. Es la base para implementar y entrenar los modelos de NLP utilizados.
- **Transformers (versión 4.30.2):** Esta biblioteca, desarrollada por Hugging Face, es esencial para NLP. Provee acceso a modelos pre-entrenados como BERT, facilitando tareas complejas como la clasificación de secuencias y el modelado de lenguaje. La versión específica asegura la compatibilidad y reproducibilidad de los experimentos.
    - **Torch Utils y Transformers:** Proporcionan componentes adicionales para la carga de datos (DataLoader), tokenización, modelado y entrenamiento (BertForMaskedLM, BertTokenizer, BertForSequenceClassification, TrainingArguments, Trainer, etc.), facilitando una amplia gama de tareas de NLP con modelos pre-entrenados y personalizados.
- **Accelerate:** Una biblioteca diseñada para simplificar el uso de aceleración de hardware (GPU) en el entrenamiento de modelos, permitiendo ejecuciones más eficientes sin necesidad de un código complejo para manejar múltiples GPUs o CPUs.
- **Evaluate:** Desarrollado también por Hugging Face, este paquete se utiliza para evaluar modelos de NLP con una variedad de métricas estándar, proporcionando un marco coherente y comparativo para la evaluación de rendimiento.

- **Weights & Biases (wandb):** Herramienta de seguimiento de experimentos que permite la visualización en tiempo real del entrenamiento de modelos, el registro de métricas, y la comparación entre diferentes ejecuciones. Es crucial para el monitoreo del progreso y la optimización de modelos.

- **NumPy y Pandas:** Herramientas fundamentales para el análisis de datos en Python. NumPy ofrece soporte para arrays y matrices de gran tamaño, mientras que Pandas proporciona estructuras de datos de alto nivel y funciones para manipulación eficiente de datos tabulares.

- **Datasets:** Otro paquete de Hugging Face que simplifica la carga, manipulación y preprocesamiento de conjuntos de datos para NLP, optimizando la eficiencia y la facilidad de uso en tareas de entrenamiento y evaluación de modelos.

- **Sklearn (Scikit-learn):** Proporciona una amplia gama de herramientas para el modelado predictivo y análisis de datos, incluyendo métricas como F1-score, precisión, recall y exactitud, esenciales para la evaluación de los modelos de clasificación.

- **PyArrow:** Utilizada para la serialización y deserialización eficiente de estructuras de datos, PyArrow facilita el manejo de grandes volúmenes de datos, especialmente en formatos columnares optimizados para operaciones rápidas de E/S.

- **Wandb (Weights & Biases):** Importado para integrar directamente en el código las funcionalidades de seguimiento de experimentos, permitiendo una gestión detallada del entrenamiento y evaluación de modelos dentro del mismo entorno de desarrollo.

El modelo BETO utilizado para la discriminación entre textos simples y complejos se puede encontrar en el siguiente link

La selección de estas herramientas y bibliotecas fue determinada por su robustez, flexibilidad y amplia aceptación en la comunidad científica, asegurando así la fiabilidad y reproducibilidad de los experimentos realizados. Cada componente software ha sido crucial para abordar las distintas fases del proyecto, desde la preparación de

los datos hasta el entrenamiento y evaluación de modelos de inteligencia artificial avanzados.

4.4. Conclusiones

En este capítulo se ha detallado la metodología propuesta para abordar la clasificación de textos simples y complejos en español, haciendo uso de modelos de lenguaje basados en *transformers* y técnicas avanzadas de estimación de incertidumbre. A continuación, se resumen los aspectos clave presentados:

- **Implementación de BETO para la Clasificación de Textos:** Se seleccionó el modelo BETO, una implementación de BERT en español, como base para la clasificación de textos. Se justificó esta elección debido a su capacidad para capturar las particularidades sintácticas y semánticas del idioma español, y su rendimiento superior en tareas de procesamiento del lenguaje natural en español. Se adaptó la arquitectura de BETO para la tarea específica de clasificación binaria, incorporando una capa de dropout y una capa lineal para obtener las predicciones.

- **Desarrollo de un Método de Estimación de Incertidumbre Basado en FDE:** Además de implementar técnicas tradicionales de estimación de incertidumbre como MCD y DEB, se propuso un método novedoso basado en la **Estimación de Densidades de Características (FDE)**. Este enfoque aprovecha las representaciones latentes generadas por BETO para estimar la densidad de probabilidad de las características y cuantificar la incertidumbre asociada a cada predicción. Se detalló el procedimiento del método, resaltando sus ventajas en términos de eficiencia computacional y adaptabilidad.

- **Evaluación de la Fiabilidad de las Estimaciones de Incertidumbre:** Se estableció el uso de la JSD como métrica para evaluar la fiabilidad de las estimaciones de incertidumbre. Este enfoque permite comparar las distribuciones de incertidumbre entre predicciones correctas e incorrectas, proporcionando una medida cuantitativa de la capacidad del modelo para distinguir entre casos seguros e inseguros.

- **Descripción del Conjunto de Datos y Definición de Complejidad Textual:** Se utilizó un conjunto de datos propio compuesto por 5,314 pares de segmentos de texto en español del dominio de la educación financiera, cada uno con su versión simplificada. Se discutieron las características que contribuyen a la complejidad textual, como el uso de palabras innecesarias, frases complejas y signos de puntuación avanzados. Esta comprensión profunda de la complejidad textual en español fundamenta la relevancia y aplicabilidad del modelo propuesto.

- **Recursos de Hardware y Software:** Se detallaron los recursos computacionales y las herramientas de software utilizadas para llevar a cabo los experimentos, asegurando la reproducibilidad y transparencia de la investigación. El uso de Google Colaboratory Pro, junto con bibliotecas como PyTorch y Transformers, permitió el entrenamiento eficiente de los modelos y la gestión adecuada del conjunto de datos.

La metodología presentada establece una base sólida para los experimentos y análisis posteriores. Al integrar un modelo potente como BETO con técnicas avanzadas de estimación de incertidumbre, se espera mejorar la precisión y confiabilidad en la clasificación de textos complejos en español. El método propuesto de FDE ofrece una alternativa eficiente a los enfoques tradicionales, potencialmente reduciendo el costo computacional y manteniendo o mejorando el desempeño en la estimación de incertidumbre.

Además, la consideración cuidadosa de la complejidad textual y la utilización de un conjunto de datos específico del dominio aseguran que el modelo esté adaptado a las particularidades del idioma y del contexto aplicado. Esto no solo contribuye a la relevancia práctica de la investigación, sino que también amplía el conocimiento en el área de procesamiento de lenguaje natural en español.

En los capítulos siguientes, se llevará a cabo una evaluación exhaustiva de los métodos implementados, comparando su desempeño y analizando su eficacia en la estimación de incertidumbre. Los resultados obtenidos permitirán validar las hipótesis planteadas y determinar las contribuciones significativas de esta investigación al campo de la simplificación automática de textos y la estimación de incertidumbre en modelos de lenguaje en español.

5. Evaluación de Incertidumbre en Clasificación de Complejidad Textual

5.1. Introducción

En este capítulo se presenta la evaluación de diversos métodos de UE en la clasificación de complejidad textual en español. La finalidad es validar los **Objetivos Específicos** presentados en la Sección 1.3 y la hipótesis planteada en la Sección 1.3.1. Se analizan tres enfoques principales para la estimación de incertidumbre: MCD, DEB y el método propuesto FDE.

> **Objetivo Específico 1**: *Implementar un modelo de aprendizaje profundo para la detección de textos complejos en español.*
>
> **Objetivo Específico 2**: *Proponer al menos un método novedoso de UE en la detección de texto complejo en español basado en FDE.*
>
> **Objetivo Específico 3**: *Proponer una métrica para cuantificar la confiabilidad de las estimaciones de incertidumbre en la clasificación de texto complejo en español.*

Hipótesis

El método FDE mejora significativamente la estimación de incertidumbre en el modelo BETO en comparación con las técnicas tradicionales de MCD y DEB. Se espera que FDE proporcione una medida de incertidumbre más informativa y confiable, demostrando superioridad estadísticamente significativa al correlacionarse de manera más precisa con el rendimiento del modelo y detectar datos fuera de distribución de forma más efectiva y con menos recursos computacionales.

Este capítulo se estructura en las siguientes secciones: se presenta el diseño experimental, se describen los métodos de estimación de incertidumbre implementados,

se exponen los resultados obtenidos y su análisis estadístico, se realiza una comparación y discusión de los resultados, incluyendo un análisis del costo computacional, y finalmente se presentan las conclusiones.

5.2. Diseño Experimental

Esta sección describe el diseño experimental seguido para evaluar y comparar la fiabilidad de los tres enfoques de UE: MCD, DEB y FDE.

5.2.1. Conjunto de Datos y Configuración Inicial

Se utiliza el conjunto de datos detallado en la Subsección 4.2, que consta de 5314 pares de segmentos de texto en español etiquetados como simples o complejos. Cada par representa un texto complejo y su versión simplificada, proveniente del dominio de la educación financiera. El conjunto de datos se divide en 10 particiones, asignando un 90 % para entrenamiento y validación, y un 10 % para prueba en cada partición.

5.2.2. Entrenamiento y Ajuste Fino de BETO

Siguiendo el **Objetivo Específico 1**, se implementa y ajusta finamente el modelo BETO, entrenado originalmente para predecir palabras faltantes en textos en español y adaptado en esta investigación para la tarea de clasificación binaria (texto simple vs. complejo):

- **Selección de Hiperparámetros:**
  - *Número de épocas:* 6.
  - *Tasa de aprendizaje:* $3{,}42 \times 10^{-5}$.
  - *Decaimiento de peso:* 0.4.
  - *Tamaño de lote:* 8.
  - *Optimizador:* AdamW con un *learning rate scheduler* lineal.
  - *Métrica para cargar el mejor modelo:* Pérdida en validación.
  - *Estrategia de evaluación:* Épocas.

- **Estrategia de Detención Temprana:** Se utiliza la técnica de *early stopping* con una paciencia de 2 épocas para prevenir el sobreajuste del modelo. Esta técnica detiene el entrenamiento cuando no hay mejora en la pérdida de validación y carga el mejor modelo obtenido.

- **Monitoreo y Evaluación:** La métrica de pérdida en validación (*eval_loss*) se utiliza para evaluar el rendimiento del modelo en el conjunto de validación, permitiendo visualizar el nivel de generalización del modelo a nuevos datos.

5.2.3. Implementación de Métodos de Estimación de Incertidumbre

Para cada uno de los métodos de UE, se siguieron las siguientes estrategias:

1. **Monte Carlo Dropout (MCD):** Se implementa MCD manteniendo las capas dropout activas durante la inferencia. Para cada muestra de prueba, se realiza N inferencias y se obtienen N predicciones diferentes, lo que permite estimar la incertidumbre a partir de la variabilidad en las predicciones. Se probaron tasas de dropout (d) entre 0.1 y 0.5, y tamaños de ensemble entre 10 y 100.

2. **Deep Ensembles (DEB):** Se entrenan y combinan N modelos de BETO con diferentes inicializaciones aleatorias. Las predicciones se promedian, y la incertidumbre se estima a partir de la varianza en las predicciones de los diferentes modelos. Se evaluaron tamaños de ensemble entre 4 y 10.

3. **Estimación de Densidades de Características (FDE):** Se implementa el método FDE utilizando dos técnicas de estimación de densidad:

   - **Histogramas:** Se construyen histogramas para todas las observaciones por cada una de las dimensiones del espacio latente.

   - **Kernel Density Estimation (KDE):** Se utiliza Kernel Density Estimation (KDE) para estimar la densidad para todas las observaciones por cada una de las dimensiones del espacio latente.

Deep Ensemble basado en UE

Tamaño del Ensemble N	4	5	6	7	8	9	10

MCD basado en UE

Drop. rate d	**Tamaño del Ensemble** N									
0.1	10	20	30	40	50	60	70	80	90	100
0.2	10	20	30	40	50	60	70	80	90	100
0.3	10	20	30	40	50	60	70	80	90	100
0.4	10	20	30	40	50	60	70	80	90	100
0.5	10	20	30	40	50	60	70	80	90	100

FDE basado en UE

FDE usando Histogramas	20 bins predeterminado
FDE usando KDE	Función Gaussiana

Cuadro 5.1: Parámetros evaluados para los métodos MCD, DEB y el método propuesto FDE

5.2.4. Evaluación de la Fiabilidad de la Estimación de Incertidumbre

La fiabilidad de cada método de UE se evalúa en función de la JSD [141, 142] entre las distribuciones de incertidumbre de las clasificaciones correctas e incorrectas. Los pasos para esta evaluación son:

1. **Cálculo de Incertidumbre:** Para cada instancia en el conjunto de prueba, se calcula la incertidumbre utilizando los métodos MCD, DEB y FDE.

2. **Construcción de Distribuciones de Incertidumbre:** Se construyen dos distribuciones de incertidumbre: una para las instancias clasificadas correctamente y otra para las clasificadas incorrectamente.

3. **Cálculo de la Distancia Jensen-Shannon (JSD):** Se calcula la JSD entre estas dos distribuciones, midiendo la discrepancia entre ellas. Una mayor JSD indica una mejor diferenciación entre la incertidumbre en las clasificaciones correctas e incorrectas, lo que sugiere una mayor fiabilidad del método de UE.

5.3. Resultados y Análisis Estadístico

En esta sección se presentan los resultados obtenidos para cada método de UE y se realiza un análisis estadístico utilizando la prueba de Wilcoxon para comparar su desempeño.

5.3.1. Resultados de Monte Carlo Dropout

Se analizaron diferentes configuraciones para MCD variando la tasa de dropout entre 0.1 y 0.5 y el tamaño del ensemble (N) entre 10 y 100. Los resultados promedian la JSD obtenida para cada configuración en 10 iteraciones con diferentes particiones de datos.

- **Tasa de Dropout Óptima:** Las tasas de dropout entre 0.2 y 0.3 produjeron los mejores resultados en términos de JSD.

- **Tamaño del Ensemble Óptimo:** Se observó una tendencia en la que aumentar el tamaño del ensemble mejora la fiabilidad. El mejor resultado se obtuvo con un tamaño de ensemble de $N = 100$.

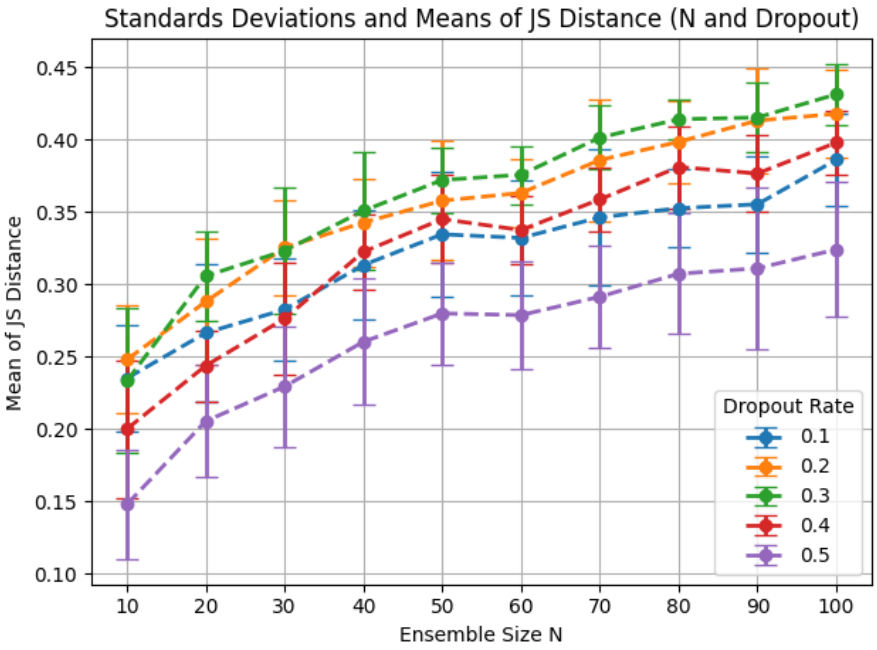

Figura 5.1: Análisis del tamaño del conjunto del método MCD comparado con el promedio de la JSD entre las UE de observaciones clasificadas correctamente e incorrectamente. Para cada punto de datos, se representó además la desviación estándar representada con las barras de error.

- **Resultado Destacado:** La configuración con una tasa de dropout $d = 0{,}3$ y un tamaño de ensemble $N = 100$ obtuvo una JSD de aproximadamente 0.432, lo que indica una alta fiabilidad del método en la tarea de clasificación.

La Figura 5.1 ilustra estos resultados, mostrando la JSD media y la desviación estándar para cada configuración.

5.3.2. Resultados de Deep Ensembles

Se evaluó la implementación del método DEB con diferentes tamaños de ensemble, variando N entre 4 y 10. Cada modelo en el ensemble se entrenó con una inicialización aleatoria distinta, y se analizaron las predicciones combinadas en términos de JSD.

- **Tamaño del Ensemble Óptimo:** Se observó una correlación positiva entre el tamaño del ensemble y la fiabilidad medida por la JSD. El mejor desempeño se

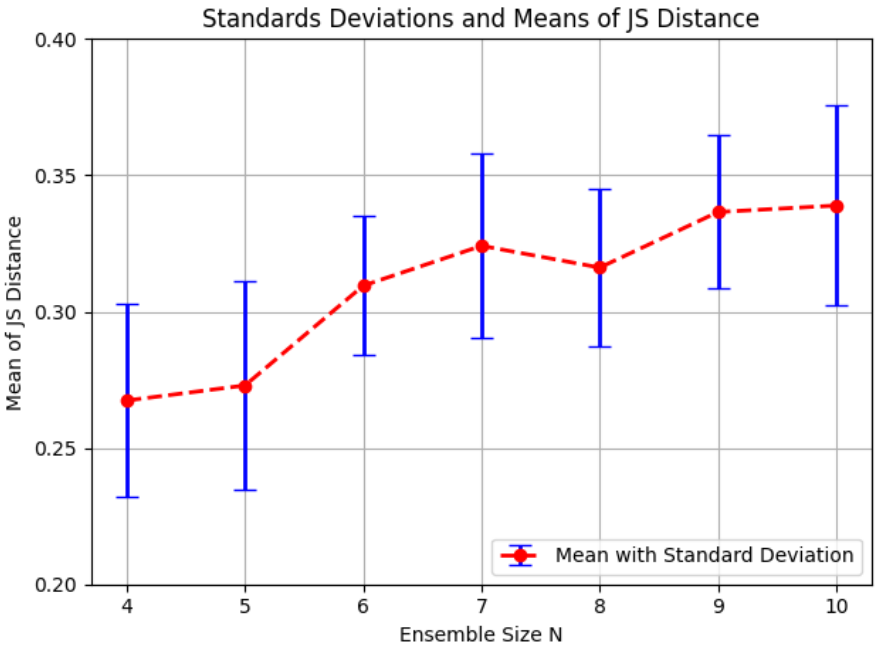

Figura 5.2: Análisis del tamaño del ensemble del método DEB y su relación con la media de la JSD en las UE de las observaciones clasificadas correctamente e incorrectamente. Para cada punto de datos, los *whiskers* reflejan la desviación estándar asociada, proporcionando una visión más detallada de la variabilidad del método DEB en diferentes condiciones de prueba.

obtuvo con un ensemble de $N = 10$, alcanzando una JSD de aproximadamente 0.339.

- **Variabilidad de Resultados:** La desviación estándar asociada con cada tamaño de ensemble se mantuvo consistente, lo que sugiere estabilidad en las estimaciones de incertidumbre independientemente del número de modelos en el ensemble.

La Figura 5.2 muestra la JSD media y la desviación estándar para cada tamaño de ensemble evaluado. Se evidencia que los ensembles más grandes ofrecen mejores resultados en términos de fiabilidad.

5.3.3. Resultados de Estimación de Densidades de Características

Para el método FDE, se realizaron dos experimentos utilizando dos técnicas de estimación de densidad: Histogramas y KDE. Se estimaron las densidades de las

representaciones latentes producidas por BETO en el conjunto de entrenamiento y se usaron estas densidades para medir la incertidumbre en el conjunto de prueba.

FDE utilizando Histograma

En este experimento, se construyeron histogramas unidimensionales para cada dimensión de las representaciones latentes, utilizando 20 bins. En este caso se utilizó solamente la cantidad de 20 bins ya que, si los bins son demasiados, podría sobre-representar pequeñas variaciones en los datos, que pueden ser ruido en lugar de información significativa, esto podría afectar los resultados del método FDE. Los resultados de la JSD obtenidos en las 10 particiones son los siguientes:

Partición	JSD
1	0.2769
2	0.5544
3	0.3853
4	0.4688
5	0.5353
6	0.3638
7	0.3244
8	0.3814
9	0.5294
10	0.3937

Cuadro 5.2: Resultados de JSD para el método FDE utilizando histogramas en las 10 particiones.

El promedio de la JSD fue de 0,421 con una desviación estándar de 0,090. Aunque algunos valores de JSD son comparables o superiores a los obtenidos con MCD y DEB, el promedio es ligeramente inferior al mejor resultado de MCD.

FDE utilizando Kernel Density Estimation (KDE)

Se decidió emplear KDE para mejorar la estimación de densidad. Los resultados de la JSD en las 10 particiones son:

Partición	JSD
1	0.2768
2	0.4733
3	0.3460
4	0.4872
5	0.5465
6	0.3045
7	0.2821
8	0.4243
9	0.5091
10	0.3490

Cuadro 5.3: Resultados de JSD para el método FDE utilizando KDE en las 10 particiones.

El promedio de la JSD fue de 0,400 con una desviación estándar de 0,095. Los resultados no mostraron una mejora significativa respecto al uso de histogramas, y el promedio sigue siendo ligeramente inferior al mejor resultado obtenido con MCD.

5.3.4. Análisis Estadístico de los Resultados

Para realizar este análisis estadístico se decidió seleccionar, de las técnicas aplicadas, sus mejores configuraciones, aquellas donde la JSD promedio de todas las particiones resultó de mayor valor. Se exceptúa FDE por ser el método propuesto, en este caso estarán sus dos configuraciones dentro de este análisis. Los datos a tomar en cuenta para este análisis se presentan en el cuadro 5.4. Con los valores individuales de la JSD por partición para cada método de estimación de incertidumbre, es posible realizar la **prueba de rangos con signo de Wilcoxon** para muestras pareadas. Esta prueba nos permitirá determinar si existen diferencias estadísticamente significativas entre los métodos comparados.

Descripción de la Prueba de Wilcoxon

La prueba de rangos con signo de Wilcoxon es una prueba no paramétrica que se utiliza para comparar dos muestras relacionadas y evaluar si sus medianas difieren significativamente. Es adecuada para muestras pequeñas y no requiere que los datos sigan una distribución normal.

Las hipótesis son:

- Hipótesis nula (H_0): No hay diferencia en las medianas de las distribuciones de JSD entre los dos métodos comparados.
- Hipótesis alternativa (H_a): Hay una diferencia significativa en las medianas de las distribuciones de JSD entre los dos métodos.

Datos Utilizados

Los valores de JSD por partición para cada método son:

Partición	MCD	DEB	FDE Histograma	FDE KDE
1	0.4210	0.3630	0.2770	0.2768
2	0.4521	0.3597	0.5544	0.4733
3	0.4118	0.3159	0.3853	0.3460
4	0.4044	0.3057	0.4688	0.4872
5	0.4636	0.3400	0.5353	0.5465
6	0.4594	0.3515	0.3638	0.3045
7	0.4455	0.2505	0.3244	0.2821
8	0.4136	0.3804	0.3814	0.4243
9	0.4098	0.3654	0.5294	0.5091
10	0.4341	0.3571	0.3937	0.3490

Cuadro 5.4: Valores de JSD por partición para cada método.

Se realizarán las siguientes comparaciones:

1. MCD vs. DEB
2. MCD vs. FDE Histograma

3. MCD vs. FDE KDE
4. DEB vs. FDE Histograma
5. DEB vs. FDE KDE
6. FDE Histograma vs. FDE KDE

Procedimiento de la Prueba de Wilcoxon

Para cada comparación, se seguirán los siguientes pasos:

1. Calcular las diferencias (D_i) entre los valores de JSD para cada partición.
2. Ordenar las diferencias por su valor absoluto y asignar rangos, teniendo en cuenta el signo de la diferencia.
3. Calcular el estadístico W, que es la suma de los rangos positivos o negativos, dependiendo de cuál sea menor.
4. Determinar el valor crítico T asociado al estadístico W en el cuadro 5.5. Como es una prueba de dos colas *(permite determinar si existen diferencias significativas en términos generales)*, y el valor de significancia es de $p = 0{,}05$, la columna que se utiliza en el cuadro 5.5 es la tercera, donde se representan los valores críticos de T para esta configuración.
5. Comparar el valor crítico T con el valor del estadístico W obtenido en las diferentes comparaciones.
6. Si W es menor que el valor crítico correspondiente del cuadro 5.5 entonces se rechaza H_0 y se acepta la hipótesis alternativa (H_a), dando como resultado la existencia de diferencia significativa entre las distribuciones de la JSD de ambos métodos.

Comparación 1: MCD vs. DEB

Cálculo de las Diferencias y Rangos

Calculamos las diferencias, se pueden ver en el cuadro 5.6:

Valores críticos de T para la prueba de rangos con signos de Wilcoxon				
			p	
n	**0.005 (una cola)** **0.01 (dos colas)**	**0.01 (una cola)** **0.02 (dos colas)**	**0.025 (una cola)** **0.05 (dos colas)**	**0.05 (una cola)** **0.10 (dos colas)**
5	-	-	-	1
6	-	-	1	2
7	-	0	2	4
8	0	2	4	6
9	2	3	6	8
10	3	5	8	11

Cuadro 5.5: Representación de los valores críticos de T para la prueba estadística de Wilcoxon. Por la naturaleza de los datos de JSD y ya que es deseable conocer si existen diferencias significativas, de forma general, entre cada par de evaluaciones realizadas, se utilizará solamente los valores presentes en la columna número 3 para un $n = 10$

.

$$D_i = \text{JSD}_{\text{MCD},i} - \text{JSD}_{\text{DEB},i} \tag{5.1}$$

Partición	JSD_{MCD}	JSD_{DEB}	D_i	**Rango**
1	0.4210	0.3630	0.0580	3
2	0.4521	0.3597	0.0924	5
3	0.4118	0.3159	0.0959	6
4	0.4044	0.3057	0.0987	7
5	0.4636	0.3400	0.1236	9
6	0.4594	0.3515	0.1079	8
7	0.4455	0.2505	0.1950	10
8	0.4136	0.3804	0.0332	1
9	0.4098	0.3654	0.0444	2
10	0.4341	0.3571	0.0770	4

Cuadro 5.6: Diferencias y rangos para MCD vs. DEB. Todas las diferencias son positivas.

- Cálculo del Estadístico W

Todas las diferencias son positivas, por lo que:

- Suma de rangos positivos (W^+): $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55$
- Suma de rangos negativos (W^-): 0

El estadístico de prueba es el menor de W^+ y W^-, por lo que $W = 0$.

- Determinación del Valor crítico T

Tal y como se detalla en el procedimiento para realizar la prueba estadística de Wilcoxon en la subsección 5.3.4, el valor crítico es $T = 8$

- Conclusión

Como $W < T$, se **rechaza la hipótesis nula** H_0. Hay evidencia suficiente para afirmar que existe una diferencia significativa entre MCD y DEB.

Comparación 2: MCD vs. FDE Histograma

Cálculo de las Diferencias y Rangos

Calculamos $D_i = \text{JSD}_{\text{MCD},i} - \text{JSD}_{\text{FDE Hist},i}$:

Partición	JSD_{MCD}	$\text{JSD}_{\text{FDE Hist}}$	D_i	Rango
1	0.4210	0.2770	0.1440	10
2	0.4521	0.5544	-0.1023	-7
3	0.4118	0.3853	0.0265	1
4	0.4044	0.4688	-0.0644	-4
5	0.4636	0.5353	-0.0717	-5
6	0.4594	0.3638	0.0956	6
7	0.4455	0.3244	0.1211	9
8	0.4136	0.3814	0.0322	2
9	0.4098	0.5294	-0.1196	-8
10	0.4341	0.3937	0.0404	3

Cuadro 5.7: Diferencias y rangos para MCD vs. FDE Histograma.

- Cálculo del Estadístico W

- Suma de rangos positivos (W^+): $1 + 2 + 3 + 6 + 9 + 10 = 31$
- Suma de rangos negativos (W^-): $4 + 5 + 7 + 8 = 24$

El estadístico de prueba es el menor de W^+ y W^-, por lo que $W = 24$.

- Determinación del Valor crítico T

Tal y como se detalla en el procedimiento para realizar la prueba estadística de Wilcoxon en la subsección 5.3.4, el valor crítico es $T = 8$

- Conclusión

Como $W > T$, **no se rechaza la hipótesis nula**. No hay evidencia suficiente para afirmar que existe una diferencia significativa entre MCD y FDE Histograma.

Comparación 3: MCD vs. FDE KDE

Cálculo de las Diferencias y Rangos

Calculamos $D_i = \text{JSD}_{\text{MCD},i} - \text{JSD}_{\text{FDE KDE},i}$:

Partición	JSD_{MCD}	$\text{JSD}_{\text{FDE KDE}}$	D_i	**Rango**
1	0.4210	0.2768	0.1442	8
2	0.4521	0.4733	-0.0212	-2
3	0.4118	0.3460	0.0658	3
4	0.4044	0.4872	-0.0828	-4
5	0.4636	0.5465	-0.0829	-5
6	0.4594	0.3045	0.1549	9
7	0.4455	0.2821	0.1634	10
8	0.4136	0.4243	-0.0107	-1
9	0.4098	0.5091	-0.0993	-7
10	0.4341	0.3490	0.0851	6

Cuadro 5.8: Diferencias y rangos para MCD vs. FDE KDE.

- Cálculo del Estadístico W

Suma de rangos positivos (W^+): $3 + 6 + 8 + 9 + 10 = 36$

Suma de rangos negativos (W^-): $1 + 2 + 4 + 5 + 7 = 19$

Estadístico de prueba $W = 19$

- Determinación del Valor crítico T

Tal y como se detalla en el procedimiento para realizar la prueba estadística de Wilcoxon en la subsección 5.3.4, el valor crítico es $T = 8$

- Conclusión

Como $W > T$, **no se rechaza la hipótesis nula**. No hay evidencia suficiente para afirmar que existe una diferencia significativa entre MCD y FDE KDE.

Comparación 4: DEB vs. FDE Histograma

Cálculo de las Diferencias y Rangos

Calculamos $D_i = \text{JSD}_{\text{DEB},i} - \text{JSD}_{\text{FDE Hist},i}$:

Partición	JSD_{DEB}	$\text{JSD}_{\text{FDE Hist}}$	D_i	**Rango**
1	0.3630	0.2770	0.0860	6
2	0.3597	0.5544	-0.1947	-9
3	0.3159	0.3853	-0.0694	-4
4	0.3057	0.4688	-0.1631	-7
5	0.3400	0.5353	-0.1953	-10
6	0.3515	0.3638	-0.0123	-2
7	0.2505	0.3244	-0.0739	-5
8	0.3804	0.3814	-0.0010	-1
9	0.3654	0.5294	-0.1640	-8
10	0.3571	0.3937	-0.0366	-3

Cuadro 5.9: Diferencias y rangos para DEB vs. FDE Histograma.

- Cálculo del Estadístico W

Suma de rangos positivos (W^+): 6

Suma de rangos negativos (W^-): $1 + 2 + 3 + 4 + 5 + 7 + 8 + 9 + 10 = 49$

Estadístico de prueba $W = 6$

- Determinación del Valor crítico T

Tal y como se detalla en el procedimiento para realizar la prueba estadística de Wilcoxon en la subsección 5.3.4, el valor crítico es $T = 8$

- Conclusión

Como $W < T$, se **rechaza la hipótesis nula**. Hay evidencia suficiente para afirmar que existe una diferencia significativa entre DEB y FDE Histograma.

Comparación 5: DEB vs. FDE KDE

Cálculo de las Diferencias y Rangos

Calculamos $D_i = \text{JSD}_{\text{DEB},i} - \text{JSD}_{\text{FDE KDE},i}$:

Partición	JSD_{DEB}	$\text{JSD}_{\text{FDE KDE}}$	D_i	**Rango**
1	0.3630	0.2768	0.0862	6
2	0.3597	0.4733	-0.1136	-7
3	0.3159	0.3460	-0.0301	-2
4	0.3057	0.4872	-0.1815	-9
5	0.3400	0.5465	-0.2065	-10
6	0.3515	0.3045	0.0470	5
7	0.2505	0.2821	-0.0316	-3
8	0.3804	0.4243	-0.0439	-4
9	0.3654	0.5091	-0.1437	-8
10	0.3571	0.3490	0.0081	1

Cuadro 5.10: Diferencias y rangos para DEB vs. FDE KDE.

- Cálculo del Estadístico W

Suma de rangos positivos (W^+): $1 + 5 + 6 = 12$

Suma de rangos negativos (W^-): $2 + 3 + 4 + 7 + 8 + 9 + 10 = 43$

Estadístico de prueba $W = 12$

- Determinación del Valor crítico T

Tal y como se detalla en el procedimiento para realizar la prueba estadística de Wilcoxon en la subsección 5.3.4, el valor crítico es $T = 8$

- Conclusión

Como $W > T$, **no se rechaza la hipótesis nula**. No hay evidencia suficiente para afirmar que existe una diferencia significativa entre DEB y FDE KDE.

Comparación 6: FDE Histograma vs. FDE KDE

Cálculo de las Diferencias y Rangos

Calculamos $D_i = \text{JSD}_{\text{FDE Hist},i} - \text{JSD}_{\text{FDE KDE},i}$:

Partición	$\text{JSD}_{\text{FDE Hist}}$	$\text{JSD}_{\text{FDE KDE}}$	D_i	**Rango**
1	0.2770	0.2768	0.0002	1
2	0.5544	0.4733	0.0811	10
3	0.3853	0.3460	0.0393	5
4	0.4688	0.4872	-0.0184	-3
5	0.5353	0.5465	-0.0112	-2
6	0.3638	0.3045	0.0593	9
7	0.3244	0.2821	0.0423	6
8	0.3814	0.4243	-0.0429	-7
9	0.5294	0.5091	0.0203	4
10	0.3937	0.3490	0.0447	8

Cuadro 5.11: Diferencias y rangos para FDE Histograma vs. FDE KDE.

Cálculo del Estadístico W

Suma de rangos positivos (W^+): $1 + 4 + 5 + 6 + 8 + 9 + 10 = 43$

Suma de rangos negativos (W^-): $2 + 3 + 7 = 12$

Estadístico de prueba $W = 12$

- Determinación del Valor crítico T

Tal y como se detalla en el procedimiento para realizar la prueba estadística de Wilcoxon en la subsección 5.3.4, el valor crítico es $T = 8$

- Conclusión

Como $W > T$, **no se rechaza la hipótesis nula**. No hay evidencia suficiente para afirmar que existe una diferencia significativa entre FDE Histograma y FDE KDE.

Media y desviación estándar de JSD por Métodos		**Diferencias estadísticas significativas**
MCD [0.4315 (±0,0223)]	DEB [0.3389 (±0,0386)]	Si
	FDE (Histogramas) [0.4214 (±0,0953)]	No
	FDE (KDE) [0.4000 (±0,1004)]	No
DEB [0.3389 (±0,0386)]	FDE (Histogramas) [0.4214 (±0,0953)]	Si
	FDE (KDE) [0.4000 (±0,1004)]	No
FDE (Histogramas) [0.4214 (±0,0953)]	FDE (KDE) [0.4000 (±0,1004)]	No

Cuadro 5.12: Resumen de la prueba estadística Wilcoxon realizada a los diferentes métodos de UE implementados

5.4. Comparación y Discusión de Resultados

En esta sección se compara el desempeño de los métodos evaluados y se discuten los hallazgos, incluyendo un análisis del costo computacional asociado a cada método.

5.4.1. Comparación del Desempeño de los Métodos

Los resultados indican que el método MCD con una tasa de dropout de 0.3 y un ensemble de 100 muestras obtuvo el mejor promedio de JSD (0.432), indicando una mayor capacidad para diferenciar entre incertidumbre en predicciones correctas e incorrectas. Aunque FDE no superó a MCD en términos de JSD, los resultados del análisis estadístico muestran que no hay diferencias significativas entre ellos, lo que sugiere que FDE es una alternativa viable con un desempeño comparable.

Por otro lado, DEB obtuvo un promedio de JSD inferior (0.339), siendo significativamente superado por MCD y FDE.

5.4.2. Análisis del Costo Computacional

El costo computacional es un factor crítico en la aplicación práctica de los métodos de UE. A continuación, se analiza la eficiencia computacional de cada método durante la fase de evaluación.

Costo Computacional del Método FDE con Histogramas

En el caso del método FDE utilizando histogramas, después de construir los histogramas para cada dimensión, el costo computacional por observación durante la evaluación es:

$$\mathcal{O}(r + d)$$

donde:

- r es el costo computacional de evaluar la red neuronal para obtener la representación latente de la nueva observación.
- d es el costo de evaluar las densidades en cada una de las d dimensiones (es decir, determinar en qué bin cae cada componente de la representación latente).

La búsqueda del bin correspondiente en un histograma es generalmente muy eficiente, ya que implica operaciones simples y el número de bins es fijo. Por lo tanto, el costo computacional asociado a d es menor en comparación con el uso de KDE.

Costo Computacional del Método FDE con KDE

En el caso del método de Estimación de Densidades de Características (FDE) utilizando KDE, después de estimar las densidades $\widetilde{p}_d$ para cada dimensión utilizando KDE, el costo computacional por observación durante la evaluación es:

$$\mathcal{O}(r + d \times k)$$

donde:

- r es el costo computacional de evaluar la red neuronal para obtener la representación latente de la nueva observación.
- d es el número de dimensiones del espacio latente (por ejemplo, 768 dimensiones en BETO).
- k es el costo computacional de evaluar la densidad en cada dimensión utilizando KDE.

La evaluación de las densidades con KDE es más costosa que con histogramas, ya que implica operaciones sobre todos los puntos de datos utilizados para estimar la densidad en cada dimensión. Sin embargo, este costo es lineal respecto al número de puntos puede ser manejable en la práctica.

Comparación con FDE con Histogramas

En comparación con FDE utilizando histogramas, donde el costo asociado a d es menor debido a la simplicidad de buscar en qué bin cae cada valor, FDE con KDE tiene un costo computacional ligeramente mayor en la fase de evaluación debido al cálculo de las densidades utilizando funciones kernel. No obstante, este costo sigue siendo significativamente menor que el de métodos que requieren múltiples evaluaciones de la red neuronal completa.

Costo Computacional del Método Monte Carlo Dropout (MCD)

Para el método **Monte Carlo Dropout (MCD)**, el costo computacional por observación durante la evaluación es:

$$\mathcal{O}\left(n \times r\right)$$

donde:

- n es el tamaño del ensemble o el número de evaluaciones estocásticas realizadas (número de muestras de dropout).
- r es el costo computacional de evaluar la red neuronal una vez.

En MCD, para estimar la incertidumbre, es necesario realizar múltiples **pases hacia adelante** (forward passes) a través de la red neuronal, cada uno con una máscara de dropout diferente. Esto implica que la red neuronal se evalúa n veces para cada nueva observación, lo que resulta en un costo computacional significativamente mayor en comparación con los métodos FDE.

Costo Computacional del Método Deep Ensembles (DEB)

Para el método DEB, el costo computacional por observación durante la evaluación es:

$$\mathcal{O}(n \times r)$$

donde:

- n es el tamaño del ensemble, es decir, el número de modelos independientes entrenados.
- r es el costo computacional de evaluar un modelo de la red neuronal.

En DEB, cada modelo del ensemble debe evaluar la nueva observación, lo que implica ejecutar n veces la red neuronal completa. Esto resulta en un costo computacional significativamente mayor en comparación con los métodos de FDE.

Conclusión sobre el Costo Computacional

Los métodos basados en FDE ofrecen una ventaja significativa en términos de eficiencia computacional durante la fase de evaluación. Dado que proporcionan resultados comparables a MCD sin diferencias significativas en términos de fiabilidad de la incertidumbre, y con un costo computacional mucho menor, son una alternativa atractiva para aplicaciones donde el tiempo de inferencia y los recursos computacionales son limitados.

Método	**Costo Computacional**
FDE (Histogramas)	$\mathcal{O}(r + d)$
FDE (KDE)	$\mathcal{O}(r + d \times k)$
MCD	$\mathcal{O}(n \times r)$
DEB	$\mathcal{O}(n \times r)$

Cuadro 5.13: Resumen del costo computacional de los diferentes métodos de UE. Donde: r es el costo de evaluar la red neuronal (una vez), d es el número de dimensiones del espacio latente, k es el costo computacional de evaluar la densidad con KDE por dimensión, y por último n es el número de modelos en el ensamble o número de evaluaciones para MCD.

5.5. Conclusiones

En este capítulo se han evaluado tres métodos de estimación de incertidumbre (UE) en la tarea de clasificación de complejidad textual en español, abordando los **Objetivos Específicos** 1 al 3 planteados en esta tesis. Los métodos evaluados fueron **Monte Carlo Dropout (MCD)**, **Deep Ensembles (DEB)** y el método propuesto **Feature Density Estimation (FDE)**.

- **Implementación de BETO para la Clasificación de Textos:** El modelo BETO fue ajustado exitosamente para clasificar textos como simples o complejos, alcanzando precisiones entre 81 % y 83 % en las pruebas realizadas. Este resultado demuestra la efectividad del modelo en tareas de clasificación de textos en español y cumple con el **Objetivo Específico 1**.

- **Desarrollo e Implementación del Método FDE:** Se implementó el método FDE utilizando tanto histogramas como KDE para la estimación de densidades en el espacio latente generado por BETO. Este enfoque innovador permite estimar la incertidumbre aprovechando la distribución de las características latentes, cumpliendo así con el **Objetivo Específico 2**.

- **Establecimiento de una Métrica para Evaluar la Confiabilidad de las Estimaciones de Incertidumbre:** Se utilizó la JSD como métrica para cuantificar la fiabilidad de las estimaciones de incertidumbre proporcionadas por los modelos. Esta métrica mide la divergencia entre las distribuciones de incertidumbre de las clasificaciones correctas e incorrectas, cumpliendo con el **Objetivo Específico 3**.

- **Análisis Comparativo de los Métodos de UE:** Aunque el método FDE no superó a MCD en términos del valor promedio de JSD, demostró un desempeño comparable con un costo computacional significativamente menor. El análisis estadístico mediante la prueba de Wilcoxon indicó que no hay diferencias significativas entre FDE y MCD, mientras que FDE mostró una ventaja significativa sobre DEB. Estos hallazgos sugieren que FDE es una alternativa viable y eficiente para la estimación de incertidumbre en tareas de clasificación de textos.

Estos resultados aportan información valiosa sobre las ventajas y desafíos de utilizar FDE en la estimación de incertidumbre en modelos de lenguaje basados en *transformers* como BETO. La combinación de un desempeño comparable al de MCD y una mayor eficiencia computacional resalta el potencial de FDE en aplicaciones prácticas donde los recursos son limitados.

Validación de la Hipótesis

La hipótesis inicial planteaba que el método FDE mejoraría significativamente la estimación de incertidumbre en el modelo BETO en comparación con técnicas tradicionales como MCD y DEB. Si bien FDE no demostró una superioridad estadísticamente significativa sobre MCD en términos de fiabilidad de la incertidumbre, alcanzó un desempeño comparable con un costo computacional mucho menor. Esta eficiencia computacional y la viabilidad práctica del método respaldan parcialmente la hipótesis propuesta. Por lo tanto, se considera que la hipótesis es **parcialmente validada**.

6. Conclusiones y Trabajo Futuro

6.1. Conclusiones Generales

La presente tesis ha abordado el desafío de estimar la incertidumbre en la clasificación de la complejidad textual en español, una tarea fundamental para mejorar la accesibilidad y personalización de contenidos en ámbitos educativos y de comunicación. A través del desarrollo e implementación de métodos avanzados de estimación de incertidumbre, se ha buscado mejorar la confiabilidad y eficacia de los modelos de clasificación basados en aprendizaje profundo.

6.1.1. Cumplimiento de los Objetivos Específicos

A lo largo de este trabajo, se han alcanzado los objetivos específicos planteados:

- **Objetivo Específico 1: Implementar un modelo de aprendizaje profundo para la detección de textos complejos en español.** Se implementó y ajustó exitosamente un modelo de aprendizaje profundo basado en BETO, logrando precisiones entre 81 % y 83 % en la clasificación de textos como simples o complejos. Esto demuestra la efectividad del modelo preentrenado en español para capturar las particularidades lingüísticas del idioma y su capacidad para abordar la tarea propuesta.

- **Objetivo Específico 2: Proponer al menos un método novedoso de estimación de incertidumbre en la detección de texto complejo en español basado en FDE.** Se propuso e implementó el método FDE, utilizando técnicas de estimación de densidades en las representaciones latentes generadas por el modelo de aprendizaje profundo. Este enfoque innovador permite estimar la incertidumbre de manera eficiente, sin incurrir en altos costos computacionales asociados con métodos tradicionales.

- **Objetivo Específico 3: Proponer una métrica para cuantificar la confiabilidad**

de las estimaciones de incertidumbre en la clasificación de texto complejo en español. Se utilizó la JSD como métrica para cuantificar la confiabilidad de las estimaciones de incertidumbre. Esta métrica permitió comparar las distribuciones de incertidumbre entre predicciones correctas e incorrectas, proporcionando una medida robusta de la capacidad del modelo para distinguir entre casos seguros e inseguros.

6.1.2. Validación de la Hipótesis

La hipótesis planteada en esta tesis sugería que el método FDE mejoraría significativamente la estimación de incertidumbre en comparación con métodos tradicionales como MCD y DEB, al tiempo que reduciría el costo computacional asociado. Los resultados obtenidos permiten afirmar que:

- **Desempeño Comparable a MCD:** Aunque FDE no mostró una superioridad estadísticamente significativa sobre MCD en términos de la métrica JSD, alcanzó un desempeño similar, lo que indica que es capaz de estimar la incertidumbre de manera efectiva.
- **Eficiencia Computacional Superior:** FDE demostró una reducción significativa en el costo computacional durante la fase de inferencia, requiriendo solo una evaluación del modelo principal y cálculos adicionales lineales en la dimensionalidad del espacio latente.
- **Superación de DEB:** El método FDE superó estadísticamente a DEB en términos de fiabilidad de las estimaciones de incertidumbre, evidenciando su eficacia y ventajas sobre otros métodos tradicionales.

Por lo tanto, la hipótesis es **parcialmente validada**, ya que si bien FDE no superó a MCD en desempeño, sí ofreció beneficios significativos en eficiencia computacional sin comprometer la calidad de las estimaciones de incertidumbre.

6.1.3. Contribuciones Significativas

Este trabajo aporta varias contribuciones al campo del procesamiento del lenguaje natural y la estimación de incertidumbre:

- **Adaptación de BETO para la Clasificación de Complejidad Textual:** Se demostró la efectividad de BETO en la tarea de clasificación de textos simples y complejos en español, sentando las bases para futuras investigaciones en este ámbito.

- **Propuesta y Validación del Método FDE:** Se introdujo un método novedoso y eficiente para la estimación de incertidumbre, adaptado a modelos de lenguaje en español, contribuyendo a la literatura existente y abriendo nuevas posibilidades en aplicaciones prácticas.

- **Evaluación Comparativa de Métodos de UE:** Se proporcionó una comparación detallada entre métodos tradicionales y el propuesto FDE, ofreciendo insights valiosos sobre sus ventajas y limitaciones, lo cual es útil para la comunidad científica y para desarrolladores de sistemas de NLP.

6.2. Limitaciones del Estudio

A pesar de los resultados positivos, esta investigación presenta algunas limitaciones:

- **Representaciones Latentes Limitadas:** Las representaciones latentes, también conocidas como *embeddings* o codificaciones internas, son las características de alto nivel que el modelo extrae de los datos de entrada para realizar predicciones. En el caso de modelos de lenguaje como BETO, estas representaciones capturan información semántica y sintáctica del texto. Una limitación identificada es que estas representaciones latentes podrían no ser lo suficientemente discriminativas para distinguir completamente entre las clases de textos simples y complejos. Si las representaciones de ambas clases son muy similares o se solapan en el espacio latente, el método de Estimación de Densidades de Características (FDE) puede tener dificultades para estimar con precisión la incertidumbre asociada a cada predicción.

- **Estimación de Densidad Unidimensional:** La simplificación al asumir independencia entre las dimensiones latentes puede conducir a la pérdida de informa-

ción sobre las correlaciones entre ellas, potencialmente limitando la precisión de las estimaciones de incertidumbre.

- **Dominio Específico del Conjunto de Datos:** El conjunto de datos utilizado se centra en textos de educación financiera, lo que puede limitar la generalización de los resultados a otros dominios o tipos de texto en español.

6.3. Trabajo Futuro

Con base en las conclusiones y las limitaciones identificadas, se proponen las siguientes líneas de investigación futura:

- **Mejora de las Representaciones Latentes:** Investigar técnicas para obtener representaciones latentes más discriminativas, como el uso de modelos preentrenados más grandes o técnicas de aprendizaje contrastivo, podría mejorar la eficacia de FDE.

- **Estimación de Densidad Multidimensional:** Implementar métodos de estimación de densidad que consideren las correlaciones entre dimensiones, como modelos de mezcla gaussianos o técnicas de aprendizaje de distribuciones en espacios latentes, puede aumentar la precisión de las estimaciones de incertidumbre [143, 144].

- **Compresión de Modelos para Mayor Eficiencia:** Explorar técnicas de compresión de modelos, como poda (*pruning*), cuantización y destilación de conocimiento (*knowledge distillation*), para reducir el tamaño y la complejidad computacional de los modelos utilizados. Estas técnicas pueden hacer que los modelos sean más adecuados para su implementación en dispositivos con recursos limitados, manteniendo un rendimiento competitivo [145, 146].

- **Aplicación a Otros Dominios y Tareas:** Extender el método propuesto a otros dominios temáticos y tareas de NLP en español, como análisis de sentimiento, detección de entidades o traducción automática, para evaluar su generalización y adaptabilidad.

6.4. Reflexiones Finales

La estimación de la incertidumbre es un componente esencial para aumentar la confiabilidad y robustez de los modelos de aprendizaje profundo en NLP. Este trabajo ha demostrado que es posible desarrollar métodos eficientes y efectivos para estimar la incertidumbre en la clasificación de textos en español, contribuyendo a mejorar la accesibilidad y personalización de contenidos.

El método FDE propuesto ofrece una alternativa viable a los métodos tradicionales, especialmente en contextos donde los recursos computacionales son limitados. Los hallazgos de esta tesis no solo aportan al conocimiento científico, sino que también tienen implicaciones prácticas para el desarrollo de aplicaciones más seguras y confiables en el procesamiento del lenguaje natural.

Se espera que las futuras investigaciones puedan ampliar y profundizar en las líneas aquí propuestas, avanzando hacia modelos de lenguaje más interpretables y confiables, y promoviendo una mayor inclusión y accesibilidad en la comunicación en español.

Bibliografía

[1] S. S. Al-Thanyyan and A. M. Azmi, "Automated text simplification: a survey," ACM Computing Surveys (CSUR), vol. 54, no. 2, pp. 1–36, 2021.

[2] Y. Lao León, A. Rivas Méndez, M. Pérez Pravia, and F. Marrero Delgado, "Procedimiento para el pronóstico de la demanda mediante redes neuronales artificiales / procedure for forecasting demand by using artificial neural networks," Ciencias Holguin, vol. 23, pp. 43–59, 01 2017.

[3] S. C. Ramirez, Improving semi-supervised deep learning under distribution mismatch for medical image analysis applications. PhD thesis, 2021.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[5] J. Mena, O. Pujol, and J. Vitria, "A survey on uncertainty estimation in deep learning classification systems from a bayesian perspective," ACM Computing Surveys (CSUR), vol. 54, no. 9, pp. 1–35, 2021.

[6] F. Fakour, A. Mosleh, and R. Ramezani, "A structured review of literature on uncertainty in machine learning & deep learning," arXiv preprint arXiv:2406.00332, 2024.

[7] R. Lehe, Uncertainty quantification in Machine learning. Nov 2022.

[8] S. Yang and K. Yee, "Towards reliable uncertainty quantification via deep ensemble in multi-output regression task," Engineering Applications of Artificial Intelligence, vol. 132, p. 107871, 2024.

[9] N. Perez-Rojas, S. Calderon-Ramirez, M. Solis-Salazar, M. Romero-Sandoval, M. Arias-Monge, and H. Saggion, "A novel dataset for financial education text simplification in spanish," arXiv preprint arXiv:2312.09897, 2023.

[10] M. Shardlow, "A survey of automated text simplification," International Journal of Advanced Computer Science and Applications, vol. 4, no. 1, pp. 58–70, 2014.

[11] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2019.

[12] M. Abreu-Cardenas, S. Calderón-Ramírez, and M. Solís, "Uncertainty estimation for complex text detection in spanish," in 2023 IEEE 5th International Conference on BioInspired Processing (BIP), pp. 1–6, 2023.

[13] M. Huang, X. Zhu, and J. Gao, "Challenges in building intelligent open-domain dialog systems," ACM Transactions on Information Systems (TOIS), vol. 38, no. 3, pp. 1–32, 2020.

[14] L. Rello and R. Baeza-Yates, "Lexical quality as a proxy for web text understandability," in Proceedings of the 21st International Conference on World Wide Web, pp. 591–592, 2012.

[15] S. S. Patil, A. Rodrigues, R. Telangi, and V. Chavan, "A review on text classification based on cnn," International Journal of Scientific Research in Science and Technology, 2022.

[16] M. Shardlow, S. Sellar, and D. Rousell, "Collaborative augmentation and simplification of text (coast): pedagogical applications of natural language processing in digital learning environments," Learning Environments Research, vol. 25, pp. 399 – 421, 2021.

[17] H. Saggion and G. Hirst, Automatic text simplification, vol. 32. Springer, 2017.

[18] J. Qiang and X. Wu, "Unsupervised statistical text simplification," IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1802–1806, 2019.

[19] A. Baez and H. Saggion, "Lsllama: Fine-tuned llama for lexical simplification," in Proceedings of the Second Workshop on Text Simplification, Accessibility and Readability, pp. 102–108, 2023.

[20] S. Stajner, D. Ibanez, and H. Saggion, "LeSS: A computationally-light lexical simplifier for Spanish," in Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing (R. Mitkov and G. Angelova, eds.), (Varna, Bulgaria), pp. 1132–1142, INCOMA Ltd., Shoumen, Bulgaria, Sept. 2023.

[21] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., "Improving language understanding by generative pre-training," 2018.

[22] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, and J. Wang, "Release strategies and the social impacts of language models," ArXiv, vol. abs/1908.09203, 2019.

[23] L. Rello, R. Baeza-Yates, L. Dempere-Marco, and H. Saggion, "Frequent words improve readability and short words improve understandability for people with dyslexia," in Human-Computer Interaction–INTERACT 2013: 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part IV 14, pp. 203–219, Springer, 2013.

[24] J. Rüsseler, S. Probst, S. Johannes, and T. F. Münte, "Recognition memory for high-and low-frequency words in adult normal and dyslexic readers: an event-related brain potential study," Journal of clinical and experimental neuropsychology, vol. 25, no. 6, pp. 815–829, 2003.

[25] R. Evans, C. Orasan, and I. Dornescu, "An evaluation of syntactic simplification rules for people with autism," Association for Computational Linguistics, 2014.

[26] I. Espinosa-Zaragoza, J. Abreu-Salas, P. Moreda, and M. Palomar, "Automatic text simplification for people with cognitive disabilities: Resource creation within the cleartext project," TSAR 2023, p. 68, 2023.

[27] M. Romero, S. Calderón-Ramírez, M. Solís, N. Pérez-Rojas, M. Chacón-Rivas, and H. Saggion, "Towards text simplification in spanish: A brief overview of deep learning approaches for text simplification," in 2022 IEEE 4th International Conference on BioInspired Processing (BIP), pp. 1–7, 2022.

[28] W. Kintsch, Comprehension: A paradigm for cognition. Cambridge university press, 1998.

[29] S. A. Crossley and D. S. McNamara, "Understanding expert ratings of essay quality: Coh-metrix analyses of first and second language writing," International Journal of Continuing Engineering Education and Life Long Learning, vol. 21, no. 2-3, pp. 170–191, 2011.

[30] L. Feng, M. Jansche, M. Huenerfauth, and N. Elhadad, "A comparison of features for automatic readability assessment," in 23rd International Conference on Computational Linguistics (COLING 2010), Poster Volume, pp. 276–284, 2010.

[31] G. Paetzold and L. Specia, "Unsupervised lexical simplification for non-native speakers," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, 2016.

[32] S. Aluísio and C. Gasperin, "Fostering digital inclusion and accessibility: the porsimples project for simplification of portuguese texts," in Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas, pp. 46–53, 2010.

[33] J. Postels, H. Blum, C. Cadena, R. Siegwart, L. Van Gool, and F. Tombari, "Quantifying aleatoric and epistemic uncertainty using density estimation in latent space," arXiv preprint arXiv:2012.03082, vol. 1, 2020.

[34] S. Calderon-Ramirez, S. Yang, D. Elizondo, and A. Moemeni, "Dealing with distribution mismatch in semi-supervised deep learning for covid-19 detection using chest x-ray images: A novel approach using feature densities," Applied Soft Computing, vol. 123, p. 108983, 2022.

[35] R. J. Fuentes-Fino, S. Calderón-Ramírez, E. Domínguez, E. López-Rubio, M. A. Hernandez-Vasquez, and M. A. Molina-Cabello, "Feature density as an uncertainty estimator method in the binary classification mammography images task for a supervised deep learning model," in International Work-Conference on Bioinformatics and Biomedical Engineering, pp. 375–388, Springer, 2022.

[36] L. Mi, H. Wang, Y. Tian, and N. Shavit, "Training-free uncertainty estimation for neural networks," 2019.

[37] S. Calderon-Ramirez, S. Yang, A. Moemeni, S. Colreavy-Donnelly, D. A. Elizondo, L. Oala, J. Rodríguez-Capitán, M. Jiménez-Navarro, E. López-Rubio, and M. A. Molina-Cabello, "Improving uncertainty estimation with semi-supervised deep learning for covid-19 detection using chest x-ray images," Ieee Access, vol. 9, pp. 85442–85454, 2021.

[38] A. Y. Odisho, B. Park, N. Altieri, J. DeNero, M. R. Cooperberg, P. R. Carroll, and B. Yu, "Natural language processing systems for pathology parsing in limited data environments with uncertainty estimation," JAMIA open, vol. 3, no. 3, pp. 431–438, 2020.

[39] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 1321–1330, 2017.

[40] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2901–2907, 2015.

[41] G. H. Paetzold and L. Specia, "A survey on lexical simplification," Journal of Artificial Intelligence Research, vol. 60, pp. 549–593, 2017.

[42] S. A. Crossley, S. Skalicky, M. Dascalu, D. S. McNamara, and K. Kyle, "Predicting text comprehension, processing, and familiarity in adult readers: New approaches to readability formulas," Discourse Processes, vol. 54, no. 5-6, pp. 340–359, 2017.

[43] S. Vajjala and D. Meurers, "Readability assessment for text simplification: From analysing documents to identifying sentential simplifications," ITL-International Journal of Applied Linguistics, vol. 165, no. 2, pp. 194–222, 2014.

[44] B. Drndarević, S. Štajner, S. Bott, S. Bautista, and H. Saggion, "Automatic text simplification in spanish: A comparative evaluation of complementing

modules," in Computational Linguistics and Intelligent Text Processing: 14th International Conference, CICLing 2013, Samos, Greece, March 24-30, 2013, Proceedings, Part II 14, pp. 488–500, Springer, 2013.

[45] F. Liu and J. S. Lee, "Hybrid models for sentence readability assessment," in Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), pp. 448–454, 2023.

[46] J. Liu and Y. Matsumoto, "Sentence complexity estimation for chinese-speaking learners of japanese," in Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation, pp. 296–302, 2017.

[47] C. Garbacea, M. Guo, S. Carton, and Q. Mei, "Explainable prediction of text complexity: The missing preliminaries for text simplification," arXiv preprint arXiv:2007.15823, 2020.

[48] G. L. Bosco, G. Pilato, and D. Schicchi, "Deepeva: a deep neural network architecture for assessing sentence complexity in italian and english languages," Array, vol. 12, p. 100097, 2021.

[49] A. Cuzzocrea, G. L. Bosco, G. Pilato, and D. Schicchi, "Multi-class text complexity evaluation via deep neural networks," in Intelligent Data Engineering and Automated Learning–IDEAL 2019: 20th International Conference, Manchester, UK, November 14–16, 2019, Proceedings, Part II 20, pp. 313–322, Springer, 2019.

[50] G. Nigusie and T. T. Asfaw, "Lexical complexity detection and simplification in amharic text using machine learning approach," in 2022 International Conference on Information and Communication Technology for Development for Africa (ICT4DA), pp. 1–6, IEEE, 2022.

[51] V. V. Ivanov, "Sentence-level complexity in russian: An evaluation of bert and graph neural networks," Frontiers in Artificial Intelligence, vol. 5, p. 1008411, 2022.

[52] I. M. Barrio-Cantalejo, P. Simón-Lorda, M. Melguizo, I. Escalona, M. I. Marijuán, and P. Hernando, "Validación de la escala inflesz para evaluar la legibilidad de

los textos dirigidos a pacientes," in Anales del sistema sanitario de Navarra, vol. 31, pp. 135–152, SciELO Espana, 2008.

[53] L. Feng, M. Jansche, M. Huenerfauth, and N. Elhadad, "A comparison of features for automatic readability assessment," in Coling 2010: Posters, pp. 276–284, 2010.

[54] S. Vajjala, D. Meurers, A. Eitel, and K. Scheiter, "Towards grounding computational linguistic approaches to readability: Modeling reader-text interaction for easy and difficult texts," in Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC), pp. 38–48, 2016.

[55] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, et al., "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," Information fusion, vol. 76, pp. 243–297, 2021.

[56] T. Glushkova, C. Zerva, R. Rei, and A. Martins, "Uncertainty-aware machine translation evaluation," ArXiv, vol. abs/2109.06352, 2021.

[57] J. V. Landeghem, M. B. Blaschko, B. Anckaert, and M.-F. Moens, "Benchmarking scalable predictive uncertainty in text classification," IEEE Access, vol. 10, pp. 43703–43737, 2022.

[58] D. Zhang, M. Sensoy, M. Makrehchi, B. Taneva-Popova, L. Gui, and Y. He, "Uncertainty quantification for text classification," Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023.

[59] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," pp. 7047–7058, 2018.

[60] A. Malinin and M. Gales, "Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness," pp. 14520–14531, 2019.

[61] G. Gordon, "Multi-dimensional linguistic complexity," Journal of Biomolecular Structure and Dynamics, vol. 20, pp. 747 – 750, 2003.

[62] W. Zhao, T. Joshi, V. Nair, and A. Sudjianto, "Shap values for explaining cnn-based text classification models," ArXiv, vol. abs/2008.11825, 2020.

[63] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[64] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez, "Spanish pre-trained bert model and evaluation data," in PML4DC at ICLR 2020, 2020.

[65] M. Wainberg, D. Merico, A. Delong, and B. J. Frey, "Deep learning in biomedicine," Nature biotechnology, vol. 36, no. 9, pp. 829–838, 2018.

[66] H. Y. Xiong, B. Alipanahi, L. J. Lee, H. Bretschneider, D. Merico, R. K. Yuen, Y. Hua, S. Gueroussov, H. S. Najafabadi, T. R. Hughes, et al., "The human splicing code reveals new insights into the genetic determinants of disease," Science, vol. 347, no. 6218, p. 1254806, 2015.

[67] S. Zhang, H. Hu, T. Jiang, L. Zhang, and J. Zeng, "Titer: predicting translation initiation sites by deep learning," Bioinformatics, vol. 33, no. 14, pp. i234–i242, 2017.

[68] D. Parks, J. X. Prochaska, S. Dong, and Z. Cai, "Deep learning of quasar spectra to discover and characterize damped lyα systems," Monthly Notices of the Royal Astronomical Society, vol. 476, no. 1, pp. 1151–1168, 2018.

[69] A. Zappone, M. D. Renzo, M. Debbah, T. T. Lam, and X. Qian, "Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks for wireless system optimization," IEEE Vehicular Technology Magazine, vol. 14, pp. 60–69, 2018.

[70] C. Cao, F. Liu, H. Tan, D. Song, W. Shu, W. Li, Y. Zhou, X. Bo, and Z. Xie, "Deep learning and its applications in biomedicine," Genomics, Proteomics Bioinformatics, vol. 16, pp. 17 – 32, 2018.

[71] M. Rizzo, M. Marcuzzo, A. Zangari, A. Gasparetto, and A. Albarelli, "Stop overkilling simple tasks with black-box models and use transparent models instead," ArXiv, vol. abs/2302.02804, 2023.

[72] L. P. Silvestrin, M. Hoogendoorn, and G. Koole, "A comparative study of state-of-the-art machine learning algorithms for predictive maintenance," 2019 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 760–767, 2019.

[73] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015.

[74] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," ArXiv, vol. abs/1506.02142, 2015.

[75] N. Gupta et al., "Artificial neural network," Network and Complex Systems, vol. 3, no. 1, pp. 24–28, 2013.

[76] S.-C. Wang and S.-C. Wang, "Artificial neural network," Interdisciplinary computing in java programming, pp. 81–100, 2003.

[77] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, A. J. Hudspeth, S. Mack, et al., Principles of neural science, vol. 4. McGraw-hill New York, 2000.

[78] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," Psychological review, vol. 65, no. 6, p. 386, 1958.

[79] T. Nitta, "Solving the xor problem and the detection of symmetry using a single complex-valued neuron," Neural Networks, vol. 16, no. 8, pp. 1101–1105, 2003.

[80] D. J. C. C. Tello, "Apuntes de redes neuronales artificiales handouts for artificial neural networks," Universidad Autónoma de San Luis Potosí, 2017.

[81] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, 2016.

[82] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers, pp. 177–186, Springer, 2010.

[83] L. Shen, C. Chen, F. Zou, Z. Jie, J. Sun, and W. Liu, "A unified analysis of adagrad with weighted aggregation and momentum acceleration.," IEEE transactions on neural networks and learning systems, vol. PP, 2018.

[84] F. Zou, L. Shen, Z. Jie, J. Sun, and W. Liu, "Weighted adagrad with unified momentum," arXiv: Learning, 2018.

[85] D. P. Kingma, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[86] T. Tieleman, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural networks for machine learning, vol. 4, no. 2, p. 26, 2012.

[87] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization.," Journal of machine learning research, vol. 12, no. 7, 2011.

[88] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[89] K. Cho, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[90] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks, vol. 5, no. 2, pp. 157–166, 1994.

[91] D. Bahdanau, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

[92] Y. Hao, L. Dong, F. Wei, and K. Xu, "Self-attention attribution: Interpreting information interactions inside transformer," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 12963–12971, 2021.

[93] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., "Language models are few-

shot learners," Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, 2020.

[94] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., "Language models are unsupervised multitask learners," OpenAI blog, vol. 1, no. 8, p. 9, 2019.

[95] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," Advances in neural information processing systems, vol. 13, 2000.

[96] C. Parsing, "Speech and language processing," Power Point Slides, 2009.

[97] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Computer Speech & Language, vol. 13, no. 4, pp. 359–394, 1999.

[98] J. Bellegarda, "Exploiting latent semantic information in statistical language modeling," Proceedings of the IEEE, vol. 88, pp. 1279–1296, 2000.

[99] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[100] 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. 1532–1543, 2014.

[101] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model.," in Interspeech, vol. 2, pp. 1045–1048, Makuhari, 2010.

[102] N. Garg, L. Schiebinger, D. Jurafsky, and J. Y. Zou, "Word embeddings quantify 100 years of gender and ethnic stereotypes," Proceedings of the National Academy of Sciences, vol. 115, pp. E3635 – E3644, 2017.

[103] P. Lauren, G. Qu, J. Yang, P. Watta, G. Huang, and A. Lendasse, "Generating word embeddings from an extreme learning machine for sentiment analysis and sequence labeling tasks," Cognitive Computation, vol. 10, pp. 625 – 638, 2018.

[104] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, A. Gonzalez-Agirre, C. Armentano-Oller, C. Rodriguez-Penagos, and M. Villegas, "Spanish language models," arXiv preprint arXiv:2107.07253, 2021.

[105] 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," arXiv preprint arXiv:1907.11692, 2019.

[106] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, C. Armentano-Oller, C. Rodriguez-Penagos, A. Gonzalez-Agirre, and M. Villegas, "Maria: Spanish language models," Procesamiento del Lenguaje Natural, vol. 68, pp. 39–60, 2022.

[107] K. Xue, Y. Zhou, Z. Ma, T. Ruan, H. Zhang, and P. He, "Fine-tuning bert for joint entity and relation extraction in chinese medical text," in 2019 IEEE international conference on bioinformatics and biomedicine (BIBM), pp. 892–897, IEEE, 2019.

[108] X. Zhang, F. Wei, and M. Zhou, "Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5059–5069, 2019.

[109] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," arXiv preprint arXiv:1910.13461, 2019.

[110] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," Journal of Machine Learning Research, vol. 21, pp. 1–67, 2020.

[111] Y. Liu, "Multilingual denoising pre-training for neural machine translation," arXiv preprint arXiv:2001.08210, 2020.

[112] A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary, et al., "Beyond english-centric multilingual machine translation," Journal of Machine Learning Research, vol. 22, pp. 1–48, 2021.

[113] M. E. Jordan and R. R. McDaniel Jr, "Managing uncertainty during collaborative problem solving in elementary school teams: The role of peer influence in robotics engineering activity," Journal of the Learning Sciences, vol. 23, no. 4, pp. 490–536, 2014.

[114] R. A. Beghetto, "Uncertainty," in The Palgrave Encyclopedia of the possible, pp. 1691–1697, Springer, 2023.

[115] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," Advances in neural information processing systems, vol. 30, 2017.

[116] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in international conference on machine learning, pp. 1050–1059, PMLR, 2016.

[117] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," Advances in neural information processing systems, vol. 32, 2019.

[118] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," Machine Learning, vol. 110, pp. 457–506, 2021.

[119] R.-R. Griffiths, M. Garcia-Ortegon, A. A. Aldrick, and A. Lee, "Achieving robustness to aleatoric uncertainty with heteroscedastic bayesian optimisation," Machine Learning: Science and Technology, vol. 3, 2019.

[120] A. Der Kiureghian and O. Ditlevsen, "Aleatory or epistemic? does it matter?," Structural safety, vol. 31, no. 2, pp. 105–112, 2009.

[121] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," Advances in neural information processing systems, vol. 30, 2017.

[122] A. Malinin, S. Chervontsev, I. Provilkov, and M. Gales, "Regression prior networks," arXiv preprint arXiv:2006.11590, 2020.

[123] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," International journal of computer vision, vol. 40, pp. 99–121, 2000.

[124] O. Pele and M. Werman, "Fast and robust earth mover's distances," in 2009 IEEE 12th international conference on computer vision, pp. 460–467, IEEE, 2009.

[125] C. E. Shannon, "A mathematical theory of communication," The Bell system technical journal, vol. 27, no. 3, pp. 379–423, 1948.

[126] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," arXiv e-prints, pp. arXiv–1701, 2017.

[127] H. Yao, D.-l. Zhu, B. Jiang, and P. Yu, "Negative log likelihood ratio loss for deep neural network classification," in Proceedings of the Future Technologies Conference (FTC) 2019: Volume 1, pp. 276–282, Springer, 2020.

[128] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," Journal of the American statistical Association, vol. 102, no. 477, pp. 359–378, 2007.

[129] P. Smyth, "Model selection for probabilistic clustering using cross-validated likelihood," Statistics and computing, vol. 10, no. 1, pp. 63–72, 2000.

[130] S. Wu and M. Dredze, "Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), (Hong Kong, China), pp. 833–844, Association for Computational Linguistics, Nov. 2019.

[131] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., "Transformers: State-of-the-art natural language processing," in Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, pp. 38–45, 2020.

[132] S. Wager, S. Wang, and P. S. Liang, "Dropout training as adaptive regularization," Advances in neural information processing systems, vol. 26, 2013.

[133] F. Verdoja and V. Kyrki, "Notes on the behavior of mc dropout," arXiv preprint arXiv:2008.02627, 2020.

[134] F. Nielsen and S. Boltz, "The burbea-rao and bhattacharyya centroids," IEEE Transactions on Information Theory, vol. 57, no. 8, pp. 5455–5466, 2011.

[135] H. Saggion, S. Štajner, S. Bott, S. Mille, L. Rello, and B. Drndarevic, "Making it simplext: Implementation and evaluation of a text simplification system for spanish," ACM Transactions on Accessible Computing (TACCESS), vol. 6, no. 4, pp. 1–36, 2015.

[136] A. Palmero Aprosio, S. Tonelli, M. Turchi, M. Negri, and A. Di Gangi Mattia, "Neural text simplification in low-resource conditions using weak supervision," in Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation (NeuralGen), pp. 37–44, Association for Computational Linguistics (ACL), 2019.

[137] G. Torres Salazar and R. Ramos Arriagada, "Manual de finanzas personales y de familia. cómo usar bien mi dinero y el tuyo," 2020.

[138] B.-C. Red Financiera and S. CREDOMATIC, "Libro maestro de educación financiera un sistema para vivir mejor," RF BAC-CREDOMATIC. San Jose, Costa rica: Innova Technology SA, 2008.

[139] J. Izaguirre Olmedo, I. M. Carhuancho Mendoza, and D. Silva Siu, Finanzas para no financieros. GUAYAQUIL/UIDE/2020, 2020.

[140] N. E. for Financial Education, Tus gastos, tus ahorros, tu futuro: Guía de preparación financiera para principiantes. 2017.

[141] D. M. Endres and J. E. Schindelin, "A new metric for probability distributions," IEEE Transactions on Information theory, vol. 49, no. 7, pp. 1858–1860, 2003.

[142] J. Lin, "Divergence measures based on the shannon entropy," IEEE Transactions on Information theory, vol. 37, no. 1, pp. 145–151, 1991.

[143] T. Elguebaly and N. Bouguila, "Simultaneous high-dimensional clustering and feature selection using asymmetric gaussian mixture models," Image Vis. Comput., vol. 34, pp. 27–41, 2015.

[144] T. Iwata and A. Kumagai, "Meta-learning for out-of-distribution detection via density estimation in latent space," ArXiv, vol. abs/2206.09543, 2022.

[145] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," arXiv preprint arXiv:1510.00149, 2015.

[146] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," Journal of Machine Learning Research, vol. 18, no. 187, pp. 1–30, 2018.